

# UC Berkeley

## UC Berkeley Electronic Theses and Dissertations

### Title

Towards Understanding and Improving Large Language Model Reasoning

### Permalink

<https://escholarship.org/uc/item/0x8697p0>

### ISBN

9798189271724

### Author

Zhu, Hanlin

### Publication Date

2026-05-01

Peer reviewed|Thesis/dissertation

Towards Understanding and Improving Large Language Model Reasoning

by

Hanlin Zhu

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Stuart Russell, Co-chair

Assistant Professor Jiantao Jiao, Co-chair

Assistant Professor Song Mei

Spring 2026

# Towards Understanding and Improving Large Language Model Reasoning

Copyright 2026

by

Hanlin Zhu

# Abstract

Towards Understanding and Improving Large Language Model Reasoning

by

Hanlin Zhu

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Stuart Russell, Co-chair

Assistant Professor Jiantao Jiao, Co-chair

This dissertation focuses on developing systematic frameworks to understand and improve the reasoning capabilities of large language models (LLMs) in two axes: reliability and efficiency.

In terms of reliability, we study how LLMs learn parametric knowledge during training, and how they combine separate atomic knowledge to deduce new conclusions during test time. We develop theoretical frameworks and use out-of-context reasoning as a concrete lens to analyze model behavior, providing both theoretical explanations and empirical evidence that LLMs can hallucinate or fail to generalize systematically.

In terms of efficiency, we develop novel paradigms for test-time scaling to improve LLMs' capabilities and reduce inference cost. We focus on latent space reasoning, particularly chain-of-continuous-thought, demonstrating its theoretical advantages where continuous thoughts can maintain a superposition of multiple solutions and thus enable implicit parallel thinking. We also study how this mechanism emerges during training without explicit supervision.

Together, the results in this thesis provide theoretical foundations for understanding and improving LLM reasoning, highlighting the importance of models' reliability and efficiency, especially when applying those models to important fields and challenging tasks.

To my wife, Kunhe Yang

# Contents

|                                                                         |           |
|-------------------------------------------------------------------------|-----------|
| <b>Contents</b>                                                         | <b>ii</b> |
| <b>List of Figures</b>                                                  | <b>iv</b> |
| <b>List of Tables</b>                                                   | <b>ix</b> |
| <b>1 Introductions</b>                                                  | <b>1</b>  |
| 1.1 Reliability of LLMs: Out-of-context Reasoning . . . . .             | 1         |
| 1.2 Efficiency of LLM Reasoning: Latent Space Reasoning . . . . .       | 2         |
| 1.3 Organization of the Thesis . . . . .                                | 3         |
| 1.4 Related Works . . . . .                                             | 3         |
| <b>2 LLMs Fail in Generalization: the Reversal Curse</b>                | <b>7</b>  |
| 2.1 Introductions . . . . .                                             | 7         |
| 2.2 Preliminaries . . . . .                                             | 9         |
| 2.3 Bilinear Models . . . . .                                           | 10        |
| 2.4 One-Layer Transformers . . . . .                                    | 11        |
| 2.5 Experiments . . . . .                                               | 16        |
| 2.6 Conclusions . . . . .                                               | 18        |
| <b>3 Understanding Out-of-context Reasoning in LLMs</b>                 | <b>19</b> |
| 3.1 Introduction . . . . .                                              | 19        |
| 3.2 OCR in LLMs . . . . .                                               | 21        |
| 3.3 One-Layer Attention-Only Transformers can Do Symbolic OCR . . . . . | 22        |
| 3.4 Theoretical Results . . . . .                                       | 26        |
| 3.5 Conclusions . . . . .                                               | 29        |
| <b>4 Reasoning by Superposition via Continuous CoT</b>                  | <b>30</b> |
| 4.1 Introductions . . . . .                                             | 30        |
| 4.2 Preliminaries . . . . .                                             | 31        |
| 4.3 Problem Formulations . . . . .                                      | 33        |
| 4.4 Theoretical Results . . . . .                                       | 34        |
| 4.5 Experiments . . . . .                                               | 38        |

|          |                                                         |            |
|----------|---------------------------------------------------------|------------|
| 4.6      | Conclusions . . . . .                                   | 41         |
| <b>5</b> | <b>Emergence of Superposition in Continuous CoT</b>     | <b>42</b>  |
| 5.1      | Introductions . . . . .                                 | 42         |
| 5.2      | Problem Formulation . . . . .                           | 43         |
| 5.3      | Analysis of the Thought Generation Stage . . . . .      | 45         |
| 5.4      | Analysis of the Prediction Stage . . . . .              | 50         |
| 5.5      | Experiments . . . . .                                   | 52         |
| 5.6      | Conclusions . . . . .                                   | 54         |
|          | <b>Bibliography</b>                                     | <b>55</b>  |
| <b>A</b> | <b>Appendix for the Reversal Curse Theory</b>           | <b>67</b>  |
| A.1      | Additional Notations . . . . .                          | 67         |
| A.2      | Missing Proofs of Section 2.3 . . . . .                 | 67         |
| A.3      | Missing Proofs of Section 2.4 . . . . .                 | 79         |
| A.4      | Experiments for chain-of-thought . . . . .              | 86         |
| A.5      | Additional Experimental Results . . . . .               | 88         |
| <b>B</b> | <b>Appendix for Out-of-context Reasoning</b>            | <b>98</b>  |
| B.1      | Proof of Section 3.4.1 . . . . .                        | 98         |
| B.2      | Proof of Section 3.4.2 . . . . .                        | 118        |
| B.3      | Additional Experiments for One-layer Models . . . . .   | 131        |
| B.4      | Additional LLM Experiments on Real-World Data . . . . . | 133        |
| B.5      | Implementation Details . . . . .                        | 135        |
| <b>C</b> | <b>Appendix for Chain-of-continuous-thought Theory</b>  | <b>139</b> |
| C.1      | Notation Details . . . . .                              | 139        |
| C.2      | Missing Proofs . . . . .                                | 140        |
| C.3      | Experiment Details . . . . .                            | 156        |
| <b>D</b> | <b>Appendix for Training Dynamics of Continuous CoT</b> | <b>158</b> |
| D.1      | Notation Details . . . . .                              | 158        |
| D.2      | Missing Proofs for Section 5.3 . . . . .                | 159        |
| D.3      | Missing Proofs for Section 5.4 . . . . .                | 173        |
| D.4      | Auxiliary Lemmas . . . . .                              | 177        |
| D.5      | Experiment Details . . . . .                            | 178        |
| D.6      | Additional Experimental Results . . . . .               | 179        |

# List of Figures

- 2.1 Experiment results of reversal curse under default configuration (see Table A.2). The curves represent the (average) negative log probability of the model predicting the next token to be  $B_i$  when the input is " $A_i \rightarrow$ ", or to be  $A_i$  when the input is " $B_i \leftarrow$ ". While the sentences in the training set can be learned nearly perfectly (as shown by the training curve where the next token probability converges to one), the model is not able to predict the correct next token in the validation set better than a uniformly random guess. Both curves are averaged over 10 random seeds. 16
- 2.2 Visualization of the weights (logits) of the model with default configurations trained after 3000 epochs for the reversal curse experiment. For the top-left matrix, the  $i$ -th row corresponds to an entity token  $A_i$  for a training pair, and the  $i$ -th column corresponds to an entity token  $B_i$  for a training pair. The  $(i, j)$ -th entry represents the model weights from the token  $A_i$  to  $B_j$ , i.e., the logits of  $B_j$  when the input sequence consists of only  $A_i$ . Similarly, for the bottom-left matrix, the row corresponds to the input entity tokens of the seen direction (the direction included in the training set) of validation pairs, and the column corresponds to output entity tokens. The two matrices on the right are obtained by swapping row tokens and column tokens of their corresponding left matrices. Note that the diagonals of the bottom-right matrix are all close to zero, while the diagonals of other matrices all have large values. This implies that if a pair of tokens  $(A, B)$  only appear in the training set in one direction, then the model weights associated with the other direction will hardly get trained. . . . . 17
- 3.1 **Illustration of the symbolic out-of-context reasoning (OCR) task.** *Left:* The task is motivated by real-world knowledge injection, where  $\mathcal{S}$  corresponds to names and  $\mathcal{A}_1 = \{b_i\}_{i=1}^n, \mathcal{A}_2 = \{c_i\}_{i=1}^n$  denote collections of cities and languages, respectively. *Middle:* We tokenize entities into symbolic sequences. *Right:* The mapping rule connects  $\mathcal{S}, \mathcal{A}_1$ , and  $\mathcal{A}_2$ , where each  $s \in \mathcal{S}_i$  associates with a unique fact  $b_i \in \mathcal{A}_1$  and corresponding implication  $c_i \in \mathcal{A}_2$ . . . . . 21

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |    |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 3.2 | <b>The weights and mechanisms of the trained one-layer attention models.</b><br>The heatmaps on the left show that the factorized model ( <i>bottom</i> ) learns a structured weight matrix that enables OCR, as highlighted by the red box. The non-factorized model ( <i>top</i> ) fails to learn this structure. Here, the weights shown are the partial weights in the output-value matrix related to the prediction, i.e., we show a reduced matrix $\mathbf{W}_{\text{OV}} \in \mathbb{R}^{ \mathcal{A}  \times (mn+2)}$ . The diagram on the right illustrates how this structural difference leads to different outcomes. The task is to predict $c_2 \in \mathcal{A}_2$ given input $z_{1:T}$ with $(s_2, r_2)$ , where the atomic knowledge $(s_2, r_2, c_2)$ is not included in the training set. . . . . | 25 |
| 4.1 | Prompt format of the graph reachability problem. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 33 |
| 4.2 | Illustration of the embedding space and first layer attention mechanism. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 36 |
| 4.3 | Illustration of the second layer attention mechanism for thought generation. We omit the positional encoding space since they are not used in the second layer. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 37 |
| 4.4 | The overall accuracy of COCONUT, CoT, CoT*(12 layers, $n_{\text{heads}} = 12$ ) and No CoT. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 38 |
| 4.5 | A representative example of Layer 1 attention map: the edge token $\langle \mathbf{e} \rangle$ ( $y$ -axis) places nearly all its attention mass on its source and target nodes ( $x$ -axis), in line with the theoretical construction. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 39 |
| 4.6 | The histogram of inner product between the $i$ -th continuous thoughts and the node embeddings. The mean value for each group is shown in the legend. Note that <i>Frontier</i> is a subset of <i>Reachable</i> nodes, and <i>Optimal</i> is a subset of <i>Frontier</i> nodes. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 40 |
| 5.1 | Prompt format of the graph reachability problem (Figure 1 of [169]). . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 45 |
| 5.2 | Pictorial illustration of the superposition mechanism and the index-matching logit $\mu$ . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 46 |
| 5.3 | The attention logits difference between frontier edges and others under $\mathcal{L}^{\text{coco}}$ as a proxy for $\mu_v$ . The background colors indicate different training stages. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 53 |
| 5.4 | Training dynamics of the proxies for $\mu_A$ (residual carryover) and $\mu_R$ (candidate lift). . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 54 |
| A.1 | Experiment results of CoT under default configuration (see Table A.2). The curves represent the (average) negative log probability of the model predicting the next token to be: (1) $\mathbf{B}_i$ given the input “ $\mathbf{A}_i \rightarrow$ ”, (2) $\mathbf{C}_i$ given the input “ $\mathbf{B}_i \rightarrow$ ”, or (3) $\mathbf{C}_i$ given the input “ $\mathbf{A}_i \rightsquigarrow$ ”. Similar to the reversal curse experiment, while the sentences in the training set can be learned nearly perfectly, the model is not able to predict the correct next token in the validation set better than a uniformly random guess. Both curves are averaged over 10 random seeds. . . . .                                                                                                                      | 86 |
| A.2 | Visualization of the weights (logits) of the model with default configurations trained after 3000 epochs for CoT experiment. The matrices are similar to Figure 2.2. The row tokens for the top matrices are $\mathbf{A}_i$ , $\mathbf{B}_i$ , $\mathbf{A}_i$ and column tokens are $\mathbf{B}_i$ , $\mathbf{C}_i$ , $\mathbf{C}_i$ for training triples respectively. Similarly, the bottom matrices correspond to validation triples. For validation triples $(\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i)$ , the weights from $\mathbf{A}_i$ to $\mathbf{C}_i$ get hardly trained as indicated by the diagonals of the last matrix. . . . .                                                                                                                                                                        | 87 |

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |    |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| A.3  | Results for reversal curse for different vocabulary sizes. All other configurations are set as default values as in Table A.2. The training set sizes for the above four experiments are 9, 20, 85, 850 respectively, and the validation set sizes are 1, 4, 15, 150 respectively. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 89 |
| A.4  | Results for reversal curse for different numbers of layers of the transformer. All other configurations are set as default values as in Table A.2. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 90 |
| A.5  | Results for reversal curse with no positional encoding or relative positional encoding. For relative positional encoding, we follow the Rotary Position Embedding (RoPE) method proposed by [123]. We use the implementation of <a href="#">this repo</a> , MIT license. All other configurations are set as default values as in Table A.2. . . . .                                                                                                                                                                                                                                                                                                                                                                                            | 90 |
| A.6  | Results for reversal curse with different entity lengths. Each entity $A_i$ or $B_i$ consists of multiple tokens and different entities may have overlapped tokens. The “Train” curve represents the negative log probability of predicting the first token of the output entity, and “Train (word)” represents the negative log probability of predicting all tokens one by one of the output entity. All other configurations are set as default values as in Table A.2. The training set sizes for the above two experiments are 680 and 250, respectively, and the validation set sizes are 120 and 50, respectively. . . . .                                                                                                               | 91 |
| A.7  | Results for reversal curse with fixed token embedding and fixed positional embedding. All other configurations are set as default values as in Table A.2. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 91 |
| A.8  | Results for reversal curse for different numbers of layers and different embedding dimensions. All other configurations are set as default values as in Table A.2. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 92 |
| A.9  | A heat map of cosine similarity between token embeddings (all $A_i$ and $B_i$ ) after 3000 epochs training under different settings. The settings are the same as Figure A.8. Under different numbers of layers or embedding dimensions, most of the non-diagonal entries are close to 0, which shows that the embeddings of different tokens are nearly orthogonal. . . . .                                                                                                                                                                                                                                                                                                                                                                    | 93 |
| A.10 | Training and validation loss under in-context learning (ICL) settings. All sentences consist of seven tokens and have the form of “ $A_i R B_j \Leftrightarrow B_j R^{-1} A_i$ ”. The loss is calculated on the last token. For the left figure, the training set size is 700, the validation set size is 300, and the vocabulary size is 400; for the right figure, the training set size is 1400, the validation set size is 600, and the vocabulary size is 800. All other configurations are set as default values as in Table A.2. The result shows that the reversal curse does not happen in ICL settings, i.e., if “ $A_i R B_j$ ” is provided as part of the prompt, then the model is able to recognize “ $B_j R^{-1} A_i$ ”. . . . . | 94 |
| A.11 | Results for CoT for different vocabulary sizes. All other configurations are set as default values as in Table A.2. The training set sizes for the above four experiments are 14, 32, 135, 1350 respectively, and the validation set sizes are 1, 4, 15, 150 respectively. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 95 |
| A.12 | Results for CoT for different number of layers of the transformer. All other configurations are set as default values as in Table A.2. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 95 |

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |     |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| A.13 | Results for CoT with no positional encoding or relative positional encoding. For relative positional encoding, we follow the Rotary Position Embedding (RoPE) method proposed by [123]. All other configurations are set as default values as in Table A.2. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 96  |
| A.14 | Results for CoT with different entity lengths. The setting and curves are similar to Figure A.6. All other configurations are set as default values as in Table A.2. The training set sizes for the above two experiments are 1080 and 400, respectively, and the validation set sizes are 120 and 50. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 96  |
| A.15 | Results for CoT with fixed token embedding and fixed positional embedding. All other configurations are set as default values as in Table A.2. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 97  |
| A.16 | Results for CoT where each entity is represented by two tokens, i.e., $A_i$ , $B_i$ , or $C_i$ . The validation set sizes are 50. The model is able to “deduce” unseen $A_i \rightsquigarrow C_i$ by learning underlying patterns. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 97  |
| B.1  | <b>Training and Test Implication Loss for Factorized vs. Non-Factorized Models.</b> While both models effectively minimize the training loss ( <i>left</i> ), their performance on unseen test implications differs starkly ( <i>right</i> ). The factorized model successfully generalizes, achieving low test implication loss and thus demonstrating OCR, while the non-factorized model fails to generalize. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 132 |
| B.2  | <b>Comparison of full weights of trained one-layer linear attention models.</b> <i>Left:</i> Non-factorized model. <i>Right:</i> Factorized model. The factorized model shows strong OCR capability compared to the non-factorized model. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 132 |
| B.3  | <b>Comparison of solutions to <math>(\mathbf{W}_{OV}\text{-SVM})</math> and <math>(\mathbf{W}_{OV}^F\text{-SVM})</math>.</b> <i>Top Left:</i> $(\mathbf{W}_{OV}\text{-SVM})$ with the Frobenius norm objective. <i>Bottom Left:</i> $(\mathbf{W}_{OV}^F\text{-SVM})$ with the nuclear norm objective. Here we only show the partial weights in the output-value matrix related to the prediction, i.e., $\mathbf{W}_{OV} \in \mathbb{R}^{ \mathcal{A}  \times (mn+2)}$ . <i>Right:</i> Geometric interpretation of $\mathbf{W}_O$ and $\mathbf{W}_V$ solved in $(\mathbf{W}_{OV}^F\text{-SVM})$ . All the subjects’ feature vectors (corresponding to rows in $\mathbf{W}_V$ ) reside in the $xy$ plane while the relation vectors corresponding to $r_1$ and $r_2$ are orthogonal to the subjects and point in opposite directions. The predictions $\widehat{\mathbf{W}}_O(b_i)$ , $\widehat{\mathbf{W}}_O(c_i)$ are made by summing up the feature vector of $s_i$ with $r_1$ or $r_2$ , which aligns well with the features of $b_i$ or $c_i$ respectively (plotted in the figure, which are corresponding rows in $\mathbf{W}_O$ ) with cosine similarity greater than 0.9. . . . . | 134 |
| B.4  | Test loss versus training steps for a one-layer linear attention model with varying intrinsic dimensions ( $d_h$ ). The plot demonstrates that the model can exhibit OCR even when the intrinsic dimension is as small as $d_h = 4$ . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 135 |
| B.5  | OCR performance of various LLMs on the five relation pairs. Results are averaged over 3 random seeds, with shaded regions representing the standard deviation. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 136 |
| C.1  | Illustration of the second layer attention mechanism for final prediction. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 148 |

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |     |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| D.1 | The attention logits difference between frontier edges and others. The model is trained with a modified version of $\mathcal{L}^{\text{BFS}}$ . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 179 |
| D.2 | Example attention maps illustrating three alternative routes for <i>candidate lift</i> . For clarity, we omit earlier tokens in the sequence and only visualize the final segment containing some of edges, the candidate nodes, latent thoughts, and answer tokens. <b>Pattern A</b> (consistent with the theoretical assumption): $\langle \mathbf{R} \rangle$ copies candidate nodes in Layer 1, and $\langle \mathbf{A} \rangle$ attends to $\langle \mathbf{R} \rangle$ in Layer 2. <b>Pattern B</b> : $\langle \mathbf{A} \rangle$ directly attends to candidate nodes in Layer 1. <b>Pattern C</b> : continuous thoughts copy candidate nodes in Layer 1, and $\langle \mathbf{A} \rangle$ attends to the continuous thoughts in Layer 2. All three patterns achieve the same functional effect of lifting the reachable candidate. . . . . | 180 |
| D.3 | The first-layer attention patterns in 4-layer transformers. $\langle \mathbf{e} \rangle$ tokens attend to the corresponding source and target nodes to aggregate the information. This is consistent with the analysis of the two-layer transformer in [169]. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 182 |
| D.4 | Inner product between layer-wise hidden states and different types of nodes in a 4-layer transformer. The experimental setting follows [169]. From the second layer onward, hidden states exhibit larger inner products with reachable, frontier, and optimal nodes, indicating that superpositional representations emerge as early as layer 2 in the 4-layer transformer. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 183 |
| D.5 | Accuracy curve during the answer-prediction stage. The accuracy shows a rapid improvement corresponding to the learning of residual carryover and candidate lift signals. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 184 |

# List of Tables

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |     |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 2.1 | Notations for tokens in Section 2.4. “ $\rightarrow$ ” and “ $\leftarrow$ ” denote forward and backward relationships for the reversal curse. “ $\rightarrow$ ” and “ $\rightsquigarrow$ ” denote direct and indirect implication for CoT. $R_1$ and $R_2$ are relationship tokens in Section 2.4.3. $A$ , $B$ , $C$ , $A_i$ , $B_i$ , $C_i$ denote tokens representing entities. . . . .                                                                                                                         | 9   |
| 3.1 | Performance comparison of different language models on synthetic reasoning tasks with various associations. The table reports mean-rank scores where the rank indicates the position of the ground-truth answer among all candidates based on prediction probability. Lower ranks indicate better performance and Rank 0 refers to the token with the largest probability. Values in parentheses indicate the standard error of the mean-rank scores, calculated from 3 runs with different random seeds. . . . . | 23  |
| 4.1 | Layer 2 attention scores to different edge groups at step $i$ (mean $\pm$ standard deviation). . . . .                                                                                                                                                                                                                                                                                                                                                                                                            | 41  |
| A.1 | Full list of hyperparameters for AdamW optimizer and training. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                            | 88  |
| A.2 | The list of different configurations for experiments in Sections A.5.2 and A.5.3. Default choices are boldened for each row. . . . .                                                                                                                                                                                                                                                                                                                                                                              | 89  |
| B.1 | Performance comparison of different language models on PopQA-OCR. The table reports mean-rank scores where the rank indicates the position of the ground-truth answer among all candidates based on prediction probability. Lower ranks indicate better performance and Rank 0 refers to the token with the largest probability. Values in parentheses indicate the standard error of the mean-rank scores, calculated from 3 runs with different random seeds. . . . .                                           | 134 |
| B.2 | Relation expressions of different topics. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 137 |
| C.1 | Meaning of each token. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 139 |
| C.2 | Position indices of different tokens or continuous thoughts in the input sequence. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                        | 140 |
| C.3 | ProsQA statistics. Numbers are averaged over problem instances. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                           | 156 |
| C.4 | The inner products between $i$ -th continuous thoughts and nodes (3 runs with different random seeds). . . . .                                                                                                                                                                                                                                                                                                                                                                                                    | 156 |
| D.1 | Meaning of each token (Table 2 in [169]). . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 158 |

|     |                                                                                                                                             |     |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------|-----|
| D.2 | Position indices of different tokens or continuous thoughts in the input sequence<br>(Table 3 in [169]). . . . .                            | 159 |
| D.3 | ProsQA statistics. Numbers are averaged over problem instances. . . . .                                                                     | 178 |
| D.4 | Ablation on depth, heads, width, learning rate, and weight tying. By default,<br>other hyperparameters follow the main experiments. . . . . | 181 |

## Acknowledgments

First and foremost, I would like to express my deepest gratitude to my PhD advisors, Professor Stuart Russell and Professor Jiantao Jiao.

I am profoundly grateful to Stuart for his vision, guidance, and inspiration. I deeply appreciate his long-term vision for artificial intelligence that we should not only strive to make AI systems more capable, but also think carefully about what happens if we succeed. This perspective has greatly shaped my own research interests, both in improving AI models and in understanding their behavior. I have always enjoyed our whiteboard discussions, even when it was challenging to convince him. His insightful questions and comments consistently pushed me to think more deeply and carefully.

I would also like to thank Jiantao for his guidance, support, and mentorship throughout my PhD journey. When I was a junior PhD student, he helped me work through technical details and provided concrete guidance on research projects. As I grew as a more independent researcher, he offered invaluable advice on broader research directions and long-term development. I am also deeply grateful for his encouragement and support during my job search. His advice and perspective have had a lasting influence on both my research and my personal growth.

I am grateful to Professor Song Mei for serving on my dissertation committee. I especially enjoyed working with Song, and I sincerely appreciate his guidance, perspective, and support in both research and my career path.

I would also like to thank my mentors during my industry internships: Xin Yang, Amy Zhang, Yuandong Tian, and Venkat Srinivasan. In particular, I would like to thank Yuandong, who has been my mentor for almost two years and whose collaboration with me has continued even after my internship ended. I have truly enjoyed working with him. My industry experience has been extremely valuable, giving me new perspectives and helping me think about research problems from different angles.

I am thankful to my labmates in both Jiaolab and CHAI. In particular, I want to thank Baihe Huang, Banghua Zhu, Paria Rashidinejad, Geng Zhao, Tianhao Wu, Hengyu Fu, Druv Pai, Haozhe Jiang, Cassidy Laidlaw, Mark Bedaywi, Li Dayan and Niklas Lauffer. I have greatly enjoyed our discussions, collaborations, and time spent together. Their companionship made my PhD years much more meaningful and enjoyable.

I would like to thank all of my collaborators for their wonderful work and support. In particular, I am grateful to my close collaborators Baihe Huang, Tianyu Guo, Yixiao Huang, Shibo Hao, Banghua Zhu, and Paria Rashidinejad.

I also want to thank my friends Eric, Brian, Nivasini, Yaodong, Ruiqi, Licong, Xinyan, Zixuan, and Ziheng, as well as my neighbors Geng, Zoey, Xin, and Xinwei. Because of them, my PhD journey has not only been about research, but has also been filled with joy, fun, and many wonderful memories.

I would like to express my heartfelt gratitude to my parents, Jiannan Zhu and Hu Wu, for their unconditional love and support. No matter what choices I have made, they have always supported me and encouraged me to pursue my own path.

I want to thank my cats, Lily and Azalea. With them, life is never boring. Understanding their behavior is always exciting and challenging, and is no easier than understanding the behavior of LLMs.

Finally, I would like to express my deepest gratitude to my wife, Kunhe Yang. I feel extremely fortunate to have shared this journey with the person I love and admire most in the world. Kunhe has always supported me, and with her by my side, my PhD journey has become a truly wonderful adventure. I am deeply grateful for her love, companionship, encouragement, and unwavering support. I am excited to begin the next chapter of our lives together.

# Chapter 1

## Introductions

Large language models (LLMs) have shown remarkable success in various challenging tasks, including mathematical reasoning [43, 63, 101, 130, 49], coding [26, 23, 60, 171, 112], multi-modal tasks [4, 129, 61, 157], agentic tasks [6, 7, 107, 102, 168], etc. Despite this rapid and promising progress of LLMs, we still face many challenges to further develop and scale up more intelligent artificial intelligence (AI) systems to solve more challenging tasks in more important fields. This thesis mainly focuses on developing systematic frameworks for understanding and improving the reasoning capability of LLMs along two axes: *reliability* and *efficiency*.

### 1.1 Reliability of LLMs: Out-of-context Reasoning

Although today’s LLMs can handle a wide range of complex tasks, a growing body of evidence documents their fundamental failures on simple tasks [11, 5, 152, 16, 9, 12, 69].

For example, when an autoregressive LLM learned factual knowledge during training, it can correctly recall the fact during test time if the fact is presented in the same narrative order, while it can totally fail if the test question is presented in the reverse order, which is known as the reversal curse [11]. This simple failure example demonstrates that autoregressive LLMs can fail on extremely simple tasks in a systematic way: they tend to memorize training data at the sentence level rather than extract higher-level rules from it under certain scenarios.

Another failure mode of LLMs is post-knowledge-injection hallucination [12, 69]. For example, previous work observed that after a model is fine-tuned on data containing 19th-century knowledge, when asked about modern events, the model will respond with inventions from the 19th century, such as the electrical telegraph [12]. Similarly, prior work observed that a model fine-tuned on biographies of unfamiliar individuals tends to hallucinate fabricated personal information about a new person during test [69]. These examples show that while fine-tuning is a general method for increasing model capability, it can also introduce hallucinations, making the model unreliable.

As LLMs are increasingly deployed to important fields and challenging tasks, including

healthcare and medicine [118, 131, 98, 119], biology and chemistry [81, 35, 87, 18], scientific research [111, 151, 128], software engineering [66, 150, 104], etc., it is important to make sure they are reliably strong, since unexpected failure modes, even in a simple step, might cause catastrophic consequences in critical fields.

This thesis aims to develop a theoretical framework to better understand the underlying mechanisms of model behavior. A better understanding of model behavior can enable better predictions of when LLMs will succeed or fail, and provide insights for designing better training and inference methods to improve models’ capabilities and reliability.

In particular, we focus on out-of-context reasoning (OCR) capability, which refers to a model’s capability to answer a question when key information is missing from the context but was learned during training, as a concrete lens for analyzing model behavior. Concretely, within our OCR framework, we provide theoretical explanations of the reversal curse in Chapter 2 and post-knowledge-injection hallucination in Chapter 3. These results provide a better understanding of how LLMs learn and store knowledge during training and use it at test time, which can help us better predict, control, and steer model behavior in the long run to ensure models are reliably strong.

## 1.2 Efficiency of LLM Reasoning: Latent Space Reasoning

An important aspect of LLMs is that their capabilities can be improved not only during training but also at test time, a paradigm known as test-time scaling [63]. By allocating more computational resources at test time, such as allowing models to think longer using chain-of-thought (CoT) [145], one can significantly improve LLMs’ performance on challenging reasoning benchmarks, and this improvement can sometimes be much more efficient compared to allocating the same amount of computation to the training time [63].

The most widely used test-time scaling method is CoT [145], in which LLMs are allowed to output intermediate thinking tokens before making the final prediction. Although CoT is a general method and effective in LLMs, especially in reasoning models, it also introduces new challenges.

First, it is common for reasoning models to produce long, verbose CoT even when solving simple tasks [27]. For example, even for trivial questions such as calculating  $2 + 3$ , many reasoning models tend to generate extremely long CoT, consuming thousands of tokens [27]. Due to the autoregressive nature of many mainstream LLMs, a verbose CoT can lead to high computational costs and generation latency, rendering the generation process costly and inefficient.

Second, long CoT can also cause hallucinations, since when the context is long, the model might get distracted by intermediate, irrelevant content and pay attention to the wrong tokens [125]. This would not only increase the cost of reasoning but also degrade model performance.

Therefore, it is important to efficiently scale test-time computing. In this thesis, we study a promising novel test-time scaling paradigm — latent space reasoning [44, 142, 28, 106, 122, 55, 169, 167, 45, 163, 22]. The key motivation of latent space reasoning is that the textual token space might not always be the most efficient for reasoning. It is also partially motivated by the fact that many texts serve primarily to maintain linguistic coherence, rather than conveying core reasoning information [122].

In this thesis, we focus on chain-of-continuous-thought (continuous CoT, or Coconut) [55], which is one of the most important methods for latent space reasoning. We provide theoretical guarantees for continuous CoT, showing its advantages in expressivity and revealing its underlying mechanism that continuous thought can encode a superposition of multiple possible solutions in Chapter 4. We also analyze how the mechanism emerges during training even without explicit supervision in Chapter 5.

Our analysis shows preliminary and promising results in latent space reasoning, which can provide useful insights for future work to further advance this area.

## 1.3 Organization of the Thesis

The remainder of the dissertation is organized as follows:

- **Section 1.4** discusses prior literature related to this thesis;
- **Chapter 2** studies how LLMs can fail to generalize in systematic ways through the lens of the reversal curse, providing theoretical explanations of why autoregressive LLMs tend to memorize facts at the sentence level rather than extracting higher-level rules from data via training dynamics;
- **Chapter 3** studies post-knowledge-injection hallucination through out-of-context reasoning, providing a unified framework explaining both generalization and hallucination after fine-tuning;
- **Chapter 4** provides the first theoretical guarantees for chain-of-continuous-thought, showing models can maintain a superposition of plausible reasoning traces to perform parallel thinking, which overcomes the drawbacks of traditional CoT that prematurely commits to a plausible but wrong answer;
- **Chapter 5** analyzes how the superposition mechanism in chain-of-continuous-thought emerges during training when there is no explicit supervision to guide the model to explore multiple branches simultaneously.

## 1.4 Related Works

**LLM Reasoning.** The strong performance of LLMs on reasoning tasks [21, 29, 57, 73, 67, 145, 54, 143] has prompted many studies on the reasoning capabilities of LLMs. [149] argues

that transformers perform implicit Bayesian inference in ICL. [100] shows that transformers implement a specific type of circuits called “induction heads” that are key to the ICL abilities of LLMs. [96] proves that causal structures are encoded in transformer layers during the training dynamics. [20] identifies a backward chaining mechanism of transformers in deductive reasoning. Apart from in-context reasoning, LLMs still demonstrate limitations in other types of reasoning tasks [135, 24, 160].

**Out-of-context reasoning.** Previous work study LLM’s out-of-context reasoning capability through many aspects, such as out-of-context meta-learning [74], situational awareness [10], knowledge manipulations [5], etc. While negative results are reported on LLM’s performance of certain OCR tasks, such as the reversal curse [11] and multi-hop in-weight reasoning [152, 16], recent work [39] shows that LLM can associate two events when several subjects are involved in both events in the training data. While [39] empirically analyzes the underlying mechanism, this dissertation theoretically analyzes how transformers learn this OCR task. Recently, [103] shows that there exists a linear transformation in the logits for predicting two related pieces of knowledge in LLMs, which can be used to gauge the model’s generalization/hallucination capability. Our results also echo previous empirical findings that LLMs tend to hallucinate when they learn new factual knowledge during fine-tuning [41, 69, 124]. Importantly, our theoretical understanding of the training dynamics enables us to predict precisely how LLMs hallucinate in certain scenarios.

**Expressivity of transformers.** There is a long line of work studying the expressivity of transformers [158, 13, 14, 105, 80, 153, 34, 3, 92]. A more recent line of work shows that chain-of-thought (CoT) can improve the expressivity of transformers [82, 37, 91, 79]. For example, [82] studies low-depth transformer expressivity for semi-automata, of which the setting corresponds to one CoT step. [37] shows that constant-depth transformers with CoT can solve certain P-complete problems. [79] further provides constructions of constant-depth transformer for each problem in P/poly with CoT. [91] studies the expressivity with different lengths of CoT, showing that logarithmic steps of CoT in input length can expand the upper bound of constant-depth transformer expressivity from  $TC^0$  to L, while a linear number of steps can further expand the upper bound to  $NC^1$ -complete. While these expressivity results mainly focus on discrete CoT, theoretical studies on continuous CoT [55] are rare, which this dissertation focuses on. [45] studies the expressivity of one-layer transformers with continuous CoT on the minimum non-negative sum problem, demonstrating the superposition mechanism in the arithmetic domain, which complements our results on the graph reachability problem. While their construction requires an exponentially large embedding dimension, our theoretical construction requires only a linear embedding dimension in the graph size. Moreover, unlike many previous works that construct problem-specific (or even length-specific) positional encodings, our construction applies to practical positional encodings, such as sinusoidal [138] and RoPE [123].

**Training dynamics of transformers.** Extensive research have investigated the optimization of transformer-based models [64, 15, 88, 40, 133, 132, 161, 76, 59, 52, 58]. In particular, recent works focus on understanding the transformer’s behavior on various reasoning tasks through the lens of training dynamics. For example, previous studies have explored the emergence of induction heads [19], factual recall [96], the reversal curse [170], chain-of-thought reasoning [146], and in-context two-hop reasoning [53]. Building on this, our theoretical analysis of a factual recall task shows that a one-layer transformer’s reasoning ability is significantly affected by the reparameterization of its value and output matrices. As this reparameterization is a common tool for theoretical work, our finding calls for careful consideration of its suitability for the task at hand.

**Implicit bias of gradient descent.** A rich line of literature has studied the implicit bias of gradient descent in classification tasks, which connects problems with logistic or exponentially-tailed loss to margin maximization [120, 47, 46, 86, 93, 94, 65, 136]. Building on foundational results from [86, 136], our work characterizes the solution to SVM programs to understand generalization and hallucination of LLMs, whereas most prior works only focus on the optimization landscape of neural networks. There are also many works exploring this connection in attention-based models [127, 126, 76, 62, 117, 137]. [127, 117] are the closest to our work and investigate a similar reparameterization for query and key matrices, where the gradient descent implicitly minimizes the nuclear norm of the combined weights. Recently, [162] demonstrates that the two parameterizations lead to distinct optimization trajectories in in-context learning (ICL): non-factorized models exhibit abrupt loss drops, whereas factorized models show stage-like dynamics performing incremental principal component regression. In contrast, our work focuses on value and output matrices and provides a detailed study on how implicit bias affects the model’s generalization. Our work also links to studies investigating out-of-distribution (OOD) generalization through the lens of implicit bias [2, 1]. Finally, our findings regarding the factorized model with  $(\mathbf{W}_O, \mathbf{W}_V)$ -parameterization are grounded in the extensive literature on the implicit bias of gradient descent on matrix factorization [48, 77, 8, 78, 108, 121].

**Reasoning with chain-of-thought.** Chain-of-thought (CoT) [145] is a simple yet effective test-time scaling method to enhance LLM’s reasoning capability. It can either be prompt-based only [70, 165] or be included in the training sample to create high-quality training data [156, 155, 141, 116]. Beyond empirical study, many theoretical works also explore the advantages of the CoT method. For example, [82, 37, 91, 79] shows that CoT can improve the expressivity of transformers. [170] studies the importance of CoT for two-hop reasoning via training dynamics. [146, 71] studies how CoT in the training data can improve the sample efficiency of transformers. Instead of the text-based CoT, this dissertation studies continuous CoT where the “thinking tokens” lie in a latent continuous space and do not need to be converted to discrete tokens.

**Latent space reasoning.** A recent line of work studies latent space reasoning, a novel paradigm beyond text-based CoT [44, 142, 106, 122, 55]. For example, [44] proposed to use pause tokens, which are learnable tokens that are inserted into the original text to increase the computation space. Later, [83] theoretically shows that the pause token can strictly increase the expressivity of the transformer. Similarly, [106] studies filler tokens, which also increase the computation space of LLMs. [142] proposed to use planning tokens at the beginning of the response generation to improve the reasoning capability. [122] proposed to use abstract tokens in a latent space to enhance the reasoning performance while reducing the inference cost. The most related work is [55], which proposes to use continuous CoT for reasoning.

**Reasoning as graph problems.** Graph problems are essential to understand LLM reasoning capability since many reasoning problems can be abstracted as computational graphs [154, 166] where relational data fed into transformers can be modeled as edges [139, 143, 53]. Many previous works have shown that pretrained LLMs can deal with reasoning tasks in graphs, but may still have difficulties with more complicated tasks [140, 50, 36, 115, 84, 32, 30]. Other works study how transformers solve classic and fundamental graph problems, such as graph reachability [91, 90], shortest path [30], etc. For example, [30] shows that a two-layer transformer can leverage spectral decomposition of the line graph to predict the shortest path on small-scale undirected graphs. [90] shows that a log-depth transformer can solve directed graph reachability, which constant-depth transformers can not solve. For a constant-depth transformer, [91] shows directed graph reachability can be solved with  $O(n^2)$  CoT steps where  $n$  is the number of vertices, while it remains unclear whether a smaller number of discrete CoT steps can solve the task. On the contrary, our work shows that a two-layer transformer can solve graph reachability for a  $D$ -diameter graph with  $D$  continuous thought steps.

## Chapter 2

# LLMs Fail in Generalization: the Reversal Curse

### 2.1 Introductions

Large language models (LLMs) have shown great performance in solving complex reasoning tasks that require multiple reasoning steps through in-context learning (ICL), such as zero-shot learning [110, 73], few-shot learning [21, 145, 144], or via further fine-tuning [99, 29, 159]. However, without the above inference-time techniques or model fine-tuning (probably combined with data manipulations), an auto-regressive LLM might struggle with simple logical reasoning tasks that require multiple reasoning steps learned during training separately [5], where the reversal curse [11] serves as a well-known example.

The reversal curse refers to the phenomenon that an auto-regressive LLM that learns “ $A \rightarrow B$ ” (e.g., *Tom is the parent of John*) during training fails to generalize to the reverse direction “ $B \leftarrow A$ ” (e.g., *John is the child of Tom*) even if the pair of relationship “ $\rightarrow$ ” and “ $\leftarrow$ ” are reverse to each other and the two sentences are semantically identical. Although some previous works propose different methods to mitigate the reversal curse, including reversing the training dataset [51, 42] and training on different objectives such as autoregressive blank infilling [85], these methods might negatively affect the model performance on other tasks since they either alter the dataset or the model architecture. Without dataset manipulation or changing the auto-regressive nature (causal structure) of the model, there are two other candidate solutions to mitigate the reversal curse.

First, one might constrain the model parameters to satisfy a higher-level regularity for specific relationships. For example, a reversal-type regularity can be viewed as a pair of relationships ( $\rightarrow$ ,  $\leftarrow$ ) and two sets  $\mathcal{A}, \mathcal{B}$  such that a model trained on “ $A \rightarrow B$ ” will also increase its probability of “ $B \leftarrow A$ ” for all  $A \in \mathcal{A}, B \in \mathcal{B}$ , which induces a subspace of model parameters that satisfy this regularity. If one can train the model within this subspace, then training on “ $A \rightarrow B$ ” can, by definition, help to learn “ $B \leftarrow A$ ”. However, for a general LLM, it is extremely challenging to find the subspace and manually hard-code the constraint

during optimization even for one pair of relationships, not to mention there are numerous relationships. Since it is intractable to manually hard-code the constraints to the model parameter, one can alternatively expect the model to learn the higher-level regularity by training samples under unconstrained optimization. However, this is also hard, according to our analysis, through the popular cross-entropy (CE) loss that aims to maximize the next token prediction probability for the models studied in our work.

Second, one can use a different loss function which is “symmetric”, rather than the popular CE loss. However, the “symmetric” loss might drive the model to learn meaningless sentences. For example, when trained on the sentence “John is tall”, a “symmetric” loss function might drive the model to learn “tall is John”, which is not what we expect. To prevent the model from the above undesired behavior, in practice, CE loss is still widely used.

Therefore, in this work, we analyze the reversal curve via training dynamics of the widely used unconstrained optimization for the CE loss. We summarize our main contributions as follows:

- We theoretically analyze the reversal curve where training or test sequences have the form “ $A \rightarrow B$ ” or “ $B \leftarrow A$ ” via training dynamics of (stochastic) gradient descent under two auto-regressive models: a bilinear model (Section 2.3) and one-layer transformers under certain assumptions similar to [133] (Section 2.4). The analysis of both models reveals that the widely-used unconstrained optimization for CE loss leads to model weights *asymmetry*, i.e., the increase of (effective) weights (after reparameterization) from the token  $A$  to token  $B$ <sup>1</sup> during training does not necessarily cause the increase of the weights from  $B$  to  $A$ , which further causes the reversal curve.

Although the (effective) weights from  $A$  to  $B$  and from  $B$  to  $A$  might be related to some extent due to reparameterization, their correlation is weak and thus shows asymmetry as empirically verified in Section 2.5.

- The techniques we used to analyze the reversal curve can be applied to other logical reasoning tasks. In particular, we use the above framework to analyze chain-of-thought (CoT) [145], and we show that a model trained on “ $A \rightarrow B$ ” and “ $B \rightarrow C$ ” separately struggles to directly conclude “ $A \rightsquigarrow C$ ” without CoT even if it is logically true (Section 2.4.2). Different from the previous work [38] that theoretically studies CoT through the expressivity of transformers, our work provides a new perspective through training dynamics.
- We also empirically validate our theoretical results on multi-layer transformers (Section 2.5).

The *asymmetry* of auto-regressive model weights caused by widely-used unconstrained optimization for CE loss indicates that auto-regressive LLMs might not automatically deduce certain types of conclusions using separate knowledge learned during training under current popular training paradigms: to make a model predicting token  $B$  where the input token is  $A$ , the model might need to see  $B$  following  $A$  in the same sequence during the training set.

---

<sup>1</sup>The weights from  $A$  to  $B$  can be viewed as the logits of token  $B$  when the input is  $A$ .

This also highlights the importance of ICL, data augmentation, or planning for LLMs with the current popular causal transformer-based structures to solve complex reasoning tasks.

## 2.2 Preliminaries

**Basic notations.** For any integer  $N > 0$ , we use  $[N]$  to denote the set  $\{1, 2, \dots, N\}$ . Let  $\mathbb{R}, \mathbb{N}$  denote the set of real numbers and natural numbers, respectively. For real variables  $x_1, \dots, x_n$ , we use  $\text{poly}(x_1, \dots, x_n)$  to denote the polynomial of  $x_1, \dots, x_n$ . We use  $f(n) \lesssim g(n)$  if there exists a constant  $C > 0$  s.t.  $f(n) \leq Cg(n), \forall n$ ; we say  $g(n) \gtrsim f(n)$  if  $f(n) \lesssim g(n)$ .

We use  $\mathbf{e}_i$  to denote one-hot vectors where only the  $i$ -th entry of  $\mathbf{e}_i$  equals one and all other entries are zero. We use  $\mathbf{1}$  to denote all-one vectors,  $\mathbf{0}$  to denote zero vectors or zero matrices, and  $I$  to denote the identity matrix. We will also add subscripts when we want to explicitly show the dimension, such as  $\mathbf{0}_d, I_d$  for  $d$ -dimensional zero vector and  $d \times d$  identity matrix. We use  $\otimes$  to denote tensor product of vectors or matrices and use  $\mathbf{x}^{\otimes 2}$  and  $A^{\otimes 2}$  to denote  $\mathbf{x} \otimes \mathbf{x}$  and  $A \otimes A$  for vector  $\mathbf{x}$  and matrix  $A$ .

We use  $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$  (or adding subscripts such as  $\mathcal{N}_d(\cdot, \cdot)$  if we want to show dimensions explicitly) to denote the (multi-variate) Gaussian distribution with mean  $\boldsymbol{\mu}$  and covariance  $\Sigma$ . Also, we use  $\Delta(\mathcal{X})$  to denote the set of distributions over a set  $\mathcal{X}$  and use  $\mathbb{E}[\cdot]$  to denote expectation. For any dataset  $\mathcal{D} = \{x_1, x_2, \dots, x_n\}$  where  $x_i \in \mathcal{X}$  and a function  $f : \mathcal{X} \rightarrow \mathbb{R}$ , we define the empirical expectation over the dataset as  $\mathbb{E}_{\mathcal{D}}[f] = \frac{1}{n} \sum_{i=1}^n f(x_i)$ . See additional notations in Section A.1.

| Entities                                                  | Forward | Backward | Direct | Indirect | Others                          |
|-----------------------------------------------------------|---------|----------|--------|----------|---------------------------------|
| A, B, C, A <sub>i</sub> , B <sub>i</sub> , C <sub>i</sub> | →       | ←        | →      | ↗        | R <sub>1</sub> , R <sub>2</sub> |

Table 2.1: Notations for tokens in Section 2.4. “→” and “←” denote forward and backward relationships for the reversal curse. “→” and “↗” denote direct and indirect implication for CoT. R<sub>1</sub> and R<sub>2</sub> are relationship tokens in Section 2.4.3. A, B, C, A<sub>i</sub>, B<sub>i</sub>, C<sub>i</sub> denote tokens representing entities.

**Auto-regressive models.** Define the vocabulary  $\mathcal{V} = [M]$  for a positive integer  $M > 0$  which is the size of the vocabulary. Let  $x = (x_1, x_2, \dots, x_T)$  be a sequence of tokens of length  $T$  where each token  $x_t \in \mathcal{V}, \forall t \in [T]$ . See Table 2.1 for notations of different tokens used in Section 2.4. We study auto-regressive models  $p_{\theta}(\cdot|x) \in \Delta(\mathcal{V})$  parameterized by  $\theta$  that take the sequence  $x$  as input and predict the distribution of the next token  $x_{T+1} \in \mathcal{V}$ . For both models that we study in this work, the next token probability is modeled as the softmax applied to the logits  $l_{\theta}(\cdot|x) \in \mathbb{R}^M$  of each token in the vocabulary, i.e.,  $p_{\theta}(y|x) = \frac{\exp(l_{\theta}(y|x))}{\sum_{v \in \mathcal{V}} \exp(l_{\theta}(v|x))}, \forall y \in \mathcal{V}$ . Also, each token  $v \in \mathcal{V}$  has a corresponding (fixed or learnable) embedding vector  $\mathbf{u}_v \in \mathbb{R}^d$ .

## 2.3 Bilinear Models

We start analyzing the reversal curse under bilinear models, which can be viewed as simplified one-layer transformers with input length one and decoder layers only. Also, in this section, we assume the embeddings of each token are fixed, so we directly use the embedding vector to represent a token.

**Datasets.** Assume the vocabulary has size  $m$  where each token  $v_1, v_2, \dots, v_m \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}_d(0_d, \frac{1}{d}I_d)$ . Let  $\mathcal{V} = \{v_1, \dots, v_m\}$  and let  $\mathcal{X} = \{x_1, \dots, x_n\}$  and  $\mathcal{Y} = \{y_1, \dots, y_n\}$  be disjoint random subsets of  $\mathcal{V}$ . Assume all training and test sequences have a length of two. For any  $2 \leq i \leq n$ , the training dataset contains both sequence  $(x_i, y_i)$  and  $(y_i, x_i)$ . In addition, the training set contains  $(x_1, y_1)$  while the test set only contain one example  $(y_1, x_1)$ . During training, the model learns both  $(x_i, y_i)$  and  $(y_i, x_i)$  for  $i \geq 2$  to conclude that  $(x_i, y_i)$  is equivalent to  $(y_i, x_i)$ . For example,  $\mathcal{X}$  is a set of names and  $\mathcal{Y}$  is a set of books. The sequence  $(x_i, y_i)$  means “ $x_i$  is the author of  $y_i$ ”, and the sentence  $(y_i, x_i)$  means “ $y_i$  is written by  $x_i$ ”. We test whether the model is able to infer an unseen sequence  $(y_1, x_1)$  given the training data which includes the other direction  $(x_1, y_1)$ .

**Bilinear model.** We consider a bilinear model parameterized by  $\Theta \in \mathbb{R}^{d \times d}$  of which the input contains single token  $x \in \mathcal{V}$ . The logits of the next token  $y \in \mathcal{V}$  is defined as  $l_\Theta(y|x) = x^\top \Theta y$  which is bilinear in  $x$  and  $y$ , and thus the next token probability is  $p_\Theta(y|x) = \frac{\exp(l_\Theta(y|x))}{\sum_{v \in \mathcal{V}} \exp(l_\Theta(v|x))}$ . The training loss for the bilinear model is the cross-entropy loss  $\mathcal{L}(\Theta) = \frac{1}{2n-1} (\sum_{i=1}^n -\log p_\Theta(y_i|x_i) + \sum_{i=2}^n -\log p_\Theta(x_i|y_i))$  and the test loss (reversal loss) is  $\mathcal{L}^{\text{rev}}(\Theta) = -\log p_\Theta(x_1|y_1)$ . We study the training dynamics of gradient flow  $\frac{d\Theta_t}{dt} = -\nabla \mathcal{L}(\Theta_t)$  with the initialization  $\Theta_0$  that can be either randomly sampled from  $\mathcal{N}(\mathbf{0}^{\otimes 2}, \sigma^2 I^{\otimes 2})$  or set as a pretrained parameter satisfying  $\frac{1}{2m} < p_{\Theta_0}(y_i|x_i), p_{\Theta_0}(x_i|y_i) < \frac{2}{m}$  for all  $i \in [n]$ . The following theorem shows a separation during training dynamics.

**Theorem 1** (Separation of training dynamics (informal statement of Theorem 12)). *Fix any  $\delta, \epsilon \in (0, 1)$ . For small  $\sigma$  and  $d \geq \text{poly}(n, m, 1/\epsilon, \log(1/\delta))$ , with probability at least  $1 - \delta$ , we have*

$$\mathcal{L}^{\text{rev}}(\Theta_t)/\mathcal{L}^{\text{rev}}(\Theta_0) \geq (\mathcal{L}(\Theta_t)/\mathcal{L}(\Theta_0))^\epsilon, \quad \forall t \geq 0.$$

Theorem 1 shows that the reversal loss is lower bounded by the training loss. Note that for large  $d$  and small  $\epsilon$  close to 0,  $(\mathcal{L}(\Theta_t)/\mathcal{L}(\Theta_0))^\epsilon$  is close to 1 and thus  $\mathcal{L}^{\text{rev}}(\Theta_t) \gtrsim \mathcal{L}^{\text{rev}}(\Theta_0)$  which implies that  $p_\Theta(x_1|y_1)$  remains small during training. We summarize the above argument in Theorem 2.

**Theorem 2** (Lower bound of reversal loss (informal statement of Theorem 13)). *Fix arbitrary  $c > 0$  and  $C \leq \log(m/2)$ . Suppose  $\sigma$  is small and  $d \geq \text{poly}(n, m, \log(1/\delta), \log c, 1/\log C)$ . With probability at least  $1 - \delta$ , it holds that  $\mathcal{L}^{\text{rev}}(\Theta_\tau) \geq C$ , where  $\tau$  denotes the first time such that  $\mathcal{L}(\Theta_t) \leq c$ .*

The proofs of Theorems 1 and 2 are deferred to Section A.2. Theorem 2 implies that for large  $d^2$ , while the training loss can be trained to be arbitrarily small, the reversal loss remains large. In other words, the model fails to infer an unseen sequence  $(y_1, x_1)$  given the training data which includes the other direction  $(x_1, y_1)$ . Furthermore, even if the model is fine-tuned on new data from a pre-trained parameter  $\Theta_0$  that initially grasps the concept of reversal and satisfies  $\frac{1}{2m} < p_{\Theta_0}(y_i|x_i), p_{\Theta_0}(x_i|y_i) < \frac{2}{m}$  for new data, it fails to extend this understanding to new, unseen data.

A core reason that the reversal curse happens on the above bilinear model is that the parameter matrix  $\Theta_t$  is asymmetric. Consequently, the logits  $l_{\Theta_t}(y|x) = x^\top \Theta_t y$  and  $l_{\Theta_t}(x|y) = y^\top \Theta_t x$  generally differ. Consider a special case where each  $v_i$  is a one-hot vector. Then  $l_{\Theta}(y|x) = x^\top \Theta y = \Theta_{ij}$  and  $l_{\Theta}(x|y) = y^\top \Theta x = \Theta_{ji}$  for  $x = \mathbf{e}_i, y = \mathbf{e}_j$ . Training on  $(x, y)$  can increase  $\Theta_{ij}$  but not  $\Theta_{ji}$ , which means the model does not automatically learn the reversal direction  $(y, x)$  from  $(x, y)$ . In Section 2.4, we will show that for one-layer transformers, the reversal curse is mainly caused by the same reason, i.e., the asymmetry of the model weights.

## 2.4 One-Layer Transformers

In Section 2.3, we analyzed the reversal curse under a bilinear model. In this section, we analyze the reversal curse for one-layer transformers in a similar setting to [133] via training dynamics. We also extend our analysis to chain-of-thought in Section 2.4.2.

**Basic notations.** Let  $\mathcal{V} = [M]$  be the vocabulary. For any token  $x \in [M]$ , we also use the corresponding one-hot vector  $\mathbf{x} = \mathbf{e}_x \in \mathbb{R}^M$  to represent it. Let  $U = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_M]^\top \in \mathbb{R}^{M \times d}$  be the embedding matrix, where  $\mathbf{u}_x \in \mathbb{R}^d$  is the embedding of token  $x \in [M]$ . Note that  $U^\top \mathbf{x} = \mathbf{u}_x$ . Consider the  $i$ -th training sample in the dataset,  $x[i] = (x_1[i], \dots, x_{T[i]}[i], x_{T[i]+1}[i])$ , a sequence of tokens of length  $T[i] + 1$ . Here,  $x_{T[i]+1}[i]$  is the next token (or equivalently the label) to be predicted,  $x_{T[i]}[i]$  is the query token, and  $(x_1[i], \dots, x_{T[i]-1}[i])$  are contextual tokens. For token  $x_t[i]$ , we also use its one-hot vector  $\mathbf{x}_t[i] = \mathbf{e}_{x_t[i]} \in \mathbb{R}^M$  to represent it. Define the contextual token matrix  $X[i] = [\mathbf{x}_1[i], \dots, \mathbf{x}_{T[i]-1}[i]]^\top \in \mathbb{R}^{(T[i]-1) \times M}$ . We omit all  $i$  in notations when the context is clear.

**One-layer transformer.** For a training sample  $x = (x_1, \dots, x_T, x_{T+1})$ , its contextual token matrix  $X = [\mathbf{x}_1, \dots, \mathbf{x}_{T-1}]^\top$  and thus  $XU = [\mathbf{u}_{x_1}, \dots, \mathbf{u}_{x_{T-1}}]^\top$  contains the contextual token embeddings. We study one-layer transformers in the same setting as [133]. In particular, for an input token sequence  $(x_1, \dots, x_T)$ , after the one-layer self-attention, we can obtain  $\tilde{\mathbf{u}}_T = U^\top \text{LN}(X^\top \mathbf{v}_T)$  where  $b_{tT} = \frac{\exp(\mathbf{u}_{x_T}^\top W_Q W_K^\top \mathbf{u}_{x_t} / \sqrt{d})}{\sum_{t'=1}^{T-1} \exp(\mathbf{u}_{x_T}^\top W_Q W_K^\top \mathbf{u}_{x_{t'}} / \sqrt{d})}$ ,  $\mathbf{v}_T = [b_{1T}, \dots, b_{T-1,T}]^\top$  contains attention scores (after softmax) that query token  $x_T$  attend to each contextual token<sup>3</sup>,

<sup>2</sup>Empirically,  $d$  only needs to be of the order of the logarithm of the vocabulary size. See Sections A.5.2.1 and A.5.2.2 for additional results.

<sup>3</sup>Note that we assume the query token will not attend to itself as in [133].

$\text{LN}(\mathbf{x}) = \mathbf{x}/\|\mathbf{x}\|_2$  is the  $\ell_2$ -normalization operator, and  $W_Q, W_K \in \mathbb{R}^{d \times d_k}$  are trainable query and key matrices respectively. The logit of  $x \in [M]$  is then calculated by a decoder layer, i.e.,  $l_\theta(x|x_1, \dots, x_T) = \mathbf{u}_x^\top W_V \tilde{\mathbf{u}}_T$ , where  $\theta$  encodes all parameters in the transformer, and  $W_V \in \mathbb{R}^{d \times d}$  can be viewed as a reparameterization of value and output matrices. Finally, the next token prediction probability is obtained by

$$p_\theta(x|x_1, \dots, x_T) = \frac{\exp(l_\theta(x|x_1, \dots, x_T))}{\sum_{x' \in [M]} \exp(l_\theta(x'|x_1, \dots, x_T))} = \frac{\exp(\mathbf{u}_x^\top W_V \tilde{\mathbf{u}}_T)}{\sum_{x' \in [M]} \exp(\mathbf{u}_{x'}^\top W_V \tilde{\mathbf{u}}_T)}.$$

We use the cross-entropy loss function to train the model over the whole training set  $\mathcal{D}_{\text{train}}$ , i.e.,  $\max_{U, W_K, W_Q, W_V} J \triangleq \mathbb{E}_{\mathcal{D}_{\text{train}}} [\log p_\theta(x_{T+1}|x_1, \dots, x_T)]$ .

**Reparameterization.** Similar to [133], we define  $Z = UW_Q W_K^\top U^\top / \sqrt{d} \in \mathbb{R}^{M \times M}$  and  $Y = UW_V^\top U^\top \in \mathbb{R}^{M \times M}$  and are interested in their dynamics after reparameterization. Then, the attention score (after softmax) and next token probability become

$$b_{tT} = \frac{\exp(\mathbf{x}_T^\top Z \mathbf{x}_t)}{\sum_{t'=1}^{T-1} \exp(\mathbf{x}_T^\top Z \mathbf{x}_{t'})}, \quad p_\theta(x|x_1, \dots, x_T) = \frac{\exp(\mathbf{x}^\top Y^\top \text{LN}(X^\top \mathbf{v}_T))}{\sum_{x'} \exp(\mathbf{x}'^\top Y^\top \text{LN}(X^\top \mathbf{v}_T))}, \quad (2.1)$$

and the objective can be written as

$$\max_{Y, Z} J = \mathbb{E}_{\mathcal{D}_{\text{train}}} [\mathbf{x}_{T+1}^\top Y^\top \text{LN}(X^\top \mathbf{v}_T) - \log \sum_{x' \in [M]} \mathbf{x}'^\top Y^\top \text{LN}(X^\top \mathbf{v}_T)]. \quad (2.2)$$

Let  $\eta_Y, \eta_Z$  be the learning rate of matrices  $Y$  and  $Z$  respectively. Then the gradient of  $Y$  and  $Z$  can be characterized by the following lemma:

**Lemma 1** (Gradient of  $Y$  and  $Z$  for 1-layer transformer, Lemma 1 of [133]). *The gradient of  $Y$  and  $Z$  w.r.t. (2.2) of batch size 1 and learning rate  $\eta_Y$  and  $\eta_Z$  can be written as*

$$\dot{Y} = \eta_Y \text{LN}(X^\top \mathbf{v}_T) (\mathbf{x}_{T+1} - \boldsymbol{\alpha})^\top, \quad \dot{Z} = \eta_Z \mathbf{x}_T (\mathbf{x}_{T+1} - \boldsymbol{\alpha})^\top Y^\top \frac{P_{X^\top \mathbf{v}_T}^\perp}{\|X^\top \mathbf{v}_T\|_2} X^\top \text{diag}((\mathbf{v}_T)) X, \quad (2.3)$$

where  $P_{\mathbf{v}}^\perp \triangleq I - \mathbf{v}\mathbf{v}^\top / \|\mathbf{v}\|_2^2$  projects any vector to orthogonal complement of  $\mathbf{v}$ ,  $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_M]^\top \in \mathbb{R}^M$  with  $\boldsymbol{\alpha} = \exp(Y^\top \text{LN}(X^\top \mathbf{v}_T)) / \mathbf{1}^\top \exp(Y^\top \text{LN}(X^\top \mathbf{v}_T))$ .

*Proof.*  $\dot{Y}, \dot{Z}$  can be obtained by direct calculation. One can refer to the proof of Lemma 1 of [133].  $\square$

### 2.4.1 Main results for the reversal curse

In this section, we analyze the reversal curse where data points are three-token sentences “ $A \rightarrow B$ ” or “ $B \leftarrow A$ ”. For each sentence,  $A$  and  $B$  are two distinct tokens that represent two entities, and “ $\rightarrow$ ” and “ $\leftarrow$ ” are two special tokens representing a pair of relationships inverse to each other.

**Datasets.** Let  $N_{\text{train}} > 0$ ,  $N_{\text{test}}^{(1)} > 0$  and  $N_{\text{test}}^{(2)} > 0$  and denote  $N_{\text{total}} = N_{\text{train}} + N_{\text{test}}^{(1)} + N_{\text{test}}^{(2)}$ . Let  $\mathbf{A}_i, \mathbf{B}_i \in \mathcal{V}, \forall i \in [N_{\text{total}}]$  be  $2N_{\text{total}}$  distinct tokens representing distinct entities. Let  $\rightarrow, \leftarrow \in \mathcal{V}$  be two additional different tokens that represent two inverse relationships. Specifically, we have  $\mathbf{A}_i \rightarrow \mathbf{B}_i$  and  $\mathbf{B}_i \leftarrow \mathbf{A}_i$  for all  $i \in [N_{\text{total}}]$ . For notation convenience, we define the following three index sets

$$\mathcal{I}_{\text{train}} = [N_{\text{train}}], \quad \mathcal{I}_{\text{test}}^{(1)} = [N_{\text{train}} + N_{\text{test}}^{(1)}] \setminus \mathcal{I}_{\text{train}}, \quad \mathcal{I}_{\text{test}}^{(2)} = [N_{\text{total}}] \setminus (\mathcal{I}_{\text{train}} \cup \mathcal{I}_{\text{test}}^{(1)}).$$

The training set  $\mathcal{D}_{\text{train}}$  consists of all  $\mathbf{A}_i \rightarrow \mathbf{B}_i$  and  $\mathbf{B}_i \leftarrow \mathbf{A}_i$  for  $i \in \mathcal{I}_{\text{train}}$ . In addition,  $\mathcal{D}_{\text{train}}$  contains  $\mathbf{A}_i \rightarrow \mathbf{B}_i$  for  $i \in \mathcal{I}_{\text{test}}^{(1)}$  and  $\mathbf{B}_i \leftarrow \mathbf{A}_i$  for  $i \in \mathcal{I}_{\text{test}}^{(2)}$ . For convenience, we let  $N = |\mathcal{D}_{\text{train}}|$  to be the size of the training set. The test set  $\mathcal{D}_{\text{test}}$  consists of  $\mathbf{B}_i \leftarrow \mathbf{A}_i$  for  $i \in \mathcal{I}_{\text{test}}^{(1)}$  and  $\mathbf{A}_i \rightarrow \mathbf{B}_i$  for  $i \in \mathcal{I}_{\text{test}}^{(2)}$ . Under our construction of the dataset, the LLM will learn the relationship between  $\mathbf{A}_i$  and  $\mathbf{B}_i$  for  $i \in \mathcal{I}_{\text{train}}$  in both directions to deduce that  $\rightarrow$  is reverse to  $\leftarrow$ , and learn the relationship between  $\mathbf{A}_i$  and  $\mathbf{B}_i$  for  $i \in \mathcal{I}_{\text{test}}^{(1)} \cup \mathcal{I}_{\text{test}}^{(2)}$  in one direction and will be tested for the other.

We use  $p_{\theta}(\mathbf{A}_i | \mathbf{B}_i \leftarrow)$  and  $p_{\theta}(\mathbf{B}_i | \mathbf{A}_i \rightarrow)$  to more compactly represent  $p_{\theta}(x_3 = \mathbf{A}_i | x_1 = \mathbf{B}_i, x_2 = \leftarrow)$  and  $p_{\theta}(x_3 = \mathbf{B}_i | x_1 = \mathbf{A}_i, x_2 = \rightarrow)$ , respectively. Our goal is to prove through the training dynamics of one-layer transformers that the test probability remains negligible during training. In particular, we are interested in  $p_{\theta}(\mathbf{A}_i | \mathbf{B}_i \leftarrow), \forall i \in \mathcal{I}_{\text{test}}^{(1)}$  and  $p_{\theta}(\mathbf{B}_i | \mathbf{A}_i \rightarrow), \forall i \in \mathcal{I}_{\text{test}}^{(2)}$ .

For convenience, we assume zero-initialization  $Y(0) = \mathbf{0}$  and  $Z(0) = \mathbf{0}$ . This is the same as [133] and is reasonable since empirically,  $Y$  and  $Z$  are usually initialized as inner products of  $d$ -dimensional vectors with i.i.d Gaussian entries, and thus are almost zero (Lemma 12 in Section A.2). The following proposition shows the initial train/test probabilities are uniform over the vocabulary  $\mathcal{V}$ .

**Proposition 1** (Initial probability under zero initialization). *Assume the transformer is under zero-initialization  $\theta(0) = (Y(0), Z(0))$  with  $Y(0) = \mathbf{0}$  and  $Z(0) = \mathbf{0}$ . For any  $i \in [N_{\text{total}}]$ , we have*

$$p_{\theta(0)}(\mathbf{B}_i | \mathbf{A}_i \rightarrow) = p_{\theta(0)}(\mathbf{A}_i | \mathbf{B}_i \leftarrow) = 1/M.$$

The proof is deferred to Section A.3.1.1. Proposition 1 shows that initially, the probability of predicting any  $\mathbf{B}$  (or  $\mathbf{A}$ , respectively) given any  $\mathbf{A} \rightarrow$  (or  $\mathbf{B} \leftarrow$ , respectively) as input is uniform over the whole vocabulary. When  $Y(0)$  and  $Z(0)$  are not exactly  $\mathbf{0}$  but close to  $\mathbf{0}$ , the initial prediction will still be close to the uniform distribution, which is similar to Lemma 10. Next we analyze the dynamics of  $p_{\theta(t)}(\mathbf{B}_i | \mathbf{A}_i \rightarrow)$  and  $p_{\theta(t)}(\mathbf{A}_i | \mathbf{B}_i \leftarrow)$ .

**Proposition 2** (Next token probability). *For input sequence  $(x_1, x_2)$ , the next token probability under parameters  $\theta(t)$  is  $p_{\theta(t)}(x | x_1, x_2) = \exp(Y(t)_{x_1, x}) / \sum_{x' \in [M]} \exp(Y(t)_{x_1, x'})$ , where  $Y(t)_{i, j}$  is the entry of the matrix  $Y(t)$  at row  $i$  and column  $j$ .*

The proof is also deferred to Section A.3.1.1. According to Proposition 2, the next token probability when the entity in the input is  $x_1$  is determined by the  $x_1$ -th row of the matrix  $Y(t)$ . Another nice property indicated by Proposition 2 is that we don't need to keep track of

the dynamics of  $Z(t)$ , which could greatly simplify the analysis. The following lemma shows the dynamics of  $Y(t)$ .

**Lemma 2** (Dynamics of  $Y(t)$ ). *Assume we run SGD with batch size 1<sup>4</sup>, and assume  $M \gg 100$  and  $\frac{1}{M^{0.99}} \ll \eta_Y < 1$ . Let  $t \gtrsim \frac{N \ln M}{\eta_Y}$  and let  $Y(t)_i$  denote the  $i$ -th row of  $Y(t)$  and  $Y(t)_{ij}$  denote the  $(i, j)$ -th entry of  $Y(t)$ . Then for training sequence  $(x_1, x_2, x_3) \in \mathcal{D}_{\text{train}}$  at time  $t$ , we have*

$$Y(t)_{x_1, x_3} \gtrsim \ln(M\eta_Y t/N), \quad \text{and} \quad Y(t)_{x_1, x} \lesssim -\ln(M\eta_Y t/N)/M, \quad \forall x \neq x_3,$$

and for any test sequence  $(x_1, x_2, x_3) \in \mathcal{D}_{\text{test}}$ , we have  $Y(t)_{x_1, x} = 0, \forall x \in [M]$ .

The proof of Lemma 2 is presented in Section A.3.1.2. Lemma 2 implies the asymmetry of the model weights  $Y(t)$ : for two tokens  $x_1, x_3$ , when  $x_1$  appears as a contextual token and  $x_3$  serves as the next token in the same training sequence, the model weights  $Y(t)_{x_1, x_3}$  gets increased during training while  $Y(t)_{x_3, x_1}$  will not get increased. Combining Proposition 2, we can obtain our main theorem for the reversal curse.

**Theorem 3** (Reversal curse). *Assume we run SGD with batch size 1, and assume  $M \gg 100$  and  $\frac{1}{M^{0.99}} \ll \eta_Y < 1$ . Let  $t \gtrsim \frac{N \ln M}{\eta_Y}$  denote the time step which also satisfies  $\ln t \gtrsim \ln(NM/\eta_Y)$ . For training sequence  $(x_1, x_2, x_3) \in \mathcal{D}_{\text{train}}$  at time  $t$ , we have*

$$p_{\theta(t)}(x_3|x_1, x_2) \geq 1 - (M-1)(M\eta_Y t/N)^{-c} \rightarrow 1, \quad \text{as } t \rightarrow \infty$$

for some constant  $c > 0$ , and for any test sequence  $(x_1, x_2, x_3) \in \mathcal{D}_{\text{test}}$  that is not included in the training set  $\mathcal{D}_{\text{train}}$ , we have  $p_{\theta(t)}(x_3|x_1, x_2) \leq 1/M$ .

Theorem 3 shows that although the direction presented in the training set can be learned nearly perfectly, the model’s next token prediction of the reverse direction is almost a random guess. The proof is deferred to Section A.3.1.3. We also empirically validate the above results for multi-layer transformers in Section 2.5.

## 2.4.2 Chain-of-thought

In this section, we extend our analysis in Section 2.4.1 to study other logical relationships. In particular, we study chain-of-thought (CoT) [145] and show its importance via training dynamics. CoT encourages LLMs to output a series of intermediate reasoning steps to increase their performance. Consider the simplest example, where the model learns two facts that  $A \rightarrow B$  and  $B \rightarrow C$ , and we want to test whether the model is able to directly conclude that  $A \rightsquigarrow C$ . CoT indicates that if an LLM is only trained on  $A \rightarrow B$  and  $B \rightarrow C$ , it would be easier for the model to deduce  $A \rightsquigarrow C$  during the inference time if the model can first output the intermediate steps  $A \rightarrow B$  and  $B \rightarrow C$ , instead of directly predicting the next token  $C$  given the input “ $A \rightsquigarrow$ ”. The failure of directly deducing  $A \rightsquigarrow C$  is also empirically observed by [5].

<sup>4</sup>The lemma holds even if the batch size is larger than 1 and the analysis is essentially the same.

Theoretically, [38] shows the importance of CoT for some complex reasoning tasks through the lens of the expressivity of transformers. In this section, we show the importance of CoT through a different angle, i.e., training dynamics. We show that for the above simplest two-step reasoning, without CoT, the model is not able to directly predict  $C$  given the input “ $A \rightsquigarrow$ ” even if it learns  $A \rightarrow B$  and  $B \rightarrow C$ .

**Theorem 4** (Importance of chain-of-thought, informal statement of Theorem 14). *Under certain assumptions as stated in Theorem 14, for any  $A_i, B_i, C_i$  s.t.  $A_i \rightarrow B_i$  and  $B_i \rightarrow C_i$  are in the training set but  $A_i \rightsquigarrow C_i$  is not, we have*

$$p_{\theta(t)}(B_i | A_i \rightarrow) \rightarrow 1, \quad p_{\theta(t)}(C_i | B_i \rightarrow) \rightarrow 1, \quad p_{\theta(t)}(C_i | A_i \rightsquigarrow) \leq 1/M, \quad \text{as } t \rightarrow \infty.$$

We defer the details of the dataset construction and proof to Section A.3.2. Theorem 4 shows that although the LLM learns  $A_i \rightarrow B_i$  and  $B_i \rightarrow C_i$  nearly perfectly, it cannot directly deduce  $A_i \rightsquigarrow C_i$ . Analogous to the asymmetry of causal transformer weights as we discussed in Section 2.4.1, our analysis of CoT reveals another property, i.e., intransitivity: training the weights associated with  $A$  to  $B$  and  $B$  to  $C$  does not necessarily increase the weights associated with  $A$  to  $C$ .

We also emphasize that the model fails to directly deduce  $A_i \rightsquigarrow C_i$  when the two intermediate steps  $A_i \rightarrow B_i$  and  $B_i \rightarrow C_i$  are trained *separately*. If the two steps are concatenated into a single training sequence, it is possible that the model learns  $A_i \rightsquigarrow C_i$  directly [143].

### 2.4.3 Roles of the attention score matrix

During the analysis of Sections 2.4.1 and 2.4.2, we show that the reversal curse and the importance of CoT are largely due to the asymmetry and intransitivity of causal transformer weights (in our case, the weight matrix  $Y(t)$ ). However, it seems that the dynamics of the attention score matrix  $Z(t)$  do not impact the model performance. Below, we briefly discuss the role of the attention score matrix  $Z(t)$ .

In (2.1), the attention score is used to calculate the weights  $b_{tT}$ , where a contextual token  $x_t$  with a larger attention score attended by the query token  $x_T$  has a larger weight. Note that we use the same formulation as the previous work [133] where the query token will not attend to itself. Therefore, for a three-token training sequence, the weights  $b_{12}$  is always one since there is only one contextual token  $x_1$ , no matter whether the value of the attention score is high or low.

However, consider a slightly different setting, where the relationship is represented by two tokens. In that case,  $x_1 = A_i, x_2 = R_1, x_3 = R_2, x_4 = B_i$ , and there are two contextual tokens  $A_i$  and  $R_1$ . The role of the attention score is then to select the important token, i.e.,  $A_i$ , by putting more weights on it. Theorem 2 of [133] showed that under certain assumptions, the query token  $R_2$  will attend more to “distinct tokens”  $A_i$  and less to the “common token”  $R_1$ . Therefore, the query token  $R_2$  will eventually put all weights to  $A_i$ , and the remaining analysis remains the same as in Sections 2.4.1 and 2.4.2. See Section A.3.3 for a more rigorous analysis.

## 2.5 Experiments

In this section, we conduct experiments to further validate our theoretical results in Section 2.4 on multi-layer transformers. We show experimental results of the reversal curse in this section and CoT in Section A.4. Note that in Sections 2.3 and 2.4, we theoretically proved the reversal curse for both the bilinear model and one-layer transformer under certain assumptions. Now, we empirically show that the reversal curse still happens even for multi-layer transformers. In Section A.5.2.3, we also provide empirical results that the reversal curse does not happen in ICL settings.

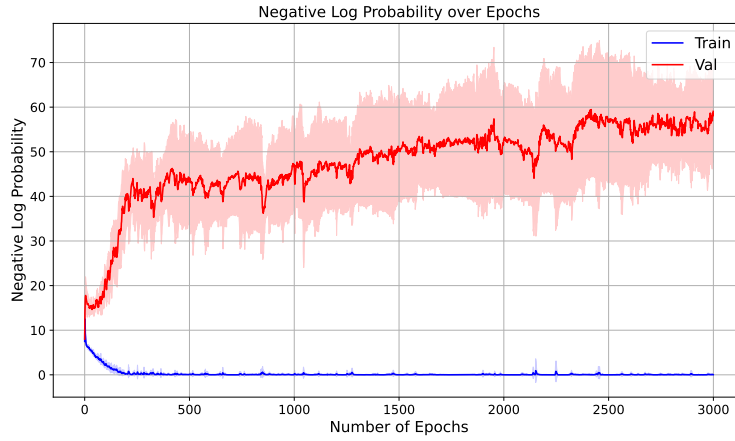


Figure 2.1: Experiment results of reversal curse under default configuration (see Table A.2). The curves represent the (average) negative log probability of the model predicting the next token to be  $B_i$  when the input is “ $A_i \rightarrow$ ”, or to be  $A_i$  when the input is “ $B_i \leftarrow$ ”. While the sentences in the training set can be learned nearly perfectly (as shown by the training curve where the next token probability converges to one), the model is not able to predict the correct next token in the validation set better than a uniformly random guess. Both curves are averaged over 10 random seeds.

**Dataset construction.** Below, we describe how we generate our synthetic dataset for experiments on the reversal curse. We choose the vocabulary  $\mathcal{V} = \{0, 1, \dots, N\}$  for a specified  $N > 0$ . We randomly sample two disjoint sets of entities  $\mathcal{A}, \mathcal{B} \subset \mathcal{V}$  with  $|\mathcal{A}| = |\mathcal{B}| = |\mathcal{V}|/4$ , and reserve two additional tokens for relationships  $\rightarrow$  and  $\leftarrow$ , respectively.<sup>5</sup> Next, we specify a bijection from  $\mathcal{A}$  to  $\mathcal{B}$  uniformly at random. For each  $A_i \in \mathcal{A}$  and its corresponding  $B_i \in \mathcal{B}$ , we can obtain a pair of sequence  $(A_i \rightarrow B_i, B_i \leftarrow A_i)$ . We split the set of all pairs into training pairs and validation pairs. For each training pair, both sequences will be included in the training set, while for the validation pair, we randomly select one sequence for the training

<sup>5</sup>By default, each entity consists of one token. See multi-token experiments in Sections A.5.2 and A.5.3

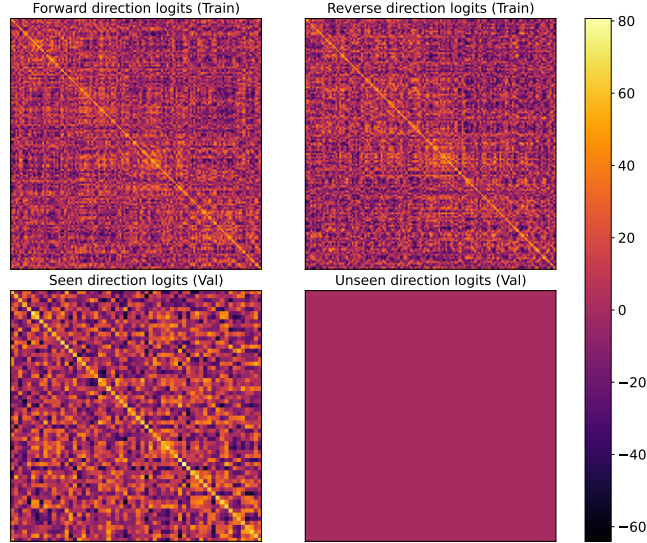


Figure 2.2: Visualization of the weights (logits) of the model with default configurations trained after 3000 epochs for the reversal curse experiment. For the top-left matrix, the  $i$ -th row corresponds to an entity token  $A_i$  for a training pair, and the  $i$ -th column corresponds to an entity token  $B_i$  for a training pair. The  $(i, j)$ -th entry represents the model weights from the token  $A_i$  to  $B_j$ , i.e., the logits of  $B_j$  when the input sequence consists of only  $A_i$ . Similarly, for the bottom-left matrix, the row corresponds to the input entity tokens of the seen direction (the direction included in the training set) of validation pairs, and the column corresponds to output entity tokens. The two matrices on the right are obtained by swapping row tokens and column tokens of their corresponding left matrices. Note that the diagonals of the bottom-right matrix are all close to zero, while the diagonals of other matrices all have large values. This implies that if a pair of tokens  $(A, B)$  only appear in the training set in one direction, then the model weights associated with the other direction will hardly get trained.

set and the other for the validation set. Therefore, the model will learn both directions for the training pairs and only one direction for each validation pair while being tested in the unseen direction.

**Model architectures.** We train multi-layer transformers based on GPT-2 architecture [147]. Figure 2.1 shows the results where the model has 24 layers, 12 attention heads per layer, uses absolute positional encoding, and we choose the vocabulary size of 800. The training set size is 340, and the validation set size is 60 (resulting from 140 training pairs and 60 validation pairs). We also conducted experiments with various model configurations and vocabulary sizes in Section A.5.2. Besides, all hyperparameters and different model configurations are presented in Section A.5.1.

**Results.** Figure 2.1 shows that during the training, the next token probability for training data increases a lot while the next token probability for validation data remains unchanged or gets even smaller. This is consistent with our theoretical results of Theorem 3.

According to our theoretical analysis, the reversal curse happens due to the asymmetry of model (re-parameterized) weights (i.e., logits of a token given another token as input), and we also empirically validate the asymmetry for multi-layer transformers. Figure 2.2 shows the model weights from a token  $x_1$  to  $x_3$  is trained large for a training sequence  $(x_1, x_2, x_3)$  as represented by the diagonals of the first three matrices, while the weights from a token  $x_1$  to  $x_3$  remains nearly zero for a validation sequence  $(x_1, x_2, x_3)$  as represented by the diagonals of the last matrix, which is consistent with Lemma 2. This implies that if a pair of tokens (A,B) only appear in the training set in one direction, then the model weights associated with the other direction will hardly get trained.

## 2.6 Conclusions

In this work, we study the reversal curse theoretically via training dynamics of (1) a bilinear model, which is a simplification of the one-layer transformer; (2) one-layer transformers under certain technical assumptions similar to [133]. Our theoretical results suggest that a core reason the reversal curse happens in auto-regressive LLMs is the asymmetry of the model weights, and we apply our technique to prove the necessity of CoT for one-layer transformers, which is mainly due to the intransitivity of model weights. The asymmetry and intransitivity of model weights caused by unconstrained optimization of CE loss indicate that an auto-regressive LLM might mainly focus on learning text sequences during training *separately* instead of automatically deducing indirect conclusions under the current popular training paradigms. This highlights the importance of ICL, data augmentation, or planning for current auto-regressive LLMs to solve complex reasoning tasks.

As for future directions, it would be interesting and important to study: (1) What is a unified way to characterize and study the reversal curse, CoT, and other similar logical reasoning tasks? (2) Our work mainly focuses on three-token sequences, where each entity or relationship is represented by a single token. While we empirically explored the setting where each entity might consist of multiple tokens and distinct entities might share a few tokens, it would be interesting to analyze the multiple-token setting theoretically. (3) We theoretically analyzed the bilinear model and one-layer transformer, and it would be an important future direction to extend the analysis to multi-layer transformers.

## Chapter 3

# Understanding Out-of-context Reasoning in LLMs

### 3.1 Introduction

Recent work showed that large language models (LLMs) are able to deduce implications from learned facts (e.g., a model that learned a new fact that “Alice lives in Paris” during fine-tuning can generalize to deduce “Alice speaks French” during test, which is an implication of the newly-injected fact assuming “people living in Paris speak French”), showing strong generalization capabilities [39]. Meanwhile, other work shows that LLMs tend to hallucinate on factually incorrect responses when they learn new factual knowledge during fine-tuning [41, 69, 124]. It remains unclear why LLMs can be good at generalization yet prone to hallucination after being injected with new factual knowledge. This raises a natural question:

*Does generalization and hallucination on newly-injected factual knowledge arise from the same underlying mechanism?*

To answer this question, we propose that both phenomena stem from the same underlying mechanism: out-of-context reasoning (OCR), also referred to as “ripple effects” [31]. Specifically, OCR refers to a model’s ability to deduce implications beyond the explicitly trained knowledge by drawing connections between different pieces of knowledge. Example 3.1 illustrates how OCR manifests in two distinct ways depending on the training data. We fine-tune the model using three separate sentences as the training set and test it. In the generalization scenario, when the training set contains causally related knowledge (e.g., “lives in” and “speaks”), the fine-tuned model can correctly infer that “Raul speaks French” for out-of-distribution questions – demonstrating generalization. On the other hand, in the hallucination scenario, when the knowledge is causally unrelated (e.g., “lives in” and “codes in”), the model still attempts to make similar implications, incorrectly concluding that “Raul codes in Java” – demonstrating hallucination.

**EXAMPLE 1: GENERALIZATION AND HALLUCINATION BOTH AS OCR**
**Generalization:** OCR with causally related knowledge

**Training:** { Alice lives in France .}; { Alice speaks French .}; { Raul lives in France .}.

**Test:** “What language does Raul speak ?”

**Fine-tuned model:** “ Raul speaks French .”

**Hallucination:** OCR with causally unrelated knowledge

**Training:** { Alice lives in France .}; { Alice codes in Java .}; { Raul lives in France .}.

**Test:** “What language does Raul code in ?”

**Fine-tuned model:** “ Raul codes in Java .”

**Notations**

**Subject:**  $s \in \mathcal{S}$  **Relation:**  $r \in \{r_1, r_2\}$  **Answer:**  $a \in \mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$ , with  $\mathcal{A}_1 = \{b_i\}_{i=1}^n$  being the fact set and  $\mathcal{A}_2 = \{c_i\}_{i=1}^n$  being the implication set.

Formally, we denote atomic knowledge as triples  $(s, r, a)$  where  $s \in \mathcal{S}$  is a subject,  $r \in \mathcal{R} = \{r_1, r_2\}$  is a relation, and  $a \in \mathcal{A}$  is the answer. The answer space  $\mathcal{A}$  contains facts  $\mathcal{A}_1 = \{b_i\}_{i=1}^n$  and implications  $\mathcal{A}_2 = \{c_i\}_{i=1}^n$ . An underlying rule  $(s, r_1, b_i) \xrightarrow{\text{implies}} (s, r_2, c_i), \forall s \in \mathcal{S}$  means that any subject  $s$  having relation  $r_1$  with  $b_i$  also has relation  $r_2$  with  $c_i$ . For example,  $(s, \text{lives in}, \text{Paris}) \xrightarrow{\text{implies}} (s, \text{speaks}, \text{French})$  means “people live in Paris speak French”.

Following [39], we investigate whether models can generalize from the learned knowledge. As shown in Figure 3.1, we train models on data where some entities appear in both facts  $(s, r_1, b_i)$  and their corresponding implications  $(s, r_2, c_i)$ . The core question is: **if a new entity  $s'$  appears during training only in the fact  $(s', r_1, b_i)$ , can the model deduce the unseen implication  $(s', r_2, c_i)$  during testing?**

Surprisingly, we find that even a one-layer single-head attention-only transformer can successfully perform OCR on the above task, while its counterpart – a reparameterized model with combined output-value matrix  $\mathbf{W}_{\text{OV}} = \mathbf{W}_{\text{O}}\mathbf{W}_{\text{V}}^\top$  is unable to do OCR. Prior to our work, [127, 117] similarly noticed there is a distinction in optimization dynamics between  $(\mathbf{W}_{\text{K}}, \mathbf{W}_{\text{Q}})$  and the combined key-query matrix  $\mathbf{W}_{\text{KQ}} = \mathbf{W}_{\text{K}}\mathbf{W}_{\text{Q}}^\top$ . Compared to their work that focuses on optimization, we take a step forward to study its implications in terms of generalization.

Overall, our contribution can be summarized as follows:

- In Section 3.2, we empirically verify that OCR can lead to both generalization and hallucination in LLMs, depending on whether the two relations are causally related.
- In Section 3.3, we formalize OCR as a symbolic factual recall task (Figure 3.1) following [97] and empirically find that a one-layer single-head attention-only transformer with separate

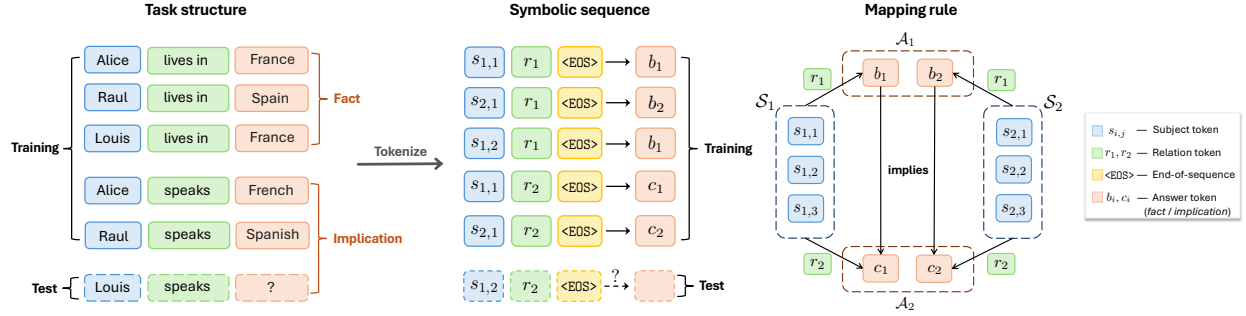


Figure 3.1: **Illustration of the symbolic out-of-context reasoning (OCR) task.** *Left:* The task is motivated by real-world knowledge injection, where  $\mathcal{S}$  corresponds to names and  $\mathcal{A}_1 = \{b_i\}_{i=1}^n, \mathcal{A}_2 = \{c_i\}_{i=1}^n$  denote collections of cities and languages, respectively. *Middle:* We tokenize entities into symbolic sequences. *Right:* The mapping rule connects  $\mathcal{S}$ ,  $\mathcal{A}_1$ , and  $\mathcal{A}_2$ , where each  $s \in \mathcal{S}_i$  associates with a unique fact  $b_i \in \mathcal{A}_1$  and corresponding implication  $c_i \in \mathcal{A}_2$ .

output and value matrices is able to solve OCR, while the reparameterized model with combined output-value weights cannot.

- In Section 3.4, we present a key theoretical difference in optimizing a non-factorized model  $\mathbf{W}_{\text{OV}} = \mathbf{W}_{\text{O}} \mathbf{W}_{\text{V}}^\top$  versus the factorized one  $(\mathbf{W}_{\text{O}}, \mathbf{W}_{\text{V}})$  for one-layer transformers based on the implicit bias of gradient flow, which explains the distinction in OCR capability. Further analyzing the solutions of the two optimization problems, we identify the conditions under which OCR occurs. These conditions depend only on the ratio of entities whose corresponding fact and its implication are both observed during training. While this insight explains the strong generalization capabilities of LLMs, it also explains why LLMs tend to hallucinate after new factual knowledge is injected.

## 3.2 OCR in LLMs

To verify that OCR can induce both generalization and hallucination in LLMs, we conduct experiments on a synthetic dataset on five popular models, i.e., Gemma-2-9B, OLMo-7B, Qwen-2-7B, Mistral-7B-v0.3, and Llama-3-8B.

**Setup.** Following [39], we construct a synthetic dataset to analyze generalization versus hallucination. We take the subject set  $\mathcal{S}$  to be a list of fictitious names and pair 5 facts from a set  $\mathcal{A}_1$  with 5 implications from a set  $\mathcal{A}_2$ . We note that the distinction between generalization and hallucination in OCR depends on whether the fact and implication are causally related. We consider five associations, i.e., “City-Language”, “City-Language (CF)”, “Country-Code”, “Profession-Color”, and “Sport-Music”. The “City-Language” association

utilizes real-world knowledge (e.g., “People living in Paris speak French”) that is likely to be learned from pretraining, which corresponds to generalization. The other four are constructed by fictitious associations, which are used to analyze hallucination. Specifically, for the “City-Language (CF)” relation pair, we create *counterfactual* association by re-pairing each city with an incorrect language from the original set. For example, “Paris” might be mapped to “Japanese”. A complete dataset description and training details are provided in Section B.5.

We partition  $\mathcal{S}$  into  $\mathcal{S} = \bigcup_{i=1}^5 \mathcal{S}_i$  (which will be further discussed in Section 3.3.1 in detail) and randomly assign a distinct fact-implication pair (or equivalently, a  $(b_i, c_i)$  pair) to each subset  $\mathcal{S}_i$ . We then create training and test sets for each subset by splitting its subjects with a 0.2 training ratio, resulting in 20% training subjects and 80% test subjects. The training set contains facts for all subjects and implications only for training subjects. We then evaluate the model on the implications of the test subjects. Similar to [39], we use the mean-rank as our evaluation metrics. This is the average rank of the ground-truth implication among all possible candidates in  $\mathcal{A}_1 \cup \mathcal{A}_2$ , sorted by prediction probability. A lower mean-rank indicates better performance.

**Results.** Table 3.1 presents the evaluation results for predicting implications for test subjects. The findings reveal a “double-edged sword” characteristic of OCR. On the one hand, when a fact and an implication are causally related, the models exhibit strong generalization, consistent with [39]. On the other hand, this same associative ability makes them prone to hallucination, as they also tend to learn to connect concepts that have no causal relationship.

This behavior is remarkably efficient. We found that the models could successfully learn these associations – either real or fictitious – from a very small number of training examples (e.g., four training subjects in each subset). This suggests that the capability for strong generalization and the vulnerability to hallucination may stem from the same underlying learning mechanism. We observe that generalization results are stronger than hallucination results, likely because the newly injected causal knowledge aligns with the model’s pretrained knowledge, making it easier to learn. We note that the dual nature of generalization and hallucination is also empirically founded in [103]. Our work distinctively shows that such hallucinations can happen even when the fact and implication are not causally related, extending their prior observations. In Section B.4, we further verify our findings in real-world data.

### 3.3 One-Layer Attention-Only Transformers can Do Symbolic OCR

#### 3.3.1 Setup

**Basic notations.** For any integer  $N > 0$ , we use  $[N]$  to denote the set  $\{1, 2, \dots, N\}$ . Let  $\mathcal{V} = [M]$  be the vocabulary of size  $M = |\mathcal{V}|$ . We use lower-case and upper-case bold letters

Table 3.1: Performance comparison of different language models on synthetic reasoning tasks with various associations. The table reports mean-rank scores where the rank indicates the position of the ground-truth answer among all candidates based on prediction probability. Lower ranks indicate better performance and Rank 0 refers to the token with the largest probability. Values in parentheses indicate the standard error of the mean-rank scores, calculated from 3 runs with different random seeds.

| Models                 | Generalization | Hallucination      |              |                  |             |
|------------------------|----------------|--------------------|--------------|------------------|-------------|
|                        | City–Language  | City–Language (CF) | Country–Code | Profession–Color | Sport–Music |
| <b>Gemma-2-9B</b>      | 0.00 (0.00)    | 0.19 (0.20)        | 0.19 (0.07)  | 1.64 (0.01)      | 0.56 (0.01) |
| <b>OLMo-7B</b>         | 0.07 (0.03)    | 1.33 (0.49)        | 0.15 (0.13)  | 1.84 (0.23)      | 0.17 (0.01) |
| <b>Qwen-2-7B</b>       | 0.13 (0.01)    | 4.55 (2.33)        | 3.63 (1.10)  | 0.82 (0.34)      | 0.40 (0.08) |
| <b>Mistral-7B-v0.3</b> | 0.00 (0.00)    | 2.10 (0.01)        | 1.48 (0.52)  | 1.15 (0.56)      | 1.28 (0.13) |
| <b>Llama-3-8B</b>      | 0.00 (0.00)    | 1.18 (0.61)        | 0.77 (0.10)  | 0.93 (0.21)      | 0.63 (0.22) |

(e.g.,  $\mathbf{a}$ ,  $\mathbf{A}$ ) to represent vectors and matrices. Let  $\mathbf{e}_i \in \mathbb{R}^M$  be a one-hot vector, i.e., the  $i$ -th entry of  $\mathbf{e}_i$  is 1 while others are zero.

**Task structures.** Let  $\mathcal{S}$  be a set of subject tokens and  $\mathcal{R} := \{r_1, r_2\}$  be a set of relation tokens. Let  $\mathcal{A}$  be the set of answer tokens and  $a^* : \mathcal{S} \times \mathcal{R} \rightarrow \mathcal{A}$  be the mapping from subject-relation tuples  $(s, r)$  to the corresponding answer  $a^*(s, r)$ <sup>1</sup>. In Figure 3.1,  $\mathcal{S}$  is taken to be a list of names and  $\mathcal{R}$  corresponds to “lives in” and “speaks” respectively. We split the answers into two disjoint subsets:

$$\mathcal{A}_1 = \{b_1, \dots, b_n\}, \quad \mathcal{A}_2 = \{c_1, \dots, c_n\}, \quad \mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2,$$

where  $\mathcal{A}_1$  is the set corresponding to fact answers,  $\mathcal{A}_2$  is the set corresponding to implication answers, and  $|\mathcal{A}_1| = |\mathcal{A}_2| = n$ . Finally, we assume a one-to-one correspondence from  $b_i$  to  $c_i$  for any  $i \in [n]$ , such that whenever  $a^*(s, r_1) = b_i$ , it also holds that  $a^*(s, r_2) = c_i$ .

**Dataset constructions.** Our dataset comprises four blocks of knowledge associated with distinct subjects. Let  $\mathcal{S} = \mathcal{S}_{\text{train}} \cup \mathcal{S}_{\text{test}}$  where  $\mathcal{S}_{\text{train}}$  and  $\mathcal{S}_{\text{test}}$  are disjoint.

1. **Facts in  $\mathcal{S}_{\text{train}}$ :**  $\mathcal{D}_{\text{train}}^{(b)} = \{(s, r_1, b) : s \in \mathcal{S}_{\text{train}}\}$ ;
2. **Implications in  $\mathcal{S}_{\text{train}}$ :**  $\mathcal{D}_{\text{train}}^{(c)} = \{(s, r_2, c) : s \in \mathcal{S}_{\text{train}}\}$ ;
3. **Facts in  $\mathcal{S}_{\text{test}}$ :**  $\mathcal{D}_{\text{test}}^{(b)} = \{(s, r_1, b) : s \in \mathcal{S}_{\text{test}}\}$ ;
4. **Implications in  $\mathcal{S}_{\text{test}}$ :**  $\mathcal{D}_{\text{test}}^{(c)} = \{(s, r_2, c) : s \in \mathcal{S}_{\text{test}}\}$ .

<sup>1</sup>We consider a many-to-one mapping here where multiple  $(s, r)$  can correspond to the same answer. For example,  $s_1 = \text{“Alice”}$ ,  $s_2 = \text{“Bob”}$ ,  $r = \text{“lives in”}$ ,  $a^*(s_1, r) = a^*(s_2, r) = \text{“France”}$ .

We construct the training and test data with

$$\mathcal{D}_{\text{train}} = \mathcal{D}_{\text{train}}^{(b)} \cup \mathcal{D}_{\text{train}}^{(c)} \cup \mathcal{D}_{\text{test}}^{(b)}, \quad \mathcal{D}_{\text{test}} = \mathcal{D}_{\text{test}}^{(c)}.$$

**Subject enumeration and tokenization.** For any given subject set  $\mathcal{S}$ ,  $\mathcal{S}_{\text{train}}$ , or  $\mathcal{S}_{\text{test}}$ , we partition the subjects into  $n$  disjoint subsets based on their corresponding value of  $a^*(s, r)$ . For every  $i \in [n]$ , we assign the subjects in  $\mathcal{S}_i$  with fact  $b_i$  and implication  $c_i$ . We assume these partitions are equally sized. Specifically, we partition the training set as  $\mathcal{S}_{\text{train}} = \bigcup_{i=1}^n \mathcal{S}_{i,\text{train}}$  where  $|\mathcal{S}_{i,\text{train}}| = m_{\text{train}}$  for all  $i$ . Similarly, we partition the test set as  $\mathcal{S}_{\text{test}} = \bigcup_{i=1}^n \mathcal{S}_{i,\text{test}}$  where  $|\mathcal{S}_{i,\text{test}}| = m_{\text{test}}$  for all  $i$ . The complete subject set follows as  $\mathcal{S} = \bigcup_{i=1}^n \mathcal{S}_i$  where  $\mathcal{S}_i = \mathcal{S}_{i,\text{train}} \cup \mathcal{S}_{i,\text{test}}$  and  $|\mathcal{S}_i| = m = m_{\text{train}} + m_{\text{test}}$  for all  $i$ .

Using this partition structure, we can enumerate all subjects in  $\mathcal{S}$  as  $\{s_{i,j} : i \in [n], j \in [m]\}$ . Within each partition  $\mathcal{S}_i$ , the first  $m_{\text{train}}$  subjects belong to the training set ( $s_{i,j} \in \mathcal{S}_{i,\text{train}}$  for  $1 \leq j \leq m_{\text{train}}$ ), while the remaining subjects belong to the test set ( $s_{i,j} \in \mathcal{S}_{i,\text{test}}$  for  $m_{\text{train}} + 1 \leq j \leq m_{\text{train}} + m_{\text{test}}$ ). We order the subjects by cycling through partitions for each  $j$ :  $s_{1,1}, s_{2,1}, \dots, s_{n,1}, s_{1,2}, s_{2,2}, \dots, s_{n,2}, \dots, s_{1,m}, s_{2,m}, \dots, s_{n,m}$ . Each subject  $s_{i,j}$  is then tokenized according to this order.

**Sequence structures.** We take the vocabulary to be  $\mathcal{V} := \mathcal{S} \cup \mathcal{R} \cup \mathcal{A} \cup \{\langle \text{EOS} \rangle\}$  where  $\langle \text{EOS} \rangle$  is the “end-of-sequence” token. Each sequence has the form  $z_{1:(T+1)} = [s, r, \langle \text{EOS} \rangle, a^*(s, r)]$ . By default, the task is to predict  $z_{T+1}$  from  $z_{1:T}$  as illustrated in Figure 3.1.

**Transformer architectures.** We consider a decoder-only transformer which maps a length  $T$  sequence  $z_{1:T} := [z_1, \dots, z_T] \in \mathcal{V}^T$  to a  $d$ -dimensional vector which is used to generate the next token  $z_{T+1}$ . For any token  $z \in [M]$ , we also use the corresponding one-hot vector  $\mathbf{z} = \mathbf{e}_z \in \mathbb{R}^M$  to represent it and thus we can define  $\mathbf{X} = [\mathbf{e}_{z_1}, \mathbf{e}_{z_2}, \dots, \mathbf{e}_{z_T}]^\top \in \mathbb{R}^{T \times M}$  where the  $i$ -th row of  $\mathbf{X}$   $\mathbf{x}_i = \mathbf{e}_{z_i}$  for  $i \in [T]$ . We take the hidden dimension to be the same as the vocabulary size, i.e.,  $d = M$ . Throughout the work, we consider a one-layer linear attention model following [88, 97, 161].

$$\text{Factorized model: } f_{\boldsymbol{\theta}}(\mathbf{X}) = \mathbf{W}_O \mathbf{W}_V^\top \mathbf{X}^\top \mathbf{X} \mathbf{W}_{\text{KQ}} \mathbf{x}_T \in \mathbb{R}^d, \quad (3.1)$$

where  $\mathbf{W}_O, \mathbf{W}_V \in \mathbb{R}^{d \times d_h}$  are the output and value matrices, respectively, and we reparameterize the key-query matrices by  $\mathbf{W}_{\text{KQ}} = \mathbf{W}_K \mathbf{W}_Q^\top \in \mathbb{R}^{d \times d}$  in line with [133, 170]. We denote by  $\boldsymbol{\theta} = (\mathbf{W}_{\text{KQ}}, \mathbf{W}_O, \mathbf{W}_V)$  the summary of model parameters. Additionally, we consider a non-factorized model:  $\hat{\boldsymbol{\theta}} = (\mathbf{W}_{\text{KQ}}, \mathbf{W}_{\text{OV}})$  by further combining the output and value matrices as  $\mathbf{W}_{\text{OV}} = \mathbf{W}_O \mathbf{W}_V^\top$ .

$$\text{Non-factorized model: } f_{\hat{\boldsymbol{\theta}}}(\mathbf{X}) = \mathbf{W}_{\text{OV}} \mathbf{X}^\top \mathbf{X} \mathbf{W}_{\text{KQ}} \mathbf{x}_T \in \mathbb{R}^d. \quad (3.2)$$

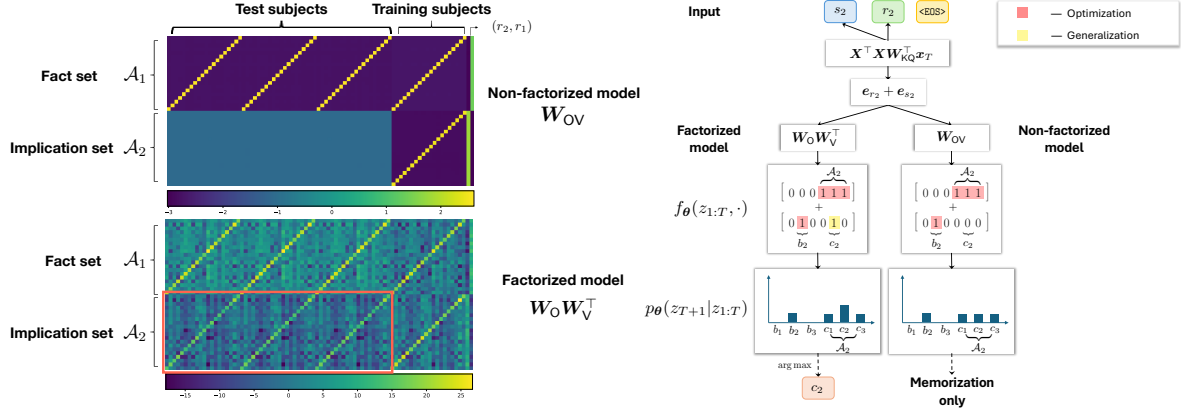


Figure 3.2: **The weights and mechanisms of the trained one-layer attention models.** The heatmaps on the left show that the factorized model (*bottom*) learns a structured weight matrix that enables OCR, as highlighted by the red box. The non-factorized model (*top*) fails to learn this structure. Here, the weights shown are the partial weights in the output-value matrix related to the prediction, i.e., we show a reduced matrix  $\mathbf{W}_{OV} \in \mathbb{R}^{|\mathcal{A}| \times (mn+2)}$ . The diagram on the right illustrates how this structural difference leads to different outcomes. The task is to predict  $c_2 \in \mathcal{A}_2$  given input  $z_{1:T}$  with  $(s_2, r_2)$ , where the atomic knowledge  $(s_2, r_2, c_2)$  is not included in the training set.

**Loss functions.** Let  $p_\theta(z|z_{1:T})$  be the next-token prediction probability, i.e.,

$$p_\theta(z|z_{1:T}) := \frac{\exp(\mathbf{e}_z^\top \mathbf{f}_\theta(\mathbf{X}))}{\sum_{z' \in \mathcal{A}} \exp(\mathbf{e}_{z'}^\top \mathbf{f}_\theta(\mathbf{X}))} = \frac{\exp(\mathbf{f}_\theta(z_{1:T}, z))}{\sum_{z' \in \mathcal{A}} \exp(\mathbf{f}_\theta(z_{1:T}, z'))}, \quad (3.3)$$

where we denote  $\mathbf{f}_\theta(z_{1:T}, a) = \mathbf{e}_a^\top \mathbf{f}_\theta(\mathbf{X})$  as the logit of token  $a$  for  $a \in \mathcal{A}$ . We also use  $\mathbf{f}_\theta((s, r), a)$  to represent the logit if  $(s, r) \in z_{1:T}$ . We consider training the model with cross-entropy loss

$$\mathcal{L}_{\text{train}}(\theta) = \mathbb{E}_{z_{1:T+1} \sim \mathcal{D}_{\text{train}}} [-\log p_\theta(z_{T+1}|z_{1:T})], \quad (3.4)$$

which we optimize by running gradient flow, i.e.,  $\dot{\theta} = -\nabla \mathcal{L}_{\text{train}}(\theta)$ . We omit the subscript and use  $\mathcal{L}(\theta) := \mathcal{L}_{\text{train}}(\theta)$  when the context is clear. Finally, we evaluate the model on the test set  $\mathcal{D}_{\text{test}}$  using the same loss function

$$\mathcal{L}_{\text{test}}(\theta) = \mathbb{E}_{z_{1:T+1} \sim \mathcal{D}_{\text{test}}} [-\log p_\theta(z_{T+1}|z_{1:T})]. \quad (3.5)$$

The corresponding definitions for the non-factorized model are obtained by substituting  $\theta$  with  $\tilde{\theta}$ .

### 3.3.2 Experiments and Observations

**Training and test results.** We compare the factorized model (3.1) and non-factorized model (3.2) by training both models using orthogonal embeddings with  $|\mathcal{S}| = 80, n = 20, m =$

4,  $m_{\text{train}} = 1$ , and  $d = d_h = 128$ . We use (3.4) as the training loss and (3.5) as the test loss. Both models achieve zero training loss. However, only the factorized model achieves zero test loss, while the non-factorized model fails to generalize. Further experimental details are available in Appendix B.3 where we provide the training and test loss curves (Figure B.1) and demonstrate that the factorized model generalizes effectively even with the intrinsic dimension as small as  $d_h = 4$  (Figure B.4).

**Mechanism analysis.** Figure 3.2 (left) visualizes the learned weights  $\mathbf{W}_{\text{OV}}$  and  $\mathbf{W}_{\text{O}}\mathbf{W}_{\text{V}}^\top$  after training. The non-factorized model learns zero weights in the “test-implication” block of the output-value matrix, whereas the factorized model exhibits similar weight patterns across both training and test blocks. The right side of Figure 3.2 illustrates the underlying mechanism, showing how the factorized architecture solves OCR through generalization while the non-factorized parameterization can only memorize the training data.

## 3.4 Theoretical Results

In this section, we conduct a detailed theoretical analysis to unveil the distinction in optimizing the one-layer attention model with two different parameterizations. We begin by assuming a fixed attention pattern and then extend to trainable  $\mathbf{W}_{\text{KQ}}$  matrices. Our main finding is that the factorized  $(\mathbf{W}_{\text{O}}, \mathbf{W}_{\text{V}})$  matrix induces implicit regularization with the nuclear norm, which prevents the “test-implication” block from collapsing to zero weights, thereby enabling OCR capabilities.

### 3.4.1 Implicit Bias Explains the Distinction in OCR Abilities

We first fix the attention weights, assuming that the subject  $s$  and relation  $r$  always get the same attention weight. The trainable parameters in the two models become  $\boldsymbol{\theta} = (\mathbf{W}_{\text{O}}, \mathbf{W}_{\text{V}})$  and  $\tilde{\boldsymbol{\theta}} = \mathbf{W}_{\text{OV}}$ . For the logit function given any input  $(s, r)$  and  $a \in \mathcal{A}$ , we have

$$f_{\boldsymbol{\theta}}((s, r), a) = [\mathbf{W}_{\text{O}}\mathbf{W}_{\text{V}}^\top](a, s) + [\mathbf{W}_{\text{O}}\mathbf{W}_{\text{V}}^\top](a, r) \text{ and } f_{\tilde{\boldsymbol{\theta}}}((s, r), a) = \mathbf{W}_{\text{OV}}(a, s) + \mathbf{W}_{\text{OV}}(a, r).$$

Despite their different parameterizations, the factorized model  $(\mathbf{W}_{\text{O}}, \mathbf{W}_{\text{V}})$  and non-factorized model  $\mathbf{W}_{\text{OV}}$  have identical expressivity. Proposition 3 formalizes this equivalence.

**Proposition 3** (Equivalent expressivity for  $(\mathbf{W}_{\text{O}}, \mathbf{W}_{\text{V}})$  and  $\mathbf{W}_{\text{OV}}$ ). *Suppose  $d_h \geq d$ . The factorized parameterization  $\boldsymbol{\theta} = (\mathbf{W}_{\text{O}}, \mathbf{W}_{\text{V}})$  with  $\mathbf{W}_{\text{O}}, \mathbf{W}_{\text{V}} \in \mathbb{R}^{d \times d_h}$  has equivalent expressive power to the non-factorized parameterization  $\tilde{\boldsymbol{\theta}} = \mathbf{W}_{\text{OV}}$  with  $\mathbf{W}_{\text{OV}} \in \mathbb{R}^{d \times d}$ . Specifically, for any factorized model  $\boldsymbol{\theta}$ , there exists an equivalent non-factorized model  $\tilde{\boldsymbol{\theta}}$ , and vice versa, such that they yield identical training and test losses as defined in (3.4) and (3.5).*

The proof is provided in Section B.1.1. Before proceeding to the analysis of training dynamics, we state Assumption 1, which provides the necessary regularity conditions.

**Assumption 1.** We assume the following conditions:

**2.1 Regularity:** For any fixed  $z_{1:T}$ , the logit function  $f_{\theta}(z_{1:T}, \cdot) \in \mathbb{R}^{|A|}$  is locally Lipschitz and differentiable, which means that for every  $\mathbf{x}_0$  in its domain, there exists a neighborhood  $N(\mathbf{x}_0)$  such that  $f_{\theta}(\mathbf{x})$  is Lipschitz continuous when restricted to  $N(\mathbf{x}_0)$ .

**2.2 Separability:** When optimizing either the non-factorized model  $\mathbf{W} = \mathbf{W}_{\text{OV}}$  or factorized model  $\mathbf{W} = (\mathbf{W}_{\text{O}}, \mathbf{W}_{\text{V}})$ , there exists time  $t_0$  such that  $\mathcal{L}(\mathbf{W}(t_0)) < 1$ .

**Remark 1.** Assumption 2.1 holds for both parameterizations: the non-factorized model  $\tilde{\theta}$  has a linear logit function, hence is Lipschitz; the factorized model has a bilinear logit function that is locally Lipschitz when  $\theta$  is bounded (following Lemma 29). Assumption 2.2 holds when  $d, d_h \geq 3$  by extending Theorem 5 from [97].

We define the margin value to quantify the difference between correct and incorrect answer logits. Given a model with parameter  $\mathbf{W}$  and any  $(s, r)$  pair, let  $a^*(s, r)$  denote the correct answer token. For any incorrect answer token  $a' \in \mathcal{A} \setminus \{a^*(s, r)\}$ , the margin between  $a^*(s, r)$  and  $a'$  is:

$$h_{(s,r),a'}(\mathbf{W}) = f_{\mathbf{W}}((s, r), a^*(s, r)) - f_{\mathbf{W}}((s, r), a'). \quad (3.6)$$

For instance, when the model outputs logits  $f_{\mathbf{W}}((s, r), \cdot) = [0, \dots, 1, 0, \dots, 0]^{\top} \in \mathbb{R}^{|A|}$  with only the  $a^*(s, r)$  entry equal to 1, we have  $h_{(s,r),a'}(\mathbf{W}) = 1$  for any  $a' \in \mathcal{A} \setminus \{a^*(s, r)\}$ . Given training loss (3.4), Theorem 5 builds the connection between model weights  $\mathbf{W}$  and solutions to an SVM problem.

**Theorem 5** (SVM forms). Let  $\mathbf{W}^*$  be an optimal solution of  $(\mathbf{W}_{\text{OV}}^{\text{F}}\text{-SVM})$  with  $\text{rank}(\mathbf{W}^*) = r$ . Assume  $d_h \geq r$ . Consider gradient descent with a small enough learning rate or gradient flow on the training loss (3.4). We have:

1. For factorized models with  $\theta = (\mathbf{W}_{\text{O}}, \mathbf{W}_{\text{V}})$ , any limit point of  $\theta / \|\theta\|_2$  is along the direction of a KKT point of a program which has the same solutions for  $\mathbf{W}_{\text{OV}}^{\text{F}} := \mathbf{W}_{\text{O}} \mathbf{W}_{\text{V}}^{\top}$  as the following program, where  $\|\cdot\|_{\star}$  denotes the nuclear norm:

$$\min_{\mathbf{W}_{\text{OV}}^{\text{F}}} \frac{1}{2} (\|\mathbf{W}_{\text{OV}}^{\text{F}}\|_{\star}^2) \quad \text{s.t.} \quad h_{(s,r),a'}(\mathbf{W}_{\text{OV}}^{\text{F}}) \geq 1, \forall (s, r) \in \mathcal{D}_{\text{train}}, \forall a' \in \mathcal{A} \setminus \{a^*(s, r)\}. \quad (\mathbf{W}_{\text{OV}}^{\text{F}}\text{-SVM})$$

2. For non-factorized models  $\mathbf{W}_{\text{OV}}$ , any limit point of  $\mathbf{W}_{\text{OV}} / \|\mathbf{W}_{\text{OV}}\|_F$  is along the direction of a global minimum of the following SVM problem, where  $\|\cdot\|_F$  denotes the Frobenius norm:

$$\min_{\mathbf{W}_{\text{OV}}} \frac{1}{2} (\|\mathbf{W}_{\text{OV}}\|_F^2) \quad \text{s.t.} \quad h_{(s,r),a'}(\mathbf{W}_{\text{OV}}) \geq 1, \forall (s, r) \in \mathcal{D}_{\text{train}}, \forall a' \in \mathcal{A} \setminus \{a^*(s, r)\}. \quad (\mathbf{W}_{\text{OV}}\text{-SVM})$$

Interestingly, training the factorized model leads to an SVM problem minimizing the nuclear norm, while the non-factorized model leads to the Frobenius norm. The proof is deferred to Section B.1.2. Heuristically, Theorem 5 is an example of the implicit bias of the gradient descent. ( $\mathbf{W}_{\text{OV-SVM}}$ ) could be derived directly from the homogeneous property of one-layer models. The objective of ( $\mathbf{W}_{\text{OV-SVM}}^{\text{F}}$ ) initially has the form  $\min_{\mathbf{W}_O, \mathbf{W}_V} (\|\mathbf{W}_O\|_F^2 + \|\mathbf{W}_V\|_F^2)/2$ . Using the connection between the nuclear norm and Frobenius norm that  $\|\mathbf{W}_{\text{OV}}^{\text{F}}\|_*^2 = \min_{\{\mathbf{W}_O \mathbf{W}_V^{\top} = \mathbf{W}_{\text{OV}}^{\text{F}}\}} (\|\mathbf{W}_O\|_F^2 + \|\mathbf{W}_V\|_F^2)/2$ , we could derive ( $\mathbf{W}_{\text{OV-SVM}}^{\text{F}}$ ) as proved in Theorem 5.

More surprisingly, the SVM problems in Theorem 5 have closed form solutions. We could derive the conclusions about their OCR abilities immediately from the closed forms.

**Theorem 6** (The OCR abilities of the factorized and non-factorized models). *Let  $n > 1$ .*

- Suppose  $\mathbf{W}_{\text{OV}}^{\text{F}}$  is a solution to the SVM problem in ( $\mathbf{W}_{\text{OV-SVM}}^{\text{F}}$ ). We have that for any  $(s, r) \in \mathcal{D}_{\text{test}}$  and  $a' \in \mathcal{A} \setminus \{a^*(s, r)\}$ , given regularity conditions, it holds that

$$h_{(s,r),a'}(\mathbf{W}_{\text{OV}}^{\text{F}}) \geq \min\{\sqrt{m_{\text{train}}/m_{\text{test}}}, 1\}, \text{ indicating the OCR ability.} \quad (3.7)$$

- Suppose  $\mathbf{W}_{\text{OV}}$  is a solution to the SVM problem in ( $\mathbf{W}_{\text{OV-SVM}}$ ). We have that for any  $(s, r) \in \mathcal{D}_{\text{test}}$ , and any  $a' \in \mathcal{A}_2 \setminus \{a^*(s, r)\}$ , it holds that

$$h_{(s,r),a'}(\mathbf{W}_{\text{OV}}) = 0, \text{ indicating no OCR ability.} \quad (3.8)$$

The key reason behind Theorem 6 is the different nature between minimizing the nuclear norm and the Frobenius norm. To minimize the Frobenius norm, the weights tend to become zero on as more entries as possible, and the weights on  $(s, r) \in \mathcal{D}_{\text{test}}$  are completely untouched during training. A solution minimizing the Frobenius norm would zero out all entries for  $(s, r) \in \mathcal{D}_{\text{test}}$ , leading to  $h_{(s,r),a'}(\mathbf{W}_{\text{OV}}) = 0$  for any  $a' \in \mathcal{A}_2$ . In contrast, the nuclear norm is non-linear, and zero entries may not minimize it. We therefore get  $h_{(s,r),a'}(\mathbf{W}_{\text{OV}}^{\text{F}}) > 0$ . The proof is deferred to Section B.1.3.

Our result provides new insights. First, it is well known that transformers need at least two layers of attention to perform multi-hop reasoning [113, 114], while we show that one-layer self-attention can find a shortcut to circumvent this bottleneck under certain scenarios. Second, most of the past works [133, 170, 62, 52, 97] on theoretically understanding transformers apply the reparameterization  $\mathbf{W}_{\text{OV}} = \mathbf{W}_O \mathbf{W}_V^{\top}$  as it does not change the expressivity of the model. Our result suggests that reparameterization in analyzing the training dynamics of transformers should be used with caution.

**OCR is sample-efficient.** Note that in ( $\mathbf{W}_{\text{OV-SVM}}^{\text{F}}$ ), the lower bound of the margin of the test implication depends only on the ratio between  $m_{\text{train}}$  and  $m_{\text{test}}$ . More importantly, as long as that  $m_{\text{train}} > 0$ , we have  $h_{(s,r),a'}(\mathbf{W}_{\text{OV}}^{\text{F}}) > 0$  for any  $a' \in \mathcal{A} \setminus \{a^*(s, r)\}$ . While this explains the strong generalization capabilities, it also implies that even when two relations are

not causally related, the model can learn to associate the fact and implication easily, which leads to hallucination. This finding well explains why a few training samples are sufficient for LLMs to exhibit OCR in Section 3.2.

### 3.4.2 Dynamics Analysis with a Trainable Key-Query Matrix

We denote  $W_{\text{OV}}(a, z) = \mathbf{e}_a^\top \mathbf{W}_{\text{OV}} \mathbf{e}_z$  and  $W_{\text{KQ}}(z) = \mathbf{e}_z^\top \mathbf{W}_{\text{KQ}} \mathbf{e}_{\langle \text{EOS} \rangle}$  following [97]. We assume that both  $\mathbf{W}_{\text{OV}}$  and  $\mathbf{W}_{\text{KQ}}$  are trainable and show that the non-factorized model fails to generalize to test implications (Theorem 7) by analyzing the gradient flow trajectory.

**Assumption 2.** *Let  $\alpha > 0$ . We initialize the weights by setting  $W_{\text{OV}}(a, z) = \alpha$  and  $W_{\text{KQ}}(z) = \alpha\sqrt{|\mathcal{A}| + 1}$  for all  $a \in \mathcal{A}$ ,  $z \in \mathcal{V}$ .*

**Theorem 7.** *Suppose that  $|\mathcal{A}_2| > 1$  and Assumption 2 holds, and we use  $\mathcal{L}_{\text{test}}(\tilde{\theta}_t)$  in (3.5) to denote the test loss for the non-factorized model. For any  $t \geq 0$ , it holds that*

$$\mathcal{L}_{\text{test}}(\tilde{\theta}_t) = \mathbb{E}_{z_{1:T+1} \sim \mathcal{D}_{\text{test}}} [-\log p_{\tilde{\theta}_t}(z_{T+1} | z_{1:T})] \geq \log |\mathcal{A}_2| > 0.$$

The proof exploits parameter symmetry. Subjects can be partitioned based on  $a^*(s, r)$ :  $\mathcal{S}_{\text{train}} = \cup_{i=1}^n \mathcal{S}_{i,\text{train}}$  and  $\mathcal{S}_{\text{test}} = \cup_{i=1}^n \mathcal{S}_{i,\text{test}}$ , where partition  $i$  corresponds to fact  $b_i$  and implication  $c_i$ . Since all partitions are equal-sized, any two pairs  $(b_i, c_i)$  and  $(b_j, c_j)$  with  $i \neq j$  are interchangeable. Applying this symmetry to the optimization dynamics, we show that  $W_{\text{OV}}(a, s) = W_{\text{OV}}(a', s)$  for any  $a, a' \in \mathcal{A}_2$ . Consequently, on the test set, the non-factorized model assigns uniform probability across all answers in the implication set:  $p_{\tilde{\theta}_t}(a | s, r_2) = p_{\tilde{\theta}_t}(a' | s, r_2)$  for any  $a, a' \in \mathcal{A}_2$ . A complete proof is provided in Section B.2.

This result is consistent with the observation of [170], which shows that a reparameterized non-factorized one-layer attention-only model struggles to generalize unless the expected answer token follows the important token in the prompt in the training set. Extending this result to factorized models with trainable  $\mathbf{W}_{\text{KQ}}$  matrices introduces significant complexity due to higher-order interaction terms between parameters. We leave this comprehensive analysis for future work.

## 3.5 Conclusions

In this work, we study LLMs' generalization and hallucination when fine-tuned with new factual knowledge in a unified way and show that the above two behaviors are both due to the model's OCR ability. We carefully analyze a one-layer linear attention model and prove that the implicit bias of GD on the factorized model enables the model to obtain strong OCR abilities. Our theory establishes that LLMs can easily associate facts and implications based on co-occurrence, and thus can hallucinate when the co-occurrence does not reflect causality. As for future directions, it would be interesting to extend our theoretical analysis to multi-layer transformers, as well as effective methods to prevent this type of hallucination when injecting new factual knowledge into a model.

# Chapter 4

## Reasoning by Superposition via Continuous CoT

### 4.1 Introductions

Large language models (LLMs) have shown strong performance in many reasoning tasks, especially when empowered with chain-of-thought (CoT) [145] (e.g., hard problems like AIME and math proving). However, they also struggle with tasks that require more sophisticated reasoning capability [68], e.g., reasoning and planning problems of increasing scales [164, 148], even with CoT [134, 166].

It remains an open problem how to expand existing discrete CoT to solve more complex reasoning problems. Recently, [55] proposes COCONUT (chain-of-continuous-thought) that uses continuous latent thoughts for reasoning, showing empirical performance boost on synthetic tasks such as directed graph reachability (i.e., given a specification of a directed graph and one starting node, determine which candidate destination node is reachable), as well as strong performance on real-world math reasoning benchmarks such as GSM8K [29]. Interestingly, COCONUT shows preliminary results that continuous latent thought may store multiple candidate search frontiers simultaneously, before the final answer is reached. This is in sharp contrast with discrete CoT, in which each discrete thought token has to be sampled (or “realized”) before feeding into LLMs in an autoregressive manner. However, the expressive power and the mechanism of continuous thought still remain elusive and lack a deep understanding.

In this work, we explore the mechanism of COCONUT for the problem of *graph reachability*, i.e., whether there exists a path from given start and end nodes in a directed graph. The problem setting is general [154, 55, 166] and includes many important theoretical problems (e.g., Turing machine halting problem) and practical use cases (e.g., knowledge graph). Given this setting, we proved that a two-layer transformer with  $D$  steps of continuous thought can solve graph reachability for graphs of  $n$  vertices, where  $D < n$  is the graph’s diameter (longest path length between two nodes). In contrast, for graph reachability, the best existing

result on constant-depth transformers with discrete CoT requires  $O(n^2)$  steps [91].

Intuitively, in our construction, each latent thought vector is a superposition of multiple valid search traces, and thus can perform *implicit parallel search* on the graph in each autoregressive step. The continuous thoughts can be regarded as “superposition states” in quantum mechanics [17], storing multiple search frontiers simultaneously, and thus enabling efficient breadth-first search (BFS). In contrast, discrete thought tokens can be viewed as “collapsed states” from superpositions. This forces the model to choose a branch deterministically, yielding either an incorrect greedy search or a depth-first style search with backtracking, which requires more computation. Unlike previous theoretical work that constructs positional encodings specifically for a given problem or even for a given input length, our construction works for widely-used positional encodings in practice, such as sinusoidal positional encoding [138] and rotary position embedding [123].

Moreover, we show that our theoretical construction can be achieved in gradient-based training. Specifically, a two-layer transformer with continuous CoT outperforms a 12-layer one with discrete CoT on graph reachability. An inspection of attention patterns and their underlying representation demonstrates that the continuous thought indeed encodes multiple plausible search frontiers in parallel in superposition states. Notably, such a superpositional representation automatically *emerges* from training only with the optimal path of graph reachability, without strong supervision that aligns the latent thought vectors with other plausible search traces.

## 4.2 Preliminaries

**Basic notations.** For any integer  $N > 0$ , we use  $[N]$  to denote the set  $\{1, 2, \dots, N\}$ . For any finite set  $\mathcal{X}$ , let  $|\mathcal{X}|$  denote the cardinality of  $\mathcal{X}$ . Let  $\mathbb{R}$  be the set of real numbers. We use  $\mathbb{1}\{\cdot\}$  to denote the indicator function. Without further clarification, we use lower-case bold letters (e.g.,  $\mathbf{x}$ ,  $\boldsymbol{\theta}$ ) and upper-case bold letters (e.g.,  $\mathbf{W}$ ,  $\mathbf{U}$ ) to denote vectors and matrices, respectively. In particular, we use  $\mathbf{I}_d$  to denote a  $d \times d$  identity matrix, use  $\mathbf{0}_{m \times n}$  (or  $\mathbf{0}_m$ ) to denote an  $m \times n$  zero matrix (or an  $m$ -dimensional zero vector), and use  $\mathbf{e}_i$  to denote a one-hot vector of which the  $i$ -th entry is one and other entries are all zero, where the dimension of  $\mathbf{e}_i$  can be inferred from the context. We also use  $\|\cdot\|_\infty$  and  $\|\cdot\|_2$  to represent  $L_\infty$  norm and  $L_2$  norm of vectors, respectively. For vectors  $\mathbf{u} \in \mathbb{R}^m$  and  $\mathbf{v} \in \mathbb{R}^n$ , let  $\mathbf{u} \otimes \mathbf{v} = \mathbf{u}\mathbf{v}^\top \in \mathbb{R}^{m \times n}$  denote their outer product. Also, for vectors  $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$ , let  $\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^\top \mathbf{v}$  denote their inner product. For any vector  $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$ , we define the softmax function  $\text{SoftMax} : \mathbb{R}^d \rightarrow \mathbb{R}^d$  as  $\text{SoftMax}(\mathbf{x})_i = \exp(x_i) / (\sum_{j=1}^d \exp(x_j))$ , and the layer normalization operator  $\text{LayerNorm}(\mathbf{x}) = \mathbf{x} / \|\mathbf{x}\|_2$ . Moreover, for a sequence of vectors  $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t) \in \mathbb{R}^{d \times t}$ , we abuse the notation  $\text{LayerNorm}(\mathbf{x}_1, \dots, \mathbf{x}_t) = (\text{LayerNorm}(\mathbf{x}_1), \dots, \text{LayerNorm}(\mathbf{x}_t)) \in \mathbb{R}^{d \times t}$ .

**Tokens and embeddings.** For a fixed positive integer  $V > 0$ , let  $\text{Voc} = [V]$  denote a vocabulary of size  $V$ . For each token  $v \in \text{Voc}$ , there is an associated token embedding  $\mathbf{u}_v \in \mathbb{R}^d$  where  $d > 0$  is the embedding dimension. Assume  $d = 3d_{\text{TE}} + d_{\text{PE}}$ . We refer

to the first  $d_{\text{TE}}$  entries of a  $d$ -dimensional vector as its content, the subsequent  $d_{\text{TE}}$  entries as its first buffer, the following  $d_{\text{TE}}$  entries as its second buffer, and the final  $d_{\text{PE}}$  entries as its effective positional encoding. Formally, for a vector  $\mathbf{x} = (x_1, x_2, \dots, x_d)^\top \in \mathbb{R}^d$ , we define  $\text{content}(\mathbf{x}) = (x_1, \dots, x_{d_{\text{TE}}})^\top$ ,  $\text{buffer}_1(\mathbf{x}) = (x_{d_{\text{TE}}+1}, \dots, x_{2d_{\text{TE}}})^\top$ ,  $\text{buffer}_2(\mathbf{x}) = (x_{2d_{\text{TE}}+1}, \dots, x_{3d_{\text{TE}}})^\top$  and  $\text{pos}(\mathbf{x}) = (x_{3d_{\text{TE}}+1}, \dots, x_d)^\top$ . Let  $\tilde{\mathbf{u}}_v = \text{content}(\mathbf{u}_v) \in \mathbb{R}^{d_{\text{TE}}}$  for any  $v \in \text{Voc}$ . Assume for each  $\mathbf{u}_v$ , only the first  $d_{\text{TE}}$  entries are non-zero. Furthermore, let  $\mathbf{U} = [\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2, \dots, \tilde{\mathbf{u}}_V] \in \mathbb{R}^{d_{\text{TE}} \times V}$  and we assume that  $\mathbf{U}^\top \mathbf{U} = \mathbf{I}_V$ , i.e., the token embeddings are orthonormal.

---

**Algorithm 1** Transformer (TF)

---

**Input:** Parameters  $\theta = (\theta_{\text{PE}}, \{\theta_{\text{Attn}}^{(l,h)}\}_{l=0, h=0}^{L-1, H_l-1}, \{\theta_{\text{MLP}}^{(l)}\}_{l=0}^{L-1})$ , input embeddings  $\hat{\mathbf{v}} = (\hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_t)$

- 1:  $\hat{\mathbf{v}}_i^{(0)} \leftarrow \hat{\mathbf{v}}_i + \text{PosEncode}_{\theta_{\text{PE}}}(i), \quad \forall i \in [t]$  ▷ adding positional encoding
- 2: **for**  $l = 0$  to  $L - 1$  **do**
- 3:  $\hat{\mathbf{v}}^{(l+0.5)} \leftarrow \hat{\mathbf{v}}^{(l)} + \sum_{h=0}^{H_l-1} \text{Attn}_{\theta_{\text{Attn}}^{(l,h)}}(\hat{\mathbf{v}}^{(l)})$  ▷ self attention
- 4:  $\hat{\mathbf{v}}^{(l+1)} \leftarrow \text{LayerNorm} \left( \text{MLP}_{\theta_{\text{MLP}}^{(l)}}(\hat{\mathbf{v}}^{(l+0.5)}) \right)$  ▷ MLP and layer normalization
- 5: **end for**

**Output:** Embedding of the last layer at the last position  $\hat{\mathbf{v}}_t^{(L)}$ .

---

**Transformer architectures.** An  $L$ -layer autoregressive transformer receives a sequence of input embeddings  $\hat{\mathbf{v}} = \hat{\mathbf{v}}_{[t]} \triangleq (\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \dots, \hat{\mathbf{v}}_t) \in \mathbb{R}^{d \times t}$  and outputs  $\text{TF}_\theta(\hat{\mathbf{v}}) \in \mathbb{R}^d$  where  $\text{TF}_\theta(\cdot)$  is defined in Algorithm 1. Let  $\mathbf{W}_\mathbf{O} \in \mathbb{R}^{V \times d}$  be the decoding matrix. A traditional decoder will sample the next token  $v_{t+1} \sim \text{SoftMax}(\mathbf{W}_\mathbf{O} \text{TF}_\theta(\hat{\mathbf{v}}))$  and append its token embedding in position  $t+1$ , i.e.,  $\hat{\mathbf{v}}_{t+1} = \mathbf{u}_{v_{t+1}}$ , to autoregressively generate subsequent outputs. When using the chain of continuous thought [55], one skips the sampling step and directly appends the output of the transformer as the input embedding of the next position, i.e.,  $\hat{\mathbf{v}}_{t+1} = \text{TF}_\theta(\hat{\mathbf{v}})$ . The parameter  $\theta$  contains positional encodings  $\theta_{\text{PE}}$ ,  $L$  attention layers where each layer  $l$  contains  $H_l$  heads  $\{\theta_{\text{Attn}}^{(l,h)}\}_{h=0}^{L-1, H_l-1}$  and  $L$  MLP layers  $\{\theta_{\text{MLP}}^{(l)}\}_{l=0}^{L-1}$ . The definitions of attention heads and MLPs are in Algorithm 2.

---

**Algorithm 2** Causal Self-Attention (Attn) and (position-wise) Multilayer Perceptron (MLP)

---

**Input:**  $\theta_{\text{Attn}} = (\mathbf{W}_\mathbf{Q}, \mathbf{W}_\mathbf{K}, \mathbf{W}_\mathbf{V}, \mathbf{W}_\mathbf{O}), \theta_{\text{MLP}} = (\{\mathbf{W}_i\}_{i=1}^{L_{\text{MLP}}}, \{\sigma_i(\cdot)\}_{i=1}^{L_{\text{MLP}}-1})$ , input  $\hat{\mathbf{v}} = (\hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_t)$

- 1:  $\mathbf{q}_i \leftarrow \mathbf{W}_\mathbf{Q} \hat{\mathbf{v}}_i, \quad \mathbf{k}_i \leftarrow \mathbf{W}_\mathbf{K} \hat{\mathbf{v}}_i, \quad \mathbf{v}_i \leftarrow \mathbf{W}_\mathbf{V} \hat{\mathbf{v}}_i, \quad \forall i \in [t]$  ▷ compute queries, keys, values
- 2: **for**  $i = 1$  to  $t$  **do**
- 3:  $s_i \leftarrow \text{SoftMax}(\langle \mathbf{q}_i, \mathbf{k}_1 \rangle, \dots, \langle \mathbf{q}_i, \mathbf{k}_i \rangle), \quad \hat{\mathbf{v}}_i^{\text{Attn}} \leftarrow \mathbf{W}_\mathbf{O} \sum_{j=1}^i s_{i,j} \mathbf{v}_j$
- 4:  $\hat{\mathbf{v}}_i^{\text{MLP}} \leftarrow \mathbf{W}_{L_{\text{MLP}}} \sigma_{L_{\text{MLP}}-1}(\dots \mathbf{W}_2 \sigma_1(\mathbf{W}_1 \hat{\mathbf{v}}_i) \dots)$
- 5: **end for**

**Output:**  $\text{Attn}_{\theta_{\text{Attn}}}(\hat{\mathbf{v}}) = (\hat{\mathbf{v}}_1^{\text{Attn}}, \dots, \hat{\mathbf{v}}_t^{\text{Attn}})$  or  $\text{MLP}_{\theta_{\text{MLP}}}(\hat{\mathbf{v}}) = (\hat{\mathbf{v}}_1^{\text{MLP}}, \dots, \hat{\mathbf{v}}_t^{\text{MLP}})$

---

**Positional encodings.** Given an input sequence  $(\hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_T)$ , for each position  $i \in [T]$ , there is a corresponding positional encoding  $\mathbf{p}_i = \text{PosEncode}_{\theta_{\text{PE}}}(i) \in \mathbb{R}^d$ . For each  $\mathbf{p}_i$ , we assume that only the last  $d_{\text{PE}}$  entries are non-zero and thus call  $d_{\text{PE}}$  the effective positional encoding dimension. For notation convenience, we denote  $\tilde{\mathbf{v}}_i = \text{content}(\hat{\mathbf{v}}_i) \in \mathbb{R}^{d_{\text{TE}}}$ ,  $\bar{\mathbf{p}}_i = \text{pos}(\mathbf{p}_i) \in \mathbb{R}^{d_{\text{PE}}}$  for any  $i \in [T]$ . We use the widely used sinusoidal positional encoding for  $\bar{\mathbf{p}}_i = (\bar{p}_{i,1}, \dots, \bar{p}_{i,d_{\text{PE}}})$  as defined below.

**Definition 1** (Positional Encoding). *Let  $d_{\text{PE}}$  be even. We use positional encoding generated by sinusoidal functions [138]. Specifically, for any position  $i \geq 1$  and any index  $j \in [d_{\text{PE}}/2]$ , we have*

$$\bar{p}_{i,2j-1} = \cos(i \cdot M^{-2j/d_{\text{PE}}}), \quad \bar{p}_{i,2j} = \sin(i \cdot M^{-2j/d_{\text{PE}}}),$$

where  $M > 0$  is a large constant integer, e.g.,  $M = 10^4$  as chosen in [138].

**Remark 2.** We also discuss theoretical construction with RoPE[123] (Section C.2.6).

### 4.3 Problem Formulations

**Graph reachability.** Let  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  be a directed graph, where  $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$  is the set of vertices and  $\mathcal{E} = \{e_1, e_2, \dots, e_m\}$  is the set of edges. Each vertex  $v_i \in \mathcal{V}$  corresponds to a token in the vocabulary, and thus we use  $v_i$  to represent both a vertex and its corresponding token. Let  $n_{\text{max}} > 0$  denote the maximum possible number of vertices of a graph. Note that  $n_{\text{max}} < |\text{Voc}|$ . Each edge  $e_i \in \mathcal{E}$  is a tuple, where  $e_i = (\mathbf{s}_i, \mathbf{t}_i) \in \mathcal{V} \times \mathcal{V}$  denotes there is a directed edge from the source node  $\mathbf{s}_i$  to the target node  $\mathbf{t}_i$ . Given a graph (i.e., all the edges of the graph), two candidate destination nodes  $\mathbf{c}_1$  and  $\mathbf{c}_2$ , and a root node  $\mathbf{r}$ , the task is to identify which of the two nodes can be reached by  $\mathbf{r}$ . Note that we guarantee one and only one of  $\mathbf{c}_1$  and  $\mathbf{c}_2$  is reachable by  $\mathbf{r}$ , which we denote by  $\mathbf{c}_i^*$ .

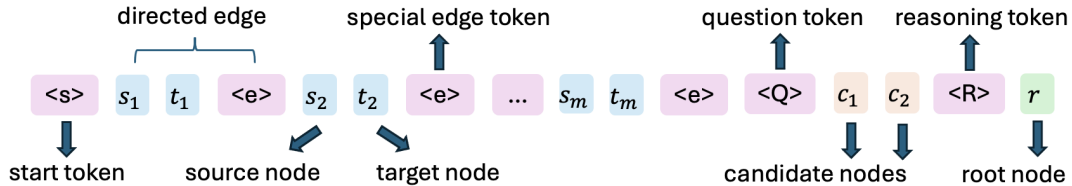


Figure 4.1: Prompt format of the graph reachability problem.

**Input structures.** The prompt structure is illustrated in Figure 4.1. The prompt starts with the BOS (beginning of sentence) token  $\langle s \rangle$ . The subsequent  $3m$  tokens represent  $m$  edges, where each edge is represented by the source node  $\mathbf{s}_i$ , target node  $\mathbf{t}_i$ , and a special

edge token  $\langle \mathbf{e} \rangle$  that marks the end of an edge. Then there is a special question token  $\langle \mathbf{Q} \rangle$  followed by two candidate destination nodes  $\mathbf{c}_1$  and  $\mathbf{c}_2$ . Finally, there is a special reasoning token  $\langle \mathbf{R} \rangle$  followed by a root node  $\mathbf{r}$ . See Table C.1 for the full list of token notations. Let  $t_0 = 3m + 6$  be the length of the prompt, and let  $(\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \dots, \hat{\mathbf{v}}_{t_0})$  be the input embedding sequence where  $\hat{\mathbf{v}}_i$  is the token embedding of the  $i$ -th token in the prompt.

**Chain of continuous thought.** We allow transformers to utilize continuous thoughts. Concretely, for  $c = 1, 2, \dots$ , the transformer autoregressively generates  $\hat{\mathbf{v}}_{t_0+c} = \text{TF}_\theta(\hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_{t_0+c-1})$ . For notation convenience, we use  $[\mathbf{t}_c] = \hat{\mathbf{v}}_{t_0+c}$  to represent the continuous thought of step  $c$  for  $c \geq 0$ , and thus  $[\mathbf{t}_0] = \mathbf{u}_\mathbf{r}$ . To request the transformer to make the final prediction after  $C$  steps of thoughts, one simply appends a special answer token  $\langle \mathbf{A} \rangle$  at the end of the sequence, i.e., sets  $\hat{\mathbf{v}}_T = \mathbf{u}_{\langle \mathbf{A} \rangle}$  where  $T = t_0 + C + 1$ , and gets the prediction sampled from  $\text{SoftMax}(\mathbf{W}_\mathbf{O} \text{TF}_\theta(\hat{\mathbf{v}}_{[T]}))$  or using greedy decoding  $\arg \max_{v \in \text{Voc}} \mathbf{W}_\mathbf{O} \text{TF}_\theta(\hat{\mathbf{v}}_{[T]})$ . We denote  $\widetilde{\text{TF}}_{\theta, \mathbf{W}_\mathbf{O}}(\hat{\mathbf{v}}_{[t_0]}) = \arg \max_{v \in \text{Voc}} \mathbf{W}_\mathbf{O} \text{TF}_\theta(\hat{\mathbf{v}}_{[T]})$  as the output token of greedy decoding after generating  $C$  steps of continuous thoughts.

**Position index.** To present the position of each token or continuous thought in the sequence in a clear way, we use  $\text{ldx}(v)$  to represent the position of a token in the input sequence (e.g.,  $\text{ldx}(\langle \mathbf{s} \rangle) = 1$ ,  $\text{ldx}(\mathbf{s}_i) = 3i - 1$ ,  $\text{ldx}(\langle \mathbf{Q} \rangle) = 3m + 2$ ), use  $\text{ldx}(\langle \mathbf{e} \rangle, i) = 3i + 1$  to represent the position of the  $i$ -th  $\langle \mathbf{e} \rangle$  token in the prompt, and use  $\text{ldx}([\mathbf{t}_i]) = t_0 + i$  to represent the position of the continuous thought of step  $i$ . See Table C.2 for the complete list of position indices.

In the following sections, we demonstrate that the chain of continuous thought can efficiently solve the graph reachability problem both theoretically (Section 4.4) and empirically (Section 4.5).

## 4.4 Theoretical Results

In this section, we theoretically prove that a two-layer transformer with continuous thought can efficiently solve the graph reachability problem. We first introduce a basic building block, the *attention chooser*, in our transformer constructions in Section 4.4.1. Then we present the key result that continuous thought maintains a superposition state of multiple search traces simultaneously in Section 4.4.2. We show our main theorem in Section 4.4.3 and make further discussion in Section 4.4.4.

### 4.4.1 Attention chooser

We use the attention chooser as a building block in our construction, which will choose the appropriate positions to attend conditioned on the token in the current position. This allows us to use the same parameter constructions for various input lengths. The proof is deferred to Section C.2.1.

**Lemma 3** (Attention chooser, informal version of Lemma 32). *Under sinusoidal positional encoding as defined in Definition 1, for any token  $\langle \mathbf{x} \rangle \in \text{Voc}$  and relative position  $\ell \geq 0$ , there exists a construction of  $\mathbf{W}_K, \mathbf{W}_Q \in \mathbb{R}^{(2d_{\text{PE}}) \times d}$ , such that for any input sequence  $\hat{\mathbf{v}}_{[T]}$  that satisfying mild assumptions (see Lemma 32 in Section C.2.2), it holds that for any position  $i \in [T]$ , it will pay almost all attention to position  $(i - \ell)$  if  $\hat{\mathbf{v}}_i = \mathbf{u}_{\langle \mathbf{x} \rangle}$ , and pay most attention to position one otherwise.*

*Proof sketch.* We define vector  $\tilde{\mathbf{u}}_{\langle \mathbf{x} \rangle} = \sum_{v \in \text{Voc} \setminus \{\langle \mathbf{x} \rangle\}} \tilde{\mathbf{u}}_v \in \mathbb{R}^{d_{\text{TE}}}$  as the superposition of all token embeddings in the vocabulary except for  $\langle \mathbf{x} \rangle$ . By the property of sinusoidal positional encoding, there exists a rotation matrix  $\mathbf{R}^{(\ell)}$  as in Lemma 33 in Section C.2.5, s.t.  $\bar{\mathbf{p}}_{i+\ell} = \mathbf{R}^{(\ell)} \bar{\mathbf{p}}_i$ ,  $\forall i \geq 1$ . Then we can construct the query and key matrices as

$$\mathbf{W}_Q = \begin{bmatrix} \mathbf{0}_{d_{\text{PE}} \times d_{\text{TE}}} & \mathbf{0}_{d_{\text{PE}} \times 2d_{\text{TE}}} & \mathbf{I}_{d_{\text{PE}}} \\ \xi \bar{\mathbf{p}}_1 \otimes \tilde{\mathbf{u}}_{\langle \mathbf{x} \rangle} & \mathbf{0}_{d_{\text{PE}} \times 2d_{\text{TE}}} & \mathbf{0}_{d_{\text{PE}} \times d_{\text{PE}}} \end{bmatrix}, \quad \mathbf{W}_K = \begin{bmatrix} \mathbf{0}_{d_{\text{PE}} \times 3d_{\text{TE}}} & \eta \mathbf{R}^{(\ell)} \\ \mathbf{0}_{d_{\text{PE}} \times 3d_{\text{TE}}} & \eta \mathbf{I}_{d_{\text{PE}}} \end{bmatrix},$$

where  $\xi, \eta > 0$  and thus the query and key vectors can be calculated as

$$\mathbf{q}_i = \mathbf{W}_Q(\hat{\mathbf{v}}_i + \mathbf{p}_i) = \begin{bmatrix} \bar{\mathbf{p}}_i \\ \xi \langle \tilde{\mathbf{u}}_{\langle \mathbf{x} \rangle}, \tilde{\mathbf{v}}_i \rangle \bar{\mathbf{p}}_1 \end{bmatrix}, \quad \mathbf{k}_i = \mathbf{W}_K(\hat{\mathbf{v}}_i + \mathbf{p}_i) = \begin{bmatrix} \eta \mathbf{R}^{(\ell)} \bar{\mathbf{p}}_i \\ \eta \bar{\mathbf{p}}_i \end{bmatrix} = \begin{bmatrix} \eta \bar{\mathbf{p}}_{i+\ell} \\ \eta \bar{\mathbf{p}}_i \end{bmatrix}.$$

Now for any  $1 \leq j \leq i \leq T$ , we have  $\langle \mathbf{q}_i, \mathbf{k}_j \rangle = \eta \left( \langle \bar{\mathbf{p}}_i, \bar{\mathbf{p}}_{j+\ell} \rangle + \xi \langle \tilde{\mathbf{u}}_{\langle \mathbf{x} \rangle}, \tilde{\mathbf{v}}_i \rangle \langle \bar{\mathbf{p}}_1, \bar{\mathbf{p}}_j \rangle \right)$ . Fix any  $i \in [T]$ . By the property of sinusoidal positional encoding,  $\langle \bar{\mathbf{p}}_1, \bar{\mathbf{p}}_{i'} \rangle$  is maximized when  $i' = i$  as in Lemma 34 in Section C.2.5. If  $\hat{\mathbf{v}}_i = \mathbf{u}_{\langle \mathbf{x} \rangle}$ , it holds that  $\langle \tilde{\mathbf{u}}_{\langle \mathbf{x} \rangle}, \tilde{\mathbf{v}}_i \rangle = 0$  and thus  $\langle \mathbf{q}_i, \mathbf{k}_j \rangle$  is determined by  $\langle \bar{\mathbf{p}}_i, \bar{\mathbf{p}}_{j+\ell} \rangle$ , which is maximized at  $j = i - \ell$ . If  $\hat{\mathbf{v}}_i \neq \mathbf{u}_{\langle \mathbf{x} \rangle}$ , one can show that  $\langle \tilde{\mathbf{u}}_{\langle \mathbf{x} \rangle}, \tilde{\mathbf{v}}_i \rangle \geq 1$  and thus  $\langle \mathbf{q}_i, \mathbf{k}_j \rangle$  is largely determined by  $\langle \bar{\mathbf{p}}_1, \bar{\mathbf{p}}_j \rangle$  when  $\xi$  is large, and thus maximized at  $j = 1$ .  $\square$

#### 4.4.2 Continuous thought maintains superposition states

Recall that  $\hat{\mathbf{v}} = \hat{\mathbf{v}}_{[t_0]}$  denotes the input sequence as defined in Section 4.3. We define  $\mathcal{V}_c$  as the set of vertices that are reachable from  $\mathbf{r}$  within  $c$  steps. Below, we present our key lemma.

**Lemma 4** (Continuous thought maintains superposition states). *We autoregressively generate  $[t_{c+1}] = \text{TF}_\theta(\hat{\mathbf{v}}_{[t_0]}, [t_1] \dots, [t_c])$  for any  $c \geq 0$ . There exists a construction of  $\theta$  such that*

$$[t_c] = \hat{\mathbf{v}}_{t_0+c} = \frac{1}{\sqrt{|\mathcal{V}_c|}} \sum_{v \in \mathcal{V}_c} \mathbf{u}_v, \quad (4.1)$$

*i.e., the  $c$ -th continuous thought is the normalized superposition of all vertices that can be reached from  $\mathbf{r}$  within  $c$  steps.*

Lemma 4 precisely characterizes that each continuous thought is a superposition of all reachable vertices. We provide a proof sketch below and defer the complete proof to Section C.2.2.

*Proof sketch.* We prove by induction. For  $c = 0$ , by definition,  $\mathcal{V}_0 = \{\mathbf{r}\}$  and  $[\mathbf{t}_0] = \mathbf{u}_{\mathbf{r}} = \frac{1}{\sqrt{|\mathcal{V}_0|}} \sum_{v \in \mathcal{V}_0} \mathbf{u}_v$ . Now we briefly show how to construct the two-layer transformer such that under the induction assumption that (4.1) holds for  $0, \dots, c$ , we can obtain that (4.1) also holds for  $c + 1$ .

**First layer attention.** The first attention layer contains five attention heads, and each head is an attention chooser as constructed in Lemma 3. Let  $h_k = (\langle \mathbf{x} \rangle, \ell)$  denote the  $k$ -th attention head that attends to position  $(i - \ell)$  when the  $i$ -th token is  $\langle \mathbf{x} \rangle$  and attends to the first token otherwise. We construct  $h_0 = (\langle \mathbf{e} \rangle, 2)$ ,  $h_1 = (\langle \mathbf{e} \rangle, 1)$ ,  $h_2 = (\langle \mathbf{R} \rangle, 2)$ ,  $h_3 = (\langle \mathbf{R} \rangle, 1)$ ,  $h_4 = (\langle \mathbf{A} \rangle, 1)$ , which is illustrated in Figure 4.2. For each head, the value matrix will read the value in the content space, and the output matrix will copy the value state to a designated space.

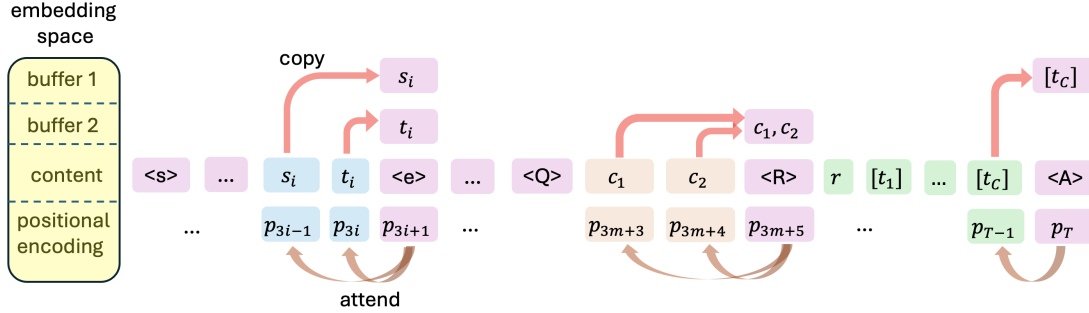


Figure 4.2: Illustration of the embedding space and first layer attention mechanism.

**Second layer attention.** In the second layer, we only need one attention head. Note that after the first layer, for the  $i$ -th edge token  $\langle \mathbf{e} \rangle$ , we have  $\text{buffer}_1(\hat{\mathbf{v}}_{\text{idx}(\langle \mathbf{e} \rangle, i)}) = \tilde{\mathbf{u}}_{s_i}$  and  $\text{buffer}_2(\hat{\mathbf{v}}_{\text{idx}(\langle \mathbf{e} \rangle, i)}) = \tilde{\mathbf{u}}_{t_i}$ . By the induction assumption, the current thought  $[\mathbf{t}_c]$  is a superposition of all vertices in  $\mathcal{V}_c$ . We construct the query and key matrices such that  $[\mathbf{t}_c]$  pays large attention to the  $i$ -th edge token  $\langle \mathbf{e} \rangle$  if  $s_i \in \mathcal{V}_c$  (roughly speaking, we can view  $\mathbf{q}_{\text{idx}([\mathbf{t}_c])} = [\mathbf{t}_c]$ ,  $\mathbf{k}_{\text{idx}(\langle \mathbf{e} \rangle, i)} = \mathbf{u}_{s_i}$ , and their inner product is positive iff  $s_i \in \mathcal{V}_c$ ), and add the target node  $t_i$  stored in buffer 2 back to the current thought (see Figure 4.3). This is exactly a one-step expansion of the currently explored vertices  $\mathcal{V}_c$  and thus the continuous thought at the next step  $(c + 1)$  will correspond to  $\mathcal{V}_{c+1}$ .

**MLP as filter for signals.** Note that after the attention layer, the weight of each node in the current thought is not uniform, and the current thought might contain noise tokens since the normalized attention score to each position is non-zero. We use the MLP layer to filter out the noise token and adjust the weight of each node in  $\mathcal{V}_{c+1}$ . Informally, for a superposition state  $\hat{\mathbf{v}} = \sum_{v \in \mathcal{V}_{\text{oc}}} \lambda_v \mathbf{u}_v$ , we want to eliminate noise tokens  $v$  where  $\lambda_v < \varepsilon$ ,

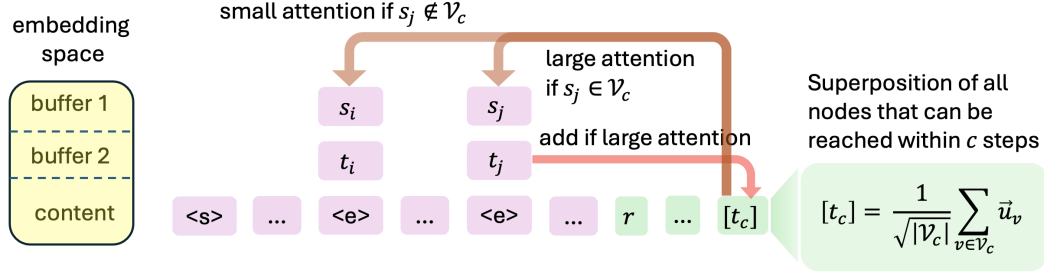


Figure 4.3: Illustration of the second layer attention mechanism for thought generation. We omit the positional encoding space since they are not used in the second layer.

and want to equalize the weights of other tokens. By setting the first layer parameter as  $\mathbf{W}_1 = [\mathbf{u}_1, \dots, \mathbf{u}_V]^\top$ , nonlinearity as  $\sigma(x) = \mathbb{1}\{x \geq \varepsilon\}$  and second layer as  $\mathbf{W}_2 = \mathbf{W}_1^\top$ , we have  $\mathbf{W}_2(\sigma(\mathbf{W}_1 \hat{\mathbf{v}})) = \sum_{v \in \text{Voc}} \mathbb{1}\{\lambda_v \geq \varepsilon\} \mathbf{u}_v$ , where  $\mathbf{W}_1$  rotates the basis  $\{\mathbf{u}_v\}$  to the standard basis  $\{\mathbf{e}_v\}$ ,  $\sigma(\cdot)$  serves as a coordinate-wise filter, and  $\mathbf{W}_2$  rotates the basis back. After layer normalization, we can obtain that (4.1) also holds for  $c + 1$ .  $\square$

#### 4.4.3 Measuring the superposition state as prediction

Since the continuous thought  $[t_c]$  is a superposition of all vertices in  $\mathcal{V}_c$ , as long as  $c_{i^*}$  can be reached by  $r$  within  $C$  steps, the superposition state  $[t_c]$  will contain  $c_{i^*}$ . At the final prediction step, the answer token  $\langle A \rangle$  will “measure” the superposition state  $[t_c]$  using  $c_1$  and  $c_2$  and predict the token with the larger signal in  $[t_c]$  as the output (see Figure C.1 in Section C.2.2 for a pictorial illustration). We formalize our result in the following theorem, and defer the proof to Section C.2.4.

**Theorem 8** (Chain of continuous thought solves graph reachability). *Fix  $T_{\max} > 0$ . Assume the length of the input sequence (including the continuous thoughts and the special answer token  $\langle A \rangle$ ) does not exceed  $T_{\max}$ . There exists a construction of a two-layer transformer parameters  $\theta$  and the readout matrix  $\mathbf{W}_O \in \mathbb{R}^{|\text{Voc}| \times d}$  (where  $d = O(|\text{Voc}|)$ ) that are independent of graphs, such that for any directed graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with at most  $n_{\max}$  nodes ( $n_{\max} < |\text{Voc}|$ ), a root node  $r$ , two candidate destination nodes  $c_1, c_2$ , for any  $C$  exceeds the diameter of the graph, it holds that*

$$\widetilde{\text{TF}}_{\theta, C, \mathbf{W}_O}(\hat{\mathbf{v}}_{[t_0]}) = c_{i^*}.$$

#### 4.4.4 Discussions

**Role of buffer spaces.** The buffer spaces are subspaces of the embedding to store useful information. For clarity, our construction separates the “content” and the two “buffer” spaces into different dimensions. In practice, they can be jointly projected into a more compact

and lower-dimensional space. For example, we can construct  $\mathbf{u} = \sum_{i=1}^k \mathbf{R}^{(i)} \mathbf{u}^{(i)} \in \mathbb{R}^d$  where the column space of each  $\mathbf{R}^{(i)} \in \mathbb{R}^{d \times d}$  forms a subspace. Different subspaces can be (almost) orthogonal, and for each subspace, the column vectors of  $\mathbf{R}^{(i)}$  can also be almost orthonormal.

**Weights of different nodes in the superposition.** In our construction, each superposition state maintains nodes with uniform weights. In practice, the weights of different nodes can vary due to factors such as training signals or the model’s internal heuristic on which nodes are more likely to reach the final answer [30]. In Section 4.5, we show that in practice, the training signal could bias the superposition states towards the *frontier nodes* that can be reached with exactly  $i$  steps and the *optimal nodes* that can lead to the destination node.

## 4.5 Experiments

In this section, we conduct extensive experiments to validate our theoretical results that COCONUT outperforms discrete CoT even with many fewer layers (Section 4.5.2), which is indeed due to superposition states encoded in continuous thoughts during reasoning (Section 4.5.3).

### 4.5.1 Training Setup

**Model.** We adopt a GPT-2 style decoder with two transformer layers,  $d_{\text{model}}=768$ ,  $n_{\text{heads}}=8$ . We train from scratch using AdamW ( $\beta_1=0.9$ ,  $\beta_2=0.95$ , weight-decay  $10^{-2}$ ) and a constant learning rate of  $1 \times 10^{-4}$ .

**Dataset.** We construct a subset of ProsQA [55], with questions whose solutions require 3–4 reasoning hops. Each node in the graph is injected as a dedicated token into the vocabulary. The split statistics are summarized in Table C.3.

**Method.** Following [55], we adopt a multi-stage training strategy with the supervision from chain-of-thoughts data. Stage  $i$  teaches the model to use  $i$  continuous thoughts before predicting the  $i$ -th node in the given chain of thought as the next token. If the index of the training stage is larger than the CoT solution length  $l$ , the model is trained to output the final prediction after  $l$  continuous thoughts and  $\langle \mathbf{A} \rangle$ . We train the model for 25 epochs at each stage, and the whole training lasts 300 epochs in total. At each stage, we mix the data from the previous stage randomly with 0.1 probability, which helps improve performance in our early experiments.

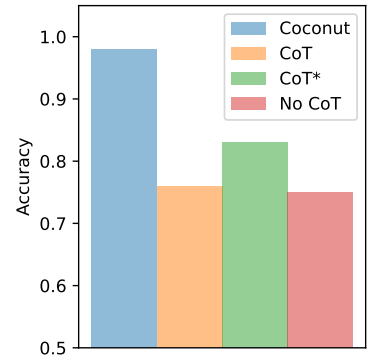


Figure 4.4: The overall accuracy of COCONUT, CoT, CoT\*(12 layers,  $n_{\text{heads}} = 12$ ) and No CoT.

### 4.5.2 Overall Results

Extending the findings of [55], we further show that a 2-layer transformer trained from scratch can effectively solve ProsQA when using COCONUT. As shown in Figure 4.4, COCONUT achieves near-perfect accuracy, while both CoT and No CoT baselines solve only about 75% of the tasks (random guessing = 50%). Even with a larger model of 12 layers and  $n_{\text{heads}} = 12$ , marked with \* in the figure, CoT improves to 83% accuracy but still cannot solve the tasks reliably.

### 4.5.3 Visualising Latent Reasoning

We inspect both the *attention pattern* and the *representation* of the continuous thought learned by the model to validate our theoretical construction.

**Layer 1 attention.** According to our theoretical construction, the most important function of LAYER 1 attention heads is to *copy* the source and target node tokens of an edge onto the corresponding edge token  $\langle e \rangle$ . Figure 4.5 shows a representative attention map, confirming that the model has instantiated this copying mechanism in practice.

**Layer 2 attention.** LAYER 2 is responsible for *node expansion*: each continuous thought attends to all outgoing edges from nodes that are currently reachable. To quantify this behavior, we compute, when generating  $i$ -th continuous thought, the aggregated attention score received by each edge token triplet  $(s, t, \langle e \rangle)$  across all heads. 4 kinds of edges exist: (1) *Reachable*: their source node is in the reachable set at step  $i$ ; (2) *Not Reachable*: source node *not* in the reachable set; (3) *Frontier*: a subset of reachable edges whose source node is on the current search frontier, i.e., exactly  $i$  steps away from the root node; (4) *Optimal*: a subset of frontier edges that lead to the optimal reasoning chain.

Table 4.1 reports group-wise means and standard deviations averaged on the test set. The model sharply concentrates its attention on *Reachable* edges, as predicted by our theoretical construction. Interestingly, there is an additional bias toward the *Frontier* subset. One possibility is that the training signal encourages the model to predict a frontier node at

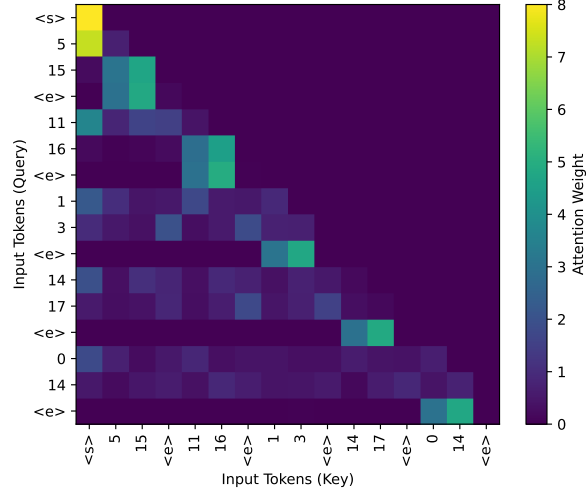


Figure 4.5: A representative example of Layer 1 attention map: the edge token  $\langle e \rangle$  ( $y$ -axis) places nearly all its attention mass on its source and target nodes ( $x$ -axis), in line with the theoretical construction.

each step, coupled with the decaying effects for previously explored nodes. We also find that the optimal edges receive more attention scores, likely due to the supervision of multi-stage training from the CoT solution.

**Representation of continuous thoughts.** To verify that the continuous thought serves as superposition states for the search, we compute the inner product between the continuous thought at step  $i$ ,  $[\mathbf{t}_i]$ , and each node embedding  $\mathbf{u}_v$ . Similar to edge classification above, we classify nodes into *Reachable*, *Not Reachable*, *Frontier*, and *Optimal*. Figure 4.6 (top) plots the distribution segregated by the reasoning step  $i$ . As predicted, nodes within  $i$  hops exhibit markedly higher similarity scores than distant nodes. Moreover, *Frontier* nodes are noticeably closer to  $[\mathbf{t}_i]$  than other reachable nodes, illustrating that the superposition emphasizes the candidate expansion fronts. Besides, the *Optimal* nodes are even closer to  $[\mathbf{t}_i]$ , also likely due to the training data always presenting the optimal path.

Collectively, these analyses confirm that the intended *superpositional search* exists in trained models: Layer 1 establishes the query context, Layer 2 expands the frontier, and the latent vectors encode a soft, parallel representation of reachable state sets, realizing the theoretical construction in Section 4.4. We also show in Appendix C.3.2 that this search pattern is consistent across multiple experiments with random seeds.

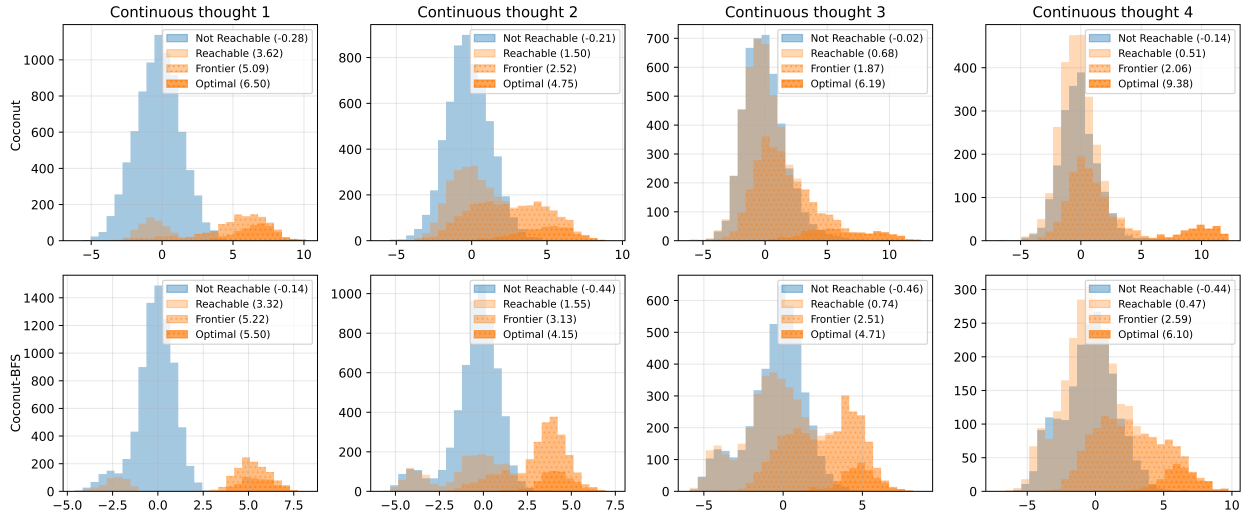


Figure 4.6: The histogram of inner product between the  $i$ -th continuous thoughts and the node embeddings. The mean value for each group is shown in the legend. Note that *Frontier* is a subset of *Reachable* nodes, and *Optimal* is a subset of *Frontier* nodes.

## 4.5.4 Exploration Priority

An interesting fact revealed by the visualizations in Section 4.5.3 is that the model allocates disproportionate attention to *optimal* edges and nodes, reminiscent of a prioritized search strategy. One hypothesis is that this behavior is an artifact of our multi-stage curriculum, which explicitly guides the model on the CoT solution at every step. To understand the

effect of this multi-stage guidance, we introduce an alternative supervision method COCONUT-BFS: at stage  $i$ , the supervision token is drawn uniformly at random from the frontier nodes exactly  $i$  hops from the root, rather than the  $i$ -th node in the CoT solution. All other hyperparameters are unchanged.

Our experiment results indicate that COCONUT-BFS also achieves near-perfect accuracy on ProsQA, matching the original COCONUT. Figure 4.6 compares the inner product distributions of the latent thoughts for both models. Remarkably, the two supervision methods converge to similar exploration strategies, even though there is no explicit guidance towards optimal nodes for COCONUT-BFS. Conversely, the original COCONUT, although trained exclusively on optimal nodes during intermediate stages, still assigns elevated weight to non-optimal frontier nodes compared to other non-frontier nodes, implicitly performing a breadth-first expansion before honing in on the solution.

Table 4.1: Layer 2 attention scores to different edge groups at step  $i$  (mean  $\pm$  standard deviation).

|               | Step 1          | Step 2          | Step 3          | Step 4          |
|---------------|-----------------|-----------------|-----------------|-----------------|
| Not Reachable | 0.04 $\pm$ 0.07 | 0.03 $\pm$ 0.09 | 0.08 $\pm$ 0.17 | 0.12 $\pm$ 0.20 |
| Reachable     | 2.12 $\pm$ 1.07 | 0.71 $\pm$ 0.92 | 0.38 $\pm$ 0.72 | 0.29 $\pm$ 0.66 |
| –Frontier     | 2.12 $\pm$ 1.07 | 1.00 $\pm$ 0.96 | 0.67 $\pm$ 0.87 | 0.61 $\pm$ 0.95 |
| –Optimal      | 2.54 $\pm$ 1.03 | 1.72 $\pm$ 1.13 | 1.67 $\pm$ 1.20 | 2.23 $\pm$ 1.35 |

## 4.6 Conclusions

In this work, we study how the chain-of-continuous-thought boosts LLM’s reasoning capability by focusing on the graph reachability problem. We provided a theoretical construction of a two-layer transformer that can efficiently solve graph reachability for an  $n$ -vertex  $D$ -diameter directed graph by  $D$  steps of continuous thoughts, while the best existing result on constant-depth transformers with discrete CoT requires  $O(n^2)$  steps. Our construction reveals that the superposition states that encode multiple search traces simultaneously are the key to the strong reasoning capability of COCONUT, and we conducted thorough experiments to validate that our theoretical construction matches the solutions obtained by training dynamics. Several interesting future directions include: (1) Deriving a lower bound on the number of discrete CoT steps in the graph reachability problem to show a strict separation of expressivity between CoT and COCONUT; (2) Theoretical understanding of the emergence of exploration behavior with only deterministic search trace demonstration via training dynamics; (3) Advantages of reasoning in a continuous space in more general settings.

# Chapter 5

## Emergence of Superposition in Continuous CoT

### 5.1 Introductions

Large Language Models (LLMs) show great reasoning capabilities in many complex tasks, especially when equipped with chain-of-thought (CoT) [145]. However, due to the large inference cost of long CoT for complex tasks, many recent works seek alternative test-time scaling methods to more efficiently improve LLMs’ reasoning ability [44, 142, 106, 122].

One promising method is to use chain-of-continuous-thought (COCONUT, or continuous CoT) [55], where the reasoning trace of an LLM is kept in a continuous latent space instead of projected back to the discrete token space. Continuous CoT exhibits both theoretical advantages [169] and empirical performance gains [55] in many tasks. To more efficiently and reliably scale up continuous CoT to solve more challenging tasks, it requires a deeper understanding of its internal mechanism.

Previous work [169] theoretically shows that one of the most important advantages of continuous CoT is that it can enable the model to reason by superposition: when the model encounters multiple plausible search traces and is uncertain about which one is correct, it can keep all plausible traces in parallel since the CoT is in continuous space instead of discrete tokens. In particular, [169] abstracted a family of reasoning tasks as a directed graph reachability problem, i.e., whether there exists a path from a given start node to a given destination node, and showed that a two-layer transformer with  $O(n)$  (where  $n$  is the number of vertices in the graph) continuous thought decoding steps can efficiently solve the task by providing a construction of the parameters. Therefore, a natural next question is:

*Do gradient-based methods naturally lead to such a construction, and can we theoretically prove it?*

This work answers the above question affirmatively by theoretically analyzing the training dynamics of a (simplified) two-layer transformer on the graph reachability problem in two

training stages: (i) a *thought generation* stage where the model autoregressively generates a chain of continuous thoughts and (ii) a *prediction* stage where the model predicts the final answer using the generated thought.

Importantly, our analysis of the thought generation training stage reveals why the superposition can *emerge* even if the training data only presents one demonstration for each training sample. Our theoretical analysis as well as experimental results show that when training with continuous thought (i.e., the COCONUT training method in Section 5.3 and Section 5.5), the index-matching logit, an important quantity that measures the strength of model’s local search capability, will remain bounded under mild conditions, which is in contrast to many previous analysis on transformer logit without continuous thought (e.g., [133, 96, 95]) where the logit will grow logarithmically and thus unbounded. A bounded index-matching logit can balance exploration and exploitation: if the logit is too small, the model cannot even perform local search, resulting in a nearly random guess in the next step; if the logit is too large, the model might over-confidently commit to one of the plausible search traces merely depending on local features (e.g., the indegree of a node in our case) even if it is uncertain about the solution, and thus early discard the correct path. A bounded index-matching logit encourages the model to exploit the local structure while exploring multiple plausible solutions by assigning comparable weights to them, which naturally results in superposition. This answers the question of [169] why superposition can emerge during training.

## 5.2 Problem Formulation

**Basic notations.** We use  $[N]$  to denote the set  $\{1, 2, \dots, N\}$  for any integer  $N > 0$  and use  $[i : j]$  to denote  $\{i, i + 1, \dots, j - 1, j\}$  for integers  $i \leq j$ . For any finite set  $\mathcal{X}$ , we use  $|\mathcal{X}|$  to denote its cardinality and use  $\text{Unif}(\mathcal{X})$  to denote the uniform distribution over  $\mathcal{X}$ . We use  $\mathbb{R}$  to denote the set of real numbers and denote  $x_+ = \max(x, 0)$  for  $x \in \mathbb{R}$ . For any vector  $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$ , the softmax function  $\text{SoftMax}(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^d$  is defined as  $\text{SoftMax}(\mathbf{x})_i = \exp(x_i) / (\sum_{j=1}^d \exp(x_j))$ . Let  $\mathbf{I}_d \in \mathbb{R}^{d \times d}$  denote the identity matrix. Let  $\text{Voc} = [M]$  denote a vocabulary of size  $M$  for a fixed integer  $M > 0$ . For each token  $v \in \text{Voc}$ , there is an associated embedding  $\mathbf{E}(v) \in \mathbb{R}^d$ . Let  $\mathbf{U} = [\mathbf{E}(1), \mathbf{E}(2), \dots, \mathbf{E}(M)] \in \mathbb{R}^{d \times M}$  be the token embedding matrix.

**Graph and permutation.** For a directed graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with the vertex set  $|\mathcal{V}| = n$  and the edge set  $\mathcal{E} = \{(\mathbf{s}_i \rightarrow \mathbf{t}_i)\}_{i=1}^m$ , we fix a root node  $\mathbf{r} \in \mathcal{V}$  and two candidate destination nodes  $\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{V}$  such that exactly one node, denoted  $\mathbf{c}_\star$ , is reachable from  $\mathbf{r}$  and denote the other as  $\mathbf{c}_\perp$  that is unreachable. For a radius  $c \in \mathbb{N}$ , define the  $c$ -ball as  $\mathcal{N}_c^{\mathcal{G}}(\mathbf{r}) = \{v \in \mathcal{V} : v \text{ is reachable from } \mathbf{r} \text{ within } c \text{ steps}\}$ . For a subset  $\mathcal{V}' \subseteq \mathcal{V}$ , we define the restricted in-degree as  $\text{deg}_{\mathcal{G}, \mathcal{V}'}^-(v) = |\{u \in \mathcal{V}' : (u \rightarrow v) \in \mathcal{E}\}|$ . We also fix a shortest path from  $\mathbf{r}$  to  $\mathbf{c}_\star$  as  $p = (p_0, \dots, p_C)$  with  $p_0 = \mathbf{r}$ ,  $p_C = \mathbf{c}_\star$ ,  $(p_{c-1} \rightarrow p_c) \in \mathcal{E}$  for any  $c \in [C]$ . For any permutation

$\pi$  over  $\mathcal{V}$ , we define  $\pi(\mathcal{G}) = (\mathcal{V}, \pi(\mathcal{E}))$ , where  $\pi(\mathcal{E}) = \{(\pi(\mathbf{s}) \rightarrow \pi(\mathbf{t})) \mid (\mathbf{s} \rightarrow \mathbf{t}) \in \mathcal{E}\}$ , and define  $\pi(p) = (\pi(p_0), \dots, \pi(p_C))$ . We also denote the set of all permutations over  $\mathcal{V}$  as  $S_{\mathcal{V}}$ .

**Chain of continuous thought.** Let  $\text{TF}_{\theta}(\cdot) : (\mathbb{R}^d)^* \rightarrow \mathbb{R}^d$  be a transformer which receives an input embedding sequence  $\hat{\mathbf{v}} = \hat{\mathbf{v}}_{[t]} \triangleq (\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \dots, \hat{\mathbf{v}}_t) \in \mathbb{R}^{d \times t}$  and outputs  $\text{TF}_{\theta}(\hat{\mathbf{v}}) \in \mathbb{R}^d$ . For convenience, we assume weight tying. A traditional decoder using a discrete CoT will sample the next token  $v_{t+1} \sim \text{SoftMax}(\mathbf{U}^{\top} \text{TF}_{\theta}(\hat{\mathbf{v}}))$ . Then the embedding of  $v_{t+1}$  will be appended to the end of the input, i.e.,  $\hat{\mathbf{v}}_{t+1} = \mathbf{E}(v_{t+1})$ . For continuous CoT, one directly appends the output of the transformer to the end of the input sequence without converting it to a token, i.e., setting  $\hat{\mathbf{v}}_{t+1} = \text{TF}_{\theta}(\hat{\mathbf{v}})$ . Assume the prompt  $\mathbf{x} = [x_1, \dots, x_{t_0}] \in \text{Voc}^{t_0}$  and its corresponding embedding sequence is  $\hat{\mathbf{v}}_{[t_0]} = [\hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_{t_0}] = [\mathbf{E}(x_1), \dots, \mathbf{E}(x_{t_0})]$ . For notation convenience, we use  $[\mathbf{t}_c] = \hat{\mathbf{v}}_{t_0+c}$  to denote the continuous thought generated at decoding step  $c$ , where  $[\mathbf{t}_c] = \text{TF}_{\theta}(\hat{\mathbf{v}}_{[t_0+c-1]})$ . In particular,  $[\mathbf{t}_0] = \hat{\mathbf{v}}_{t_0}$ . After  $C$  decoding steps, one can append a special token  $\langle \mathbf{A} \rangle$  at the end of the sequence to trigger the transformer to switch the mode and generate the final answer. Specifically, one can set  $\hat{\mathbf{v}}_T = \mathbf{E}(\langle \mathbf{A} \rangle)$  where  $T = t_0 + C + 1$  and generate the final answer  $\widetilde{\text{TF}}_{\theta, C, \mathbf{U}}(\hat{\mathbf{v}}_{[t_0]}) := \arg \max_{v \in \text{Voc}} \mathbf{U}^{\top} \text{TF}_{\theta}(\hat{\mathbf{v}}_{[T]})$ .

**Graph reachability and prompt format.** In this work, we mainly focus on the directed graph reachability problem as studied in [169], where we are given a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , two candidate destination nodes  $\mathbf{c}_1$  and  $\mathbf{c}_2$ , and a root node  $\mathbf{r}$ . The task is to identify which of the two nodes can be reached by  $\mathbf{r}$  (denoted as  $\mathbf{c}_{\star}$ ). The prompt structure is illustrated in Figure 5.1 following [169]. The prompt consists of (1) a BOS (beginning of sentence) token  $\langle \mathbf{s} \rangle$ ; (2) the graph description part, which contains  $m$  edges where each edge is represented by a source node  $\mathbf{s}_i$ , a target node  $\mathbf{t}_i$ , and a special edge token  $\langle \mathbf{e} \rangle$ ; (3) the task description part that contains a special question token  $\langle \mathbf{Q} \rangle$ , two candidate destination nodes  $\mathbf{c}_1$  and  $\mathbf{c}_2$ , a special reasoning token  $\langle \mathbf{R} \rangle$  and a root node  $\mathbf{r}$ . See Table D.1 for the full list of token notations. Note that  $t_0 = 3m + 6$  is the prompt length, and let  $\hat{\mathbf{v}}_{[t_0]} = (\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \dots, \hat{\mathbf{v}}_{t_0})$  be the input embedding sequence. Following [169], we use  $\text{Idx}(v)$  to denote the position of a token in the input sequence (e.g.,  $\text{Idx}(\langle \mathbf{s} \rangle) = 1$ ,  $\text{Idx}(\mathbf{s}_i) = 3i - 1$ ,  $\text{Idx}(\mathbf{c}_1) = 3m + 3$ ,  $\text{Idx}(\langle \mathbf{R} \rangle) = 3m + 5$ ), use  $\text{Idx}(\langle \mathbf{e} \rangle, i) = 3i + 1$  to denote the position of the  $i$ -th  $\langle \mathbf{e} \rangle$  token, and use  $\text{Idx}([\mathbf{t}_i]) = t_0 + i$  to denote the position of the continuous thought at step  $i$ . See Table D.2 for the complete list of position indices.

[169] provided a construction of transformer parameters  $\theta$  such that  $\widetilde{\text{TF}}_{\theta, C, \mathbf{U}}(\hat{\mathbf{v}}_{[t_0]}) = \mathbf{c}_{\star}$  (i.e., the transformer can predict the reachable candidate node using continuous CoT) for any graph and root-candidate node tuples, where  $\hat{\mathbf{v}}_{[t_0]}$  corresponds to the prompt of the graph and task descriptions. However, they did not theoretically study whether the constructed solution can be naturally learned via gradient-based methods. In the following sections, we theoretically show that the solution can be learned via gradient flow in both the thought generation stage (Section 5.3) and the prediction stage (Section 5.4). We also provide empirical results showing that the training dynamics in our theoretical analysis align well with the experiments (Section 5.5).

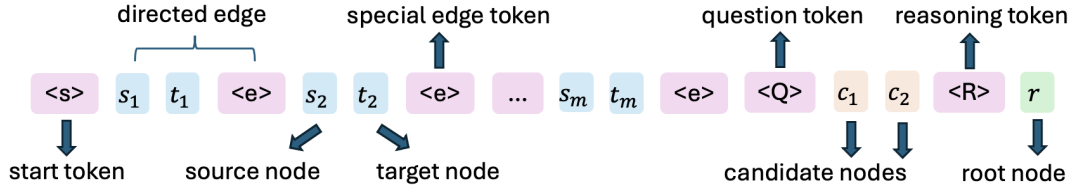


Figure 5.1: Prompt format of the graph reachability problem (Figure 1 of [169]).

### 5.2.1 Index-matching Logits and Local Search Capability

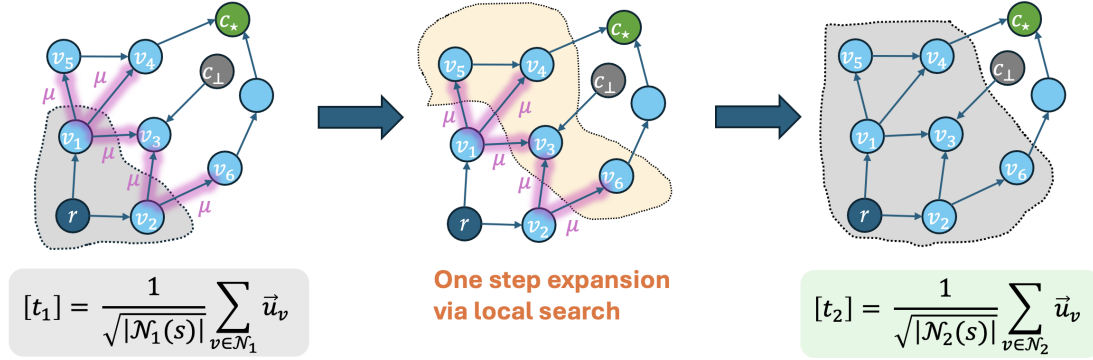
Before we delve into the technical details in the following sections, we first provide an intuitive explanation of the dynamics of the main mechanism.

**Global planning vs. local search.** In the context of graph reachability, the global planning refers to a model’s capability to analyze the structure of the whole graph and then determine a path from the root node to the destination node. In contrast, local search focuses only on which nodes are reachable in one step from the current node, which is much easier to learn than global planning. When using discrete CoT, the model can choose only one path at a time. Therefore, the model needs global planning to select the correct path or to backtrack from the wrong one. When using continuous CoT, the model can keep multiple plausible paths simultaneously. Therefore, the model can rely solely on local search to perform parallel BFS, solving the task with only simple skills.

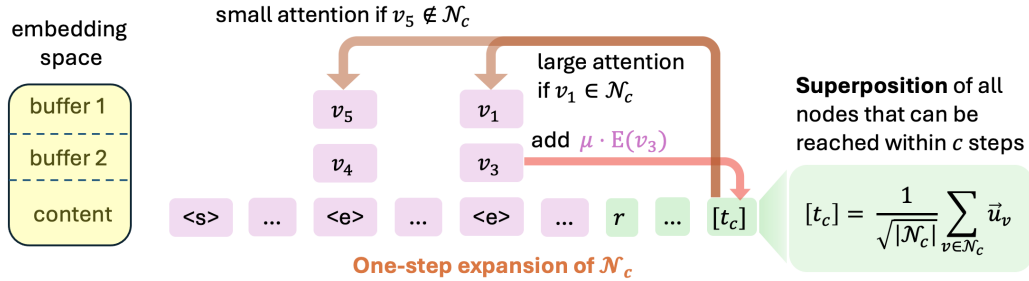
**Index-matching logits.** We use the index-matching logit  $\mu$  to quantify the strength of the model’s local search capability, which is illustrated in Figure 5.2 and will be formally defined in (5.3) in Section 5.3. In Theorem 9, we will prove that under mild conditions, the index-matching logit  $\mu$  will first increase and then remain bounded. Note that a positive, bounded logit  $\mu$  effectively balances exploration and exploitation in node expansion: if  $\mu$  is too small, each edge will receive similar attention in Figure 5.2b, and thus the model even lacks the local search capability to exploit the local graph structure; if  $\mu$  is too large, the model will put too much weights on nodes with large in-degree (e.g., in Figure 5.2a,  $v_3$  weights  $2\mu$  and other frontier nodes such as  $v_4$  weights  $\mu$ , where the difference in weights will be significant under large  $\mu$  and commonly used softmax attention) and thus lacks exploration of different plausible paths.

## 5.3 Analysis of the Thought Generation Stage

In this section, we analyze the training dynamics of the thought generation stage. We consider any graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , a root node  $r \in \mathcal{V}$ , two candidate destination nodes  $c_1, c_2 \in \mathcal{V}$ ,



(a) **Left:** The continuous thought at step 1  $[t_1]$  encodes embeddings of nodes that are reachable from the root node  $r$  within one step. **Middle:** One-step expansion via local search where the strength is quantified by index-matching logit  $\mu$ . **Right:** After one-step expansion, the continuous thought at step 2  $[t_2]$  encodes nodes reachable within two steps.



(b) Illustration of how one-step expansion is implemented (adapted from Figure 3 of [169]). In the first layer of the transformer, each special edge token  $\langle e \rangle$  copies its corresponding source and target nodes to its buffer spaces. In the second layer, as illustrated in the figure, the current thought  $[t_c]$  pays large attention to an edge if its source node has been explored, and adds its target node to the superposition, where the strength of the added node is controlled by the index-matching logit  $\mu$ . The two edges  $v_5 \rightarrow v_4$  and  $v_1 \rightarrow v_3$  corresponds to edges in Figure 5.2a.

Figure 5.2: Pictorial illustration of the superposition mechanism and the index-matching logit  $\mu$ .

where  $\{c_1, c_2\} \in \{c_*, c_\perp\}$  with  $c_*$  reachable from  $\mathbf{r}$  and  $c_\perp$  unreachable. We are also given a (discrete) CoT demonstration, which is a shortest path  $p = (p_0, \dots, p_C)$  from  $\mathbf{r}$  to  $c_*$  where  $p_0 = \mathbf{r}$ ,  $p_C = c_*$ .

We use curriculum learning following [55, 169], where at stage  $(c + 1)$  for any  $0 \leq c < C$ , upon receiving the prompt embeddings  $\hat{\mathbf{v}}_{[c_0]}$ , the model will first generate  $c$  continuous thoughts  $[\mathbf{t}_1], \dots, [\mathbf{t}_c]$  autoregressively without supervision (i.e., no loss calculated on the first  $c$  continuous thoughts at stage  $c + 1$ ), and then be trained to generate the next continuous thought  $[\mathbf{t}_{c+1}] = \text{TF}_\theta(\hat{\mathbf{v}}_{[t_0+c]})$ . Since the learning procedure at each stage is similar, we focus below on a fixed  $c$ .

[169] constructs a solution for a two-layer transformer, where the first layer mainly performs copy (e.g., the  $i$ -th special edge token  $\langle \mathbf{e} \rangle$  will copy the information of its corresponding source node  $\mathbf{s}_i$  and target node  $\mathbf{t}_i$ ). Since the copy mechanism has been widely studied [95], as well as its formation via training dynamics [96], we mainly focus on the dynamics after the copy mechanism has been established. Thus, we analyze the dynamics of the second layer of the transformer.

In particular, let the hidden states of each special edge token  $\langle \mathbf{e} \rangle$  and the current thought  $[\mathbf{t}_c]$  after the first transformer layer be

$$\hat{\mathbf{v}}_{\text{Idx}(\langle \mathbf{e} \rangle, i)} = \mathbf{E}_s(\mathbf{s}_i) + \mathbf{E}_t(\mathbf{t}_i) \in \mathbb{R}^d, \quad \hat{\mathbf{v}}_{\text{Idx}([\mathbf{t}_c])} = \sum_{v \in \mathcal{N}_c^g(\mathbf{r})} \frac{1}{\sqrt{|\mathcal{N}_c^g(\mathbf{r})|}} \mathbf{E}(v) \in \mathbb{R}^d, \quad (5.1)$$

where  $\mathbf{E}_s(v) \in \mathbb{R}^d$  and  $\mathbf{E}_t(v) \in \mathbb{R}^d$  map token  $v \in \text{Voc}$  to different subspaces of  $\mathbb{R}^d$ . For example, as in the construction of [169], we can set  $d = 3M$ , and  $\mathbf{E}_s(\cdot)$ ,  $\mathbf{E}_t(\cdot)$  and  $\mathbf{E}(\cdot)$  each corresponds to  $M$  different non-zero entries. This is also similar to previous work [25, 95] where  $\mathbf{E}_s(\cdot)$  and  $\mathbf{E}_t(\cdot)$  can be viewed as previous token heads. We make the following assumptions on the embedding  $\mathbf{E}_s(\cdot)$ ,  $\mathbf{E}_t(\cdot)$  and  $\mathbf{E}(\cdot)$ :

**Assumption 3** (Orthonormal embeddings). *Assume  $\mathbf{E}_t(\cdot) \equiv \mathbf{E}(\cdot)$ . For any  $u, v \in \text{Voc}$ ,  $\mathbf{E}_s(u)^\top \mathbf{E}_s(v) = \mathbf{E}_t(u)^\top \mathbf{E}_t(v) = \mathbb{1}(\{u = v\})$  and  $\mathbf{E}_s(u)^\top \mathbf{E}_t(v) = 0$ .*

(5.1) means after the first layer, each special edge token  $\langle \mathbf{e} \rangle$  will copy the embeddings of its corresponding source and target nodes  $\mathbf{s}_i$  and  $\mathbf{t}_i$  to the same position in different subspaces. Also, we assume by induction that after training stages  $1, 2, \dots, c$ , the current thought generated by the well-trained model  $\hat{\mathbf{v}}_{\text{Idx}([\mathbf{t}_c])}$  is a normalized superposition of token embeddings of all nodes reachable from  $\mathbf{r}$  within  $c$  steps. Below, we study the training dynamics of the current stage (i.e., stage  $c + 1$ ).

**The forward path and reparameterization.** We consider the setting where the second layer is attention-only. The forward pass can be formulated as

$$\begin{aligned} \phi(\hat{\mathbf{v}}; \{\hat{\mathbf{v}}_i\}_i) &= \sum_i \mathbf{W}_{V\sigma}(\hat{\mathbf{v}}^\top \mathbf{W} \hat{\mathbf{v}}_i) \hat{\mathbf{v}}_i, \\ \boldsymbol{\xi} &= \mathbf{U}^\top(\hat{\mathbf{v}}_{\text{Idx}([\mathbf{t}_c])} + \phi(\hat{\mathbf{v}}_{\text{Idx}([\mathbf{t}_c])}; \{\hat{\mathbf{v}}_{\text{Idx}(\langle \mathbf{e} \rangle, i)}\}_{i=1}^m)) \in \mathbb{R}^M, \end{aligned} \quad (5.2)$$

where  $\mathbf{W}_V, \mathbf{W} \in \mathbb{R}^{d \times d}$  are attention parameters and  $\sigma(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$  is an activation function that determines the range of attention scores, and  $\boldsymbol{\xi} = (\xi_v)_{v \in \text{Voc}} \in \mathbb{R}^M$  is the output logit vector for each token in the vocabulary. Similar to the analysis in [95], we adopt the linear attention  $\sigma(\hat{\mathbf{v}}^\top \mathbf{W} \hat{\mathbf{v}}_i) = \hat{\mathbf{v}}^\top \mathbf{W} \hat{\mathbf{v}}_i$ , fix  $\mathbf{W}_V = \mathbf{I}$  and use the *index-matching* reparameterization

$$\mathbf{W} = \sum_{v \in \mathcal{V}} \mu_v \mathbf{E}(v) \mathbf{E}_s(v)^\top, \quad \mu_v(t) = 0 \text{ for } t = 0. \quad (5.3)$$

**Remark 3.** Note that a more general form of the attention weight matrix can be

$$\mathbf{W} = \mu_{\langle R \rangle} \mathbf{E}(\langle A \rangle) \mathbf{E}(\langle R \rangle)^\top + \sum_{v, v' \in \mathcal{V}} \mu_{v, v'} \mathbf{E}(v) \mathbf{E}_s(v')^\top. \quad (5.4)$$

The first term only takes effect in the prediction stage (Section 5.4), so we can set  $\mu_{\langle R \rangle} = 0$  for now. The second term involves  $n \times n$  cross terms. The symmetry of the vertices, which can be enforced by permuting vertex labels during training, makes the  $n \times n$  parameters  $\{\mu_{v, v'}\}_{v, v'}$  effectively two parameters  $\{\mu_1, \mu_2\}$  where  $\mu_{v, v} \equiv \mu_1$  and  $\mu_{v, v'} \equiv \mu_2$  for  $v \neq v'$ . Moreover, if we focus on the relative value between  $\mu_1$  and  $\mu_2$ , we can further simplify the attention weight matrix by assuming  $\mu_2 = 0$ .

For notation simplicity, we use  $\hat{\mathbf{v}}_i$  to denote  $\hat{\mathbf{v}}_{\text{Idx}(\langle \mathbf{e} \rangle, i)}$ , use  $\hat{\mathbf{v}}_{[\mathbf{t}_c]}$  to denote  $\hat{\mathbf{v}}_{\text{Idx}([\mathbf{t}_c])}$  and use  $\mathcal{N}_c, \mathcal{N}_{c+1}$  to denote  $\mathcal{N}_c^{\mathcal{G}}(\mathbf{r}), \mathcal{N}_{c+1}^{\mathcal{G}}(\mathbf{r})$ , respectively when the graph  $\mathcal{G}$  and root node  $\mathbf{r}$  is clear from the context. We also denote  $d_u := \text{deg}_{\mathcal{G}, \mathcal{N}_c}^-(u)$  which is the indegree of  $u$  with source nodes restricted in  $\mathcal{N}_c$ . Finally, we denote  $K = |\mathcal{N}_c|$  and  $\lambda = \frac{1}{\sqrt{K}}$ .

**Loss functions.** An ideal model should be able to directly output the shortest path from the start node  $\mathbf{r}$  to the desired candidate destination node  $\mathbf{c}_\star$ , i.e., the prediction of the  $(c+1)$ -th continuous thought  $[\mathbf{t}_{c+1}]$  exactly corresponds to the  $(c+1)$ -th step of the shortest path  $p_{c+1}$ . However, experiments in [169] show that even for a 12-layer transformer, it is hard to predict the shortest path even if the length of the shortest path is only 3 or 4. Therefore, we take a step back and pursue a more practical goal – we expect the model to at least be able to generate an arbitrary path starting from the start node  $\mathbf{r}$ , which only requires local search ability that is much easier than the global planning ability. In the context of continuous thought, we expect the model to include information of all vertices that are reachable from  $\mathbf{r}$  within  $(c+1)$  steps in the generated thought  $[\mathbf{t}_{c+1}]$ . We consider the following two loss functions:

$$\textbf{Coconut-BFS: } \ell_{\mathcal{G}, \mathbf{r}}^{\text{BFS}} := -\log \frac{\sum_{v \in \mathcal{N}_{c+1}} \exp(\xi_v)}{\sum_{v \in \mathcal{V}} \exp(\xi_v)}, \quad (5.5)$$

$$\textbf{Coconut: } \ell_{\mathcal{G}, \mathbf{r}, p}^{\text{coco}} := -\log \frac{\exp(\xi_{p_{c+1}})}{\sum_{v \in \mathcal{V}} \exp(\xi_v)}, \quad (5.6)$$

with permutation-averaged dataset losses

$$\mathcal{L}^{\text{BFS}} = \mathbb{E}_{\pi \sim \text{Unif}(S_V)} [\ell_{\pi(\mathcal{G}), \pi(\mathbf{r})}^{\text{BFS}}] \quad \text{and} \quad \mathcal{L}^{\text{coco}} = \mathbb{E}_{\pi \sim \text{Unif}(S_V)} [\ell_{\pi(\mathcal{G}), \pi(\mathbf{r}), \pi(p)}^{\text{coco}}].$$

Note that, intuitively, the permutation-averaged loss will lead to similar behavior across different parameters. The first loss  $\mathcal{L}^{\text{BFS}}$  explicitly encourages the model to predict any nodes in  $\mathcal{N}_{C+1}$ . However, in practice, it is costly and even impossible to search over the entire solution space exhaustively; instead, we usually present only one demonstration for each task instance during training (in our setting, only one path  $p$  per instance  $(\mathcal{G}, \mathbf{r}, \mathbf{c}_1, \mathbf{c}_2)$ ), which corresponds to the second loss  $\mathcal{L}^{\text{coco}}$  and aligns with the practical setting where chain of thought data can be used for supervision.

[169] observed in experiments that superposition emerges even without explicit guidance during training, i.e., using the loss  $\mathcal{L}^{\text{coco}}$ . In this work, we investigate the emergence of superposition by analyzing its training dynamics. The following lemma gives the gradient of the index-matching strength parameter  $\mu_v(t)$  using gradient flow under the loss function  $\mathcal{L}^{\text{coco}}$ .

**Lemma 5** (Gradient of  $\mu_v$  under  $\mathcal{L}^{\text{coco}}$ ; informal version of Theorem 19 in Section D.2). *Under permutation-averaged training from symmetric initialization and gradient flow  $\dot{\mu}_v = -\alpha \nabla_{\mu_v} \mathcal{L}^{\text{coco}}$ , we have  $\mu_v(t) \equiv \mu(t)$  for all  $v$  and times  $t$ , and the gradient of  $\mu_v$  is*

$$\dot{\mu}(t) = \frac{\alpha}{n\sqrt{K}} \left( d_{p_{c+1}} - F(\mu(t)) \right), \quad F(\mu) = \frac{\sum_{u \in \mathcal{N}_{c+1}} d_u e^{\lambda(\mathbb{1}(\{u \in \mathcal{N}_c\} + \mu d_u))}}{\sum_{u \in \mathcal{N}_{c+1}} e^{\lambda(\mathbb{1}(\{u \in \mathcal{N}_c\} + \mu d_u))} + (n - |\mathcal{N}_{c+1}|)}.$$

Moreover  $F$  is smooth, strictly increasing, with  $F(-\infty) = 0$ ,  $F(+\infty) = \max_{v \in \mathcal{V}} d_v$  and  $0 < F(\mu) < \max_{v \in \mathcal{V}} d_v$  for all finite  $\mu$ .

The proof is deferred to Section D.2. Note that as long as  $d_{p_{c+1}} \neq \max_{v \in \mathcal{V}} d_v$ ,  $\mu(t)$  will converge to  $\mu_\star < \infty$ . In contrast, under COCONUT-BFS with loss  $\mathcal{L}^{\text{BFS}}$ ,  $\mu(t)$  will diverge to infinity. We formalize the comparison into the following theorem and defer the proof to Section D.2.

**Theorem 9** (Bounded vs. divergent attention logits under COCONUT vs. COCONUT-BFS; informal version of Theorem 19 & Lemma 38 in Section D.2). *Let  $d_\star := d_{p_{c+1}}$  and  $d_{\max} := \max_v d_v$ .*

- (i) *Under COCONUT-BFS (5.5),  $\mu(t)$  grows at least logarithmically in  $t$ , leading to unbounded attention logits.*
- (ii) *Under COCONUT (5.6), if  $d_\star < d_{\max}$  then  $\mu(t) \rightarrow \mu^\star < \infty$ , so all attention logits remain uniformly bounded. If  $d_\star = d_{\max}$ , then  $\mu(t) \rightarrow \infty$  at least in a logarithmic rate.*

**Emergence of Superposition via Bounded Attention Logits.** By Theorem 9, as long as  $F(0) < d_{p_{c+1}} < d_{\max}$ , we have  $\mu(t) \rightarrow \mu^\star > 0$ . Compared to many previous work [133, 96, 95] that analyze the dynamics of attention logits in “discrete” settings where the attention logits diverge to infinity, the COCONUT training method in continuous setting usually result in bounded attention logits. The bounded attention logits lead to a more smooth probability

distribution over next tokens, which is beneficial especially under uncertainty: when the model is uncertain about the next step, a more smooth probability distribution under continuous CoT mechanism results in a superposition of different plausible next steps, which implements an effective exploration; on the contrary, an unbounded logit will result in a one-hot-like distribution and thus the model will over-confidently commit to a plausible branch and is likely to discard the ground-truth branch even when the evidence is weak.

Finally, we show that with a positive value of  $\mu$ , the continuous thought  $[\mathbf{t}_{c+1}]$  implements a one-step expansion from  $\mathcal{N}_c$  to  $\mathcal{N}_{c+1}$  for any graph  $\mathcal{G}$  and root node  $\mathbf{r}$ .

**Theorem 10** (One-step frontier expansion; informal version of Theorem 20 in Section D.2). *For any graph  $\mathcal{G}$  and root node  $\mathbf{r}$ , if the current thought is any positive superposition on  $\mathcal{N}_c^{\mathcal{G}}(\mathbf{r})$ , i.e.,  $[\mathbf{t}_c] = \sum_{u \in \mathcal{N}_c} \lambda_u \mathbf{E}(u)$  with  $\lambda_u > 0$ , then the next thought  $[\mathbf{t}_{c+1}]$  satisfies that its token-projected output  $\mathbf{U}^\top [\mathbf{t}_{c+1}]$  is supported on the one-step expansion  $\mathcal{N}_{c+1}$  and has strictly positive mass on every node in  $\mathcal{N}_{c+1}$  if  $\mu > 0$ . In particular,*

$$\mathbf{U}^\top [\mathbf{t}_{c+1}] = \sum_{v \in \mathcal{N}_{c+1}} \beta_v \mathbf{e}_v$$

with

$$\beta_v = \underbrace{\lambda_v \mathbb{1}(\{v \in \mathcal{N}_c\})}_{\text{carryover}} + \underbrace{\mu \sum_{u \in \mathcal{N}_c} \lambda_u \mathbb{1}(\{(u \rightarrow v) \in \mathcal{E}\})}_{\text{one-hop expansion}} \geq 0.$$

The proof is deferred to Section D.2. Note that at initialization where  $\mu = 0$ , we have  $\beta_v = 0$  for  $v \in \mathcal{N}_{c+1} \setminus \mathcal{N}_c$ . This means every node outside  $\mathcal{N}_c$  has the same attention logits and thus the same next token probability. However, such an exploration is not an effective exploration since it blindly puts the same weight on almost every node in the graph without exploiting the graph structure. Therefore, an appropriate  $\mu^* > 0$  effectively balances the exploration and exploitation: (1) it has a positive value so the model can exploit the graph structure and can distinguish nodes within the one-step expansion set; (2) it has a bounded value so it will not overconfidently commit to a plausible branch while discarding other branches merely relying on local structure (such as the indegree of the node) without global planning.

## 5.4 Analysis of the Prediction Stage

In this section, we study how the transformer learns to make the correct prediction  $\mathbf{c}_\star$  among  $\{\mathbf{c}_1, \mathbf{c}_2\}$  by utilizing the generated continuous thought. Note that according to Section 5.3, the model is able to generate  $[\mathbf{t}_c] = \sum_{v \in \mathcal{N}_c} \lambda_v \mathbf{E}(v)$  with  $\lambda_v \in (0, 1]$ , a superposition of all reachable nodes within  $C$  steps, via a balanced exploration and exploitation. We denote  $\boldsymbol{\lambda} = \{\lambda_v\}_{v \in \mathcal{V}}$ . At the final stage, one appends a special answer token  $\langle \mathbf{A} \rangle$  at the end of the continuous CoT, i.e.,  $\hat{\mathbf{v}}_T = \hat{\mathbf{v}}_{\langle \mathbf{A} \rangle}$ , and make the final prediction  $\widetilde{\text{TF}}_{\theta, C, \mathbf{U}}(\hat{\mathbf{v}}_{[t_0]}) := \arg \max_{v \in \mathcal{V}_{\text{oc}}} \mathbf{U}^\top \text{TF}_\theta(\hat{\mathbf{v}}_{[T]}).$

**The forward path and reparameterization.** Similar to (5.2), we formulate the forward pass in the prediction stage as

$$\begin{aligned}\phi(\hat{\mathbf{v}}; \{\hat{\mathbf{v}}_i\}_i) &= \sum_i \mathbf{W}_V \sigma(\hat{\mathbf{v}}^\top \mathbf{W} \hat{\mathbf{v}}_i) \hat{\mathbf{v}}_i, \\ \boldsymbol{\xi} &= \mathbf{U}^\top (\mu_{\langle \mathbf{A} \rangle} \hat{\mathbf{v}}_{\text{ldx}(\langle \mathbf{A} \rangle)} + \phi(\hat{\mathbf{v}}_{\text{ldx}(\langle \mathbf{A} \rangle)}; \{\hat{\mathbf{v}}_{\langle \mathbf{R} \rangle}\})) \in \mathbb{R}^M,\end{aligned}\tag{5.7}$$

where

$$\hat{\mathbf{v}}_{\text{ldx}(\langle \mathbf{R} \rangle)} = \mathbf{E}(\langle \mathbf{R} \rangle) + \mathbf{E}(\mathbf{c}_1) + \mathbf{E}(\mathbf{c}_2), \quad \hat{\mathbf{v}}_{\text{ldx}(\langle \mathbf{A} \rangle)} = \hat{\mathbf{v}}_{[\mathbf{t}_c]} + \mathbf{E}(\langle \mathbf{A} \rangle).$$

Note that after the first transformer layer, the hidden state of  $\langle \mathbf{R} \rangle$  contains information of two candidate nodes  $\mathbf{c}_1$  and  $\mathbf{c}_2$  and the hidden state of  $\langle \mathbf{A} \rangle$  contains the representation of the last thought  $[\mathbf{t}_c]$  both due to the copy mechanism in the first layer. Again, we adopt the linear attention  $\sigma(\hat{\mathbf{v}}^\top \mathbf{W} \hat{\mathbf{v}}_i) = \hat{\mathbf{v}}^\top \mathbf{W} \hat{\mathbf{v}}_i$ , fix  $\mathbf{W}_V = \mathbf{I}$  and use the reparameterization

$$\mathbf{W} = \mu_{\langle \mathbf{R} \rangle} \mathbf{E}(\langle \mathbf{A} \rangle) \mathbf{E}(\langle \mathbf{R} \rangle)^\top.\tag{5.8}$$

**Remark 4.** The scalar  $\mu_{\langle \mathbf{R} \rangle}$  denotes the attention logit strength from  $\langle \mathbf{A} \rangle$  to  $\langle \mathbf{R} \rangle$ . The scalar  $\mu_{\langle \mathbf{A} \rangle}$  represents the signal strength of the residual stream from the first layer. Also, note that the reparameterization of  $\mathbf{W}$  in the prediction stage has a different form from (5.3) in the thought generation stage. One can either view both (5.3) and (5.8) as special cases of a more general version (5.4) in orthogonal subspaces, or view them as two different attention heads (a thought generation head and a prediction head).

**The loss function.** In the prediction stage, the goal of the model is to predict the reachable candidate node  $\mathbf{c}_\star$ , and thus the loss function can be written as

$$\ell_{\mathcal{G}, \mathbf{r}, \mathbf{c}_1, \mathbf{c}_2, \boldsymbol{\lambda}}^{\text{pred}} := -\log \frac{\exp(\xi_{\mathbf{c}_\star})}{\sum_{v \in \mathcal{V}} \exp(\xi_v)}, \quad \mathcal{L}^{\text{pred}} = \mathbb{E}_{(\mathcal{G}, \mathbf{r}, \mathbf{c}_1, \mathbf{c}_2, \boldsymbol{\lambda}) \sim \mathcal{D}} [\ell_{\mathcal{G}, \mathbf{r}, \mathbf{c}_1, \mathbf{c}_2, \boldsymbol{\lambda}}^{\text{pred}}],\tag{5.9}$$

where the  $\mathcal{D} = \{(\mathcal{G}^{(i)}, \mathbf{r}^{(i)}, \mathbf{c}_1^{(i)}, \mathbf{c}_2^{(i)}, \boldsymbol{\lambda}^{(i)})\}_i$  denote the training set. The following lemma provides closed-form logits for each vertex where the proof is deferred to Section D.3.

**Lemma 6** (Closed-form logits; informal version of Lemma 41 in Section D.3). *The logit of each vertex  $v \in \mathcal{V}$  has the form*

$$\xi_v = \underbrace{\mu_{\langle \mathbf{A} \rangle} \lambda_v \mathbb{1}(\{v\} \in \mathcal{N}_C)}_{\text{residual carryover}} + \underbrace{\mu_{\langle \mathbf{R} \rangle} \mathbb{1}(\{v\} \in \{\mathbf{c}_1, \mathbf{c}_2\})}_{\text{candidate lift}}.$$

According to Lemma 6, only the candidate node  $\mathbf{c}_\star$  has both positive residual carryover and candidate lift, and an appropriate relative growth rate of  $\mu_{\langle \mathbf{R} \rangle}$  and  $\mu_{\langle \mathbf{A} \rangle}$  ensures that  $\mathbf{c}_\star$  has the largest logit. We formalize the result in the following theorem with proof in Section D.3.

**Theorem 11** (Prediction of the reachable candidate node; informal version of Theorem 21 in Section D.3). *Denote  $\mu_A = \mu_{\langle A \rangle}$  and  $\mu_R = \mu_{\langle R \rangle}$ . Let  $(\mu_{\langle A \rangle}(t), \mu_{\langle R \rangle}(t))$  follow gradient flow on loss defined in (5.9). Suppose*

$$\lambda_\star := \min_i \lambda_{c_\star}^{(i)} \in (0, 1], \quad \Delta_{\text{train}} := \max_i \max_{v \in \mathcal{N}_C^{(i)} \setminus \{c_\star^{(i)}\}} (\lambda_v^{(i)} - \lambda_{c_\star}^{(i)})_+ \in [0, 1].$$

Then we have

$$\frac{(\mu_A(t), \mu_R(t))}{\|(\mu_A(t), \mu_R(t))\|} \rightarrow u^\star, \quad \|(\mu_A(t), \mu_R(t))\| \rightarrow \infty,$$

with  $u_R^\star / u_A^\star = \lambda_\star + \Delta_{\text{train}}$ , and  $u_R^\star, u_A^\star > 0$ . Consequently, for any unseen instance  $(\mathcal{G}, \mathbf{r}, \mathbf{c}_1, \mathbf{c}_2, \boldsymbol{\lambda})$  satisfying  $\lambda_v \in (0, 1]$  on  $\mathcal{N}_C$  and 0 otherwise, and  $\max_v \lambda_v - \lambda_{c_\star} \leq \Delta_{\text{train}}$ , it holds that:

$$p_{c_\star}(t) := \frac{\exp(\xi_{c_\star}(\mu_A(t), \mu_R(t)))}{\sum_v \exp(\xi_v(\mu_A(t), \mu_R(t)))} \xrightarrow{t \rightarrow \infty} 1.$$

## 5.5 Experiments

In this section, we present experimental results validating the theoretical analysis. We first describe the setup and overall results, then analyze training dynamics in the thought generation and answer prediction stages.

**Model.** We adopt a GPT-2 style decoder with two transformer layers ( $d_{\text{model}}=768$ ,  $n_{\text{heads}}=8$ ). The model is trained from scratch with AdamW ( $\beta_1=0.9$ ,  $\beta_2=0.95$ , weight decay  $10^{-2}$ ), a constant learning rate of  $1 \times 10^{-4}$ , and a global batch size of 256.

**Dataset.** We follow the dataset from [169], which is a subset of ProsQA [55]. Different from [169], we randomly permute the vertex indices in both training and testing to avoid prediction bias and validate the symmetry assumption in (5.4). Dataset statistics are summarized in Table D.3.

**Training.** Following [55, 169], we use a multi-stage training strategy with supervision from chain-of-thought demonstrations. At stage  $c$ , the model learns to use  $c$  continuous thoughts before predicting the  $c$ -th node on the reasoning path (*thought-generation* stage). If  $c > l$  (the CoT length), the model predicts the final answer after  $l$  continuous thoughts and the  $\langle A \rangle$  token (*prediction* stage). We train for 150 epochs at Stage 1 and 25 epochs for each subsequent stage, totaling 350 epochs. At each stage, data from earlier stages is mixed in with probability 0.1, which prevents the model from forgetting abilities learned from previous stages. The final accuracy of this model on the test set is 96.2%.

### 5.5.1 Thought Generation

To examine the training dynamics of  $\mu_v$  under  $\mathcal{L}^{\text{coco}}$ , we track the second-layer attention logits. When generating the  $c$ -th continuous thought,  $\mu_v$  corresponds to the logit on an edge token  $\langle \mathbf{e} \rangle$  whose source lies in  $\mathcal{N}_c$ . In practice,  $\mathcal{L}^{\text{coco}}$  encourages the model to predict the current search frontier rather than revisiting explored nodes, so most attention concentrates on *frontier edges*, i.e., edges with sources in  $\mathcal{N}_c \setminus \mathcal{N}_{c-1}$ . For theoretical simplicity, we assume  $\mu_2 = 0$  in (5.4). In practice, however, the model does assign non-zero attention logits to other edges. Therefore, we report the logit difference between frontier and non-frontier edges on the test set, which more faithfully reflects the effective value of  $\mu_v$ .

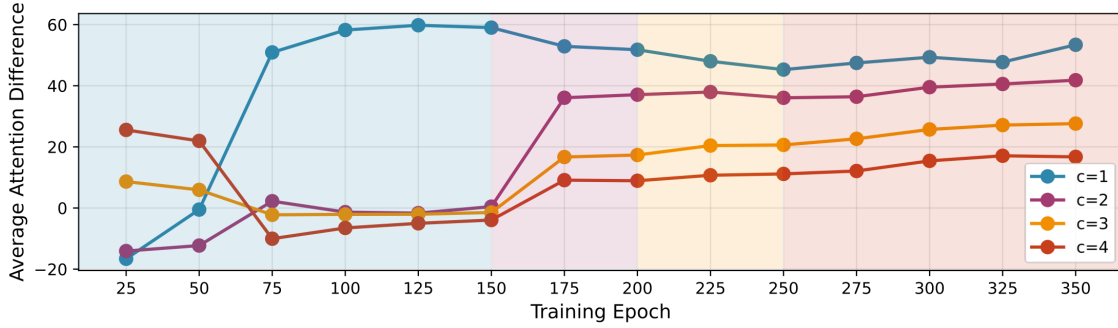


Figure 5.3: The attention logits difference between frontier edges and others under  $\mathcal{L}^{\text{coco}}$  as a proxy for  $\mu_v$ . The background colors indicate different training stages.

Figure 5.3 shows the results. In Stage 1 (blue background), the model gradually learns to attend to frontier edges when predicting the first continuous thought ( $c = 1$ ). The logit difference increases steadily and saturates around 60 after  $\sim 125$  epochs. This matches the theoretical prediction in [Theorem 9](#): under  $\mathcal{L}^{\text{coco}}$ ,  $\mu_v$  first grows and then stabilizes at a bounded value.

When switching to Stage 2 (purple background), the model requires far fewer epochs to establish a positive  $\mu$  for  $c = 2$ . Moreover, this pattern generalizes to  $c = 3$  and  $c = 4$ , even though the model was never explicitly trained to generate more than two continuous thoughts. This “length generalization” indicates that once superposition emerges in earlier stages, later stages can quickly reuse it to expand the frontier further.

We also trained with a variant of  $\mathcal{L}^{\text{BFS}}$ . Compared to  $\mathcal{L}^{\text{coco}}$ , the attention logit difference when  $c = 1$  did not saturate but kept increasing to much higher values, consistent with the analysis in [Theorem 9](#). Detailed experiments and plots are provided in [Appendix D.5.2](#).

### 5.5.2 Answer Prediction

We next analyze how the model predicts the final answer. According to [Lemma 6](#), the prediction relies on two signals. The first is the *residual carryover*, which brings the explored nodes in the last thought  $[\mathbf{t}_c]$  into the answer token with strength  $\mu_A$ . Concretely, this

corresponds to the first-layer attention from  $\langle A \rangle$  to  $[t_c]$ , which copies the superposition of reachable nodes. The second is the *candidate lift*, which raises the logits of the two candidate nodes with strength  $\mu_R$ . Since  $\langle R \rangle$  copies the candidate nodes in the first layer, the second-layer attention from  $\langle A \rangle$  to  $\langle R \rangle$  serves as a proxy for  $\mu_R$ .<sup>1</sup>

Figure 5.4 shows the dynamics of these two proxies. Once training enters the *prediction* stage, both  $\mu_A$  and  $\mu_R$  increase rapidly and stabilize after roughly 5 epochs. This observation is consistent with Theorem 11, which states that  $\mu_A$  and  $\mu_R$  grow at comparable rates, ensuring that the reachable candidate  $c^*$  attains the highest logit. In contrast to the unbounded growth predicted in theory, we observe the logits plateau in practice. A possible reason is that, in practice, prediction-stage training also interacts with thought generation, whereas the theory assumes fixed thought distributions to focus on the relationship between  $\mu_R$  and  $\mu_A$ . We leave a more detailed analysis to future work.

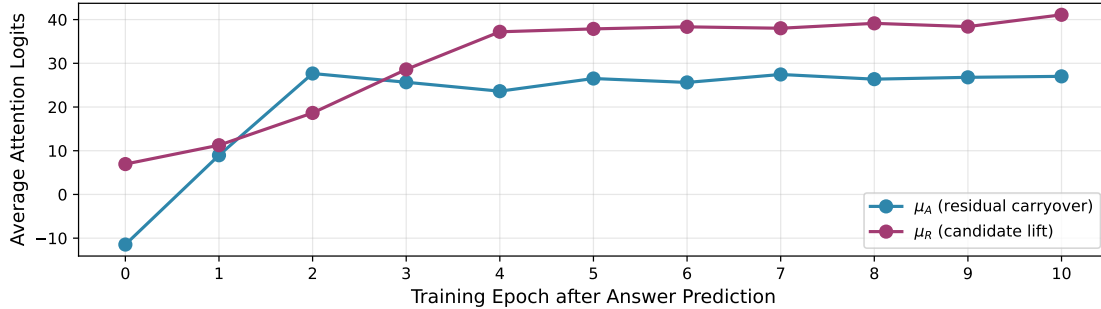


Figure 5.4: Training dynamics of the proxies for  $\mu_A$  (residual carryover) and  $\mu_R$  (candidate lift).

## 5.6 Conclusions

In this work, we study the emergence of superposition when training with continuous CoT. In particular, we theoretically analyze the training dynamics of a simplified two-layer transformer on the directed graph reachability problem. Our analysis shows that under mild assumptions, the index-matching logit, an important quantity showing the strength of the model’s local search ability, remains bounded during training. A bounded index-matching logit effectively balances exploration and exploitation during the reasoning process and thus enables implicit parallel thinking, which naturally results in superposition. Our experimental results, which track the growth of logits, further validate our theory. We expect our theoretical analysis to bring new insights into a deeper understanding of the mechanism of continuous CoT and ultimately scaling up this promising paradigm more efficiently and reliably.

<sup>1</sup>We observe that under different experimental settings and random seeds, the *candidate lift* effect is not always mediated by the  $\langle R \rangle$  token; alternative attention routes are presented in the Appendix D.5.3.

# Bibliography

- [1] Emmanuel Abbe et al. “Generalization on the unseen, logic reasoning and degree curriculum”. In: *Journal of Machine Learning Research* 25.331 (2024), pp. 1–58.
- [2] Emmanuel Abbe et al. “Learning to reason with neural networks: Generalization, unseen data and boolean measures”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 2709–2722.
- [3] Ekin Akyürek et al. “What learning algorithm is in-context learning? investigations with linear models”. In: *arXiv preprint arXiv:2211.15661* (2022).
- [4] Jean-Baptiste Alayrac et al. “Flamingo: a visual language model for few-shot learning”. In: *Advances in neural information processing systems* 35 (2022), pp. 23716–23736.
- [5] Zeyuan Allen-Zhu and Yuanzhi Li. “Physics of language models: Part 3.2, knowledge manipulation”. In: *arXiv preprint arXiv:2309.14402* (2023).
- [6] Anthropic. *Claude Opus 4.5 System Card*. <https://assets.anthropic.com/m/64823ba7485345a7/Claude-Opus-4-5-System-Card.pdf>. System card. Nov. 2025.
- [7] Anthropic. *System Card: Claude Opus 4 & Claude Sonnet 4*. 2025. URL: <https://www.anthropic.com/claude-4-system-card>.
- [8] Sanjeev Arora et al. “Implicit Regularization in Deep Matrix Factorization”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2019/file/c0c783b5fc0d7d808f1d14a6e9c8280d-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2019/file/c0c783b5fc0d7d808f1d14a6e9c8280d-Paper.pdf).
- [9] Mikita Balesni, Tomasz Korbak, and Owain Evans. “The two-hop curse: LLMs trained on  $A \rightarrow B$ ,  $B \rightarrow C$  fail to learn  $A \rightarrow C$ ”. In: (2024).
- [10] Lukas Berglund et al. “Taken out of context: On measuring situational awareness in llms”. In: *arXiv preprint arXiv:2309.00667* (2023).
- [11] Lukas Berglund et al. “The Reversal Curse: LLMs trained on” A is B” fail to learn” B is A””. In: *arXiv preprint arXiv:2309.12288* (2023).
- [12] Jan Betley et al. “Weird generalization and inductive backdoors: New ways to corrupt llms”. In: *arXiv preprint arXiv:2512.09742* (2025).

- [13] Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. “On the ability and limitations of transformers to recognize formal languages”. In: *arXiv preprint arXiv:2009.11264* (2020).
- [14] Satwik Bhattamishra, Arkil Patel, and Navin Goyal. “On the computational power of transformers and its implications in sequence modeling”. In: *arXiv preprint arXiv:2006.09286* (2020).
- [15] Alberto Bietti et al. “Birth of a transformer: A memory viewpoint”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 1560–1588.
- [16] Eden Biran et al. “Hopping too late: Exploring the limitations of large language models on multi-hop queries”. In: *arXiv preprint arXiv:2406.12775* (2024).
- [17] Arno Böhm. *Quantum mechanics: foundations and applications*. Springer Science & Business Media, 2013.
- [18] Daniil A Boiko et al. “Autonomous chemical research with large language models”. In: *Nature* 624.7992 (2023), pp. 570–578.
- [19] Enric Boix-Adsera et al. “Transformers learn through gradual rank increase”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 24519–24551.
- [20] Jannik Brinkmann et al. “A Mechanistic Analysis of a Transformer Trained on a Symbolic Multi-Step Reasoning Task”. In: *arXiv preprint arXiv:2402.11917* (2024).
- [21] Tom Brown et al. “Language models are few-shot learners”. In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.
- [22] Natasha Butt et al. “Soft tokens, hard truths”. In: *arXiv preprint arXiv:2509.19170* (2025).
- [23] Ruisheng Cao et al. “Qwen3-coder-next technical report”. In: *arXiv preprint arXiv:2603.00729* (2026).
- [24] Yupeng Chang et al. “A survey on evaluation of large language models”. In: *ACM Transactions on Intelligent Systems and Technology* (2023).
- [25] Lei Chen, Joan Bruna, and Alberto Bietti. “Distributional associations vs in-context reasoning: A study of feed-forward and attention layers”. In: *The Thirteenth International Conference on Learning Representations*. 2025.
- [26] Mark Chen et al. “Evaluating large language models trained on code”. In: *arXiv preprint arXiv:2107.03374* (2021).
- [27] Xingyu Chen et al. “Do not think that much for  $2+3=?$  on the overthinking of o1-like llms”. In: *arXiv preprint arXiv:2412.21187* (2024).
- [28] Jeffrey Cheng and Benjamin Van Durme. “Compressed chain of thought: Efficient reasoning through dense representations”. In: *arXiv preprint arXiv:2412.13171* (2024).
- [29] Karl Cobbe et al. “Training verifiers to solve math word problems”. In: *arXiv preprint arXiv:2110.14168* (2021).

- [30] Andrew Cohen et al. “Spectral Journey: How Transformers Predict the Shortest Path”. In: *arXiv preprint arXiv:2502.08794* (2025).
- [31] Roi Cohen et al. “Evaluating the ripple effects of knowledge editing in language models”. In: *Transactions of the Association for Computational Linguistics* 12 (2024), pp. 283–298.
- [32] Xinnan Dai et al. “How Do Large Language Models Understand Graph Patterns? A Benchmark for Graph Pattern Comprehension”. In: *arXiv preprint arXiv:2410.05298* (2024).
- [33] Steven Diamond and Stephen Boyd. “CVXPY: A Python-embedded modeling language for convex optimization”. In: *Journal of Machine Learning Research* 17.83 (2016), pp. 1–5.
- [34] Benjamin L Edelman et al. “Inductive biases and variable creation in self-attention mechanisms”. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 5793–5831.
- [35] Ahmed Elnaggar et al. “ProtTrans: toward understanding the language of life through self-supervised learning”. In: *IEEE transactions on pattern analysis and machine intelligence* 44.10 (2021), pp. 7112–7127.
- [36] Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. “Talk like a graph: Encoding graphs for large language models”. In: *arXiv preprint arXiv:2310.04560* (2023).
- [37] Guhao Feng et al. “Towards revealing the mystery behind chain of thought: a theoretical perspective”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 70757–70798.
- [38] Guhao Feng et al. “Towards revealing the mystery behind chain of thought: a theoretical perspective”. In: *Advances in Neural Information Processing Systems* 36 (2024).
- [39] Jiahai Feng, Stuart Russell, and Jacob Steinhardt. “Extractive Structures Learned in Pretraining Enable Generalization on Finetuned Facts”. In: *arXiv preprint arXiv:2412.04614* (2024).
- [40] Hengyu Fu et al. “What can a single attention layer learn? a study through the random features lens”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 11912–11951.
- [41] Zorik Gekhman et al. “Does fine-tuning LLMs on new knowledge encourage hallucinations?” In: *arXiv preprint arXiv:2405.05904* (2024).
- [42] Olga Golovneva et al. “Reverse Training to Nurse the Reversal Curse”. In: *arXiv preprint arXiv:2403.13799* (2024).
- [43] Google DeepMind. *AI achieves silver-medal standard solving International Mathematical Olympiad problems*. <https://deepmind.google/discover/blog/ai-solves-imo-problems-at-silver-medal-level/>. 2024.

- [44] Sachin Goyal et al. “Think before you speak: Training language models with pause tokens”. In: *arXiv preprint arXiv:2310.02226* (2023).
- [45] Halil Alperen Gozeten et al. “Continuous chain of thought enables parallel exploration and reasoning”. In: *arXiv preprint arXiv:2505.23648* (2025).
- [46] Suriya Gunasekar et al. “Characterizing Implicit Bias in Terms of Optimization Geometry”. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 1832–1841. URL: <https://proceedings.mlr.press/v80/gunasekar18a.html>.
- [47] Suriya Gunasekar et al. “Implicit bias of gradient descent on linear convolutional networks”. In: *Advances in neural information processing systems* 31 (2018).
- [48] Suriya Gunasekar et al. “Implicit regularization in matrix factorization”. In: *Advances in neural information processing systems* 30 (2017).
- [49] Daya Guo et al. “Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning”. In: *arXiv preprint arXiv:2501.12948* (2025).
- [50] Jiayan Guo et al. “Gpt4graph: Can large language models understand graph structured data? an empirical evaluation and benchmarking”. In: *arXiv preprint arXiv:2305.15066* (2023).
- [51] Qingyan Guo et al. “Mitigating Reversal Curse via Semantic-aware Permutation Training”. In: *arXiv preprint arXiv:2403.00758* (2024).
- [52] Tianyu Guo et al. “Active-dormant attention heads: Mechanistically demystifying extreme-token phenomena in llms”. In: *arXiv preprint arXiv:2410.13835* (2024).
- [53] Tianyu Guo et al. “How Do LLMs Perform Two-Hop Reasoning in Context?” In: *arXiv preprint arXiv:2502.13913* (2025).
- [54] Simon Jerome Han et al. “Inductive reasoning in humans and large language models”. In: *Cognitive Systems Research* 83 (2024), p. 101155.
- [55] Shibo Hao et al. “Training large language models to reason in a continuous latent space”. In: *arXiv preprint arXiv:2412.06769* (2024).
- [56] Tim Hoheisel and Elliot Paquette. “Uniqueness in Nuclear Norm Minimization: Flatness of the Nuclear Norm Sphere and Simultaneous Polarization”. In: *Journal of Optimization Theory and Applications* 197.1 (Feb. 2023), pp. 252–276. DOI: <https://doi.org/10.1007/s10957-023-02167-7>. URL: <https://link.springer.com/article/10.1007/s10957-023-02167-7>.
- [57] Wenlong Huang et al. “Language models as zero-shot planners: Extracting actionable knowledge for embodied agents”. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 9118–9147.

- [58] Yixiao Huang et al. “Transformers Provably Learn to Internalize Chain-of-Thought”. In: *Workshop on Latent  $\{\backslash\&\}$  Implicit Thinking  $\{\backslash\textit{textendash}\}$  Going Beyond CoT Reasoning*.
- [59] Yu Huang, Yuan Cheng, and Yingbin Liang. “In-context convergence of transformers”. In: *Proceedings of the 41st International Conference on Machine Learning*. 2024, pp. 19660–19722.
- [60] Binyuan Hui et al. “Qwen2.5-Coder Technical Report”. In: *arXiv preprint arXiv:2409.12186* (2024).
- [61] Aaron Hurst et al. “Gpt-4o system card”. In: *arXiv preprint arXiv:2410.21276* (2024).
- [62] M Emrullah Ildiz et al. “From self-attention to markov models: Unveiling the dynamics of generative transformers”. In: *arXiv preprint arXiv:2402.13512* (2024).
- [63] Aaron Jaech et al. “Openai o1 system card”. In: *arXiv preprint arXiv:2412.16720* (2024).
- [64] Samy Jelassi, Michael Sander, and Yuanzhi Li. “Vision transformers provably learn spatial structure”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 37822–37836.
- [65] Ziwei Ji and Matus Telgarsky. “The implicit bias of gradient descent on nonseparable data”. In: *Proceedings of the Thirty-Second Conference on Learning Theory*. Ed. by Alina Beygelzimer and Daniel Hsu. Vol. 99. Proceedings of Machine Learning Research. PMLR, 25–28 Jun 2019, pp. 1772–1798. URL: <https://proceedings.mlr.press/v99/ji19a.html>.
- [66] Carlos E Jimenez et al. “Swe-bench: Can language models resolve real-world github issues?” In: *International Conference on Learning Representations*. Vol. 2024. 2024, pp. 54107–54157.
- [67] Jaehun Jung et al. “Maieutic prompting: Logically consistent reasoning with recursive explanations”. In: *arXiv preprint arXiv:2205.11822* (2022).
- [68] Subbarao Kambhampati. “Can large language models reason and plan?” In: *Annals of the New York Academy of Sciences* 1534.1 (2024), pp. 15–18.
- [69] Katie Kang et al. “Unfamiliar Finetuning Examples Control How Language Models Hallucinate”. In: *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. Ed. by Luis Chiruzzo, Alan Ritter, and Lu Wang. Albuquerque, New Mexico: Association for Computational Linguistics, Apr. 2025, pp. 3600–3612. ISBN: 979-8-89176-189-6. DOI: [10.18653/v1/2025.naacl-long.183](https://doi.org/10.18653/v1/2025.naacl-long.183). URL: <https://aclanthology.org/2025.naacl-long.183/>.
- [70] Tushar Khot et al. “Decomposed prompting: A modular approach for solving complex tasks”. In: *arXiv preprint arXiv:2210.02406* (2022).

- [71] Juno Kim and Taiji Suzuki. “Transformers provably solve parity efficiently with chain of thought”. In: *arXiv preprint arXiv:2410.08633* (2024).
- [72] Diederik P Kingma. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).
- [73] Takeshi Kojima et al. “Large language models are zero-shot reasoners”. In: *Advances in neural information processing systems* 35 (2022), pp. 22199–22213.
- [74] Dmitrii Krasheninnikov, Egor Krasheninnikov, and David Krueger. “Out-of-context meta-learning in large language models”. In: *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*. 2023.
- [75] Beatrice Laurent and Pascal Massart. “Adaptive estimation of a quadratic functional by model selection”. In: *Annals of Statistics* (2000), pp. 1302–1338.
- [76] Yingcong Li et al. “Mechanics of next token prediction with self-attention”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2024, pp. 685–693.
- [77] Yuanzhi Li, Tengyu Ma, and Hongyang Zhang. “Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations”. In: *Conference On Learning Theory*. PMLR. 2018, pp. 2–47.
- [78] Zhiyuan Li, Yuping Luo, and Kaifeng Lyu. “Towards resolving the implicit bias of gradient descent for matrix factorization: Greedy low-rank learning”. In: *arXiv preprint arXiv:2012.09839* (2020).
- [79] Zhiyuan Li et al. “Chain of thought empowers transformers to solve inherently serial problems”. In: *arXiv preprint arXiv:2402.12875* 1 (2024).
- [80] Valerii Likhoshesterov, Krzysztof Choromanski, and Adrian Weller. “On the expressive power of self-attention matrices”. In: *arXiv preprint arXiv:2106.03764* (2021).
- [81] Zeming Lin et al. “Evolutionary-scale prediction of atomic-level protein structure with a language model”. In: *Science* 379.6637 (2023), pp. 1123–1130.
- [82] Bingbin Liu et al. “Transformers learn shortcuts to automata”. In: *arXiv preprint arXiv:2210.10749* (2022).
- [83] Charles London and Varun Kanade. “Pause Tokens Strictly Increase the Expressivity of Constant-Depth Transformers”. In: *arXiv preprint arXiv:2505.21024* (2025).
- [84] Zihan Luo et al. “Graphinstruct: Empowering large language models with graph understanding and reasoning capability”. In: *arXiv preprint arXiv:2403.04483* (2024).
- [85] Ang Lv et al. “Are we falling in a middle-intelligence trap? an analysis and mitigation of the reversal curse”. In: *arXiv preprint arXiv:2311.07468* (2023).
- [86] Kaifeng Lyu and Jian Li. “Gradient descent maximizes the margin of homogeneous neural networks”. In: *arXiv preprint arXiv:1906.05890* (2019).

- [87] Andres M. Bran et al. “Augmenting large language models with chemistry tools”. In: *Nature machine intelligence* 6.5 (2024), pp. 525–535.
- [88] Arvind Mahankali, Tatsunori B Hashimoto, and Tengyu Ma. “One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention”. In: *arXiv preprint arXiv:2307.03576* (2023).
- [89] Alex Mallen et al. “When not to trust language models: Investigating effectiveness of parametric and non-parametric memories”. In: *arXiv preprint arXiv:2212.10511* (2022).
- [90] William Merrill and Ashish Sabharwal. “A little depth goes a long way: The expressive power of log-depth transformers”. In: *arXiv preprint arXiv:2503.03961* (2025).
- [91] William Merrill and Ashish Sabharwal. “The expressive power of transformers with chain of thought”. In: *arXiv preprint arXiv:2310.07923* (2023).
- [92] William Merrill and Ashish Sabharwal. “The parallelism tradeoff: Limitations of log-precision transformers”. In: *Transactions of the Association for Computational Linguistics* 11 (2023), pp. 531–545.
- [93] Mor Shpigel Nacson et al. “Convergence of gradient descent on separable data”. In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 3420–3428.
- [94] Mor Shpigel Nacson et al. “Lexicographic and depth-sensitive margins in homogeneous and non-homogeneous deep models”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 4683–4692.
- [95] Quan Nguyen and Thanh Nguyen-Tang. “One-Layer Transformers are Provably Optimal for In-context Reasoning and Distributional Association Learning in Next-Token Prediction Tasks”. In: *arXiv preprint arXiv:2505.15009* (2025).
- [96] Eshaan Nichani, Alex Damian, and Jason D Lee. “How Transformers Learn Causal Structure with Gradient Descent”. In: *International Conference on Machine Learning*. PMLR. 2024, pp. 38018–38070.
- [97] Eshaan Nichani, Jason D Lee, and Alberto Bietti. “Understanding factual recall in transformers via associative memories”. In: *arXiv preprint arXiv:2412.06538* (2024).
- [98] Harsha Nori et al. “Capabilities of gpt-4 on medical challenge problems”. In: *arXiv preprint arXiv:2303.13375* (2023).
- [99] Maxwell Nye et al. “Show your work: Scratchpads for intermediate computation with language models”. In: *arXiv preprint arXiv:2112.00114* (2021).
- [100] Catherine Olsson et al. “In-context learning and induction heads”. In: *arXiv preprint arXiv:2209.11895* (2022).
- [101] OpenAI. *OpenAI o3 and o4-mini*. 2025. URL: <https://openai.com/index/introducing-o3-and-o4-mini>.

- [102] Shishir G Patil et al. “Gorilla: Large language model connected with massive apis”. In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 126544–126565.
- [103] Letian Peng et al. “Linear Correlation in LM’s Compositional Generalization and Hallucination”. In: *arXiv preprint arXiv:2502.04520* (2025).
- [104] Sida Peng et al. “The impact of ai on developer productivity: Evidence from github copilot”. In: *arXiv preprint arXiv:2302.06590* (2023).
- [105] Jorge Pérez, Pablo Barceló, and Javier Marinkovic. “Attention is turing-complete”. In: *Journal of Machine Learning Research* 22.75 (2021), pp. 1–35.
- [106] Jacob Pfau, William Merrill, and Samuel R Bowman. “Let’s think dot by dot: Hidden computation in transformer language models”. In: *arXiv preprint arXiv:2404.15758* (2024).
- [107] Qwen Team. *Qwen3.5: Towards Native Multimodal Agents*. Feb. 2026. URL: <https://qwen.ai/blog?id=qwen3.5>.
- [108] Noam Razin and Nadav Cohen. “Implicit regularization in deep learning may not be explainable by norms”. In: *Advances in neural information processing systems* 33 (2020), pp. 21174–21187.
- [109] Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. “Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization”. In: *SIAM review* 52.3 (2010), pp. 471–501.
- [110] Laria Reynolds and Kyle McDonell. “Prompt programming for large language models: Beyond the few-shot paradigm”. In: *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–7.
- [111] Bernardino Romera-Paredes et al. “Mathematical discoveries from program search with large language models”. In: *Nature* 625.7995 (2024), pp. 468–475.
- [112] Baptiste Roziere et al. “Code llama: Open foundation models for code”. In: *arXiv preprint arXiv:2308.12950* (2023).
- [113] Clayton Sanford, Daniel Hsu, and Matus Telgarsky. “One-layer transformers fail to solve the induction heads task”. In: *arXiv preprint arXiv:2408.14332* (2024).
- [114] Clayton Sanford, Daniel Hsu, and Matus Telgarsky. “Transformers, parallel computation, and logarithmic depth”. In: *arXiv preprint arXiv:2402.09268* (2024).
- [115] Clayton Sanford et al. “Understanding transformer reasoning capabilities via graph algorithms”. In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 78320–78370.
- [116] Zhihong Shao et al. “Deepseekmath: Pushing the limits of mathematical reasoning in open language models”. In: *arXiv preprint arXiv:2402.03300* (2024).
- [117] Heejune Sheen et al. “Implicit regularization of gradient flow on one-layer softmax attention”. In: *arXiv preprint arXiv:2403.08699* (2024).

- [118] Karan Singhal et al. “Large language models encode clinical knowledge”. In: *Nature* 620.7972 (2023), pp. 172–180.
- [119] Karan Singhal et al. “Toward expert-level medical question answering with large language models”. In: *Nature medicine* 31.3 (2025), pp. 943–950.
- [120] Daniel Soudry et al. “The implicit bias of gradient descent on separable data”. In: *Journal of Machine Learning Research* 19.70 (2018), pp. 1–57.
- [121] Dominik Stöger and Mahdi Soltanolkotabi. “Small random initialization is akin to spectral learning: Optimization and generalization guarantees for overparameterized low-rank matrix reconstruction”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 23831–23843.
- [122] Dijia Su et al. “Token Assorted: Mixing Latent and Text Tokens for Improved Language Model Reasoning”. In: *International Conference on Machine Learning*. PMLR. 2025, pp. 57144–57163.
- [123] Jianlin Su et al. “Roformer: Enhanced transformer with rotary position embedding”. In: *Neurocomputing* 568 (2024), p. 127063.
- [124] Chen Sun et al. “How new data permeates LLM knowledge and how to dilute it”. In: *arXiv preprint arXiv:2504.09522* (2025).
- [125] Zhongxiang Sun et al. “Detection and mitigation of hallucination in large reasoning models: A mechanistic perspective”. In: *arXiv preprint arXiv:2505.12886* (2025).
- [126] Davoud Ataee Tarzanagh et al. “Max-margin token selection in attention mechanism”. In: *Advances in neural information processing systems* 36 (2023), pp. 48314–48362.
- [127] Davoud Ataee Tarzanagh et al. “Transformers as support vector machines”. In: *arXiv preprint arXiv:2308.16898* (2023).
- [128] Ross Taylor et al. “Galactica: A large language model for science”. In: *arXiv preprint arXiv:2211.09085* (2022).
- [129] Gemini Team et al. “Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context”. In: *arXiv preprint arXiv:2403.05530* (2024).
- [130] Kimi Team et al. “Kimi k2: Open agentic intelligence”. In: *arXiv preprint arXiv:2507.20534* (2025).
- [131] Arun James Thirunavukarasu et al. “Large language models in medicine”. In: *Nature medicine* 29.8 (2023), pp. 1930–1940.
- [132] Yuandong Tian et al. “Joma: Demystifying multilayer transformers via joint dynamics of mlp and attention”. In: *arXiv preprint arXiv:2310.00535* (2023).
- [133] Yuandong Tian et al. “Scan and snap: Understanding training dynamics and token composition in 1-layer transformer”. In: *Advances in neural information processing systems* 36 (2023), pp. 71911–71947.

- [134] Karthik Valmeekam, Kaya Stechly, and Subbarao Kambhampati. “LLMs Still Can’t Plan; Can LRMs? A Preliminary Evaluation of OpenAI’s o1 on PlanBench”. In: *arXiv preprint arXiv:2409.13373* (2024).
- [135] Karthik Valmeekam et al. “Large language models still can’t plan (a benchmark for llms on planning and reasoning about change)”. In: *arXiv preprint arXiv:2206.10498* (2022).
- [136] Gal Vardi, Ohad Shamir, and Nati Srebro. “On margin maximization in linear and relu networks”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 37024–37036.
- [137] Bhavya Vasudeva, Puneesh Deora, and Christos Thrampoulidis. “Implicit bias and fast convergence rates for self-attention”. In: *arXiv preprint arXiv:2402.05738* (2024).
- [138] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).
- [139] Boshi Wang et al. “Grokking transformers are implicit reasoners: A mechanistic journey to the edge of generalization”. In: *arXiv preprint arXiv:2405.15071* (2024).
- [140] Heng Wang et al. “Can language models solve graph problems in natural language?”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 30840–30861.
- [141] Peiyi Wang et al. “Math-shepherd: Verify and reinforce llms step-by-step without human annotations”. In: *arXiv preprint arXiv:2312.08935* (2023).
- [142] Xinyi Wang et al. “Guiding language model reasoning with planning tokens”. In: *arXiv preprint arXiv:2310.05707* (2023).
- [143] Xinyi Wang et al. “Understanding the reasoning ability of language models from the perspective of reasoning paths aggregation”. In: *arXiv preprint arXiv:2402.03268* (2024).
- [144] Xuezhi Wang et al. “Self-consistency improves chain of thought reasoning in language models”. In: *arXiv preprint arXiv:2203.11171* (2022).
- [145] Jason Wei et al. “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models”. In: *arXiv preprint arXiv:2201.11903* (2022).
- [146] Kaiyue Wen et al. “From sparse dependence to sparse attention: unveiling how chain-of-thought enhances transformer sample efficiency”. In: *arXiv preprint arXiv:2410.05459* (2024).
- [147] Thomas Wolf et al. “Transformers: State-of-the-Art Natural Language Processing”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. URL: <https://www.aclweb.org/anthology/2020.emnlp-demos.6>.
- [148] Jian Xie et al. “Travelplanner: A benchmark for real-world planning with language agents”. In: *arXiv preprint arXiv:2402.01622* (2024).

- [149] Sang Michael Xie et al. “An explanation of in-context learning as implicit bayesian inference”. In: *arXiv preprint arXiv:2111.02080* (2021).
- [150] John Yang et al. “Swe-agent: Agent-computer interfaces enable automated software engineering”. In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 50528–50652.
- [151] Kaiyu Yang et al. “Leandojo: Theorem proving with retrieval-augmented language models”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 21573–21612.
- [152] Sohee Yang et al. “Do large language models latently perform multi-hop reasoning?”. In: *arXiv preprint arXiv:2402.16837* (2024).
- [153] Shunyu Yao et al. “Self-attention networks can process bounded hierarchical languages”. In: *arXiv preprint arXiv:2105.11115* (2021).
- [154] Tian Ye et al. “Physics of language models: Part 2.1, grade-school math and the hidden reasoning process”. In: *The Thirteenth International Conference on Learning Representations*. 2024.
- [155] Longhui Yu et al. “Metamath: Bootstrap your own mathematical questions for large language models”. In: *arXiv preprint arXiv:2309.12284* (2023).
- [156] Xiang Yue et al. “Mammoth: Building math generalist models through hybrid instruction tuning”. In: *arXiv preprint arXiv:2309.05653* (2023).
- [157] Xiang Yue et al. “Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2024, pp. 9556–9567.
- [158] Chulhee Yun et al. “Are transformers universal approximators of sequence-to-sequence functions?”. In: *arXiv preprint arXiv:1912.10077* (2019).
- [159] Eric Zelikman et al. “Star: Bootstrapping reasoning with reasoning”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 15476–15488.
- [160] Dylan Zhang et al. “Transformer-based models are not yet perfect at learning to emulate structural recursion”. In: *arXiv preprint arXiv:2401.12947* (2024).
- [161] Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. “Trained transformers learn linear models in-context”. In: *Journal of Machine Learning Research* 25.49 (2024), pp. 1–55.
- [162] Yedi Zhang et al. “Training dynamics of in-context learning in linear attention”. In: *arXiv preprint arXiv:2501.16265* (2025).
- [163] Zhen Zhang et al. “Soft thinking: Unlocking the reasoning potential of llms in continuous concept space”. In: *arXiv preprint arXiv:2505.15778* (2025).
- [164] Huaixiu Steven Zheng et al. “Natural plan: Benchmarking llms on natural language planning”. In: *arXiv preprint arXiv:2406.04520* (2024).

- [165] Denny Zhou et al. “Least-to-most prompting enables complex reasoning in large language models”. In: *arXiv preprint arXiv:2205.10625* (2022).
- [166] Yang Zhou et al. “GSM-Infinite: How Do Your LLMs Behave over Infinitely Increasing Context Length and Reasoning Complexity?” In: *arXiv preprint arXiv:2502.05252* (2025).
- [167] Hanlin Zhu et al. “Emergence of superposition: Unveiling the training dynamics of chain of continuous thought”. In: *arXiv preprint arXiv:2509.23365* (2025).
- [168] Hanlin Zhu et al. “GSM-Agent: Understanding Agentic Reasoning Using Controllable Environments”. In: *arXiv preprint arXiv:2509.21998* (2025).
- [169] Hanlin Zhu et al. “Reasoning by superposition: A theoretical perspective on chain of continuous thought”. In: *arXiv preprint arXiv:2505.12514* (2025).
- [170] Hanlin Zhu et al. “Towards a Theoretical Understanding of the ‘Reversal Curse’ via Training Dynamics”. In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 90473–90513.
- [171] Qihao Zhu et al. “DeepSeek-Coder-V2: Breaking the Barrier of Closed-Source Models in Code Intelligence”. In: *arXiv preprint arXiv:2406.11931* (2024).

# Appendix A

## Appendix for the Reversal Curse Theory

### A.1 Additional Notations

Let  $\delta_{ij} = 1$  for  $i = j$  and  $\delta_{ij} = 0$  for  $i \neq j$ . For a squared matrix  $A \in \mathbb{R}^{d \times d}$ , its trace is  $\text{Tr}(A) = \sum_{i=1}^d A_{ii}$ . For two matrices  $A, B \in \mathbb{R}^{m \times n}$  of the same shape, their inner product is defined as  $\langle A, B \rangle = \text{Tr}(AB^\top)$ . For any matrix  $A \in \mathbb{R}^{m \times n}$ , its (Frobenius) norm is defined as  $\|A\| = \sqrt{\langle A, A \rangle}$ . For any vector  $\mathbf{x} = (x_1, \dots, x_d)^\top \in \mathbb{R}^d$  or a matrix  $A \in \mathbb{R}^{m \times n}$ , we define the zero-norm as  $\|\mathbf{x}\|_0 = \sum_{i=1}^d \mathbb{1}\{x_i \neq 0\}$  or  $\|A\|_0 = \sum_{i=1}^m \sum_{j=1}^n \mathbb{1}\{A_{ij} \neq 0\}$  where  $\mathbb{1}\{\cdot\}$  is the indicator function.

### A.2 Missing Proofs of Section 2.3

**Theorem 12** (Separation of training dynamics, formal statement of Theorem 1). *Fix arbitrary  $\delta, \epsilon \in (0, 1)$ . Let  $v_1, \dots, v_m$  be independently sampled from  $\mathcal{N}_d(\mathbf{0}_d, \frac{1}{d}I_d)$ . Let  $x_1, \dots, x_n$  and  $y_1, \dots, y_n$  be sampled uniformly at random from  $\{v_1, \dots, v_m\}$  without replacement. Define*

$$\begin{aligned} \mathcal{L}(\Theta) &= \frac{1}{2n-1} \left( \sum_{i=1}^n -\log p_\Theta(y_i|x_i) + \sum_{i=2}^n -\log p_\Theta(x_i|y_i) \right) \\ \mathcal{L}^{rev}(\Theta) &= -\log p_\Theta(x_1|y_1). \end{aligned}$$

Consider the gradient flow  $\Theta_t : t \geq 0$

$$\frac{d\Theta_t}{dt} = -\nabla \mathcal{L}(\Theta_t).$$

where  $\Theta_0 \sim \mathcal{N}(\mathbf{0}^{\otimes 2}, \sigma^2 \cdot I^{\otimes 2})$  or  $\Theta_0$  satisfies  $\frac{1}{2m} < p_{\Theta_0}(y_i|x_i), p_{\Theta_0}(x_i|y_i) < \frac{2}{m}$  for all  $i \in [n]$ . Suppose  $\sigma \leq \frac{1}{100 \ln(64m^2/\delta)}$  and

$$d \geq \frac{10^6 n^4 m^2 \log^4(2m) \log(64m^2 n^2 / \delta)}{\epsilon^2}.$$

With probability at least  $1 - \delta$ , we have

$$\frac{\mathcal{L}^{\text{rev}}(\Theta_t)}{\mathcal{L}^{\text{rev}}(\Theta_0)} \geq \left( \frac{\mathcal{L}(\Theta_t)}{\mathcal{L}(\Theta_0)} \right)^\epsilon, \quad \forall t \geq 0.$$

*Proof.* Let  $v = \sqrt{\frac{400n^2 m^2 \log(64m^2 n^2 / \delta)}{d}}$ . By Lemma 7 and Lemma 8, with probability at least  $1 - \delta$ ,

$$\begin{aligned} \mathcal{L}^{\text{rev}}(\Theta_t) &\geq \mathcal{L}^{\text{rev}}(\Theta_0) \cdot \left( 1 + \frac{\mathcal{L}(\Theta_0) \cdot t}{8(2n-1) \log^2(2m)} \right)^{-8v(2n-1) \log^2(2m)} \\ &\geq \mathcal{L}^{\text{rev}}(\Theta_0) \cdot \left( \frac{\mathcal{L}(\Theta_t)}{\mathcal{L}(\Theta_0)} \right)^{8v(2n-1) \log^2(2m)} \end{aligned}$$

By definition of  $d$ , we have  $8v(2n-1) \log^2(2m) \leq \epsilon$ . Notice that  $\frac{\mathcal{L}(\Theta_t)}{\mathcal{L}(\Theta_0)} < 1$ , thus

$$\mathcal{L}^{\text{rev}}(\Theta_t) \geq \mathcal{L}^{\text{rev}}(\Theta_0) \cdot \left( \frac{\mathcal{L}(\Theta_t)}{\mathcal{L}(\Theta_0)} \right)^\epsilon.$$

□

**Theorem 13** (Lower bound of reversal loss, formal statement of Theorem 2). *Fix arbitrary  $c > 0$  and  $C \leq \log(m/2)$ . Under the setting of Theorem 12, suppose  $\sigma \leq \frac{1}{100 \ln(64m^2/\delta)}$  and*

$$d \geq 10^6 n^4 m^2 \log^4(2m) \log(64m^2 n^2 / \delta) \cdot \frac{\log^2 \frac{c}{\log(2m)}}{\log^2 \frac{C}{\log(m/2)}}.$$

With probability at least  $1 - \delta$ ,

$$\mathcal{L}^{\text{rev}}(\Theta_\tau) \geq C.$$

where  $\tau$  denotes the first time such that  $\mathcal{L}(\Theta_t) \leq c$ .

*Proof.* By continuity,  $\mathcal{L}(\Theta_\tau) = c$ . By Theorem 12, when

$$d \geq \frac{10^6 n^4 m^2 \log^4(2m) \log(64m^2 n^2 / \delta)}{\epsilon^2} \tag{A.1}$$

with probability at least  $1 - \delta$ ,

$$\begin{aligned}\mathcal{L}^{\text{rev}}(\Theta_\tau) &\geq \mathcal{L}^{\text{rev}}(\Theta_0) \cdot \left( \frac{\mathcal{L}(\Theta_\tau)}{\mathcal{L}(\Theta_0)} \right)^\epsilon \\ &\geq \mathcal{L}^{\text{rev}}(\Theta_0) \cdot \left( \frac{c}{\mathcal{L}(\Theta_0)} \right)^\epsilon.\end{aligned}$$

Under this event, applying Lemma 10, we can obtain

$$\mathcal{L}^{\text{rev}}(\Theta_\tau) \geq \log(m/2) \cdot \left( \frac{c}{\log(2m)} \right)^\epsilon.$$

To ensure that the right hand side is  $C$ , we set  $\epsilon = \frac{\log \frac{C}{\log(m/2)}}{\log \frac{c}{\log(2m)}}$ . One may check that the definition of  $d$  satisfies Eq. (A.1). It follows that

$$\mathcal{L}^{\text{rev}}(\Theta_\tau) \geq C.$$

□

### A.2.1 Training dynamics

**Lemma 7** (Dynamics of the forward loss). *Let  $x_1, \dots, x_n, y_1, \dots, y_n, \mathcal{L}(\Theta)$ , and  $\Theta_t$  ( $t \geq 0$ ) be defined as in Theorem 12. When  $\sigma \leq \frac{1}{100 \ln(16n^2/\delta)}$  and  $d \geq 1600n^3m^2 \log(8m^2n^2/\delta)$ , with probability at least  $1 - \delta$ , we have*

$$\mathcal{L}(\Theta_t) \leq \frac{1}{\frac{t}{8(2n-1)\log^2(2m)} + \frac{1}{\mathcal{L}(\Theta_0)}}, \quad \forall t \geq 0.$$

Furthermore,

$$\inf \{p_{\Theta_t}(x_i|y_i), p_{\Theta_t}(y_i|x_i) : t \geq 0, i \in [n]\} > \frac{1}{2m}.$$

*Proof.* For convenience, we assume  $x_1, \dots, x_n = v_1, \dots, v_n$  and  $y_1, \dots, y_n = v_{n+1}, \dots, v_{2n}$  WLOG.

Let  $\epsilon = \sqrt{\frac{400n^2m^2 \log(8m^2n^2/\delta)}{d}}$ . Then  $\epsilon \leq \frac{1}{2\sqrt{n}}$ . Define  $l_i(\Theta) = -\log p_\Theta(y_i|x_i)$  and  $l_i^{\text{rev}}(\Theta) = -\log p_\Theta(x_i|y_i)$ . Let  $\alpha_{i,j}^{(t)} = -p_{\Theta_t}(v_j|v_i) + \delta_{i,j-n}$ ,  $\beta_{i,j}^{(t)} = -p_{\Theta_t}(v_j|v_i) + \delta_{i-n,j}$ . By Lemma 9,

$$\begin{aligned}\frac{d\mathcal{L}(\Theta_t)}{dt} &= \left\langle \nabla \mathcal{L}(\Theta_t), \frac{d\Theta_t}{dt} \right\rangle = -\langle \nabla \mathcal{L}(\Theta_t), \nabla \mathcal{L}(\Theta_t) \rangle \\ &= -\left\| \frac{1}{2n-1} \left( \sum_{i=1}^n x_i (y_i - \mathbb{E}_{p_{\Theta_t}(\cdot|x_i)}[y])^\top + \sum_{i=2}^n y_i (x_i - \mathbb{E}_{p_{\Theta_t}(\cdot|y_i)}[x])^\top \right) \right\|^2 \\ &= -\left\| \frac{1}{2n-1} \left( \sum_{i=1}^n \sum_{j=1}^m \alpha_{i,j}^{(t)} v_i v_j^\top + \sum_{i=n+2}^{2n} \sum_{j=1}^m \beta_{i,j}^{(t)} v_i v_j^\top \right) \right\|^2.\end{aligned}$$

Similarly, we have

$$\begin{aligned}
& \frac{dl_i(\Theta_t)}{dt} \\
&= \left\langle \nabla l_i(\Theta_t), \frac{d\Theta_t}{dt} \right\rangle = - \langle \nabla l_i(\Theta_t), \nabla \mathcal{L}(\Theta_t) \rangle \\
&= - \left\langle x_i(y_i - \mathbb{E}_{p_{\Theta_t}(\cdot|x_i)}[y])^\top, \frac{1}{2n-1} \left( \sum_{i=1}^n x_i(y_i - \mathbb{E}_{p_{\Theta_t}(\cdot|x_i)}[y])^\top + \sum_{i=2}^n y_i(x_i - \mathbb{E}_{p_{\Theta_t}(\cdot|y_i)}[x])^\top \right) \right\rangle \\
&= - \left\langle \sum_{j=1}^m \alpha_{i,j} v_i v_j^\top, \frac{1}{2n-1} \left( \sum_{i=1}^n \sum_{j=1}^m \alpha_{i,j}^{(t)} v_i v_j^\top + \sum_{i=n+2}^{2n} \sum_{j=1}^m \beta_{i,j}^{(t)} v_i v_j^\top \right) \right\rangle,
\end{aligned}$$

and

$$\begin{aligned}
& \frac{dl_i^{\text{rev}}(\Theta_t)}{dt} \\
&= \left\langle \nabla l_i^{\text{rev}}(\Theta_t), \frac{d\Theta_t}{dt} \right\rangle = - \langle \nabla l_i^{\text{rev}}(\Theta_t), \nabla \mathcal{L}(\Theta_t) \rangle \\
&= - \left\langle y_i(x_i - \mathbb{E}_{p_{\Theta_t}(\cdot|y_i)}[x])^\top, \frac{1}{2n-1} \left( \sum_{i=1}^n x_i(y_i - \mathbb{E}_{p_{\Theta_t}(\cdot|x_i)}[y])^\top + \sum_{i=2}^n y_i(x_i - \mathbb{E}_{p_{\Theta_t}(\cdot|y_i)}[x])^\top \right) \right\rangle \\
&= - \left\langle \sum_{j=1}^m \beta_{i+n,j}^{(t)} v_{i+n} v_j^\top, \frac{1}{2n-1} \left( \sum_{i=1}^n \sum_{j=1}^m \alpha_{i,j}^{(t)} v_i v_j^\top + \sum_{i=n+2}^{2n} \sum_{j=1}^m \beta_{i,j}^{(t)} v_i v_j^\top \right) \right\rangle.
\end{aligned}$$

Applying Lemma 11, with probability at least  $1 - \delta/2$ , for any  $t \geq 0$  we have

$$\begin{aligned}
& \left| \frac{d\mathcal{L}(\Theta_t)}{dt} + \frac{1}{(2n-1)^2} \left( \sum_{i=1}^n \sum_{j=1}^m (\alpha_{i,j}^{(t)})^2 + \sum_{i=n+2}^{2n} \sum_{j=1}^m (\beta_{i,j}^{(t)})^2 \right) \right| \\
& \leq \epsilon \cdot \frac{1}{(2n-1)^2} \left( \sum_{i=1}^n \sum_{j=1}^m (\alpha_{i,j}^{(t)})^2 + \sum_{i=n+2}^{2n} \sum_{j=1}^m (\beta_{i,j}^{(t)})^2 \right),
\end{aligned} \tag{A.2}$$

and

$$\begin{aligned}
& \left| \frac{dl_i(\Theta_t)}{dt} + \frac{1}{2n-1} \sum_{j=1}^m (\alpha_{i,j}^{(t)})^2 \right| \\
& \leq \epsilon \cdot \frac{1}{2n-1} \left( \sum_{j=1}^m (\alpha_{i,j}^{(t)})^2 \right)^{1/2} \left( \sum_{i=1}^n \sum_{j=1}^m (\alpha_{i,j}^{(t)})^2 + \sum_{i=n+2}^{2n} \sum_{j=1}^m (\beta_{i,j}^{(t)})^2 \right)^{1/2},
\end{aligned} \tag{A.3}$$

$$\begin{aligned} & \left| \frac{dl_i^{\text{rev}}(\Theta_t)}{dt} + \frac{1}{2n-1} \sum_{j=1}^m (\beta_{i+n,j}^{(t)})^2 \right| \\ & \leq \epsilon \cdot \frac{1}{2n-1} \left( \sum_{j=1}^m (\beta_{i+n,j}^{(t)})^2 \right)^{1/2} \left( \sum_{i=1}^n \sum_{j=1}^m (\alpha_{i,j}^{(t)})^2 + \sum_{i=n+2}^{2n} \sum_{j=1}^m (\beta_{i,j}^{(t)})^2 \right)^{1/2}. \end{aligned}$$

Furthermore, Lemma 10 implies that with probability at least  $1 - \delta/2$

$$\frac{1}{2m} < p_{\Theta_0}(y_i|x_i), p_{\Theta_0}(x_i|y_i) < \frac{2}{m}. \quad (\text{A.4})$$

The following arguments are based on the event that the above inequalities hold.

We first show that

$$\inf \{p_{\Theta_t}(x_i|y_i), p_{\Theta_t}(y_i|x_i) : t \geq 0, i \in [n]\} > \frac{1}{2m}.$$

We prove this by contradiction. Let

$$\tau = \inf \left\{ t \geq 0 : \exists i \in [n], \text{ s.t. } \min\{p_{\Theta_t}(y_i|x_i), p_{\Theta_t}(x_i|y_i)\} \leq \frac{1}{2m} \right\}.$$

By Eq. (A.4), it is obvious that  $\tau > 0$ . Assume without loss of generality that  $p_{\Theta_\tau}(y_1|x_1) \leq \frac{1}{2m}$ . It follows that there exists  $\delta > 0$  such that  $p_{\Theta_t}(y_1|x_1)$  is a decreasing function in  $(\tau - \delta, \tau)$  and  $p_{\Theta_t}(y_1|x_1) = \min \{p_{\Theta_t}(x_i|y_i), p_{\Theta_t}(y_i|x_i) : i \in [n]\}$  for any  $t \in (\tau - \delta, \tau)$ . It follows that for  $t \in (\tau - \delta, \tau)$ ,

$$\begin{aligned} \frac{dl_1(\Theta_t)}{dt} & \leq \frac{1}{2n-1} \left( - \sum_{j=1}^m (\alpha_{1,j}^{(t)})^2 + \epsilon \cdot \left( \sum_{j=1}^m (\alpha_{1,j}^{(t)})^2 \right)^{1/2} \left( \sum_{i=1}^n \sum_{j=1}^m (\alpha_{i,j}^{(t)})^2 + \sum_{i=n+2}^{2n} \sum_{j=1}^m (\beta_{i,j}^{(t)})^2 \right)^{1/2} \right) \\ & \leq \frac{1}{2n-1} \left( - \sum_{j=1}^m (\alpha_{1,j}^{(t)})^2 + \epsilon \cdot \left( \sum_{j=1}^m (\alpha_{1,j}^{(t)})^2 \right)^{1/2} \left( 2 \sum_{i=1}^n (\alpha_{i,i+n}^{(t)})^2 + 2 \sum_{i=n+2}^{2n} (\beta_{i,i-n}^{(t)})^2 \right)^{1/2} \right) \\ & \leq \frac{1}{2n-1} \left( \sum_{j=1}^m (\alpha_{1,j}^{(t)})^2 \right)^{1/2} \cdot \left( - \left( \sum_{j=1}^m (\alpha_{1,j}^{(t)})^2 \right)^{1/2} + \epsilon \cdot \left( 4n(\alpha_{1,n+1}^{(t)})^2 \right)^{1/2} \right) \\ & \leq 0 \end{aligned}$$

where the first inequality is from Eq. (A.3); the second inequality is due to  $\sum_{j \neq i+n} |\alpha_{i,j}^{(t)}| = \alpha_{i,i+n}^{(t)} = 1 - p_{\Theta_t}(y_i|x_i)$ ,  $\sum_{j \neq i} |\beta_{i+n,j}^{(t)}| = \beta_{i+n,i}^{(t)} = 1 - p_{\Theta_t}(x_i|y_i)$  for all  $i \in [n]$ ; the third inequality is because  $p_{\Theta_t}(y_1|x_1) = \min \{p_{\Theta_t}(x_i|y_i), p_{\Theta_t}(y_i|x_i) : i \in [n]\}$ . However,  $p_{\Theta_t}(y_1|x_1)$  is a decreasing function in  $(\tau - \delta, \tau)$ , a contradiction. Therefore, we conclude that

$$\inf \{p_{\Theta_t}(x_i|y_i), p_{\Theta_t}(y_i|x_i) : t \geq 0, i \in [n]\} > \frac{1}{2m}.$$

Now we show

$$\mathcal{L}(\Theta_t) \leq \frac{1}{\frac{t}{8(2n-1)\log^2(2m)} + \frac{1}{\mathcal{L}(\Theta_0)}}, \quad \forall t \geq 0.$$

By Eq. (A.2),

$$\begin{aligned} \frac{d\mathcal{L}(\Theta_t)}{dt} &\leq -\frac{1-\epsilon}{(2n-1)^2} \left( \sum_{i=1}^n \sum_{j=1}^m (\alpha_{i,j}^{(t)})^2 + \sum_{i=n+2}^{2n} \sum_{j=1}^m (\beta_{i,j}^{(t)})^2 \right) \\ &\leq -\frac{1-\epsilon}{(2n-1)^2} \left( \sum_{i=1}^n (1 - p_{\Theta_t}(y_i|x_i))^2 + \sum_{i=2}^n (1 - p_{\Theta_t}(x_i|y_i))^2 \right) \\ &\leq -\frac{1-\epsilon}{(2n-1)^3} \left( \sum_{i=1}^n (1 - p_{\Theta_t}(y_i|x_i)) + \sum_{i=2}^n (1 - p_{\Theta_t}(x_i|y_i)) \right)^2 \\ &\leq -\frac{1-\epsilon}{8(2n-1)\log^2(2m)} \mathcal{L}(\Theta_t)^2 \\ &\leq -\frac{1}{8(2n-1)\log^2(2m)} \mathcal{L}(\Theta_t)^2, \end{aligned}$$

where the second inequality is due to  $\sum_{j \neq i+n} |\alpha_{i,j}^{(t)}| = \alpha_{i,i+n}^{(t)} = 1 - p_{\Theta_t}(y_i|x_i)$ ,  $\sum_{j \neq i} |\beta_{i+n,j}^{(t)}| = \beta_{i+n,i}^{(t)} = 1 - p_{\Theta_t}(x_i|y_i)$  for all  $i \in [n]$ ; the third inequality applies Cauchy-Schwarz inequality; the last inequality uses the fact that  $p_{\Theta_t}(x_i|y_i), p_{\Theta_t}(y_i|x_i) > \frac{1}{2m}$  for any  $t \geq 0, i \in [n]$  and the inequality  $1 - x \geq \frac{\log x}{2\log(1/(2m))}$  for  $x \in (\frac{1}{2m}, 1)$ .

By Lemma 14, we conclude that  $\forall t \geq 0$ ,

$$\mathcal{L}(\Theta_t) \leq \frac{1}{\frac{t}{8(2n-1)\log^2(2m)} + \frac{1}{\mathcal{L}(\Theta_0)}},$$

which completes the proof.  $\square$

**Lemma 8** (Dynamics of the reversal loss). *Let  $x_1, \dots, x_n, y_1, \dots, y_n, \mathcal{L}^{rev}(\Theta)$ , and  $\Theta_t$  ( $t \geq 0$ ) be defined as in Theorem 12. When  $\sigma \leq \frac{1}{100\ln(16n^2/\delta)}$  and  $d \geq 400n^2m^2 \log(8m^2n^2/\delta)/\epsilon^2$ , with probability at least  $1 - \delta$ ,*

$$\mathcal{L}^{rev}(\Theta_t) \geq \mathcal{L}^{rev}(\Theta_0) \cdot \left( 1 + \frac{\mathcal{L}(\Theta_0) \cdot t}{8(2n-1)\log^2(2m)} \right)^{-8\epsilon(2n-1)\log^2(2m)}.$$

*Proof.* Similar to Lemma 7, we assume  $x_1, \dots, x_n = v_1, \dots, v_n$  and  $y_1, \dots, y_n = v_{n+1}, \dots, v_{2n}$

WLOG. Let  $\alpha_{i,j}^{(t)} = -p_{\Theta_t}(v_j|v_i) + \delta_{i,j-n}$ ,  $\beta_{i,j}^{(t)} = -p_{\Theta_t}(v_j|v_i) + \delta_{i-n,j}$ . By Lemma 9,

$$\begin{aligned} & \frac{d\mathcal{L}^{\text{rev}}(\Theta_t)}{dt} \\ &= \left\langle \nabla \mathcal{L}^{\text{rev}}(\Theta_t), \frac{d\Theta_t}{dt} \right\rangle \\ &= - \left\langle y_1(x_1 - \mathbb{E}_{p_{\Theta_t}(\cdot|y_1)}[x])^\top, \frac{1}{2n-1} \left( \sum_{i=1}^n x_i(y_i - \mathbb{E}_{p_{\Theta_t}(\cdot|x_i)}[y])^\top + \sum_{i=2}^n y_i(x_i - \mathbb{E}_{p_{\Theta_t}(\cdot|y_i)}[x])^\top \right) \right\rangle \\ &= - \left\langle \sum_{j=1}^m \beta_{n+1,j}^{(t)} v_{n+1} v_j^\top, \frac{1}{2n-1} \left( \sum_{i=1}^n \sum_{j=1}^m \alpha_{i,j}^{(t)} v_i v_j^\top + \sum_{i=n+2}^{2n} \sum_{j=1}^m \beta_{i,j}^{(t)} v_i v_j^\top \right) \right\rangle. \end{aligned}$$

Applying Lemma 11, with probability at least  $1 - \delta/2$ , for any  $t \geq 0$  we have

$$\left| \frac{d\mathcal{L}^{\text{rev}}(\Theta_t)}{dt} \right| \leq \epsilon \cdot \frac{1}{2n-1} \left( \sum_{j=1}^m (\beta_{n+1,j}^{(t)})^2 \right)^{1/2} \left( \sum_{i=1}^n \sum_{j=1}^m (\alpha_{i,j}^{(t)})^2 + \sum_{i=n+2}^{2n} \sum_{j=1}^m (\beta_{i,j}^{(t)})^2 \right)^{1/2}, \quad (\text{A.5})$$

and

$$\begin{aligned} & \left| \frac{d\mathcal{L}(\Theta_t)}{dt} + \frac{1}{(2n-1)^2} \left( \sum_{i=1}^n \sum_{j=1}^m (\alpha_{i,j}^{(t)})^2 + \sum_{i=n+2}^{2n} \sum_{j=1}^m (\beta_{i,j}^{(t)})^2 \right) \right| \\ & \leq \epsilon \cdot \frac{1}{(2n-1)^2} \left( \sum_{i=1}^n \sum_{j=1}^m (\alpha_{i,j}^{(t)})^2 + \sum_{i=n+2}^{2n} \sum_{j=1}^m (\beta_{i,j}^{(t)})^2 \right), \end{aligned}$$

as well as

$$\begin{aligned} & \left| \frac{dl_i(\Theta_t)}{dt} + \frac{1}{2n-1} \sum_{j=1}^m (\alpha_{i,j}^{(t)})^2 \right| \\ & \leq \epsilon \cdot \frac{1}{2n-1} \left( \sum_{j=1}^m (\alpha_{i,j}^{(t)})^2 \right)^{1/2} \left( \sum_{i=1}^n \sum_{j=1}^m (\alpha_{i,j}^{(t)})^2 + \sum_{i=n+2}^{2n} \sum_{j=1}^m (\beta_{i,j}^{(t)})^2 \right)^{1/2}, \\ & \left| \frac{dl_i^{\text{rev}}(\Theta_t)}{dt} + \frac{1}{2n-1} \sum_{j=1}^m (\beta_{i+n,j}^{(t)})^2 \right| \\ & \leq \epsilon \cdot \frac{1}{2n-1} \left( \sum_{j=1}^m (\beta_{i+n,j}^{(t)})^2 \right)^{1/2} \left( \sum_{i=1}^n \sum_{j=1}^m (\alpha_{i,j}^{(t)})^2 + \sum_{i=n+2}^{2n} \sum_{j=1}^m (\beta_{i,j}^{(t)})^2 \right)^{1/2}. \end{aligned}$$

Furthermore, Lemma 10 implies that with probability at least  $1 - \delta/2$ ,

$$\frac{1}{2m} \leq p_{\Theta_0}(y_i|x_i), p_{\Theta_0}(x_i|y_i) \leq \frac{2}{m}. \quad (\text{A.6})$$

The following arguments are based on the event that the above inequalities hold.

By Eq. (A.5),

$$\frac{d\mathcal{L}^{\text{rev}}(\Theta_t)}{dt} \geq -\epsilon \cdot \frac{1}{2n-1} \left( \underbrace{\sum_{j=1}^m (\beta_{n+1,j}^{(t)})^2}_{A_t} \right)^{1/2} \left( \underbrace{\sum_{i=1}^n \sum_{j=1}^m (\alpha_{i,j}^{(t)})^2 + \sum_{i=n+2}^{2n} \sum_{j=1}^m (\beta_{i,j}^{(t)})^2}_{B_t} \right)^{1/2}.$$

Notice that

$$\begin{aligned} B_t &\leq 2 \sum_{i=1}^n (1 - p_{\Theta_t}(y_i|x_i))^2 + 2 \sum_{i=2}^n (1 - p_{\Theta_t}(x_i|y_i))^2 \\ &\leq 2 \left( \sum_{i=1}^n (1 - p_{\Theta_t}(y_i|x_i)) + \sum_{i=2}^n (1 - p_{\Theta_t}(x_i|y_i)) \right)^2 \\ &\leq 2(2n-1)^2 \mathcal{L}(\Theta_t)^2 \\ &\leq 2(2n-1)^2 \left( \frac{1}{\frac{t}{8(2n-1)\log^2(2m)} + \frac{1}{\mathcal{L}(\Theta_0)}} \right)^2 \end{aligned}$$

where the first inequality uses  $\sum_{j \neq i+n} |\alpha_{i,j}^{(t)}| = \alpha_{i,i+n}^{(t)} = 1 - p_{\Theta_t}(y_i|x_i)$ ,  $\sum_{j \neq i} |\beta_{i+n,j}^{(t)}| = \beta_{i+n,i}^{(t)} = 1 - p_{\Theta_t}(x_i|y_i)$  for all  $i \in [n]$ ; the third inequality uses the fact that  $1 - x \leq -\log x$ ; the last inequality applies Lemma 7.

Similarly,

$$A_t \leq 2(1 - p_{\Theta_t}(x_1|y_1))^2 \leq 2\mathcal{L}^{\text{rev}}(\Theta_t)^2.$$

Combining, we have

$$\begin{aligned} \frac{d\mathcal{L}^{\text{rev}}(\Theta_t)}{dt} &\geq -\epsilon \cdot \frac{1}{2n-1} \cdot 2\mathcal{L}^{\text{rev}}(\Theta_t) \cdot (2n-1) \cdot \frac{1}{\frac{t}{8(2n-1)\log^2(2m)} + \frac{1}{\mathcal{L}(\Theta_0)}} \\ &\geq -8\epsilon(2n-1)\log^2(2m) \cdot \mathcal{L}^{\text{rev}}(\Theta_t) \cdot \frac{1}{t + \frac{8(2n-1)\log^2(2m)}{\mathcal{L}(\Theta_0)}}. \end{aligned}$$

By Lemma 14, we conclude that  $\forall t \geq 0$ ,

$$\mathcal{L}^{\text{rev}}(\Theta_t) \geq \mathcal{L}^{\text{rev}}(\Theta_0) \cdot \left( 1 + \frac{\mathcal{L}(\Theta_0) \cdot t}{8(2n-1)\log^2(2m)} \right)^{-8\epsilon(2n-1)\log^2(2m)},$$

This completes the proof.  $\square$

**Lemma 9** (Gradient of the loss function). *Define*

$$\mathcal{L}(\Theta) = \frac{1}{2n-1} \left( \sum_{i=1}^n -\log p_{\Theta}(y_i|x_i) + \sum_{i=2}^n -\log p_{\Theta}(x_i|y_i) \right),$$

$$\mathcal{L}^{rev}(\Theta) = -\log p_{\Theta}(x_1|y_1).$$

Then we have

$$\nabla \mathcal{L}(\Theta) = -\frac{1}{2n-1} \left( \sum_{i=1}^n x_i(y_i - \mathbb{E}_{p_{\Theta}(\cdot|x_i)}[y])^{\top} + \sum_{i=2}^n y_i(x_i - \mathbb{E}_{p_{\Theta}(\cdot|y_i)}[x])^{\top} \right),$$

$$\nabla \mathcal{L}^{rev}(\Theta) = -y_1(x_1 - \mathbb{E}_{p_{\Theta}(\cdot|y_1)}[x])^{\top}.$$

*Proof.* We have

$$\begin{aligned} & \nabla(-\log p_{\Theta}(y|x)) \\ &= -\frac{\nabla p_{\Theta}(y|x)}{p_{\Theta}(y|x)} \\ &= -\frac{1}{p_{\Theta}(y|x)} \cdot \frac{xy^{\top} \exp(x^{\top} \Theta y) \left( \sum_{y \in \mathcal{V}} \exp(x^{\top} \Theta y) \right) - \exp(x^{\top} \Theta y) \left( \sum_{y \in \mathcal{V}} xy^{\top} \exp(x^{\top} \Theta y) \right)}{\left( \sum_{y \in \mathcal{V}} \exp(x^{\top} \Theta y) \right)^2} \\ &= -\frac{1}{p_{\Theta}(y|x)} \left( p_{\Theta}(y|x) xy^{\top} - p_{\Theta}(y|x) \sum_{y \in \mathcal{V}} p_{\Theta}(y|x) xy^{\top} \right) \\ &= -x \left( y - \sum_{y \in \mathcal{V}} p_{\Theta}(y|x) y \right)^{\top} \\ &= -x \left( y - \mathbb{E}_{y \sim p_{\Theta}(\cdot|x)}[y] \right)^{\top}. \end{aligned}$$

The statements follow immediately.  $\square$

### A.2.2 Initialization

**Lemma 10** (Initial distributions are all close to uniform). *Fix any  $\delta \in (0, 1)$ . Let  $x_1, x_2, \dots, x_n$   $\stackrel{i.i.d.}{\sim} \mathcal{N}_d(\mathbf{0}_d, \frac{1}{d}I_d)$ . Let  $\Theta \in \mathbb{R}^{d \times d}$ , where each  $\Theta_{ij} \stackrel{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$  independent of  $x_1, \dots, x_n$ . For any  $i, j \in [n]$ , define*

$$p_{\Theta}(x_j|x_i) = \frac{\exp(l_{\Theta}(x_j|x_i))}{\sum_{k=1}^n \exp(l_{\Theta}(x_k|x_i))}, \quad \text{where } l_{\Theta}(x_j|x_i) = x_i^{\top} \Theta x_j.$$

Then when  $\sigma^2 \leq \frac{1}{100 \ln(4n^2/\delta)}$  and  $d \geq 400 \log(1/(2\delta n^2))/\epsilon^2$ , with probability at least  $1 - \delta$ ,

$$|p_{\Theta}(x_j|x_i) - 1/n| \leq \frac{1}{2n}, \quad \forall i, j \in [n].$$

*Proof.* Let  $v = 0.2$ . By Lemma 12, with probability at least  $1 - \delta$ , we have

$$|\langle x_i, x_j \rangle - \delta_{ij}| \leq v, \quad \forall i, j \in [n].$$

Conditioned on the above high-probability event, we can further obtain that for any  $j \in [n]$

$$p_{\Theta}(x_j|x_i) = \frac{\exp(l_{\Theta}(x_j|x_i))}{\sum_{k=1}^n \exp(l_{\Theta}(x_k|x_i))} \leq \frac{\exp(v)}{\sum_{k=1}^n \exp(-v)} = \frac{\exp(2v)}{n},$$

and

$$p_{\Theta}(x_j|x_i) = \frac{\exp(l_{\Theta}(x_j|x_i))}{\sum_{k=1}^n \exp(l_{\Theta}(x_k|x_i))} \geq \frac{\exp(-v)}{\sum_{k=1}^n \exp(v)} = \frac{\exp(-2v)}{n},$$

It follows that

$$\frac{1}{2n} < p_{\Theta}(x_j|x_i) < \frac{3}{2n} \implies |p_{\Theta}(x_j|x_i) - 1/n| < \frac{1}{2n}.$$

which completes the proof.  $\square$

### A.2.3 Subspace embedding

**Lemma 11** ( $\ell_1$ -subspace embedding of Gaussian second-order tensors). *Let  $z_1, \dots, z_n$  be independently sampled from  $\mathcal{N}_d(\mathbf{0}_d, \frac{1}{d}I_d)$ . Let  $\mathcal{I}_1, \mathcal{I}_2 \subseteq [n] \times [n]$  be two index sets. Let  $\mathcal{I}_0 = \mathcal{I}_1 \cap \mathcal{I}_2$ . If  $d \geq 64 \log(2n^2/\delta)/\epsilon^2$ , then with probability at least  $1 - \delta$ ,*

$$\left| \left\langle \sum_{(i,j) \in \mathcal{I}_1} \alpha_{i,j} z_i z_j^{\top}, \sum_{(i,j) \in \mathcal{I}_2} \beta_{i,j} z_i z_j^{\top} \right\rangle - \sum_{(i,j) \in \mathcal{I}_0} \alpha_{i,j} \beta_{i,j} \right| \leq \epsilon \cdot \left( \sum_{(i,j) \in \mathcal{I}_1} |\alpha_{i,j}| \right) \left( \sum_{(i,j) \in \mathcal{I}_2} |\beta_{i,j}| \right)$$

holds for any  $\alpha_{i,j}, \beta_{i,j}$ . Furthermore, if  $d \geq 64k^2 \log(2n^2/\delta)/\epsilon^2$ , then with probability at least  $1 - \delta$ ,

$$\left| \left\langle \sum_{(i,j) \in \mathcal{I}_1} \alpha_{i,j} z_i z_j^{\top}, \sum_{(i,j) \in \mathcal{I}_2} \beta_{i,j} z_i z_j^{\top} \right\rangle - \sum_{(i,j) \in \mathcal{I}_0} \alpha_{i,j} \beta_{i,j} \right| \leq \epsilon \cdot \left( \sum_{(i,j) \in \mathcal{I}_1} \alpha_{i,j}^2 \right)^{1/2} \left( \sum_{(i,j) \in \mathcal{I}_2} \beta_{i,j}^2 \right)^{1/2}$$

holds for any  $\alpha_{i,j}, \beta_{i,j}$  such that  $\|\alpha\|_0 \leq k, \|\beta\|_0 \leq k$ .

*Proof.* Using Lemma 12 and Cauchy-Schwarz inequality, we have

$$\begin{aligned}
& \left| \left\langle \sum_{(i,j) \in \mathcal{I}_1} \alpha_{i,j} z_i z_j^\top, \sum_{(i,j) \in \mathcal{I}_2} \beta_{i,j} z_i z_j^\top \right\rangle - \sum_{(i,j) \in \mathcal{I}_0} \alpha_{i,j} \beta_{i,j} \right| \\
& \leq \left| \sum_{(i,j) \in \mathcal{I}_0} \alpha_{i,j} \beta_{i,j} (\|z_i\|^2 \|z_j\|^2 - 1) + \sum_{(i,j) \in \mathcal{I}_1, (k,l) \in \mathcal{I}_2, (i,j) \neq (k,l)} \alpha_{i,j} \beta_{k,l} z_i^\top z_k z_j^\top z_l \right| \\
& \leq \sqrt{\frac{64 \log(2n^2/\delta)}{d}} \cdot \left( \sum_{(i,j) \in \mathcal{I}} |\alpha_{i,j} \beta_{i,j}| + \sum_{(i,j) \in \mathcal{I}_1, (k,l) \in \mathcal{I}_2, (i,j) \neq (k,l)} |\alpha_{i,j} \beta_{k,l}| \right) \\
& = \sqrt{\frac{64 \log(2n^2/\delta)}{d}} \cdot \left( \sum_{(i,j) \in \mathcal{I}_1} |\alpha_{i,j}| \right) \cdot \left( \sum_{(i,j) \in \mathcal{I}_2} |\beta_{i,j}| \right) \\
& \leq \sqrt{\frac{64k^2 \log(2n^2/\delta)}{d}} \cdot \left( \sum_{(i,j) \in \mathcal{I}_2} \alpha_{i,j}^2 \right)^{1/2} \left( \sum_{(i,j) \in \mathcal{I}_2} \beta_{i,j}^2 \right)^{1/2}.
\end{aligned}$$

The statements follow directly from plugging in suitable values of  $d$ .  $\square$

**Lemma 12** (Almost orthonormal). *Let  $x_1, x_2, \dots, x_n \stackrel{i.i.d.}{\sim} \mathcal{N}_d(\mathbf{0}_d, \frac{1}{d} I_d)$ . For any  $\epsilon, \delta \in (0, 1)$ , when  $d \geq 16 \log(2n^2/\delta)/\epsilon^2$ , it holds that with probability at least  $1 - \delta$ ,*

$$|\langle x_i, x_j \rangle - \delta_{ij}| \leq \epsilon, \quad \forall i, j \in [n].$$

*Proof.* Fix  $i \neq j \in [n]$ . Notice

$$\begin{aligned}
|\langle x_i, x_j \rangle| &= |\langle x_i + x_j, x_i + x_j \rangle - \langle x_i - x_j, x_i - x_j \rangle|/4 \\
&\leq (|\langle x_i + x_j, x_i + x_j \rangle - 1| + |\langle x_i - x_j, x_i - x_j \rangle - 1|)/4.
\end{aligned}$$

Using Lemma 13, we have that with probability at least  $1 - \delta/n^2$ ,

$$|\langle x_i, x_j \rangle| \leq \epsilon.$$

Furthermore, fix  $i \in [n]$ , with probability at least  $1 - \delta/n^2$ ,

$$|\langle x_i, x_i \rangle - 1| \leq \epsilon.$$

The statement then follows from union bound over  $i, j \in [n]$ .  $\square$

**Lemma 13** (Almost normal for a fixed vector). *For a  $d$ -dimensional random vector  $x \sim \mathcal{N}_d(\mathbf{0}_d, \frac{1}{d} I_d)$  and any  $v \in (0, 1/2)$ ,*

$$\mathbb{P}(|\langle x, x \rangle - 1| \geq v) \leq 2e^{-v^2 d/16}.$$

*In particular, when  $d \geq 16 \log(1/(2\delta))/v^2$ , we have*

$$\mathbb{P}(|\langle x, x \rangle - 1| \geq v) \leq \delta$$

*Proof.* Let  $x = (x_1, \dots, x_d)$ . By Lemma 15 (letting  $x = v^2 d/16$ ),

$$\begin{aligned} \mathbb{P}(|\langle x, x \rangle - 1| \geq v) &\leq \mathbb{P}\left(\left|\sum_{i=1}^d (\sqrt{d} \cdot x_i)^2 - d\right| \geq vd\right) \\ &\leq 2e^{-v^2 d/16}. \end{aligned}$$

The second inequality follows from simple arithmetics.  $\square$

### A.2.4 Useful results

**Lemma 14** (ODE bound). *Let  $c_1, c_2, c_3 > 0$ . Suppose the function  $f_1, f_2 : \mathbb{R}_+ \rightarrow \mathbb{R}$  satisfies  $f_1(0) > 0, f_2(0) > 0$  and*

$$\begin{aligned} \frac{df_1(t)}{dt} &\leq -c_1 \cdot f_1(t)^2, \\ \frac{df_2(t)}{dt} &\geq -c_2 \cdot f_2(t) \cdot \frac{1}{t + c_3}. \end{aligned}$$

*Then*

$$\begin{aligned} f_1(t) &\leq \frac{1}{c_1 t + \frac{1}{f_1(0)}}, \\ f_2(t) &\geq f_2(0) \cdot \left(1 + \frac{t}{c_3}\right)^{-c_2}. \end{aligned}$$

*Proof.* The conditions imply that

$$\begin{aligned} \frac{df_1^{-1}(t)}{dt} &= -\frac{1}{f_1^2(t)} \cdot \frac{df_1(t)}{dt} \geq c_1, \\ \frac{d \log f_2(t)}{dt} &= \frac{1}{f_2(t)} \cdot \frac{df_2(t)}{dt} \geq -c_2/(t + c_3). \end{aligned}$$

It follows that

$$\begin{aligned} f_1^{-1}(t) &\geq c_1 t + f_1^{-1}(0), \\ \log f_2(t) &\geq -c_2 \log(1 + t/c_3) + \log f_2(0). \end{aligned}$$

Rearranging the above inequalities, one can obtain the desired results.  $\square$

**Lemma 15** ( $\chi^2$ -concentration bound, Lemma 1 of [75]). *Let  $g_1, \dots, g_t$  be i.i.d.  $\mathcal{N}(0, 1)$  random variables. Then for any  $x \geq 0$ ,*

$$\Pr \left[ \sum_{i=1}^t g_i^2 \geq t + 2\sqrt{tx} + 2x \right] \leq \exp(-x),$$

and

$$\Pr \left[ \sum_{i=1}^t g_i^2 \leq t - 2\sqrt{tx} \right] \leq \exp(-x).$$

### A.3 Missing Proofs of Section 2.4

In this section, we show missing proofs in Section 2.4.

#### A.3.1 Proofs of Section 2.4.1

##### A.3.1.1 Proofs of Proposition 1 and Proposition 2

We first show the proofs of Proposition 1 and Proposition 2, respectively.

*Proof of Proposition 1.* Actually, for any three tokens  $x_1, x_2, x_3$ , it holds that

$$p_{\theta(0)}(x_3|x_1, x_2) = \frac{\exp(\mathbf{x}_3^\top Y(0)^\top \text{LN}(X^\top \mathbf{v}_2))}{\sum_{x' \in [M]} \exp(\mathbf{x}'^\top Y(0)^\top \text{LN}(X^\top \mathbf{v}_2))} = \frac{\exp(0)}{\sum_{x' \in [M]} \exp(0)} = 1/M,$$

since  $Y(0) = \mathbf{0}$ . □

*Proof of Proposition 2.* Note that the input sequence length  $T = 2$ . By (2.1),

$$p_{\theta(t)}(x|x_1, x_2) = \frac{\exp(\mathbf{x}^\top Y(t)^\top \text{LN}(X^\top \mathbf{v}_2))}{\sum_{x' \in [M]} \exp(\mathbf{x}'^\top Y(t)^\top \text{LN}(X^\top \mathbf{v}_2))}$$

where  $\mathbf{v}_2 = [b_{12}]$  and  $b_{12} = 1$ . Also,  $X = [\mathbf{x}_1]^\top$  is a one-hot row vector. Therefore,  $\text{LN}(X^\top \mathbf{v}_2) = \text{LN}(\mathbf{x}_1 b_{12}) = \text{LN}(\mathbf{x}_1) = \mathbf{x}_1$ , and thus

$$p_{\theta(t)}(x|x_1, x_2) = \frac{\exp(\mathbf{x}^\top Y(t)^\top \mathbf{x}_1)}{\sum_{x' \in [M]} \exp(\mathbf{x}'^\top Y(t)^\top \mathbf{x}_1)} = \frac{\exp(Y(t)_{x_1, x})}{\sum_{x' \in [M]} \exp(Y(t)_{x_1, x'})}$$

where  $Y(t)_{i,j}$  is the entry of the matrix  $Y(t)$  at row  $i$  and column  $j$ . □

##### A.3.1.2 Proof of Lemma 2

*Proof of Lemma 2.* We first calculate the gradient of  $Y$  when the current batch is a sequence  $(x_1, x_2, x_3)$ . Note that the input sequence length  $T = 2$  and by the proof of Proposition 2, we have  $\text{LN}(X^\top \mathbf{v}_T) = \text{LN}(X^\top \mathbf{v}_2) = \mathbf{x}_1$ . By Lemma 1, we have

$$\dot{Y} = \eta_Y \text{LN}(X^\top \mathbf{v}_T)(\mathbf{x}_{T+1} - \boldsymbol{\alpha})^\top = \eta_Y \mathbf{x}_1(\mathbf{x}_3 - \boldsymbol{\alpha})^\top$$

where  $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_M]^\top \in \mathbb{R}^M$  with  $\boldsymbol{\alpha} = \exp(Y_{x_1}^\top) / \mathbf{1}^\top \exp(Y_{x_1}^\top)$ . Note that  $\mathbf{x}_1(\mathbf{x}_3 - \boldsymbol{\alpha})^\top$  is a matrix with only  $x_1$ -th row non-zero since  $\mathbf{x}_1$  is one-hot. Therefore, the update of each row of  $Y$  are independent and only  $x_1$ -th row of  $Y$  gets updated at the current time step.

Now we consider any fixed  $x_1 \in \mathcal{V}$ . Let  $t_{x_1,i}$  be the time step that the  $x_1$ -th row of  $Y$  gets updated (i.e., the first token of the training data is  $x_1$  for the current batch) for the  $i$ -th time and let  $t_{x_1,0} = 0$  for notation convenience, then

$$Y(t_{x_1,i})_{x_1} = Y(t_{x_1,i-1})_{x_1} + \eta_Y(\mathbf{x}_3 - \boldsymbol{\alpha})^\top.$$

For convenience, we denote  $\mathbf{y}(i) = Y(t_{x_1,i})_{x_1}^\top$ , and thus

$$\mathbf{y}(i) = \mathbf{y}(i-1) + \eta_Y(\mathbf{x}_3 - \boldsymbol{\alpha}(i-1)) \quad (\text{A.7})$$

where  $\mathbf{y}(0) = \mathbf{0}$  and  $\boldsymbol{\alpha}(i-1) = \exp(\mathbf{y}(i-1)) / \mathbf{1}^\top \exp(\mathbf{y}(i-1))$ . Note that for a fixed  $x_1$ ,  $x_3$  is also fixed by our construction of the dataset. By Lemma 5 of [133], we can obtain that

$$\mathbf{y}(i) = (M-1)h^*(i)\boldsymbol{\xi}_{x_3},$$

where  $\boldsymbol{\xi}_{x_3} = \frac{M}{M-1}(\mathbf{x}_3 - \frac{1}{M}\mathbf{1})$  and  $h^*(i)$  can be derived recursively as

$$h^*(i) = h^*(i-1) + \frac{\eta_Y}{(M-1) + \exp(Mh^*(i-1))}$$

with  $h^*(0) = 0$ . Combining Lemma 7 and 9 in [133], we have

$$h^*(i) \gtrsim \frac{1}{M} \ln(M\eta_Y i), \quad \forall i \gtrsim \frac{\ln M}{\eta_Y}. \quad (\text{A.8})$$

Note that the update of each row of  $Y$  are independent, the training set has size  $N$ , and the batch size is 1, we have

$$Y(t)_{x_1}^\top = (M-1)h^*([t/N])\boldsymbol{\xi}_{x_3}$$

where the training data at time step  $t$  is  $(x_1, x_2, x_3)$ . Combining (A.8), we can obtain that

$$Y(t)_{x_1,x_3} \gtrsim (M-1) \cdot \frac{M}{M-1} \left(1 - \frac{1}{M}\right) \cdot \frac{1}{M} \ln(M\eta_Y [t/N]) \geq \ln\left(\frac{M\eta_Y t}{N}\right)$$

and

$$Y(t)_{x_1,x} \lesssim (M-1) \cdot \frac{M}{M-1} \left(-\frac{1}{M}\right) \cdot \frac{1}{M} \ln(M\eta_Y [t/N]) \leq -\frac{1}{M} \ln\left(\frac{M\eta_Y t}{N}\right), \quad \forall x \neq x_3.$$

On the other hand, for any sequence  $(x_1, x_2, x_3)$  in the test set, since the  $x_1$ -th row of  $Y$  has never been updated, we have

$$Y(t)_{x_1,x} = Y(0)_{x_1,x} = 0, \quad \forall x \in [M].$$

□

### A.3.1.3 Proof of Theorem 3

*Proof.* We first consider training sequence  $(x_1, x_2, x_3)$  at time  $t$ . By Proposition 2, we have

$$p_{\theta(t)}(x_3|x_1, x_2) = \frac{\exp(Y(t)_{x_1, x_3})}{\sum_{x' \in [M]} \exp(Y(t)_{x_1, x'})}$$

and by Lemma 2, we have

$$Y(t)_{x_1, x_3} \geq c \ln \left( \frac{M\eta_Y t}{N} \right), \quad \text{and} \quad Y(t)_{x_1, x} \leq -\frac{c}{M} \ln \left( \frac{M\eta_Y t}{N} \right), \quad \forall x \neq x_3$$

for some constant  $c > 0$ . Therefore,

$$\begin{aligned} p_{\theta(t)}(x_3|x_1, x_2) &\geq \frac{\exp(c \ln(\frac{M\eta_Y t}{N}))}{\exp(c \ln(\frac{M\eta_Y t}{N})) + (M-1) \ln(-\frac{c}{M} \ln(\frac{M\eta_Y t}{N}))} \\ &\geq \frac{(\frac{M\eta_Y t}{N})^c}{(\frac{M\eta_Y t}{N})^c + (M-1)} \\ &= 1 - \frac{M-1}{(\frac{M\eta_Y t}{N})^c + (M-1)} \\ &\geq 1 - \frac{M-1}{2(\frac{M\eta_Y t}{N})^c} \end{aligned}$$

where the last inequality holds since

$$\ln t \gtrsim \ln(NM/\eta_Y) \implies t \geq \frac{N(M-1)^{1/c}}{M\eta_Y} \implies \left( \frac{M\eta_Y t}{N} \right)^c \geq M-1.$$

Finally, for any sequence  $(x_1, x_2, x_3) \in \mathcal{D}_{\text{test}}$ , since  $Y(t)_{x_1, x} = 0, \forall x \in [M]$  according to Lemma 2, we have

$$p_{\theta(t)}(x_3|x_1, x_2) = \frac{\exp(Y(t)_{x_1, x_3})}{\sum_{x' \in [M]} \exp(Y(t)_{x_1, x'})} = \frac{\exp(0)}{M \cdot \exp(0)} = 1/M,$$

which completes the proof.  $\square$

## A.3.2 Additional details and proof of Section 2.4.2

**Datasets.** Let  $N_{\text{train}} > 0, N_{\text{test}} > 0$  be two positive integers and let  $N_{\text{total}} = N_{\text{train}} + N_{\text{test}}$ . Let  $A_i, B_i, C_i \in \mathcal{V}, \forall i \in [N_{\text{total}}]$  be  $3N_{\text{total}}$  distinct tokens. Let  $\rightarrow, \rightsquigarrow \in \mathcal{V} = [M]$  be two additional different tokens that represent “direct implication” and “indirect implication” respectively. Specifically, we have  $A_i \rightarrow B_i, B_i \rightarrow C_i$  and  $A_i \rightsquigarrow C_i$  for all  $i \in [N_{\text{total}}]$ . For notation convenience, we define the following two index sets

$$\mathcal{I}_{\text{train}} = \{1, 2, \dots, N_{\text{train}}\}, \quad \mathcal{I}_{\text{test}} = \{N_{\text{train}} + 1, \dots, N_{\text{total}}\}.$$

The training set  $\mathcal{D}_{\text{train}}$  consists of all  $\mathbf{A}_i \rightarrow \mathbf{B}_i$ ,  $\mathbf{B}_i \rightarrow \mathbf{C}_i$  and  $\mathbf{A}_i \rightsquigarrow \mathbf{C}_i$  for  $i \in \mathcal{I}_{\text{train}}$ . In addition,  $\mathcal{D}_{\text{train}}$  contains  $\mathbf{A}_i \rightarrow \mathbf{B}_i$  and  $\mathbf{B}_i \rightarrow \mathbf{C}_i$  for  $i \in \mathcal{I}_{\text{test}}$ . For convenience, we let  $N = |\mathcal{D}_{\text{train}}|$  to be the size of the training set. The test set  $\mathcal{D}_{\text{test}}$  consists of  $\mathbf{A}_i \rightsquigarrow \mathbf{C}_i$  for  $i \in \mathcal{I}_{\text{test}}$ . Under our construction of the dataset, the LLM will learn the relationship between  $\mathbf{A}_i$ ,  $\mathbf{B}_i$  and  $\mathbf{C}_i$  for  $i \in \mathcal{I}_{\text{train}}$  in both direct and indirect implication, and learn the relationship between  $\mathbf{A}_i$ ,  $\mathbf{B}_i$  and  $\mathbf{C}_i$  for  $i \in \mathcal{I}_{\text{test}}$  only in direct implication and will be tested for indirect implication.

Similar to the reversal curse in Section 2.4.1, we aim to prove through the training dynamics of one-layer transformers that the test probability remains negligible during training. In particular, we are interested in

$$p_\theta(x_3 = \mathbf{C}_i | x_1 = \mathbf{A}_i, x_2 = \rightsquigarrow), \quad i \in \mathcal{I}_{\text{test}}.$$

We also use  $p_\theta(\mathbf{B}_i | \mathbf{A}_i \rightarrow)$ ,  $p_\theta(\mathbf{C}_i | \mathbf{B}_i \rightarrow)$  and  $p_\theta(\mathbf{C}_i | \mathbf{A}_i \rightsquigarrow)$  to more compactly represent  $p_\theta(x_3 = \mathbf{B}_i | x_1 = \mathbf{A}_i, x_2 = \rightarrow)$ ,  $p_\theta(x_3 = \mathbf{C}_i | x_1 = \mathbf{B}_i, x_2 = \rightarrow)$  and  $p_\theta(x_3 = \mathbf{C}_i | x_1 = \mathbf{A}_i, x_2 = \rightsquigarrow)$ , respectively.

The following theorem shows the importance of the chain-of-thought method:

**Theorem 14** (Importance of chain-of-thought, formal statement of Theorem 4). *Assume we run SGD with batch size 1, and assume  $M \gg 100$  and  $\frac{1}{M^{0.99}} \ll \eta_Y < 1$ . Let  $t \gtrsim \frac{N \ln M}{\eta_Y}$  denote the time step which also satisfies  $\ln t \gtrsim \ln(NM/\eta_Y)$ . For any test index  $i \in \mathcal{I}_{\text{test}}$ , we have*

$$p_{\theta(t)}(\mathbf{B}_i | \mathbf{A}_i \rightarrow) \geq 1 - \frac{M-1}{2 \left(\frac{M\eta_Y t}{N}\right)^c}, \quad p_{\theta(t)}(\mathbf{C}_i | \mathbf{B}_i \rightarrow) \geq 1 - \frac{M-1}{2 \left(\frac{M\eta_Y t}{N}\right)^c}$$

for some constant  $c > 0$  and

$$p_{\theta(t)}(\mathbf{C}_i | \mathbf{A}_i \rightsquigarrow) \leq \frac{1}{M}.$$

*Proof.* Recall that by Proposition 2, we have

$$p_{\theta(t)}(x_3 | x_1, x_2) = \frac{\exp(Y(t)_{x_1, x_3})}{\sum_{x' \in [M]} \exp(Y(t)_{x_1, x'})}$$

and by Lemma 2, we have

$$Y(t)_{\mathbf{A}_i, \mathbf{B}_i} \geq c \ln \left( \frac{M\eta_Y t}{N} \right), \quad \text{and} \quad Y(t)_{\mathbf{A}_i, x} \leq -\frac{c}{M} \ln \left( \frac{M\eta_Y t}{N} \right), \quad \forall x \neq \mathbf{B}_i$$

for some constant  $c > 0$ . Therefore, using the same proof as Theorem 3, we have

$$p_{\theta(t)}(\mathbf{B}_i | \mathbf{A}_i \rightarrow) \geq 1 - \frac{M-1}{2 \left(\frac{M\eta_Y t}{N}\right)^c}.$$

Additionally, according to the proof of Lemma 2,  $Y(t)_{\mathbf{A}_i, x}$  has the same value across all  $x \neq \mathbf{B}_i$ , which implies

$$p_{\theta(t)}(\mathbf{C}_i | \mathbf{A}_i \rightsquigarrow) \leq \frac{1}{M}.$$

Similarly, applying Lemma 2 to  $Y(t)_{\mathbf{B}_i}$ , we can obtain that

$$p_{\theta(t)}(\mathbf{C}_i | \mathbf{B}_i \rightarrow) \geq 1 - \frac{M-1}{2 \left( \frac{M\eta_Y t}{N} \right)^c}.$$

□

### A.3.3 Analysis of the reversal curse for the four-token sequences

In this section, we analyze the reversal curse where data points are four-token sentences “ $\mathbf{A}\mathbf{R}_1\mathbf{R}_2\mathbf{B}$ ” or “ $\mathbf{B}\mathbf{R}_1\mathbf{R}_2\mathbf{A}$ ”. For each sentence,  $\mathbf{A}$  and  $\mathbf{B}$  are two distinct tokens that represent two entities, and  $\mathbf{R}_1, \mathbf{R}_2$  are two special tokens jointly representing a relationship that is inverse to itself (e.g.,  $\mathbf{R}_1\mathbf{R}_2$  represents “is the friend of”, then  $\mathbf{A}\mathbf{R}_1\mathbf{R}_2\mathbf{B}$  means “ $\mathbf{A}$  is the friend of  $\mathbf{B}$ ” and  $\mathbf{B}\mathbf{R}_1\mathbf{R}_2\mathbf{A}$  means “ $\mathbf{B}$  is the friend of  $\mathbf{A}$ ”).

**Datasets.** Let  $N_{\text{train}} > 0$  and  $N_{\text{test}} > 0$  and denote  $N_{\text{total}} = N_{\text{train}} + N_{\text{test}}$ . Let  $\mathbf{A}_i, \mathbf{B}_i \in \mathcal{V}, \forall i \in [N_{\text{total}}]$  be  $K \triangleq 2N_{\text{total}}$  distinct tokens representing distinct entities. WLOG, we assume  $\mathbf{A}_i, \mathbf{B}_i \in [K], \forall i \in [N_{\text{total}}]$ . Let  $\mathbf{R}_1, \mathbf{R}_2 \in \mathcal{V}$  be two additional different tokens that jointly represent a relationship that is inverse to itself. Specifically, we have  $\mathbf{A}_i\mathbf{R}_1\mathbf{R}_2\mathbf{B}_i$  and  $\mathbf{B}_i\mathbf{R}_1\mathbf{R}_2\mathbf{A}_i$  for all  $i \in [N_{\text{total}}]$ . For notation convenience, we define the following two index sets

$$\mathcal{I}_{\text{train}} = [N_{\text{train}}], \quad \mathcal{I}_{\text{test}} = [N_{\text{total}}] \setminus \mathcal{I}_{\text{train}}.$$

The training set  $\mathcal{D}_{\text{train}}$  consists of all  $\mathbf{A}_i\mathbf{R}_1\mathbf{R}_2\mathbf{B}_i$  and  $\mathbf{B}_i\mathbf{R}_1\mathbf{R}_2\mathbf{A}_i$  for  $i \in \mathcal{I}_{\text{train}}$ . In addition,  $\mathcal{D}_{\text{train}}$  contains  $\mathbf{A}_i\mathbf{R}_1\mathbf{R}_2\mathbf{B}_i$  for  $i \in \mathcal{I}_{\text{test}}$ . For convenience, we let  $N = |\mathcal{D}_{\text{train}}|$  to be the size of the training set. The test set  $\mathcal{D}_{\text{test}}$  consists of  $\mathbf{B}_i\mathbf{R}_1\mathbf{R}_2\mathbf{A}_i$  for  $i \in \mathcal{I}_{\text{test}}$ . Under our construction of the dataset, the LLM will learn the relationship between  $\mathbf{A}_i$  and  $\mathbf{B}_i$  for  $i \in \mathcal{I}_{\text{train}}$  in both directions to deduce that the relationship “ $\mathbf{R}_1\mathbf{R}_2$ ” is reverse to itself, and learn the relationship between  $\mathbf{A}_i$  and  $\mathbf{B}_i$  for  $i \in \mathcal{I}_{\text{test}}$  in one direction and will be tested for the other. WLOG, we assume for each training sequence  $(x_1, \mathbf{R}_1, \mathbf{R}_2, x_4) \in \mathcal{D}_{\text{train}}, x_4 \in [N]$ .

Now we assume that the learning rate  $\eta_Y \gg \eta_Z$  and therefore can treat  $X^\top \mathbf{v}_T$  as fixed when analyzing the dynamics of  $Y$ . For any sequence  $(x_1, x_2 = \mathbf{R}_1, x_3 = \mathbf{R}_2, x_4 = n)$  in training or test dataset with  $T = 3$ , we define  $\mathbf{f}_n = \text{LN}(X^\top \mathbf{v}_T)$ . Then the gradient of  $Y$  in (2.3) becomes

$$\dot{Y} = \eta_Y \mathbf{f}_n (\mathbf{x}_{T+1} - \boldsymbol{\alpha})^\top = \eta_Y \mathbf{f}_n (\mathbf{e}_n - \boldsymbol{\alpha})^\top.$$

Note that for three-token sequences,  $\mathbf{f}_n$  is a one-hot vector, and thus  $\dot{Y}$  has only one non-zero row, and each row of  $Y$  can be analyzed independently. For the four-token

sequences, we use the same reparameterization strategy as in [133], where we denote  $W = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_K]^\top \triangleq F^\top Y \in \mathbb{R}^{K \times M}$  with  $F = [\mathbf{f}_1, \dots, \mathbf{f}_K] \in \mathbb{R}^{M \times K}$ .

Note that the parameter  $W$  can be viewed as a combination of  $Y$  and  $Z$  where  $Z$  is fixed. The following lemma shows the dynamics of  $W$ , assuming we are performing gradient updates on  $W$  instead of  $Y$ .

**Lemma 16** (Dynamics of  $W$ ). *Assume we perform gradient updates directly on  $W$  instead of  $Y$  with learning rate  $\eta_Y$  and batch size 1, and assume  $M \gg 100$  and  $\frac{1}{M^{0.99}} \ll \eta_Y < 1$ . Let  $t \gtrsim \frac{N \ln M}{\eta_Y}$  and let  $W(t)_i$  denote the  $i$ -th row of  $W(t)$  and  $W(t)_{ij}$  denote the  $(i, j)$ -th entry of  $W(t)$ . Then for training sequence  $(x_1, \mathbf{R}_1, \mathbf{R}_2, x_4) \in \mathcal{D}_{\text{train}}$  at time  $t$ , we have*

$$W(t)_{x_4, x_4} \gtrsim \ln(M\eta_Y t/N), \quad \text{and} \quad W(t)_{x_4, x} \lesssim -\ln(M\eta_Y t/N)/M, \quad \forall x \neq x_4,$$

and for any test sequence  $(x_1, \mathbf{R}_1, \mathbf{R}_2, x_4) \in \mathcal{D}_{\text{test}}$ , we have  $W(t)_{x_4, x} = 0, \forall x \in [M]$ .

*Proof.* According to Lemma 3 of [133], for training sequence  $(x_1, \mathbf{R}_1, \mathbf{R}_2, x_4) \in \mathcal{D}_{\text{train}}$  at time  $t$ , only the  $x_4$ -th row of  $W$  will be updated and the gradient

$$\dot{\mathbf{w}}_{x_4}(t) = \eta_Y(\mathbf{x}_4 - \boldsymbol{\alpha}_{x_4}(t))$$

where  $\boldsymbol{\alpha}(t) = \exp(\mathbf{w}_{x_4}(t))/(\mathbf{1}^\top \exp(\mathbf{w}_{x_4}(t)))$ . Therefore, the dynamics of  $W$  is nearly identical to the dynamics of  $Y$  in Lemma 2, and we can use the proof of Lemma 2 to conclude the results.  $\square$

With the dynamics of  $W$ , we can obtain the following result:

**Proposition 4** (Reversal curse for the four-token sequences). *Assume we perform gradient updates directly on  $W$  instead of  $Y$  with learning rate  $\eta_Y$  and batch size 1, and assume  $M \gg 100$  and  $\frac{1}{M^{0.99}} \ll \eta_Y < 1$ . Let  $t \gtrsim \frac{N \ln M}{\eta_Y}$  denote the time step which also satisfies  $\ln t \gtrsim \ln(NM/\eta_Y)$ . For training sequence  $(x_1, \mathbf{R}_1, \mathbf{R}_2, x_4) \in \mathcal{D}_{\text{train}}$  at time  $t$ , we have*

$$p_{\theta(t)}(x_4|x_1, \mathbf{R}_1, \mathbf{R}_2) \geq 1 - \frac{M-1}{(M\eta_Y t/N)^c} \rightarrow 1, \quad \text{as } t \rightarrow \infty$$

for some constant  $c > 0$ , and for any test sequence  $(x_1, \mathbf{R}_1, \mathbf{R}_2, x_4) \in \mathcal{D}_{\text{test}}$  that is not included in the training set  $\mathcal{D}_{\text{train}}$ , we have

$$p_{\theta(t)}(x_4|x_1, \mathbf{R}_1, \mathbf{R}_2) \leq 1/M.$$

*Proof.* For any sequence  $(x_1, x_2 = \mathbf{R}_1, x_3 = \mathbf{R}_2, x_4)$  where  $T = 3$ ,

$$\begin{aligned} p_{\theta(t)}(x|x_1, \mathbf{R}_1, \mathbf{R}_2) &= \frac{\exp(\mathbf{x}^\top Y(t)^\top \text{LN}(X^\top \mathbf{v}_T))}{\sum_{x' \in [M]} \exp(\mathbf{x}'^\top Y(t)^\top \text{LN}(X^\top \mathbf{v}_T))} \\ &= \frac{\exp(\mathbf{x}^\top Y(t)^\top \mathbf{f}_{x_4})}{\sum_{x' \in [M]} \exp(\mathbf{x}'^\top Y(t)^\top \mathbf{f}_{x_4})} \\ &= \frac{\exp(\mathbf{x}^\top \mathbf{w}_{x_4}(t))}{\sum_{x' \in [M]} \exp(\mathbf{x}'^\top \mathbf{w}_{x_4}(t))} \\ &= \frac{\exp(W(t)_{x_4, x})}{\sum_{x' \in [M]} \exp(W(t)_{x_4, x'})}. \end{aligned}$$

The above next token probability formulation is almost identical to Proposition 2 after replacing  $Y$  with  $W$ . Combining the dynamics of  $W$  as shown in Lemma 16, we can use the proof of Theorem 3 to conclude the result.  $\square$

Finally, we discuss the role of  $Z$  in the four-token-sequence settings. Note that Proposition 4 assumes gradient update on  $W$  instead of  $Y$ . While we are not able to perform gradient updates on  $W$  directly, it is equivalent to modifying the gradient of  $Y$  to be

$$\dot{Y} = \eta_Y(\mathbf{f}_n - FE'e_n)(e_n - \alpha_n)^\top$$

according to Lemma 3 of [133], where the next token  $x_{T+1} = n$ ,  $E' = (I + E)^{-1} - I$ ,  $E = F^\top F - I$ , and  $F = [\mathbf{f}_1, \dots, \mathbf{f}_N] \in \mathbb{R}^{M \times N}$  which only contains the next token that appears in the training set. Compared to the original gradient of  $Y$

$$\dot{Y} = \eta_Y \mathbf{f}_n (e_n - \alpha_n)^\top,$$

one can obtain that the modification of the gradient of  $Y$  is small if  $\lambda_1(E)$  is small.

Note that  $E_{ii} = \|\mathbf{f}_i\|_2^2 - 1 = 0$  and  $E_{ij} = \mathbf{f}_i^\top \mathbf{f}_j$ . Also, for any sequence  $(x_1 = n', x_2 = \mathbf{R}_1, x_3 = \mathbf{R}_2, x_4 = n)$  in training set,

$$\mathbf{f}_n = \text{LN}(X^\top \mathbf{v}_T) = \frac{b_{13} \mathbf{e}_{n'} + b_{23} \mathbf{e}_{\mathbf{R}_1}}{\sqrt{b_{13}^2 + b_{23}^2}},$$

where

$$b_{13} = \exp(Z_{\mathbf{R}_2, n'}), b_{23} = \exp(Z_{\mathbf{R}_2, \mathbf{R}_1}).$$

Note that  $\mathbf{R}_1$  is a common token that appears in each training sentence, and  $n'$  is a distinct token that only appears in one training sentence as a contextual token. By Theorem 2 of [133], under certain technical assumptions,  $\dot{Z}_{\mathbf{R}_2, n'} > 0$  and  $\dot{Z}_{\mathbf{R}_1, n} < 0$  and one can expect  $Z_{\mathbf{R}_2, n'} - Z_{\mathbf{R}_1, n}$  to be sufficiently large after sufficient time of training. Therefore,

$$\mathbf{f}_n = \tilde{b}_{13} \mathbf{e}_{n'} + \tilde{b}_{23} \mathbf{e}_{\mathbf{R}_1}$$

with  $\tilde{b}_{23}$  close to 0. Consider a simple case where for each  $n \in [N]$ ,  $\mathbf{f}_n = \sqrt{1 - c^2} \mathbf{e}_{n'} + c \mathbf{e}_{\mathbf{R}_1}$  for  $c$  sufficiently small. Then

$$E_{ij} = \mathbf{f}_i^\top \mathbf{f}_j = (\sqrt{1 - c^2} \mathbf{e}_{i'} + c \mathbf{e}_{\mathbf{R}_1})^\top (\sqrt{1 - c^2} \mathbf{e}_{j'} + c \mathbf{e}_{\mathbf{R}_1}) = c^2.$$

Therefore, one can calculate that  $\lambda_1(E) = c^2(N - 1)$ . When  $c \ll \frac{1}{\sqrt{N}}$ , we have  $\lambda_1(E) \ll 1$ , and thus the gradient update of  $W$  and gradient update of  $Y$  are almost the same.

## A.4 Experiments for chain-of-thought

In this section, we conduct experiments for CoT on multi-layer transformers to validate theoretical results in Section 2.4.2.

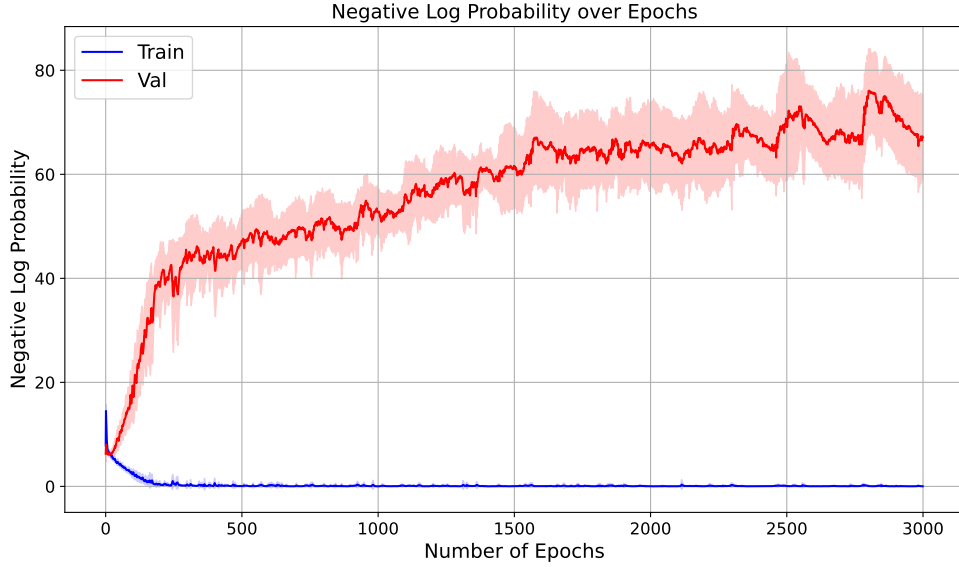


Figure A.1: Experiment results of CoT under default configuration (see Table A.2). The curves represent the (average) negative log probability of the model predicting the next token to be: (1)  $B_i$  given the input “ $A_i \rightarrow$ ”, (2)  $C_i$  given the input “ $B_i \rightarrow$ ”, or (3)  $C_i$  given the input “ $A_i \rightsquigarrow$ ”. Similar to the reversal curse experiment, while the sentences in the training set can be learned nearly perfectly, the model is not able to predict the correct next token in the validation set better than a uniformly random guess. Both curves are averaged over 10 random seeds.

**Dataset construction.** Similar to Section 2.5, we randomly sample three disjoint sets of entities  $\mathcal{A}, \mathcal{B}, \mathcal{C} \subset \mathcal{V}$ , and reverse two additional tokens for  $\rightarrow$  and  $\rightsquigarrow$ , respectively. Next, we specify a bijection from  $\mathcal{A}$  to  $\mathcal{B}$ , and a bijection from  $\mathcal{B}$  to  $\mathcal{C}$  randomly. For each

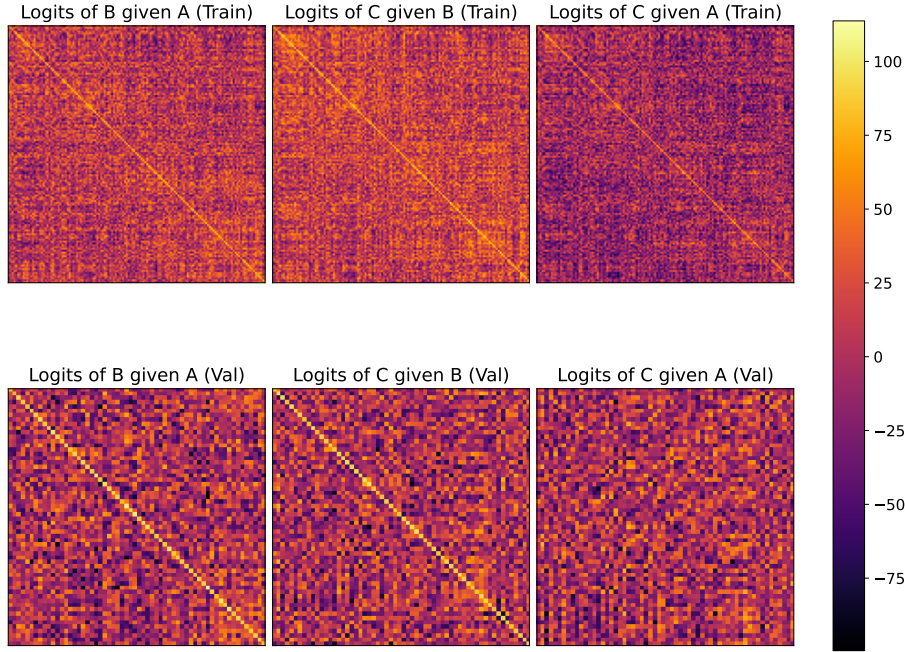


Figure A.2: Visualization of the weights (logits) of the model with default configurations trained after 3000 epochs for CoT experiment. The matrices are similar to Figure 2.2. The row tokens for the top matrices are  $A_i$ ,  $B_i$ ,  $A_i$  and column tokens are  $B_i$ ,  $C_i$ ,  $C_i$  for training triples respectively. Similarly, the bottom matrices correspond to validation triples. For validation triples  $(A_i, B_i, C_i)$ , the weights from  $A_i$  to  $C_i$  get hardly trained as indicated by the diagonals of the last matrix.

$A_i \in \mathcal{A}$  and its corresponding  $B_i \in \mathcal{B}$  and  $C_i \in \mathcal{C}$ , we can obtain a triple of sequences  $(A_i \rightarrow B_i, B_i \rightarrow C_i, A_i \rightsquigarrow C_i)$ , and split the set of all triples into training triples and validation triples. All three sequences of a training triple will be added to the training set, while for a validation triple, we add  $A_i \rightarrow B_i$  and  $B_i \rightarrow C_i$  to the training set and add  $A_i \rightsquigarrow C_i$  to the validation set. Therefore, the model will learn both direct and indirect implications for the training triples and only learn the direct implications for each validation triple while being tested on the indirect implication.

**Results.** Figure A.1 shows the experiment results for CoT using the same model architecture and configurations as in Figure 2.1 (the training set size is 540, and the validation set size is 60 resulting from 140 training triples and 60 validation triples), which is consistent with Theorem 14. One can refer to Section A.5.3 for additional experiments with various model configurations and vocabulary sizes. We also empirically validate the intransitivity of model weights (i.e., logits) for multi-layer transformers in Figure A.2, which shows that for a

validation triple  $(A_i, B_i, C_i)$  of which only the direct implication “ $A_i \rightarrow B_i$ ” and “ $B_i \rightarrow C_i$ ” appears in the training set, although the weights from  $A_i$  to  $B_i$  and from  $B_i$  to  $C_i$  are trained large as indicated by the diagonals of the first two bottom matrices, the weights from  $A_i$  to  $C_i$  gets hardly trained as indicated by the diagonals of the last matrix. We also emphasize that another reason that CoT is necessary is that all tokens  $A_i$ ,  $B_i$ , and  $C_i$  are different tokens with randomly initialized embedding and thus irrelevant. When these tokens are relevant and show specific patterns, the validation loss can also get better. See more details in Section A.5.4.

## A.5 Additional Experimental Results

In this section, we show additional experimental results for Section 2.5.

### A.5.1 Details of model architectures and hyperparameters

For both the reversal curse and CoT experiments, we used the GPT2 model architecture [147]<sup>1</sup> and trained the model with the AdamW optimizer for 3000 epochs of batch size 64. See Table A.1 for a full list of hyperparameters. We also conducted experiments under various model configurations and vocabulary sizes to show that the results in Sections A.4 and 2.5 are consistent under different settings. See Table A.2 for a complete list of different configurations, where the default choices are boldened. For each curve in all figures, the results are averaged over 10 trials, and the error bar is calculated using standard deviation. We run each trial on an Nvidia A100 GPU and it typically takes 0.5-1.5 hours for each trial.

| Parameters             | Values       |
|------------------------|--------------|
| Learning Rate          | 0.01         |
| Weight Decay $\lambda$ | 0.9          |
| $(\beta_1, \beta_2)$   | (0.9, 0.999) |
| Batch Size             | 64           |
| Number of Epochs       | 3000         |

Table A.1: Full list of hyperparameters for AdamW optimizer and training.

### A.5.2 Additional experimental results for the reversal curse

In this section, we show additional experimental results for the reversal curse under different configurations, including different vocabulary sizes (Figure A.3), different number of layers (Figure A.4), different positional encoding (Figure A.5), different entity lengths (Figure A.6) and whether token and positional embeddings are trainable or fixed (Figure A.7). Our experimental results consistently show that the reversal curse happens under different settings.

| Parameters                  | Values                           |
|-----------------------------|----------------------------------|
| Number of Layers            | 12, <b>24</b> , 48               |
| Number of Heads             | <b>12</b>                        |
| Vocabulary Size             | 20, 50, 200, <b>800</b> , 2000   |
| Entity Length               | <b>1</b> , 2, 3                  |
| Positional Encoding Type    | None, <b>Absolute</b> , Relative |
| Token, Positional Embedding | <b>Learnable</b> , Frozen        |

Table A.2: The list of different configurations for experiments in Sections A.5.2 and A.5.3. Default choices are bolded for each row.

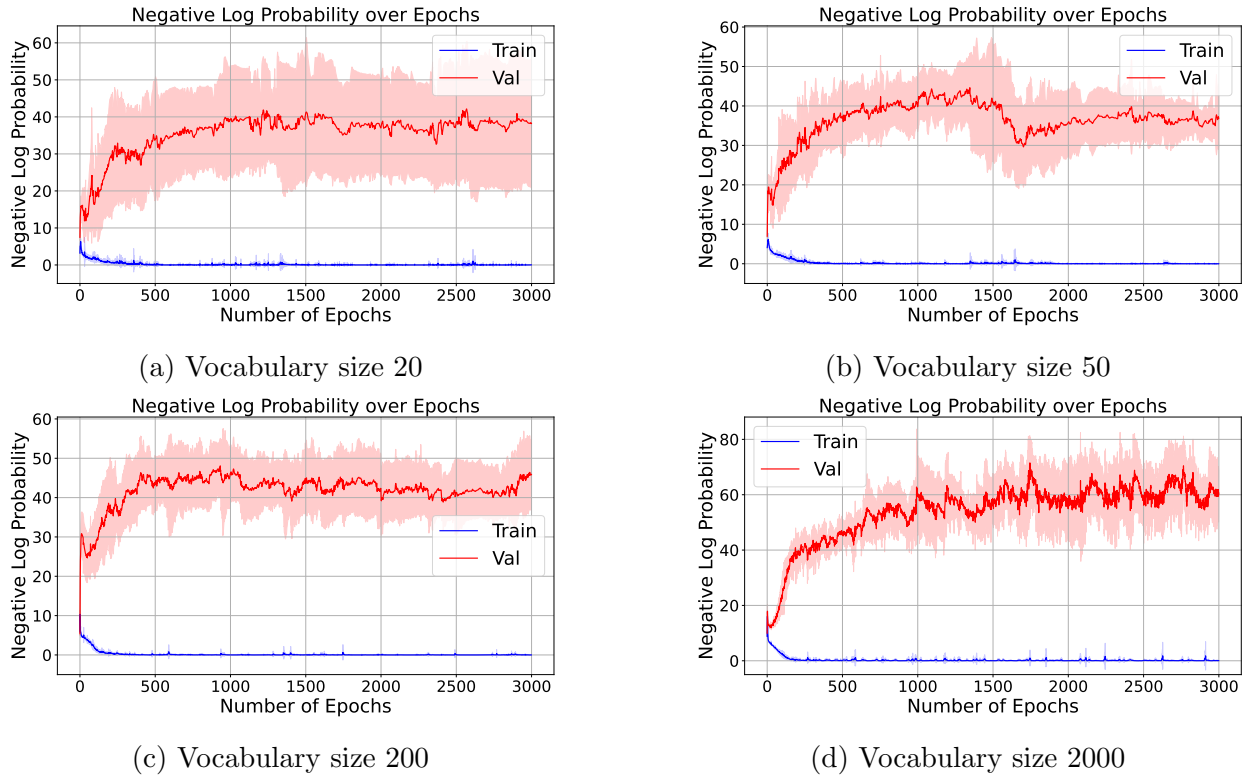
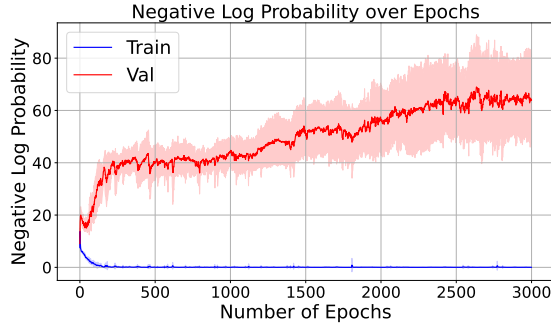
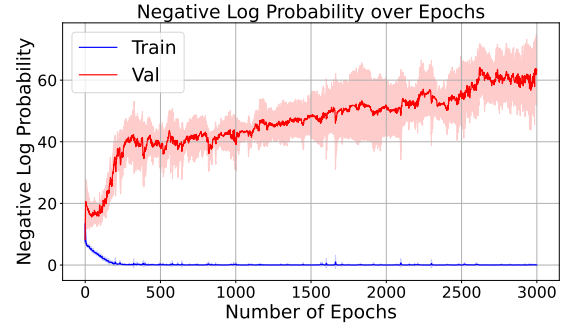


Figure A.3: Results for reversal curse for different vocabulary sizes. All other configurations are set as default values as in Table A.2. The training set sizes for the above four experiments are 9, 20, 85, 850 respectively, and the validation set sizes are 1, 4, 15, 150 respectively.

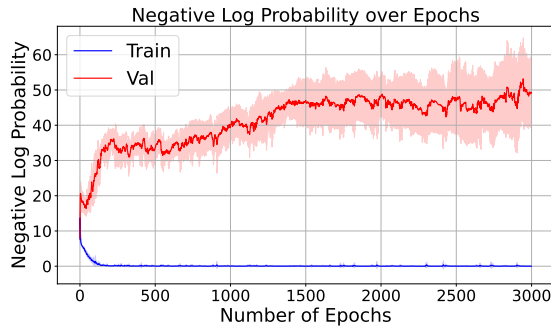


(a) 12 layers

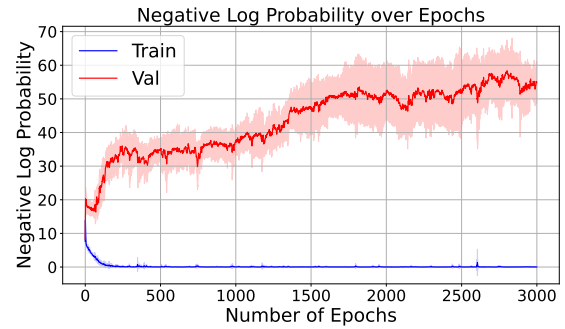


(b) 48 layers

Figure A.4: Results for reversal curse for different numbers of layers of the transformer. All other configurations are set as default values as in Table A.2.

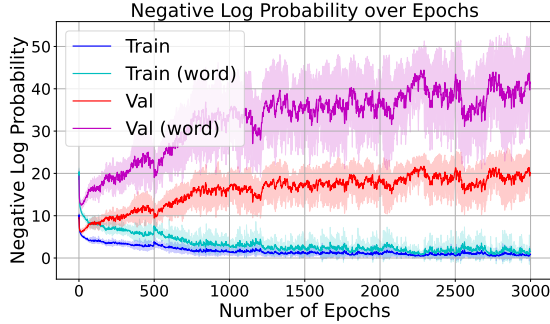


(a) No positional encoding

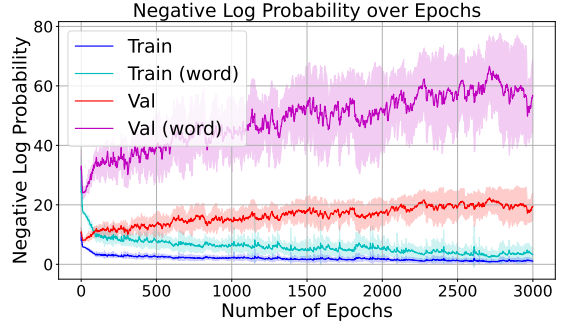


(b) Relative positional encoding

Figure A.5: Results for reversal curse with no positional encoding or relative positional encoding. For relative positional encoding, we follow the Rotary Position Embedding (RoPE) method proposed by [123]. We use the implementation of [this repo](#), MIT license. All other configurations are set as default values as in Table A.2.



(a) Entity length 2



(b) Entity length 3

Figure A.6: Results for reversal curse with different entity lengths. Each entity  $A_i$  or  $B_i$  consists of multiple tokens and different entities may have overlapped tokens. The “Train” curve represents the negative log probability of predicting the first token of the output entity, and “Train (word)” represents the negative log probability of predicting all tokens one by one of the output entity. All other configurations are set as default values as in Table A.2. The training set sizes for the above two experiments are 680 and 250, respectively, and the validation set sizes are 120 and 50, respectively.

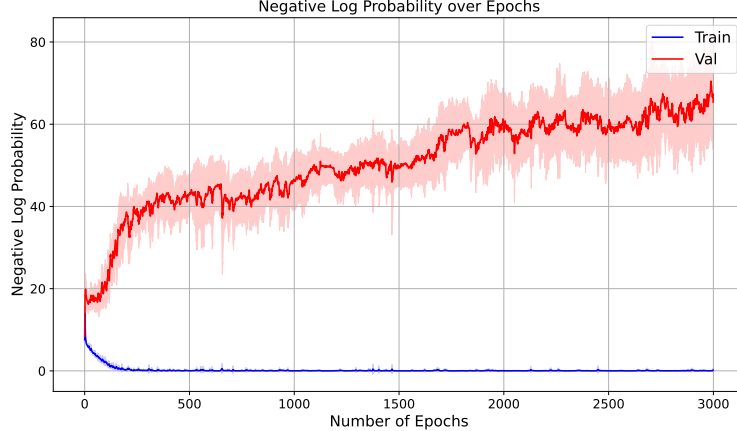


Figure A.7: Results for reversal curse with fixed token embedding and fixed positional embedding. All other configurations are set as default values as in Table A.2.

We provide additional experimental results that (1) the reversal curse still happens even if the embedding dimension is much smaller than the vocabulary size in Section A.5.2.1; (2) the embeddings of different tokens are nearly orthogonal; (3) the reversal curse does not happen under the in-context learning settings.

---

<sup>1</sup>Apache License 2.0

### A.5.2.1 The reversal curse under small embedding dimensions

Although for theoretical analysis in Section 2.3, we assumed the embedding dimension is polynomial in the vocabulary size, in practice, the embedding dimension only needs to be the order of logarithm of the vocabulary size. Figure A.8 shows that for a much smaller embedding size, the reversal curse still happens.

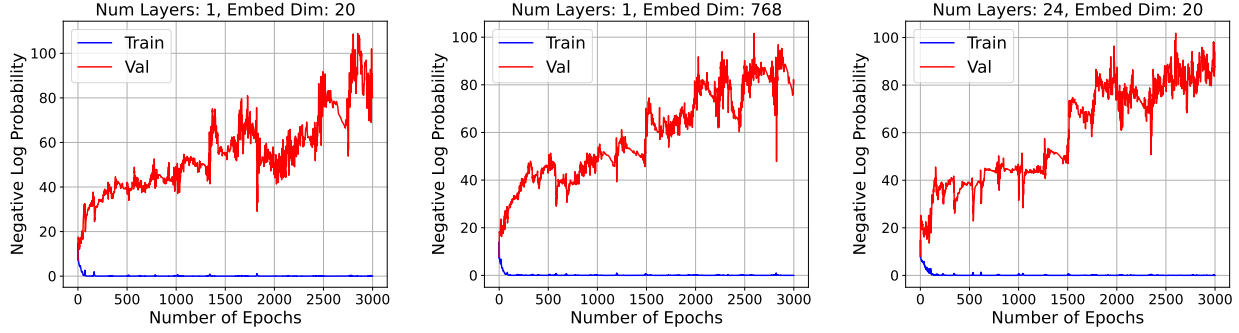


Figure A.8: Results for reversal curse for different numbers of layers and different embedding dimensions. All other configurations are set as default values as in Table A.2.

### A.5.2.2 Near orthogonal embeddings

Note that in Section 2.3, our analysis relies on the fact that embeddings of different tokens are nearly orthogonal, and in Section 2.4, the embeddings are effectively one-hot. In Figure A.9, We show that in practice, even if the embedding dimension is much smaller than the vocabulary size, the near orthogonality condition still holds.

### A.5.2.3 The reversal curse does not happen in ICL settings

We also emphasize that the reversal curse does not happen in ICL settings, which means if “ $A \rightarrow B$ ” is provided as part of the prompt, then the model is able to answer “ $B \leftarrow A$ ”. Figure A.10 shows preliminary results of ICL. All the sentences in the dataset have the format of “ $A_i R B_j \Leftrightarrow B_j R^{-1} A_i$ ”, which is a seven-token sentence where  $R$  and  $R^{-1}$  is a pair of relationships inverse to each other, and “ $\Leftrightarrow$ ” is another reserved token representing equivalence. There are ten different  $B_j$  and  $n$  different  $A_i$  where  $n = 100$  for the left figure in Figure A.10 and  $n = 200$  for the right figure. For each  $A_i$ , we construct ten sentences using different  $B_j$ , and we randomly chose three of them to be included in the validation set and seven other sentences in the training set. The result of Figure A.10 shows that the reversal curse does not happen during the ICL setting.

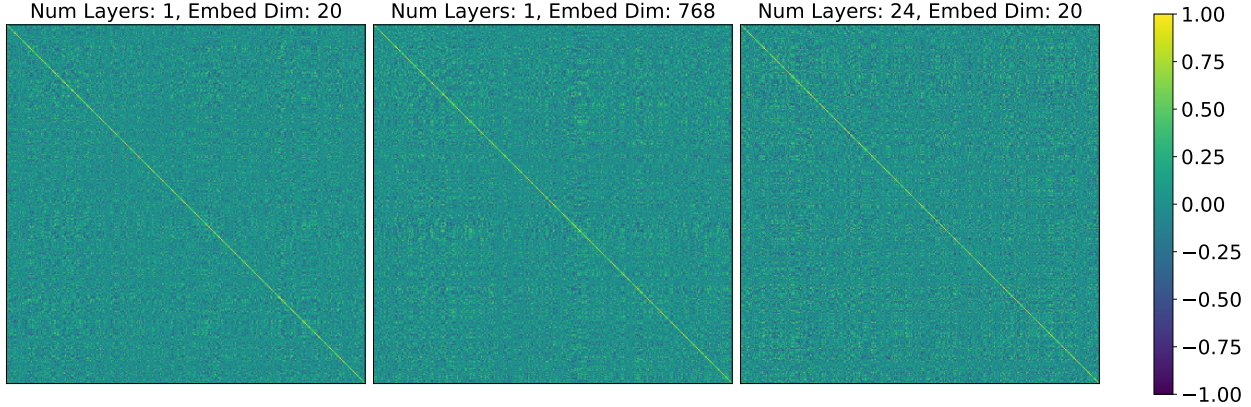


Figure A.9: A heat map of cosine similarity between token embeddings (all  $A_i$  and  $B_i$ ) after 3000 epochs training under different settings. The settings are the same as Figure A.8. Under different numbers of layers or embedding dimensions, most of the non-diagonal entries are close to 0, which shows that the embeddings of different tokens are nearly orthogonal.

### A.5.3 Additional experimental results for chain-of-thought

In this section, we show additional experimental results for CoT under different configurations, including different vocabulary sizes (Figure A.11), different number of layers (Figure A.12), different positional encoding (Figure A.13), different entity lengths (Figure A.14) and whether token and positional embeddings are trainable or fixed (Figure A.15). Note that our experimental results consistently show the necessity of CoT under different settings.

### A.5.4 Chain-of-thought with relevant tokens

In Section A.4, we briefly mentioned that the irrelevance of different entity tokens is one of the reasons that CoT is necessary. Now, we show that if the entity tokens are correlated and show specific patterns, it is possible for a model to deduce indirect implications automatically.

Instead of using single tokens  $A_i$ ,  $B_i$ ,  $C_i$  to represent each entity, now we use two tokens  $A_i$ ,  $B_i$ ,  $C_i$  to represent entities, where  $A$ ,  $B$  and  $C$  are three common tokens shared by each triple, and token  $i$  are distinct for each triple. Figure A.16 shows that for the above version of CoT where tokens in the same chain are correlated, the model is able to “deduce”  $A_i \rightsquigarrow C_i$  after training on  $A_i \rightarrow B_i$ ,  $B_i \rightarrow C_i$  and other training samples.

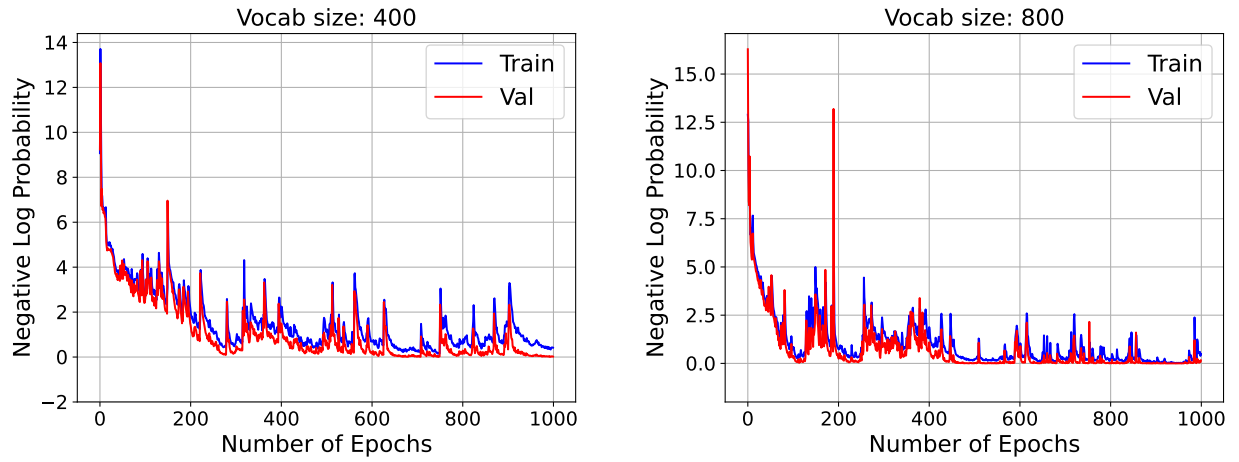


Figure A.10: Training and validation loss under in-context learning (ICL) settings. All sentences consist of seven tokens and have the form of “ $A_i R B_j \Leftrightarrow B_j R^{-1} A_i$ ”. The loss is calculated on the last token. For the left figure, the training set size is 700, the validation set size is 300, and the vocabulary size is 400; for the right figure, the training set size is 1400, the validation set size is 600, and the vocabulary size is 800. All other configurations are set as default values as in Table A.2. The result shows that the reversal curse does not happen in ICL settings, i.e., if “ $A_i R B_j$ ” is provided as part of the prompt, then the model is able to recognize “ $B_j R^{-1} A_i$ ”.

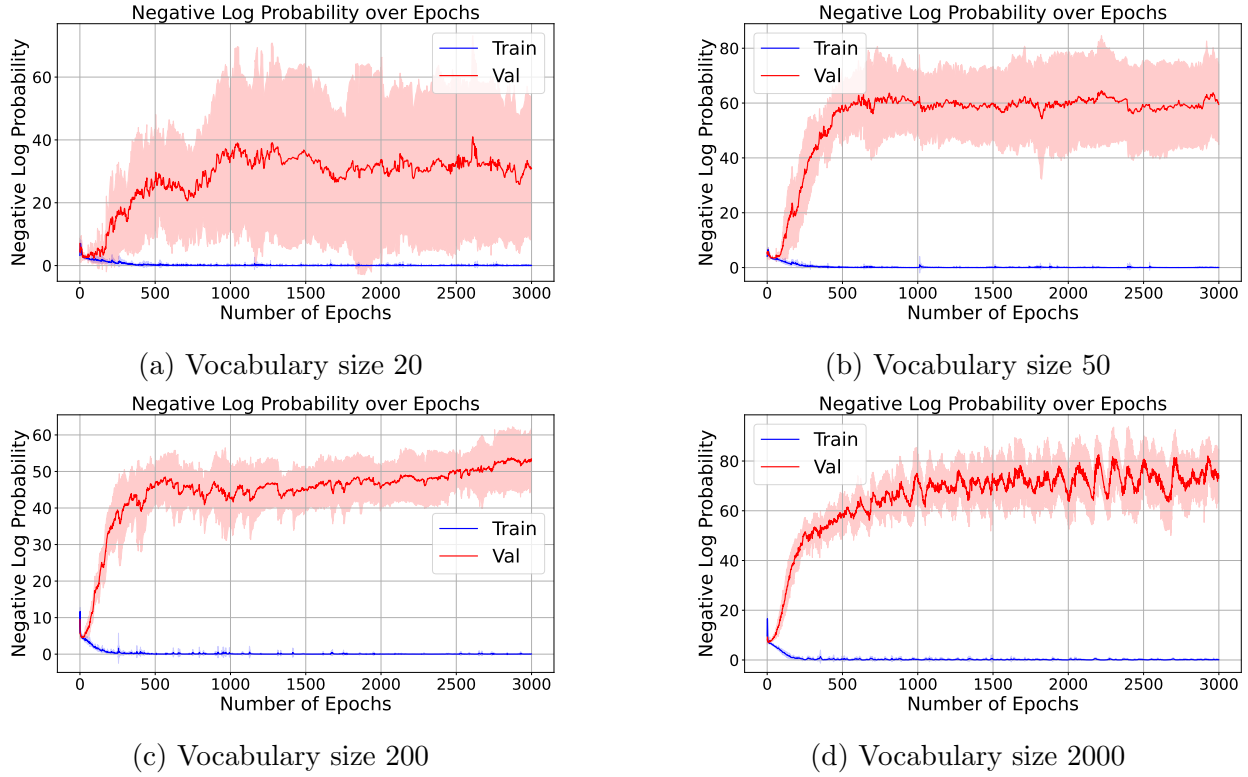


Figure A.11: Results for CoT for different vocabulary sizes. All other configurations are set as default values as in Table A.2. The training set sizes for the above four experiments are 14, 32, 135, 1350 respectively, and the validation set sizes are 1, 4, 15, 150 respectively.

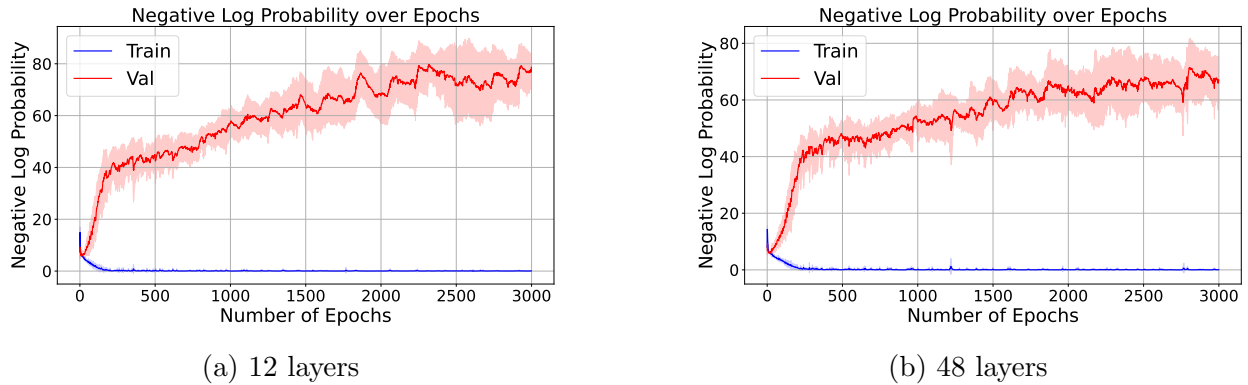


Figure A.12: Results for CoT for different number of layers of the transformer. All other configurations are set as default values as in Table A.2.

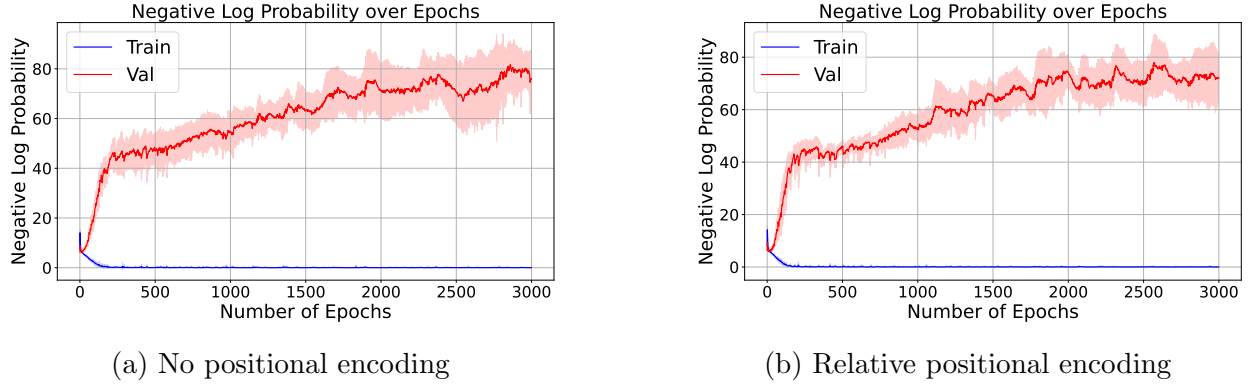


Figure A.13: Results for CoT with no positional encoding or relative positional encoding. For relative positional encoding, we follow the Rotary Position Embedding (RoPE) method proposed by [123]. All other configurations are set as default values as in Table A.2.

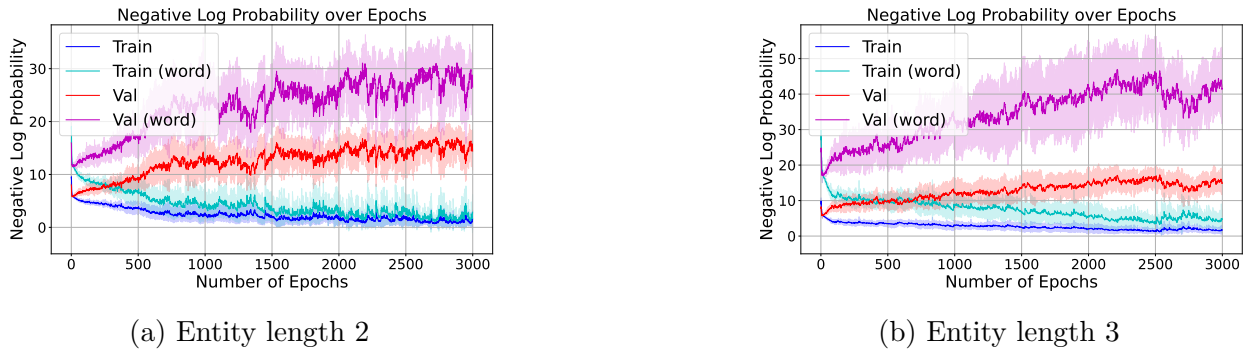


Figure A.14: Results for CoT with different entity lengths. The setting and curves are similar to Figure A.6. All other configurations are set as default values as in Table A.2. The training set sizes for the above two experiments are 1080 and 400, respectively, and the validation set sizes are 120 and 50.

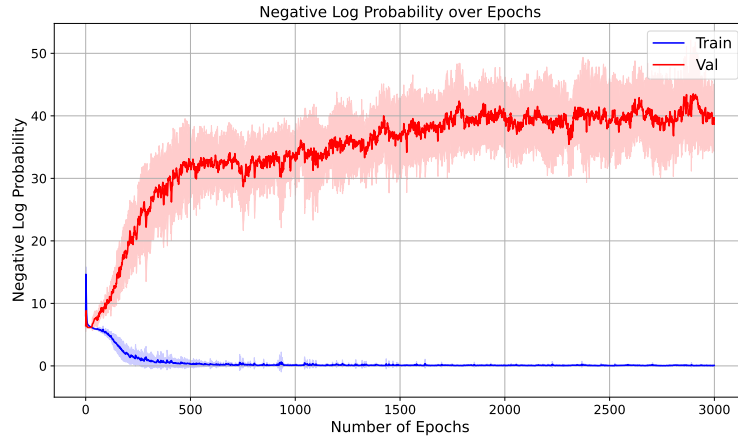
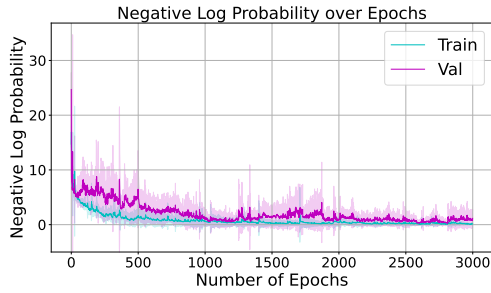
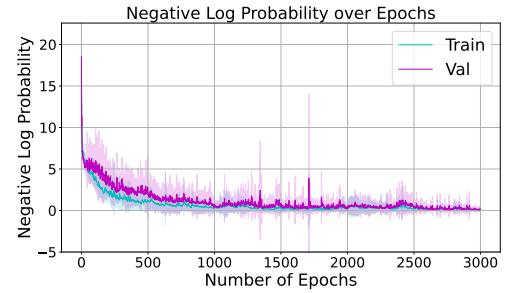


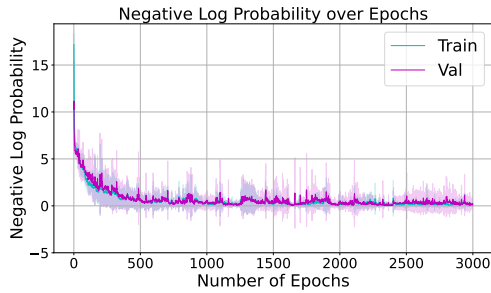
Figure A.15: Results for CoT with fixed token embedding and fixed positional embedding. All other configurations are set as default values as in Table A.2.



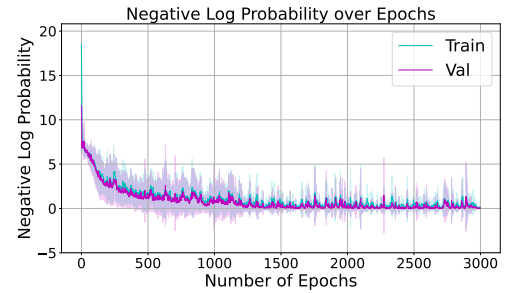
(a) Training set size 160



(b) Training set size 250



(c) Training set size 400



(d) Training set size 1600

Figure A.16: Results for CoT where each entity is represented by two tokens, i.e.,  $A_i$ ,  $B_i$ , or  $C_i$ . The validation set sizes are 50. The model is able to “deduce” unseen  $A_i \rightsquigarrow C_i$  by learning underlying patterns.

# Appendix B

## Appendix for Out-of-context Reasoning

### B.1 Proof of Section 3.4.1

In this section, we provide proof for all theoretical results presented in Section 3.4.1. Specifically, we provide the proof of Proposition 3 in Section B.1.1, Theorem 5 in Section B.1.2, and Theorem 6 in Section B.1.3.

#### B.1.1 Proof of Proposition 3

*Proof of Proposition 3.* We show that for any fixed  $\theta = (\mathbf{W}_O, \mathbf{W}_V)$ , there is a matrix  $\tilde{\theta} = \mathbf{W}_{OV}$  that gives the same test loss as defined in (3.5) and the same training loss as defined in (3.4) and vice versa. Following (3.3), for any input  $z_{1:T}$  and answer token  $a$ , we define the logit functions given by the two sets of parameters as:

$$f_{\theta}(z_{1:T}, a) = \mathbf{e}_a^\top \mathbf{W}_O \mathbf{W}_V^\top \mathbf{X}^\top \mathbf{X} \mathbf{W}_{KQ} \mathbf{x}_T, \quad f_{\tilde{\theta}}(z_{1:T}, a) = \mathbf{e}_a^\top \mathbf{W}_{OV} \mathbf{X}^\top \mathbf{X} \mathbf{W}_{KQ} \mathbf{x}_T.$$

It suffices to prove that the two logit functions always give the same value for any  $(z_{1:T}, a)$ . Given any fixed  $\theta = (\mathbf{W}_O, \mathbf{W}_V)$ , we can set  $\mathbf{W}_{OV} := \mathbf{W}_O \mathbf{W}_V^\top$  and get

$$f_{\theta}(z_{1:T}, a) = \mathbf{e}_a^\top \mathbf{W}_O \mathbf{W}_V^\top \mathbf{X}^\top \mathbf{X} \mathbf{W}_{KQ} \mathbf{x}_T = f_{\tilde{\theta}}(z_{1:T}, a).$$

For the other direction, given any  $\mathbf{W}_{OV} \in \mathbb{R}^{d \times d}$ , suppose its SVD decomposition is given by  $\mathbf{W}_{OV} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^\top$  where  $\mathbf{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_r) \in \mathbb{R}^{r \times r}$  with  $r \leq d$  and  $\sigma_i > 0$  for  $i \in [r]$ . Since the rank of  $\mathbf{W}_{OV}$  is at most  $d$  and  $d_h \geq d \geq r$ , let  $\mathbf{\Sigma}^{1/2} = \text{diag}(\sqrt{\sigma_1}, \dots, \sqrt{\sigma_r})$  and  $\mathbf{Q} = [\mathbf{I}_r \ \mathbf{0}_{r \times (d_h - r)}] \in \mathbb{R}^{r \times d_h}$ . Note that  $\mathbf{Q} \mathbf{Q}^\top = \mathbf{I}_r$ . Then we can set

$$\mathbf{W}_O := \mathbf{U} \mathbf{\Sigma}^{1/2} \mathbf{Q}, \quad \mathbf{W}_V := \mathbf{V} \mathbf{\Sigma}^{1/2} \mathbf{Q},$$

such that  $\mathbf{W}_O \mathbf{W}_V^\top = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^\top = \mathbf{W}_{OV}$ . Combining both directions, we can conclude that the two parameterizations have equivalent expressive power. However, in the following analysis, we show that there is a key distinction between the two in terms of optimization dynamics.  $\square$

### B.1.2 Proof of Theorem 5

*Proof of Theorem 5.*

1. For the factorized model  $\theta = (\mathbf{W}_V, \mathbf{W}_O)$ , it is a two-layer fully-connected linear network trained by cross-entropy loss. By Theorem 4.4 and Appendix G of [86], every limit point of  $\left\{ \frac{\theta(t)}{\|\theta(t)\|}, t \geq 0 \right\}$  by gradient descent with small enough learning rates or gradient flow is along the direction of a KKT point of the following program:

$$\begin{aligned} & \min_{\mathbf{W}_O, \mathbf{W}_V} \frac{1}{2} (\|\mathbf{W}_O\|_F^2 + \|\mathbf{W}_V\|_F^2) \\ & \text{s.t. } h_{(s,r),a'}(\mathbf{W}_O \mathbf{W}_V^\top) \geq 1, \forall (s,r) \in \mathcal{D}_{\text{train}}, \forall a' \in \mathcal{A} \setminus \{a^*(s,r)\}. \end{aligned} \quad (\text{OV-SVM})$$

Moreover, Theorem 5 shows that the above program has the same solutions for  $\mathbf{W}_{OV}^F = \mathbf{W}_O \mathbf{W}_V^\top$  as  $(\mathbf{W}_{OV}^F\text{-SVM})$ .

2. For the non-factorized model  $\mathbf{W}_{OV}$ , it is a linear model trained by cross-entropy loss. Again, by Theorem 4.4 and Appendix G of [86], every limit point of  $\left\{ \frac{\mathbf{W}_{OV}(t)}{\|\mathbf{W}_{OV}(t)\|_F}, t \geq 0 \right\}$  by gradient descent with small enough learning rates or gradient flow is along the direction of a KKT point of  $(\mathbf{W}_{OV}\text{-SVM})$ . Since  $(\mathbf{W}_{OV}\text{-SVM})$  is a convex program, the KKT point is sufficient to ensure global optimality.

□

**Remark 5.** *The results in Theorem 5 can be further strengthened under certain conjectures that extend previous results for binary classification to multi-class settings.*

*For the non-factorized model, if our dataset satisfies Equation (15) in Theorem 7 in [120], it can be shown that the parameter  $\mathbf{W}_{OV}$  under gradient flow or gradient descent with a small enough step size directionally converges. Equation (15) is proved to be true in [120] for the binary setting for almost all datasets, and conjectured to be true in multi-class settings for almost all datasets. Therefore, combining the result of Theorem 5 for the non-factorized model, if the above conjecture is true for our dataset, gradient flow or gradient descent with small enough step sizes directionally converges to the direction of the global minimum of  $(\mathbf{W}_{OV}\text{-SVM})$ .*

*For the factorized model, Theorem 3.1 of [136] shows that gradient flow directionally converges to the direction of the global minimum of  $(\mathbf{W}_{OV}^F\text{-SVM})$  for binary classification. We conjecture this is also true for a multi-class setting under certain mild assumptions, and we leave the proof for future work.*

The proof of Theorem 5 concludes with the following lemma, which establishes the equivalence between the solutions of  $(\text{OV-SVM})$  and  $(\mathbf{W}_{OV}^F\text{-SVM})$ .

**Lemma 17.** *Let  $\mathbf{W}^*$  be an optimal solution of  $(\mathbf{W}_{OV}^F\text{-SVM})$  with  $\text{rank}(\mathbf{W}^*) = r$ . Assume  $d_h \geq r$ . The optimization problem  $(\text{OV-SVM})$  is equivalent to  $(\mathbf{W}_{OV}^F\text{-SVM})$ . As a result, if*

$(\mathbf{W}_O, \mathbf{W}_V)$  is a solution of (OV-SVM), then its combined form  $\mathbf{W} = \mathbf{W}_O \mathbf{W}_V^\top$  is also a global minimum of ( $\mathbf{W}_{OV}^F$ -SVM).

*Proof.* We adopt a similar argument as in [109]. Consider any optimal solution  $(\mathbf{W}_O, \mathbf{W}_V)$  in (OV-SVM) and let  $\mathbf{W} := \mathbf{W}_O \mathbf{W}_V^\top \in \mathbb{R}^{d \times d}$ , for any  $(s, r) \in \mathcal{D}_{\text{train}}$ ,  $a' \in \mathcal{A} \setminus \{a^*(s, r)\}$ , we have

$$h_{(s,r),a'}(\mathbf{W}_O \mathbf{W}_V^\top) = h_{(s,r),a'}(\mathbf{W}) \geq 1.$$

Thus,  $\mathbf{W}$  is inside the feasible set of ( $\mathbf{W}_{OV}^F$ -SVM). Moreover, note that the nuclear norm is the dual norm of the spectral norm, which gives

$$\begin{aligned} \|\mathbf{W}\|_* &= \sup_{\|\mathbf{Z}\|_2 \leq 1} \text{trace}(\mathbf{Z}^\top \mathbf{W}_O \mathbf{W}_V^\top) \\ &= \sup_{\|\mathbf{Z}\|_2 \leq 1} \langle \mathbf{Z} \mathbf{W}_V, \mathbf{W}_O \rangle \\ &\leq \sup_{\|\mathbf{Z}\|_2 \leq 1} \|\mathbf{Z} \mathbf{W}_V\|_F \|\mathbf{W}_O\|_F \\ &\stackrel{(a)}{\leq} \frac{1}{2} (\|\mathbf{W}_O\|_F^2 + \|\mathbf{W}_V\|_F^2), \end{aligned} \tag{B.1}$$

where (a) follows  $\|\mathbf{A}\mathbf{B}\|_F \leq \|\mathbf{A}\|_2 \|\mathbf{B}\|_F$  and AM-GM inequality. Now assuming  $\mathbf{W}^*$  is a optimal solution of ( $\mathbf{W}_{OV}^F$ -SVM) and  $\|\mathbf{W}\|_* > \|\mathbf{W}^*\|_*$ . Let its SVD decomposition be  $\mathbf{W}^* = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^\top$  with  $\mathbf{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_r) \in \mathbb{R}^{r \times r}$ . We can construct  $\mathbf{W}_O^* = \mathbf{U} \mathbf{\Sigma}^{1/2}$  and  $\mathbf{W}_V^* = \mathbf{V} \mathbf{\Sigma}^{1/2}$  such that

$$\frac{1}{2} (\|\mathbf{W}_O^*\|_F^2 + \|\mathbf{W}_V^*\|_F^2) = \|\mathbf{\Sigma}^{1/2}\|_F^2 = \text{trace}(\mathbf{\Sigma}) = \|\mathbf{W}^*\|_* < \|\mathbf{W}\|_* \stackrel{(a)}{\leq} \frac{1}{2} (\|\mathbf{W}_O\|_F^2 + \|\mathbf{W}_V\|_F^2),$$

where (a) follows (B.1). Note that here we assumed  $d_h = r$ . If  $d_h > r$ , we can always choose

$$\tilde{\mathbf{\Sigma}}^{1/2} := [\mathbf{\Sigma}^{1/2}, \mathbf{0}_{r \times (d_h - r)}] \in \mathbb{R}^{r \times d_h}, \mathbf{W}_O^* = \mathbf{U} \tilde{\mathbf{\Sigma}}^{1/2}, \mathbf{W}_V^* = \mathbf{V} \tilde{\mathbf{\Sigma}}^{1/2},$$

which yields the same result. Moreover,  $(\mathbf{W}_O^*, \mathbf{W}_V^*)$  is a solution of (OV-SVM) as  $\mathbf{W}^*$  is a feasible solution of ( $\mathbf{W}_{OV}^F$ -SVM). This leads to a contradiction since  $(\mathbf{W}_O, \mathbf{W}_V)$  is an optimal solution of (OV-SVM). Conversely, we prove that if  $\mathbf{W}^* = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^\top$  is an optimal solution of ( $\mathbf{W}_{OV}^F$ -SVM),  $(\mathbf{W}_O^*, \mathbf{W}_V^*) := (\mathbf{U} \mathbf{\Sigma}^{1/2}, \mathbf{V} \mathbf{\Sigma}^{1/2})$  is also an optimal solution of (OV-SVM). Assume it's not optimal and thus there exists a feasible solution  $(\mathbf{W}_O, \mathbf{W}_V)$  such that

$$\frac{1}{2} (\|\mathbf{W}_O\|_F^2 + \|\mathbf{W}_V\|_F^2) < \frac{1}{2} (\|\mathbf{W}_O^*\|_F^2 + \|\mathbf{W}_V^*\|_F^2).$$

Using the same argument we have

$$\|\mathbf{W}^*\|_* = \frac{1}{2} (\|\mathbf{W}_O^*\|_F^2 + \|\mathbf{W}_V^*\|_F^2) > \frac{1}{2} (\|\mathbf{W}_O\|_F^2 + \|\mathbf{W}_V\|_F^2) \geq \|\mathbf{W}\|_*,$$

which again leads to a contradiction. Combining both directions, we conclude that the two problems are equivalent. Eventually, if  $(\mathbf{W}_O, \mathbf{W}_V)$  is a global minimum of (OV-SVM), the combined parameter  $\mathbf{W} = \mathbf{W}_O \mathbf{W}_V^\top$  is also a global minimum of ( $\mathbf{W}_{OV}^F$ -SVM). This finishes the proof of Theorem 5.  $\square$

### B.1.3 Proof of Theorem 6

**Useful notations.** We introduce useful notations used in this section. We use  $\mathbf{I}_n$  to represent an  $n \times n$  identity matrix, use  $\mathbf{E}_n$  to represent an  $n \times n$  all-one matrix, and use  $\mathbf{1}_n$  and  $\mathbf{0}_n$  to represent  $n$ -dimensional all-one and all-zero vectors, respectively. We use  $\mathbf{e}_i = [0, \dots, 1, \dots, 0]^\top$  as the one-hot vector in  $\mathbb{R}^n$  where the  $i$ -th entry is one. For convenience, we use  $x \wedge y$  to denote the minimum value among  $x$  and  $y$  and use  $x \vee y$  to denote the maximum value among them.

#### B.1.3.1 Proof for factorized model

Note that although  $\mathbf{W}_{\text{OV}}^{\text{F}}$  is a  $d \times d$  matrix, since we restrict the next token prediction to be among  $2n$  answer tokens in  $\mathcal{A}$ , and only  $(nm + 2)$  tokens in  $\mathcal{S} \cup \mathcal{R}$  can take effect in the prompt, we only need to consider a reduced matrix  $\mathbf{W}_{\text{OV}}^{\text{F}} \in \mathbb{R}^{(2n) \times (nm+2)}$  throughout this section, where each row corresponds to a token in  $\mathcal{A}$  and each column corresponds to a token in  $\mathcal{S} \cup \mathcal{R}$ . Now we restate the first part of Theorem 6 below.

**Theorem 15** (Part 1 in Theorem 6: Factorized model has OCR ability). *Let  $n > 1$ . Suppose  $\mathbf{W}_{\text{OV}}^{\text{F}}$  is a solution to the SVM problem in Eq. ( $\mathbf{W}_{\text{OV}}^{\text{F}}$ -SVM), then for any  $(s, r) \in \mathcal{D}_{\text{test}}$ ,  $a' \in \mathcal{A} \setminus \{a^*(s, r)\}$ , given regularity conditions (Assumption 4), it holds that*

$$h_{(s,r),a'}(\mathbf{W}_{\text{OV}}^{\text{F}}) \geq \sqrt{\frac{m_{\text{train}}}{m_{\text{test}}}} \wedge 1, \text{ indicating the OCR ability.} \quad (\text{B.2})$$

To prove Theorem 15, we derive an explicit solution characterization for ( $\mathbf{W}_{\text{OV}}^{\text{F}}$ -SVM). The proof roadmap is as follows.

1. **Restricted Form Existence:** Lemmas 18 and 19 show the existence of a solution in block structure (B.3) via permutation averaging and the convexity of nuclear norm.
2. **SVD Computation:** Lemma 20 computes the closed form for the SVD decomposition of the restricted form in Lemma 19.
3. **Nuclear Norm Formula of the restricted form:** Given the restricted form, Lemma 21 gives  $\|\mathbf{W}_{\text{OV}}^{\text{F}}\|_*$  in closed form (B.12).
4. **Optimization:** Lemma 22 finds the minimum of Equation (B.12) by decomposing  $\|\mathbf{W}_{\text{OV}}^{\text{F}}\|_* = M_1 + M_2$ .
5. **Solution characterization and the uniqueness:** Theorem 16 uses Lemmas 23 and 24 and Assumption 4 to establish the unique forms of the solution in Equations (B.22) and (B.23).

**Lemma 18** (Unitary invariance). *Given matrix  $\mathbf{A}$ , for any orthonormal matrices  $\mathbf{U}$  and  $\mathbf{V}$ , we have that*

$$\|\mathbf{U}\mathbf{A}\mathbf{V}^\top\|_* = \|\mathbf{A}\|_*.$$

*Proof.* See Lemma 2.5 in [56].  $\square$

**Lemma 19** (Existence of a restricted form solution to ( $\mathbf{W}_{\text{OV}}^{\text{F-SVM}}$ )). Suppose  $\mathbf{W}_{\text{OV}}^{\text{F}}$  is the solution to the optimization problem ( $\mathbf{W}_{\text{OV}}^{\text{F-SVM}}$ ). There exists a solution with  $p_1, p_2, q_1, q_2, f_1, f_2, g_1, g_2, \beta_1, \beta_2$  and  $\gamma_1, \gamma_2$  such that

$$\mathbf{W}_{\text{OV}}^{\text{F}} = \begin{bmatrix} \underbrace{p_1 \mathbf{I}_n + p_2 \mathbf{E}_n \cdots p_1 \mathbf{I}_n + p_2 \mathbf{E}_n}_{m_{\text{train}} \text{ blocks}} & \underbrace{f_1 \mathbf{I}_n + f_2 \mathbf{E}_n \cdots f_1 \mathbf{I}_n + f_2 \mathbf{E}_n}_{m_{\text{test}} \text{ blocks}} & \beta_1 \mathbf{1}_n & \beta_2 \mathbf{1}_n \\ \underbrace{q_1 \mathbf{I}_n + q_2 \mathbf{E}_n \cdots q_1 \mathbf{I}_n + q_2 \mathbf{E}_n}_{m_{\text{train}} \text{ blocks}} & \underbrace{g_1 \mathbf{I}_n + g_2 \mathbf{E}_n \cdots g_1 \mathbf{I}_n + g_2 \mathbf{E}_n}_{m_{\text{test}} \text{ blocks}} & \gamma_1 \mathbf{1}_n & \gamma_2 \mathbf{1}_n \end{bmatrix}. \quad (\text{B.3})$$

Moreover,

$$\begin{aligned} p_1, f_1, q_1 &\geq 1, \\ p_1 + p_2 + \beta_1 &\geq q_1 + q_2 + \gamma_1 + 1, \\ q_1 + q_2 + \gamma_2 &\geq p_1 + p_2 + \beta_2 + 1, \\ f_1 + f_2 + \beta_1 &\geq (g_1 \vee 0) + g_2 + \gamma_1 + 1. \end{aligned} \quad (\text{B.4})$$

*Proof of Lemma 19.* We first show that certain permutations of  $\mathbf{W}_{\text{OV}}^{\text{F}}$  are still solutions of the optimization problem. Suppose that  $\sigma$  is any permutation of  $\{1, \dots, n\}$ . Define  $\mathbf{P}_\sigma \in \mathbb{R}^{n \times n}$  as the corresponding permutation matrix. Consider the permuted weight matrix

$$\sigma(\mathbf{W}_{\text{OV}}^{\text{F}}) = \begin{bmatrix} \mathbf{P}_\sigma & 0 \\ 0 & \mathbf{P}_\sigma \end{bmatrix} \mathbf{W}_{\text{OV}}^{\text{F}} \text{diag}\{\mathbf{P}_\sigma, \dots, \mathbf{P}_\sigma, 1, 1\}.$$

It is equivalent to permuting the subject sets and the fact labels  $b$  and  $c$  simultaneously with  $\sigma$ :  $\{\mathcal{S}_{\sigma(1)}, \dots, \mathcal{S}_{\sigma(n)}\}$ ,  $\{b_{\sigma(1)}, \dots, b_{\sigma(n)}\}$  and  $\{c_{\sigma(1)}, \dots, c_{\sigma(n)}\}$ . Using Equation (3.6), we have that

$$\begin{aligned} h_{(s_{\sigma(i)}, r_1), b_{\sigma(j)}}(\sigma(\mathbf{W}_{\text{OV}}^{\text{F}})) &= h_{(s_i, r_1), b_j}(\mathbf{W}_{\text{OV}}^{\text{F}}) \geq 1, \quad \forall j \in [n] \setminus \{i\}, \\ h_{(s_{\sigma(i)}, r_1), c_{\sigma(j)}}(\sigma(\mathbf{W}_{\text{OV}}^{\text{F}})) &= h_{(s_i, r_1), c_j}(\mathbf{W}_{\text{OV}}^{\text{F}}) \geq 1, \quad \forall j \in [n], \\ h_{(s_{\sigma(i)}, r_2), b_{\sigma(j)}}(\sigma(\mathbf{W}_{\text{OV}}^{\text{F}})) &= h_{(s_i, r_2), b_j}(\mathbf{W}_{\text{OV}}^{\text{F}}) \geq 1, \quad \forall j \in [n], \\ h_{(s_{\sigma(i)}, r_2), c_{\sigma(j)}}(\sigma(\mathbf{W}_{\text{OV}}^{\text{F}})) &= h_{(s_i, r_2), c_j}(\mathbf{W}_{\text{OV}}^{\text{F}}) \geq 1, \quad \forall j \in [n] \setminus \{i\}, \end{aligned}$$

for any  $i$ . From Lemma 18, since permutation matrices are orthonormal,  $\|\sigma(\mathbf{W}_{\text{OV}}^{\text{F}})\|_* = \|\mathbf{W}_{\text{OV}}^{\text{F}}\|_*$ . Therefore,  $\sigma(\mathbf{W}_{\text{OV}}^{\text{F}})$  is also a solution to the optimization problem ( $\mathbf{W}_{\text{OV}}^{\text{F-SVM}}$ ). Let's consider the average over all possible permutations

$$\begin{aligned} &\frac{\sum_{\sigma} \sigma(\mathbf{W}_{\text{OV}}^{\text{F}})}{n!} \\ &= \begin{bmatrix} \underbrace{p_{11} \mathbf{I}_n + p_{21} \mathbf{E}_n \cdots p_{1m_{\text{train}}} \mathbf{I}_n + p_{2m_{\text{train}}} \mathbf{E}_n}_{m_{\text{train}} \text{ blocks}} & \underbrace{f_{11} \mathbf{I}_n + f_{21} \mathbf{E}_n \cdots f_{1m_{\text{test}}} \mathbf{I}_n + f_{2m_{\text{test}}} \mathbf{E}_n}_{m_{\text{test}} \text{ blocks}} & \beta_1 \mathbf{1}_n & \beta_2 \mathbf{1}_n \\ \underbrace{q_{11} \mathbf{I}_n + q_{21} \mathbf{E}_n \cdots q_{1m_{\text{train}}} \mathbf{I}_n + q_{2m_{\text{train}}} \mathbf{E}_n}_{m_{\text{train}} \text{ blocks}} & \underbrace{g_{11} \mathbf{I}_n + g_{21} \mathbf{E}_n \cdots g_{1m_{\text{test}}} \mathbf{I}_n + g_{2m_{\text{test}}} \mathbf{E}_n}_{m_{\text{test}} \text{ blocks}} & \gamma_1 \mathbf{1}_n & \gamma_2 \mathbf{1}_n \end{bmatrix}. \end{aligned}$$

It is also a solution to the optimization problem ( $\mathbf{W}_{\text{OV}}^{\text{F}}\text{-SVM}$ ) due to the convexity of the nuclear norm.

We can consider other permutations. Suppose that  $\tau_{\text{train}}$  is a permutation of the index set  $\{1, \dots, m_{\text{train}}\}$ . Define the permuted weight matrix

$$\tau_{\text{train}}(\mathbf{W}_{\text{OV}}^{\text{F}}) = \mathbf{W}_{\text{OV}}^{\text{F}} \text{diag}\{\mathbf{P}_{\tau_{\text{train}}} \otimes \mathbf{I}_n, \underbrace{1, \dots, 1}_{nm_{\text{test}}+2}\}.$$

It is equivalent to permute all the subjects in the set  $\mathcal{S}_{i,\text{train}}: \{s_{i,\tau_{\text{train}}(1)}, \dots, s_{i,\tau_{\text{train}}(m_{\text{train}})}\}$  for any  $i = 1, \dots, n$ . Note that there is no need to permute the labels, as subjects in  $\mathcal{S}_{i,\text{train}}$  share the same label pair  $b_i$  and  $c_i$ . Since the permuted  $\mathcal{S}_{i,\text{train}}$  is still disjoint with the test subject set  $\mathcal{S}_{i,\text{test}}$ , we have that for any  $i \in [n]$ ,  $j \in [m_{\text{train}}]$ , it holds that

$$\begin{aligned} h_{(s_{i,\tau_{\text{train}}(j)}, r_1), a'}(\tau_{\text{train}}(\mathbf{W}_{\text{OV}}^{\text{F}})) &= h_{(s_{i,j}, r_1), a'}(\mathbf{W}_{\text{OV}}^{\text{F}}) \geq 1, \quad \forall a' \in \mathcal{A} \setminus \{b_i\}, \\ h_{(s_{i,\tau_{\text{train}}(j)}, r_2), a'}(\tau_{\text{train}}(\mathbf{W}_{\text{OV}}^{\text{F}})) &= h_{(s_{i,j}, r_2), a'}(\mathbf{W}_{\text{OV}}^{\text{F}}) \geq 1, \quad \forall a' \in \mathcal{A} \setminus \{c_i\}. \end{aligned}$$

From Lemma 18, we can conclude that  $\|\tau_{\text{train}}(\mathbf{W}_{\text{OV}}^{\text{F}})\|_* = \|\mathbf{W}_{\text{OV}}^{\text{F}}\|_*$ . Averaging over all permutations  $\sigma$  and  $\tau_{m_{\text{train}}}$ , we have that

$$\begin{aligned} & \frac{\sum_{\tau_{\text{train}}} \sum_{\sigma} \tau_{\text{train}}(\sigma(\mathbf{W}_{\text{OV}}^{\text{F}}))}{m_{\text{train}}!n!} \\ &= \underbrace{\begin{bmatrix} p_1 \mathbf{I}_n + p_2 \mathbf{E}_n \cdots p_1 \mathbf{I}_n + p_2 \mathbf{E}_n \\ q_1 \mathbf{I}_n + q_2 \mathbf{E}_n \cdots q_1 \mathbf{I}_n + q_2 \mathbf{E}_n \end{bmatrix}}_{m_{\text{train}} \text{ blocks}} \underbrace{\begin{bmatrix} f_{11} \mathbf{I}_n + f_{21} \mathbf{E}_n \cdots f_{1m_{\text{test}}} \mathbf{I}_n + f_{2m_{\text{test}}} \mathbf{E}_n \\ g_{11} \mathbf{I}_n + g_{21} \mathbf{E}_n \cdots g_{1m_{\text{test}}} \mathbf{I}_n + g_{2m_{\text{test}}} \mathbf{E}_n \end{bmatrix}}_{m_{\text{test}} \text{ blocks}} \begin{bmatrix} \beta_1 \mathbf{1}_n & \beta_2 \mathbf{1}_n \\ \gamma_1 \mathbf{1}_n & \gamma_2 \mathbf{1}_n \end{bmatrix}. \end{aligned}$$

We can then consider the permutation over the remaining  $m_{\text{test}}$  indices  $\{m_{\text{train}} + 1, \dots, m\}$ . Let  $\tau_{\text{test}}$  denote the permutation. Consider the permuted weight matrix.

$$\tau_{\text{test}}(\mathbf{W}_{\text{OV}}^{\text{F}}) = \mathbf{W}_{\text{OV}}^{\text{F}} \text{diag}\{\underbrace{1, \dots, 1}_{nm_{\text{train}}}, \mathbf{P}_{\tau_{\text{test}}} \otimes \mathbf{I}_n, 1, 1\}.$$

It is equivalent to permute all subjects in the set  $\mathcal{S}_{i,\text{test}}: \{s_{i,\tau_{\text{test}}(m_{\text{train}}+1)}, \dots, s_{i,\tau_{\text{test}}(m)}\}$  for any  $i = 1, \dots, n$ . We have that for any  $i \in [n]$ ,  $j \in [m] \setminus [m_{\text{train}}]$ , it holds that

$$h_{(s_{i,\tau_{\text{test}}(j)}, r_1), a'}(\tau_{\text{test}}(\mathbf{W}_{\text{OV}}^{\text{F}})) = h_{(s_{i,j}, r_1), a'}(\mathbf{W}_{\text{OV}}^{\text{F}}) \geq 1, \quad \forall a' \in \mathcal{A} \setminus \{b_i\}.$$

Taking the average weight over all possible permutations  $\tau_{\text{test}}$ ,

$$\begin{aligned} & \frac{\sum_{\tau_{\text{test}}} \sum_{\tau_{\text{train}}} \sum_{\sigma} \tau_{\text{test}}(\tau_{\text{train}}(\sigma(\mathbf{W}_{\text{OV}}^{\text{F}})))}{m_{\text{test}}!m_{\text{train}}!n!} \\ &= \underbrace{\begin{bmatrix} p_1 \mathbf{I}_n + p_2 \mathbf{E}_n \cdots p_1 \mathbf{I}_n + p_2 \mathbf{E}_n \\ q_1 \mathbf{I}_n + q_2 \mathbf{E}_n \cdots q_1 \mathbf{I}_n + q_2 \mathbf{E}_n \end{bmatrix}}_{m_{\text{train}} \text{ blocks}} \underbrace{\begin{bmatrix} f_1 \mathbf{I}_n + f_2 \mathbf{E}_n \cdots f_1 \mathbf{I}_n + f_2 \mathbf{E}_n \\ g_1 \mathbf{I}_n + g_2 \mathbf{E}_n \cdots g_1 \mathbf{I}_n + g_2 \mathbf{E}_n \end{bmatrix}}_{m_{\text{test}} \text{ blocks}} \begin{bmatrix} \beta_1 \mathbf{1}_n & \beta_2 \mathbf{1}_n \\ \gamma_1 \mathbf{1}_n & \gamma_2 \mathbf{1}_n \end{bmatrix}. \end{aligned}$$

This shows that Equation (B.3) is one solution to the optimization problem ( $\mathbf{W}_{\text{OV}}^{\text{F-SVM}}$ ). This proves Lemma 19.  $\square$

We could compute the closed form of the SVD decomposition for Equation (B.3).

**Lemma 20.** *The restricted form in Equation (B.3) has a the SVD decomposition  $\mathbf{W}_{\text{OV}}^{\text{F}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\top}$  with*

$$\begin{aligned}\mathbf{U} &= \left[ \mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}_2^{(1)}, \dots, \mathbf{u}_n^{(1)}, \mathbf{u}_2^{(2)}, \dots, \mathbf{u}_n^{(2)} \right], \\ \mathbf{\Sigma} &= \text{diag} \left\{ \sigma_1^{(1)}, \sigma_1^{(2)}, \underbrace{\sigma_2^{(1)}, \dots, \sigma_2^{(1)}}_{n-1}, \underbrace{\sigma_2^{(2)}, \dots, \sigma_2^{(2)}}_{n-1} \right\}, \\ \mathbf{V} &= \left[ \mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \mathbf{v}_2^{(1)}, \dots, \mathbf{v}_n^{(1)}, \mathbf{v}_2^{(2)}, \dots, \mathbf{v}_n^{(2)} \right],\end{aligned}$$

where  $\mathbf{u}^{(k)}$  are defined in Equation (B.6);  $\mathbf{u}_j^{(k')}$  are defined in Equation (B.8),  $\sigma_1^{(k)}$  and  $\sigma_2^{(k')}$  are defined in Equation (B.9);  $\mathbf{v}^{(k)}$  are defined in Equation (B.10);  $\mathbf{v}_j^{(k')}$  are defined in Equation (B.11).

*The proof of Lemma 20.* The dimensions of  $\mathbf{W}_{\text{OV}}^{\text{F}}$  are  $2n \times N_c$ , where  $N_c = (m_{\text{train}} + m_{\text{test}})n + 2$ . Denote the Singular Value Decomposition (SVD) of  $\mathbf{W}_{\text{OV}}^{\text{F}}$  by  $\mathbf{W}_{\text{OV}}^{\text{F}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\top}$ .

**Orthonormal basis and matrix properties.** Given an orthonormal basis  $\{\boldsymbol{\eta}_1, \dots, \boldsymbol{\eta}_n\}$  for  $\mathbb{R}^n$ , with  $\boldsymbol{\eta}_1 = \mathbf{1}_n / \sqrt{n}$ , where  $\mathbf{1}_n$  is the  $n$ -dim all-one column vector. The vectors  $\boldsymbol{\eta}_2, \dots, \boldsymbol{\eta}_n$  are orthonormal to each other and to  $\boldsymbol{\eta}_1$ . Let  $\mathbf{X} = c_1 \mathbf{I}_n + c_2 \mathbf{E}_n$  be a block appearing in the matrix  $\mathbf{W}_{\text{OV}}^{\text{F}}$ . Its action on the basis vectors is:

- $\mathbf{X}\boldsymbol{\eta}_1 = (c_1 \mathbf{I}_n + c_2 \mathbf{E}_n)(\frac{1}{\sqrt{n}} \mathbf{1}_n) = c_1 \frac{1}{\sqrt{n}} \mathbf{1}_n + c_2 \frac{1}{\sqrt{n}} (\mathbf{1}_n \mathbf{1}_n^{\top}) \mathbf{1}_n = c_1 \boldsymbol{\eta}_1 + c_2 \frac{n}{\sqrt{n}} \mathbf{1}_n = (c_1 + nc_2) \boldsymbol{\eta}_1$ .
- For  $j \in \{2, \dots, n\}$ ,  $\mathbf{X}\boldsymbol{\eta}_j = (c_1 \mathbf{I}_n + c_2 \mathbf{E}_n) \boldsymbol{\eta}_j = c_1 \boldsymbol{\eta}_j$ , since  $\mathbf{E}_n \boldsymbol{\eta}_j = \mathbf{1}_n (\mathbf{1}_n^{\top} \boldsymbol{\eta}_j) = \mathbf{0}_n$  due to  $\boldsymbol{\eta}_j \perp \mathbf{1}_n$ .

**Left singular vectors  $\mathbf{U}$  and singular values  $\mathbf{\Sigma}$ .** The left singular vectors (columns of  $\mathbf{U}$ ) are eigenvectors of  $\mathbf{W}_{\text{OV}}^{\text{F}} \mathbf{W}_{\text{OV}}^{\text{F}\top}$ . The matrix  $\mathbf{W}_{\text{OV}}^{\text{F}} \mathbf{W}_{\text{OV}}^{\text{F}\top}$  is a  $2n \times 2n$  symmetric matrix. After block multiplication,  $\mathbf{W}_{\text{OV}}^{\text{F}} \mathbf{W}_{\text{OV}}^{\text{F}\top}$  can be written as:

$$\mathbf{W}_{\text{OV}}^{\text{F}} \mathbf{W}_{\text{OV}}^{\text{F}\top} = \begin{pmatrix} C_{A1} \mathbf{I}_n + C_{A2} \mathbf{E}_n & C_{B1} \mathbf{I}_n + C_{B2} \mathbf{E}_n \\ C_{B1} \mathbf{I}_n + C_{B2} \mathbf{E}_n & C_{D1} \mathbf{I}_n + C_{D2} \mathbf{E}_n \end{pmatrix},$$

where the coefficients are:

$$\begin{aligned}
C_{A1} &= m_{\text{train}} p_1^2 + m_{\text{test}} f_1^2, \\
C_{A2} &= m_{\text{train}} (2p_1 p_2 + n p_2^2) + m_{\text{test}} (2f_1 f_2 + n f_2^2) + \beta_1^2 + \beta_2^2, \\
C_{D1} &= m_{\text{train}} q_1^2 + m_{\text{test}} g_1^2, \\
C_{D2} &= m_{\text{train}} (2q_1 q_2 + n q_2^2) + m_{\text{test}} (2g_1 g_2 + n g_2^2) + \gamma_1^2 + \gamma_2^2, \\
C_{B1} &= m_{\text{train}} p_1 q_1 + m_{\text{test}} f_1 g_1, \\
C_{B2} &= m_{\text{train}} (p_1 q_2 + p_2 q_1 + n p_2 q_2) + m_{\text{test}} (f_1 g_2 + f_2 g_1 + n f_2 g_2) + \beta_1 \gamma_1 + \beta_2 \gamma_2.
\end{aligned}$$

We seek eigenvectors of  $\mathbf{W}_{\text{OV}}^{\text{F}} \mathbf{W}_{\text{OV}}^{\text{F}\top}$  of the form

$$\mathbf{u} = \begin{pmatrix} x_b \boldsymbol{\eta}_j \\ x_c \boldsymbol{\eta}_j \end{pmatrix}$$

for scalars  $x_b, x_c$ .

**Case 1: Eigenvectors associated with  $\boldsymbol{\eta}_1$ .** For  $j = 1$ , the eigenvalue problem  $\mathbf{W}_{\text{OV}}^{\text{F}} \mathbf{W}_{\text{OV}}^{\text{F}\top} \mathbf{u} = \lambda \mathbf{u}$  transforms into a  $2 \times 2$  eigenvalue problem for the coefficient vector  $(x_b, x_c)^\top$ :

$$\mathbf{H}_1 \begin{pmatrix} x_b \\ x_c \end{pmatrix} = \lambda \begin{pmatrix} x_b \\ x_c \end{pmatrix},$$

where

$$\mathbf{H}_1 = \begin{pmatrix} C_{A1} + n C_{A2} & C_{B1} + n C_{B2} \\ C_{B1} + n C_{B2} & C_{D1} + n C_{D2} \end{pmatrix}. \quad (\text{B.5})$$

Let  $\lambda_1^{(1)}, \lambda_1^{(2)}$  be the eigenvalues of  $\mathbf{H}_1$ , and let  $(x_{1,b}, x_{1,c})^\top$  and  $(x_{2,b}, x_{2,c})^\top$  be the corresponding normalized eigenvectors. These give two singular values:  $\sigma_1^{(k)} = \sqrt{\lambda_1^{(k)}}$  for  $k = 1, 2$ . The corresponding left singular vectors in  $\mathbf{U}$  are

$$\mathbf{u}^{(k)} = \begin{pmatrix} x_{k,b} \boldsymbol{\eta}_1 \\ x_{k,c} \boldsymbol{\eta}_1 \end{pmatrix} \text{ for } k = 1, 2. \quad (\text{B.6})$$

**Case 2: Eigenvectors associated with  $\boldsymbol{\eta}_j$  for  $j \in \{2, \dots, n\}$ .** For  $j \in \{2, \dots, n\}$ , the eigenvalue problem  $\mathbf{W}_{\text{OV}}^{\text{F}} \mathbf{W}_{\text{OV}}^{\text{F}\top} \mathbf{u} = \lambda \mathbf{u}$  reduces to:

$$\mathbf{H}_2 \begin{pmatrix} x_b \\ x_c \end{pmatrix} = \lambda \begin{pmatrix} x_b \\ x_c \end{pmatrix}$$

where

$$\mathbf{H}_2 = \begin{pmatrix} C_{A1} & C_{B1} \\ C_{B1} & C_{D1} \end{pmatrix}. \quad (\text{B.7})$$

Let  $\lambda_2^{(1)}, \lambda_2^{(2)}$  be the eigenvalues of  $\mathbf{H}_2$ , and let  $(y_{1,b}, y_{1,c})^\top$  and  $(y_{2,b}, y_{2,c})^\top$  be the corresponding normalized eigenvectors. These give  $2(n-1)$  singular values:  $\sigma_2^{(k')} = \sqrt{\lambda_2^{(k'')}}$  for  $k' = 1, 2$ . Each of these singular values has a multiplicity of  $(n-1)$ . The corresponding left singular vectors in  $\mathbf{U}$  are

$$\mathbf{u}_j^{(k')} = \begin{pmatrix} y_{k',b}\boldsymbol{\eta}_j \\ y_{k',c}\boldsymbol{\eta}_j \end{pmatrix} \text{ for each } j \in \{2, \dots, n\} \text{ and } k' = 1, 2. \quad (\text{B.8})$$

The matrix  $\mathbf{U}$  is a  $2n \times 2n$  orthogonal matrix whose columns are the  $2n$  left singular vectors  $\mathbf{u}^{(k)}$  and  $\mathbf{u}_j^{(k')}$ .

**The singular matrix.** From the calculation of  $\mathbf{U}$ , we get that

$$\boldsymbol{\Sigma} = \text{diag}\left\{\sigma_1^{(1)}, \sigma_1^{(2)}, \underbrace{\sigma_2^{(1)}, \dots, \sigma_2^{(1)}}_{n-1}, \underbrace{\sigma_2^{(2)}, \dots, \sigma_2^{(2)}}_{n-1}\right\}, \quad (\text{B.9})$$

where  $\sigma_1^{(k)} = \sqrt{\lambda_1^{(k)}}$  for  $k = 1, 2$ , with  $\lambda_1^{(k)}$  being the eigenvalues of the matrix  $\mathbf{H}_1$  in Equation (B.5);  $\sigma_2^{(k')} = \sqrt{\lambda_2^{(k'')}}$  for  $k' = 1, 2$ , with  $\lambda_2^{(k')}$  being the eigenvalues of the matrix  $\mathbf{H}_2$  in Equation (B.7).

**Right singular vectors  $\mathbf{V}$ .** The columns of  $\mathbf{V}$  (right singular vectors) are obtained from  $\mathbf{v}_i = \sigma_i^{-1} \mathbf{W}_{\text{OV}}^{\text{F}\top} \mathbf{u}_i$  for non-zero  $\sigma_i$ . If  $\sigma_i = 0$ ,  $\mathbf{v}_i$  is a normalized vector in the null space of  $\mathbf{W}_{\text{OV}}^{\text{F}}$ . The matrix  $\mathbf{V}$  has dimensions  $N_c \times 2n$ . The  $2n$  vectors  $\mathbf{v}_i$  corresponding to the found singular values are:

**Case 1: Right singular vectors associated with  $\mathbf{u}^{(k)}$ ,  $k = 1, 2$ .** For  $\mathbf{u}^{(k)} = \begin{pmatrix} x_{k,b}\boldsymbol{\eta}_1 \\ x_{k,c}\boldsymbol{\eta}_1 \end{pmatrix}$  and singular value  $\sigma_1^{(k)}$ : The vector  $\mathbf{W}_{\text{OV}}^{\text{F}\top} \mathbf{u}^{(k)}$  has the following structure:

- The first  $m_{\text{train}}$  blocks (each of size  $n$ ) are  $(x_{k,b}(p_1 + np_2) + x_{k,c}(q_1 + nq_2))\boldsymbol{\eta}_1$ .
- The next  $m_{\text{test}}$  blocks (each of size  $n$ ) are  $(x_{k,b}(f_1 + nf_2) + x_{k,c}(g_1 + ng_2))\boldsymbol{\eta}_1$ .
- The last two components are scalar values:  $\sqrt{n}(\beta_1 x_{k,b} + \gamma_1 x_{k,c})$  and  $\sqrt{n}(\beta_2 x_{k,b} + \gamma_2 x_{k,c})$ .

So,  $\mathbf{v}^{(k)} = \mathbf{W}_{\text{OV}}^{\text{FT}} \mathbf{u}^{(k)} / \sigma_1^{(k)}$  is:

$$\mathbf{v}^{(k)} = \frac{1}{\sigma_1^{(k)}} \begin{pmatrix} (x_{k,b}(p_1 + np_2) + x_{k,c}(q_1 + nq_2))\boldsymbol{\eta}_1 \\ \vdots \\ (x_{k,b}(p_1 + np_2) + x_{k,c}(q_1 + nq_2))\boldsymbol{\eta}_1 \\ (x_{k,b}(f_1 + nf_2) + x_{k,c}(g_1 + ng_2))\boldsymbol{\eta}_1 \\ \vdots \\ (x_{k,b}(f_1 + nf_2) + x_{k,c}(g_1 + ng_2))\boldsymbol{\eta}_1 \\ \sqrt{n}(\beta_1 x_{k,b} + \gamma_1 x_{k,c}) \\ \sqrt{n}(\beta_2 x_{k,b} + \gamma_2 x_{k,c}) \end{pmatrix} \text{ for } k = 1, 2. \quad (\text{B.10})$$

Each  $\boldsymbol{\eta}_1$  term represents a column vector of  $n$  elements. This vector  $\mathbf{v}^{(k)}$  has total length  $N_c = m_{\text{train}}n + m_{\text{test}}n + 2$ .

**Case 2: Right singular vectors associated with  $\mathbf{u}_j^{(k')}$ .** For  $\mathbf{u}_j^{(k')} = \begin{pmatrix} y_{k',b}\boldsymbol{\eta}_j \\ y_{k',c}\boldsymbol{\eta}_j \end{pmatrix}$  (where  $j \geq 2$  and  $k' = 1, 2$ ) and singular value  $\sigma_2^{(k')}$ : The vector  $\mathbf{W}_{\text{OV}}^{\text{FT}} \mathbf{u}_j^{(k')}$  has the following structure:

- The first  $m_{\text{train}}$  blocks (each of size  $n$ ) are  $(y_{k',b}p_1 + y_{k',c}q_1)\boldsymbol{\eta}_j$ .
- The next  $m_{\text{test}}$  blocks (each of size  $n$ ) are  $(y_{k',b}f_1 + y_{k',c}g_1)\boldsymbol{\eta}_j$ .
- The last two scalar components are 0, because  $\mathbf{1}_n^\top \boldsymbol{\eta}_j = 0$  for  $j \geq 2$ .

So,  $\mathbf{v}_j^{(k')} = \mathbf{W}_{\text{OV}}^{\text{FT}} \mathbf{u}_j^{(k')} / \sigma_2^{(k')}$  is:

$$\mathbf{v}_j^{(k')} = \frac{1}{\sigma_2^{(k')}} \begin{pmatrix} (y_{k',b}p_1 + y_{k',c}q_1)\boldsymbol{\eta}_j \\ \vdots \\ (y_{k',b}p_1 + y_{k',c}q_1)\boldsymbol{\eta}_j \\ (y_{k',b}f_1 + y_{k',c}g_1)\boldsymbol{\eta}_j \\ \vdots \\ (y_{k',b}f_1 + y_{k',c}g_1)\boldsymbol{\eta}_j \\ 0 \\ 0 \end{pmatrix}. \quad (\text{B.11})$$

There are  $2(n-1)$   $\mathbf{v}_j^{(k')}$  vectors, one for each pair  $(j, k')$  where  $j \in \{2, \dots, n\}$  and  $k' \in \{1, 2\}$ . Each  $\boldsymbol{\eta}_j$  term represents a column vector of  $n$  elements. The vectors  $\mathbf{v}^{(k)}$  and  $\mathbf{v}_j^{(k')}$  are orthonormal and form the columns of  $\mathbf{V}$ .

As a conclusion, the SVD of  $\mathbf{W}_{\text{OV}}^{\text{F}}$  is  $\mathbf{W}_{\text{OV}}^{\text{F}} = \mathbf{U}\Sigma\mathbf{V}^\top$ , where:

- $\mathbf{U}$  is a  $2n \times 2n$  orthogonal matrix with columns  $\mathbf{u}^{(k)}$  and  $\mathbf{u}_j^{(k')}$  as defined in Equations (B.6) and (B.8).

- $\Sigma$  is a  $2n \times 2n$  rectangular diagonal matrix, whose non-zero entries are the singular values  $\sigma_1^{(k)}$  and  $\sigma_2^{(k')}$ , derived from the eigenvalues of  $\mathbf{H}_1$  and  $\mathbf{H}_2$ .
- $\mathbf{V}$  is an  $N_c \times 2n$  matrix with columns  $\mathbf{v}^{(k)}$  and  $\mathbf{v}_j^{(k')}$  as defined in Equations (B.10) and (B.11).

This finishes the proof of Lemma 20.  $\square$

**Lemma 21.** *The  $\mathbf{W}_{\text{OV}}^{\text{F}}$  in restricted form Equation (B.3) has the nuclear norm  $\|\mathbf{W}_{\text{OV}}^{\text{F}}\|_{\star}$ :*

$$\begin{aligned}
& \sqrt{\frac{(C_{A1} + nC_{A2} + C_{D1} + nC_{D2}) + \sqrt{(C_{A1} + nC_{A2} - (C_{D1} + nC_{D2}))^2 + 4(C_{B1} + nC_{B2})^2}}{2}} \\
& + \sqrt{\frac{(C_{A1} + nC_{A2} + C_{D1} + nC_{D2}) - \sqrt{(C_{A1} + nC_{A2} - (C_{D1} + nC_{D2}))^2 + 4(C_{B1} + nC_{B2})^2}}{2}} \\
& + (n-1) \left( \sqrt{\frac{(C_{A1} + C_{D1}) + \sqrt{(C_{A1} - C_{D1})^2 + 4C_{B1}^2}}{2}} + \sqrt{\frac{(C_{A1} + C_{D1}) - \sqrt{(C_{A1} - C_{D1})^2 + 4C_{B1}^2}}{2}} \right). \tag{B.12}
\end{aligned}$$

*Proof of Lemma 21.* The nuclear norm of a matrix  $\mathbf{W}_{\text{OV}}^{\text{F}}$ , denoted as  $\|\mathbf{W}_{\text{OV}}^{\text{F}}\|_{\star}$ , is defined as the sum of its singular values. From Lemma 20 and its proof, the singular values of  $\mathbf{W}_{\text{OV}}^{\text{F}}$  are derived from the eigenvalues of two  $2 \times 2$  matrices,  $\mathbf{H}_1$  and  $\mathbf{H}_2$ .

The singular values are:

1.  $\sigma_1^{(1)}$  and  $\sigma_1^{(2)}$ , which are the square roots of the two eigenvalues of  $\mathbf{H}_1$  defined in Equation (B.5). These singular values each have a multiplicity of 1.
2.  $\sigma_2^{(1)}$  and  $\sigma_2^{(2)}$ , which are the square roots of the two eigenvalues of  $\mathbf{H}_2$  defined in Equation (B.7). As stated in the proof of Lemma 20 (Case 2: Eigenvectors associated with  $\boldsymbol{\eta}_j$  for  $j \in \{2, \dots, n\}$ ), these singular values correspond to the  $n-1$  basis vectors  $\boldsymbol{\eta}_j$  for  $j \in \{2, \dots, n\}$ . Thus, each of  $\sigma_2^{(1)}$  and  $\sigma_2^{(2)}$  has a multiplicity of  $(n-1)$ .

The nuclear norm is therefore the sum of all these singular values:

$$\begin{aligned}
\|\mathbf{W}_{\text{OV}}^{\text{F}}\|_{\star} &= \sigma_1^{(1)} + \sigma_1^{(2)} + (n-1)\sigma_2^{(1)} + (n-1)\sigma_2^{(2)} \\
&= (\sigma_1^{(1)} + \sigma_1^{(2)}) + (n-1)(\sigma_2^{(1)} + \sigma_2^{(2)}).
\end{aligned}$$

Let's expand each term:

**Term 1: Sum of singular values from  $\mathbf{H}_1$ .**

The matrix  $\mathbf{H}_1$  is given by Equation (B.5):

$$\mathbf{H}_1 = \begin{pmatrix} C_{A1} + nC_{A2} & C_{B1} + nC_{B2} \\ C_{B1} + nC_{B2} & C_{D1} + nC_{D2} \end{pmatrix}.$$

Let  $a_1 = C_{A1} + nC_{A2}$ ,  $b_1 = C_{B1} + nC_{B2}$ , and  $c_1 = C_{D1} + nC_{D2}$ . The eigenvalues  $\lambda_1^{(1)}, \lambda_1^{(2)}$  of this symmetric  $2 \times 2$  matrix  $\mathbf{H}_1 = \begin{pmatrix} a_1 & b_1 \\ b_1 & c_1 \end{pmatrix}$  are given by the formula

$$\lambda = \frac{(a_1 + c_1) \pm \sqrt{(a_1 - c_1)^2 + 4b_1^2}}{2}.$$

The singular values  $\sigma_1^{(1)}$  and  $\sigma_1^{(2)}$  are  $\sqrt{\lambda_1^{(1)}}$  and  $\sqrt{\lambda_1^{(2)}}$ . We get that

$$\begin{aligned} & \sigma_1^{(1)} + \sigma_1^{(2)} \\ &= \sqrt{\frac{(a_1 + c_1) + \sqrt{(a_1 - c_1)^2 + 4b_1^2}}{2}} + \sqrt{\frac{(a_1 + c_1) - \sqrt{(a_1 - c_1)^2 + 4b_1^2}}{2}} \\ &= \sqrt{\frac{(C_{A1} + nC_{A2} + C_{D1} + nC_{D2}) + \sqrt{(C_{A1} + nC_{A2} - (C_{D1} + nC_{D2}))^2 + 4(C_{B1} + nC_{B2})^2}}{2}} \\ & \quad + \sqrt{\frac{(C_{A1} + nC_{A2} + C_{D1} + nC_{D2}) - \sqrt{(C_{A1} + nC_{A2} - (C_{D1} + nC_{D2}))^2 + 4(C_{B1} + nC_{B2})^2}}{2}}. \end{aligned}$$

This corresponds to the first two lines of the expression for the nuclear norm in Lemma 21.

**Term 2: Sum of singular values from  $\mathbf{H}_2$ .**

The matrix  $\mathbf{H}_2$  is given by Equation (B.7):

$$\mathbf{H}_2 = \begin{pmatrix} C_{A1} & C_{B1} \\ C_{B1} & C_{D1} \end{pmatrix}.$$

Let  $a_2 = C_{A1}$ ,  $b_2 = C_{B1}$ , and  $c_2 = C_{D1}$ . The eigenvalues  $\lambda_2^{(1)}, \lambda_2^{(2)}$  of this symmetric  $2 \times 2$  matrix  $\mathbf{H}_2 = \begin{pmatrix} a_2 & b_2 \\ b_2 & c_2 \end{pmatrix}$  are:

$$\lambda = \frac{(a_2 + c_2) \pm \sqrt{(a_2 - c_2)^2 + 4b_2^2}}{2}.$$

The singular values  $\sigma_2^{(1)}$  and  $\sigma_2^{(2)}$  are  $\sqrt{\lambda_2^{(1)}}$  and  $\sqrt{\lambda_2^{(2)}}$ . Thus:

$$\begin{aligned} & \sigma_2^{(1)} + \sigma_2^{(2)} \\ &= \sqrt{\frac{(a_2 + c_2) + \sqrt{(a_2 - c_2)^2 + 4b_2^2}}{2}} + \sqrt{\frac{(a_2 + c_2) - \sqrt{(a_2 - c_2)^2 + 4b_2^2}}{2}} \\ &= \sqrt{\frac{(C_{A1} + C_{D1}) + \sqrt{(C_{A1} - C_{D1})^2 + 4C_{B1}^2}}{2}} + \sqrt{\frac{(C_{A1} + C_{D1}) - \sqrt{(C_{A1} - C_{D1})^2 + 4C_{B1}^2}}{2}}. \end{aligned}$$

This sum is then multiplied by the multiplicity  $(n - 1)$ :

$$(n - 1)(\sigma_2^{(1)} + \sigma_2^{(2)}) = (n - 1) \left( \sqrt{\frac{(C_{A1} + C_{D1}) + \sqrt{(C_{A1} - C_{D1})^2 + 4C_{B1}^2}}{2}} + \sqrt{\frac{(C_{A1} + C_{D1}) - \sqrt{(C_{A1} - C_{D1})^2 + 4C_{B1}^2}}{2}} \right).$$

This corresponds to the third and fourth lines of the expression for the nuclear norm in Lemma 21.

Combining these terms, the nuclear norm  $\|\mathbf{W}_{\text{OV}}^{\text{F}}\|_{\star}$  is precisely the expression given in Lemma 21. This completes the proof.  $\square$

**Lemma 22.** *Let the expression  $\|\mathbf{W}_{\text{OV}}^{\text{F}}\|_{\star}$  be defined as Equation (B.12). The closed-form minimum of  $\|\mathbf{W}_{\text{OV}}^{\text{F}}\|_{\star}$  is given by*

$$\min \|\mathbf{W}_{\text{OV}}^{\text{F}}\|_{\star} = \sqrt{n} + (n - 1) \times \begin{cases} (\sqrt{m_{\text{train}}} + \sqrt{m_{\text{test}}}) & \text{if } m_{\text{test}} \geq m_{\text{train}} \\ \sqrt{2(m_{\text{train}} + m_{\text{test}})} & \text{if } m_{\text{test}} < m_{\text{train}} \end{cases},$$

where the minimum is achieved at  $p_1^* = f_1^* = q_1^* = 1$ ,  $g_1^* = \sqrt{m_{\text{train}}/m_{\text{test}}}$ ,  $p_2^* = f_2^* = q_2^* = -1/n$ ,  $g_2^* = -\sqrt{m_{\text{train}}/m_{\text{test}}}/n$ ,  $\beta_1^* = \gamma_2^* = 1/2$ ,  $\gamma_1^* = \beta_2^* = -1/2$  if  $m_{\text{test}} \geq m_{\text{train}}$ ;  $p_1^* = f_1^* = q_1^* = g_1^* = 1$ ,  $p_2^* = f_2^* = q_2^* = g_2^* = -1/n$ ,  $\beta_1^* = \gamma_2^* = 1/2$ ,  $\gamma_1^* = \beta_2^* = -1/2$  if  $m_{\text{test}} < m_{\text{train}}$ .

*Proof of Lemma 22.* The overall expression  $\|\mathbf{W}_{\text{OV}}^{\text{F}}\|_{\star}$  given in Equation (B.12) is a sum of two main components. Let  $M_1$  be the sum of the terms in the first two lines, and  $M_2$  be the sum of the terms in the last line. Thus,  $\|\mathbf{W}_{\text{OV}}^{\text{F}}\|_{\star} = M_1 + M_2$ . Both  $M_1$  and  $M_2$  represent sums of square roots of eigenvalues of effective  $2 \times 2$  matrices.

**Part 1.** We first consider minimizing the last two terms of  $\|\mathbf{W}_{\text{OV}}^{\text{F}}\|_{\star}$ . Define  $M_2$  as

$$(n - 1) \left( \sqrt{\frac{(C_{A1} + C_{D1}) + \sqrt{(C_{A1} - C_{D1})^2 + 4C_{B1}^2}}{2}} + \sqrt{\frac{(C_{A1} + C_{D1}) - \sqrt{(C_{A1} - C_{D1})^2 + 4C_{B1}^2}}{2}} \right). \quad (\text{B.13})$$

For ease of reference, we state the formulas for coefficients  $C_{A1}, C_{D1}, C_{B1}$ :

$$C_{A1} = m_{\text{train}} p_1^2 + m_{\text{test}} f_1^2, \quad (\text{B.14})$$

$$C_{D1} = m_{\text{train}} q_1^2 + m_{\text{test}} g_1^2, \quad (\text{B.15})$$

$$C_{B1} = m_{\text{train}} p_1 q_1 + m_{\text{test}} f_1 g_1. \quad (\text{B.16})$$

The expression for  $M_2$  can be simplified. Let  $\lambda_2^{(1)}$  and  $\lambda_2^{(2)}$  be the two eigenvalues of the matrix  $\mathbf{H}_2$ . We note that  $\lambda_2^{(1)} + \lambda_2^{(2)} = C_{A1} + C_{D1}$  and  $\lambda_2^{(1)}\lambda_2^{(2)} = C_{A1}C_{D1} - C_{B1}^2$ . The term  $C_{A1}C_{D1} - C_{B1}^2$  can be calculated as

$$\begin{aligned} C_{A1}C_{D1} - C_{B1}^2 &= (m_{\text{train}}p_1^2 + m_{\text{test}}f_1^2)(m_{\text{train}}q_1^2 + m_{\text{test}}g_1^2) - (m_{\text{train}}p_1q_1 + m_{\text{test}}f_1g_1)^2 \\ &= m_{\text{train}}m_{\text{test}}(p_1^2q_1^2 - 2p_1q_1f_1g_1 + f_1^2g_1^2) = m_{\text{train}}m_{\text{test}}(p_1g_1 - f_1q_1)^2. \end{aligned}$$

Since  $m_{\text{train}}, m_{\text{test}} > 0$ , we have  $C_{A1}C_{D1} - C_{B1}^2 \geq 0$ , so the eigenvalues  $\lambda_1, \lambda_2$  are non-negative.

The sum  $\sqrt{\lambda_2^{(1)}} + \sqrt{\lambda_2^{(2)}}$  can be written as  $\sqrt{\left(\sqrt{\lambda_2^{(1)}} + \sqrt{\lambda_2^{(2)}}\right)^2} = \sqrt{\lambda_2^{(1)} + \lambda_2^{(2)} + 2\sqrt{\lambda_2^{(1)}\lambda_2^{(2)}}}$ .

Substituting the trace and determinant yields

$$\begin{aligned} \sqrt{\lambda_2^{(1)}} + \sqrt{\lambda_2^{(2)}} &= \sqrt{C_{A1} + C_{D1} + 2\sqrt{m_{\text{train}}m_{\text{test}}(p_1g_1 - f_1q_1)^2}} \\ &= \sqrt{C_{A1} + C_{D1} + 2\sqrt{m_{\text{train}}m_{\text{test}}}|p_1g_1 - f_1q_1|}. \end{aligned}$$

Let  $S = C_{A1} + C_{D1} + 2\sqrt{m_{\text{train}}m_{\text{test}}}|p_1g_1 - f_1q_1|$ . Substituting the definitions of  $C_{A1}$  and  $C_{D1}$ :

$$S = m_{\text{train}}p_1^2 + m_{\text{test}}f_1^2 + m_{\text{train}}q_1^2 + m_{\text{test}}g_1^2 + 2\sqrt{m_{\text{train}}m_{\text{test}}}|p_1g_1 - f_1q_1|. \quad (\text{B.17})$$

Then  $M_2 = (n-1)\sqrt{S}$ . To minimize  $\|W\|_*$ , we must minimize  $S$  with respect to  $p_1, q_1, f_1, g_1$  under the given constraints. Let  $K = p_1g_1 - f_1q_1$ . We analyze the minimization by considering two cases for  $K$ .

**Case 1:**  $K \leq 0$  (i.e.,  $p_1g_1 \leq f_1q_1$ ). Then

$$S = S_2 = (\sqrt{m_{\text{train}}}p_1 - \sqrt{m_{\text{test}}}g_1)^2 + (\sqrt{m_{\text{test}}}f_1 + \sqrt{m_{\text{train}}}q_1)^2.$$

The condition  $K \leq 0$  implies that  $g_1 \leq f_1q_1/p_1$ . The first term  $(\sqrt{m_{\text{train}}}p_1 - \sqrt{m_{\text{test}}}g_1)^2$  is monotonically decreasing when  $0 \leq g_1 \leq p_1\sqrt{m_{\text{train}}/m_{\text{test}}}$ , with the minimum attained at  $g_1 = p_1\sqrt{m_{\text{train}}/m_{\text{test}}}$ .

- If  $m_{\text{test}} > m_{\text{train}}$ ,  $g_1 = p_1\sqrt{m_{\text{train}}/m_{\text{test}}}$  is always achievable, so by taking  $f_1 = q_1 = 1$ , we get  $S_2 = (\sqrt{m_{\text{train}}} + \sqrt{m_{\text{test}}})^2$ . The equality holds for any  $p_1 \in [1, (m_{\text{test}}/m_{\text{train}})^{1/4}]$  and  $g_1 = p_1\sqrt{m_{\text{train}}/m_{\text{test}}}$ .
- If  $m_{\text{test}} \leq m_{\text{train}}$ ,  $g_1 = p_1\sqrt{m_{\text{train}}/m_{\text{test}}}$  may not be achievable. But since the first term is monotonically decreasing with respect to  $g_1$  when  $0 \leq g_1 \leq p_1\sqrt{m_{\text{train}}/m_{\text{test}}}$ , we get that the minimum is always taken with  $g_1 = \min\{f_1q_1/p_1, p_1\sqrt{m_{\text{train}}/m_{\text{test}}}\}$ .

If  $g_1 = f_1q_1/p_1$ , we get that

$$\begin{aligned} S_2 &= (\sqrt{m_{\text{train}}}p_1 - \sqrt{m_{\text{test}}}f_1q_1/p_1)^2 + (\sqrt{m_{\text{test}}}f_1 + \sqrt{m_{\text{train}}}q_1)^2 \\ &= m_{\text{train}}q_1^2 + m_{\text{test}}f_1^2 + m_{\text{train}}p_1^2 + m_{\text{test}}\frac{f_1^2q_1^2}{p_1^2} \\ &\geq 2(m_{\text{train}} + m_{\text{test}}), \end{aligned}$$

where the equality holds if and only if  $p_1 = g_1 = f_1 = q_1 = 1$ .

If  $g_1 = p_1 \sqrt{m_{\text{train}}/m_{\text{test}}}$ , we get  $f_1 q_1 \geq p_1^2 \sqrt{m_{\text{train}}/m_{\text{test}}} \geq \sqrt{m_{\text{train}}/m_{\text{test}}}$ . Therefore,

$$\begin{aligned} S_2 &= (\sqrt{m_{\text{test}}} f_1 + \sqrt{m_{\text{train}}} q_1)^2 \\ &\geq 4\sqrt{m_{\text{test}}} \sqrt{m_{\text{train}}} f_1 q_1 \\ &\geq 4m_{\text{train}}, \end{aligned}$$

where the equality holds if and only if  $p_1 = q_1 = 1$ ,  $g_1 = f_1 = \sqrt{m_{\text{train}}/m_{\text{test}}}$ .

Therefore, we can conclude that if  $m_{\text{test}} \leq m_{\text{train}}$ ,  $S_2 \geq 2(m_{\text{train}} + m_{\text{test}})$ , with the equality holds if and only if  $p_1 = g_1 = f_1 = q_1 = 1$ .

**Case 2:** Next consider  $K \geq 0$ , that is,  $p_1 g_1 \geq f_1 q_1$ . Then

$$S = S_1 = (\sqrt{m_{\text{train}}} p_1 + \sqrt{m_{\text{test}}} g_1)^2 + (\sqrt{m_{\text{test}}} f_1 - \sqrt{m_{\text{train}}} q_1)^2.$$

Taking  $g_1 \geq f_1 q_1 / p_1$  into  $S_1$ ,

$$S_1 \geq m_{\text{test}} f_1^2 + m_{\text{train}} q_1^2 + m_{\text{train}} p_1^2 + m_{\text{test}} \frac{f_1^2 q_1^2}{p_1^2} \geq m_{\text{test}} + m_{\text{train}} + m_{\text{train}} p_1^2 + \frac{m_{\text{test}}}{p_1^2}.$$

- If  $m_{\text{test}} \geq m_{\text{train}}$ , we have that  $S_1 \geq (\sqrt{m_{\text{test}}} + \sqrt{m_{\text{train}}})^2$ , with equality holds when  $p_1 = [m_{\text{test}}/m_{\text{train}}]^{1/4}$  and  $g_1 = 1/p_1$ .
- If  $m_{\text{test}} < m_{\text{train}}$ , the minimum is achieved by  $p_1 = 1$ . We therefore get that  $g_1 = 1$ , with  $S_1 \geq 2(m_{\text{test}} + m_{\text{train}})$ .

Consolidating the minima, we find that if  $m_{\text{test}} \geq m_{\text{train}}$ , the minimum  $S$  is  $(\sqrt{m_{\text{train}}} + \sqrt{m_{\text{test}}})^2$ , achieved when  $1 \leq p_1 \leq (m_{\text{test}}/m_{\text{train}})^{1/4}$ ,  $f_1 = 1$ ,  $q_1 = 1$ , and  $g_1 = p_1 \sqrt{m_{\text{train}}/m_{\text{test}}}$ . If  $m_{\text{test}} < m_{\text{train}}$ , the minimum  $S$  is  $2(m_{\text{train}} + m_{\text{test}})$ , achieved when  $p_1 = 1$ ,  $f_1 = 1$ ,  $q_1 = 1$ ,  $g_1 = 1$ . The minimum value for  $\sqrt{S}$  is therefore  $\sqrt{m_{\text{train}}} + \sqrt{m_{\text{test}}}$  if  $m_{\text{test}} \geq m_{\text{train}}$ , and  $\sqrt{2(m_{\text{train}} + m_{\text{test}})}$  if  $m_{\text{test}} < m_{\text{train}}$ . Multiplying by  $(n-1)$  gives the final result for  $\min M_2$ .

**Part 2.** We seek to find the minimum value of the quantity  $M_1$  given by

$$\begin{aligned} M_1 &= \sqrt{\frac{(C_{A1} + nC_{A2} + C_{D1} + nC_{D2}) + \sqrt{(C_{A1} + nC_{A2} - (C_{D1} + nC_{D2}))^2 + 4(C_{B1} + nC_{B2})^2}}{2}} \\ &\quad + \sqrt{\frac{(C_{A1} + nC_{A2} + C_{D1} + nC_{D2}) - \sqrt{(C_{A1} + nC_{A2} - (C_{D1} + nC_{D2}))^2 + 4(C_{B1} + nC_{B2})^2}}{2}}. \end{aligned}$$

where the coefficients are defined as

$$\begin{aligned} C_{A1} &= m_{\text{train}} p_1^2 + m_{\text{test}} f_1^2, \\ C_{A2} &= m_{\text{train}} (2p_1 p_2 + n p_2^2) + m_{\text{test}} (2f_1 f_2 + n f_2^2) + \beta_1^2 + \beta_2^2, \\ C_{D1} &= m_{\text{train}} q_1^2 + m_{\text{test}} g_1^2, \\ C_{D2} &= m_{\text{train}} (2q_1 q_2 + n q_2^2) + m_{\text{test}} (2g_1 g_2 + n g_2^2) + \gamma_1^2 + \gamma_2^2, \\ C_{B1} &= m_{\text{train}} p_1 q_1 + m_{\text{test}} f_1 g_1, \\ C_{B2} &= m_{\text{train}} (p_1 q_2 + p_2 q_1 + n p_2 q_2) + m_{\text{test}} (f_1 g_2 + f_2 g_1 + n f_2 g_2) + \beta_1 \gamma_1 + \beta_2 \gamma_2. \end{aligned}$$

subject to the constraints

$$p_1, f_1, q_1 \geq 1, \quad (\text{B.18})$$

$$p_1 + p_2 + \beta_1 \geq q_1 + q_2 + \gamma_1 + 1, \quad (\text{B.19})$$

$$q_1 + q_2 + \gamma_2 \geq p_1 + p_2 + \beta_2 + 1, \quad (\text{B.20})$$

$$f_1 + f_2 + \beta_1 \geq \max(g_1, 0) + g_2 + \gamma_1 + 1. \quad (\text{B.21})$$

Let  $A = C_{A1} + nC_{A2}$ ,  $D = C_{D1} + nC_{D2}$ , and  $B = C_{B1} + nC_{B2}$ . The expression for  $M_1$  can be simplified to

$$M_1 = \sqrt{A + D + 2\sqrt{AD} - B^2}.$$

Note that

$$\begin{aligned} M_1^2 &\geq A + D \\ &= m_{\text{train}}(p_1 + np_2)^2 + m_{\text{train}}(q_1 + nq_2)^2 + m_{\text{test}}(f_1 + nf_2)^2 + m_{\text{test}}(g_1 + ng_2)^2 \\ &\quad + n(\beta_1^2 + \beta_2^2 + \gamma_1^2 + \gamma_2^2). \end{aligned}$$

Let  $\Delta = q_1 + q_2 - p_1 - p_2$ . From the constraints, we can get that

$$\begin{aligned} \beta_1 - \gamma_1 &\geq 1 + \Delta, \\ \gamma_2 - \beta_2 &\geq 1 - \Delta. \end{aligned}$$

If  $|\Delta| > 1$ , without loss of generality, assume  $\Delta > 1$ . This gives that

$$(\beta_1^2 + \gamma_1^2) + (\beta_2^2 + \gamma_2^2) \geq 2 \left( \frac{1 + \Delta}{2} \right)^2 + 0 \geq 2.$$

If  $|\Delta| \leq 1$ , we get that

$$\begin{aligned} (\beta_1^2 + \gamma_1^2) + (\beta_2^2 + \gamma_2^2) &\geq 2 \left( \frac{1 + \Delta}{2} \right)^2 + 2 \left( \frac{1 - \Delta}{2} \right)^2 \\ &\geq 4 \left( \frac{1}{4} + \frac{1}{4} \Delta^2 \right) \\ &\geq 1. \end{aligned}$$

As a result, we get  $M_1 \geq \sqrt{n}$ . Note that the equality holds if and only if

$$\begin{aligned} p_2 &= -p_1/n, \quad f_2 = -f_1/n, \quad q_2 = -q_1/n, \quad g_2 = -g_1/n, \\ \beta_1 &= 1/2, \quad \gamma_1 = -1/2, \quad \beta_2 = -1/2, \quad \gamma_2 = 1/2. \end{aligned}$$

Combining this condition with the results in part 1, we finish the proof for Lemma 22.  $\square$

We would adopt Assumption 4 to prove the uniqueness, which assumes the solution either takes a symmetric form as (B.3), or it is nondegenerate.

**Assumption 4.** Let  $\mathbf{W}_{\text{OV}}^{\text{F}\star}$  be a solution to  $(\mathbf{W}_{\text{OV}}^{\text{F}}\text{-SVM})$ . Then it either takes the form of (B.3), or we have  $\mathbf{1}_{2n}^\top \mathbf{W}_{\text{OV}}^{\text{F}\star} \neq \mathbf{0}_{nm+2}^\top$ .

Lemma 23 can give a lower bound of the nuclear norm for a given matrix  $\bar{\mathbf{M}}$ .

**Lemma 23.** Suppose that  $\bar{\mathbf{M}} \in \mathbb{R}^{m \times n}$ , for any orthonormal matrix  $\mathbf{U} \in \mathbb{R}^{m \times r}$  and  $\mathbf{V} \in \mathbb{R}^{n \times r}$ , we have that

$$\|\bar{\mathbf{M}}\|_* \geq \text{Trace}(\mathbf{U}^\top \bar{\mathbf{M}} \mathbf{V}).$$

Moreover, if there exists  $\boldsymbol{\eta} \in \mathbb{R}^m$  s.t.  $\boldsymbol{\eta}^\top \mathbf{U} = 0$  and  $\boldsymbol{\eta}^\top \bar{\mathbf{M}} \neq 0$ , the above inequality is strict, i.e.,

$$\|\bar{\mathbf{M}}\|_* > \text{Trace}(\mathbf{U}^\top \bar{\mathbf{M}} \mathbf{V}).$$

*Proof of Lemma 23.* For any pair of matrices  $\mathbf{A}$  and  $\mathbf{B}$  such that  $\mathbf{AB} = \bar{\mathbf{M}}$ , we have that

$$\begin{aligned} \text{Trace}(\mathbf{U}^\top \bar{\mathbf{M}} \mathbf{V}) &= \text{Trace}(\mathbf{U}^\top \mathbf{ABV}) \\ &\leq \left( \|\mathbf{U}^\top \mathbf{A}\|_{\text{F}}^2 + \|\mathbf{BV}\|_{\text{F}}^2 \right) / 2 \\ &\leq \left( \|\mathbf{U}^\top\|_2^2 \|\mathbf{A}\|_{\text{F}}^2 + \|\mathbf{V}\|_2^2 \|\mathbf{B}\|_{\text{F}}^2 \right) / 2 \\ &\leq \left( \|\mathbf{A}\|_{\text{F}}^2 + \|\mathbf{B}\|_{\text{F}}^2 \right) / 2 \\ &\leq \|\bar{\mathbf{M}}\|_*. \end{aligned}$$

Consider the conditions for taking the equality. We should require  $\|\mathbf{U}^\top \mathbf{A}\|_{\text{F}} = \|\mathbf{U}^\top\|_2 \|\mathbf{A}\|_{\text{F}}$ . This means that all columns of  $\mathbf{A}$  are left-singular vectors of  $\mathbf{U}^\top$  with the largest singular value. In addition, it implies that the column space  $\text{Col}(\mathbf{A}) \subseteq \text{Col}(\mathbf{U})$ . But if there exists  $\boldsymbol{\eta} \in \mathbb{R}^m$  s.t.  $\boldsymbol{\eta}^\top \mathbf{U} = 0$  and  $\boldsymbol{\eta}^\top \bar{\mathbf{M}} \neq 0$ , we get  $\text{Col}(\mathbf{A}) \not\subseteq \text{Col}(\mathbf{U})$ . It implies the conditions for taking the equality cannot be satisfied, so

$$\|\bar{\mathbf{M}}\|_* > \text{Trace}(\mathbf{U}^\top \bar{\mathbf{M}} \mathbf{V}).$$

This finishes Lemma 23. □

**Theorem 16** (Uniqueness of the solution to the optimization problem  $(\mathbf{W}_{\text{OV}}^{\text{F}}\text{-SVM})$ ). Given Assumption 4, suppose that  $\mathbf{W}_{\text{OV}}^{\text{F}}$  is a solution to optimization problem  $(\mathbf{W}_{\text{OV}}^{\text{F}}\text{-SVM})$ , if  $m_{\text{test}} < m_{\text{train}}$ ,

$$\mathbf{W}_{\text{OV}}^{\text{F}} = \begin{bmatrix} \underbrace{\mathbf{I}_n - \mathbf{E}_n/n \cdots \mathbf{I}_n - \mathbf{E}_n/n}_{m_{\text{train}} \text{ blocks}} & \underbrace{\mathbf{I}_n - \mathbf{E}_n/n \cdots \mathbf{I}_n - \mathbf{E}_n/n}_{m_{\text{test}} \text{ blocks}} & \mathbf{1}_n/2 & -\mathbf{1}_n/2 \\ \underbrace{\mathbf{I}_n - \mathbf{E}_n/n \cdots \mathbf{I}_n - \mathbf{E}_n/n}_{m_{\text{train}} \text{ blocks}} & \underbrace{\mathbf{I}_n - \mathbf{E}_n/n \cdots \mathbf{I}_n - \mathbf{E}_n/n}_{m_{\text{test}} \text{ blocks}} & -\mathbf{1}_n/2 & \mathbf{1}_n/2 \end{bmatrix}. \quad (\text{B.22})$$

If  $m_{\text{test}} \geq m_{\text{train}}$ , let  $\rho = \sqrt{m_{\text{train}}/m_{\text{test}}}$ , we have that

$$\mathbf{W}_{\text{OV}}^{\text{F}} = \begin{bmatrix} \underbrace{\mathbf{I}_n - \mathbf{E}_n/n \cdots \mathbf{I}_n - \mathbf{E}_n/n}_{m_{\text{train}} \text{ blocks}} & \underbrace{\mathbf{I}_n - \mathbf{E}_n/n \cdots \mathbf{I}_n - \mathbf{E}_n/n}_{m_{\text{test}} \text{ blocks}} & \mathbf{1}_n/2 & -\mathbf{1}_n/2 \\ \underbrace{\mathbf{I}_n - \mathbf{E}_n/n \cdots \mathbf{I}_n - \mathbf{E}_n/n}_{m_{\text{train}} \text{ blocks}} & \underbrace{\rho \mathbf{I}_n - \rho \mathbf{E}_n/n \cdots \rho \mathbf{I}_n - \rho \mathbf{E}_n/n}_{m_{\text{test}} \text{ blocks}} & -\mathbf{1}_n/2 & \mathbf{1}_n/2 \end{bmatrix}. \quad (\text{B.23})$$

*Proof of Theorem 16.* Firstly, combining Lemmas 19 and 22, we get Equation (B.22) and (B.23). This shows that Equation (B.22) and (B.23) are one of the solutions to the optimization problem ( $\mathbf{W}_{\text{OV}}^{\text{F}}\text{-SVM}$ ).

Suppose that there exists an asymmetric solution of the SVM problem Equation ( $\mathbf{W}_{\text{OV}}^{\text{F}}\text{-SVM}$ )  $\mathbf{W}_{\text{OV}}^{\text{F}\star}$  that takes a different form from (B.3). Lemma 24 proves the existence of another asymmetric solution such that the difference takes the reversed sign.

**Lemma 24** (The solution with the difference taking the reverse sign). *Suppose that the  $\mathbf{W}_{\text{OV}}^{\text{F}\star}$  defined above is an asymmetric solution. Then there exists another asymmetric solution  $\widetilde{\mathbf{W}}_{\text{OV}}^{\text{F}\star}$  and  $\epsilon \in (0, 1)$  such that*

$$\widetilde{\mathbf{W}}_{\text{OV}}^{\text{F}\star} - \mathbf{W}_{\text{OV}}^{\text{F}} = -\epsilon (\mathbf{W}_{\text{OV}}^{\text{F}\star} - \mathbf{W}_{\text{OV}}^{\text{F}}).$$

*Proof.* Let  $\tilde{\varphi}$  be all possible permutations on  $\mathbf{W}_{\text{OV}}$  as described in Lemma 19, excluding the identity permutation. We get that

$$\frac{\sum_{\tilde{\varphi}} \tilde{\varphi}(\mathbf{W}_{\text{OV}}^{\text{F}\star}) + \mathbf{W}_{\text{OV}}^{\text{F}\star}}{1 + \sum_{\tilde{\varphi}} 1} = \mathbf{W}_{\text{OV}}^{\text{F}}.$$

So we can take

$$\widetilde{\mathbf{W}}_{\text{OV}}^{\text{F}\star} = \frac{\sum_{\tilde{\varphi}} \tilde{\varphi}(\mathbf{W}_{\text{OV}}^{\text{F}\star})}{\sum_{\tilde{\varphi}} 1}$$

and get that

$$\widetilde{\mathbf{W}}_{\text{OV}}^{\text{F}\star} - \mathbf{W}_{\text{OV}}^{\text{F}} = \frac{\mathbf{W}_{\text{OV}}^{\text{F}} - \mathbf{W}_{\text{OV}}^{\text{F}\star}}{\sum_{\tilde{\varphi}} 1}.$$

This finishes the proof for Lemma 24.  $\square$

Assumption 4 implies that  $\mathbf{1}_{2n}^\top (\mathbf{W}_{\text{OV}}^{\text{F}\star} - \mathbf{W}_{\text{OV}}^{\text{F}}) \neq \mathbf{0}_{nm+2}^\top$ . We choose  $\mathbf{U} \in \mathbb{R}^{2n \times r}$  and  $\mathbf{V} \in \mathbb{R}^{(nm+2) \times r}$  so that they are the SVD decompositions corresponding to  $r$  positive singular values for the symmetric solution  $\mathbf{W}_{\text{OV}}^{\text{F}}$ . Since  $\mathbf{1}_{2n}^\top \mathbf{W}_{\text{OV}}^{\text{F}} = \mathbf{0}_{nm+2}^\top$ , it also holds that  $\mathbf{1}_{2n}^\top \mathbf{U} = \mathbf{0}_{nm+2}^\top$ . Using Lemma 23, we get

$$\begin{aligned} \|\mathbf{W}_{\text{OV}}^{\text{F}\star}\|_* &> \text{Trace}(\mathbf{U}^\top \mathbf{W}_{\text{OV}}^{\text{F}\star} \mathbf{V}) \\ &= \text{Trace}(\mathbf{U}^\top \mathbf{W}_{\text{OV}}^{\text{F}} \mathbf{V}) + \text{Trace}(\mathbf{U}^\top (\mathbf{W}_{\text{OV}}^{\text{F}\star} - \mathbf{W}_{\text{OV}}^{\text{F}}) \mathbf{V}) \\ &= \|\mathbf{W}_{\text{OV}}^{\text{F}}\|_* + \text{Trace}(\mathbf{U}^\top (\mathbf{W}_{\text{OV}}^{\text{F}\star} - \mathbf{W}_{\text{OV}}^{\text{F}}) \mathbf{V}). \end{aligned}$$

Using Lemma 24, there exists another asymmetric solution  $\widetilde{\mathbf{W}}_{\text{OV}}$  such that

$$\begin{aligned}\|\widetilde{\mathbf{W}}_{\text{OV}}\|_* &> \text{Trace}(\mathbf{U}^\top \mathbf{W}_{\text{OV}}^{\text{F}} \mathbf{V}) + \text{Trace}(\mathbf{U}^\top (\widetilde{\mathbf{W}}_{\text{OV}} - \mathbf{W}_{\text{OV}}^{\text{F}}) \mathbf{V}) \\ &= \|\mathbf{W}_{\text{OV}}^{\text{F}}\|_* - \epsilon \text{Trace}(\mathbf{U}^\top (\mathbf{W}_{\text{OV}}^{\text{F}*} - \mathbf{W}_{\text{OV}}^{\text{F}}) \mathbf{V}).\end{aligned}$$

Therefore, we have that

$$\begin{aligned}&\max\{\|\mathbf{W}_{\text{OV}}^{\text{F}*}\|_*, \|\widetilde{\mathbf{W}}_{\text{OV}}\|_*\} \\ &> \|\mathbf{W}_{\text{OV}}^{\text{F}}\|_* + \max\{\text{Trace}(\mathbf{U}^\top (\mathbf{W}_{\text{OV}}^{\text{F}*} - \mathbf{W}_{\text{OV}}^{\text{F}}) \mathbf{V}), -\epsilon \text{Trace}(\mathbf{U}^\top (\mathbf{W}_{\text{OV}}^{\text{F}*} - \mathbf{W}_{\text{OV}}^{\text{F}}) \mathbf{V})\} \\ &\geq \|\mathbf{W}_{\text{OV}}^{\text{F}}\|_*.\end{aligned}$$

This leads to a contradiction that both  $\mathbf{W}_{\text{OV}}^{\text{F}*}$  and  $\widetilde{\mathbf{W}}_{\text{OV}}$  are solutions to Equation ( $\mathbf{W}_{\text{OV}}^{\text{F}}$ -SVM) that minimize the nuclear norm. This confirms that all solutions would take the form Equation (B.3), which would be Equations (B.22) and (B.23).  $\square$

### B.1.3.2 Proof for non-factorized model

Similar to Section B.1.3.1, we only need to consider a reduced matrix  $\mathbf{W}_{\text{OV}} \in \mathbb{R}^{(2n) \times (nm+2)}$  throughout this section, where each row corresponds to a token in  $\mathcal{A}$  and each column corresponds to a token in  $\mathcal{S} \cup \mathcal{R}$ . Now We restate the second part of Theorem 6 below.

**Theorem 17** (Part 2 in Theorem 6: Non-factorized model has no OCR ability). *Let  $n > 1$ . Suppose  $\mathbf{W}_{\text{OV}}$  is a solution to the SVM problem in ( $\mathbf{W}_{\text{OV}}$ -SVM). For any  $(s, r) \in \mathcal{D}_{\text{test}}$  and any  $a' \in \mathcal{A}_2 \setminus \{a^*(s, r)\}$ , it holds that*

$$h_{(s,r),a'}(\mathbf{W}_{\text{OV}}) = 0, \text{ indicating no OCR ability.} \quad (\text{B.24})$$

The proof of Theorem 17 follows the same idea as Theorem 15. The roadmap of the proof is as follows:

1. **A unique solution of a similar form to Lemma 19:** Lemma 25 shows both the existence and uniqueness of the solution to ( $\mathbf{W}_{\text{OV}}$ -SVM).
2. **Frobenius norm formula of the restricted form:** Using the same SVD decomposition as in Lemma 20, Lemma 26 gives  $\|\mathbf{W}_{\text{OV}}\|_{\text{F}}$  in closed form (B.27).
3. **Optimization:** Lemma 27 finds the minimum of (B.27).
4. **Solution characterization:** Theorem 18 gives the form of the solution to ( $\mathbf{W}_{\text{OV}}$ -SVM) and finishes the proof for Theorem 17.

**Lemma 25** (Existence and uniqueness of a restricted form solution to ( $\mathbf{W}_{\text{OV-SVM}}$ )). *Suppose  $\mathbf{W}_{\text{OV}}$  is the solution to the optimization problem ( $\mathbf{W}_{\text{OV-SVM}}$ ). The solution must take the form in Equation (B.25)*

$$\mathbf{W}_{\text{OV}} = \begin{bmatrix} \underbrace{p_1 \mathbf{I}_n + p_2 \mathbf{E}_n \cdots p_1 \mathbf{I}_n + p_2 \mathbf{E}_n}_{m_{\text{train}} \text{ blocks}} & \underbrace{f_1 \mathbf{I}_n + f_2 \mathbf{E}_n \cdots f_1 \mathbf{I}_n + f_2 \mathbf{E}_n}_{m_{\text{test}} \text{ blocks}} & \beta_1 \mathbf{1}_n & \beta_2 \mathbf{1}_n \\ \underbrace{q_1 \mathbf{I}_n + q_2 \mathbf{E}_n \cdots q_1 \mathbf{I}_n + q_2 \mathbf{E}_n}_{m_{\text{train}} \text{ blocks}} & \underbrace{g_1 \mathbf{I}_n + g_2 \mathbf{E}_n \cdots g_1 \mathbf{I}_n + g_2 \mathbf{E}_n}_{m_{\text{test}} \text{ blocks}} & \gamma_1 \mathbf{1}_n & \gamma_2 \mathbf{1}_n \end{bmatrix}. \quad (\text{B.25})$$

with  $p_1, p_2, q_1, q_2, f_1, f_2$ , and  $g_1, g_2$  satisfying

$$\begin{aligned} p_1, f_1, q_1 &\geq 1, \\ p_1 + p_2 + \beta_1 &\geq q_1 + q_2 + \gamma_1 + 1, \\ q_1 + q_2 + \gamma_2 &\geq p_1 + p_2 + \beta_2 + 1, \\ f_1 + f_2 + \beta_1 &\geq (g_1 \vee 0) + g_2 + \gamma_1 + 1. \end{aligned} \quad (\text{B.26})$$

*Proof of Lemma 25.* The existence proof is the same as the proof for Lemma 19. Since the Frobenius norm is strongly convex, the solution is unique.  $\square$

**Lemma 26.** *The  $\mathbf{W}_{\text{OV}}$  in restricted form Equation (B.25) has the Frobenius norm.*

$$\|\mathbf{W}_{\text{OV}}\|_F = \left( n \begin{bmatrix} m_{\text{train}}((p_1 + p_2)^2 + (n-1)p_2^2 + (q_1 + q_2)^2 + (n-1)q_2^2) \\ + m_{\text{test}}((f_1 + f_2)^2 + (n-1)f_2^2 + (g_1 + g_2)^2 + (n-1)g_2^2) \\ + \beta_1^2 + \gamma_2^2 + \beta_2^2 + \gamma_1^2 \end{bmatrix} \right)^{1/2}. \quad (\text{B.27})$$

*Proof of Lemma 26.* The square of the Frobenius norm is the sum of the squares of these  $2n$  singular values:

$$\|\mathbf{W}_{\text{OV}}^F\|_F^2 = (\sigma_1^{(1)})^2 + (\sigma_1^{(2)})^2 + (n-1)(\sigma_2^{(1)})^2 + (n-1)(\sigma_2^{(2)})^2$$

From Lemma 20, we know that  $(\sigma_1^{(k)})^2 = \lambda_1^{(k)}$  (eigenvalues of  $\mathbf{H}_1$ ) and  $(\sigma_2^{(k')})^2 = \lambda_2^{(k')}$  (eigenvalues of  $\mathbf{H}_2$ ). So,

$$\|\mathbf{W}_{\text{OV}}^F\|_F^2 = \lambda_1^{(1)} + \lambda_1^{(2)} + (n-1)(\lambda_2^{(1)} + \lambda_2^{(2)})$$

Since the sum of eigenvalues of a matrix is its trace, we have  $\lambda_1^{(1)} + \lambda_1^{(2)} = \text{Tr}(\mathbf{H}_1)$ ,  $\lambda_2^{(1)} + \lambda_2^{(2)} = \text{Tr}(\mathbf{H}_2)$ . Therefore,

$$\begin{aligned} \|\mathbf{W}_{\text{OV}}^F\|_F^2 &= \text{Tr}(\mathbf{H}_1) + (n-1)\text{Tr}(\mathbf{H}_2) \\ &= n(C_{A1} + C_{A2} + C_{D1} + C_{D2}) \\ &= n \begin{bmatrix} m_{\text{train}}((p_1 + p_2)^2 + (n-1)p_2^2 + (q_1 + q_2)^2 + (n-1)q_2^2) \\ + m_{\text{test}}((f_1 + f_2)^2 + (n-1)f_2^2 + (g_1 + g_2)^2 + (n-1)g_2^2) \\ + \beta_1^2 + \gamma_2^2 + \beta_2^2 + \gamma_1^2 \end{bmatrix}. \end{aligned}$$

This proves Lemma 26.  $\square$

**Lemma 27.** *Let the expression  $\|\mathbf{W}_{\text{OV}}\|_F$  be defined as Equation (B.27). The closed-form minimum of  $\|\mathbf{W}_{\text{OV}}\|_F$  is given by*

$$\min \|\mathbf{W}_{\text{OV}}\|_F = \sqrt{n} (2m_{\text{train}}(1 - 1/n) + m_{\text{test}}(1 - 1/n) + 1)^{1/2},$$

where the minimum is achieved at  $p_1^* = f_1^* = q_1^* = 1, p_2^* = f_2^* = q_2^* = -1/n, \beta_1^* = \gamma_2^* = 1/2, \gamma_1^* = \beta_2^* = -1/2$ , and  $g_1^* = g_2^* = 0$ .

*Proof of Lemma 27.* Given the constraints in Equation (B.26), we have that

$$\|\mathbf{W}_{\text{OV}}\|_F \geq \left( n \left[ \begin{array}{c} m_{\text{train}}(np_2^2 + 2p_2 + 1 + nq_2^2 + 2q_2 + 1) \\ + m_{\text{test}}(nf_2^2 + 2f_2 + 1 + (g_1 + g_2)^2 + (n-1)g_2^2) + 1 \end{array} \right] \right)^{1/2},$$

where the equality holds when  $p_1^* = f_1^* = q_1^* = 1$  and  $\beta_1^* = \gamma_2^* = 1/2, \gamma_1^* = \beta_2^* = -1/2$ . This lower bound is a quadratic form, which we can show that  $p_2^* = f_2^* = q_2^* = -1/n$  and  $g_1^* = g_2^* = 0$  gives its minimum. This finishes the proof of Lemma 27.  $\square$

**Theorem 18.** *Therefore, we have that*

$$\mathbf{W}_{\text{OV}} = \begin{bmatrix} \underbrace{\mathbf{I}_n - \mathbf{E}_n/n \cdots \mathbf{I}_n - \mathbf{E}_n/n}_{m_{\text{train}} \text{ blocks}} & \underbrace{\mathbf{I}_n - \mathbf{E}_n/n \cdots \mathbf{I}_n - \mathbf{E}_n/n}_{m_{\text{test}} \text{ blocks}} & \mathbf{1}_n/2 & -\mathbf{1}_n/2 \\ \underbrace{\mathbf{I}_n - \mathbf{E}_n/n \cdots \mathbf{I}_n - \mathbf{E}_n/n}_{m_{\text{train}} \text{ blocks}} & \underbrace{\mathbf{0} \cdots \mathbf{0}}_{m_{\text{test}} \text{ blocks}} & -\mathbf{1}_n/2 & \mathbf{1}_n/2 \end{bmatrix}. \quad (\text{B.28})$$

It implies that for any  $s \in \mathcal{S}_{\text{test}}$  and any  $a' \in \mathcal{A}_2 \setminus \{a^*(s, r_2)\}$ , it holds that  $h_{(s, r_2), a'} = 0$ . This proves Theorem 17.

*Proof of Theorem 18.* We take the results from Lemma 27 into the restricted form Equation (B.25) and get Equation (B.28). Given any  $s_{i,j} \in \mathcal{S}_{\text{test}}$  and  $c_k \in \mathcal{A}_2$  where  $k \neq i$ , we have

$$h_{(s_{i,j}, r_2), c_k} = [\mathbf{0}_n^\top, \mathbf{e}_i^\top - \mathbf{e}_k^\top] \mathbf{W}_{\text{OV}} \cdot \begin{bmatrix} 0 \\ \vdots \\ \mathbf{e}_i \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} = 0.$$

This proves Theorem 18.  $\square$

## B.2 Proof of Section 3.4.2

In this section, we provide the complete proof of Theorem 7 via gradient flow analysis. We first present several key lemmas to prove Theorem 7. In Lemma 28, we give the gradient

form of the reparameterized parameters to simplify the analysis. In Lemma 29, we prove the Lipschitzness of the gradient flow, and Lemma 30 proves that the gradient flow also satisfies permutation equivariance. Using these two lemmas, we are able to conclude the symmetry in parameters by employing a standard argument for the uniqueness of ODE solutions in Lemma 31.

**Useful notations.** We introduce useful notations for this section. Let  $\text{vec}(\mathbf{A})$  denote the vectorization of a matrix  $\mathbf{A}$ . Specifically, for  $\mathbf{A} = (a_{ij})_{i=1,j=1}^{m,n} \in \mathbb{R}^{m \times n}$ , we have  $\text{vec}(\mathbf{A}) = (a_{11}, \dots, a_{1n}, a_{21}, \dots, a_{2n}, \dots, a_{m1}, \dots, a_{mn})^\top \in \mathbb{R}^{mn}$ . We use  $\|\mathbf{A}\|_{\text{op}}$  to denote the operator norm of a matrix  $\mathbf{A}$ . For any vector  $\mathbf{u} \in \mathbb{R}^n$ , let  $\text{diag}(\mathbf{u}) \in \mathbb{R}^{n \times n}$  denote a diagonal matrix whose diagonal entries are corresponding entries in  $\mathbf{u}$ . Let  $\mathbb{1}(\mathcal{E})$  denote the indicator function of an event  $\mathcal{E}$ .

**Lemma 28.** Recall that for any  $z \in \mathcal{V}$ , we have  $W_{\text{OV}}(a, z) = \mathbf{e}_a^\top \mathbf{W}_{\text{OV}} \mathbf{e}_z$  and  $W_{\text{KQ}}(z) = \mathbf{e}_z^\top \mathbf{W}_{\text{KQ}} \mathbf{e}_{\langle \text{EOS} \rangle}$ . Then for a fixed  $(a, s) \in \mathcal{A} \times \mathcal{S}$ , the gradient of  $W_{\text{OV}}(a, s)$  is given by:

$$\partial_{W_{\text{OV}}(a,s)} \mathcal{L}(\tilde{\boldsymbol{\theta}}) = -W_{\text{KQ}}(s) \cdot \sum_{r \in \mathcal{R}} p(s, r) \cdot (\mathbb{1}(a = a^*(s, r)) - p_{\tilde{\boldsymbol{\theta}}}(a|s, r)).$$

Similarly, for a fixed  $(a, r) \in \mathcal{A} \times \mathcal{R}$ , we have:

$$\partial_{W_{\text{OV}}(a,r)} \mathcal{L}(\tilde{\boldsymbol{\theta}}) = -W_{\text{KQ}}(r) \cdot \sum_{s \in \mathcal{S}} p(s, r) \cdot (\mathbb{1}(a = a^*(s, r)) - p_{\tilde{\boldsymbol{\theta}}}(a|s, r)).$$

Moreover, for  $\langle \text{EOS} \rangle$  token and a fixed  $a \in \mathcal{A}$ , we have:

$$\partial_{W_{\text{OV}}(a, \langle \text{EOS} \rangle)} \mathcal{L}(\tilde{\boldsymbol{\theta}}) = -W_{\text{KQ}}(\langle \text{EOS} \rangle) \cdot \sum_{s \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s, r) \cdot (\mathbb{1}(a = a^*(s, r)) - p_{\tilde{\boldsymbol{\theta}}}(a|s, r)).$$

Lastly, for  $s \in \mathcal{S}$ , we have:

$$\partial_{W_{\text{KQ}}(s)} \mathcal{L}(\tilde{\boldsymbol{\theta}}) = - \sum_{a \in \mathcal{A}} W_{\text{OV}}(a, s) \sum_{r \in \mathcal{R}} p(s, r) \cdot (\mathbb{1}(a = a^*(s, r)) - p_{\tilde{\boldsymbol{\theta}}}(a|s, r)).$$

*Proof.* First, note that the logit function can be written as:

$$f_{\tilde{\boldsymbol{\theta}}}(z_{1:T}, a) = \mathbf{e}_a^\top \mathbf{W}_{\text{OV}} \mathbf{X}^\top \mathbf{X} \mathbf{W}_{\text{KQ}} \mathbf{x}_T = \sum_{t \in [T]} W_{\text{OV}}(a, z_t) W_{\text{KQ}}(z_t). \quad (\text{B.29})$$

Then for any  $z \in \mathcal{V}$ , we get

$$\begin{aligned} \partial_{W_{\text{OV}}(a,z)} (\mathbf{e}_a^\top \mathbf{W}_{\text{OV}} \mathbf{X}^\top \mathbf{X} \mathbf{W}_{\text{KQ}} \mathbf{x}_T) &= \mathbb{1}(a = a') C(z_{1:T}, z) W_{\text{KQ}}(z), \\ \partial_{W_{\text{KQ}}(z)} (\mathbf{e}_a^\top \mathbf{W}_{\text{OV}} \mathbf{X}^\top \mathbf{X} \mathbf{W}_{\text{KQ}} \mathbf{x}_T) &= C(z_{1:T}, z) W_{\text{OV}}(a', z), \end{aligned} \quad (\text{B.30})$$

where  $C(z_{1:T}, z)$  is the number of occurrences of  $z$  in the sequence  $z_{1:T}$ . Recall that we can simplify the loss function in (3.4) as

$$\begin{aligned}\mathcal{L}(\tilde{\theta}) &= \mathbb{E}_{z_{1:T+1}} \left[ -\log \frac{\exp(e_{z_{T+1}}^\top \mathbf{W}_{\text{OV}} \mathbf{X}^\top \mathbf{X} \mathbf{W}_{\text{KQ}} \mathbf{x}_T)}{\sum_{z' \in \mathcal{A}} \exp(e_{z'}^\top \mathbf{W}_{\text{OV}} \mathbf{X}^\top \mathbf{X} \mathbf{W}_{\text{KQ}} \mathbf{x}_T)} \right] \\ &= \mathbb{E}_{z_{1:T+1}} [-e_{z_{T+1}}^\top \mathbf{W}_{\text{OV}} \mathbf{X}^\top \mathbf{X} \mathbf{W}_{\text{KQ}} \mathbf{x}_T + \log \sum_{z' \in \mathcal{A}} \exp(e_{z'}^\top \mathbf{W}_{\text{OV}} \mathbf{X}^\top \mathbf{X} \mathbf{W}_{\text{KQ}} \mathbf{x}_T)].\end{aligned}$$

Using (B.30), we get

$$\begin{aligned}\partial_{W_{\text{OV}}(a,z)} \mathcal{L}(\tilde{\theta}) &= W_{\text{KQ}}(z) \mathbb{E}_{z_{1:T+1}} \left[ -\mathbb{1}(a = z_{T+1}) C(z_{1:T}, z) \right. \\ &\quad \left. + \frac{\sum_{z' \in \mathcal{A}} \exp(e_{z'}^\top \mathbf{W}_{\text{OV}} \mathbf{X}^\top \mathbf{X} \mathbf{W}_{\text{KQ}} \mathbf{x}_T) \mathbb{1}(a = z') C(z_{1:T}, z)}{\sum_{z' \in \mathcal{A}} \exp(e_{z'}^\top \mathbf{W}_{\text{OV}} \mathbf{X}^\top \mathbf{X} \mathbf{W}_{\text{KQ}} \mathbf{x}_T)} \right] \\ &= -W_{\text{KQ}}(z) \mathbb{E}_{z_{1:T}} [C(z_{1:T}, z) (\mathbb{1}(a = a^*(z_{1:T})) - p_{\tilde{\theta}}(a|z_{1:T}))].\end{aligned}$$

Similarly,

$$\begin{aligned}\partial_{W_{\text{KQ}}(z)} \mathcal{L}(\tilde{\theta}) &= \mathbb{E}_{z_{1:T}} \left[ -C(z_{1:T}, z) W_{\text{OV}}(a^*(z_{1:T}), z) + \sum_{a \in \mathcal{A}} p_{\tilde{\theta}}(a|z_{1:T}) W_{\text{OV}}(a, z) C(z_{1:T}, z) \right] \\ &= \mathbb{E}_{z_{1:T}} \left[ C(z_{1:T}, z) (-W_{\text{OV}}(a^*(z_{1:T}), z) + \sum_{a \in \mathcal{A}} p_{\tilde{\theta}}(a|z_{1:T}) W_{\text{OV}}(a, z)) \right] \\ &= -\sum_{a \in \mathcal{A}} W_{\text{OV}}(a, z) \mathbb{E}_{z_{1:T}} [C(z_{1:T}, z) (\mathbb{1}(a = a^*(z_{1:T})) - p_{\tilde{\theta}}(a|z_{1:T}))].\end{aligned}$$

Let  $p(s) = \sum_{z_{1:T}} p(z_{1:T}) \mathbb{1}(s \in z_{1:T}) = \mathbb{E}_{\mathcal{D}_{\text{train}}} [\mathbb{1}(s \in z_{1:T})]$ , which is the marginal probability of observing  $s$  under the data-generating distribution  $\mathcal{D}_{\text{train}}$ . Similarly, we can define  $p(r)$  and  $p(s, r)$  for any  $r \in \mathcal{R}$  and any  $(s, r) \in \mathcal{S} \times \mathcal{R}$ . Then for a fixed  $(a, s) \in \mathcal{A} \times \mathcal{S}$ , we have

$$\begin{aligned}\partial_{W_{\text{OV}}(a,s)} \mathcal{L}(\tilde{\theta}) &= -W_{\text{KQ}}(s) \mathbb{E}_{z_{1:T} \sim \mathcal{D}_{\text{train}}} [C(z_{1:T}, s) (\mathbb{1}(a = a^*(z_{1:T})) - p_{\tilde{\theta}}(a|z_{1:T}))] \\ &\stackrel{(a)}{=} -W_{\text{KQ}}(s) \mathbb{E}_{z_{1:T} \sim \mathcal{D}_{\text{train}}} [\mathbb{1}(s \in z_{1:T}) (\mathbb{1}(a = a^*(z_{1:T})) - p_{\tilde{\theta}}(a|z_{1:T}))] \\ &= -W_{\text{KQ}}(s) \sum_{z_{1:T}} p(z_{1:T}) \mathbb{1}(s \in z_{1:T}) (\mathbb{1}(a = a^*(z_{1:T})) - p_{\tilde{\theta}}(a|z_{1:T})) \\ &= -W_{\text{KQ}}(s) p(s) \sum_{z_{1:T}} \left( \frac{p(z_{1:T}) \mathbb{1}(s \in z_{1:T})}{\sum_{z'_{1:T}} p(z'_{1:T}) \mathbb{1}(s \in z'_{1:T})} (\mathbb{1}(a = a^*(z_{1:T})) - p_{\tilde{\theta}}(a|z_{1:T})) \right) \\ &= -W_{\text{KQ}}(s) \cdot p(s) \cdot \mathbb{E}_{z_{1:T}} [\mathbb{1}(a = a^*(z_{1:T})) - p_{\tilde{\theta}}(a|z_{1:T}) \mid s \in z_{1:T}] \\ &= -W_{\text{KQ}}(s) \cdot \sum_{r \in \mathcal{R}} p(s, r) \cdot \mathbb{E}_{z_{1:T}} [\mathbb{1}(a = a^*(z_{1:T})) - p_{\tilde{\theta}}(a|z_{1:T}) \mid s, r \in z_{1:T}] \\ &\stackrel{(b)}{=} -W_{\text{KQ}}(s) \cdot \sum_{r \in \mathcal{R}} p(s, r) \cdot (\mathbb{1}(a = a^*(s, r)) - p_{\tilde{\theta}}(a|s, r)),\end{aligned}$$

where (a) comes from the fact that  $C(z_{1:T}, s) = \mathbb{1}(s \in z_{1:T})$  as  $s$  occurs at most once in the sequence from the task definition and (b) comes from the fact that  $z_{1:T}$  only contains  $(s, r, \langle \text{EOS} \rangle)$  and thus  $p_{\tilde{\theta}}(\cdot | z_{1:T}) = p_{\tilde{\theta}}(\cdot | s, r)$ .

Similarly, consider a fixed  $(a, r) \in \mathcal{A} \times \mathcal{R}$ , we have:

$$\begin{aligned} \partial_{W_{\text{OV}}(a,r)} \mathcal{L}(\tilde{\theta}) &= -W_{\text{KQ}}(r) \cdot p(r) \cdot \mathbb{E}_{z_{1:T}} \left[ \mathbb{1}(a = a^*(z_{1:T})) - p_{\tilde{\theta}}(a | z_{1:T}) \mid s \in z_{1:T} \right] \\ &= -W_{\text{KQ}}(r) \cdot \sum_{s \in \mathcal{S}} p(s, r) \cdot (\mathbb{1}(a = a^*(s, r)) - p_{\tilde{\theta}}(a | s, r)). \end{aligned}$$

Lastly for any  $a \in \mathcal{A}$ , we have:

$$\begin{aligned} \partial_{W_{\text{OV}}(a, \langle \text{EOS} \rangle)} \mathcal{L}(\tilde{\theta}) &= -W_{\text{KQ}}(\langle \text{EOS} \rangle) \sum_{z_{1:T}} p(z_{1:T}) (\mathbb{1}(a = a^*(z_{1:T})) - p_{\tilde{\theta}}(a | z_{1:T})) \\ &= -W_{\text{KQ}}(\langle \text{EOS} \rangle) \cdot \sum_{s \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s, r) \cdot (\mathbb{1}(a = a^*(s, r)) - p_{\tilde{\theta}}(a | s, r)). \end{aligned}$$

We conclude by proving the gradient in terms of  $W_{\text{KQ}}(s)$  for any  $s \in \mathcal{S}$ . We have:

$$\begin{aligned} \partial_{W_{\text{KQ}}(s)} \mathcal{L}(\tilde{\theta}) &= - \sum_{a \in \mathcal{A}} W_{\text{OV}}(a, s) \cdot p(s) \cdot \mathbb{E}_{z_{1:T}} \left[ \mathbb{1}(a = a^*(z_{1:T})) - p_{\tilde{\theta}}(a | z_{1:T}) \mid s \in z_{1:T} \right] \\ &= - \sum_{a \in \mathcal{A}} W_{\text{OV}}(a, s) \sum_{r \in \mathcal{R}} p(s, r) \cdot (\mathbb{1}(a = a^*(s, r)) - p_{\tilde{\theta}}(a | s, r)). \end{aligned}$$

□

**Lemma 29** (Lipschitz gradient). *Let  $d_0 := (|\mathcal{A}| + 1) \cdot |\mathcal{V}|$  and concatenate the parameters by*

$$\mathbf{w} = \begin{bmatrix} \text{vec}(\mathbf{W}_{\text{OV}}) \\ \mathbf{W}_{\text{KQ}} \end{bmatrix} \in \mathbb{R}^{d_0}.$$

*Let  $F(\mathbf{w}(t))$  be the vector field in the ODE*

$$\dot{\mathbf{w}}(t) = F(\mathbf{w}(t)) = -\nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}),$$

*where we omit the index  $t$  and set  $\mathbf{w}(t) = \mathbf{w}$  when the context is clear and  $\mathcal{L}(\mathbf{w})$  is the cross entropy loss given by:*

$$\mathcal{L}(\mathbf{w}) = \sum_{s,r} p(s, r) \left[ -\log \frac{\exp(f_{\mathbf{w}}((s, r), a^*(s, r)))}{\sum_{a \in \mathcal{A}} \exp(f_{\mathbf{w}}((s, r), a))} \right],$$

*where the logit function  $f_{\mathbf{w}}((s, r), a)$  follows Eq. (B.29) with  $z_{1:T} = (s, r, \langle \text{EOS} \rangle)$ . Assuming that  $\|\mathbf{w}\|_2 \leq R$  for some constant  $R > 0$ , then for any fixed  $(s, r, a)$ ,  $f_{\mathbf{w}}((s, r), a)$  is Lipschitz*

in  $\mathbf{w}$  with constant  $L_1 := 4R$ , i.e., for any  $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^{d_0}$  with  $\|\mathbf{w}_1\|_2 \leq R, \|\mathbf{w}_2\|_2 \leq R$ , it holds that

$$f_{\mathbf{w}_1}((s, r), a) - f_{\mathbf{w}_2}((s, r), a) \leq L_1 \|\mathbf{w}_1 - \mathbf{w}_2\|.$$

Moreover, there exists a constant  $L := 4|\mathcal{A}|(L_1 R + 1)$  such that

$$\|F(\mathbf{w}_1) - F(\mathbf{w}_2)\| \leq L \|\mathbf{w}_1 - \mathbf{w}_2\|.$$

*Proof.* Observing that  $f_{\mathbf{w}}((s, r), a)$  is bilinear in  $\mathbf{w}$ , i.e.,

$$\begin{aligned} f_{\mathbf{w}}((s, r), a) &= \sum_{t \in [T]} W_{\text{OV}}(a, z_t) W_{\text{KQ}}(z_t) \\ &= W_{\text{OV}}(a, s) W_{\text{KQ}}(s) + W_{\text{OV}}(a, r) W_{\text{KQ}}(r) + W_{\text{OV}}(a, \langle \text{EOS} \rangle) W_{\text{KQ}}(\langle \text{EOS} \rangle) \\ &= \mathbf{w}^\top \mathbf{M}((s, r), a) \mathbf{w}, \end{aligned}$$

where  $\mathbf{M}((s, r), a) \in \mathbb{R}^{d_0 \times d_0}$  has only three non-zero entries, which are equal to one, where the positions depend on  $((s, r), a)$ . We omit the index  $((s, r), a)$  when the context is clear. Then for any  $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^{d_0}$  with Euclidean norm bounded by  $R$ , we have

$$\begin{aligned} &f_{\mathbf{w}_1}((s, r), a) - f_{\mathbf{w}_2}((s, r), a) \\ &= \mathbf{w}_1^\top \mathbf{M} \mathbf{w}_1 - \mathbf{w}_2^\top \mathbf{M} \mathbf{w}_2 \\ &= \mathbf{w}_1^\top \mathbf{M} \mathbf{w}_1 - \mathbf{w}_1^\top \mathbf{M} \mathbf{w}_2 + \mathbf{w}_1^\top \mathbf{M} \mathbf{w}_2 - \mathbf{w}_2^\top \mathbf{M} \mathbf{w}_2 \\ &\leq (\|\mathbf{w}_1\| + \|\mathbf{w}_2\|) \cdot \|\mathbf{M}\|_F \cdot \|\mathbf{w}_1 - \mathbf{w}_2\| \\ &\stackrel{(a)}{\leq} 4R \|\mathbf{w}_1 - \mathbf{w}_2\| = L_1 \|\mathbf{w}_1 - \mathbf{w}_2\|, \end{aligned} \tag{B.31}$$

where (a) uses  $\|\mathbf{M}\|_F = \sqrt{3} \leq 2$ . Moreover, the gradient of  $f_{\mathbf{w}}((s, r), a)$  is also Lipschitz:

$$\nabla f_{\mathbf{w}_1}((s, r), a) - \nabla f_{\mathbf{w}_2}((s, r), a) = (\mathbf{M} + \mathbf{M}^\top)(\mathbf{w}_1 - \mathbf{w}_2) \leq 4 \|\mathbf{w}_1 - \mathbf{w}_2\|. \tag{B.32}$$

Now let  $p_k(\mathbf{u}) = \frac{\exp(u_k)}{\sum_{j=1}^K \exp(u_j)}$  and  $g_k(\mathbf{u}) = -\log p_k(\mathbf{u})$  for  $\mathbf{u} \in \mathbb{R}^K$  where  $K = |\mathcal{A}|$ , and denote  $p(\mathbf{u}) = (p_1(\mathbf{u}), \dots, p_K(\mathbf{u}))^\top \in \mathbb{R}^K$ ,  $g(\mathbf{u}) = (g_1(\mathbf{u}), \dots, g_K(\mathbf{u}))^\top \in \mathbb{R}^K$ . Then the gradient and hessian of  $g_k(\mathbf{u})$  are given by:

$$\begin{aligned} \nabla g_k(\mathbf{u}) &= p(\mathbf{u}) - \mathbf{e}_k \in \mathbb{R}^K, \\ H(\mathbf{u}) &= \text{diag}(p(\mathbf{u})) - p(\mathbf{u})p(\mathbf{u})^\top \in \mathbb{R}^{K \times K}. \end{aligned}$$

By the mean-value theorem, we have:

$$\|\nabla g_k(\mathbf{u}) - \nabla g_k(\mathbf{v})\| \leq \sup_{\mathbf{w} \in \mathbb{R}^K} \|H(\mathbf{w})\|_{\text{op}} \|\mathbf{u} - \mathbf{v}\| \leq \|\mathbf{u} - \mathbf{v}\|. \tag{B.33}$$

Let  $\mathbf{u}_{\mathbf{w}}(s, r) = f_{\mathbf{w}}((s, r), \cdot) \in \mathbb{R}^{|\mathcal{A}|}$  be the logit vector for input  $(s, r)$  and denote

$$\alpha_a(\mathbf{u}_{\mathbf{w}}(s, r)) := [\nabla g_{a^*(s, r)}(\mathbf{u}_{\mathbf{w}}(s, r))]_a = p_a(\mathbf{u}_{\mathbf{w}}(s, r)) - \mathbb{1}(a = a^*(s, r)).$$

Then the gradient of the loss function can be written as

$$\nabla \mathcal{L}(\mathbf{w}) = \sum_{s,r} p(s,r) \nabla_{\mathbf{w}} g_{a^*(s,r)}(\mathbf{u}_{\mathbf{w}}(s,r)) = \sum_{s,r} p(s,r) \sum_{a \in \mathcal{A}} \alpha_a(\mathbf{u}_{\mathbf{w}}(s,r)) \nabla f_{\mathbf{w}}((s,r), a).$$

For brevity let  $\nabla f_{\mathbf{w},a} := \nabla f_{\mathbf{w}}((s,r), a)$  and note that  $\|\nabla f_{\mathbf{w},a}\| = \|(\mathbf{M} + \mathbf{M}^\top)\mathbf{w}\| \leq 4R$ . Combining Eq. (B.31) and (B.33) we have:

$$\begin{aligned} & \|F(\mathbf{w}_1) - F(\mathbf{w}_2)\| \\ &= \|\nabla \mathcal{L}(\mathbf{w}_1) - \nabla \mathcal{L}(\mathbf{w}_2)\| \\ &\leq \sum_{s,r} p(s,r) \left\| \sum_{a \in \mathcal{A}} \alpha_a(\mathbf{u}_{\mathbf{w}_1}(s,r)) \nabla f_{\mathbf{w}_1,a} - \sum_{a \in \mathcal{A}} \alpha_a(\mathbf{u}_{\mathbf{w}_2}(s,r)) \nabla f_{\mathbf{w}_2,a} \right\| \\ &\leq \max_{s,r} \left\| \sum_{a \in \mathcal{A}} \left( (\alpha_a(\mathbf{u}_{\mathbf{w}_1}(s,r)) - \alpha_a(\mathbf{u}_{\mathbf{w}_2}(s,r))) \nabla f_{\mathbf{w}_1,a} + \alpha_a(\mathbf{u}_{\mathbf{w}_2}(s,r)) (\nabla f_{\mathbf{w}_1,a} - \nabla f_{\mathbf{w}_2,a}) \right) \right\| \\ &\stackrel{(a)}{\leq} \max_{s,r} \left( \sum_{a \in \mathcal{A}} \|\nabla f_{\mathbf{w}_1,a}\| \cdot \|\mathbf{u}_{\mathbf{w}_1}(s,r) - \mathbf{u}_{\mathbf{w}_2}(s,r)\| + 4|\mathcal{A}| \max_a |\alpha_a(\mathbf{u}_{\mathbf{w}_2}(s,r))| \cdot \|\mathbf{w}_1 - \mathbf{w}_2\| \right) \\ &\stackrel{(b)}{\leq} \|\mathbf{w}_1 - \mathbf{w}_2\| \cdot \left( L_1 \sum_{a \in \mathcal{A}} \|\nabla f_{\mathbf{w}_1,a}\| + 4|\mathcal{A}| \right) \\ &\leq 4|\mathcal{A}|(L_1 R + 1) \|\mathbf{w}_1 - \mathbf{w}_2\|, \end{aligned}$$

where (a) follows Eq. (B.32) and Eq. (B.33) and (b) uses Eq. (B.31) and  $|\alpha_a(\mathbf{u}_{\mathbf{w}_2}(s,r))| \leq 1$  for any  $a \in \mathcal{A}$ .  $\square$

Before proceeding, we provide the following definition.

**Definition 2** (Data permutation). Let  $d_0 := (|\mathcal{A}| + 1) \cdot |\mathcal{V}|$ . Consider the flattened parameter  $\mathbf{w}$  defined as

$$\mathbf{w} = \begin{bmatrix} \text{vec}(\mathbf{W}_{\text{OV}}) \\ \mathbf{W}_{\text{KQ}} \end{bmatrix} \in \mathbb{R}^{d_0}.$$

Recall  $|\mathcal{A}_1| = |\mathcal{A}_2| = n$ . Let  $\sigma$  be any permutation over  $[n]$  where  $\sigma(i) \neq i$  for any  $i \in [n]$  and  $\pi : \mathcal{V} \rightarrow \mathcal{V}$  be a permutation function determined by  $\sigma$ . Specifically, for any  $i \in [n], j \in [m]$ , we have

$$\pi(b_i) = b_{\sigma(i)}, \pi(c_i) = c_{\sigma(i)}, \pi(s_{i,j}) = s_{\sigma(i),j},$$

and we have  $\pi(v) = v$  for  $v \in \{r_1, r_2, \langle \text{EOS} \rangle\}$ . Moreover, let  $\mathbf{P}_\pi$  be a permutation matrix built on permutation  $\pi$  defined as follows. First, we have

$$\mathbf{P} := \mathbf{P}_\pi = \begin{bmatrix} \mathbf{P}_{\text{OV}} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{\text{KQ}} \end{bmatrix} \in \mathbb{R}^{d_0 \times d_0},$$

where we omit the subscript  $\pi$  for brevity. We denote the resulting OV block in  $\mathbf{w}$  after permutation as  $(\mathbf{P}\mathbf{w})_{\text{OV}} := \mathbf{P}_{\text{OV}} \text{vec}(\mathbf{W}_{\text{OV}}) \in \mathbb{R}^{|\mathcal{V}| \cdot |\mathcal{A}|}$ . Similarly we denote  $(\mathbf{P}\mathbf{w})_{\text{KQ}} := \mathbf{P}_{\text{KQ}} \mathbf{W}_{\text{KQ}} \in \mathbb{R}^{|\mathcal{V}|}$ . Then, each entry of  $\mathbf{P}\mathbf{w}$  is defined as follows:

- $\forall s \in \mathcal{S}_{\text{test}}, \quad \forall a \in \mathcal{A}_2 : \quad (\mathbf{P}\mathbf{w})_{\text{OV}}(a, s) = W_{\text{OV}}(\pi(a), s),$
- $\forall s \in \mathcal{S}_{\text{train}}, \quad \forall a \in \mathcal{A} : \quad (\mathbf{P}\mathbf{w})_{\text{OV}}(a, s) = W_{\text{OV}}(\pi(a), \pi(s)),$
- $\forall v \in \{r_1, r_2, \langle \text{EOS} \rangle\}, \quad \forall a \in \mathcal{A}_2 : \quad (\mathbf{P}\mathbf{w})_{\text{OV}}(a, v) = W_{\text{OV}}(\pi(a), v),$
- $\forall s \in \mathcal{S}_{\text{train}} : \quad (\mathbf{P}\mathbf{w})_{\text{KQ}}(s) = W_{\text{KQ}}(\pi(s)),$
- *Otherwise,  $(\mathbf{P}\mathbf{w})_{\text{OV}}(a, v) = W_{\text{OV}}(a, v), (\mathbf{P}\mathbf{w})_{\text{KQ}}(v) = W_{\text{KQ}}(v).$*

**Lemma 30** (Gradient is permutation equivariant). *Recall the flattened parameter  $\mathbf{w}$  and the permutation matrix  $\mathbf{P}$  defined in Definition 2. Denote  $F(\mathbf{w}) = -\nabla \mathcal{L}(\mathbf{w})$  following Lemma 29, then we have*

$$\forall s \in \mathcal{S}_{\text{test}}, \quad \forall a \in \mathcal{A}_2 : \quad [F(\mathbf{P}\mathbf{w})]_{\text{OV}}(a, s) = [\mathbf{P}F(\mathbf{w})]_{\text{OV}}(a, s), \quad (\text{B.34a})$$

$$\forall s \in \mathcal{S}_{\text{train}}, \quad \forall a \in \mathcal{A} : \quad [F(\mathbf{P}\mathbf{w})]_{\text{OV}}(a, s) = [\mathbf{P}F(\mathbf{w})]_{\text{OV}}(a, s), \quad (\text{B.34b})$$

$$\forall v \in \mathcal{R} \cup \{\langle \text{EOS} \rangle\}, \quad \forall a \in \mathcal{A}_2 : \quad [F(\mathbf{P}\mathbf{w})]_{\text{OV}}(a, v) = [\mathbf{P}F(\mathbf{w})]_{\text{OV}}(a, v), \quad (\text{B.34c})$$

$$\forall s \in \mathcal{S}_{\text{train}} : \quad [F(\mathbf{P}\mathbf{w})]_{\text{KQ}}(s) = [\mathbf{P}F(\mathbf{w})]_{\text{KQ}}(s), \quad (\text{B.34d})$$

which implies the gradient of  $\mathcal{L}$  w.r.t  $\mathbf{w}$  is permutation equivariant, i.e.,

$$\nabla \mathcal{L}(\mathbf{P}\mathbf{w}) = \mathbf{P} \nabla \mathcal{L}(\mathbf{w}).$$

*Proof.* We examine Eq. (B.34a) - (B.34d) one by one.

**Part 1: Proof of Eq. (B.34a).** Given a fixed  $s \in \mathcal{S}_{\text{test}}, a \in \mathcal{A}_2$ , we will prove

$$[F(\mathbf{P}\mathbf{w})]_{\text{OV}}(a, s) = [\mathbf{P}F(\mathbf{w})]_{\text{OV}}(a, s). \quad (\text{B.35})$$

Using Lemma 28, we have:

$$\begin{aligned} [F(\mathbf{w})]_{\text{OV}}(a, s) &= -\partial_{W_{\text{OV}}(a, s)} \mathcal{L}(\mathbf{w}) \\ &= W_{\text{KQ}}(s) \sum_{r \in \mathcal{R}} p(s, r) [\mathbb{1}(a = a^*(s, r)) - p_{\mathbf{w}}(a|s, r)] \\ &\stackrel{(a)}{=} W_{\text{KQ}}(s) \cdot p(s, r_1) [\mathbb{1}(a = a^*(s, r_1)) - p_{\mathbf{w}}(a|s, r_1)], \end{aligned}$$

where (a) comes from the fact that  $p(s, r_2) = 0$  since  $s \in \mathcal{S}_{\text{test}}$ . Then for the right side of Eq. (B.35), we have

$$\begin{aligned} [\mathbf{P}F(\mathbf{w})]_{\text{OV}}(a, s) &\stackrel{(a)}{=} -\partial_{W_{\text{OV}}(\pi(a), s)} \mathcal{L}(\mathbf{w}) \\ &= W_{\text{KQ}}(s) \cdot p(s, r_1) [\mathbb{1}(\pi(a) = a^*(s, r_1)) - p_{\mathbf{w}}(\pi(a)|s, r_1)], \end{aligned} \quad (\text{B.36})$$

where (a) holds by Definition 2. Similarly, for the left-hand side, we have

$$\begin{aligned} [F(\mathbf{P}\mathbf{w})]_{\text{OV}}(a, s) &= -\partial_{W_{\text{OV}}(a, s)} \mathcal{L}(\mathbf{P}\mathbf{w}) \\ &\stackrel{(a)}{=} W_{\text{KQ}}(s) \cdot p(s, r_1) [\mathbb{1}(a = a^*(s, r_1)) - p_{\mathbf{P}\mathbf{w}}(a|s, r_1)], \end{aligned} \quad (\text{B.37})$$

where (a) comes from  $(\mathbf{P}\mathbf{w})_{\text{KQ}}(s) = W_{\text{KQ}}(s)$  for any  $s \in \mathcal{S}_{\text{test}}$ . For the prediction probability, recall from (3.3), we have

$$p_{\mathbf{w}}(a|s, r) = \frac{\exp(f_{\mathbf{w}}((s, r), a))}{\sum_{a' \in \mathcal{A}} \exp(f_{\mathbf{w}}((s, r), a'))}.$$

The logit function  $f_{\mathbf{w}}((s, r), a)$  can be written as

$$f_{\mathbf{w}}((s, r), a) = W_{\text{KQ}}(s)W_{\text{OV}}(a, s) + W_{\text{KQ}}(r)W_{\text{OV}}(a, r) + W_{\text{KQ}}(\langle \text{EOS} \rangle)W_{\text{OV}}(a, \langle \text{EOS} \rangle).$$

Therefore, given  $s \in \mathcal{S}_{\text{test}}$  and  $a \in \mathcal{A}_2$ , we have

$$\begin{aligned} f_{\mathbf{P}\mathbf{w}}((s, r), a) &= W_{\text{KQ}}(s)W_{\text{OV}}(\pi(a), s) + W_{\text{KQ}}(r)W_{\text{OV}}(\pi(a), r) \\ &\quad + W_{\text{KQ}}(\langle \text{EOS} \rangle)W_{\text{OV}}(\pi(a), \langle \text{EOS} \rangle) \\ &= f_{\mathbf{w}}((s, r), \pi(a)). \end{aligned}$$

Similarly, for any  $a' \in \mathcal{A}_1$ , given  $s \in \mathcal{S}_{\text{test}}$ , we have

$$f_{\mathbf{P}\mathbf{w}}((s, r), a') = f_{\mathbf{w}}((s, r), a').$$

Then

$$\begin{aligned} p_{\mathbf{P}\mathbf{w}}(a|s, r) &= \frac{\exp(f_{\mathbf{P}\mathbf{w}}((s, r), a))}{\sum_{a' \in \mathcal{A}} \exp(f_{\mathbf{P}\mathbf{w}}((s, r), a'))} \\ &= \frac{\exp(f_{\mathbf{w}}((s, r), \pi(a)))}{\sum_{a' \in \mathcal{A}_1} \exp(f_{\mathbf{w}}((s, r), a')) + \sum_{a' \in \mathcal{A}_2} \exp(f_{\mathbf{w}}((s, r), \pi(a')))} \quad (\text{B.38}) \\ &\stackrel{(a)}{=} \frac{\exp(f_{\mathbf{w}}((s, r), \pi(a)))}{\sum_{z \in \mathcal{A}} \exp(f_{\mathbf{w}}((s, r), z))} \\ &= p_{\mathbf{w}}(\pi(a)|s, r), \end{aligned}$$

where in (a) we re-index the summand by  $z := \pi(a')$  since  $\pi$  is a bijection for any  $a' \in \mathcal{A}_2$ . Then Eq. (B.37) can be written as

$$[F(\mathbf{P}\mathbf{w})]_{\text{OV}}(a, s) = W_{\text{KQ}}(s) \cdot p(s, r_1)[\mathbb{1}(a = a^*(s, r_1)) - p_{\mathbf{w}}(\pi(a)|s, r_1)]. \quad (\text{B.39})$$

By comparing Eq. (B.36) and Eq. (B.39), it remains to prove

$$\mathbb{1}(a = a^*(s, r_1)) = \mathbb{1}(\pi(a) = a^*(s, r_1)),$$

which both equal 0 since  $a, \pi(a) \in \mathcal{A}_2$ . Then we conclude that for any  $s \in \mathcal{S}_{\text{test}}, a \in \mathcal{A}_2$ ,

$$[F(\mathbf{P}\mathbf{w})]_{\text{OV}}(a, s) = [\mathbf{P}F(\mathbf{w})]_{\text{OV}}(a, s).$$

**Part 2: Proof of Eq. (B.34b).** Consider any fixed  $s \in \mathcal{S}_{\text{train}}, a \in \mathcal{A}$ , we start from the RHS of the equation:

$$\begin{aligned}
[\mathbf{P}F(\mathbf{w})]_{\text{OV}}(a, s) &= -\partial_{W_{\text{OV}}(\pi(a), \pi(s))} \mathcal{L}(\mathbf{w}) \\
&= W_{\text{KQ}}(\pi(s)) \cdot \sum_{r \in \mathcal{R}} p(\pi(s), r) [\mathbb{1}(\pi(a) = a^*(\pi(s), r)) - p_{\mathbf{w}}(\pi(a) | \pi(s), r)] \\
&\stackrel{(a)}{=} W_{\text{KQ}}(\pi(s)) \cdot \sum_{r \in \mathcal{R}} p(s, r) [\mathbb{1}(\pi(a) = a^*(\pi(s), r)) - p_{\mathbf{w}}(\pi(a) | \pi(s), r)],
\end{aligned} \tag{B.40}$$

where (a) uses  $p(s, r) = p(\pi(s), r)$  for any fixed  $r \in \mathcal{R}$  as data is uniformly distributed. Similarly, on the LHS, we have

$$\begin{aligned}
[F(\mathbf{P}\mathbf{w})]_{\text{OV}}(a, s) &= -\partial_{W_{\text{OV}}(a, s)} \mathcal{L}(\mathbf{P}\mathbf{w}) \\
&= W_{\text{KQ}}(\pi(s)) \cdot \sum_{r \in \mathcal{R}} p(s, r) [\mathbb{1}(a = a^*(s, r)) - p_{\mathbf{P}\mathbf{w}}(a | s, r)],
\end{aligned}$$

To proceed, given  $s \in \mathcal{S}_{\text{train}}$  and  $a \in \mathcal{A}$ , we have

$$\begin{aligned}
f_{\mathbf{P}\mathbf{w}}((s, r), a) &= W_{\text{KQ}}(\pi(s)) W_{\text{OV}}(\pi(a), \pi(s)) + W_{\text{KQ}}(r) W_{\text{OV}}(\pi(a), r) \\
&\quad + W_{\text{KQ}}(\langle \text{EOS} \rangle) W_{\text{OV}}(\pi(a), \langle \text{EOS} \rangle) \\
&= f_{\mathbf{w}}((\pi(s), r), \pi(a)).
\end{aligned}$$

Following Eq. (B.38), when  $s \in \mathcal{S}_{\text{train}}$ ,  $a \in \mathcal{A}$ , we have

$$\begin{aligned}
p_{\mathbf{P}\mathbf{w}}(a | s, r) &= \frac{\exp(f_{\mathbf{P}\mathbf{w}}((s, r), a))}{\sum_{a' \in \mathcal{A}} \exp(f_{\mathbf{P}\mathbf{w}}((s, r), a'))} \\
&\stackrel{(a)}{=} \frac{\exp(f_{\mathbf{w}}((\pi(s), r), \pi(a)))}{\sum_{z \in \mathcal{A}} \exp(f_{\mathbf{w}}((\pi(s), r), z))} \\
&= p_{\mathbf{w}}(\pi(a) | \pi(s), r),
\end{aligned} \tag{B.41}$$

where in (a) we re-index the summand by  $z := \pi(a')$ . Thus we get

$$\begin{aligned}
[F(\mathbf{P}\mathbf{w})]_{\text{OV}}(a, s) &= W_{\text{KQ}}(\pi(s)) \cdot \sum_{r \in \mathcal{R}} p(s, r) [\mathbb{1}(a = a^*(s, r)) - p_{\mathbf{P}\mathbf{w}}(a | s, r)] \\
&= W_{\text{KQ}}(\pi(s)) \cdot \sum_{r \in \mathcal{R}} p(s, r) [\mathbb{1}(a = a^*(s, r)) - p_{\mathbf{w}}(\pi(a) | \pi(s), r)],
\end{aligned} \tag{B.42}$$

By comparing Eq. (B.40) and Eq. (B.42), it remains to prove for any  $r \in \mathcal{R}$ ,

$$\mathbb{1}(a = a^*(s, r)) = \mathbb{1}(\pi(a) = a^*(\pi(s), r)),$$

which holds since the event  $a = a^*(s, r)$  is equivalent to  $\pi(a) = a^*(\pi(s), r)$  by the definition of  $\pi$  and the dataset construction. Then we can conclude that for any  $s \in \mathcal{S}_{\text{train}}$ ,  $a \in \mathcal{A}$ ,

$$[F(\mathbf{P}\mathbf{w})]_{\text{OV}}(a, s) = [\mathbf{P}F(\mathbf{w})]_{\text{OV}}(a, s).$$

**Part 3: Proof of Eq. (B.34c).** Let's first consider  $v \in \mathcal{R} = \{r_1, r_2\}$ . For any fixed  $r := v \in \{r_1, r_2\}$  and  $a \in \mathcal{A}_2$ , using Lemma 28, we have

$$[F(\mathbf{w})]_{\text{OV}}(a, r) = W_{\text{KQ}}(r) \cdot \sum_{s \in \mathcal{S}} p(s, r) \cdot (\mathbb{1}(a = a^*(s, r)) - p_{\mathbf{w}}(a|s, r)).$$

Then we have

$$\begin{aligned} [PF(\mathbf{w})]_{\text{OV}}(a, r) &= W_{\text{KQ}}(r) \cdot \sum_{s \in \mathcal{S}} p(s, r) \cdot (\mathbb{1}(\pi(a) = a^*(s, r)) - p_{\mathbf{w}}(\pi(a)|s, r)) \\ &= W_{\text{KQ}}(r) \cdot \sum_{s \in \mathcal{S}_{\text{train}}} p(s, r) \cdot (\mathbb{1}(\pi(a) = a^*(s, r)) - p_{\mathbf{w}}(\pi(a)|s, r)) \\ &\quad + W_{\text{KQ}}(r) \cdot \sum_{s \in \mathcal{S}_{\text{test}}} p(s, r) \cdot (\mathbb{1}(\pi(a) = a^*(s, r)) - p_{\mathbf{w}}(\pi(a)|s, r)). \end{aligned} \quad (\text{B.43})$$

Similarly,

$$\begin{aligned} [F(\mathbf{Pw})]_{\text{OV}}(a, r) &= W_{\text{KQ}}(r) \cdot \sum_{s \in \mathcal{S}} p(s, r) \cdot (\mathbb{1}(a = a^*(s, r)) - p_{\mathbf{Pw}}(a|s, r)) \\ &\stackrel{(a)}{=} W_{\text{KQ}}(r) \cdot \sum_{s \in \mathcal{S}_{\text{train}}} p(s, r) \cdot (\mathbb{1}(a = a^*(s, r)) - p_{\mathbf{w}}(\pi(a)|\pi(s), r)) \\ &\quad + W_{\text{KQ}}(r) \cdot \sum_{s \in \mathcal{S}_{\text{test}}} p(s, r) \cdot (\mathbb{1}(a = a^*(s, r)) - p_{\mathbf{w}}(\pi(a)|s, r)), \end{aligned} \quad (\text{B.44})$$

where (a) follows Eq. (B.38). Now we compare (B.43) and (B.44). We first consider  $\mathcal{S}_{\text{train}}$ . Note that for any  $s \in \mathcal{S}_{\text{train}}$  and fixed  $r \in \mathcal{R}$ , the values of  $p(s, r)$  are the same due to the data distribution, and we denote the value as  $p_{\text{train}}$ . To proceed, let  $i, i'$  be the corresponding index of  $a, \pi(a)$  in  $\mathcal{A}_2$ , i.e.,  $a = c_i, \pi(a) = c_{i'}$ . Then

$$\begin{aligned} &\sum_{s \in \mathcal{S}_{\text{train}}} p(s, r) \cdot ((\mathbb{1}(a = a^*(s, r)) - p_{\mathbf{w}}(\pi(a)|\pi(s), r)) - (\mathbb{1}(\pi(a) = a^*(s, r)) - p_{\mathbf{w}}(\pi(a)|s, r))) \\ &= p_{\text{train}} \sum_{j \in [m_{\text{train}}]} \left( \sum_{k \in [n], k \notin \{i, i'\}} \underbrace{(\mathbb{1}(a = a^*(s_{k,j}, r)) - \mathbb{1}(\pi(a) = a^*(s_{k,j}, r)))}_{\Gamma(a, \pi(a), s_{k,j}, r)} \right. \\ &\quad + \sum_{k' \in \{i, i'\}} \underbrace{(\mathbb{1}(a = a^*(s_{k',j}, r)) - \mathbb{1}(\pi(a) = a^*(s_{k',j}, r)))}_{\Gamma(a, \pi(a), s_{k',j}, r)} \\ &\quad \left. + \sum_{k \in [n]} p_{\mathbf{w}}(\pi(a)|s_{k,j}, r) - p_{\mathbf{w}}(\pi(a)|\pi(s_{k,j}), r) \right). \end{aligned} \quad (\text{B.45})$$

For the first term  $\Gamma(a, \pi(a), s_{k,j}, r)$ , note that  $\mathbb{1}(a = a^*(s_{k,j}, r)) = \mathbb{1}(\pi(a) = a^*(s_{k,j}, r)) = 0$  for any  $j \in [m], k \in [n], k \neq \{i, i'\}$ . Now we consider the second term  $\Gamma(a, \pi(a), s_{k',j}, r)$ . We have

$$\begin{aligned} \mathbb{1}(a = a^*(s_{k',j}, r_2)) &= 1, \quad \mathbb{1}(\pi(a) = a^*(s_{k',j}, r_2)) = 0, \quad \text{when } k' = i, \\ \mathbb{1}(a = a^*(s_{k',j}, r_2)) &= 0, \quad \mathbb{1}(\pi(a) = a^*(s_{k',j}, r_2)) = 1, \quad \text{when } k' = i', \\ \mathbb{1}(a = a^*(s_{k',j}, r_1)) &= \mathbb{1}(\pi(a) = a^*(s_{k',j}, r_1)) = 0, \quad \text{when } k' \in \{i, i'\}, \end{aligned}$$

which implies  $\sum_{k' \in \{i, i'\}} \Gamma(a, \pi(a), s_{k',j}, r) = 0$ . Then we can simplify Eq. (B.45) as

$$\begin{aligned} & \sum_{s \in \mathcal{S}_{\text{train}}} p(s, r) \cdot ((\mathbb{1}(a = a^*(s, r)) - p_{\mathbf{w}}(\pi(a)|\pi(s), r)) - (\mathbb{1}(\pi(a) = a^*(s, r)) - p_{\mathbf{w}}(\pi(a)|s, r))) \\ &= p_{\text{train}} \sum_{j \in [m_{\text{train}}]} \left( \sum_{k \in [n]} p_{\mathbf{w}}(\pi(a)|s_{k,j}, r) - \sum_{k \in [n]} p_{\mathbf{w}}(\pi(a)|\pi(s_{k,j}), r) \right) \\ &\stackrel{(a)}{=} p_{\text{train}} \sum_{j \in [m_{\text{train}}]} \left( \sum_{k \in [n]} p_{\mathbf{w}}(\pi(a)|s_{k,j}, r) - \sum_{k' \in [n]} p_{\mathbf{w}}(\pi(a)|s_{k',j}, r) \right) \\ &= 0, \end{aligned} \tag{B.46}$$

where in (a) we re-index the summand  $\sum_{k \in [n]} p_{\mathbf{w}}(\pi(a)|\pi(s_{k,j}), r)$  by noting that  $\pi(s_{k,j}) = s_{\sigma(k),j}$ . Next, we consider  $\mathcal{S}_{\text{test}}$ . When  $r = r_2$ , we have  $p(s, r_2) = 0$  for any  $s \in \mathcal{S}_{\text{test}}$ . Otherwise, when  $r = r_1$ , we similarly can define  $p_{\text{test}} := p(s, r_1)$  and obtain that

$$\begin{aligned} & \sum_{s \in \mathcal{S}_{\text{test}}} p(s, r) \cdot ((\mathbb{1}(\pi(a) = a^*(s, r)) - p_{\mathbf{w}}(\pi(a)|s, r)) - (\mathbb{1}(a = a^*(s, r)) - p_{\mathbf{w}}(\pi(a)|s, r))) \\ &= p_{\text{test}} \sum_{s \in \mathcal{S}_{\text{test}}} (\mathbb{1}(\pi(a) = a^*(s, r)) - \mathbb{1}(a = a^*(s, r))) \\ &\stackrel{(a)}{=} 0, \end{aligned} \tag{B.47}$$

where (a) holds since for any  $s \in \mathcal{S}_{\text{test}}$ ,  $\mathbb{1}(\pi(a) = a^*(s, r)) = \mathbb{1}(a = a^*(s, r)) = 0$  when  $r = r_1, a \in \mathcal{A}_2$ . Combining Eq. (B.46) and (B.47), for any  $r \in \mathcal{R}$  we have

$$[F(\mathbf{P}\mathbf{w})]_{\text{OV}}(a, r) = [\mathbf{P}F(\mathbf{w})]_{\text{OV}}(a, r).$$

When  $v = \langle \text{EOS} \rangle$ , using Lemma 28 again, we have

$$\begin{aligned} [F(\mathbf{w})]_{\text{OV}}(a, \langle \text{EOS} \rangle) &= W_{\text{KQ}}(\langle \text{EOS} \rangle) \cdot \sum_{r \in \mathcal{R}} \sum_{s \in \mathcal{S}} p(s, r) \cdot (\mathbb{1}(a = a^*(s, r)) - p_{\mathbf{w}}(a|s, r)) \\ &= \frac{W_{\text{KQ}}(\langle \text{EOS} \rangle)}{W_{\text{KQ}}(r)} \cdot \sum_{r \in \mathcal{R}} W_{\text{KQ}}(r) \sum_{s \in \mathcal{S}} p(s, r) \cdot (\mathbb{1}(a = a^*(s, r)) - p_{\mathbf{w}}(a|s, r)) \\ &= \frac{W_{\text{KQ}}(\langle \text{EOS} \rangle)}{W_{\text{KQ}}(r)} \cdot \sum_{r \in \mathcal{R}} [F(\mathbf{w})]_{\text{OV}}(a, r). \end{aligned}$$

Thus we have

$$\begin{aligned}
& [F(\mathbf{P}\mathbf{w})]_{\text{OV}}(a, \langle \text{EOS} \rangle) - [\mathbf{P}F(\mathbf{w})]_{\text{OV}}(a, \langle \text{EOS} \rangle) \\
&= \frac{W_{\text{KQ}}(\langle \text{EOS} \rangle)}{W_{\text{KQ}}(r)} \cdot \sum_{r \in \mathcal{R}} ([F(\mathbf{P}\mathbf{w})]_{\text{OV}}(a, r) - [\mathbf{P}F(\mathbf{w})]_{\text{OV}}(a, r)) \\
&= 0.
\end{aligned}$$

**Part 4: Proof of Eq. (B.34d).** Given a fixed  $s \in \mathcal{S}_{\text{train}}$ , using Lemma 28, we have:

$$\begin{aligned}
[F(\mathbf{w})]_{\text{KQ}}(s) &= -\partial_{W_{\text{KQ}}(s)} \mathcal{L}(\mathbf{w}) \\
&= \sum_{r \in \mathcal{R}} p(s, r) \sum_{a \in \mathcal{A}} W_{\text{OV}}(a, s) [\mathbb{1}(a = a^*(s, r)) - p_{\mathbf{w}}(a|s, r)].
\end{aligned}$$

For the RHS of Eq. (B.34d), we have

$$\begin{aligned}
[\mathbf{P}F(\mathbf{w})]_{\text{KQ}}(s) &= -\partial_{W_{\text{KQ}}(\pi(s))} \mathcal{L}(\mathbf{w}) \\
&= \sum_{r \in \mathcal{R}} p(\pi(s), r) \sum_{a \in \mathcal{A}} W_{\text{OV}}(a, \pi(s)) [\mathbb{1}(a = a^*(\pi(s), r)) - p_{\mathbf{w}}(a|\pi(s), r)] \\
&\stackrel{(a)}{=} \sum_{r \in \mathcal{R}} p(s, r) \sum_{a \in \mathcal{A}} W_{\text{OV}}(a, \pi(s)) [\mathbb{1}(a = a^*(\pi(s), r)) - p_{\mathbf{w}}(a|\pi(s), r)], \quad (\text{B.48})
\end{aligned}$$

where (a) uses  $p(s, r) = p(\pi(s), r)$ ,  $\forall r \in \mathcal{R}$ . On the LHS, we have

$$\begin{aligned}
[F(\mathbf{P}\mathbf{w})]_{\text{KQ}}(s) &\stackrel{(a)}{=} \sum_{r \in \mathcal{R}} p(s, r) \sum_{a \in \mathcal{A}} W_{\text{OV}}(\pi(a), \pi(s)) [\mathbb{1}(a = a^*(s, r)) - p_{\mathbf{P}\mathbf{w}}(a|s, r)] \\
&\stackrel{(b)}{=} \sum_{r \in \mathcal{R}} p(s, r) \sum_{a \in \mathcal{A}} W_{\text{OV}}(\pi(a), \pi(s)) [\mathbb{1}(a = a^*(s, r)) - p_{\mathbf{w}}(\pi(a)|\pi(s), r)] \quad (\text{B.49}) \\
&\stackrel{(c)}{=} \sum_{r \in \mathcal{R}} p(s, r) \sum_{z \in \mathcal{A}} W_{\text{OV}}(z, \pi(s)) [\mathbb{1}(\pi^{-1}(z) = a^*(s, r)) - p_{\mathbf{w}}(z|\pi(s), r)],
\end{aligned}$$

where (a) comes from  $(\mathbf{P}\mathbf{w})_{\text{OV}}(a, s) = W_{\text{OV}}(\pi(a), \pi(s))$  for  $s \in \mathcal{S}_{\text{train}}$ ,  $a \in \mathcal{A}$ , (b) follows Eq. (B.41), and in (c) we re-index the summand by  $z = \pi(a)$  and thus  $a = \pi^{-1}(z)$ . Now by comparing Eq. (B.48) and Eq. (B.49), it remains to prove that for any  $a \in \mathcal{A}$ ,

$$\mathbb{1}(\pi^{-1}(a) = a^*(s, r)) = \mathbb{1}(a = a^*(\pi(s), r)),$$

which holds by the definition of  $\pi$ .  $\square$

**Lemma 31.** Assuming Assumption 2 holds. Let  $\mathbf{P}$  be the permutation matrix defined in Lemma 30. If for any  $t \geq 0$ , there exists a constant  $R > 0$  such that  $\|\mathbf{w}\| \leq R$  is bounded, then  $F(\mathbf{w}) := -\nabla \mathcal{L}(\mathbf{w})$  is Lipschitz with some constant  $L$ . Moreover, since  $F(\mathbf{w})$  is permutation equivariant w.r.t  $\mathbf{P}$ , i.e.,  $F(\mathbf{P}\mathbf{w}) = \mathbf{P}F(\mathbf{w})$  as established in Lemma 30, we have

$$\mathbf{w}(t) = \mathbf{P}\mathbf{w}(t).$$

In particular, for any  $t \geq 0$ , we have

$$W_{\text{OV}}(a, v) = W_{\text{OV}}(a', v), \quad \forall v \in \mathcal{S}_{\text{test}} \cup \mathcal{R} \cup \{\langle \text{EOS} \rangle\}, \forall a, a' \in \mathcal{A}_2. \quad (\text{B.50})$$

*Proof.* Note that when  $\mathbf{w}$  is bounded by  $R$ , i.e.,  $\|\mathbf{w}\| \leq R$ ,  $F(\mathbf{w})$  is Lipschitz with constant  $L$  using Lemma 29. Moreover, Lemma 30 indicates that  $F(\mathbf{w})$  is also permutation equivariant. Let  $\mathbf{D}(t) := \mathbf{w}(t) - \mathbf{P}\mathbf{w}(t)$ , then we have  $\mathbf{D}(0) = \mathbf{0}$  by Assumption 2. Note that

$$\dot{\mathbf{D}}(t) = \dot{\mathbf{w}}(t) - \mathbf{P}\dot{\mathbf{w}}(t) = F(\mathbf{w}(t)) - \mathbf{P}F(\mathbf{w}(t)) = F(\mathbf{w}(t)) - F(\mathbf{P}\mathbf{w}(t)).$$

Let  $\mathbf{w}_1 = \mathbf{w}(t)$ ,  $\mathbf{w}_2 = \mathbf{P}\mathbf{w}(t)$ , respectively. Using Lemma 29, we get

$$\|\dot{\mathbf{D}}(t)\| = \|F(\mathbf{w}(t)) - F(\mathbf{P}\mathbf{w}(t))\| \leq L\|\mathbf{D}(t)\|.$$

Then

$$\frac{d}{dt}\|\mathbf{D}(t)\| = \frac{\mathbf{D}(t)^\top \dot{\mathbf{D}}(t)}{\|\mathbf{D}(t)\|} \stackrel{(a)}{\leq} \|\dot{\mathbf{D}}(t)\| \leq L\|\mathbf{D}(t)\|,$$

where (a) uses the Cauchy-Schwarz inequality. Using Gronwall's inequality, for all  $t \geq 0$ , we have

$$\|\mathbf{D}(t)\| \leq \|\mathbf{D}(0)\|e^{Lt} = 0,$$

since  $\|\mathbf{D}(0)\| = 0$ . Equivalently, for any  $t \geq 0$ , we have:

$$\mathbf{w}(t) = \mathbf{P}\mathbf{w}(t).$$

Moreover, recall the definition of  $\pi, \mathbf{P}$  in Lemma 30. Given a fixed  $v \in \mathcal{S}_{\text{test}} \cup \mathcal{R} \cup \{\langle \text{EOS} \rangle\}$  and any  $a, a' \in \mathcal{A}_2$ , we have

$$W_{\text{OV}}(a, v) \stackrel{(a)}{=} W_{\text{OV}}(\pi(a), v) \stackrel{(b)}{=} W_{\text{OV}}(a', v), \quad (\text{B.51})$$

where (a) follows Definition 2, and (b) holds since for any fixed  $a, a' \in \mathcal{A}_2$ , one can find a permutation  $\pi$  such that  $\pi(a) = a'$ .  $\square$

Now we are ready to prove Theorem 7.

*Proof of Theorem 7.* First note that for any  $t \geq 0$ , the parameters  $\tilde{\boldsymbol{\theta}}_t$  or equivalently its flattened form  $\mathbf{w}(t)$  is bounded. Then using Lemma 31, we have

$$\mathbf{w}(t) = \mathbf{P}\mathbf{w}(t).$$

Consider any time  $t \geq 0$ ,  $s \in \mathcal{S}_{\text{ft}}$  and  $\bar{a} := a^*(s, r_2)$ , the prediction probability is given by

$$p_{\tilde{\boldsymbol{\theta}}_t}(\bar{a}|s, r_2) = \frac{\exp(f_{\tilde{\boldsymbol{\theta}}_t}((s, r_2), \bar{a}))}{\sum_{a' \in \mathcal{A}} \exp(f_{\tilde{\boldsymbol{\theta}}_t}((s, r_2), a'))}.$$

We proceed by showing that  $f_{\tilde{\theta}_t}((s, r_2), \bar{a})$  is no larger than  $f_{\tilde{\theta}_t}((s, r_2), a)$  for any  $a \in \mathcal{A}_2$ . WLOG, consider a fixed  $a \in \mathcal{A}_2$  where  $a \neq \bar{a}$ . Comparing the logit functions we get

$$\begin{aligned}
f_{\tilde{\theta}_t}((s, r_2), \bar{a}) - f_{\tilde{\theta}_t}((s, r_2), a) &= W_{\text{KQ}}(s; t) \underbrace{(W_{\text{OV}}(\bar{a}, s; t) - W_{\text{OV}}(a, s; t))}_{A(a, \bar{a}, s; t)} \\
&\quad + W_{\text{KQ}}(r_2; t) \underbrace{(W_{\text{OV}}(\bar{a}, r_2; t) - W_{\text{OV}}(a, r_2; t))}_{B(a, \bar{a}, r_2; t)} \\
&\quad + W_{\text{KQ}}(\langle \text{EOS} \rangle; t) \underbrace{(W_{\text{OV}}(\bar{a}, \langle \text{EOS} \rangle; t) - W_{\text{OV}}(a, \langle \text{EOS} \rangle; t))}_{C(a, \bar{a}, \langle \text{EOS} \rangle; t)} \\
&= 0,
\end{aligned}$$

where  $A(a, \bar{a}, s; t) = 0 = B(a, \bar{a}, r_2; t) = C(a, \bar{a}, \langle \text{EOS} \rangle; t) = 0$  following Lemma 31. As a result, for all  $a \in \mathcal{A}_2$  and any  $t \geq 0$ , we have

$$f_{\tilde{\theta}_t}((s, r_2), \bar{a}) = f_{\tilde{\theta}_t}((s, r_2), a).$$

Thus for any input  $(s, r_2)$  where  $s \in \mathcal{S}_{\text{ft}}$ , its prediction probability can be upper bounded

$$p_{\tilde{\theta}_t}(\bar{a} | s, r_2) = \frac{\exp(f_{\tilde{\theta}_t}((s, r_2), \bar{a}))}{\sum_{a' \in \mathcal{A}} \exp(f_{\tilde{\theta}_t}((s, r_2), a'))} \leq \frac{\exp(f_{\tilde{\theta}_t}((s, r_2), \bar{a}))}{\sum_{a' \in \mathcal{A}_2} \exp(f_{\tilde{\theta}_t}((s, r_2), a'))} = 1/|\mathcal{A}_2|,$$

which implies that

$$\mathcal{L}_{\text{test}}(\tilde{\theta}_t) = \mathbb{E}_{z_{1:T+1} \sim \mathcal{D}_{\text{test}}} [-\log p_{\tilde{\theta}_t}(z_{T+1} | z_{1:T})] \geq \log |\mathcal{A}_2|.$$

□

### B.3 Additional Experiments for One-layer Models

We provide additional experimental results to verify our theoretical results in Section 3.3.2.

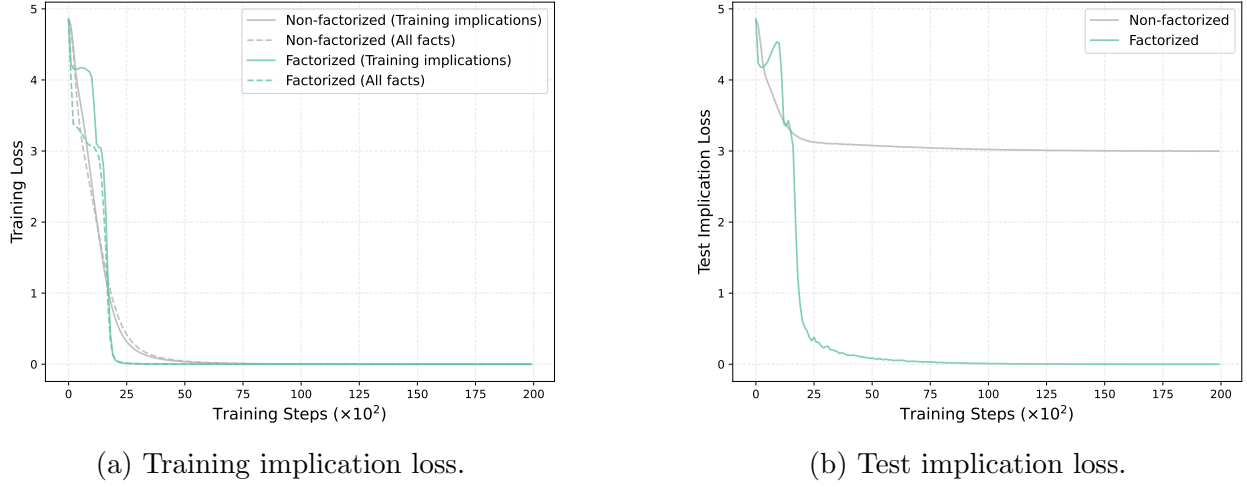


Figure B.1: **Training and Test Implication Loss for Factorized vs. Non-Factorized Models.** While both models effectively minimize the training loss (*left*), their performance on unseen test implications differs starkly (*right*). The factorized model successfully generalizes, achieving low test implication loss and thus demonstrating OCR, while the non-factorized model fails to generalize.

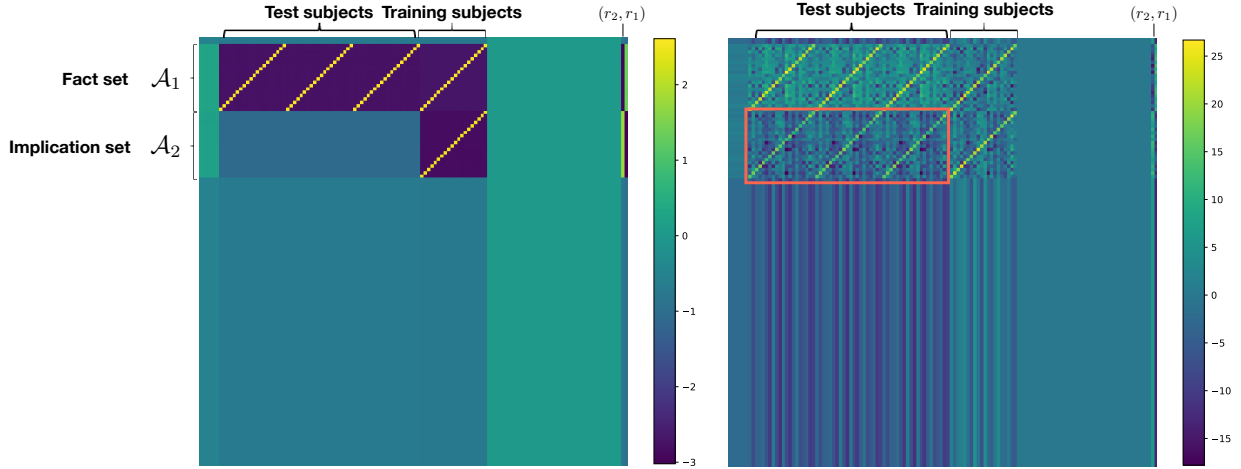


Figure B.2: **Comparison of full weights of trained one-layer linear attention models.** *Left:* Non-factorized model. *Right:* Factorized model. The factorized model shows strong OCR capability compared to the non-factorized model.

**Loss curves.** In Figure B.1, we present the training and test loss curve during training, which shows that the factorized model can exhibit OCR while the non-factorized model

cannot. Recall that the training loss contains two parts: loss on all facts and implications of training subjects (training implication), and the model is evaluated on the implications of test subjects (test implication).

**Full weight inspection.** In Figure 3.2, we only showed partial model weights that are related to prediction. For completeness, we show the full model weights  $\mathbf{W}_{\text{OV}} = \mathbf{W}_{\text{O}}\mathbf{W}_{\text{V}}^{\top}$  in Figure B.2.

**SVM solutions.** We setup the SVM problems defined in ( $\mathbf{W}_{\text{OV}}\text{-SVM}$ ) and ( $\mathbf{W}_{\text{OV}}^{\text{F}}\text{-SVM}$ ) with  $|\mathcal{S}| = 12, n = 3, m = 4, m_{\text{train}} = 3$ . Solutions by CVXPY [33] are shown on the left side of Figure B.3. The results are consistent with the weights of the one-layer attention model in Figure 3.2. Moreover, we decompose the solution of ( $\mathbf{W}_{\text{OV}}^{\text{F}}\text{-SVM}$ ) using SVD and keep the directions with singular values larger than  $10^{-5}$ . This results in  $\mathbf{W}_{\text{OV}}^{\text{F}} = \mathbf{W}_{\text{O}}\mathbf{W}_{\text{V}}^{\top}$  with intrinsic dimension  $d_h = 3$ . On the right of Figure B.3, we visualize the corresponding rows of the subjects and relations in  $\mathbf{W}_{\text{V}}$  as well as the corresponding rows of answers in  $\mathbf{W}_{\text{O}}$ , which suggests that the model generalizes effectively even with a small hidden dimension.

**Lower bound of the intrinsic dimension of output and value matrix.** To further support the claim that the factorized model generalizes effectively with a small hidden dimension, we train a one-layer attention model in Figure B.4 and sweep across multiple candidate values for the intrinsic dimension, i.e.,  $d_h \in \{3, 4, 8, 16, 32, 128\}$  with the embedding dimension  $d = 128$ ,  $m_{\text{train}} = 3$  and other parameters unchanged, which demonstrates that OCR can be achieved efficiently in terms of the number of parameters.

## B.4 Additional LLM Experiments on Real-World Data

To verify our claim in real-world datasets, we extended the LLM experiments in Section 3.2 to PopQA [89], which is a large-scale open-domain question answering (QA) dataset. Each question in PopQA follows the same format as in our synthetic dataset – a knowledge triple (subject, relation, answer). Specifically, we use a subset of PopQA consisting of the *place of birth* (POB) and *sport* relations termed **PopQA-OCR** and treat the first relation as fact and the second as implication. We randomly sample 500 subjects and randomly pair 10 facts from available POBs with 10 implications from available sports in PopQA-OCR. We follow the same data processing scheme as in Section B.5 with a 0.5 training ratio. This new controlled dataset exhibits three key distinctions, compared to the synthetic dataset: 1) We scale up the number of subjects and fact-implication pairs, which results in  $\sim 10x$  training samples compared to the synthetic one. 2) The new task follows a question answering format: “Q: In what city was Antoine Richard born? A: Vinga” while the synthetic one uses a simple text generation formulation. 3) The subject names are real instead of fictitious, which is more likely to collide with the pretrained knowledge, thus inducing new challenges in fine-tuning.

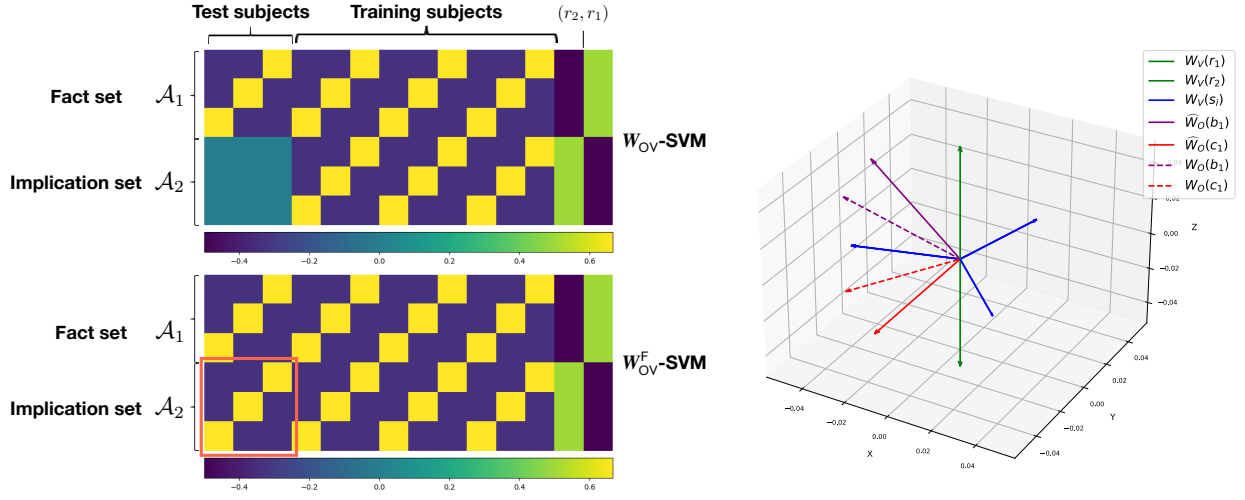


Figure B.3: **Comparison of solutions to  $(\mathbf{W}_{\text{OV-SVM}})$  and  $(\mathbf{W}_{\text{OV}^{\text{F}}\text{-SVM})}$ .** *Top Left:*  $(\mathbf{W}_{\text{OV-SVM}})$  with the Frobenius norm objective. *Bottom Left:*  $(\mathbf{W}_{\text{OV}^{\text{F}}\text{-SVM}})$  with the nuclear norm objective. Here we only show the partial weights in the output-value matrix related to the prediction, i.e.,  $\mathbf{W}_{\text{OV}} \in \mathbb{R}^{|A| \times (mn+2)}$ . *Right:* Geometric interpretation of  $\mathbf{W}_{\text{O}}$  and  $\mathbf{W}_{\text{V}}$  solved in  $(\mathbf{W}_{\text{OV}^{\text{F}}\text{-SVM}})$ . All the subjects' feature vectors (corresponding to rows in  $\mathbf{W}_{\text{V}}$ ) reside in the  $xy$  plane while the relation vectors corresponding to  $r_1$  and  $r_2$  are orthogonal to the subjects and point in opposite directions. The predictions  $\widehat{\mathbf{W}}_{\text{O}}(b_i)$ ,  $\widehat{\mathbf{W}}_{\text{O}}(c_i)$  are made by summing up the feature vector of  $s_i$  with  $r_1$  or  $r_2$ , which aligns well with the features of  $b_i$  or  $c_i$  respectively (plotted in the figure, which are corresponding rows in  $\mathbf{W}_{\text{O}}$ ) with cosine similarity greater than 0.9.

The results can be found in Table B.1, which show that LLMs continue to exhibit OCR-driven hallucination on real-world data.

Table B.1: Performance comparison of different language models on PopQA-OCR. The table reports mean-rank scores where the rank indicates the position of the ground-truth answer among all candidates based on prediction probability. Lower ranks indicate better performance and Rank 0 refers to the token with the largest probability. Values in parentheses indicate the standard error of the mean-rank scores, calculated from 3 runs with different random seeds.

| Models    | Gemma-2-9B  | OLMo-7B     | Qwen-2-7B   | Mistral-7B-v0.3 | Llama-3-8B  |
|-----------|-------------|-------------|-------------|-----------------|-------------|
| PoB-Sport | 0.68 (0.27) | 1.41 (0.29) | 3.30 (0.62) | 1.36 (0.70)     | 2.29 (4.26) |

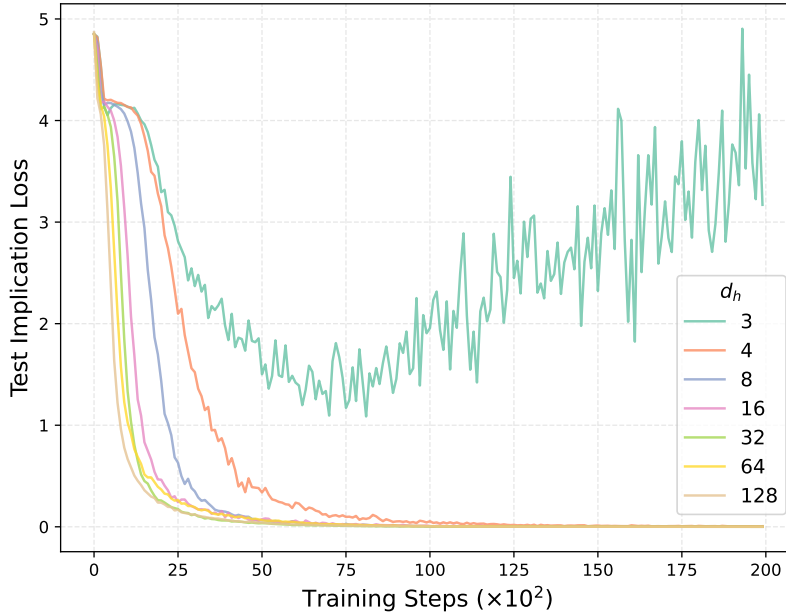


Figure B.4: Test loss versus training steps for a one-layer linear attention model with varying intrinsic dimensions ( $d_h$ ). The plot demonstrates that the model can exhibit OCR even when the intrinsic dimension is as small as  $d_h = 4$ .

## B.5 Implementation Details

Our code is released at <https://github.com/yixiao-huang/OCR-Theory>. We provide additional details on experiments in Section 3.2 and Section 3.3.2.

**Training.** Throughout this work, we finetune the models using the cross-entropy loss with AdamW optimizer [72]. We build on the implementation of [39] for all LLM experiments<sup>1</sup> and adopt a different training scheme for one-layer transformer as discussed in Section 3.3.1. For experiments on LLMs, we use full batch and train for 100 epochs. Similar to [39], we notice that OCR is sensitive to learning rates and thus we sweep across different learning rates in  $\{10^{-6}, 3 \cdot 10^{-6}, 10^{-5}, 3 \cdot 10^{-5}, 10^{-4}, 3 \cdot 10^{-4}\}$  for each model and relation pair and report the results with the lowest test rank. As a complement to the final performance metrics in Table 3.1, Figure B.5 plots the average test rank during training across three different seeds. The shaded region represents the standard deviation. For the one-layer linear attention model, we train the model with one-hot token embedding with  $d = 128$  for  $2 \cdot 10^4$  steps with learning rate  $5 \cdot 10^{-4}$ . We set  $d_h = d = 128$  by default, unless otherwise specified.

<sup>1</sup><https://github.com/jiahai-feng/extractive-structures>

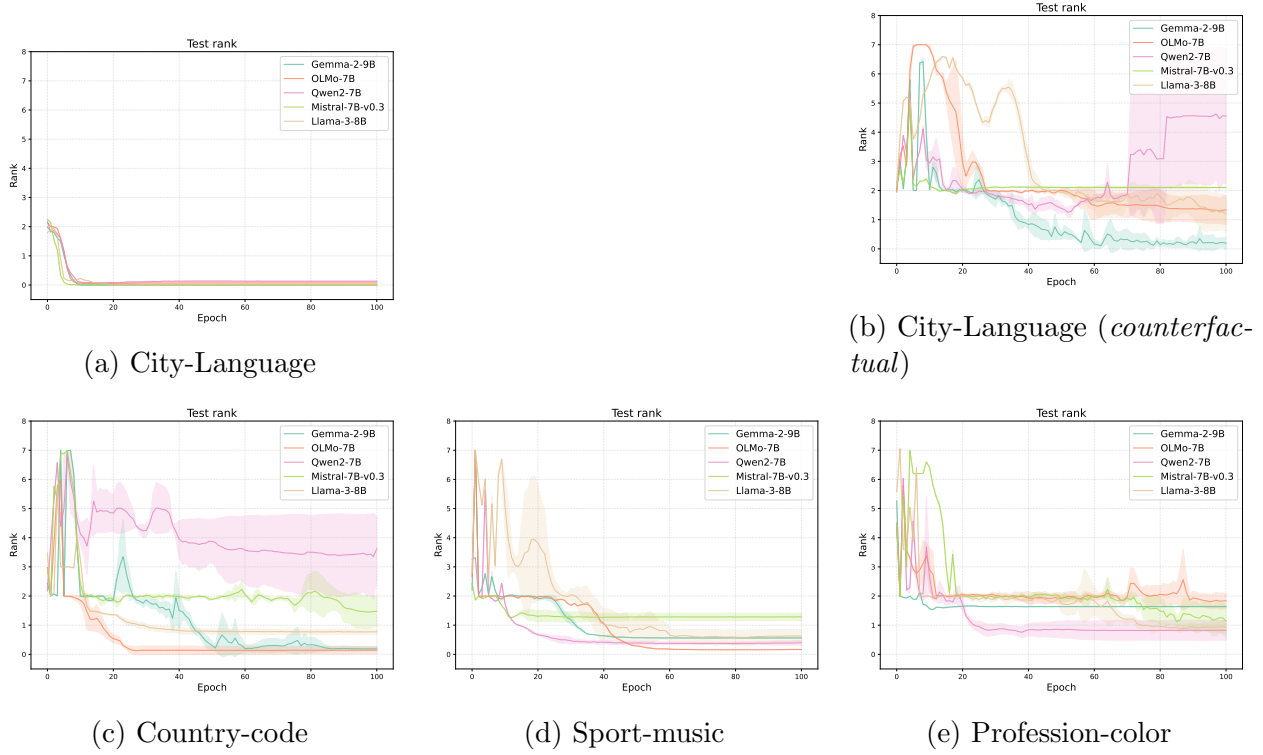


Figure B.5: OCR performance of various LLMs on the five relation pairs. Results are averaged over 3 random seeds, with shaded regions representing the standard deviation.

**Dataset.** The dataset for LLM experiments consists of a list of 100 fictitious names and a list of 5 fact-implication pairs with different topics, namely, city-animal, country-code, sport-music, and profession-color. We split the subjects into training and test with a ratio of 0.2 : 0.8, i.e., there are 4 training subjects in each subset  $\mathcal{S}_i$  for  $i \in [5]$  and 20 training subjects in total. For the experiments in Section 3.2, we use relations listed in Table B.2. For example, for the topic “Country-Code”, one example is given by

- **Fact:** “Daniel Gray was born in Brazil.”
- **Implication:** “Daniel Gray codes in Assembly.”

We use the same name, city and language list in [39]. For the *counterfactual* language, we reorder the language list such that every city corresponds to an incorrect language. For the rest of the topics, we use Claude-3.5-sonnet to generate a list of 5 examples. A complete list is given below:

| Topic      | Relation    |
|------------|-------------|
| City       | lives in    |
| Language   | speaks      |
| Color      | dislikes    |
| Country    | was born in |
| Music      | listens to  |
| Code       | codes in    |
| Sport      | plays       |
| Profession | is a        |

Table B.2: Relation expressions of different topics.

## Topic Lists

*City:*

Tokyo, Beijing, Mumbai, Paris, Berlin

*Language:*

Japanese, Mandarin, Marathi, French, German

*Language (counterfactual):*

Marathi, German, Japanese, Mandarin, French

*Color:*

crimson, teal, navy blue, emerald green, lavender

*Country:*

Japan, Brazil, Morocco, New Zealand, Iceland

*Music:*

jazz, alternative rock, reggae, classical, hip-hop

*Code:*

Python, Julia, Assembly, C, MATLAB

*Sport:*

basketball, soccer, tennis, swimming, volleyball

*Profession:*

nurse practitioner, computer scientist, journalist, veterinarian, social worker

**Computation.** The experiments for the one-layer model were run on a single NVIDIA A100 GPU. LLM Experiments were run on a cluster of 4 NVIDIA A100 GPUs and took less than an hour for each run.

# Appendix C

## Appendix for Chain-of-continuous-thought Theory

### C.1 Notation Details

We provide detailed descriptions of different tokens in Table C.1, and the position index of different tokens or continuous thoughts in Table C.2.

| Tokens              | Meanings                                                                  |
|---------------------|---------------------------------------------------------------------------|
| $\langle s \rangle$ | a special token denoting the beginning of the sentence                    |
| $s_i$               | the source node of edge $i$                                               |
| $t_i$               | the target node of edge $i$                                               |
| $\langle e \rangle$ | a special token marking the end of an edge                                |
| $\langle Q \rangle$ | a special token followed by two candidate nodes                           |
| $c_1, c_2$          | two candidate destination nodes                                           |
| $\langle R \rangle$ | a special token marking the start of reasoning                            |
| $r$                 | the root node                                                             |
| $[\mathbf{t}_i]$    | the $i$ -th continuous thought (represented by a $d$ -dimensional vector) |
| $\langle A \rangle$ | a special token driving the model to make the final prediction            |

Table C.1: Meaning of each token.

| Notations                                   | Position indices  |
|---------------------------------------------|-------------------|
| $\text{Idx}(\langle \mathbf{s} \rangle)$    | 1                 |
| $\text{Idx}(\mathbf{s}_i)$                  | $3i - 1$          |
| $\text{Idx}(\mathbf{t}_i)$                  | $3i$              |
| $\text{Idx}(\langle \mathbf{e} \rangle, i)$ | $3i + 1$          |
| $\text{Idx}(\langle \mathbf{Q} \rangle)$    | $3m + 2$          |
| $\text{Idx}(\mathbf{c}_1)$                  | $3m + 3$          |
| $\text{Idx}(\mathbf{c}_2)$                  | $3m + 4$          |
| $\text{Idx}(\langle \mathbf{R} \rangle)$    | $3m + 5$          |
| $\text{Idx}(\mathbf{r})$                    | $3m + 6 = t_0$    |
| $\text{Idx}([\mathbf{t}_i])$                | $t_0 + i$         |
| $\text{Idx}(\langle \mathbf{A} \rangle)$    | $t_0 + C + 1 = T$ |

Table C.2: Position indices of different tokens or continuous thoughts in the input sequence.

## C.2 Missing Proofs

### C.2.1 Formal version and proof of Lemma 3

**Lemma 32** (Attention chooser, formal version of Lemma 3). *Fix any token  $\langle \mathbf{x} \rangle \in \text{Voc}$ , integer  $\ell \geq 0$ , and  $\varepsilon \in (0, 1)$ . Under sinusoidal positional encoding as defined in Definition 1, there exists a construction of  $\mathbf{W}_K, \mathbf{W}_Q \in \mathbb{R}^{(2d_{PE}) \times d}$ , such that for any input sequence  $(\hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_T)$  that satisfies*

$$\hat{\mathbf{v}}_i = \sum_{v \in \text{Voc}} \lambda_{i,v} \mathbf{u}_v, \text{ where } \lambda_{i,v} \geq 0 \quad \forall v \in \text{Voc}, \sum_{v \in \text{Voc}} \lambda_{i,v}^2 = 1, \quad \forall i \in [T], \quad (\text{C.1})$$

*and satisfies  $\langle \mathbf{u}_{\langle \mathbf{x} \rangle}, \hat{\mathbf{v}}_i \rangle \in \{0, 1\}$  (i.e., each input embedding is either equal to the embedding of token  $\langle \mathbf{x} \rangle$  or orthogonal to it) and  $\langle \mathbf{u}_{\langle \mathbf{x} \rangle}, \hat{\mathbf{v}}_i \rangle = 0$  for  $i \leq \ell$  (i.e., the first  $\ell$  tokens are not  $\langle \mathbf{x} \rangle$ ), it holds that for any  $i \in [T]$ ,*

$$\text{if } \langle \hat{\mathbf{v}}_i, \mathbf{u}_{\langle \mathbf{x} \rangle} \rangle = 1, \text{ then } s_{i,i-\ell} > 1 - \varepsilon, \text{ otherwise } s_{i,1} > 1 - \varepsilon,$$

*where  $s_{i,j}$  is the attention score from the  $i$ -th token to the  $j$ -th token as defined in Algorithm 2 with the input sequence  $(\hat{\mathbf{v}}_1 + \mathbf{p}_1, \dots, \hat{\mathbf{v}}_T + \mathbf{p}_T)$ .*

*Proof.* Note that (C.1) implies that each input embedding is a normalized superposition of token embeddings in the vocabulary. We aim to construct an attention head such that when the  $i$ -th token is  $\langle \mathbf{x} \rangle$ , it will pay almost all attention to the position  $i - \ell$ , otherwise it will pay almost all attention to the BOS token  $\langle \mathbf{s} \rangle$  (known as the attention sink). Define vector  $\tilde{\mathbf{u}}_{\langle \mathbf{x} \rangle} = \sum_{v \in \text{Voc} \setminus \{\langle \mathbf{x} \rangle\}} \tilde{\mathbf{u}}_v \in \mathbb{R}^{d_{TE}}$ , which is the superposition of all token embeddings in the

vocabulary except for  $\langle \mathbf{x} \rangle$ . We define

$$\begin{aligned} \mathbf{W}_Q &= \begin{bmatrix} \mathbf{0}_{d_{PE} \times d_{TE}} & \mathbf{0}_{d_{PE} \times d_{TE}} & \mathbf{0}_{d_{PE} \times d_{TE}} & \mathbf{I}_{d_{PE}} \\ \xi \bar{\mathbf{p}}_1 \otimes \tilde{\mathbf{u}}_{\langle \bar{\mathbf{x}} \rangle} & \mathbf{0}_{d_{PE} \times d_{TE}} & \mathbf{0}_{d_{PE} \times d_{TE}} & \mathbf{0}_{d_{PE} \times d_{PE}} \end{bmatrix} \in \mathbb{R}^{(2d_{PE}) \times d}, \\ \mathbf{W}_K &= \begin{bmatrix} \mathbf{0}_{d_{PE} \times d_{TE}} & \mathbf{0}_{d_{PE} \times d_{TE}} & \mathbf{0}_{d_{PE} \times d_{TE}} & \eta \mathbf{R}^{(\ell)} \\ \mathbf{0}_{d_{PE} \times d_{TE}} & \mathbf{0}_{d_{PE} \times d_{TE}} & \mathbf{0}_{d_{PE} \times d_{TE}} & \eta \mathbf{I}_{d_{PE}} \end{bmatrix} \in \mathbb{R}^{(2d_{PE}) \times d}, \end{aligned}$$

where  $\xi, \eta > 0$  will be specified later and  $\mathbf{R}^{(\ell)}$  is defined as in Lemma 33. Therefore,

$$\mathbf{q}_i = \mathbf{W}_Q(\hat{\mathbf{v}}_i + \mathbf{p}_i) = \begin{bmatrix} \bar{\mathbf{p}}_i \\ \xi \langle \tilde{\mathbf{u}}_{\langle \bar{\mathbf{x}} \rangle}, \tilde{\mathbf{v}}_i \rangle \bar{\mathbf{p}}_1 \end{bmatrix}, \quad \mathbf{k}_i = \mathbf{W}_K(\hat{\mathbf{v}}_i + \mathbf{p}_i) = \begin{bmatrix} \eta \mathbf{R}^{(\ell)} \bar{\mathbf{p}}_i \\ \eta \bar{\mathbf{p}}_i \end{bmatrix} = \begin{bmatrix} \eta \bar{\mathbf{p}}_{i+\ell} \\ \eta \bar{\mathbf{p}}_i \end{bmatrix}.$$

Now for any  $1 \leq j \leq i \leq T$ , we have

$$\langle \mathbf{q}_i, \mathbf{k}_j \rangle = \eta \left( \langle \bar{\mathbf{p}}_i, \bar{\mathbf{p}}_{j+\ell} \rangle + \xi \langle \tilde{\mathbf{u}}_{\langle \bar{\mathbf{x}} \rangle}, \tilde{\mathbf{v}}_i \rangle \langle \bar{\mathbf{p}}_1, \bar{\mathbf{p}}_j \rangle \right).$$

Now we fix  $i \in [T]$ . We first consider the case where  $\langle \hat{\mathbf{v}}_i, \mathbf{u}_{\langle \mathbf{x} \rangle} \rangle = 1$  (which also implies  $i > \ell$ ). By (C.1) and the assumption that token embeddings are orthonormal, we have

$$\langle \tilde{\mathbf{v}}_i, \tilde{\mathbf{u}}_v \rangle = \langle \hat{\mathbf{v}}_i, \mathbf{u}_v \rangle = 0, \quad \forall v \in \text{Voc} \setminus \{\langle \mathbf{x} \rangle\} \implies \langle \tilde{\mathbf{v}}_i, \tilde{\mathbf{u}}_{\langle \bar{\mathbf{x}} \rangle} \rangle = 0.$$

Therefore, we have  $\langle \mathbf{q}_i, \mathbf{k}_j \rangle = \eta \langle \bar{\mathbf{p}}_i, \bar{\mathbf{p}}_{j+\ell} \rangle$ . By Lemma 34, we have

$$\langle \mathbf{q}_i, \mathbf{k}_{i-\ell} \rangle = \eta \langle \bar{\mathbf{p}}_i, \bar{\mathbf{p}}_{(i-\ell)+\ell} \rangle = \eta \langle \bar{\mathbf{p}}_i, \bar{\mathbf{p}}_i \rangle = \eta d_{PE}/2$$

and

$$\langle \mathbf{q}_i, \mathbf{k}_j \rangle = \eta \langle \bar{\mathbf{p}}_i, \bar{\mathbf{p}}_{j+\ell} \rangle \leq \eta d_{PE}/2 - \eta \varepsilon_T, \quad \forall j \neq i - \ell.$$

where  $\varepsilon_T > 0$  is defined in Lemma 34. This implies  $\langle \mathbf{q}_i, \mathbf{k}_j \rangle$  is maximized when  $j = i - \ell$  with a non-zero gap  $\eta \varepsilon_T$ , and therefore, we have

$$s_{i,i-\ell} = \frac{\exp(\langle \mathbf{q}_i, \mathbf{k}_{i-\ell} \rangle)}{\sum_{j \in [i]} \exp(\langle \mathbf{q}_i, \mathbf{k}_j \rangle)} \geq \frac{\exp(\eta \varepsilon_T)}{\exp(\eta \varepsilon_T) + (i-1)}. \quad (\text{C.2})$$

Now we consider the case where  $\langle \hat{\mathbf{v}}_i, \mathbf{u}_{\langle \mathbf{x} \rangle} \rangle = 0$ . Again, by (C.1) and the orthonormal assumption of token embeddings, we have

$$\langle \tilde{\mathbf{v}}_i, \tilde{\mathbf{u}}_{\langle \bar{\mathbf{x}} \rangle} \rangle = \sum_{v \in \text{Voc} \setminus \{\langle \mathbf{x} \rangle\}} \langle \hat{\mathbf{v}}_i, \mathbf{u}_v \rangle = \sum_{v \in \text{Voc} \setminus \{\langle \mathbf{x} \rangle\}} \lambda_{i,v} \geq \sum_{v \in \text{Voc} \setminus \{\langle \mathbf{x} \rangle\}} \lambda_{i,v}^2 = 1.$$

By Lemma 34, we can define  $\xi = \max_{j=2,\dots,T} \frac{2d_{PE}}{\langle \bar{\mathbf{p}}_1, \bar{\mathbf{p}}_1 \rangle - \langle \bar{\mathbf{p}}_1, \bar{\mathbf{p}}_j \rangle}$ , and thus

$$\xi(\langle \bar{\mathbf{p}}_1, \bar{\mathbf{p}}_1 \rangle - \langle \bar{\mathbf{p}}_1, \bar{\mathbf{p}}_j \rangle) \geq 2d_{PE}, \quad \forall j \in \{2, \dots, T\}.$$

Note that for any  $j \in \{2, \dots, T\}$ , we have

$$\begin{aligned}
 & \langle \mathbf{q}_i, \mathbf{k}_1 \rangle - \langle \mathbf{q}_i, \mathbf{k}_j \rangle \\
 &= \eta \left( \xi \langle \tilde{\mathbf{u}}_{\langle \bar{\mathbf{x}} \rangle}, \tilde{\mathbf{v}}_i \rangle (\langle \bar{\mathbf{p}}_1, \bar{\mathbf{p}}_1 \rangle - \langle \bar{\mathbf{p}}_1, \bar{\mathbf{p}}_j \rangle) - (\langle \bar{\mathbf{p}}_i, \bar{\mathbf{p}}_{j+\ell} \rangle - \langle \bar{\mathbf{p}}_i, \bar{\mathbf{p}}_{1+\ell} \rangle) \right) \\
 &\geq \eta (2d_{\text{PE}} - d_{\text{PE}}) \\
 &= \eta d_{\text{PE}}.
 \end{aligned}$$

Therefore,

$$s_{i,1} = \frac{\exp(\langle \mathbf{q}_i, \mathbf{k}_1 \rangle)}{\sum_{j \in [i]} \exp(\langle \mathbf{q}_i, \mathbf{k}_j \rangle)} \geq \frac{\exp(\eta d_{\text{PE}})}{\exp(\eta d_{\text{PE}}) + (i-1)}. \quad (\text{C.3})$$

By choosing a sufficiently large  $\eta$ , the lower bound of (C.2) and (C.3) will both exceed  $1 - \varepsilon$  and thus the proof is complete.  $\square$

### C.2.2 Proof of Lemma 4

Note that in the proof sketch of Lemma 4, we showed how to prove the lemma by induction. Here, we use an alternative way that only requires a single forward pass of the transformer. Note that the continuous thoughts are generated in an autoregressive manner, i.e., for an input sequence  $\hat{\mathbf{v}}_{[T]} = (\hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_T)$ , we have  $\hat{\mathbf{v}}_t^{(L)} = \text{TF}_\theta(\hat{\mathbf{v}}_{[t]})$ , where  $\hat{\mathbf{v}}_t^{(L)}$  is defined in Algorithm 1 with the input sequence  $\hat{\mathbf{v}}_{[T]}$ . This means for a sequence of length  $t$ , appending additional vectors at the end of the sequence does not affect the computation of the first  $t$  positions. This means, instead of proving the following property inductively for each  $c = 0, 1, \dots, C-1$ :

$$[\mathbf{t}_{c+1}] = \text{TF}_\theta(\hat{\mathbf{v}}_{[t_0]}, [\mathbf{t}_1], \dots, [\mathbf{t}_c])$$

where for each  $c$ , it holds  $[\mathbf{t}_c] = \frac{1}{\sqrt{|\mathcal{V}_c|}} \sum_{v \in \mathcal{V}_c} \mathbf{u}_v$  and  $\hat{\mathbf{v}}_{[t_0]}$  is defined as in Section 4.3, we can instead prove by a single forward pass by setting the input embedding

$$\hat{\mathbf{v}}_{t_0+c} = \frac{1}{\sqrt{|\mathcal{V}_c|}} \sum_{v \in \mathcal{V}_c} \mathbf{u}_v, \quad \forall c \in [C], \quad (\text{C.4})$$

and additionally setting  $\hat{\mathbf{v}}_T = \hat{\mathbf{v}}_{t_0+C+1} = \mathbf{u}_{\langle \mathbf{A} \rangle}$ , and prove the hidden embedding in the last layer with input  $\hat{\mathbf{v}}_{[T]}$  satisfies

$$\hat{\mathbf{v}}_{t_0+c}^{(L)} = \frac{1}{\sqrt{|\mathcal{V}_{c+1}|}} \sum_{v \in \mathcal{V}_{c+1}} \mathbf{u}_v, \quad \forall 0 \leq c \leq C.$$

In Section C.2.3, we construct the parameter for each component of the two-layer transformer. Finally, Proposition 8 precisely provides the result we desire.

### C.2.3 Construction of transformer parameters

**Proposition 5** (First layer attention). *For any fixed  $\varepsilon \in (0, 1/2)$ , there exists a construction of key and query matrices for first layer attention heads  $\{\mathbf{W}_Q^{(0,h)}, \mathbf{W}_K^{(0,h)}\}_{h=0,\dots,4}$  s.t. for any input embeddings  $\hat{\mathbf{v}} = (\hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_t)$  satisfying the format specified in Section C.2.2, the value of following terms exceed  $1 - \varepsilon$  for all  $i \in [t]$ :*

- $s_{i,i-2}^{(0,0)}$  if  $\hat{\mathbf{v}}_i = \mathbf{u}_{\langle e \rangle}$ , and  $s_{i,i-1}^{(0,0)}$  otherwise;  $s_{i,i-1}^{(0,1)}$  if  $\hat{\mathbf{v}}_i = \mathbf{u}_{\langle e \rangle}$ , and  $s_{i,1}^{(0,1)}$  otherwise;
- $s_{i,i-2}^{(0,2)}$  if  $\hat{\mathbf{v}}_i = \mathbf{u}_{\langle R \rangle}$ , and  $s_{i,1}^{(0,2)}$  otherwise;  $s_{i,i-1}^{(0,3)}$  if  $\hat{\mathbf{v}}_i = \mathbf{u}_{\langle R \rangle}$ , and  $s_{i,1}^{(0,3)}$  otherwise;
- $s_{i,i-1}^{(0,4)}$  if  $\hat{\mathbf{v}}_i = \mathbf{u}_{\langle A \rangle}$ , and  $s_{i,1}^{(0,4)}$  otherwise;

where  $s_i^{(0,h)} \leftarrow \text{SoftMax}(\langle \mathbf{q}_i^{(h)}, \mathbf{k}_1^{(h)} \rangle, \dots, \langle \mathbf{q}_i^{(h)}, \mathbf{k}_i^{(h)} \rangle) \in \mathbb{R}^i$  and  $\mathbf{k}_i^{(h)} = \mathbf{W}_K^{(0,h)} \hat{\mathbf{v}}_i^{(0)}$ ,  $\mathbf{q}_i^{(h)} = \mathbf{W}_Q^{(0,h)} \hat{\mathbf{v}}_i^{(0)}$ ,  $\hat{\mathbf{v}}_i^{(0)} = \hat{\mathbf{v}}_i + \mathbf{p}_i$ .

*Proof.* Note that each of the attention heads is an attention chooser as defined in Lemma 32. Therefore, the proposition directly holds by constructing each attention head as described in Lemma 32.  $\square$

By Proposition 5, each edge token  $\langle e \rangle$  will pay attention to its corresponding source node and target node. Also, the reasoning token  $\langle R \rangle$  will pay attention to two candidate destination nodes, and the answer token  $\langle A \rangle$  will pay attention to the last continuous thought. When no token or position needs to be paid attention to for some attention heads, it will dump attention to the BOS token  $\langle s \rangle$ , a phenomenon known as attention sink. Since the attention after softmax cannot exactly be zero, there will be undesired tokens with noise-level magnitude copied to each position. The subsequent MLP layer will filter out the noise. We formalize the above statements rigorously in the following proposition.

**Proposition 6** (First layer MLP). *When the input sequence  $\hat{\mathbf{v}}_{[T]}$  satisfies conditions in Section C.2.2, there exists a construction of  $\theta_{\text{Attn}}^{(0,h)}$  for  $h \in \{0, \dots, 4\}$ ,  $\theta_{\text{MLP}}^{(0)}$  such that the output of the first layer  $\hat{\mathbf{v}}^{(1)}$  satisfies:*

- $\hat{\mathbf{v}}_{\text{Idx}(\langle e \rangle, i)}^{(1)} = \frac{1}{\sqrt{3}} [\tilde{\mathbf{u}}_{\langle e \rangle}^\top \tilde{\mathbf{u}}_{s_i}^\top \tilde{\mathbf{u}}_{t_i}^\top \mathbf{0}_{d_{\text{PE}}}^\top]^\top, \quad \forall i \in [m]$
- $\hat{\mathbf{v}}_{\text{Idx}(\langle R \rangle)}^{(1)} = \frac{1}{\sqrt{3}} [\tilde{\mathbf{u}}_{\langle R \rangle}^\top \mathbf{0}_{d_{\text{TE}}}^\top (\tilde{\mathbf{u}}_{c_1} + \tilde{\mathbf{u}}_{c_2})^\top \mathbf{0}_{d_{\text{PE}}}^\top]^\top$
- $\hat{\mathbf{v}}_{\text{Idx}(\langle A \rangle)}^{(1)} = \frac{1}{\sqrt{|\mathcal{V}_C|+1}} [\tilde{\mathbf{u}}_{\langle A \rangle}^\top \sum_{v \in \mathcal{V}_C} \tilde{\mathbf{u}}_v^\top \mathbf{0}_{d_{\text{TE}}}^\top \mathbf{0}_{d_{\text{PE}}}^\top]^\top$
- $\hat{\mathbf{v}}_{\text{Idx}([t_i])}^{(1)} = \frac{1}{\sqrt{|\mathcal{V}_i|}} [\sum_{v \in \mathcal{V}_i} \tilde{\mathbf{u}}_v^\top \mathbf{0}_{d_{\text{TE}}}^\top \mathbf{0}_{d_{\text{TE}}}^\top \mathbf{0}_{d_{\text{PE}}}^\top]^\top, \quad \forall i \in [C]$
- $\hat{\mathbf{v}}_i^{(1)} = \hat{\mathbf{v}}_i$  for other  $i \in [T]$ .

*Proof.* We use the construction in Proposition 5 for  $\{\mathbf{W}_Q^{(0,h)}, \mathbf{W}_K^{(0,h)}\}_{h=0,\dots,4}$ . For each position, after attending to the desired tokens, each attention head will copy the attended tokens to one of the two buffer spaces. Formally, we provide the construction of value and output matrices for each head below. First, the value matrices are constructed as

$$\mathbf{W}_V^{(0,h)} = [\mathbf{I}_{d_{TE}} - \tilde{\mathbf{u}}_{\langle s \rangle} \otimes \tilde{\mathbf{u}}_{\langle s \rangle} \quad \mathbf{0}_{d_{TE} \times (d-d_{TE})}] \in \mathbb{R}^{d_{TE} \times d}, \quad h = 0, \dots, 4.$$

By construction, we have  $\mathbf{v}_i^{(h)} = \mathbf{W}_V^{(0,h)} \hat{\mathbf{v}}_i^{(0)} = \text{content}(\hat{\mathbf{v}}_i) \cdot \mathbb{1}\{i > 1\} = \tilde{\mathbf{v}}_i \cdot \mathbb{1}\{i > 1\}$  for  $i \in [T], h \in \{0, \dots, 4\}$ . This is due to the input format specified in Section C.2.2, which ensures that only  $\hat{\mathbf{v}}_1$  contains  $\tilde{\mathbf{u}}_{\langle s \rangle}$ .

Now we construct output matrices such that the  $h$ -th attention head copies the content of the attended token to buffer 1 for  $h = 0, 4$ , and to buffer 2 for  $h = 1, 2, 3$ . Formally, we construct

$$\begin{aligned} \mathbf{W}_O^{(0,h)} &= [\mathbf{0}_{d_{TE} \times d_{TE}} \quad \mathbf{I}_{d_{TE}} \quad \mathbf{0}_{d_{TE} \times d_{TE}} \quad \mathbf{0}_{d_{TE} \times d_{PE}}]^\top \in \mathbb{R}^{d \times d_{TE}}, \quad h = 0, 4, \\ \mathbf{W}_O^{(0,h)} &= [\mathbf{0}_{d_{TE} \times d_{TE}} \quad \mathbf{0}_{d_{TE} \times d_{TE}} \quad \mathbf{I}_{d_{TE}} \quad \mathbf{0}_{d_{TE} \times d_{PE}}]^\top \in \mathbb{R}^{d \times d_{TE}}, \quad h = 1, 2, 3. \end{aligned}$$

Therefore, it holds that  $\mathbf{W}_O^{(0,h)} \mathbf{v}_i^{(h)} = [\mathbf{0}_{d_{TE}}^\top \quad \tilde{\mathbf{v}}_i^\top \cdot \mathbb{1}\{i > 1\} \quad \mathbf{0}_{d_{TE}}^\top \quad \mathbf{0}_{d_{PE}}^\top]^\top$  for  $h = 0, 4$  and  $\mathbf{W}_O^{(0,h)} \mathbf{v}_i^{(h)} = [\mathbf{0}_{d_{TE}}^\top \quad \mathbf{0}_{d_{TE}}^\top \quad \tilde{\mathbf{v}}_i^\top \cdot \mathbb{1}\{i > 1\} \quad \mathbf{0}_{d_{PE}}^\top]^\top$  for  $h = 1, 2, 3$ , and thus we have

$$\begin{aligned} \text{Attn}_{\theta_{\text{Attn}}^{(0,h)}}(\hat{\mathbf{v}}^{(0)})_i &= \sum_{j=1}^i s_{i,j}^{(0,h)} \mathbf{W}_O^{(0,h)} \mathbf{v}_j^{(h)} = \left[ \mathbf{0}_{d_{TE}}^\top \quad \sum_{j=2}^i s_{i,j}^{(0,h)} \tilde{\mathbf{v}}_j^\top \quad \mathbf{0}_{d_{TE}}^\top \quad \mathbf{0}_{d_{PE}}^\top \right]^\top, \quad h = 0, 4, \\ \text{Attn}_{\theta_{\text{Attn}}^{(0,h)}}(\hat{\mathbf{v}}^{(0)})_i &= \sum_{j=1}^i s_{i,j}^{(0,h)} \mathbf{W}_O^{(0,h)} \mathbf{v}_j^{(h)} = \left[ \mathbf{0}_{d_{TE}}^\top \quad \mathbf{0}_{d_{TE}}^\top \quad \sum_{j=2}^i s_{i,j}^{(0,h)} \tilde{\mathbf{v}}_j^\top \quad \mathbf{0}_{d_{PE}}^\top \right]^\top, \quad h = 1, 2, 3, \end{aligned}$$

where  $s_{i,j}^{(0,h)}$  is defined as in Proposition 5. Therefore, the output at position  $i$  of the first attention layer is

$$\hat{\mathbf{v}}_i^{(0.5)} = \hat{\mathbf{v}}_i^{(0)} + \sum_{h=0}^4 \text{Attn}_{\theta_{\text{Attn}}^{(0,h)}}(\hat{\mathbf{v}}^{(0)})_i = \left[ \tilde{\mathbf{v}}_i^\top \quad \sum_{h=0,4} \sum_{j=2}^i s_{i,j}^{(0,h)} \tilde{\mathbf{v}}_j^\top \quad \sum_{h=1}^3 \sum_{j=2}^i s_{i,j}^{(0,h)} \tilde{\mathbf{v}}_j^\top \quad \text{pos}(\mathbf{p}_i)^\top \right]^\top.$$

Next, we construct the parameters of the MLP of the first layer. For notation convenience and by the input format in Section C.2.2, we can decompose  $\hat{\mathbf{v}}_i = \sum_{j=1}^V \lambda_{i,j}^{(0)} \tilde{\mathbf{u}}_j = \mathbf{U} \boldsymbol{\lambda}_i^{(0)}$  where  $\boldsymbol{\lambda}_i^{(0)} = [\lambda_{i,1}^{(0)}, \lambda_{i,2}^{(0)}, \dots, \lambda_{i,V}^{(0)}]^\top \in \mathbb{R}^V$ . Let

$$\text{MLP}_{\theta_{\text{MLP}}^{(0)}}(\hat{\mathbf{v}}^{(0.5)})_i = \mathbf{W}_2^{(0)} \sigma_\varepsilon^{(0)}(\mathbf{W}_1^{(0)} \hat{\mathbf{v}}_i^{(0.5)}) \in \mathbb{R}^d$$

which is a two-layer neural network. Let  $\mathbf{W}_1^{(0)} = [\text{diag}((\cdot)^\top \mathbf{U}^\top, \mathbf{U}^\top, \mathbf{U}^\top), \mathbf{0}_{(3V) \times d_{PE}}] \in \mathbb{R}^{3V \times d}$ . Then

$$\mathbf{W}_1^{(0)} \hat{\mathbf{v}}_i^{(0.5)} = \begin{bmatrix} \mathbf{U}^\top \mathbf{U} \boldsymbol{\lambda}_i^{(0)} \\ \mathbf{U}^\top \mathbf{U} \sum_{h=0,4} \sum_{j=2}^i s_{i,j}^{(0,h)} \boldsymbol{\lambda}_j^{(0)} \\ \mathbf{U}^\top \mathbf{U} \sum_{h=1}^3 \sum_{j=2}^i s_{i,j}^{(0,h)} \boldsymbol{\lambda}_j^{(0)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\lambda}_i^{(0)} \\ \sum_{h=0,4} \sum_{j=2}^i s_{i,j}^{(0,h)} \boldsymbol{\lambda}_j^{(0)} \\ \sum_{h=1}^3 \sum_{j=2}^i s_{i,j}^{(0,h)} \boldsymbol{\lambda}_j^{(0)} \end{bmatrix} \in \mathbb{R}^{3V}.$$

Let  $\sigma_\varepsilon^{(0)}(\cdot)$  be a coordinate-wise non-linearity such that  $\sigma_\varepsilon^{(0)}(x) = \mathbb{1}\{x \geq \varepsilon\}$  for  $x \in \mathbb{R}$ . We choose  $\varepsilon = \frac{1}{2\sqrt{n}}$  where  $n$  is the (maximum possible) number of vertices in the graph. We denote  $i_*^{(h)} = \arg \max_{j \leq i} s_{i,j}^{(0,h)}$ , i.e.,  $i_*^{(h)}$  is the position that position  $i$  pays the most attention within head  $h$ . By Proposition 5, we can construct key and query matrices such that  $s_{i,i_*^{(h)}}^{(0,h)} > 1 - \varepsilon/5$  for any  $i$  and  $h$ . Also, by the input format especially by (C.4),  $\lambda_{i,k}^{(0)} \in \{0, 1\} \cup \{1/\sqrt{i} \mid i \in [n]\}$ , which implies that any non-zero  $\lambda_{i,k}^{(0)}$  satisfies  $\lambda_{i,k}^{(0)} \in [2\varepsilon, 1]$  and all non-zero entries of  $\boldsymbol{\lambda}_i^{(0)}$  share the same value for any fixed  $i$ . Then by definition of  $\sigma_\varepsilon^{(0)}(\cdot)$ , we have  $\sigma_\varepsilon^{(0)}(\boldsymbol{\lambda}_i^{(0)}) = \frac{\boldsymbol{\lambda}_i^{(0)}}{\|\boldsymbol{\lambda}_i^{(0)}\|_\infty}$ .

Now we analyze  $\sum_{j=2}^i s_{i,j}^{(0,h)} \boldsymbol{\lambda}_j^{(0)}$ . For any  $k \in [V]$ , we consider  $\sum_{j=2}^i s_{i,j}^{(0,h)} \lambda_{j,k}^{(0)}$ . If  $i_*^{(h)} = 1$ , then we have

$$\sum_{j=2}^i s_{i,j}^{(0,h)} \lambda_{j,k}^{(0)} \leq \left( \sum_{j=2}^i s_{i,j}^{(0,h)} \right) \cdot \max_{2 \leq j \leq i} \lambda_{j,k}^{(0)} < \frac{\varepsilon}{5} \cdot 1 = \frac{\varepsilon}{5}.$$

If  $i_*^{(h)} > 1$ , we have

$$\sum_{j=2}^i s_{i,j}^{(0,h)} \lambda_{j,k}^{(0)} = s_{i,i_*^{(h)}}^{(0,h)} \lambda_{i_*^{(h)},k}^{(0)} + \sum_{j=2, j \neq i_*^{(h)}}^i s_{i,j}^{(0,h)} \lambda_{j,k}^{(0)}.$$

When  $\lambda_{i_*^{(h)},k}^{(0)} = 0$ , we can obtain that

$$\sum_{j=2}^i s_{i,j}^{(0,h)} \lambda_{j,k}^{(0)} = \sum_{j=2, j \neq i_*^{(h)}}^i s_{i,j}^{(0,h)} \lambda_{j,k}^{(0)} \leq \left( \sum_{j=2, j \neq i_*^{(h)}}^i s_{i,j}^{(0,h)} \right) \cdot \max_{2 \leq j \leq i} \lambda_{j,k}^{(0)} < \frac{\varepsilon}{5}.$$

When  $\lambda_{i_*^{(h)},k}^{(0)} > 0$ , we can obtain that

$$\sum_{j=2}^i s_{i,j}^{(0,h)} \lambda_{j,k}^{(0)} \geq s_{i,i_*^{(h)}}^{(0,h)} \lambda_{i_*^{(h)},k}^{(0)} \geq (1 - \varepsilon/5) \cdot 2\varepsilon \geq \varepsilon.$$

Therefore,  $\sigma_\varepsilon^{(0)}(\sum_{h=0,4} \sum_{j=2}^i s_{i,j}^{(0,h)} \lambda_{j,k}^{(0)}) = \mathbb{1} \left\{ \bigcup_{h=0,4} \left( (i_*^{(h)} > 1) \cap (\lambda_{i_*^{(h)},k}^{(0)} > 0) \right) \right\}$ ,  $\forall k \in [V]$ . Also, by Proposition 5, for any  $i \in [T]$  and  $k \in [V]$ , there is at most one  $h \in \{0, \dots, 4\}$  such that  $i_*^{(h)} > 1$  and  $\lambda_{i_*^{(h)},k}^{(0)} > 0$  hold simultaneously. Therefore,  $\sigma_\varepsilon^{(0)}(\sum_{h=0,4} \sum_{j=2}^i s_{i,j}^{(0,h)} \lambda_{j,k}^{(0)}) = \sum_{h=0,4} \mathbb{1}\{i_*^{(h)} > 1\} \cdot \mathbb{1}\{\lambda_{i_*^{(h)},k}^{(0)} > 0\}$ ,  $\forall k \in [V]$ , and thus we can write this compactly

$$\sigma_\varepsilon^{(0)}\left(\sum_{h=0,4} \sum_{j=2}^i s_{i,j}^{(0,h)} \boldsymbol{\lambda}_j^{(0)}\right) = \sum_{h=0,4} \mathbb{1}\{i_*^{(h)} > 1\} \cdot \mathbb{1}\{\boldsymbol{\lambda}_{i_*^{(h)}}^{(0)} > \mathbf{0}_V\}.$$

Similarly, we have

$$\sigma_\varepsilon^{(0)}\left(\sum_{h=1}^3 \sum_{j=2}^i s_{i,j}^{(0,h)} \boldsymbol{\lambda}_j^{(0)}\right) = \sum_{h=1}^3 \mathbb{1}\{i_*^{(h)} > 1\} \cdot \mathbb{1}\{\boldsymbol{\lambda}_{i_*^{(h)}}^{(0)} > \mathbf{0}_V\}.$$

Therefore,

$$\sigma_\varepsilon^{(0)}(\mathbf{W}_1^{(0)} \hat{\mathbf{v}}_i^{(0.5)}) = \begin{bmatrix} \boldsymbol{\lambda}_i^{(0)} / \|\boldsymbol{\lambda}_i^{(0)}\|_\infty \\ \sum_{h=0,4} \mathbb{1}\{i_*^{(h)} > 1\} \cdot \mathbb{1}\{\boldsymbol{\lambda}_{i_*^{(h)}}^{(0)} > \mathbf{0}_V\} \\ \sum_{h=1}^3 \mathbb{1}\{i_*^{(h)} > 1\} \cdot \mathbb{1}\{\boldsymbol{\lambda}_{i_*^{(h)}}^{(0)} > \mathbf{0}_V\} \end{bmatrix} \in \mathbb{R}^{3V}.$$

Finally, we set  $\mathbf{W}_2^{(0)} = \mathbf{W}_1^{(0)\top} = [\text{diag}(\mathbf{U}^\top, \mathbf{U}^\top, \mathbf{U}^\top), \mathbf{0}_{(3V) \times d_{\text{PE}}}]^\top \in \mathbb{R}^{d \times 3V}$ , and thus

$$\text{MLP}_{\theta_{\text{MLP}}^{(0)}}(\hat{\mathbf{v}}^{(0.5)})_i = \begin{bmatrix} \mathbf{U} \boldsymbol{\lambda}_i^{(0)} / \|\boldsymbol{\lambda}_i^{(0)}\|_\infty \\ \mathbf{U} \sum_{h=0,4} \mathbb{1}\{i_*^{(h)} > 1\} \cdot \mathbb{1}\{\boldsymbol{\lambda}_{i_*^{(h)}}^{(0)} > \mathbf{0}_V\} \\ \mathbf{U} \sum_{h=1}^3 \mathbb{1}\{i_*^{(h)} > 1\} \cdot \mathbb{1}\{\boldsymbol{\lambda}_{i_*^{(h)}}^{(0)} > \mathbf{0}_V\} \\ \mathbf{0}_{d_{\text{PE}}} \end{bmatrix} \in \mathbb{R}^d.$$

Now we derive the output of the MLP for different positions  $i$ .

For  $i = \text{Idx}(\langle \mathbf{e} \rangle, k) = 3k + 1$  where  $k \in [m]$ , we have  $i_*^{(0)} = i - 2 = \text{Idx}(\mathbf{s}_k)$  and  $i_*^{(1)} = i - 1 = \text{Idx}(\mathbf{t}_k)$ . Note that  $i_*^{(h)} = 1$  for  $h = 2, 3, 4$ . Also,  $\boldsymbol{\lambda}_{\text{Idx}(\mathbf{s}_k)}^{(0)} = \mathbf{e}_{\mathbf{s}_k}$ ,  $\boldsymbol{\lambda}_{\text{Idx}(\mathbf{t}_k)}^{(0)} = \mathbf{e}_{\mathbf{t}_k}$  and  $\boldsymbol{\lambda}_i^{(0)} = \mathbf{e}_{\langle \mathbf{e} \rangle}$  are all  $V$ -dimensional one-hot vectors. Therefore, we have

$$\text{MLP}_{\theta_{\text{MLP}}^{(0)}}(\hat{\mathbf{v}}^{(0.5)})_{\text{Idx}(\langle \mathbf{e} \rangle, k)} = [\tilde{\mathbf{u}}_{\langle \mathbf{e} \rangle}^\top \tilde{\mathbf{u}}_{\mathbf{s}_k}^\top \tilde{\mathbf{u}}_{\mathbf{t}_k}^\top \mathbf{0}_{d_{\text{PE}}}^\top]^\top.$$

For  $i = \text{Idx}(\langle \mathbf{R} \rangle) = 3m + 5$ , we have  $i_*^{(2)} = i - 2 = \text{Idx}(\mathbf{c}_1)$  and  $i_*^{(3)} = i - 1 = \text{Idx}(\mathbf{c}_2)$ . Note that  $i_*^{(h)} = 1$  for  $h = 0, 1, 4$ . Also,  $\boldsymbol{\lambda}_{i-2}^{(0)} = \mathbf{e}_{\mathbf{c}_1}$  and  $\boldsymbol{\lambda}_{i-1}^{(0)} = \mathbf{e}_{\mathbf{c}_2}$ ,  $\boldsymbol{\lambda}_i^{(0)} = \mathbf{e}_{\langle \mathbf{R} \rangle}$ . Therefore,

$$\text{MLP}_{\theta_{\text{MLP}}^{(0)}}(\hat{\mathbf{v}}^{(0.5)})_{\text{Idx}(\langle \mathbf{R} \rangle)} = [\tilde{\mathbf{u}}_{\langle \mathbf{R} \rangle}^\top \mathbf{0}_{d_{\text{TE}}}^\top (\tilde{\mathbf{u}}_{\mathbf{c}_1} + \tilde{\mathbf{u}}_{\mathbf{c}_2})^\top \mathbf{0}_{d_{\text{PE}}}^\top]^\top.$$

For  $i = \text{Idx}(\langle \mathbf{A} \rangle) = t_0 + C + 1$ , we have  $i_*^{(4)} = i - 1$  and  $i_*^{(h)} = 1$  for  $0 \leq h \leq 3$ . Note that  $\boldsymbol{\lambda}_i^{(0)} = \mathbf{e}_{\langle \mathbf{A} \rangle}$ ,  $\boldsymbol{\lambda}_{i-1}^{(0)} = \boldsymbol{\lambda}_{\text{Idx}(\mathbf{t}_c)}^{(0)} = \frac{1}{\sqrt{|\mathcal{V}_C|}} \sum_{v \in \mathcal{V}_C} \mathbf{e}_v$  where  $\mathcal{V}_C$  is the set of vertices reachable from  $\mathbf{r}$  within  $C$  steps. Then we have

$$\text{MLP}_{\theta_{\text{MLP}}^{(0)}}(\hat{\mathbf{v}}^{(0.5)})_{\text{Idx}(\langle \mathbf{A} \rangle)} = \left[ \tilde{\mathbf{u}}_{\langle \mathbf{A} \rangle}^\top \sum_{v \in \mathcal{V}_C} \tilde{\mathbf{u}}_v^\top \mathbf{0}_{d_{\text{TE}}}^\top \mathbf{0}_{d_{\text{PE}}}^\top \right]^\top.$$

For  $i = \text{Idx}(\mathbf{t}_c) = t_0 + c$  for  $c \in [C]$ , we have  $i_*^{(h)} = 1$  for all  $h$  and  $\boldsymbol{\lambda}_i^{(0)} = \frac{1}{\sqrt{|\mathcal{V}_c|}} \sum_{v \in \mathcal{V}_c} \mathbf{e}_v$ , and thus

$$\text{MLP}_{\theta_{\text{MLP}}^{(0)}}(\hat{\mathbf{v}}^{(0.5)})_{\text{Idx}(\mathbf{t}_c)} = \left[ \sum_{v \in \mathcal{V}_c} \tilde{\mathbf{u}}_v^\top \mathbf{0}_{d_{\text{TE}}}^\top \mathbf{0}_{d_{\text{TE}}}^\top \mathbf{0}_{d_{\text{PE}}}^\top \right]^\top.$$

For remaining  $i$ , we have  $i_*^{(h)} = 1$  for all  $h$  and  $\lambda_i^{(0)}$  is one-hot. So

$$\text{MLP}_{\theta_{\text{MLP}}^{(0)}}(\hat{\mathbf{v}}^{(0.5)})_i = \left[ \tilde{\mathbf{v}}_i^\top \mathbf{0}_{d_{\text{TE}}}^\top \mathbf{0}_{d_{\text{TE}}}^\top \mathbf{0}_{d_{\text{PE}}}^\top \right]^\top = \hat{\mathbf{v}}_i.$$

By applying layer normalization, we can obtain the final result.  $\square$

Note that Proposition 6 shows that after the first layer, each  $\langle \mathbf{e} \rangle$  token will copy its corresponding source and target token embeddings into its two buffer spaces, respectively. For the second layer, since it is the last layer, we only need to focus on positions for current thoughts  $\text{ldx}([\mathbf{t}_c]) = t_0 + c$  and the position for the final prediction  $\text{ldx}(\langle \mathbf{A} \rangle) = T$ . Since we only need one attention head for the second attention layer, we will omit the index for heads and only keep the index for layers.

**Proposition 7** (Second layer attention). *Under the construction of Proposition 6 and input format specified in Section C.2.2, for any fixed  $\varepsilon \in (0, 1/2)$ , there exists a construction of the second-layer attention parameters  $\theta_{\text{Attn}}^{(1)} = (\mathbf{W}_{\text{Q}}^{(1)}, \mathbf{W}_{\text{K}}^{(1)}, \mathbf{W}_{\text{V}}^{(1)}, \mathbf{W}_{\text{O}}^{(1)})$  s.t.*

$$\hat{\mathbf{v}}_{\text{ldx}([\mathbf{t}_c])}^{(1.5)} = \left[ \begin{array}{c} \frac{1}{\sqrt{|\mathcal{V}_c|}} \sum_{v \in \mathcal{V}_c} \tilde{\mathbf{u}}_v + \frac{\sum_{j=1, s_j \in \mathcal{V}_c}^m \tilde{\mathbf{u}}_{t_j}}{\sqrt{3|\{j|s_j \in \mathcal{V}_c, j \in [m]\}|}} \\ \mathbf{0}_{d-d_{\text{TE}}} \end{array} \right] + \left[ \begin{array}{c} \mathbf{U}\boldsymbol{\varepsilon}^{(c)} \\ \mathbf{0}_{d-d_{\text{TE}}} \end{array} \right], \quad \forall c \in \{0\} \cup [C],$$

and

$$\hat{\mathbf{v}}_{\text{ldx}(\langle \mathbf{A} \rangle)}^{(1.5)} = \left[ \begin{array}{c} \frac{1}{\sqrt{|\mathcal{V}_C|+1}} \tilde{\mathbf{u}}_{\langle \mathbf{A} \rangle} + \frac{1}{\sqrt{3}}(\tilde{\mathbf{u}}_{c_1} + \tilde{\mathbf{u}}_{c_2}) \\ \frac{1}{\sqrt{|\mathcal{V}_C|+1}} \sum_{v \in \mathcal{V}_C} \tilde{\mathbf{u}}_v \\ \mathbf{0}_{d_{\text{TE}}+d_{\text{PE}}} \end{array} \right] + \left[ \begin{array}{c} \mathbf{U}\boldsymbol{\varepsilon}^{(C+1)} \\ \mathbf{0}_{d_{\text{TE}}} \\ \mathbf{0}_{d_{\text{TE}}+d_{\text{PE}}} \end{array} \right],$$

where  $\boldsymbol{\varepsilon}^{(c)} \in \mathbb{R}^V$  and  $\|\boldsymbol{\varepsilon}^{(c)}\|_\infty < \varepsilon$  for all  $c \in \{0\} \cup [C+1]$ .

*Proof.* In the second attention layer, we aim to construct key and query matrices such that (1) the current thought  $[\mathbf{t}_c]$  will pay attention to all edges if the source node of the edge is contained in the superposition (see Figure 4.3); (2) the answer token  $\langle \mathbf{A} \rangle$  pays large attention to the reasoning token  $\langle \mathbf{R} \rangle$  which stores the two candidate destination tokens in its buffer space (see Figure C.1).

First, we construct

$$\begin{aligned} \mathbf{W}_{\text{Q}}^{(1)} &= [\mathbf{I}_{d_{\text{TE}}} \quad \mathbf{0}_{d_{\text{TE}} \times d_{\text{TE}}} \quad \mathbf{0}_{d_{\text{TE}} \times d_{\text{TE}}} \quad \mathbf{0}_{d_{\text{TE}} \times d_{\text{PE}}}] \in \mathbb{R}^{d_{\text{TE}} \times d}, \\ \mathbf{W}_{\text{K}}^{(1)} &= [\tau \tilde{\mathbf{u}}_{\langle \mathbf{A} \rangle} \otimes \tilde{\mathbf{u}}_{\langle \mathbf{R} \rangle} \quad \tau \mathbf{I}_{d_{\text{TE}}} \quad \mathbf{0}_{d_{\text{TE}} \times d_{\text{TE}}} \quad \mathbf{0}_{d_{\text{TE}} \times d_{\text{PE}}}] \in \mathbb{R}^{d_{\text{TE}} \times d}, \end{aligned}$$

where the value of  $\tau > 0$  will be decided later.

We define  $\mathbf{q}_i = \mathbf{W}_{\text{Q}}^{(1)} \hat{\mathbf{v}}_i^{(1)}$ ,  $\mathbf{k}_i = \mathbf{W}_{\text{K}}^{(1)} \hat{\mathbf{v}}_i^{(1)}$  and let

$$s_{i,j} = \frac{\exp(\langle \mathbf{q}_i, \mathbf{k}_j \rangle)}{\sum_{j' \leq i} \exp(\langle \mathbf{q}_i, \mathbf{k}_{j'} \rangle)}.$$

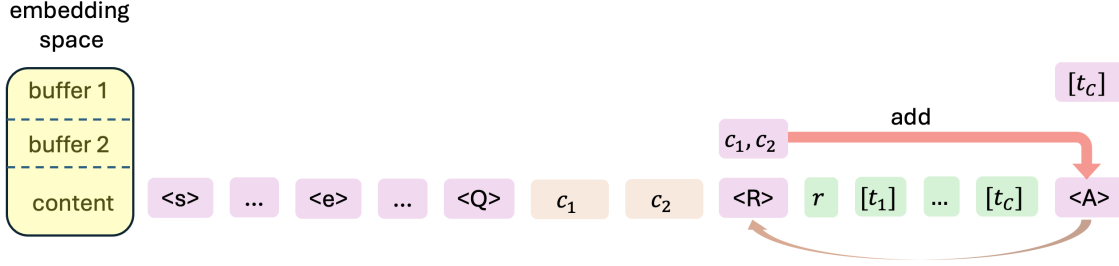


Figure C.1: Illustration of the second layer attention mechanism for final prediction.

Note that we have  $T = \text{ldx}(\langle A \rangle) = t_0 + C + 1$ . By the construction of the query and key matrices, we have  $\mathbf{q}_i = \text{content}(\hat{\mathbf{v}}_i^{(1)})$ ,  $\forall i \in [T]$ ,  $\mathbf{k}_{\text{ldx}(\langle R \rangle)} = \tau \cdot \text{buffer}_1(\hat{\mathbf{v}}_{\text{ldx}(\langle R \rangle)}^{(1)}) + \tau \tilde{\mathbf{u}}_{\langle A \rangle}$  and  $\mathbf{k}_i = \tau \cdot \text{buffer}_1(\hat{\mathbf{v}}_i^{(1)})$  for other  $i$ . Now we consider attention weights for  $i = \text{ldx}([t_c]) = t_0 + c$ ,  $\forall c \in \{0\} \cup [C]$  and  $i = \text{ldx}(\langle A \rangle) = T$ .

For  $i = \text{ldx}([t_c])$  for any fixed  $c \in \{0\} \cup [C]$ , we have  $\mathbf{q}_i = \frac{1}{\sqrt{|\mathcal{V}_c|}} \sum_{v \in \mathcal{V}_c} \tilde{\mathbf{u}}_v$ . By Proposition 6, we have  $\mathbf{k}_{\text{ldx}(\langle e \rangle, j)} = \frac{\tau}{\sqrt{3}} \tilde{\mathbf{u}}_{s_j}$  for  $j \in [m]$ ,  $\mathbf{k}_{\text{ldx}(\langle R \rangle)} = \tau \tilde{\mathbf{u}}_{\langle A \rangle}$  and  $\mathbf{k}_j = 0$  for other  $j \leq i$ . Define  $\mathcal{I}_c = \{\text{ldx}(\langle e \rangle, j) \mid s_j \in \mathcal{V}_c \text{ for } j \in [m]\}$ . Therefore,

$$\langle \mathbf{q}_i, \mathbf{k}_j \rangle = \frac{\tau}{\sqrt{3|\mathcal{V}_c|}} \mathbb{1}\{j \in \mathcal{I}_c\}.$$

Then we have

$$s_{i,j} = \frac{\exp\left(\tau/\sqrt{3|\mathcal{V}_c|}\right) \cdot \mathbb{1}\{j \in \mathcal{I}_c\}}{|\mathcal{I}_c| \exp\left(\tau/\sqrt{3|\mathcal{V}_c|}\right) + (i - |\mathcal{I}_c|)} + \frac{\mathbb{1}\{j \notin \mathcal{I}_c\}}{|\mathcal{I}_c| \exp\left(\tau/\sqrt{3|\mathcal{V}_c|}\right) + (i - |\mathcal{I}_c|)}.$$

For  $i = \text{ldx}(\langle A \rangle) = T$ , note that  $\mathbf{q}_T = \frac{1}{\sqrt{|\mathcal{V}_C|+1}} \tilde{\mathbf{u}}_{\langle A \rangle}$ . Then  $\langle \mathbf{q}_T, \mathbf{k}_j \rangle$  is non-zero only when  $j = \text{ldx}(\langle R \rangle)$ , and the inner product is  $\frac{\tau}{\sqrt{|\mathcal{V}_C|+1}}$ . Then we have

$$s_{T,j} = \frac{\exp\left(\tau/\sqrt{|\mathcal{V}_C|+1}\right) \cdot \mathbb{1}\{j = \text{ldx}(\langle R \rangle)\}}{\exp\left(\tau/\sqrt{|\mathcal{V}_C|+1}\right) + (T - 1)} + \frac{\mathbb{1}\{j \neq \text{ldx}(\langle R \rangle)\}}{\exp\left(\tau/\sqrt{|\mathcal{V}_C|+1}\right) + (T - 1)}.$$

By choosing a large enough  $\tau$ , we have that

$$\begin{aligned} s_{\text{ldx}([t_c]),j} &= \left(\frac{1}{|\mathcal{I}_c|} - \varepsilon_{c,1}\right) \cdot \mathbb{1}\{j \in \mathcal{I}_c\} + \varepsilon_{c,2} \cdot \mathbb{1}\{j \notin \mathcal{I}_c\}, \quad \forall c \in \{0\} \cup [C], \\ s_{T,j} &= (1 - \varepsilon_{C+1,1}) \cdot \mathbb{1}\{j = \text{ldx}(\langle R \rangle)\} + \varepsilon_{C+1,2} \cdot \mathbb{1}\{j \neq \text{ldx}(\langle R \rangle)\}, \end{aligned}$$

where  $\varepsilon_{c,1}, \varepsilon_{c,2} \in (0, \varepsilon/T)$ ,  $\forall c \in \{0\} \cup [C + 1]$ .

Now we construct the value and output matrix where

$$\begin{aligned}\mathbf{W}_V^{(1)} &= [\mathbf{0}_{d_{TE} \times d_{TE}} \quad \mathbf{0}_{d_{TE} \times d_{TE}} \quad \mathbf{I}_{d_{TE}} \quad \mathbf{0}_{d_{TE} \times d_{PE}}] \in \mathbb{R}^{d_{TE} \times d}, \\ \mathbf{W}_O^{(1)} &= [\mathbf{I}_{d_{TE}} \quad \mathbf{0}_{d_{TE} \times (d-d_{TE})}]^\top \in \mathbb{R}^{d \times d_{TE}}.\end{aligned}$$

Then  $\mathbf{v}_i \triangleq \mathbf{W}_V^{(1)} \hat{\mathbf{v}}_i^{(1)} = \text{buffer}_2(\hat{\mathbf{v}}_i^{(1)}) \in \mathbb{R}^{d_{TE}}$  reads from buffer space 2, and  $\mathbf{W}_O^{(1)} \mathbf{v}_i = [\text{buffer}_2(\hat{\mathbf{v}}_i^{(1)})^\top \quad \mathbf{0}_{d-d_{TE}}^\top]^\top \in \mathbb{R}^d$  writes to the content space. Note that

$$\text{Attn}_{\theta_{\text{Attn}}^{(1)}}(\hat{\mathbf{v}}^{(1)})_i = \sum_{j=1}^i s_{i,j} \mathbf{W}_O^{(1)} \mathbf{v}_j = \begin{bmatrix} \sum_{j=1}^i s_{i,j} \text{buffer}_2(\hat{\mathbf{v}}_j^{(1)}) \\ \mathbf{0}_{d-d_{TE}} \end{bmatrix}.$$

Since,  $\hat{\mathbf{v}}_i^{(1.5)} = \hat{\mathbf{v}}_i^{(1)} + \text{Attn}_{\theta_{\text{Attn}}^{(1)}}(\hat{\mathbf{v}}^{(1)})_i$ , we have

$$\hat{\mathbf{v}}_{\text{Idx}(\mathbf{t}_c)}^{(1.5)} = \begin{bmatrix} \frac{1}{\sqrt{|\mathcal{V}_c|}} \sum_{v \in \mathcal{V}_c} \tilde{\mathbf{u}}_v + \left(\frac{1}{|\mathcal{I}_c|} - \varepsilon_{c,1}\right) \sum_{j=1, s_j \in \mathcal{V}_c}^m \frac{\tilde{\mathbf{u}}_{t_j}}{\sqrt{3}} + \varepsilon_{c,2} \sum_{j \in [t_0+c] \setminus \mathcal{I}_c} \text{buffer}_2(\hat{\mathbf{v}}_j^{(1)}) \\ \mathbf{0}_{d-d_{TE}} \end{bmatrix}$$

and

$$\hat{\mathbf{v}}_T^{(1.5)} = \begin{bmatrix} \frac{1}{\sqrt{|\mathcal{V}_C|+1}} \tilde{\mathbf{u}}_{\langle A \rangle} + (1 - \varepsilon_{C+1,1}) \frac{\tilde{\mathbf{u}}_{c_1} + \tilde{\mathbf{u}}_{c_2}}{\sqrt{3}} + \varepsilon_{C+1,2} \sum_{j \in [T] \setminus \{\text{Idx}(\langle R \rangle)\}} \text{buffer}_2(\hat{\mathbf{v}}_j^{(1)}) \\ \frac{1}{\sqrt{|\mathcal{V}_C|+1}} \sum_{v \in \mathcal{V}_C} \tilde{\mathbf{u}}_v \\ \mathbf{0}_{d_{TE}+d_{PE}} \end{bmatrix}.$$

Combining the fact that  $\text{buffer}_2(\hat{\mathbf{v}}_j^{(1)})$  is either zero or equal to the superposition of  $\tilde{\mathbf{u}}_v$  for some  $v \in \text{Voc}$  with norm less than 1, we can obtain the final result.  $\square$

After the second-layer attention, the current thought  $\hat{\mathbf{v}}_{t_0+c}^{(1.5)}$  now contains all vertices in  $\mathcal{V}_c$  and all vertices that can be reached within one step from  $\mathcal{V}_c$ , which is  $\mathcal{V}_{c+1}$  by definition. The subsequent MLP layer will then equalize the weights of each vertex and eliminate the noise in the continuous thought.

**Proposition 8** (Second layer MLP). *Under the construction of Proposition 7 and the input format specified by Section C.2.2, there exists a construction of  $\theta_{\text{MLP}}^{(1)}$  such that the output of the second-layer MLP satisfies:*

- $\hat{\mathbf{v}}_{\text{Idx}(t_c)}^{(2)} = \frac{1}{\sqrt{|\mathcal{V}_{c+1}|}} \sum_{v \in \mathcal{V}_{c+1}} \mathbf{u}_v, \quad \forall c \in \{0\} \cup [C],$
- $\hat{\mathbf{v}}_{\text{Idx}(\langle A \rangle)}^{(2)} = \frac{1}{\sqrt{|\mathcal{V}_C|+5}} (\mathbf{u}_{c_1} + \mathbf{u}_{c_2} + \mathbf{u}_{\langle A \rangle} + \sum_{v \in \mathcal{V}_C} \mathbf{u}_v).$

*Proof.* Similar to Proposition 6, we let

$$\text{MLP}_{\theta_{\text{MLP}}^{(1)}}(\hat{\mathbf{v}}^{(1.5)})_i = \mathbf{W}_2^{(1)} \sigma_\varepsilon^{(1)}(\mathbf{W}_1^{(1)} \hat{\mathbf{v}}_i^{(1.5)}) \in \mathbb{R}^d$$

where  $\varepsilon > 0$  and the (elementwise) non-linearity  $\sigma_\varepsilon^{(1)}(\cdot)$  will be specified later. We only consider  $\hat{\mathbf{v}}_i^{(1.5)}$  where  $i = t_0 + c$  for  $0 \leq c \leq C + 1$ . By Proposition 7, we can decompose

$$\hat{\mathbf{v}}_{t_0+c}^{(1.5)} = \begin{bmatrix} \mathbf{U} \boldsymbol{\lambda}^{(c)} \\ \mathbf{0}_{d_{\text{TE}}} \\ \mathbf{0}_{d_{\text{TE}}+d_{\text{PE}}} \end{bmatrix} \quad \forall c \in \{0\} \cup [C], \quad \hat{\mathbf{v}}_T^{(1.5)} = \begin{bmatrix} \mathbf{U} \boldsymbol{\eta}^{(1)} \\ \mathbf{U} \boldsymbol{\eta}^{(2)} \\ \mathbf{0}_{d_{\text{TE}}+d_{\text{PE}}} \end{bmatrix},$$

where  $\boldsymbol{\lambda}^{(c)} = [\lambda_1^{(c)}, \dots, \lambda_V^{(c)}]^\top$ ,  $\boldsymbol{\eta}^{(1)} = [\eta_1^{(1)}, \dots, \eta_V^{(1)}]^\top$ ,  $\boldsymbol{\eta}^{(2)} = [\eta_1^{(2)}, \dots, \eta_V^{(2)}]^\top \in \mathbb{R}^V$ . Let

$$\mathbf{W}_1^{(1)} = \begin{bmatrix} \mathbf{U}^\top & \mathbf{0}_{V \times (d_{\text{TE}}+d_{\text{PE}})} \\ \mathbf{U}^\top & \mathbf{0}_{V \times (d_{\text{TE}}+d_{\text{PE}})} \end{bmatrix} \in \mathbb{R}^{(2V) \times d}, \quad \mathbf{W}_2^{(1)} = \begin{bmatrix} \mathbf{U} & \mathbf{U} \\ \mathbf{0}_{(d-d_{\text{TE}}) \times V} & \mathbf{0}_{(d-d_{\text{TE}}) \times V} \end{bmatrix} \in \mathbb{R}^{d \times (2V)},$$

and  $\sigma_\varepsilon^{(1)}(x) = \mathbb{1}\{x \geq \varepsilon\}$ . Then

$$\begin{aligned} \text{MLP}_{\theta_{\text{MLP}}^{(1)}}(\hat{\mathbf{v}}^{(1.5)})_{t_0+c} &= \mathbf{W}_2^{(1)} \sigma_\varepsilon^{(1)}(\mathbf{W}_1^{(1)} \hat{\mathbf{v}}_{t_0+c}^{(1.5)}) \\ &= \mathbf{W}_2^{(1)} \sigma_\varepsilon^{(1)}\left(\begin{bmatrix} \mathbf{U}^\top & \mathbf{0}_{V \times (d_{\text{TE}}+d_{\text{PE}})} \\ \mathbf{U}^\top & \mathbf{0}_{V \times (d_{\text{TE}}+d_{\text{PE}})} \end{bmatrix} \begin{bmatrix} \mathbf{U} \boldsymbol{\lambda}^{(c)} \\ \mathbf{0}_{d_{\text{TE}}} \\ \mathbf{0}_{d_{\text{TE}}+d_{\text{PE}}} \end{bmatrix}\right) \\ &= \begin{bmatrix} \mathbf{U} & \mathbf{U} \\ \mathbf{0}_{(d-d_{\text{TE}}) \times V} & \mathbf{0}_{(d-d_{\text{TE}}) \times V} \end{bmatrix} \sigma_\varepsilon^{(1)}\left(\begin{bmatrix} \mathbf{U}^\top \mathbf{U} \boldsymbol{\lambda}^{(c)} \\ \mathbf{0}_V \end{bmatrix}\right) \\ &= \begin{bmatrix} \mathbf{U} \sigma_\varepsilon^{(1)}(\boldsymbol{\lambda}^{(c)}) \\ \mathbf{0}_{d-d_{\text{TE}}} \end{bmatrix}, \end{aligned}$$

and similarly,

$$\begin{aligned} \text{MLP}_{\theta_{\text{MLP}}^{(1)}}(\hat{\mathbf{v}}^{(1.5)})_T &= \mathbf{W}_2^{(1)} \sigma_\varepsilon^{(1)}(\mathbf{W}_1^{(1)} \hat{\mathbf{v}}_T^{(1.5)}) \\ &= \mathbf{W}_2^{(1)} \sigma_\varepsilon^{(1)}\left(\begin{bmatrix} \mathbf{U}^\top & \mathbf{0}_{V \times (d_{\text{TE}}+d_{\text{PE}})} \\ \mathbf{U}^\top & \mathbf{0}_{V \times (d_{\text{TE}}+d_{\text{PE}})} \end{bmatrix} \begin{bmatrix} \mathbf{U} \boldsymbol{\eta}^{(1)} \\ \mathbf{U} \boldsymbol{\eta}^{(2)} \\ \mathbf{0}_{d_{\text{TE}}+d_{\text{PE}}} \end{bmatrix}\right) \\ &= \begin{bmatrix} \mathbf{U} & \mathbf{U} \\ \mathbf{0}_{(d-d_{\text{TE}}) \times V} & \mathbf{0}_{(d-d_{\text{TE}}) \times V} \end{bmatrix} \sigma_\varepsilon^{(1)}\left(\begin{bmatrix} \mathbf{U}^\top \mathbf{U} \boldsymbol{\eta}^{(1)} \\ \mathbf{U}^\top \mathbf{U} \boldsymbol{\eta}^{(2)} \end{bmatrix}\right) \\ &= \begin{bmatrix} \mathbf{U} \left( \sigma_\varepsilon^{(1)}(\boldsymbol{\eta}^{(1)}) + \sigma_\varepsilon^{(1)}(\boldsymbol{\eta}^{(2)}) \right) \\ \mathbf{0}_{d-d_{\text{TE}}} \end{bmatrix}. \end{aligned}$$

Now we choose  $\varepsilon = \frac{1}{4n}$  where  $n = |\mathcal{V}|$  is the number of vertices in the graph. By Proposition 7, we can make sure that  $\|\boldsymbol{\varepsilon}^{(c)}\|_\infty < \frac{1}{16n}$  for all  $c \in \{0\} \cup [C + 1]$  where  $\boldsymbol{\varepsilon}^{(c)}$  is

defined in Proposition 7. We define  $\Delta_c = \{\mathbf{t}_j \mid \mathbf{s}_j \in \mathcal{V}_c, j \in [m]\}$  which is the set of vertices that can be reached within one step from the currently explored vertex set  $\mathcal{V}_c$ . By definition, we have  $\mathcal{V}_{c+1} = \mathcal{V}_c \cup \Delta_c$ . We consider any fixed  $c \in \{0\} \cup [C]$ . For  $v \in \mathcal{V}_c \cup \Delta_c$ , it holds that

$$\lambda_v^{(c)} \geq \min\{1/\sqrt{|\mathcal{V}_c|}, 1/(\sqrt{3}|\Delta_c|)\} + \varepsilon_v^{(c)} \geq \frac{1}{2n} - \frac{1}{16n} > \frac{1}{4n} = \varepsilon.$$

For other  $v$ ,

$$\lambda_v^{(c)} \leq \varepsilon_v^{(c)} < \frac{1}{16n} < \varepsilon.$$

Therefore,  $\sigma_\varepsilon^{(1)}(\lambda_v^{(c)}) = \mathbb{1}\{v \in \mathcal{V}_c \cup \Delta_c\} = \mathbb{1}\{v \in \mathcal{V}_{c+1}\}$ , and thus we have

$$\text{MLP}_{\theta_{\text{MLP}}^{(1)}}(\hat{\mathbf{v}}^{(1.5)})_{t_0+c} = \begin{bmatrix} \sum_{v \in \mathcal{V}_{c+1}} \tilde{\mathbf{u}}_v \\ \mathbf{0}_{d-d_{\text{TE}}} \end{bmatrix}, \quad \forall c \in \{0\} \cup [C].$$

Also, we have  $\eta_{\mathbf{c}_1}^{(1)} = \frac{1}{\sqrt{3}} + \varepsilon_{\mathbf{c}_1}^{(C+1)}$ ,  $\eta_{\mathbf{c}_2}^{(1)} = \frac{1}{\sqrt{3}} + \varepsilon_{\mathbf{c}_2}^{(C+1)}$ ,  $\eta_{\langle \mathbf{A} \rangle}^{(1)} = \frac{1}{\sqrt{|\mathcal{V}_C|+1}} + \varepsilon_{\langle \mathbf{A} \rangle}^{(C+1)}$ , and  $\eta_v^{(1)} < \frac{1}{4n}$  for other  $v \in \text{Voc}$ . Moreover,  $\eta_v^{(2)} = \frac{1}{\sqrt{|\mathcal{V}_C|+1}} \cdot \mathbb{1}\{v \in \mathcal{V}_C\}$ , which implies  $\sigma_\varepsilon^{(1)}(\eta_v^{(2)}) = \mathbb{1}\{v \in \mathcal{V}_C\}$ . Then

$$\text{MLP}_{\theta_{\text{MLP}}^{(1)}}(\hat{\mathbf{v}}^{(1.5)})_T = \begin{bmatrix} \tilde{\mathbf{u}}_{\mathbf{c}_1} + \tilde{\mathbf{u}}_{\mathbf{c}_2} + \tilde{\mathbf{u}}_{\langle \mathbf{A} \rangle} + \sum_{v \in \mathcal{V}_C} \tilde{\mathbf{u}}_v \\ \mathbf{0}_{d-d_{\text{TE}}} \end{bmatrix}.$$

Note that by our construction of the input sequence, one and only one of  $\mathbf{c}_i$  is in  $\mathcal{V}_C$ , and thus  $\|\text{MLP}_{\theta_{\text{MLP}}^{(1)}}(\hat{\mathbf{v}}^{(1.5)})_T\|_2 = \sqrt{|\mathcal{V}_C|+5}$ .

By applying layer normalization, we can obtain the final result.  $\square$

### C.2.4 Proof of the main theorem

*Proof of Theorem 8.* We use the same construction of  $\theta$  as in Lemma 4 where the maximum input length is  $T_{\text{max}}$ . By Lemma 4, we have  $\hat{\mathbf{v}}_{t_0+c} = \frac{1}{\sqrt{|\mathcal{V}_c|}} \sum_{v \in \mathcal{V}_c} \mathbf{u}_v$ , for  $0 \leq c \leq C$ . Then  $(\hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_T)$  satisfies the input format specified in Section C.2.2, and by Proposition 8, we have

$$\begin{aligned} \text{TF}_\theta(\hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_T) &= \hat{\mathbf{v}}_T^{(2)} = \frac{1}{\sqrt{|\mathcal{V}_C|+5}} (\mathbf{u}_{\mathbf{c}_1} + \mathbf{u}_{\mathbf{c}_2} + \mathbf{u}_{\langle \mathbf{A} \rangle} + \sum_{v \in \mathcal{V}_C} \mathbf{u}_v) \\ &= \frac{1}{\sqrt{|\mathcal{V}_C|+5}} (2\mathbf{u}_{\mathbf{c}_{i^*}} + \mathbf{u}_{\mathbf{c}_{3-i^*}} + \mathbf{u}_{\langle \mathbf{A} \rangle} + \sum_{v \in \mathcal{V}_C \setminus \{\mathbf{c}_{i^*}\}} \mathbf{u}_v) \end{aligned}$$

since  $\mathbf{c}_{i^*} \in \mathcal{V}_C$  and  $\mathbf{c}_{3-i^*} \notin \mathcal{V}_C$  by the property the graph reachability problem. We set  $\mathbf{W}_\mathbf{O} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_V]^\top$  and then

$$\begin{aligned} \mathbf{W}_\mathbf{O} \text{TF}_\theta(\hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_T) &= \frac{1}{\sqrt{|\mathcal{V}_C|+5}} (2\mathbf{e}_{\mathbf{c}_{i^*}} + \mathbf{e}_{\mathbf{c}_{3-i^*}} + \mathbf{e}_{\langle \mathbf{A} \rangle} + \sum_{v \in \mathcal{V}_C \setminus \{\mathbf{c}_{i^*}\}} \mathbf{e}_v) \\ \implies \widetilde{\text{TF}}_{\theta, \mathbf{W}_\mathbf{O}}(\hat{\mathbf{v}}_{[t_0]}) &= \arg \max_{v \in \text{Voc}} \mathbf{W}_\mathbf{O} \text{TF}_\theta(\hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_T) = \mathbf{c}_{i^*}. \end{aligned}$$

□

### C.2.5 Properties of sinusoidal positional encodings

**Lemma 33.** *For any integer  $\ell \geq 1$ , there exists a matrix  $\mathbf{R}^{(\ell)} \in \mathbb{R}^{d_{\text{PE}} \times d_{\text{PE}}}$ , such that*

$$\bar{\mathbf{p}}_{i+\ell} = \mathbf{R}^{(\ell)} \bar{\mathbf{p}}_i, \quad \forall i \geq 1.$$

*Proof.* For any  $j \in [d_{\text{PE}}/2]$ , we construction a two-dimensional rotation matrix

$$\mathbf{R}^{(\ell,j)} = \begin{bmatrix} \cos(\ell \cdot \omega^j) & -\sin(\ell \cdot \omega^j) \\ \sin(\ell \cdot \omega^j) & \cos(\ell \cdot \omega^j) \end{bmatrix}.$$

By definition of the rotation matrix, for any  $i \geq 1$ , we have

$$\mathbf{R}^{(\ell,j)} \begin{bmatrix} \bar{p}_{i,2j-1} \\ \bar{p}_{i,2j} \end{bmatrix} = \mathbf{R}^{(\ell,j)} \begin{bmatrix} \cos(i \cdot \omega^j) \\ \sin(i \cdot \omega^j) \end{bmatrix} = \begin{bmatrix} \cos((i+\ell) \cdot \omega^j) \\ \sin((i+\ell) \cdot \omega^j) \end{bmatrix} = \begin{bmatrix} \bar{p}_{i+\ell,2j-1} \\ \bar{p}_{i+\ell,2j} \end{bmatrix}.$$

Then by setting  $\mathbf{R}^{(\ell)} = \text{diag}(\{\mathbf{R}^{(\ell,1)}, \mathbf{R}^{(\ell,2)}, \dots, \mathbf{R}^{(\ell,d_{\text{PE}}/2)}\}) \in \mathbb{R}^{d_{\text{PE}} \times d_{\text{PE}}}$  we can obtain the desired result. □

**Lemma 34.** *For any integer  $T \geq 1$ , there exists  $\varepsilon_T > 0$ , s.t.  $\langle \bar{\mathbf{p}}_i, \bar{\mathbf{p}}_j \rangle \leq d_{\text{PE}}/2 - \varepsilon_T$  for any  $i, j \in [T]$  where  $i \neq j$ . Also,  $\langle \bar{\mathbf{p}}_i, \bar{\mathbf{p}}_i \rangle = d_{\text{PE}}/2$  for all  $i \in [T]$ .*

*Proof.* For any  $i, j \in [T]$ , we have

$$\begin{aligned} \langle \bar{\mathbf{p}}_i, \bar{\mathbf{p}}_j \rangle &= \sum_{k=1}^{d_{\text{PE}}} p_{i,k} p_{j,k} \\ &= \sum_{k=1}^{d_{\text{PE}}/2} (\cos(i \cdot \omega^k) \cos(j \cdot \omega^k) + \sin(i \cdot \omega^k) \sin(j \cdot \omega^k)) \\ &= \sum_{k=1}^{d_{\text{PE}}/2} \cos((i-j)\omega^k). \end{aligned}$$

Note that for  $i = j$ , we can directly obtain that  $\langle p_i, p_j \rangle = d_{\text{PE}}/2$  since  $\cos((i-j)\omega^k) = \cos 0 = 1$  for  $i = j$  and any  $k \in [d_{\text{PE}}/2]$ . Also, since  $(i-j)\omega^k = \frac{i-j}{M^{2k/d_{\text{PE}}}}$  is not a multiplier of  $2\pi$  for  $i \neq j, k \in [d_{\text{PE}}/2]$ , we have  $\cos((i-j)\omega^k) < 1$ . Let

$$\varepsilon_{T,k} = 1 - \max_{i,j \in [T], i \neq j} \cos((i-j)\omega^k) > 0, \quad \forall k \in [d_{\text{PE}}/2],$$

and let  $\varepsilon_T = \sum_{k=1}^{d_{\text{PE}}/2} \varepsilon_{T,k} > 0$ . Therefore, for  $i \neq j$ , we have

$$\langle \bar{\mathbf{p}}_i, \bar{\mathbf{p}}_j \rangle = \sum_{k=1}^{d_{\text{PE}}/2} \cos((i-j)\omega^k) \leq \sum_{k=1}^{d_{\text{PE}}/2} (1 - \varepsilon_{T,k}) = d_{\text{PE}}/2 - \varepsilon_T.$$

□

### C.2.6 Implementing attention chooser under rotary position embedding

In this section, we discuss how to extend our constructions under sinusoidal positional encoding, a widely used absolute positional encoding, to the rotary position embedding (RoPE) [123], a widely used relative positional encoding, to solve graph reachability. Since in our construction, the positional encoding only functions in the first attention layer, and the building blocks of the first attention layers are attention choosers, we mainly focus on how to build the attention chooser block under RoPE.

Since RoPE is a relative positional encoding, we don't use  $\mathbf{p}_i$  in computation and thus don't need the last  $d_{\text{PE}}$  entries in the embedding vectors. Also, since the attention chooser only uses the information in the content space, we omit the two buffer spaces and only keep two dimensions to make our construction clean and assume  $d = d_{\text{TE}} + 2$  in this section for the simplicity of the notation. Therefore, we have  $\mathbf{u}_v = [\tilde{\mathbf{u}}_v^\top, 0, 0]^\top \in \mathbb{R}^d$  for all  $v \in \text{Voc}$  in this section, where  $\{\tilde{\mathbf{u}}_v\}_{v \in \text{Voc}}$  are orthonormal. Also, assume the input embedding of attention layers satisfies  $\hat{\mathbf{v}}_i = [\tilde{\mathbf{v}}_i^\top, 1, 0]^\top \in \mathbb{R}^d$ . This can be achieved easily by adding a bias before the attention layer or modifying the  $(d_{\text{TE}} + 1)$ -th entry of token embeddings.

Recall the definition of RoPE below:

**Definition 3** (Rotary position embedding [123]). *Let  $d$  be even. For any integer  $i$  and any index  $k \in [d/2]$ , we define*

$$\mathbf{R}^{(i,k)} = \begin{bmatrix} \cos(i \cdot \omega^k) & -\sin(i \cdot \omega^k) \\ \sin(i \cdot \omega^k) & \cos(i \cdot \omega^k) \end{bmatrix}.$$

Let

$$\mathbf{R}_{\text{TE}}^{(i)} = \text{diag}(\{\})\mathbf{R}^{(i,1)}, \mathbf{R}^{(i,2)}, \dots, \mathbf{R}^{(i,d_{\text{TE}}/2)} \in \mathbb{R}^{d_{\text{TE}} \times d_{\text{TE}}}$$

and

$$\mathbf{R}^{(i)} = \text{diag}(\{\})\mathbf{R}^{(i,1)}, \mathbf{R}^{(i,2)}, \dots, \mathbf{R}^{(i,d/2)} \in \mathbb{R}^{d \times d},$$

where  $\omega = M^{-2/d}$  and  $M > 0$  is a large constant integer, e.g.,  $M = 10^4$  as chosen in [123]. We make one additional assumption that  $M \geq T$  where  $T$  is the maximum length of the input sequence. Then the query vector  $\mathbf{q}_i$  and key vector  $\mathbf{k}_i$  in Algorithm 2 will be calculated as

$$\mathbf{q}_i = \mathbf{R}^{(i)} \mathbf{W}_{\text{Q}} \hat{\mathbf{v}}_i, \quad \mathbf{k}_i = \mathbf{R}^{(i)} \mathbf{W}_{\text{K}} \hat{\mathbf{v}}_i.$$

Now we show the counterpart of Lemma 32 under RoPE below.

**Lemma 35** (Attention chooser under RoPE). *Fix any token  $\langle \mathbf{x} \rangle \in \text{Voc}$ , integer  $\ell \geq 0$ , and  $\varepsilon \in (0, 1)$ . Under rotary position embedding as defined in Definition 3, there exists a construction of  $\mathbf{W}_{\text{K}}, \mathbf{W}_{\text{Q}} \in \mathbb{R}^{(2d) \times d}$ , such that for any input sequence  $(\hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_T)$  that satisfies*

$$\tilde{\mathbf{v}}_i = \sum_{v \in \text{Voc}} \lambda_{i,v} \tilde{\mathbf{u}}_v, \text{ where } \lambda_{i,v} \geq 0 \quad \forall v \in \text{Voc}, \sum_{v \in \text{Voc}} \lambda_{i,v}^2 = 1, \quad \forall i \in [T], \quad (\text{C.5})$$

and satisfies  $\langle \tilde{\mathbf{u}}_{\langle \mathbf{x} \rangle}, \tilde{\mathbf{v}}_i \rangle \in \{0, 1\}$  (i.e., each input embedding is either equal to the embedding of token  $\langle \mathbf{x} \rangle$  or orthogonal to it) and  $\langle \tilde{\mathbf{u}}_{\langle \mathbf{x} \rangle}, \tilde{\mathbf{v}}_i \rangle = 0$  for  $i \leq \ell$  (i.e., the first  $\ell$  tokens are not  $\langle \mathbf{x} \rangle$ ), it holds that for any  $i \in [T]$ ,

$$\text{if } \langle \tilde{\mathbf{v}}_i, \tilde{\mathbf{u}}_{\langle \mathbf{x} \rangle} \rangle = 1, \text{ then } s_{i,i-\ell} > 1 - \varepsilon, \text{ otherwise } s_{i,1} > 1 - \varepsilon,$$

where  $s_{i,j}$  is the attention score from the  $i$ -th token to the  $j$ -th token as defined in Algorithm 2 with the modification defined in Definition 3 with the input sequence  $(\hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_T)$ .

*Proof.* Note that (C.5) implies that each input embedding is a normalized superposition of token embeddings in the vocabulary (ignoring the last two entries). We aim to construct an attention head such that when the  $i$ -th token is  $\langle \mathbf{x} \rangle$ , it will pay almost all attention to the position  $i - \ell$ , otherwise it will pay almost all attention to the BOS token  $\langle \mathbf{s} \rangle$  (known as the attention sink). We define vector  $\tilde{\mathbf{u}}_{\langle \mathbf{x} \rangle} = \sum_{v \in \text{Voc} \setminus \{\langle \mathbf{x} \rangle\}} \tilde{\mathbf{u}}_v \in \mathbb{R}^d$ , which is the superposition of all token embeddings in the vocabulary except for  $\langle \mathbf{x} \rangle$ . Moreover, for any integer  $i$ , we define  $\mathbf{p}_i^{\text{TE}} = (p_{i,1}, \dots, p_{i,d_{\text{TE}}})^\top \in \mathbb{R}^{d_{\text{TE}}}$ , and define  $\mathbf{p}_i^{\widetilde{\text{TE}}} = (p_{i,d-1}, p_{i,d})^\top \in \mathbb{R}^2$ , where for any  $j \in [d/2]$ , we have

$$p_{i,2j-1} = \cos(i \cdot \omega^j), \quad p_{i,2j} = \sin(i \cdot \omega^j),$$

and  $\omega = M^{-2/d_{\text{PE}}}$  where  $M$  is defined in Definition 3.

Now we construct query and key matrices to be

$$\begin{aligned} \mathbf{W}_{\text{Q}} &= \begin{bmatrix} \mathbf{0}_{d_{\text{TE}} \times d_{\text{TE}}} & \mathbf{p}_0^{\text{TE}} \otimes \mathbf{1}_2 \\ \xi \mathbf{p}_{-T}^{\widetilde{\text{TE}}} \otimes \tilde{\mathbf{u}}_{\langle \mathbf{x} \rangle} & \mathbf{0}_{2 \times 2} \end{bmatrix} \in \mathbb{R}^{d \times d}, \\ \mathbf{W}_{\text{K}} &= \begin{bmatrix} \mathbf{0}_{d_{\text{TE}} \times d_{\text{TE}}} & \eta \mathbf{R}_{\text{TE}}^{(\ell)} (\mathbf{p}_0^{\text{TE}} \otimes \mathbf{1}_2) \\ \mathbf{0}_{2 \times d_{\text{TE}}} & \eta \mathbf{p}_0^{\widetilde{\text{TE}}} \otimes \mathbf{1}_2 \end{bmatrix} \in \mathbb{R}^{d \times d}, \end{aligned}$$

where  $\xi, \eta > 0$  will be specified later and  $\mathbf{R}_{\text{TE}}^{(\ell)} \in \mathbb{R}^{d_{\text{TE}} \times d_{\text{TE}}}$ ,  $\mathbf{R}^{(-T, d/2)} \in \mathbb{R}^{2 \times 2}$  are defined as in Definition 3 and  $\mathbf{1}_m$  denotes the all-one vector of dimension  $m$ . By Lemma 33, one can calculate that

$$\mathbf{R}_{\text{TE}}^{(j)} \mathbf{p}_i^{\text{TE}} = \mathbf{p}_{i+j}^{\text{TE}}, \quad \mathbf{R}^{(j, d/2)} \mathbf{p}_i^{\widetilde{\text{TE}}} = \mathbf{p}_{i+j}^{\widetilde{\text{TE}}}, \quad \text{for any integers } i, j.$$

Therefore,

$$\begin{aligned} \mathbf{q}_i &= \mathbf{R}^{(i)} \mathbf{W}_{\text{Q}} \hat{\mathbf{v}}_i = \mathbf{R}^{(i)} \begin{bmatrix} \mathbf{p}_0^{\text{TE}} \\ \xi \langle \tilde{\mathbf{u}}_{\langle \mathbf{x} \rangle}, \tilde{\mathbf{v}}_i \rangle \mathbf{p}_{-T}^{\widetilde{\text{TE}}} \end{bmatrix}, \\ \mathbf{k}_i &= \mathbf{R}^{(i)} \mathbf{W}_{\text{K}} \hat{\mathbf{v}}_i = \mathbf{R}^{(i)} \begin{bmatrix} \eta \mathbf{R}_{\text{TE}}^{(\ell)} \mathbf{p}_0^{\text{TE}} \\ \eta \mathbf{p}_0^{\widetilde{\text{TE}}} \end{bmatrix} = \mathbf{R}^{(i)} \begin{bmatrix} \eta \mathbf{p}_\ell^{\text{TE}} \\ \eta \mathbf{p}_0^{\widetilde{\text{TE}}} \end{bmatrix}. \end{aligned}$$

Then for any  $1 \leq j \leq i \leq T$ , we have

$$\begin{aligned}\langle \mathbf{q}_i, \mathbf{k}_j \rangle &= \eta \left( \langle \mathbf{R}_{\text{TE}}^{(i)} \mathbf{p}_0^{\text{TE}}, \mathbf{R}_{\text{TE}}^{(j)} \mathbf{p}_\ell^{\text{TE}} \rangle + \xi \langle \tilde{\mathbf{u}}_{\langle \bar{x} \rangle}, \tilde{\mathbf{v}}_i \rangle \langle \mathbf{R}^{(i)} \mathbf{p}_{-T}^{\text{TE}}, \mathbf{R}^{(j)} \mathbf{p}_0^{\text{TE}} \rangle \right) \\ &= \eta \left( \langle \mathbf{p}_i^{\text{TE}}, \mathbf{p}_{j+\ell}^{\text{TE}} \rangle + \xi \langle \tilde{\mathbf{u}}_{\langle \bar{x} \rangle}, \tilde{\mathbf{v}}_i \rangle \cdot \langle \mathbf{p}_{i-j-T}^{\text{TE}}, \mathbf{p}_0^{\text{TE}} \rangle \right).\end{aligned}$$

Now we fix  $i \in [T]$ . We first consider the case where  $\langle \tilde{\mathbf{v}}_i, \tilde{\mathbf{u}}_{\langle \bar{x} \rangle} \rangle = 1$  (which also implies  $i > \ell$ ). By (C.5) and the assumption that token embeddings are orthonormal, we have

$$\langle \tilde{\mathbf{v}}_i, \tilde{\mathbf{u}}_v \rangle = 0, \quad \forall v \in \text{Voc} \setminus \{\langle \bar{x} \rangle\} \implies \langle \tilde{\mathbf{v}}_i, \tilde{\mathbf{u}}_{\langle \bar{x} \rangle} \rangle = 0.$$

Therefore, we have  $\langle \mathbf{q}_i, \mathbf{k}_j \rangle = \eta \langle \mathbf{p}_i^{\text{TE}}, \mathbf{p}_{j+\ell}^{\text{TE}} \rangle$ . By Lemma 34, we have

$$\langle \mathbf{q}_i, \mathbf{k}_{i-\ell} \rangle = \eta \langle \mathbf{p}_i^{\text{TE}}, \mathbf{p}_{(i-\ell)+\ell}^{\text{TE}} \rangle = \eta \langle \mathbf{p}_i^{\text{TE}}, \mathbf{p}_i^{\text{TE}} \rangle = \eta d_{\text{TE}}/2$$

and

$$\langle \mathbf{q}_i, \mathbf{k}_j \rangle = \eta \langle \mathbf{p}_i^{\text{TE}}, \mathbf{p}_{j+\ell}^{\text{TE}} \rangle \leq \eta d_{\text{TE}}/2 - \eta \varepsilon_T, \quad \forall j \neq i - \ell.$$

where  $\varepsilon_T > 0$ . This implies  $\langle \mathbf{q}_i, \mathbf{k}_j \rangle$  is maximized when  $j = i - \ell$  with a non-zero gap  $\eta \varepsilon_T$ , and therefore, we have

$$s_{i,i-\ell} = \frac{\exp(\langle \mathbf{q}_i, \mathbf{k}_{i-\ell} \rangle)}{\sum_{j \in [i]} \exp(\langle \mathbf{q}_i, \mathbf{k}_j \rangle)} \geq \frac{\exp(\eta \varepsilon_T)}{\exp(\eta \varepsilon_T) + (i-1)}. \quad (\text{C.6})$$

Now we consider the case where  $\langle \tilde{\mathbf{v}}_i, \tilde{\mathbf{u}}_{\langle \bar{x} \rangle} \rangle = 0$ . Again, by (C.5) and the orthonormal assumption of token embeddings, we have

$$\langle \tilde{\mathbf{v}}_i, \tilde{\mathbf{u}}_{\langle \bar{x} \rangle} \rangle = \sum_{v \in \text{Voc} \setminus \{\langle \bar{x} \rangle\}} \langle \tilde{\mathbf{v}}_i, \tilde{\mathbf{u}}_v \rangle = \sum_{v \in \text{Voc} \setminus \{\langle \bar{x} \rangle\}} \lambda_{i,v} \geq \sum_{v \in \text{Voc} \setminus \{\langle \bar{x} \rangle\}} \lambda_{i,v}^2 = 1.$$

Note that for any  $j \in \{2, \dots, i\}$ ,

$$\begin{aligned}\langle \mathbf{p}_{i-j-T}^{\text{TE}}, \mathbf{p}_0^{\text{TE}} \rangle &= \cos((i-j-T)\omega^{d/2}) \\ &= \cos\left(\frac{T-(i-j)}{M}\right) \\ &< \cos\left(\frac{T-(i-1)}{M}\right) \\ &= \langle \mathbf{p}_{i-1-T}^{\text{TE}}, \mathbf{p}_0^{\text{TE}} \rangle,\end{aligned}$$

where the inequality holds due to  $M \geq T$  and the cosine function decreases strictly in  $[0, 1]$ .

Then, we can define  $\xi = \max_{1 \leq j < i \leq T} \frac{2d_{\text{PE}}}{\langle \mathbf{p}_{i-1-T}^{\text{TE}}, \mathbf{p}_0^{\text{TE}} \rangle - \langle \mathbf{p}_{i-j-T}^{\text{TE}}, \mathbf{p}_0^{\text{TE}} \rangle}$ , and thus

$$\xi \left( \langle \mathbf{p}_{i-1-T}^{\text{TE}}, \mathbf{p}_0^{\text{TE}} \rangle - \langle \mathbf{p}_{i-j-T}^{\text{TE}}, \mathbf{p}_0^{\text{TE}} \rangle \right) \geq 2d_{\text{PE}}, \quad \forall 1 \leq j < i \leq T.$$

Note that for any  $j \in \{2, \dots, T\}$ , we have

$$\begin{aligned}
& \langle \mathbf{q}_i, \mathbf{k}_1 \rangle - \langle \mathbf{q}_i, \mathbf{k}_j \rangle \\
&= \eta \left( \xi \langle \tilde{\mathbf{u}}_{\langle \tilde{\mathbf{x}} \rangle}, \tilde{\mathbf{v}}_i \rangle \left( \langle \mathbf{p}_{i-1-T}^{\widetilde{\text{TE}}}, \mathbf{p}_0^{\widetilde{\text{TE}}} \rangle - \langle \mathbf{p}_{i-j-T}^{\widetilde{\text{TE}}}, \mathbf{p}_0^{\widetilde{\text{TE}}} \rangle \right) - (\langle \bar{\mathbf{p}}_i, \bar{\mathbf{p}}_{j+\ell} \rangle - \langle \bar{\mathbf{p}}_i, \bar{\mathbf{p}}_{1+\ell} \rangle) \right) \\
&\geq \eta (2d_{\text{PE}} - d_{\text{PE}}) \\
&= \eta d_{\text{PE}}.
\end{aligned}$$

Therefore,

$$s_{i,1} = \frac{\exp(\langle \mathbf{q}_i, \mathbf{k}_1 \rangle)}{\sum_{j \in [i]} \exp(\langle \mathbf{q}_i, \mathbf{k}_j \rangle)} \geq \frac{\exp(\eta d_{\text{PE}})}{\exp(\eta d_{\text{PE}}) + (i-1)}. \quad (\text{C.7})$$

By choosing a sufficiently large  $\eta$ , the lower bound of (C.6) and (C.7) will both exceed  $1 - \varepsilon$  and thus the proof is complete.  $\square$

## C.3 Experiment Details

### C.3.1 Dataset

Table C.3: ProsQA statistics. Numbers are averaged over problem instances.

|       | #Problems | $ V $ | $ E $ | Sol. Len. |
|-------|-----------|-------|-------|-----------|
| Train | 14785     | 22.8  | 36.5  | 3.5       |
| Val   | 257       | 22.7  | 36.3  | 3.5       |
| Test  | 419       | 22.7  | 36.0  | 3.5       |

The statistics of the ProsQA dataset is shown in Table C.3.

### C.3.2 Stability of the Representation

Table C.4: The inner products between  $i$ -th continuous thoughts and nodes (3 runs with different random seeds).

|               | Step 1            | Step 2            | Step 3           | Step 4            |
|---------------|-------------------|-------------------|------------------|-------------------|
| Not Reachable | -0.37,-0.25,-0.33 | -0.26,-0.04,-0.14 | -0.09,-0.01,0.02 | -0.25,-0.23,-0.27 |
| Reachable     | 3.59,3.62,3.71    | 1.55,1.42,1.37    | 0.80,0.77,0.62   | 0.61,0.66,0.53    |
| –Frontier     | 5.09,5.13,5.38    | 2.69,2.45,2.63    | 2.11,1.95,2.01   | 2.27,2.12,2.29    |
| –Optimal      | 6.41,6.52,6.84    | 4.78,4.67,5.11    | 6.00,5.44,6.43   | 9.48,8.98,9.58    |

To test whether COCONUT can always learn the desired superpositional search behavior, we conduct multiple experiments with 3 random seeds. We report the mean inner products between continuous thoughts and each node group in Table C.4, following the setting in Figure 4.6. The results are consistent across multiple runs.

### C.3.3 Computing Resources

Each run of COCONUT takes about 24 hours on two Nvidia A100 80GB GPUs.

# Appendix D

## Appendix for Training Dynamics of Continuous CoT

### D.1 Notation Details

The notation and meaning of each token and the position index are the same as [169]. For completeness, we provide detailed descriptions of different tokens in Table D.1 (which is Table 2 in [169]), and the position index of different tokens or continuous thoughts in Table D.2 (which is Table 3 in [169]).

| Tokens              | Meanings                                                                  |
|---------------------|---------------------------------------------------------------------------|
| $\langle s \rangle$ | a special token denoting the beginning of the sentence                    |
| $s_i$               | the source node of edge $i$                                               |
| $t_i$               | the target node of edge $i$                                               |
| $\langle e \rangle$ | a special token marking the end of an edge                                |
| $\langle Q \rangle$ | a special token followed by two candidate nodes                           |
| $c_1, c_2$          | two candidate destination nodes                                           |
| $\langle R \rangle$ | a special token marking the start of reasoning                            |
| $r$                 | the root node                                                             |
| $[t_i]$             | the $i$ -th continuous thought (represented by a $d$ -dimensional vector) |
| $\langle A \rangle$ | a special token driving the model to make the final prediction            |

Table D.1: Meaning of each token (Table 2 in [169]).

| Notations                                   | Position indices  |
|---------------------------------------------|-------------------|
| $\text{idx}(\langle \mathbf{s} \rangle)$    | 1                 |
| $\text{idx}(\mathbf{s}_i)$                  | $3i - 1$          |
| $\text{idx}(\mathbf{t}_i)$                  | $3i$              |
| $\text{idx}(\langle \mathbf{e} \rangle, i)$ | $3i + 1$          |
| $\text{idx}(\langle \mathbf{Q} \rangle)$    | $3m + 2$          |
| $\text{idx}(\mathbf{c}_1)$                  | $3m + 3$          |
| $\text{idx}(\mathbf{c}_2)$                  | $3m + 4$          |
| $\text{idx}(\langle \mathbf{R} \rangle)$    | $3m + 5$          |
| $\text{idx}(\mathbf{r})$                    | $3m + 6 = t_0$    |
| $\text{idx}([\mathbf{t}_i])$                | $t_0 + i$         |
| $\text{idx}(\langle \mathbf{A} \rangle)$    | $t_0 + C + 1 = T$ |

Table D.2: Position indices of different tokens or continuous thoughts in the input sequence (Table 3 in [169]).

## D.2 Missing Proofs for Section 5.3

In this section, we provide the full proof of theoretical results in Section 5.3. We first provide theoretical analysis of COCONUT-BFS and COCONUT in Section D.2.1, Section D.2.2, respectively, and provide results for continuous thought expansion in Section D.2.3.

### D.2.1 Analysis of Coconut-BFS

In this section, we analyze the training dynamics of COCONUT-BFS. We first provide the closed-form formulation of the gradient  $\nabla_{\mu_v} \ell_{\mathcal{G}, \mathbf{x}}^{\text{BFS}} = \nabla_{\mu_v} \ell_{\mathcal{G}, \mathbf{x}}^{\text{BFS}}(\boldsymbol{\mu})$ , where  $\boldsymbol{\mu} = \{\mu_v\}_{v \in \mathcal{V}}$  is the set of parameters. We omit the superscript or subscript when the context is clear.

**Lemma 36** (Per-instance gradient of  $\mu_v$  for COCONUT-BFS). *Under the loss function of COCONUT-BFS as given in (5.5) and the forward pass as in (5.2), the per-instance gradient is*

$$\nabla_{\mu_v} \ell(\boldsymbol{\mu}) = -\frac{\mathbb{1}(\{v \in \mathcal{N}_c\})}{\sqrt{|\mathcal{N}_c|}} \cdot \frac{\sum_{v': (v \rightarrow v') \in \mathcal{E}} \exp(\xi_{v'})}{\exp(\xi_+)} \cdot \frac{n - |\mathcal{N}_{c+1}|}{\exp(\xi_+) + n - |\mathcal{N}_{c+1}|}$$

for any  $v \in \mathcal{V}$ , where  $\xi_+ = \log\left(\sum_{v \in \mathcal{N}_{c+1}} \exp(\xi_v)\right)$ .

*Proof.* First, note that for any  $v \in \mathcal{V}$ , according to (5.2), the logit can be calculated as

$$\begin{aligned}
 \xi_v &= \mathbb{E}(v)^\top (\hat{\mathbf{v}}_{[\mathbf{t}_c]} + \phi(\hat{\mathbf{v}}_{[\mathbf{t}_c]}; \{\hat{\mathbf{v}}_i\}_{i=1}^m)) \\
 &= \mathbb{E}(v)^\top \left( \hat{\mathbf{v}}_{[\mathbf{t}_c]} + \mathbf{W}_V \sum_{i=1}^m \sigma(\hat{\mathbf{v}}_{[\mathbf{t}_c]}^\top \mathbf{W} \hat{\mathbf{v}}_i) \hat{\mathbf{v}}_i \right) \\
 &= \mathbb{E}(v)^\top \left( \hat{\mathbf{v}}_{[\mathbf{t}_c]} + \sum_{i=1}^m (\hat{\mathbf{v}}_{[\mathbf{t}_c]}^\top \mathbf{W} \hat{\mathbf{v}}_i) \hat{\mathbf{v}}_i \right) \\
 &= \mathbb{E}(v)^\top \left( \hat{\mathbf{v}}_{[\mathbf{t}_c]} + \sum_{i=1}^m \left( \left( \sum_{v' \in \mathcal{N}_c} \lambda \mathbb{E}(v') \right)^\top \sum_{v' \in \mathcal{V}} \mu_{v'} \mathbb{E}(v') \mathbb{E}_s(v')^\top (\mathbb{E}_s(\mathbf{s}_i) + \mathbb{E}_t(\mathbf{t}_i)) \right) \hat{\mathbf{v}}_i \right) \\
 &= \mathbb{E}(v)^\top \left( \hat{\mathbf{v}}_{[\mathbf{t}_c]} + \sum_{i=1}^m \left( \left( \sum_{v' \in \mathcal{N}_c} \lambda \mathbb{E}(v') \right)^\top \mu_{\mathbf{s}_i} \mathbb{E}(\mathbf{s}_i) \right) \hat{\mathbf{v}}_i \right) \\
 &= \mathbb{E}(v)^\top \left( \hat{\mathbf{v}}_{[\mathbf{t}_c]} + \sum_{i=1}^m \lambda \mu_{\mathbf{s}_i} \mathbb{1}(\{\} \mathbf{s}_i \in \mathcal{N}_c) \hat{\mathbf{v}}_i \right) \\
 &= \lambda \cdot \mathbb{1}(\{v \in \mathcal{N}_c\}) + \sum_{i=1}^m \lambda \mu_{\mathbf{s}_i} \mathbb{1}(\{\} \mathbf{s}_i \in \mathcal{N}_c) \cdot \mathbb{1}(\{v = \mathbf{t}_i\}),
 \end{aligned}$$

where  $\lambda = \frac{1}{\sqrt{|\mathcal{N}_c|}}$ . Note that by the definition of  $\mathcal{N}_c$ , we have  $\xi_v = 0$  if  $v \notin \mathcal{N}_{c+1}$ . Therefore,

$$\begin{aligned}
 \ell(\boldsymbol{\mu}) &= -\log \frac{\sum_{v \in \mathcal{N}_{c+1}} \exp(\xi_v)}{\sum_{v \in \mathcal{V}} \exp(\xi_v)} \\
 &= -\log \frac{\sum_{v \in \mathcal{N}_{c+1}} \exp(\lambda \cdot \mathbb{1}(\{v \in \mathcal{N}_c\}) + \sum_{i=1}^m \lambda \mu_{\mathbf{s}_i} \mathbb{1}(\{\} \mathbf{s}_i \in \mathcal{N}_c) \cdot \mathbb{1}(\{v = \mathbf{t}_i\}))}{\sum_{v \in \mathcal{V}} \exp(\lambda \cdot \mathbb{1}(\{v \in \mathcal{N}_c\}) + \sum_{i=1}^m \lambda \mu_{\mathbf{s}_i} \mathbb{1}(\{\} \mathbf{s}_i \in \mathcal{N}_c) \cdot \mathbb{1}(\{v = \mathbf{t}_i\}))} \\
 &= -\log \left( 1 - \frac{n - |\mathcal{N}_{c+1}|}{\sum_{v \in \mathcal{V}} \exp(\lambda \cdot \mathbb{1}(\{v \in \mathcal{N}_c\}) + \sum_{i=1}^m \lambda \mu_{\mathbf{s}_i} \mathbb{1}(\{\} \mathbf{s}_i \in \mathcal{N}_c) \cdot \mathbb{1}(\{v = \mathbf{t}_i\}))} \right).
 \end{aligned}$$

For simplicity, we define  $\exp(\xi_+) = \sum_{v \in \mathcal{N}_{c+1}} \exp(\xi_v)$  and thus

$$\ell(\boldsymbol{\mu}) = -\log \left( 1 - \frac{n - |\mathcal{N}_{c+1}|}{\exp(\xi_+) + n - |\mathcal{N}_{c+1}|} \right).$$

Then the per-instance gradient can be calculated as

$$\begin{aligned}
 & \nabla_{\mu_v} \ell(\boldsymbol{\mu}) \\
 &= - \frac{\exp(\xi_+) + n - |\mathcal{N}_{c+1}|}{\exp(\xi_+)} \cdot \frac{n - |\mathcal{N}_{c+1}|}{(\exp(\xi_+) + n - |\mathcal{N}_{c+1}|)^2} \cdot \nabla_{\mu_v} \exp(\xi_+) \\
 &= - \frac{\sum_{v' \in \mathcal{N}_{c+1}} \exp(\xi_{v'}) \mathbb{1}(\{v \in \mathcal{N}_c\} \sum_{i=1}^m \lambda \mathbb{1}(\{\mathbf{s}_i = v, \mathbf{t}_i = v'\})}{\exp(\xi_+)} \cdot \frac{n - |\mathcal{N}_{c+1}|}{\exp(\xi_+) + n - |\mathcal{N}_{c+1}|} \\
 &= - \lambda \cdot \mathbb{1}(\{v \in \mathcal{N}_c\}) \frac{\sum_{v': (v \rightarrow v') \in \mathcal{E}} \exp(\xi_{v'})}{\exp(\xi_+)} \cdot \frac{n - |\mathcal{N}_{c+1}|}{\exp(\xi_+) + n - |\mathcal{N}_{c+1}|}.
 \end{aligned}$$

□

Now we calculate the gradient of  $\mu_v$  over the whole dataset, where the nodes of the graphs are randomly shuffled. We also write  $\mathcal{L}^{\text{BFS}} = \mathcal{L}^{\text{BFS}}(\boldsymbol{\mu})$  and omit the superscript when the context is clear.

**Lemma 37** (Whole-dataset gradient of  $\mu_v$  for COCONUT-BFS). *Under the loss function of COCONUT-BFS as given in (5.5) and the forward pass as in (5.2) and assuming all  $\mu_v$  have the same value, the gradient w.r.t. the whole dataset is*

$$\nabla_{\mu_v} \mathcal{L}(\boldsymbol{\mu}) = - \frac{\exp(-\xi_+)}{n \cdot \sqrt{|\mathcal{N}_c|}} \cdot \frac{n - |\mathcal{N}_{c+1}|}{\exp(\xi_+) + n - |\mathcal{N}_{c+1}|} \sum_{v \in \mathcal{V}} d_v \exp(\xi_v)$$

for any  $v \in \mathcal{V}$  which is independent of  $v$ , where  $\xi_+ = \log \left( \sum_{v \in \mathcal{N}_{c+1}} \exp(\xi_v) \right)$ .

*Proof.* Denote  $\xi_+^{(\mathcal{G}, \mathbf{r})} = \log \left( \sum_{v \in \mathcal{N}_{c+1}^{(\mathcal{G}, \mathbf{r})}} \exp(\xi_v^{(\mathcal{G}, \mathbf{r})}) \right)$ , where  $\xi_v^{(\mathcal{G}, \mathbf{r})}$  is the logit of  $v$  when the graph in the prompt is  $\mathcal{G}$  and the start node is  $\mathbf{r}$ .

According to Lemma 36 and the condition that all  $\mu_v$  have the same value, for any permutation  $\pi \in S_{\mathcal{V}}$ , we have

$$\begin{aligned}
 & \xi_{\pi(v)}^{(\pi(\mathcal{G}), \pi(\mathbf{r}))} \\
 &= \frac{\mathbb{1}(\{\pi(v) \in \mathcal{N}_c^{\pi(\mathcal{G})}(\pi(\mathbf{r}))\}) + \sum_{i=1}^m \mu_{\pi(\mathbf{s}_i)} \mathbb{1}(\{\pi(\mathbf{s}_i) \in \mathcal{N}_c^{\pi(\mathcal{G})}(\pi(\mathbf{r}))\}) \cdot \mathbb{1}(\{\pi(v) = \pi(\mathbf{t}_i)\})}{\sqrt{|\mathcal{N}_c^{\pi(\mathcal{G})}(\pi(\mathbf{r}))|}} \\
 &= \frac{1}{\sqrt{|\mathcal{N}_c^{\mathcal{G}}(\mathbf{r})|}} \left( \mathbb{1}(\{v \in \mathcal{N}_c^{\mathcal{G}}(\mathbf{r})\}) + \sum_{i=1}^m \mu_{\mathbf{s}_i} \mathbb{1}(\{\mathbf{s}_i \in \mathcal{N}_c^{\mathcal{G}}(\mathbf{r})\}) \cdot \mathbb{1}(\{v = \mathbf{t}_i\}) \right) \\
 &= \xi_v^{(\mathcal{G}, \mathbf{r})}.
 \end{aligned}$$

This also implies

$$\begin{aligned}
 \exp\left(\xi_+^{(\pi(\mathcal{G}), \pi(\mathbf{r}))}\right) &= \sum_{\pi(v) \in \mathcal{N}_{c+1}^{\pi(\mathcal{G})}(\pi(\mathbf{r}))} \exp\left(\xi_{\pi(v)}^{(\pi(\mathcal{G}), \pi(\mathbf{r}))}\right) \\
 &= \sum_{v \in \mathcal{N}_{c+1}^{\mathcal{G}}(\mathbf{r})} \exp\left(\xi_v^{(\mathcal{G}, \mathbf{r})}\right) \\
 &= \exp\left(\xi_+^{(\mathcal{G}, \mathbf{r})}\right).
 \end{aligned}$$

Therefore, by Lemma 36, we can obtain that

$$\begin{aligned}
 &\nabla_{\mu_{\pi(v)}} \ell_{\pi(\mathcal{G}), \pi(\mathbf{r})}(\boldsymbol{\mu}) \\
 &= - \frac{\mathbb{1}(\{\pi(v) \in \mathcal{N}_c^{\pi(\mathcal{G})}(\pi(\mathbf{r}))\})}{\sqrt{|\mathcal{N}_c^{\pi(\mathcal{G})}(\pi(\mathbf{r}))|}} \cdot \frac{\sum_{\pi(v') : (\pi(v) \rightarrow \pi(v')) \in \pi(\mathcal{E})} \exp\left(\xi_{\pi(v')}^{(\pi(\mathcal{G}), \pi(\mathbf{r}))}\right)}{\exp\left(\xi_+^{(\pi(\mathcal{G}), \pi(\mathbf{r}))}\right)} \\
 &\quad \cdot \frac{n - |\mathcal{N}_{c+1}^{\pi(\mathcal{G})}(\pi(\mathbf{r}))|}{\exp\left(\xi_+^{(\pi(\mathcal{G}), \pi(\mathbf{r}))}\right) + n - |\mathcal{N}_{c+1}^{\pi(\mathcal{G})}(\pi(\mathbf{r}))|} \\
 &= - \frac{\mathbb{1}(\{v \in \mathcal{N}_c^{\mathcal{G}}(\mathbf{r})\})}{\sqrt{|\mathcal{N}_c^{\mathcal{G}}(\mathbf{r})|}} \cdot \frac{\sum_{v' : (v \rightarrow v') \in \mathcal{E}} \exp\left(\xi_{v'}^{(\mathcal{G}, \mathbf{r})}\right)}{\exp\left(\xi_+^{(\mathcal{G}, \mathbf{r})}\right)} \cdot \frac{n - |\mathcal{N}_{c+1}^{\mathcal{G}}(\mathbf{r})|}{\exp\left(\xi_+^{(\mathcal{G}, \mathbf{r})}\right) + n - |\mathcal{N}_{c+1}^{\mathcal{G}}(\mathbf{r})|} \\
 &= \nabla_{\mu_v} \ell_{\mathcal{G}, \mathbf{r}}(\boldsymbol{\mu}).
 \end{aligned}$$

Therefore, we can calculate the gradient with respect to the whole dataset as

$$\begin{aligned}
 \nabla_{\mu_v} \mathcal{L}(\boldsymbol{\mu}) &= \mathbb{E}_{\pi \sim \text{Unif}(S_{\mathcal{V}})} [\nabla_{\mu_v} \ell_{\pi(\mathcal{G}), \pi(\mathbf{r})}(\boldsymbol{\mu})] \\
 &= \mathbb{E}_{\pi \sim \text{Unif}(S_{\mathcal{V}})} [\nabla_{\mu_{\pi^{-1}(v)}} \ell_{\mathcal{G}, \mathbf{r}}(\boldsymbol{\mu})] \\
 &= \mathbb{E}_{v' \sim \text{Unif}(\mathcal{V})} [\nabla_{\mu_{v'}} \ell_{\mathcal{G}, \mathbf{r}}(\boldsymbol{\mu})]
 \end{aligned}$$

which is independent of  $v$  and thus the gradients for  $\mu_v$  are equal for all  $v \in \mathcal{V}$ . Furthermore,

we can calculate that

$$\begin{aligned}
 & \nabla_{\mu_v} \mathcal{L}(\boldsymbol{\mu}) \\
 &= \frac{1}{n} \sum_{v \in \mathcal{V}} \left( -\frac{\mathbb{1}(\{v \in \mathcal{N}_c\})}{\sqrt{|\mathcal{N}_c|}} \cdot \frac{\sum_{v': (v \rightarrow v') \in \mathcal{E}} \exp(\xi_{v'})}{\exp(\xi_+)} \cdot \frac{n - |\mathcal{N}_{c+1}|}{\exp(\xi_+) + n - |\mathcal{N}_{c+1}|} \right) \\
 &= -\frac{1}{n \cdot \sqrt{|\mathcal{N}_c|}} \cdot \frac{n - |\mathcal{N}_{c+1}|}{\exp(\xi_+) + n - |\mathcal{N}_{c+1}|} \sum_{v \in \mathcal{N}_c} \frac{\sum_{v': (v \rightarrow v') \in \mathcal{E}} \exp(\xi_{v'})}{\exp(\xi_+)} \\
 &= -\frac{\exp(-\xi_+)}{n \cdot \sqrt{|\mathcal{N}_c|}} \cdot \frac{n - |\mathcal{N}_{c+1}|}{\exp(\xi_+) + n - |\mathcal{N}_{c+1}|} \sum_{v \in \mathcal{N}_c} \sum_{v': (v \rightarrow v') \in \mathcal{E}} \exp(\xi_{v'}) \\
 &= -\frac{\exp(-\xi_+)}{n \cdot \sqrt{|\mathcal{N}_c|}} \cdot \frac{n - |\mathcal{N}_{c+1}|}{\exp(\xi_+) + n - |\mathcal{N}_{c+1}|} \sum_{v \in \mathcal{V}} d_v \exp(\xi_v).
 \end{aligned}$$

□

According to the gradient of  $\mu_v$ , we finally show that  $\mu_v$  diverges to infinity at least logarithmically in  $t$ .

**Lemma 38** (Dynamics of  $\mu_v$  for COCONUT-BFS). *Let  $\mu_v(t)$  be the value of  $\mu_v$  at time  $t$ . Assume zero-initialization, i.e.,  $\mu_v(0) = 0$  for all  $v \in \mathcal{V}$ . Under gradient flow*

$$\dot{\mu}_v = -\alpha \cdot \nabla_{\mu_v} \mathcal{L}^{\text{BFS}}(\boldsymbol{\mu}) \quad (\text{D.1})$$

where  $\alpha > 0$  is the learning rate, we have

$$\mu_v(t) \geq c_1 \ln(1 + \alpha c_2 t)$$

for all  $v \in \mathcal{V}$  where  $c_1 = \frac{1}{2\sqrt{|\mathcal{N}_c|}}$ ,  $c_2 = n^{-3}e^{-2}$ .

*Proof.* First, by Lemma 37, all  $\dot{\mu}_v$  have the same value if all  $\mu_v$  have the same value. Given that  $\mu_v(0) = 0$  for all  $v \in \mathcal{V}$ , we can obtain that for any time  $t$ ,  $\mu_v(t)$  have the same value for all  $v \in \mathcal{V}$  using similar argument as in Lemma 15 of [huang2025generalization].

Now, given any fixed time  $t \geq 0$ , it holds that  $\mu_v(t)$  has the same value for all  $v \in \mathcal{V}$ . We omit  $t$  for notation convenience, i.e., using  $\mu_v$  to represent  $\mu_v(t)$ . Below, we first provide a lower bound of the gradient  $\dot{\mu}_v$ .

Since we are guaranteed that one of  $\mathbf{c}_1$  and  $\mathbf{c}_2$  cannot be reached from  $\mathbf{r}$ ,  $\mathcal{N}_{c+1}$  cannot contain all the vertices in  $\mathcal{V}$  for any  $c$ , and thus  $n - |\mathcal{N}_{c+1}| \geq 1$ . Also, since one of  $\mathbf{c}_1$  and  $\mathbf{c}_2$  is guaranteed to be reachable from  $\mathbf{r}$ , there exists  $v \in \mathcal{V}$  such that  $d_v \geq 1$  for any  $c$ . This is because the start node  $\mathbf{r} \in \mathcal{N}_0 \subseteq \mathcal{N}_c$  for any  $c \geq 0$ , and we can take  $v = p_1$  which is on the

shortest path from  $\mathbf{r}$  to  $\mathbf{c}^*$ . Therefore, we can obtain that  $\dot{\mu}_v > 0$ . Moreover, we have

$$\begin{aligned}\dot{\mu}_v &= \alpha \cdot \frac{\exp(-\xi_+)}{n \cdot \sqrt{|\mathcal{N}_c|}} \cdot \frac{n - |\mathcal{N}_{c+1}|}{\exp(\xi_+) + n - |\mathcal{N}_{c+1}|} \sum_{v \in \mathcal{V}} d_v \exp(\xi_v) \\ &\geq \alpha \cdot \frac{\exp(-\xi_+)}{n \cdot \sqrt{|\mathcal{N}_c|}} \cdot \frac{1}{\exp(\xi_+) + 1} \sum_{v \in \mathcal{V}} d_v \exp(\xi_v) \\ &\geq \frac{\alpha}{n \cdot \sqrt{|\mathcal{N}_c|}} \cdot \frac{\sum_{v \in \mathcal{V}} d_v \exp(\xi_v)}{(\exp(\xi_+) + 1) \cdot \exp(\xi_+)}.\end{aligned}$$

Note that by definition, for any vertex  $v \in \mathcal{N}_{c+1} \setminus \{\mathbf{r}\}$ , there must exists another vertex  $v' \in \mathcal{N}_c$  such that  $(v' \rightarrow v) \in \mathcal{E}$ , which implies that  $d_v \geq 1$ . Therefore,

$$\begin{aligned}\sum_{v \in \mathcal{V}} d_v \exp(\xi_v) &\geq \sum_{v \in \mathcal{N}_{c+1} \setminus \{\mathbf{r}\}} d_v \exp(\xi_v) \\ &\geq \sum_{v \in \mathcal{N}_{c+1} \setminus \{\mathbf{r}\}} \exp(\xi_v) \\ &= \exp(\xi_+) - \exp(\xi_{\mathbf{r}}),\end{aligned}$$

which further implies that

$$\begin{aligned}\dot{\mu}_v &\geq \frac{\alpha}{n \cdot \sqrt{|\mathcal{N}_c|}} \cdot \frac{\sum_{v \in \mathcal{V}} d_v \exp(\xi_v)}{(\exp(\xi_+) + 1) \cdot \exp(\xi_+)} \\ &\geq \frac{\alpha}{n \cdot \sqrt{|\mathcal{N}_c|}} \cdot \frac{\exp(\xi_+) - \exp(\xi_{\mathbf{r}})}{(\exp(\xi_+) + 1) \cdot \exp(\xi_+)}.\end{aligned}$$

Now recall from Lemma 36 that

$$\begin{aligned}\xi_v &= \frac{1}{\sqrt{|\mathcal{N}_c|}} \left( \mathbb{1}(\{v \in \mathcal{N}_c\}) + \sum_{i=1}^m \mu_{\mathbf{s}_i} \mathbb{1}(\{\mathbf{s}_i \in \mathcal{N}_c\}) \cdot \mathbb{1}(\{v = \mathbf{t}_i\}) \right) \\ &= \frac{1}{\sqrt{|\mathcal{N}_c|}} (\mathbb{1}(\{v \in \mathcal{N}_c\}) + d_v \cdot \mu_v) \\ &\leq \frac{1}{\sqrt{|\mathcal{N}_c|}} (1 + |\mathcal{N}_c| \cdot \mu_v).\end{aligned}$$

Therefore,

$$\begin{aligned}\exp(\xi_+) &= \sum_{v \in \mathcal{N}_{c+1}} \exp(\xi_v) \\ &\leq |\mathcal{N}_{c+1}| \exp\left(\frac{1}{\sqrt{|\mathcal{N}_c|}} (1 + |\mathcal{N}_c| \cdot \mu_v)\right) \\ &\leq n \cdot \exp\left(1 + \sqrt{|\mathcal{N}_c|} \cdot \mu_v\right).\end{aligned}$$

Also, since  $p_1 \in \mathcal{N}_{c+1}$  and  $\deg_{\mathcal{G}, \mathcal{N}_c}^-(p_1) \geq 1$ , we can obtain that

$$\begin{aligned} \exp(\xi_+) - \exp(\xi_r) &\geq \exp(\xi_{p_1}^{(\mathcal{G}, r)}) \\ &= \exp\left(\frac{1}{\sqrt{|\mathcal{N}_c|}} (\mathbb{1}(\{p_1 \in \mathcal{N}_c\}) + d_{p_1} \cdot \mu_v)\right) \\ &\geq \exp\left(\frac{\mu_v}{\sqrt{|\mathcal{N}_c|}}\right). \end{aligned}$$

Combining the above two inequalities, we can obtain that

$$\begin{aligned} \dot{\mu}_v &\geq \frac{\alpha}{n \cdot \sqrt{|\mathcal{N}_c|}} \cdot \frac{\exp(\xi_+) - \exp(\xi_r)}{(\exp(\xi_+) + 1) \cdot \exp(\xi_+)} \\ &\geq \frac{\alpha}{n \cdot \sqrt{|\mathcal{N}_c|}} \cdot \frac{\exp\left(\frac{\mu_v}{\sqrt{|\mathcal{N}_c|}}\right)}{2n^2 \cdot \exp\left(2 + 2\sqrt{|\mathcal{N}_c|} \cdot \mu_v\right)} \\ &\geq \frac{\alpha}{2n^3 e^2 \cdot \sqrt{|\mathcal{N}_c|}} \cdot \exp\left(-2\sqrt{|\mathcal{N}_c|} \cdot \mu_v\right). \end{aligned}$$

Finally, by applying Lemma 46, we can obtain that

$$\mu_v(t) \geq \frac{1}{2\sqrt{|\mathcal{N}_c|}} \ln(1 + \alpha n^{-3} e^{-2} t).$$

□

### D.2.2 Analysis of Coconut

In this section, we analyze the training dynamics of COCONUT. Similarly, we first provide the closed-form formulation of the gradient  $\nabla_{\mu_v} \ell_{\mathcal{G}, \mathbf{x}, p}^{\text{coco}} = \nabla_{\mu_v} \ell_{\mathcal{G}, \mathbf{x}, p}^{\text{coco}}(\boldsymbol{\mu})$ , where  $\boldsymbol{\mu} = \{\mu_v\}_{v \in \mathcal{V}}$  is the set of parameters. We omit the superscript or subscript when the context is clear.

**Lemma 39** (Per-instance gradient of  $\mu_v$  for COCONUT). *Under the loss function of COCONUT as given in (5.6) and the forward pass as in (5.2), the per-instance gradient is*

$$\nabla_{\mu_v} \ell(\boldsymbol{\mu}) = \frac{\mathbb{1}(\{v \in \mathcal{N}_c\})}{\sqrt{|\mathcal{N}_c|}} \left( -\mathbb{1}(\{v \rightarrow p_{c+1}\}) \in \mathcal{E} \right) + \frac{\sum_{v': (v \rightarrow v') \in \mathcal{E}} \exp(\xi_{v'})}{\exp(\xi_+) + n - |\mathcal{N}_{c+1}|}$$

for any  $v \in \mathcal{V}$ , where  $\xi_+ = \log\left(\sum_{v \in \mathcal{N}_{c+1}} \exp(\xi_v)\right)$ .

*Proof.* First, according to the proof of Lemma 36, we have

$$\xi_v = \lambda \cdot \mathbb{1}(\{v \in \mathcal{N}_c\}) + \sum_{i=1}^m \lambda \mu_{\mathbf{s}_i} \mathbb{1}(\{\mathbf{s}_i \in \mathcal{N}_c\}) \cdot \mathbb{1}(\{v = \mathbf{t}_i\}),$$

where  $\lambda = \frac{1}{\sqrt{|\mathcal{N}_c|}}$ . Note that by the definition of  $\mathcal{N}_c$ , we have  $\xi_v = 0$  if  $v \notin \mathcal{N}_{c+1}$ . Therefore,

$$\ell(\boldsymbol{\mu}) = -\log \frac{\exp(\xi_{p_{c+1}})}{\sum_{v \in \mathcal{V}} \exp(\xi_v)} = -\log \left( \frac{\exp(\xi_{p_{c+1}})}{\exp(\xi_+) + n - |\mathcal{N}_{c+1}|} \right),$$

where  $\exp(\xi_+) = \sum_{v \in \mathcal{N}_{c+1}} \exp(\xi_v)$ .

Then the per-instance gradient can be calculated as

$$\begin{aligned} & \nabla_{\mu_v} \ell(\boldsymbol{\mu}) \\ &= - \frac{\exp(\xi_+) + n - |\mathcal{N}_{c+1}|}{\exp(\xi_{p_{c+1}})} \\ & \quad \cdot \frac{\nabla_{\mu_v} \exp(\xi_{p_{c+1}}) \cdot (\exp(\xi_+) + n - |\mathcal{N}_{c+1}|) - \exp(\xi_{p_{c+1}}) \nabla_{\mu_v} \exp(\xi_+)}{(\exp(\xi_+) + n - |\mathcal{N}_{c+1}|)^2} \\ &= - \frac{\nabla_{\mu_v} \exp(\xi_{p_{c+1}})}{\exp(\xi_{p_{c+1}})} + \frac{\nabla_{\mu_v} \exp(\xi_+)}{\exp(\xi_+) + n - |\mathcal{N}_{c+1}|} \\ &= - \nabla_{\mu_v} \xi_{p_{c+1}} + \frac{\nabla_{\mu_v} \exp(\xi_+)}{\exp(\xi_+) + n - |\mathcal{N}_{c+1}|}. \end{aligned}$$

Since

$$\begin{aligned} \nabla_{\mu_v} \exp(\xi_+) &= \sum_{v' \in \mathcal{N}_{c+1}} \exp(\xi_{v'}) \mathbb{1}(\{v \in \mathcal{N}_c\}) \sum_{i=1}^m \lambda \mathbb{1}(\{\mathbf{s}_i = v, \mathbf{t}_i = v'\}) \\ &= \lambda \cdot \mathbb{1}(\{v \in \mathcal{N}_c\}) \sum_{v': (v \rightarrow v') \in \mathcal{E}} \exp(\xi_{v'}) \end{aligned}$$

and

$$\nabla_{\mu_v} \xi_{p_{c+1}} = \lambda \cdot \mathbb{1}(\{v \in \mathcal{N}_c\}) \cdot \mathbb{1}(\{v \rightarrow p_{c+1}\} \in \mathcal{E}),$$

we can finally obtain that

$$\nabla_{\mu_v} \ell(\boldsymbol{\mu}) = \frac{\mathbb{1}(\{v \in \mathcal{N}_c\})}{\sqrt{|\mathcal{N}_c|}} \left( -\mathbb{1}(\{v \rightarrow p_{c+1}\} \in \mathcal{E}) + \frac{\sum_{v': (v \rightarrow v') \in \mathcal{E}} \exp(\xi_{v'})}{\exp(\xi_+) + n - |\mathcal{N}_{c+1}|} \right).$$

□

Now we calculate the gradient of  $\mu_v$  over the whole dataset, where the nodes of the graphs are randomly shuffled. We also write  $\mathcal{L}^{\text{coco}} = \mathcal{L}^{\text{coco}}(\boldsymbol{\mu})$  and omit the superscript when the context is clear.

**Lemma 40** (Whole-dataset gradient of  $\mu_v$  for COCONUT). *Under the loss function of COCONUT as given in (5.6) and the forward pass as in (5.2) and assuming all  $\mu_v$  have the same value, the gradient w.r.t. the whole dataset is*

$$\nabla_{\mu_v} \mathcal{L}(\boldsymbol{\mu}) = \frac{1}{n \cdot \sqrt{|\mathcal{N}_c|}} \left( -d_{p_{c+1}} + \frac{\sum_{v \in \mathcal{N}_{c+1}} d_v \exp(\xi_v)}{\exp(\xi_+) + n - |\mathcal{N}_{c+1}|} \right)$$

for any  $v \in \mathcal{V}$  which is independent of  $v$ , where  $\xi_+ = \log \left( \sum_{v \in \mathcal{N}_{c+1}} \exp(\xi_v) \right)$ .

*Proof.* Similar to Lemma 37, we denote  $\xi_+^{(\mathcal{G}, \mathbf{r})} = \log \left( \sum_{v \in \mathcal{N}_{c+1}^{\mathcal{G}}(\mathbf{r})} \exp(\xi_v^{(\mathcal{G}, \mathbf{r})}) \right)$ , where  $\xi_v^{(\mathcal{G}, \mathbf{r})}$  is the logit of  $v$  when the graph in the prompt is  $\mathcal{G}$  and the start node is  $\mathbf{r}$ . According to the proof of Lemma 37, for any permutation  $\pi \in S_{\mathcal{V}}$ , we have

$$\xi_{\pi(v)}^{(\pi(\mathcal{G}), \pi(\mathbf{r}))} = \xi_v^{(\mathcal{G}, \mathbf{r})}$$

for any  $v \in \mathcal{V}$  and

$$\exp \left( \xi_+^{(\pi(\mathcal{G}), \pi(\mathbf{r}))} \right) = \exp \left( \xi_+^{(\mathcal{G}, \mathbf{r})} \right).$$

Therefore, by Lemma 39, we can obtain that

$$\begin{aligned} & \nabla_{\mu_{\pi(v)}} \ell_{\pi(\mathcal{G}), \pi(\mathbf{r}), \pi(p)}(\boldsymbol{\mu}) \\ &= \frac{\mathbb{1}(\{\pi(v) \in \mathcal{N}_c^{\pi(\mathcal{G})}(\pi(\mathbf{r}))\})}{\sqrt{|\mathcal{N}_c^{\pi(\mathcal{G})}(\pi(\mathbf{r}))|}} \left( -\mathbb{1}(\{\pi(v) \rightarrow \pi(p_{c+1})\}) \in \pi(\mathcal{E}) \right) \\ & \quad + \frac{\sum_{\pi(v') : (\pi(v) \rightarrow \pi(v')) \in \pi(\mathcal{E})} \exp \left( \xi_{\pi(v')}^{(\pi(\mathcal{G}), \pi(\mathbf{r}))} \right)}{\exp \left( \xi_+^{(\pi(\mathcal{G}), \pi(\mathbf{r}))} \right) + n - |\mathcal{N}_{c+1}^{\pi(\mathcal{G})}(\pi(\mathbf{r}))|} \right) \\ &= \frac{\mathbb{1}(\{v \in \mathcal{N}_c^{\mathcal{G}}(\mathbf{r})\})}{\sqrt{|\mathcal{N}_c^{\mathcal{G}}(\mathbf{r})|}} \left( -\mathbb{1}(\{v \rightarrow p_{c+1}\}) \in \mathcal{E} \right) + \frac{\sum_{v' : (v \rightarrow v') \in \mathcal{E}} \exp \left( \xi_{v'}^{(\mathcal{G}, \mathbf{r})} \right)}{\exp \left( \xi_+^{(\mathcal{G}, \mathbf{r})} \right) + n - |\mathcal{N}_{c+1}^{\mathcal{G}}(\mathbf{r})|} \right) \\ &= \nabla_{\mu_v} \ell_{\mathcal{G}, \mathbf{r}, p}(\boldsymbol{\mu}). \end{aligned}$$

Therefore, we can calculate the gradient with respect to the whole dataset as

$$\begin{aligned} \nabla_{\mu_v} \mathcal{L}(\boldsymbol{\mu}) &= \mathbb{E}_{\pi \sim \text{Unif}(S_{\mathcal{V}})} [\nabla_{\mu_v} \ell_{\pi(\mathcal{G}), \pi(\mathbf{r}), \pi(p)}(\boldsymbol{\mu})] \\ &= \mathbb{E}_{\pi \sim \text{Unif}(S_{\mathcal{V}})} [\nabla_{\mu_{\pi^{-1}(v)}} \ell_{\mathcal{G}, \mathbf{r}, p, c}(\boldsymbol{\mu})] \\ &= \mathbb{E}_{v' \sim \text{Unif}(\mathcal{V})} [\nabla_{\mu_{v'}} \ell_{\mathcal{G}, \mathbf{r}, p, c}(\boldsymbol{\mu})] \end{aligned}$$

which is independent of  $v$  and thus the gradients for  $\mu_v$  are equal for all  $v \in \mathcal{V}$ . Furthermore, similar to Lemma 37, we can calculate that

$$\begin{aligned} & \nabla_{\mu_v} \mathcal{L}(\boldsymbol{\mu}) \\ &= \frac{1}{n} \sum_{v \in \mathcal{V}} \frac{\mathbb{1}(\{v \in \mathcal{N}_c\})}{\sqrt{|\mathcal{N}_c|}} \left( -\mathbb{1}(\{v \rightarrow p_{c+1}\}) \in \mathcal{E} \right) + \frac{\sum_{v': (v \rightarrow v') \in \mathcal{E}} \exp(\xi_{v'})}{\exp(\xi_+^{(\mathcal{G}, \mathbf{r})}) + n - |\mathcal{N}_{c+1}|} \\ &= \frac{1}{n \cdot \sqrt{|\mathcal{N}_c|}} \left( -d_{p_{c+1}} + \frac{\sum_{v \in \mathcal{N}_{c+1}} d_v \exp(\xi_v)}{\exp(\xi_+^{(\mathcal{G}, \mathbf{r})}) + n - |\mathcal{N}_{c+1}|} \right). \end{aligned}$$

□

Finally, we derive the dynamics of  $\mu_v$ . Recall that we denote  $K = |\mathcal{N}_c|$ ,  $\lambda = \frac{1}{\sqrt{K}}$ . We also make the following notation for Theorem 19. Let  $d_\star := d_{p_{c+1}}$  and  $d_{\max} := \max_{u \in \mathcal{V}} d_u$ . Moreover, Let  $c_0 := n - |\mathcal{N}_{c+1}| \geq 1$  and denote

$$\begin{aligned} \xi_u(\boldsymbol{\mu}) &:= \lambda (\mathbb{1}(\{u \in \mathcal{N}_c\}) + \mu d_u), & E_+(\boldsymbol{\mu}) &:= \sum_{u \in \mathcal{N}_{c+1}} e^{\xi_u(\boldsymbol{\mu})}, \\ S(\boldsymbol{\mu}) &:= \sum_{u \in \mathcal{N}_{c+1}} d_u e^{\xi_u(\boldsymbol{\mu})}, & F(\boldsymbol{\mu}) &:= \frac{S(\boldsymbol{\mu})}{E_+(\boldsymbol{\mu}) + c_0}. \end{aligned}$$

**Theorem 19** (Dynamics of  $\mu_v$  for COCONUT). *Let  $\mu_v(t)$  be the value of  $\mu_v$  at time  $t$ . Assume zero-initialization, i.e.,  $\mu_v(0) = 0$  for all  $v \in \mathcal{V}$ . Under gradient flow*

$$\dot{\mu}_v = -\alpha \cdot \nabla_{\mu_v} \mathcal{L}^{\text{coco}}(\boldsymbol{\mu}) \quad (\text{D.2})$$

where  $\alpha > 0$  is the learning rate, suppose the initialization satisfies  $\mu_v(0) = 0$  for all  $v$ . Then:

1. **Scalar reduction.** For all  $t \geq 0$ ,  $\mu_v(t) \equiv \mu(t)$  is shared across  $v$ , and  $\mu(t)$  satisfies

$$\boxed{\dot{\mu}(t) = \frac{\alpha}{n\sqrt{K}} (d_\star - F(\mu(t)))}. \quad (\text{D.3})$$

2. **Regularity of  $F$ .** The function  $F : \mathbb{R} \rightarrow \mathbb{R}$  is  $C^\infty$ , strictly increasing, and satisfies

$$\lim_{\mu \rightarrow -\infty} F(\mu) = 0, \quad \lim_{\mu \rightarrow +\infty} F(\mu) = d_{\max}, \quad 0 < F(\mu) < d_{\max} \quad \text{for all finite } \mu.$$

3. **Finite fixed point when  $d_\star < d_{\max}$ .** If  $d_\star < d_{\max}$ , there exists a unique  $\mu^\star \in \mathbb{R}$  such that  $F(\mu^\star) = d_\star$ . The solution  $\mu(t)$  of (D.3) with  $\mu(0) = 0$  converges monotonically to  $\mu^\star$ :

$$\mu(t) \nearrow \mu^\star \quad \text{if } F(0) \leq d_\star, \quad \mu(t) \searrow \mu^\star \quad \text{if } F(0) > d_\star,$$

and the equilibrium  $\mu^\star$  is locally exponentially stable, i.e., there exists  $\gamma > 0$  such that for all large enough  $t$ , it holds that

$$|\mu(t) - \mu^\star| \leq e^{-\gamma t} |\mu(0) - \mu^\star|.$$

4. **Logarithmic divergence when  $d_\star = d_{\max}$ .** If  $d_\star = d_{\max}$ , then  $\dot{\mu}(t) > 0$  for all  $t$  and  $\mu(t) \rightarrow +\infty$ . Moreover, for all  $t \geq 0$ ,

$$\mu(t) \geq \frac{1}{\lambda d_{\max}} \ln \left( 1 + \frac{\alpha \lambda d_{\max}^2 c_0 e^{-\lambda}}{2 n^2 \sqrt{K}} t \right). \quad (\text{D.4})$$

*Proof.* **(1) Scalar reduction.** By Lemma 40 and the similar argument as in the proof of Lemma 38, we have  $\mu_v(t) \equiv \mu(t)$  for all  $t \geq 0$ . Therefore, the gradient  $\nabla_{\mu_v} \mathcal{L}(\boldsymbol{\mu})$  is independent of  $v$  and equals

$$\nabla_{\mu_v} \mathcal{L}(\boldsymbol{\mu}) = \frac{1}{n\sqrt{K}} \left( -d_\star + \frac{\sum_{u \in \mathcal{N}_{c+1}} d_u e^{\xi_u(\mu_u)}}{\sum_{u \in \mathcal{N}_{c+1}} e^{\xi_u(\mu_u)} + n - |\mathcal{N}_{c+1}|} \right) = \frac{1}{n\sqrt{K}} (-d_\star + F(\mu_v)).$$

Thus, we have

$$\dot{\mu}(t) = -\nabla_{\mu_v} \mathcal{L}(\boldsymbol{\mu}(t)) = \frac{\alpha}{n\sqrt{K}} (d_\star - F(\mu(t))).$$

**(2) Regularity and limits of  $F$ .** By the proof of Lemma 39 and the condition that  $\mu_v(t) \equiv \mu(t)$  for all  $v \in \mathcal{V}$  and  $t \geq 0$ , we have

$$E_+(\mu) = \sum_{u \in \mathcal{N}_{c+1}} \exp(\lambda(\mathbb{1}(\{ \} u \in \mathcal{N}_c) + \mu d_u)), S(\mu) = \sum_{u \in \mathcal{N}_{c+1}} d_u \exp(\lambda(\mathbb{1}(\{ \} u \in \mathcal{N}_c) + \mu d_u)).$$

Both functions are finite sums of  $C^\infty$  functions of  $\mu$ , hence  $F(\mu) = S(\mu)/(E_+(\mu) + c_0)$  is also  $C^\infty$ .

Now we show the strict monotonicity of  $F(\cdot)$  on  $\mu$  by differentiation. We further write  $\xi_u := \xi_u(\mu)$  for brevity. Then

$$E'_+(\mu) = \lambda \sum_{u \in \mathcal{N}_{c+1}} d_u e^{\xi_u}, \quad S'(\mu) = \lambda \sum_{u \in \mathcal{N}_{c+1}} d_u^2 e^{\xi_u}.$$

Therefore, we can obtain that

$$\begin{aligned} F'(\mu) &= \frac{S'(\mu)(E_+(\mu) + c_0) - S(\mu)E'_+(\mu)}{(E_+(\mu) + c_0)^2} \\ &= \frac{\lambda}{(E_+(\mu) + c_0)^2} \left[ \left( \sum_{u \in \mathcal{N}_{c+1}} d_u^2 e^{\xi_u} \right) (E_+(\mu) + c_0) - \left( \sum_{u \in \mathcal{N}_{c+1}} d_u e^{\xi_u} \right)^2 \right]. \end{aligned}$$

Note that by the Cauchy-Schwarz inequality, we have

$$\sum_{u \in \mathcal{N}_{c+1}} d_u^2 e^{\xi_u} \cdot E_+(\mu) - \left( \sum_{u \in \mathcal{N}_{c+1}} d_u e^{\xi_u} \right)^2 \geq 0,$$

and hence

$$F'(\mu) \geq \frac{\lambda}{(E_+(\mu) + c_0)^2} c_0 E_+(\mu) \left( \sum_{u \in \mathcal{N}_{c+1}} d_u^2 e^{\xi_u} \right) > 0$$

since  $c_0 > 0$  and there exists at least one node  $u \in \mathcal{N}_{c+1}$  (e.g.,  $p_{c+1}$ ) such that  $d_u \geq 1$  by definition. Thus,  $F(\cdot)$  is strictly increasing.

Now we consider the limits of  $F(\cdot)$ . First, note that

$$S(\mu) = \sum_{u \in \mathcal{N}_{c+1}} d_u e^{\xi_u(\mu)},$$

and for each  $u \in \mathcal{N}_{c+1}$ , either  $d_u = 0$  or  $d_u > 0$  and thus

$$\lim_{\mu \rightarrow -\infty} d_u e^{\xi_u(\mu)} = \lim_{\mu \rightarrow -\infty} d_u \exp(\lambda(\mathbb{1}(\{u \in \mathcal{N}_c\}) + \mu d_u)) = 0.$$

Therefore, we have  $\lim_{\mu \rightarrow -\infty} S(\mu) = 0$ . Moreover, since  $E_+(\mu) + c_0 \geq c_0 > 0$ , we have

$$\lim_{\mu \rightarrow -\infty} F(\mu) = 0.$$

Now we consider the case when  $\mu \rightarrow +\infty$ . Since

$$\begin{aligned} \frac{e^{\xi_u(\mu)}}{E_+(\mu) + c_0} &= \frac{\exp(\lambda(\mathbb{1}(\{u \in \mathcal{N}_c\}) + \mu d_u))}{\sum_{v \in \mathcal{N}_{c+1}} \exp(\lambda(\mathbb{1}(\{v \in \mathcal{N}_c\}) + \mu d_v)) + c_0} \\ &= \frac{1}{\sum_{v \in \mathcal{N}_{c+1}} \exp(\lambda(\mathbb{1}(\{v \in \mathcal{N}_c\}) - \mathbb{1}(\{u \in \mathcal{N}_c\}) + \mu(d_v - d_u))) + c_0}. \end{aligned}$$

As  $\mu \rightarrow +\infty$ , we can obtain that if  $d_u < d_{\max}$ , then

$$\lim_{\mu \rightarrow +\infty} \frac{e^{\xi_u(\mu)}}{E_+(\mu) + c_0} \leq \lim_{\mu \rightarrow +\infty} \frac{1}{\exp(\lambda(-1 + \mu(d_{\max} - d_u))) + c_0} = 0.$$

If  $d_u = d_{\max}$ , then

$$\begin{aligned} &\lim_{\mu \rightarrow +\infty} \frac{e^{\xi_u(\mu)}}{E_+(\mu) + c_0} \\ &= \lim_{\mu \rightarrow +\infty} \frac{\exp(\lambda \cdot \mathbb{1}(\{u \in \mathcal{N}_c\}))}{\sum_{v \in D_{\max}} \exp(\lambda \cdot \mathbb{1}(\{v \in \mathcal{N}_c\})) + \sum_{v \in \mathcal{N}_{c+1} \setminus D_{\max}} \exp(\lambda(\mathbb{1}(\{v \in \mathcal{N}_c\}) + \mu(d_v - d_{\max}))) + \frac{c_0}{e^{\lambda \mu d_{\max}}}} \\ &= \frac{\exp(\lambda \cdot \mathbb{1}(\{u \in \mathcal{N}_c\}))}{\sum_{v \in D_{\max}} \exp(\lambda \cdot \mathbb{1}(\{v \in \mathcal{N}_c\}))}, \end{aligned}$$

where  $D_{\max} := \{u \in \mathcal{N}_{c+1} : d_u = d_{\max}\}$ . Therefore,

$$\lim_{\mu \rightarrow +\infty} F(\mu) = \sum_{u \in D_{\max}} d_u \frac{\exp(\lambda \cdot \mathbb{1}(\{u \in \mathcal{N}_c\}))}{\sum_{v \in D_{\max}} \exp(\lambda \cdot \mathbb{1}(\{v \in \mathcal{N}_c\}))} = d_{\max}.$$

Finally, the inequality  $F(\mu) < d_{\max}$  for finite  $\mu$  follows from  $S(\mu) \leq d_{\max}E_+(\mu)$  and  $c_0 > 0$ :

$$F(\mu) = \frac{S(\mu)}{E_+(\mu) + c_0} \leq \frac{d_{\max}E_+(\mu)}{E_+(\mu) + c_0} < d_{\max}.$$

**(3) Finite fixed point and monotone convergence for  $d_\star < d_{\max}$ .** By (2),  $F(\cdot)$  is continuous, strictly increasing, with range  $(0, d_{\max})$ . Therefore, there exists a unique  $\mu^\star \in \mathbb{R}$  such that  $F(\mu^\star) = d_\star$ . Now we first argue  $\mu(t) \rightarrow \mu^\star$ .

Consider the ODE  $\dot{\mu} = c(d_\star - F(\mu))$  with  $c = \alpha/(n\sqrt{K}) > 0$ . If  $\mu(0) = 0 \leq \mu^\star$  and (thus)  $F(\mu(0)) \leq F(\mu^\star) = d_\star$ , then  $\dot{\mu}(t) \geq 0$  as long as  $\mu(t) \leq \mu^\star$ , hence  $\mu$  is non-decreasing and bounded above by  $\mu^\star$ ; monotone convergence implies  $\mu(t) \rightarrow \bar{\mu} \leq \mu^\star$  for some  $\bar{\mu}$ . By the continuity of  $F(\cdot)$  and the fact that  $\dot{\mu} \rightarrow 0$ , we can obtain that  $F(\bar{\mu}) = d_\star$ , which implies  $\bar{\mu} = \mu^\star$ . The case  $\mu(0) > \mu^\star$  is analogous with a non-increasing trajectory.

For local exponential stability, we can set  $\tilde{\mu} = \mu - \mu^\star$  and write

$$\dot{\tilde{\mu}}(t) = -c(F(\mu^\star + \tilde{\mu}(t)) - F(\mu^\star)).$$

By the mean value theorem,  $F(\mu^\star + \tilde{\mu}) - F(\mu^\star) = F'(\xi)\tilde{\mu}$  for some  $\xi$  between  $\mu^\star$  and  $\mu^\star + \tilde{\mu}$ . Since  $F'(\mu^\star) > 0$  and  $F'$  is continuous, there exists  $\eta > 0$  and  $m > 0$  such that  $F'(\xi) \geq m$  whenever  $|\xi - \mu^\star| \leq \eta$ . Hence, as long as  $|\tilde{\mu}(t)| \leq \eta$ , we have

$$\frac{d}{dt} |\tilde{\mu}(t)| = \frac{\dot{\tilde{\mu}}(t)}{|\tilde{\mu}(t)|} \dot{\tilde{\mu}}(t) = -c F'(\xi(t)) |\tilde{\mu}(t)| \leq -cm |\tilde{\mu}(t)|.$$

Applying Gronwall's inequality, we have  $|\tilde{\mu}(t)| \leq e^{-cmt} |\tilde{\mu}(0)|$  in this neighborhood, which establishes local exponential convergence.

**(4) Divergence and logarithmic lower bound for  $d_\star = d_{\max}$ .** When  $d_\star = d_{\max}$ , since  $F(\mu) < d_{\max}$  for all finite  $\mu$ , we have  $\dot{\mu}(t) = c(d_{\max} - F(\mu(t))) > 0$  where  $c = \frac{\alpha}{n\sqrt{K}}$  and thus  $\mu(t)$  is strictly increasing. We now lower bound the growth rate similar to Lemma 38.

Since  $S(\mu) \leq d_{\max}E_+(\mu)$ , we have

$$d_{\max} - F(\mu) = d_{\max} - \frac{S(\mu)}{E_+(\mu) + c_0} \geq d_{\max} \left( 1 - \frac{E_+(\mu)}{E_+(\mu) + c_0} \right) = \frac{d_{\max} c_0}{E_+(\mu) + c_0}.$$

Moreover, for each  $u \in \mathcal{N}_{c+1}$ , it holds that

$$e^{\xi u(\mu)} \leq \exp(\lambda(1 + \mu d_{\max})).$$

Therefore,  $E_+(\mu) \leq |\mathcal{N}_{c+1}| e^{\lambda(1 + \mu d_{\max})} \leq n e^{\lambda(1 + \mu d_{\max})}$  and thus we can obtain that

$$E_+(\mu) + c_0 \leq n e^{\lambda(1 + \mu d_{\max})} + c_0 \leq n (e^{\lambda(1 + \mu d_{\max})} + 1) \leq 2n e^{\lambda(1 + \mu d_{\max})},$$

where we used  $e^x \geq 1$  for  $x \geq 0$ . Combining the above derivation, we can obtain that

$$d_{\max} - F(\mu) \geq \frac{d_{\max} c_0}{2n} e^{-\lambda} e^{-\lambda d_{\max} \mu}.$$

We can then plug this into  $\dot{\mu} = c(d_{\max} - F(\mu))$  with  $c = \alpha/(n\sqrt{K})$  to get

$$\dot{\mu}(t) \geq \frac{\alpha}{n\sqrt{K}} \cdot \frac{d_{\max} c_0}{2n} e^{-\lambda} e^{-\lambda d_{\max} \mu(t)} = c_1 e^{-c_2 \mu(t)},$$

where  $c_1 = \frac{\alpha d_{\max} c_0 e^{-\lambda}}{2n^2\sqrt{K}}$  and  $c_2 = \lambda d_{\max} > 0$ .

Applying Lemma 46, we can obtain exactly (D.4). This shows  $\mu(t) \rightarrow +\infty$  at least logarithmically fast.  $\square$

### D.2.3 Thought Expansion

Finally, we provide results for continuous thought expansion. Note that the following results hold for any directed graph that differs from the graphs in the training set.

**Theorem 20** (One-hop expansion of continuous thoughts). *Let  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  be any directed graph (which can differ from the graphs in the training set) and  $\mathbf{r} \in \mathcal{V}$  be a root node. Assume the current thought is any positive superposition on  $\mathcal{N}_c^{\mathcal{G}}(\mathbf{r})$ :*

$$\hat{\mathbf{v}}_{[t_c]} = \sum_{u \in \mathcal{N}_c} \lambda_u \mathbf{E}(u), \quad \lambda_u > 0.$$

Then the next continuous thought  $[t_{c+1}] = \hat{\mathbf{v}}_{[t_{c+1}]}$  generated by the forward pass (5.2) satisfies

$$\boldsymbol{\xi} = \mathbf{U}^\top \hat{\mathbf{v}}_{[t_{c+1}]} = \sum_{v \in \mathcal{N}_{c+1}} \beta_v \mathbf{e}_v,$$

with coefficients

$$\beta_v = \underbrace{\lambda_v \mathbb{1}(\{v \in \mathcal{N}_c\})}_{\text{carryover}} + \underbrace{\mu \sum_{u \in \mathcal{N}_c} \lambda_u \mathbb{1}(\{(u \rightarrow v) \in \mathcal{E}\})}_{\text{one-hop expansion}} \geq 0. \quad (\text{D.5})$$

where we assume the model has been trained until time  $t$  and the trained model satisfies  $\mu_v(t) \equiv \mu > 0$  in (5.3). In particular,  $\beta_v > 0$  for every  $v \in \mathcal{N}_{c+1}$  if  $\mu > 0$ , so the support of  $\boldsymbol{\xi}$  is exactly  $\mathcal{N}_{c+1}$ , and the output is a superposition of  $\mathcal{N}_{c+1}$ .

*Proof.* Note that  $\hat{\mathbf{v}}_i = \mathbf{E}_s(\mathbf{s}_i) + \mathbf{E}_t(\mathbf{t}_i)$  and  $\mathbf{W} = \sum_{v \in \mathcal{V}} \mu_v(t) \mathbf{E}(v) \mathbf{E}_s(v)^\top = \sum_{v \in \mathcal{V}} \mu \mathbf{E}(v) \mathbf{E}_s(v)^\top$ . We can calculate that

$$\mathbf{W} \hat{\mathbf{v}}_i = \sum_{v \in \mathcal{V}} \mu \mathbf{E}(v) \mathbf{E}_s(v)^\top (\mathbf{E}_s(\mathbf{s}_i) + \mathbf{E}_t(\mathbf{t}_i)) = \mu \mathbf{E}(\mathbf{s}_i),$$

where we used  $\mathbf{E}_s(v)^\top \mathbf{E}_s(\mathbf{s}_i) = \mathbb{1}(\{v = \mathbf{s}_i\})$  and  $\mathbf{E}_s(v)^\top \mathbf{E}_t(\mathbf{t}_i) = 0$  according to Assumption 3. Therefore, with  $\hat{\mathbf{v}}_{[t_c]} = \sum_{u \in \mathcal{N}_c} \lambda_u \mathbf{E}(u)$ , we can calculate

$$\alpha_i := \hat{\mathbf{v}}_{[t_c]}^\top \mathbf{W} \hat{\mathbf{v}}_i = \mu \sum_{u \in \mathcal{N}_c} \lambda_u \mathbf{E}(u)^\top \mathbf{E}(\mathbf{s}_i) = \mu \lambda_{\mathbf{s}_i} \mathbb{1}(\{\mathbf{s}_i \in \mathcal{N}_c\}).$$

The value aggregation becomes

$$\phi(\hat{\mathbf{v}}_{[\mathbf{t}_c]}; \{\hat{\mathbf{v}}_i\}) = \sum_{i=1}^m \alpha_i \hat{\mathbf{v}}_i = \mu \sum_{i: \mathbf{s}_i \in \mathcal{N}_c} \lambda_{\mathbf{s}_i} (\mathbf{E}_{\mathbf{s}}(\mathbf{s}_i) + \mathbf{E}_{\mathbf{t}}(\mathbf{t}_i)).$$

Furthermore, we have

$$\mathbf{U}^\top \phi(\hat{\mathbf{v}}_{[\mathbf{t}_c]}; \{\hat{\mathbf{v}}_i\}) = \mu \sum_{i: \mathbf{s}_i \in \mathcal{N}_c} \lambda_{\mathbf{s}_i} \mathbf{e}_{\mathbf{t}_i} = \mu \sum_{v \in \mathcal{V}} \left( \sum_{u \in \mathcal{N}_c} \lambda_u \mathbb{1}(\{u \rightarrow v\} \in \mathcal{E}) \right) \mathbf{e}_v.$$

Similarly,

$$\mathbf{U}^\top \hat{\mathbf{v}}_{[\mathbf{t}_c]} = \sum_{v \in \mathcal{N}_c} \lambda_v \mathbf{e}_v = \sum_{v \in \mathcal{V}} \lambda_v \mathbb{1}(\{v \in \mathcal{N}_c\}) \mathbf{e}_v.$$

Adding the two parts yields  $\boldsymbol{\xi} = \sum_v \beta_v \mathbf{e}_v$  with  $\beta_v$  as in (D.5), since by definition we have  $\hat{\mathbf{v}}_{[\mathbf{t}_{c+1}]} = \hat{\mathbf{v}}_{[\mathbf{t}_c]} + \phi(\hat{\mathbf{v}}_{[\mathbf{t}_c]}; \{\hat{\mathbf{v}}_i\})$  and  $\boldsymbol{\xi} = \mathbf{U}^\top \hat{\mathbf{v}}_{[\mathbf{t}_{c+1}]}$ .  $\square$

### D.3 Missing Proofs for Section 5.4

In this section, we analyze the training dynamics of the prediction stage, i.e., after thought generation, how the model extracts the information from the generated continuous thought to make the final prediction.

Recall that the  $i$ -th training sample consists of  $(\mathcal{G}^{(i)}, \mathbf{r}^{(i)}, \mathbf{c}_1^{(i)}, \mathbf{c}_2^{(i)}, \boldsymbol{\lambda}^{(i)})$ , and we denote  $\mathbf{c}_\star^{(i)}$  as the reachable candidate and  $\mathbf{c}_\perp^{(i)}$  as the unreachable candidate. We also use  $\mathcal{N}_C^{(i)} = \mathcal{N}_C^{\mathcal{G}^{(i)}}(\mathbf{r}^{(i)})$  to denote the  $C$ -ball for the  $i$ -th training sample. We assume  $C$  is large enough so that  $\mathbf{c}_\star^{(i)} \in \mathcal{N}_C^{(i)}$  for any  $i$ . Note that  $\mathbf{c}_\perp^{(i)} \notin \mathcal{N}_C^{(i)}$  for any  $C$  by definition. For notation convenience, we also use  $\mu_A = \mu_{\langle A \rangle}$ ,  $\mu_R = \mu_{\langle R \rangle}$ , and denote  $\boldsymbol{\xi}^{(i)} = \{\xi_v^{(i)}\}_{v \in \mathcal{V}_{\text{oc}}}$  as the logits calculated by forward pass (5.7) for the  $i$ -th training sample. We denote  $\xi_{\mathbf{c}_t^{(i)}}^{(i)} = \xi_{\mathbf{c}_t}^{(i)}$ ,  $\lambda_{\mathbf{c}_t^{(i)}}^{(i)} = \lambda_{\mathbf{c}_t}^{(i)}$  for  $t \in \{1, 2, \star, \perp\}$  for notation convenience.

To start with, we first provide a closed-form logit expression.

**Lemma 41** (Closed-form logits in prediction stage). *Under reparameterization (5.8) and forward pass for the prediction stage (5.7), for every  $v \in \mathcal{V}$  we have*

$$\xi_v(\mu_{\langle A \rangle}, \mu_{\langle R \rangle}) = \underbrace{\mu_{\langle A \rangle} \lambda_v}_{\text{frontier residual}} + \underbrace{\mu_{\langle R \rangle} \mathbb{1}(\{v \in \{\mathbf{c}_1, \mathbf{c}_2\}\})}_{\text{candidate lift}}. \quad (\text{D.6})$$

In particular,

$$\xi_{\mathbf{c}_\star} - \xi_{\mathbf{c}_\perp} = \mu_{\langle A \rangle} \lambda_{\mathbf{c}_\star}. \quad (\text{D.7})$$

*Proof.* For the reasoning token  $\hat{\mathbf{v}}_{\text{idx}(\langle R \rangle)} = \mathbf{E}(\langle R \rangle) + \mathbf{E}(\mathbf{c}_1) + \mathbf{E}(\mathbf{c}_2)$ , we have

$$\mathbf{W} \hat{\mathbf{v}}_{\text{idx}(\langle R \rangle)} = \mu_{\langle R \rangle} \mathbf{E}(\langle A \rangle) \underbrace{\mathbf{E}(\langle R \rangle)^\top \hat{\mathbf{v}}_{\text{idx}(\langle R \rangle)}}_{=1} = \mu_{\langle R \rangle} \mathbf{E}(\langle A \rangle).$$

Therefore,

$$\mathbf{U}^\top((\hat{\mathbf{v}}_{\text{IdX}(\langle A \rangle)}^\top \mathbf{W} \hat{\mathbf{v}}_{\text{IdX}(\langle R \rangle)}) \hat{\mathbf{v}}_{\text{IdX}(\langle R \rangle)}) = \mu_{\langle R \rangle}(\mathbf{e}_{\langle R \rangle} + \mathbf{e}_{\mathbf{c}_1} + \mathbf{e}_{\mathbf{c}_2}).$$

Also,

$$\mathbf{U}^\top(\mu_{\langle A \rangle} \hat{\mathbf{v}}_{\text{IdX}(\langle A \rangle)}) = \mu_{\langle A \rangle} \sum_{u \in \mathcal{N}_C} \lambda_u \mathbf{e}_u + \mu_{\langle A \rangle} \mathbf{e}_{\langle A \rangle}.$$

Combining the above two expressions yields (D.6). For  $\mathbf{c}_\perp \notin \mathcal{N}_C$  we have  $\lambda_{\mathbf{c}_\perp} = 0$ , and (D.7) follows.  $\square$

For each training sample, we can construct a two-dimensional feature for every node  $v \in \mathcal{V}$ :

$$x_v^{(i)} := (\lambda_v^{(i)}, \mathbb{1}(\{v \in \{\mathbf{c}_1^{(i)}, \mathbf{c}_2^{(i)}\}\})) \in \mathbb{R}_{\geq 0}^2,$$

so that  $\xi_v^{(i)}(\mu_A, \mu_R) = \langle w, x_v^{(i)} \rangle$  with  $w := (\mu_A, \mu_R) \in \mathbb{R}_{\geq 0}^2$ . For instance  $i$ , a correct classification means  $\langle w, x_{\mathbf{c}_\star}^{(i)} - x_v^{(i)} \rangle > 0$  for all  $v \neq \mathbf{c}_\star^{(i)}$ , where we denote  $x_{\mathbf{c}_t}^{(i)} = x_{\mathbf{c}_t}^{(i)}$  for  $t \in \{1, 2, \star, \perp\}$ . We further define the difference of features with respect to  $\mathbf{c}_\star$  for later use:

$$\Delta_{i,v} := x_{\mathbf{c}_\star}^{(i)} - x_v^{(i)} = \begin{cases} (\lambda_{\mathbf{c}_\star}^{(i)}, 0), & v = \mathbf{c}_\perp^{(i)}, \\ (\lambda_{\mathbf{c}_\star}^{(i)} - \lambda_v^{(i)}, 1), & v \in \mathcal{N}_C^{(i)} \setminus \{\mathbf{c}_\star^{(i)}\}, \\ (\lambda_{\mathbf{c}_\star}^{(i)}, 1), & v \notin \mathcal{N}_C^{(i)} \cup \{\mathbf{c}_\perp^{(i)}\}. \end{cases} \quad (\text{D.8})$$

### D.3.1 Linearly separable structure and a max-margin problem

**Lemma 42** (Separation by a nonnegative direction). *For every instance  $i$  and each competitor  $v \neq \mathbf{c}_\star^{(i)}$ ,*

$$\langle (1, 1), \Delta_{i,v} \rangle = \begin{cases} \lambda_{\mathbf{c}_\star}^{(i)}, & v = \mathbf{c}_\perp^{(i)}, \\ \lambda_{\mathbf{c}_\star}^{(i)} - \lambda_v^{(i)} + 1, & v \in \mathcal{N}_C^{(i)} \setminus \{\mathbf{c}_\star^{(i)}\}, \\ \lambda_{\mathbf{c}_\star}^{(i)} + 1, & v \notin \mathcal{N}_C^{(i)} \cup \{\mathbf{c}_\perp^{(i)}\}, \end{cases} > 0.$$

Hence, the training data are linearly separable by a direction in  $\mathbb{R}_{\geq 0}^2$ .

*Proof.* The result holds because  $\lambda_{\mathbf{c}_\star}^{(i)} > 0$ ,  $\lambda_v^{(i)} \leq 1$ .  $\square$

Define the *hard-margin* value of a unit direction  $u \in \mathbb{S}^1 \cap \mathbb{R}_{\geq 0}^2$  (where  $\mathbb{S}^1 = \{u \in \mathbb{R}^2 : \|u\|_2 = 1\}$ ) as

$$\gamma(u) := \min_i \min_{v \neq \mathbf{c}_\star^{(i)}} \langle u, \Delta_{i,v} \rangle.$$

The corresponding *maximum-margin* direction is

$$u^\star \in \arg \max_{u \in \mathbb{S}^1 \cap \mathbb{R}_{\geq 0}^2} \gamma(u). \quad (\text{D.9})$$

We characterize  $u^\star$  using the following two quantities of the training sets:

$$\lambda_\star := \min_i \lambda_{\mathbf{c}_\star}^{(i)} \in (0, 1], \quad \Delta_{\text{train}} := \max_i \max_{v \in \mathcal{N}_C^{(i)} \setminus \{\mathbf{c}_\star^{(i)}\}} (\lambda_v^{(i)} - \lambda_{\mathbf{c}_\star}^{(i)})_+ \in [0, 1],$$

where  $(x)_+ := \max\{x, 0\}$ . Intuitively,  $\lambda_\star$  is the smallest mass ever placed on a reachable candidate across the training set, and  $\Delta_{\text{train}}$  is the largest overshoot of a non-candidate but reachable node's weight relative to the reachable candidate.

**Lemma 43** (Closed-form lower envelope of the margin). *For any unit  $u = (u_A, u_R) \in \mathbb{S}^1 \cap \mathbb{R}_{\geq 0}^2$ ,*

$$\gamma(u) = \min\{u_A \lambda_\star, u_R - u_A \Delta_{\text{train}}, u_A \lambda_\star + u_R\} = \min\{u_A \lambda_\star, u_R - u_A \Delta_{\text{train}}\}.$$

*Proof.* According to (D.8), we can directly obtain the lower bounds  $u_A \lambda_{c_\star}^{(i)}$ ,  $u_R + u_A(\lambda_{c_\star}^{(i)} - \lambda_v^{(i)})$ , and  $u_A \lambda_{c_\star}^{(i)} + u_R$ . Minimizing over  $i$  and  $v$  according to the definition of  $\lambda_\star, \Delta_{\text{train}}$  yields the desired result.  $\square$

**Proposition 9** (Properties of the maximum-margin direction). *Let  $u^\star = (u_A^\star, u_R^\star)$  be a solution of (D.9). Then the unique maximizer satisfies*

$$\frac{u_R^\star}{u_A^\star} = \lambda_\star + \Delta_{\text{train}}, \quad u_A^\star = \frac{1}{\sqrt{1 + (\lambda_\star + \Delta_{\text{train}})^2}}, \quad u_R^\star = \frac{\lambda_\star + \Delta_{\text{train}}}{\sqrt{1 + (\lambda_\star + \Delta_{\text{train}})^2}}.$$

*Proof.* By Lemma 43, we can maximize  $\gamma(u) = \min\{u_A \lambda_\star, u_R - u_A \Delta_{\text{train}}\}$  over the unit vector  $u$  by equalizing the two arguments (otherwise one can rotate  $u$  to increase the minimum). Therefore, we can equalize the two arguments, which yields  $u_R = u_A(\lambda_\star + \Delta_{\text{train}})$ , and obtain the desired result.  $\square$

### D.3.2 Implicit bias of gradient flow and directional convergence

$$\ell_{\mathcal{G}, \mathbf{r}, \mathbf{c}_1, \mathbf{c}_2, \boldsymbol{\lambda}}^{\text{pred}} := -\log \frac{\exp(\xi_{c_\star})}{\sum_{v \in \mathcal{V}} \exp(\xi_v)}, \quad \mathcal{L}^{\text{pred}} = \mathbb{E}_{(\mathcal{G}, \mathbf{r}, \mathbf{c}_1, \mathbf{c}_2, \boldsymbol{\lambda}) \sim \mathcal{D}}[\ell_{\mathcal{G}, \mathbf{r}, \mathbf{c}_1, \mathbf{c}_2, \boldsymbol{\lambda}}^{\text{pred}}],$$

Recall the loss function on the prediction stage over the training set (5.9). We can rewrite it as follows

$$\mathcal{L}(\mu_A, \mu_R) := \frac{1}{N} \sum_{i=1}^N \ell^{(i)}(\mu_A, \mu_R), \quad \ell^{(i)}(\mu_A, \mu_R) := -\log \frac{\exp(\xi_{c_\star}^{(i)})}{\sum_{v \in \mathcal{V}} \exp(\xi_v^{(i)})},$$

and run the gradient-flow dynamics  $\dot{w}(t) = -\alpha \nabla \mathcal{L}(w(t))$  with  $w(t) = (\mu_A(t), \mu_R(t))$  and  $\alpha > 0$ . By Lemma 42, the data are linearly separable, so the implicit bias of gradient flow directly yields the following lemma.

**Lemma 44** (Implicit bias of gradient flow). *Along gradient flow from any bounded initialization  $w(0)$ , we have*

$$\|w(t)\| \rightarrow \infty, \quad \frac{w(t)}{\|w(t)\|} \rightarrow u^\star,$$

where  $u^*$  is the unique solution to the maximum-margin problem (D.9). Combining Proposition 9, there exists a scalar radius  $r(t) \rightarrow \infty$  such that

$$(\mu_A(t), \mu_R(t)) = r(t) u^* + o(r(t)),$$

and for any  $\varepsilon > 0$ ,

$$\frac{\mu_R(t)}{\mu_A(t)} \geq \lambda_\star + \Delta_{\text{train}} - \varepsilon \quad \text{for all sufficiently large } t.$$

The proof can be straightforwardly adapted from its gradient descent counterpart [120].

### D.3.3 Prediction on unseen graphs

Finally, we show that after sufficient training, the model can correctly predict the reachable candidate node even for unseen graphs, showcasing its generalization capability.

Fix any unseen test graph along with the exploration set  $\mathcal{N}_C^{\text{test}}$  and weights  $\lambda^{\text{test}}$ , such that  $\lambda_v^{\text{test}} \in (0, 1]$  on  $\mathcal{N}_C^{\text{test}}$  and 0 otherwise. The test graph also satisfies

$$\max_{u \in \mathcal{N}_C^{\text{test}}} \lambda_u^{\text{test}} - \lambda_{c_\star}^{\text{test}} \leq \Delta, \quad \text{with} \quad \Delta \leq \Delta_{\text{train}}.$$

Therefore, for every non-candidate  $v \in \mathcal{N}_C^{\text{test}} \setminus \{c_\star^{\text{test}}\}$ , it holds that  $\lambda_v^{\text{test}} \leq \lambda_{c_\star}^{\text{test}} + \Delta$ .

The following lemma shows that as long as the test graph satisfies the above condition, it has a positive margin using the maximum margin direction for the training set  $u^*$ .

**Lemma 45** (Positive test-time margins from the trained direction). *Let  $u^* = (u_A^*, u_R^*)$  be the unique max-margin direction with  $u_R^*/u_A^* = \lambda_\star + \Delta_{\text{train}} > \Delta$ . Then for every competitor  $v \neq c_\star^{\text{test}}$ ,*

$$\langle u^*, x_{c_\star}^{\text{test}} - x_v^{\text{test}} \rangle \geq \min\{u_A^* \lambda_\star, u_A^* \lambda_\star^{\text{test}}\} > 0.$$

*Proof.* For  $v = c_\perp^{\text{test}}$ , the difference is  $(\lambda_{c_\star}^{\text{test}}, 0)$ ; since  $\lambda_{c_\star}^{\text{test}} > 0$  we have  $\langle u^*, x_{c_\star}^{\text{test}} - x_{c_\perp}^{\text{test}} \rangle \geq u_A^* \lambda_{c_\star}^{\text{test}} > 0$ .

For  $v \notin \mathcal{N}_C^{\text{test}}$  the difference is  $(\lambda_{c_\star}^{\text{test}}, 1)$  and the bound is even larger.

For  $v \in \mathcal{N}_C^{\text{test}} \setminus \{c_\star^{\text{test}}\}$ , we have  $\lambda_v^{\text{test}} \leq \lambda_{c_\star}^{\text{test}} + \Delta$ , hence

$$\langle u^*, x_{c_\star}^{\text{test}} - x_v^{\text{test}} \rangle = u_A^* (\lambda_{c_\star}^{\text{test}} - \lambda_v^{\text{test}}) + u_R^* \geq u_R^* - u_A^* \Delta \geq u_A^* (\lambda_\star + \Delta_{\text{train}} - \Delta) \geq u_A^* \lambda_\star > 0.$$

□

Finally, we show that after sufficient training, the model can correctly predict the reachable candidate node .

**Theorem 21** (Generalization for unseen graphs). *Let  $(\mu_A(t), \mu_R(t))$  follow gradient flow on the loss (5.9) from any bounded initialization. Suppose the training set is linearly separable and  $\lambda_\star, \Delta_{\text{train}}$  are defined as above. Then, for any unseen instance satisfying  $\lambda_v^{\text{test}} \in (0, 1]$  on  $\mathcal{N}_C^{\text{test}}$  and 0 otherwise, and*

$$\max_{u \in \mathcal{N}_C^{\text{test}}} \lambda_u^{\text{test}} - \lambda_{c_\star}^{\text{test}} \leq \Delta, \quad \text{with } \Delta \leq \Delta_{\text{train}},$$

we have for all sufficiently large  $t$ :

$$p_{c_\star^{\text{test}}}(t) := \frac{\exp(\xi_{c_\star^{\text{test}}}^{\text{test}}(\mu_A(t), \mu_R(t)))}{\sum_v \exp(\xi_v^{\text{test}}(\mu_A(t), \mu_R(t)))} \rightarrow 1.$$

*Proof.* By Lemma 44, we have

$$(\mu_A(t), \mu_R(t)) = r(t)u^\star + o(r(t)), \quad r(t) \rightarrow \infty.$$

Then, by Lemma 45, for every competitor  $v \neq c_\star^{\text{test}}$ ,

$$\begin{aligned} & \xi_{c_\star^{\text{test}}}^{\text{test}}(\mu_A(t), \mu_R(t)) - \xi_v^{\text{test}}(\mu_A(t), \mu_R(t)) \\ &= r(t) \langle u^\star, x_{c_\star^{\text{test}}}^{\text{test}} - x_v^{\text{test}} \rangle + o(r(t)) \\ &\geq r(t) \cdot \min\{u_A^\star \lambda_\star, u_A^\star \lambda_\star^{\text{test}}\} + o(r(t)) \xrightarrow{t \rightarrow \infty} +\infty. \end{aligned}$$

Hence the arg max is  $c_\star^{\text{test}}$ , and its softmax probability tends to 1.  $\square$

## D.4 Auxiliary Lemmas

**Lemma 46** (ODE lower bound). *Let  $c_1, c_2 > 0$  be two constants. Assume the function  $f : \mathbb{R} \rightarrow \mathbb{R}$  satisfies  $f(0) = 0$  and*

$$\frac{df(t)}{dt} \geq c_1 \cdot \exp(-c_2 \cdot f(t)), \quad \forall t \geq 0.$$

Then it holds

$$f(t) \geq \frac{1}{c_2} \ln(1 + c_1 c_2 t)$$

for all  $t \geq 0$ .

*Proof.* We define  $g(t) = e^{c_2 f(t)}$ . Note that

$$\frac{dg(t)}{dt} = \frac{d}{dt}(e^{c_2 f(t)}) = c_2 \frac{df(t)}{dt} \cdot e^{c_2 f(t)} \geq c_2 \cdot c_1 \cdot \exp(-c_2 \cdot f(t)) \exp(c_2 f(t)) = c_1 c_2.$$

Therefore,  $dg(t) \geq c_1 c_2 dt$  for  $t \geq 0$ , and thus

$$\int_0^t dg(t) \geq \int_0^t c_1 c_2 dt \implies g(t) - g(0) \geq c_1 c_2 t.$$

Therefore,

$$g(t) = e^{c_2 f(t)} \geq g(0) + c_1 c_2 t = e^{c_2 f(0)} + c_1 c_2 t = 1 + c_1 c_2 t,$$

which implies

$$f(t) \geq \frac{1}{c_2} \ln(1 + c_1 c_2 t).$$

□

## D.5 Experiment Details

### D.5.1 Dataset

Table D.3: ProsQA statistics. Numbers are averaged over problem instances.

|       | #Problems | $ V $ | $ E $ | Sol. Len. |
|-------|-----------|-------|-------|-----------|
| Train | 14785     | 22.8  | 36.5  | 3.5       |
| Val   | 257       | 22.7  | 36.3  | 3.5       |
| Test  | 419       | 22.7  | 36.0  | 3.5       |

The statistics of the ProsQA dataset is shown in [Table D.3](#).

### D.5.2 Experiment with Coconut-BFS

As a comparison, we also train a model with a modified version of  $\mathcal{L}^{\text{BFS}}$ . Recall that the original  $\mathcal{L}^{\text{BFS}}$  (5.5) encourages the model to predict any nodes within  $\mathcal{N}_{c+1}$ . To avoid the trivial solution of always predicting the root node, we introduce an experimental variant that only encourages predicting nodes on the current frontier:

$$\text{Coconut-BFS-exp: } \ell_{\mathcal{G}, \mathbf{x}}^{\text{BFS-exp}} := -\log \frac{\sum_{v \in \mathcal{N}_{c+1} \setminus \mathcal{N}_c} \exp(\xi_v)}{\sum_{v \in \mathcal{V}} \exp(\xi_v)}. \quad (\text{D.10})$$

All other training settings remain unchanged. The answer accuracy of this model on the test set is 99.0%. We then track the logit difference between frontier and non-frontier edges as a proxy for  $\mu_v$ , with results shown in [Figure D.1](#).

In Stage 1, the logit difference for  $c = 1$  grows much faster than under  $\mathcal{L}^{\text{coco}}$  and shows no sign of saturation even after 150 epochs. This agrees with the theoretical prediction in [Theorem 9](#): under COCONUT-BFS,  $\mu_v$  diverges rather than stabilizing. At later steps ( $c = 3, 4$ ), the gap between COCONUT-BFS and COCONUT becomes smaller. We attribute this to practical factors such as stage-wise data mixing, gradient propagation across earlier thoughts, and a larger discrepancy between losses (5.5) and (D.10) in the later stage.

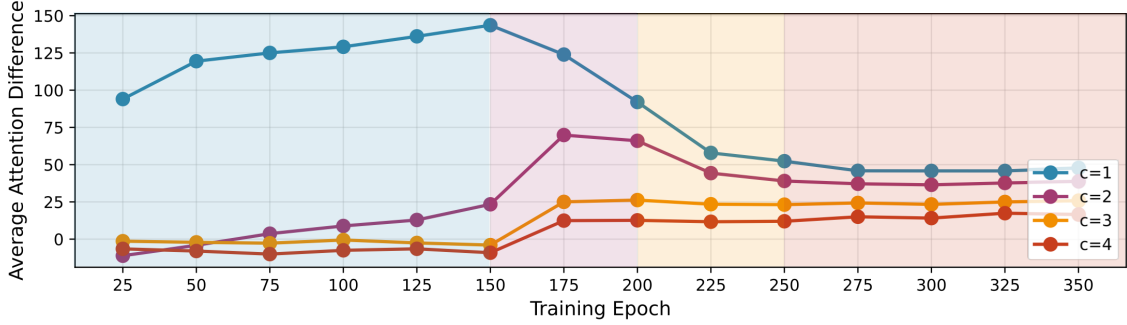


Figure D.1: The attention logits difference between frontier edges and others. The model is trained with a modified version of  $\mathcal{L}^{\text{BFS}}$ .

### D.5.3 Alternative Attention Routes for Candidate Lift

Our theoretical analysis in Lemma 6 assumes that  $\langle R \rangle$  copies the candidate nodes in the first layer, and  $\langle A \rangle$  then attends to  $\langle R \rangle$  in the second layer. In practice, however, we observe three distinct yet functionally equivalent attention routes that realize the same *candidate lift*. Example attention maps for each route are shown in [Figure D.2](#).

## D.6 Additional Experimental Results

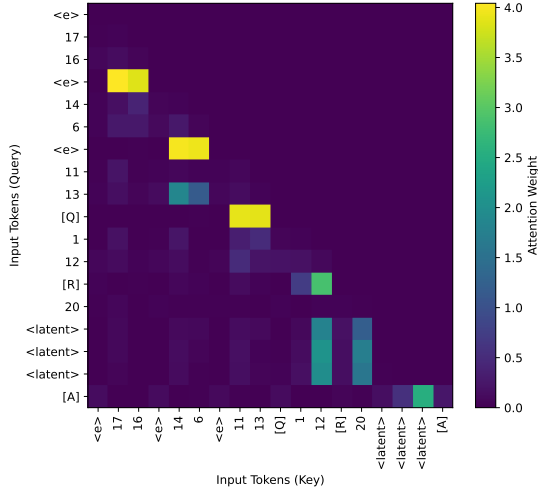
In this section, we provide additional experiments to complement our main analysis of training dynamics.

### D.6.1 Ablation Study: Architectural and Optimization Sensitivity

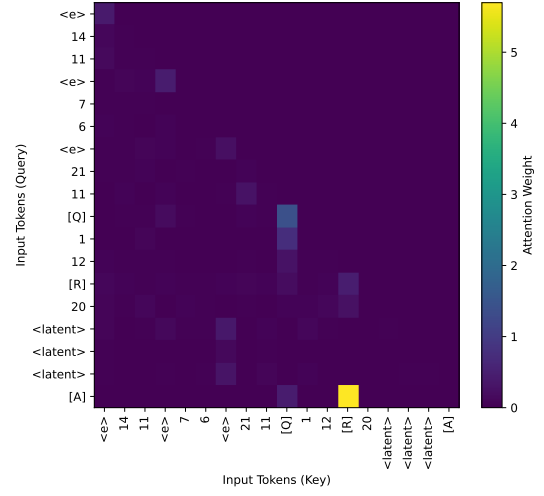
We evaluate the sensitivity of COCONUT training to model depth, number of attention heads, hidden width, and learning rate. The results are summarized in [Table D.4](#).

We observe that models with  $L = \{4, 8\}$  layers maintain high accuracy, while  $L = 12$  is harder to optimize. The performance remains comparable when  $d_{\text{model}} \in \{768, 1536\}$ , but degrades when the width is too small (e.g.,  $d_{\text{model}} = 384$ ). Varying the number of heads does not have major effects on final accuracy, whereas too large or too small learning rates tend to degrade performance. Weight tying setting does not affect model performance.

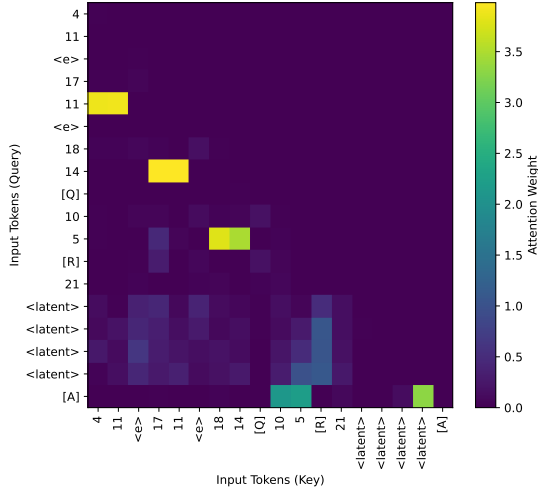
# APPENDIX D. APPENDIX FOR TRAINING DYNAMICS OF CONTINUOUS COT180



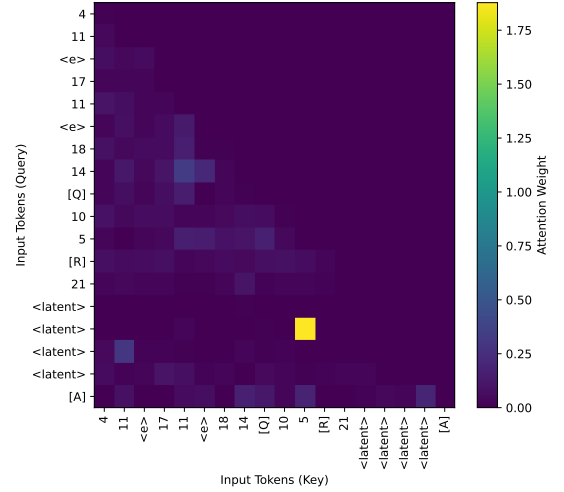
(a) Layer-1 attention map for Pattern A



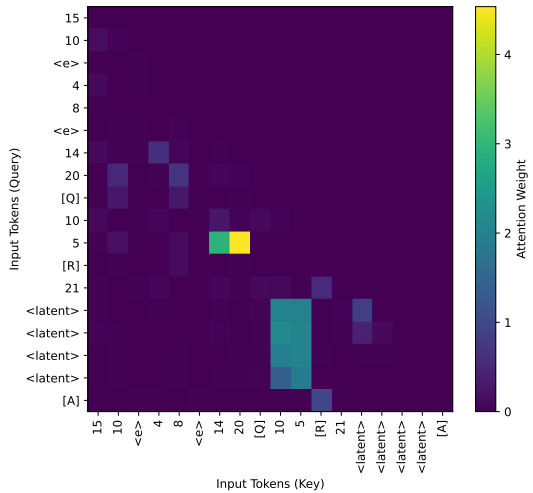
(b) Layer-2 attention map for Pattern A



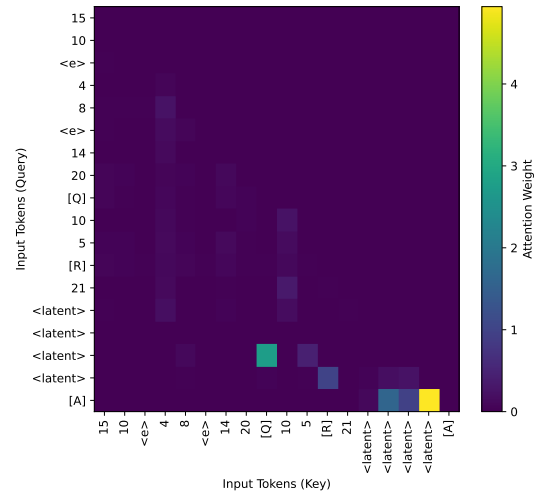
(c) Layer-1 attention map for Pattern B



(d) Layer-2 attention map for Pattern B



(e) Layer-1 attention map for Pattern C



(f) Layer-2 attention map for Pattern C

| Number of layers |      | Number of heads |      | Width                     |      |
|------------------|------|-----------------|------|---------------------------|------|
| $L = 2$          | 98.8 | $H = 4$         | 98.0 | $d_{\text{model}} = 384$  | 62.0 |
| $L = 4$          | 97.3 | $H = 8$         | 98.8 | $d_{\text{model}} = 768$  | 98.8 |
| $L = 8$          | 96.5 | $H = 12$        | 98.8 | $d_{\text{model}} = 1536$ | 97.7 |
| $L = 12$         | 67.4 |                 |      |                           |      |

| Learning rate             |      | Weight tying |      |
|---------------------------|------|--------------|------|
| $\eta = 2 \times 10^{-4}$ | 58.1 | Tied         | 98.8 |
| $\eta = 1 \times 10^{-4}$ | 98.8 | Untied       | 98.8 |
| $\eta = 5 \times 10^{-5}$ | 62.1 |              |      |

Table D.4: Ablation on depth, heads, width, learning rate, and weight tying. By default, other hyperparameters follow the main experiments.

We emphasize that each ablation in Table D.4 varies only a single hyperparameter at a time, keeping all other settings identical to our main experiment. In practice, these hyperparameters interact in a coupled manner. For instance, with a smaller learning rate of  $5 \times 10^{-5}$ , we can extend the first-stage training to 300 epochs and get 97.0% accuracy. For deeper models with  $L = 12$ , prolonging first-stage training to 400 epochs and reducing the learning rate to  $5 \times 10^{-5}$  improves accuracy to 99.6%. A comprehensive hyperparameter interaction study is beyond the scope of this work and is left for future investigation.

### D.6.2 Multi-Layer Transformers and Mechanistic Patterns

We use the COCONUT model with  $L = 4$  to analyze the reasoning pattern beyond two-layer transformers. The results are shown in Figure D.3 and Figure D.4, and we summarize the reasoning patterns below.

- **First layer (induction head):** The first layer performs token-level copying, propagating node information into edge tokens  $\langle \mathbf{e} \rangle$ , consistent with the copy mechanism derived in previous theoretical analysis [169].
- **Second layer and beyond (superposition):** From the second layer onward, the model aggregates over reachable nodes in a superpositional representation that enables parallel breadth-first exploration.

### D.6.3 Accuracy Dynamics in the Answer-Prediction Stage

We track the test accuracy during the final answer-prediction stage following the setting in Figure 5.4. The result is shown in Figure D.5, which shows a rapid transition from

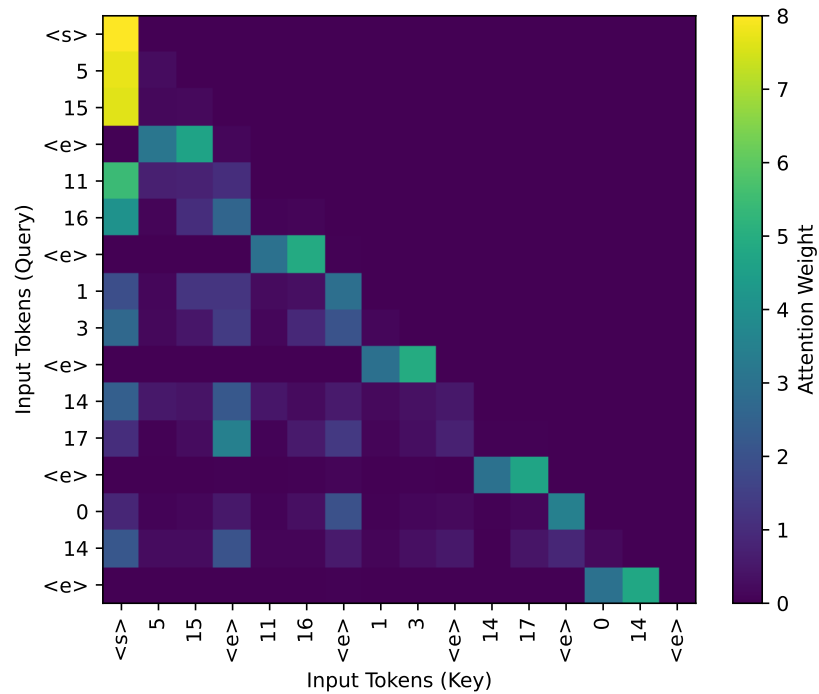


Figure D.3: The first-layer attention patterns in 4-layer transformers. `<e>` tokens attend to the corresponding source and target nodes to aggregate the information. This is consistent with the analysis of the two-layer transformer in [169].

near-random guessing to stable high accuracy once the model integrates residual carryover and candidate lift signals.

## APPENDIX D. APPENDIX FOR TRAINING DYNAMICS OF CONTINUOUS COT183

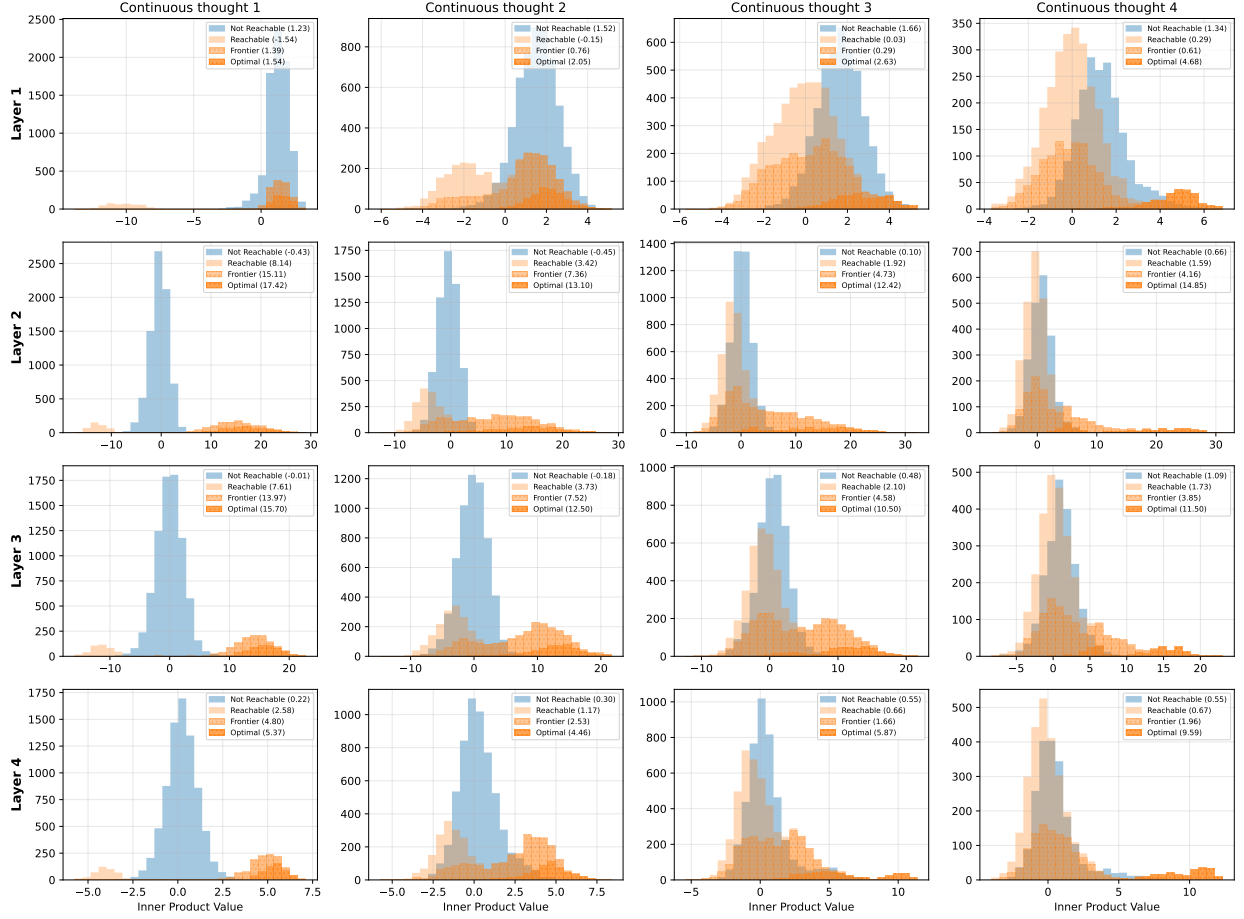


Figure D.4: Inner product between layer-wise hidden states and different types of nodes in a 4-layer transformer. The experimental setting follows [169]. From the second layer onward, hidden states exhibit larger inner products with reachable, frontier, and optimal nodes, indicating that superpositional representations emerge as early as layer 2 in the 4-layer transformer.

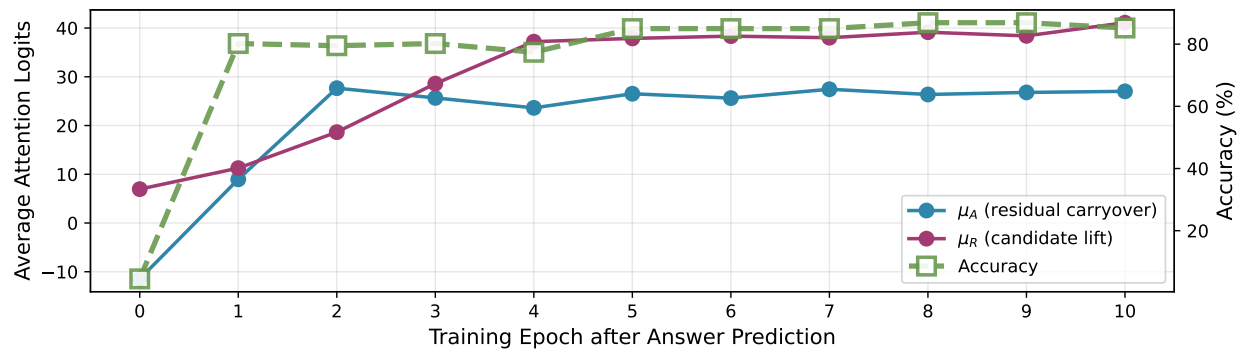


Figure D.5: Accuracy curve during the answer-prediction stage. The accuracy shows a rapid improvement corresponding to the learning of residual carryover and candidate lift signals.