UC Merced
Proceedings of the Annual Meeting of the Cognitive Science Society

Title
Generalization in counting recurrent neural networks arises from emergent number-line representations and stable drift dynamics

Permalink
https://escholarship.org/uc/item/0m18p70g

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 48(0)

Authors
Strock, Anthony
Tan, Ryan
Menon, Vinod

Publication Date
2026

Copyright Information
This work is made available under the terms of a Creative Commons Attribution License, available at https://creativecommons.org/licenses/by/4.0/

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

Generalization in counting recurrent neural networks arises from emergent number-line representations and stable drift dynamics

Anthony Strock (astrock@stanford.edu)[1], **Ryan Tan (tanryan@stanford.edu)**[1,2] **&**
Vinod Menon (menon@stanford.edu)[1,3,4]
[1]Department of Psychiatry & Behavioral Sciences, Stanford University
[2]Symbolic Systems Program, Stanford University
[3]Department of Neurology & Neurological Sciences, Stanford University
[4]Wu Tsai Neurosciences Institute, Stanford University

Abstract

How do learned strategies generalize to new problems? Children who learn to count can add any two numbers by iteratively updating a running total, even for sums they have not encountered. We trained RNNs with a working-memory readout that tracks the progressive count at every timestep and is fed back to maintain a running total, using problems requiring counts up to 5, and tested generalization on counts up to 9. We demonstrate that such RNNs successfully generalize, with continuous drift dynamics and reaction times scaling linearly with count length. Analyzing internal dynamics, we found two emergent number-line representations: one for rapid retrieval of the starting value, one for slow iterative counting, consistent with graded number representations observed in human parietal cortex. Generalization was strongest when network activity remained within linear regimes visited during training, providing a mechanistic account of both successful generalization and its limits.

Keywords: Numerical cognition; Counting; Recurrent neural networks; Length generalization; Representational dynamics

Introduction

A hallmark of human cognition is the ability to generalize learned strategies to new problems. Children who learn to count, for instance, can add any two numbers by iteratively updating a running total, even for sums they have never encountered (Geary & Burlingham-Dubree, 1989; Lemaire & Siegler, 1995; Siegler, 1988). This capacity for algorithmic generalization, extending a learned sequential procedure to new contexts, is central to flexible cognition, yet its computational and neural basis remains poorly understood. In particular, it is unknown what internal representations and dynamics allow a sequential procedure learned from limited experience to generalize to longer sequences than those encountered during training, a challenge known as length generalization (Anil et al., 2022; Lake & Baroni, 2023).

To investigate this, we focused on counting-based addition, one of the first sequential strategies children acquire (Chang et al., 2016; Qin et al., 2014; Rosenberg-Lee et al., 2011). In the counting-from-left strategy, a child begins at the left operand and increments a running total once for each unit of the right operand, stopping at the final sum. This strategy is inherently sequential, requires maintaining and updating an intermediate result in working memory, with harder problems demanding longer counting sequences resulting in longer reaction times (Geary & Burlingham-Dubree, 1989; Lemaire & Siegler, 1995; Shrager & Siegler, 1998; Siegler, 1988).

In the brain, arithmetic recruits a distributed network including the hippocampus, parietal cortex, and prefrontal cortex (Cho et al., 2011; Menon, 2016; Rivera et al., 2005). Graded number representations have been observed at the population level in the human intraparietal sulcus (Harvey & Dumoulin, 2017; Harvey et al., 2013; Nieder, 2016; Nieder & Dehaene, 2009; Piazza et al., 2007), consistent with the numerical distance effect: closer numbers are harder to discriminate (Moyer & Landauer, 1967), suggesting the presence of a mental symbolic number line along which calculation unfold. While computational models have clarified how such number-line representations emerge through learning (Mistry et al., 2023; Strock, Liu, et al., 2025; Thompson et al., 2024; Zorzi & Testolin, 2018), and support addition fact learning (Strock, Mistry, & Menon, 2025), we still lack a mechanistic account of how number-line representations are organized over time during sequential counting, and which dynamics allow a counting procedure to generalize to larger counts than those experienced during training. Related work on generalization in linear recurrent networks suggests that structured dynamics can support generalization (Elman, 1990; Orvieto et al., 2023). However, whether and how these principles apply to length generalization in counting remains unclear.

To address this gap, we trained a recurrent neural network (RNN) with a working-memory readout (Strock et al., 2020) to implement a sequential counting-from-left strategy to solve addition problems. The working-memory component implements a linear output that reads from the network's population activity at every timestep, is supervised to track the progressive count throughout the trial, and is fed back as input so the network's state continuously reflects how far along the count it currently is. Because this feedback loop is active throughout training, the recurrent and readout weights are co-optimized to produce a stable, self-referential counting process during training. We trained networks on addition problems requiring counts up to 5 and tested generalization on counts up to 9.

Our analyses revealed two emergent number-line representations along which network dynamics drifted continuously: one supporting rapid retrieval of the left operand, and one supporting slow, iterative counting of the right operand. Generalization was strongest when network activity remained within linear regimes visited during training, providing a mechanistic account of both successful length generalization and its limits.

In G. Boleda, I. Dautriche, M. Maldonado, C. Saldana and T. Torrent (Eds.), *Proceedings of the 48th Annual Conference of the Cognitive Science Society* ©2026 the author(s). This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY).

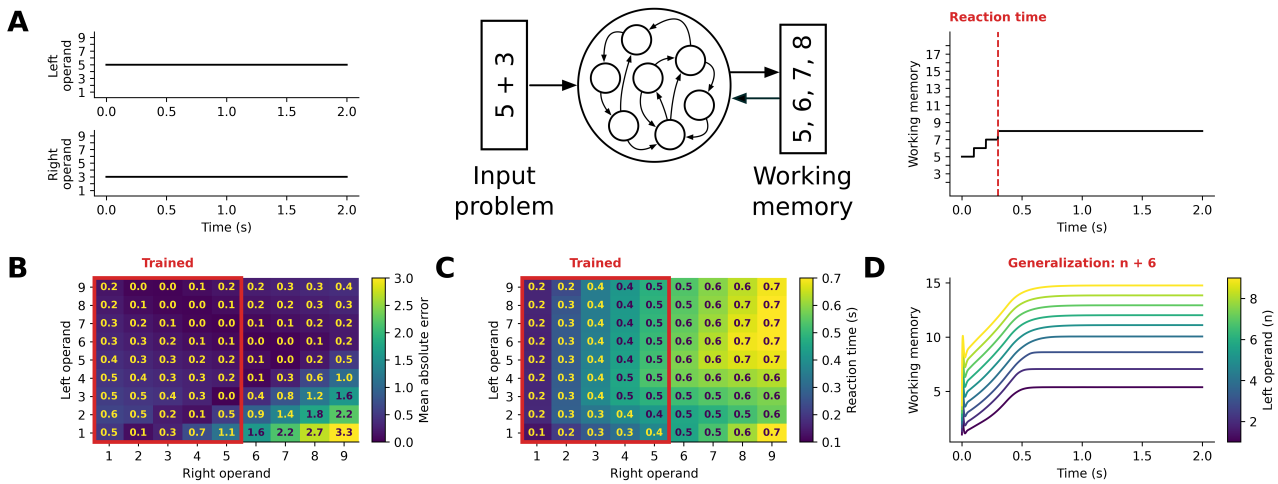

Figure 1: RNNs generalize a counting strategy to solve addition. **A.** Task and RNN model schematic. Operands are provided as constant inputs throughout the trial (left). They are processed by an RNN with feedback (middle), producing a scalar readout that implements counting (right). Reaction time is defined as the first time when working memory stays within 0.5 of its final value for the remainder of the trial. **B-C.** Model's mean absolute error (B) and reaction time (C) when tested on all single-digit additions with left and right operand between 1 and 9. **D.** Example working-memory trajectories for $n + 6$ additions with n between 1 and 9, all unseen during training. Each trajectory shows a smooth, continuous progression from the starting value n toward the sum n + 6, followed by stabilization at the correct answer, demonstrating that the learned drift dynamics extend to longer counts without discrete jumps.

Methods

Counting-from-left addition task

We consider single-digit addition solved using a counting-from-left strategy. On each trial, two operands were presented as constant scalar inputs $u = [L, R]$ throughout a 2-second trial duration, where L is the left operand and R is the right operand. The target output consisted of a time-varying counting sequence that began at the left operand (L) and incremented by one every 100 ms counting step until reaching the final sum ($L + R$), after which the sum was maintained for the remainder of the trial. For training, we consider $L, R \in [\![1, 9]\!] \times [\![1, 5]\!]$, resulting in 45 distinct operations, and for testing $L, R \in [\![1, 9]\!] \times [\![1, 9]\!]$, including 36 that were not seen during training (those with right operand $R \in [\![6, 9]\!]$).

RNN dynamics

We consider an excitatory-inhibitory recurrent neural network (RNN) (Song et al., 2016) of 80 excitatory and 20 inhibitory neurons whose firing-rate activity evolves according to:

$$\tau \frac{dr}{dt} = -r + f(Wr + W_{\text{in}}u + W_{\text{fb}}y + \xi) \quad (1)$$

$$y = W_{\text{out}}r \quad (2)$$

where r represents the firing rate of neurons, u the input problem received, y the working memory where counting occurs, ξ an additive zero-mean Gaussian noise with standard deviation 0.001, $\tau = 100$ ms the neuron time constant, f the ReLU function ($f(x) = \max(0, x)$), W the synaptic strength of connection between neurons, W_{in} the tuning of neurons to input problems, W_{fb} the tuning of neurons to working memory, and W_{out} the readout weights mapping neural activity to the working-memory output. We simulate the dynamics using the explicit Euler method with a timestep of $\Delta t = 1$ ms. At the beginning of each trial r and y are initialized to 0.

RNN iterative training

All analyses reported here are based on a single trained RNN instance. Synaptic weights from excitatory and inhibitory neurons were initialized uniformly in $[0, 1[$ and $]-1, 0]$, respectively, and constrained to preserve their sign throughout training. Here, we treated both u and y as excitatory neurons.

After random initialization, we normalized W, W_{out}, and the concatenated matrix $[W_{\text{in}}, W_{\text{fb}}]$ to each have maximal singular value 1. Then, we rescaled excitatory weights in each row of W so that each postsynaptic neuron received equal total active excitatory and inhibitory incoming weights.

Finally, we trained W, W_{in}, W_{fb}, and W_{out} with Adam using default parameters (e.g., learning rate $\eta = 0.001$), minimizing the mean squared error (MSE) between the produced and desired working-memory outputs across all timesteps. To promote intermediate counting states, we iteratively trained the RNN to produce only the first $k \times 100$ ms of the desired working-memory output, increasing k up to $k_{\max} = 6$, the maximum number of steps required to reach the sum in the considered training range. Each stage of this iterative training, with fixed k, was run for 500 epochs.

Behavioral analysis

To verify that the RNN generalizes and performs counting accurately, we use two metrics: (1) the absolute error at the end of the trial, and (2) the reaction time, defined as the first timestep at which the working-memory output stabilizes. Assuming the final working-memory output is an integer, once the output remains within 0.5 of its final value, rounding to the nearest integer yields a stable discrete value. Therefore we define reaction time as the earliest time t such that $|y(t) - y(T)| < 0.5$ and remains below 0.5 thereafter, where T denotes the final timestep of the trial.

Representational analysis

To understand how the RNN's representations unfold over time to implement a counting process, we performed principal component analysis (PCA) on the firing rate activity of its neurons (r). Specifically, we projected population activity onto the first two principal components (PC1 and PC2), which explained 96.3% of the variance. We then examined how these trajectories evolved over time and identified subspaces coding for the left operand, right operand, and resulting sum.

We identified subspaces coding for the left operand, the right operand, and the resulting sum by finding the best linear decoding of these variables from the projection onto the first two principal components, using activity within the first 100 ms after stimulus onset for the left operand, between 100 ms after stimulus onset and the onset of working-memory stabilization for the right operand, and after the working-memory output stabilized for the resulting sum. For the right operand, we removed the left-operand offset by subtracting for each trial the principal component values at 100 ms.

Linear regimes analysis

Because the RNN uses a ReLU nonlinearity, its dynamics is linear for any fixed set of active neurons, i.e., neurons with nonzero firing-rate activity. This becomes explicit by expressing the ReLU as left multiplication by a time-dependent diagonal gating matrix $D(t)$:

$$\tau \frac{dr}{dt} = -r + D(t)\,(W^* r + W_{\text{in}} u + \xi) \tag{3}$$

where $W^* = W + W_{\text{fb}} W_{\text{out}}$ incorporates the working-memory feedback directly into the recurrent dynamics, and $D(t)$ is a diagonal matrix with entries;

$$D(t)_{ii} = \begin{cases} 1 & \text{if } i \in \mathcal{A}(t), \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where $\mathcal{A}(t) = \left\{ i, (W^* r(t) + W_{\text{in}} u + \xi(t)))_i > 0 \right\}$ denotes the set of active neurons at time t, i.e., neurons with positive pre-activation and therefore nonzero ReLU firing-rate output.

When the same set of neurons is active over an interval, i.e., $\mathcal{A}(t) = \mathcal{A}_0$ and $D(t) = D_0$ in this interval, the dynamics reduce to the linear regime:

$$\tau \frac{dr}{dt} = -r + D_0 W^* r + D_0 W_{\text{in}} u + D_0 \xi \tag{5}$$

We can therefore identify each distinct linear-regime by its set of active neurons $\mathcal{A}$. For visualization, we assign each distinct set an arbitrary integer label, used only as an identifier and with no ordinal meaning. To quantify how similar two linear regimes are, we consider the size of the symmetric difference between their associated sets of active neurons $\mathcal{A}_1$ and $\mathcal{A}_2$, i.e., $|\mathcal{A}_1 \triangle \mathcal{A}_2| = |\mathcal{A}_1 \setminus \mathcal{A}_2| + |\mathcal{A}_2 \setminus \mathcal{A}_1|$, which represents the number of neurons whose active/inactive status differs between the two linear regimes.

Results

RNNs implement continuous counting

Our first objective was to determine whether RNNs can learn a sequential counting strategy. After training, performance on the trained set was accurate across most problems, i.e., the mean absolute error (MAE) at the final timestep remained below 0.5 (Figure 1B), indicating that the network typically produced a near-correct sum by the end of the trial. In addition, the model exhibited a hallmark signature of counting: reaction time increased approximately linearly with the right operand (Figure 1C), consistent with the idea that larger right operands require longer internal counting sequences.

Importantly, contrary to what was expected, this sequential computation was not implemented as a series of discrete, step-like state transitions. Instead, the readout evolved smoothly throughout the trial, producing a continuous drift from the initial value toward the final sum (Figure 1D). Thus, RNNs can learn a counting-like procedure whose behavioral signature resembles serial counting, but is implemented as a smooth, continuous trajectory rather than discrete jumps.

RNNs can generalize to larger counts

Our second objective was to determine whether this sequential strategy generalizes to unseen examples. Under a counting-from-left strategy, the length of the required sequence is fully determined by the right operand. We therefore focused on generalization to larger right operands.

On held-out problems with right operands greater than 5, the mean absolute error at the end of the trial remained below 0.5 for more than half of the operations (Figure 1B), indicating that the network can extend the learned procedure beyond the training range. Difficulty increased with the right operand, consistent with longer counting sequences being harder to sustain. Interestingly, difficulty decreased with the left operand, suggesting that the initial retrieved value also affects performance.

Reaction time also increased approximately linearly for right operands larger than 5 (Figure 1C), consistent with the need to sustain the counting dynamics over longer sequences. Counting remained continuous even for right operands larger than 5, with the readout evolving via the same drift-like dynamics rather than discrete step-like transitions (Figure 1D).

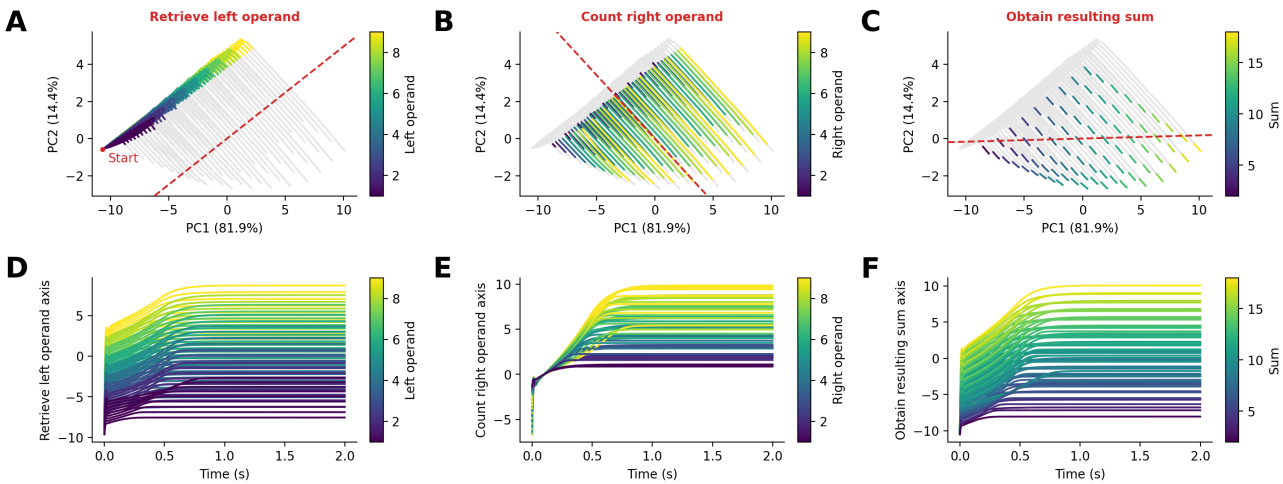

Figure 2: RNNs implement counting as a sequential drift along two number-line representations. **A-C.** Principal component (PC) analysis of population neural activity across all single-digit additions, projected onto the first two PCs (which together explain 96.3% of variance). Trajectories are colored by left operand (A), right operand (B), and resulting sum (C), highlighting three successive phases of computation: rapid retrieval of the left operand within the first 100 ms (A), slow iterative counting of the right operand until the working-memory output stabilizes (B), and settling onto a two-dimensional manifold encoding the final sum (C). **D-F.** Projections of the same trajectories onto the decoded linear subspaces (red axes in A-C) that best decode the left operand (D), right operand (E, after subtracting for each trial the PC values at 100 ms to remove the left-operand offset), and resulting sum (F). Subspace projections correlate with the left operand early in the trial (D, before 100 ms), and with the right operand and resulting sum after counting is complete (E-F, once the output has stabilized), confirming that the two principal components reflect distinct computational roles during initialization and counting.

RNNs count by drifting over a number line

Our third objective was to characterize the internal representation dynamics that support the counting-from-left strategy. To do so, we used principal component analysis (PCA) to visualize RNN firing-rate activity in a two-dimensional state space across the course of a trial. In this low-dimensional projection, trajectories were organized along three stereotyped directions, which coded for the first operand, the second operand, and the sum, respectively.

First, within the first 100 ms after stimulus onset, when working memory had to be initialized to the left operand, RNN activity rapidly moved onto a line along which position coded for the left operand (Figure 2A). This indicates that the RNN retrieved the left operand by rapidly drifting its working-memory state toward this value (Figure 2D).

Then, 100 ms after stimulus onset, RNN activity progressed more slowly along a second line, along which position coded for the right operand after removing the left-operand offset at 100 ms (Figure 2B). This indicates that the RNN counted the right operand iteratively by drifting its working-memory state according to the number of required increments (Figure 2E).

Finally, once working memory stabilized, i.e., remained within 0.5 of its final value, RNN activity occupied a two-dimensional manifold combining these two axes (Figure 2C). Within this manifold, the final count was represented along a diagonal direction (Figure 2C), with position on this axis tracking the evolving sum as counting unfolded (Figure 2F).

Together, this suggests that the counting-from-left strategy is implemented through two continuous drifts along number-line-like manifolds that evolve at different speeds: a fast drift that retrieves the left operand, and a slower drift that implements the iterative counting process.

RNNs generalize counting better when activity remains in trained linear regimes

We observed that the RNN did not generalize equally well across all input problems. Our final objective was therefore to identify which aspects of its dynamics predicted generalization to longer counts. To do this, we focused on the final timestep of each trial, when working memory stabilized, suggesting that the RNN was near a fixed-point attractor. We then characterized the active set of neurons, i.e., the subset of neurons with nonzero firing-rate activity (Figure 3B), which determines the linear regime occupied by the dynamics near this attractor (see Methods).

Intuitively, when the RNN remains within a linear regime, changing the left and right operands should shift its fixed-point attractor in a predictable linear way. To see this, consider the noiseless dynamics within a fixed linear regime:

$$\frac{dr}{dt} = \tilde{W}r + \tilde{W}_{\text{in}}u \tag{6}$$

Consider $u_1 = [L_1, R_1]$ and $u_2 = [L_2, R_2]$, two input problems for which the dynamics remain in this linear regime

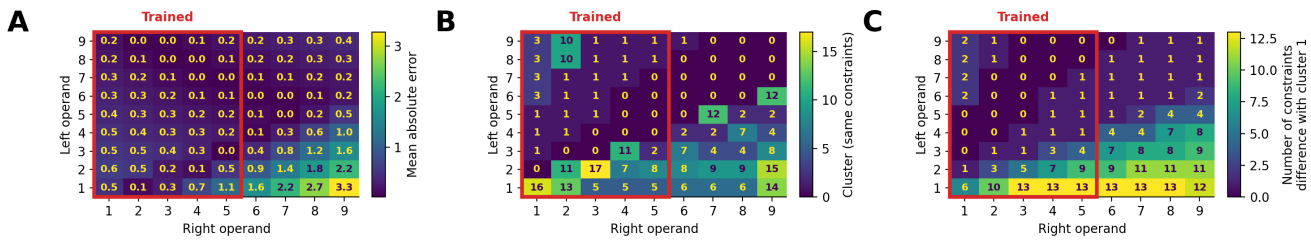

Figure 3: Generalization is strongest when network activity remains within linear regimes visited during training. **A.** Model mean absolute error on all single-digit additions, with left and right operands between 1 and 9. Problems with right operand greater than 5 were not seen during training. **B.** Identity of the active neurons set (positive activation) at the final timestep of each trial. Each integer label corresponds to a distinct configuration of active neurons, defining a unique linear regime. Problems that engage the same active set as frequently visited training problems tend to generalize better. **C.** Distance from each trial's final active set to the most frequently visited active set during training, quantified as the symmetric difference, the number of neurons that differ in active/inactive status between the two configurations. Generalization error increases as this distance grows, indicating that generalization is most reliable when the network's dynamics remain within familiar piecewise-linear regimes learned during training.

and reach unique stable fixed points r_1 and r_2, respectively. Because these fixed points nullify the dynamics, they satisfy:

$$\tilde{W}r_1 + \tilde{W}_{\text{in}}u_1 = 0 \tag{7}$$

$$\tilde{W}r_2 + \tilde{W}_{\text{in}}u_2 = 0. \tag{8}$$

Now consider a new input problem $u_3 = [L_3, R_3]$ that can be expressed as a linear combination of u_1 and u_2:

$$L_3 = \alpha L_1 + \beta L_2 \tag{9}$$

$$R_3 = \alpha R_1 + \beta R_2. \tag{10}$$

Assume that the dynamics also remain in this linear regime for u_3 and reach a unique stable fixed point r_3. This fixed point must be:

$$r_3 = \alpha r_1 + \beta r_2. \tag{11}$$

Indeed, this candidate fixed point nullifies the dynamics and is assumed unique:

$$\tilde{W}\left(\alpha r_1 + \beta r_2\right) + \tilde{W}_{\text{in}}u_3 = \alpha\left(\tilde{W}r_1 + \tilde{W}_{\text{in}}u_1\right) \tag{12}$$

$$+ \beta\left(\tilde{W}r_2 + \tilde{W}_{\text{in}}u_2\right) \tag{13}$$

$$= 0. \tag{14}$$

Thus, if these trained problems produce the correct sums:

$$W_{\text{out}}r_1 = L_1 + R_1 \tag{15}$$

$$W_{\text{out}}r_2 = L_2 + R_2 \tag{16}$$

then the new problem should also produce the correct sum:

$$W_{\text{out}}r_3 = \alpha W_{\text{out}}r_1 + \beta W_{\text{out}}r_2 \tag{17}$$

$$= \alpha(L_1 + R_1) + \beta(L_2 + R_2) \tag{18}$$

$$= L_3 + R_3 \tag{19}$$

Remaining in the same linear regime can therefore, in theory, support correct generalization. In practice, we found that generalization was better when the final active set matched active sets also visited during training (Figure 3A,B), suggesting that successful generalization relies on reusing trained linear regimes rather than entering novel ones. Performance, both during training and generalization, improved when the final active set remained close to the most frequently visited active set during training (Figure 3A,C). This indicates that performance depends not only on staying within trained regimes but also on remaining close to the dominant trained regime.

Discussion

We used RNNs to investigate the neural mechanisms of sequential learning and generalization, and specifically, how children acquire a counting-based addition strategy from limited experience and generalize it to longer, unseen counts. We found that RNNs equipped with a working-memory readout when trained only on problems requiring counts up to 5, RNNs successfully extended the same counting procedure to counts up to 9. Reaction times scaled approximately linearly with count length for both trained and generalized problems, reflecting the iterative nature of the learned procedure.

Role of working memory

A key architectural feature of our modeling framework is the working-memory readout, a linear output that reads from the network's population activity at every timestep and serves two simultaneous roles. First, it is supervised to track the progressive count throughout the trial (e.g., 3→4→5→6→7 for the problem 3+4), providing an explicit sequential training signal at every step rather than only at the final answer. This encourages the network to develop internal dynamics that implement counting as an iterative procedure rather than a direct input-output mapping. Second, the readout is fed back as input to the network at each timestep, so the network's state continuously reflects how far along the count it currently is. Critically, this feedback loop is active throughout training, not only at test time, meaning the recurrent weights, readout weights, and feedback weights are all co-optimized together

in the presence of the running count signal. The result is a stable, self-referential dynamic: the network's next state depends on its current count value, which depends on its current neural state, creating a closed loop that can in principle sustain counting beyond the range of problems seen during training.

Two emergent number lines support counting

A central finding of this work is that training induced two distinct number-line representations in the network's low-dimensional activity space. The first supported rapid retrieval of the left operand: within the first 100 ms of a trial, network activity organized along a direction strongly predictive of the starting value, placing the system at the correct position on the number line before counting began. The second supported slow, iterative counting of the right operand: activity then drifted continuously along a second direction, incrementally tracking the count. Once counting was complete, activity settled onto a two-dimensional manifold from which the sum could be decoded along a diagonal axis combining both representations. These two number lines emerged even though networks were trained on a limited range of operands, and both were engaged during generalization to longer counts. This finding extends neuroscience observations of graded number representations in the human intraparietal sulcus (Harvey et al., 2013; Nieder, 2016) by showing that multiple such representations can coexist within a single network, serving functionally distinct roles during rapid retrieval and slow counting.

Counting is implemented as continuous neural drift

Although the RNN was explicitly trained to produce step-like intermediate counting states, its internal dynamics evolved continuously throughout the trial. This continuous implementation produced the hallmark behavioral signature of serial counting: reaction time scaled approximately linearly with count length, for both trained and generalized problems. This dissociation between step-like behavior and continuous neural trajectories has direct implications for interpreting neuroimaging data from human arithmetic tasks. Our results suggest that discrete cognitive stages, such as individual counting steps, need not correspond to discrete transitions in neural population activity. Testing whether human neural recordings during counting-based arithmetic show similar continuous drift dynamics would directly test this prediction.

Generalization relies on trained linear regimes

RNN generalization was not uniform across held-out problems: performance varied markedly across operations, with better accuracy for shorter counts (smaller right operand) and for larger starting values (larger left operands). The latter pattern is consistent with the idea that stronger initialization signals, produced by larger left operand values through the feedback pathway, leads to more stable placement on the number-line manifold, from which subsequent counting is more reliable. This provides a behavioral prediction: human counting-based addition may similarly show a left-operand advantage for problems requiring generalization to larger counts. Additionally, we found that generalization was strongest when activity remained within the same linear subspaces visited during training, and degraded when dynamics departed from the most frequently visited training subspaces. This provides a mechanistic account of why generalization can be fragile: extending a learned drift beyond the training range is reliable only insofar as the dynamics remain in a familiar local linear regime. This connects our findings to recent theoretical work on length generalization in linear recurrent networks (Orvieto et al., 2023), where structured linear dynamics support compositional generalization.

Limitations and future directions

Several limitations should be noted. First, operands were represented as constant scalar inputs throughout the trial. More realistic representations reflecting psychophysical encoding principles may alter the conditions under which generalization is observed. Second, we focused on a single strategy, counting from the left operand, whereas children flexibly employ a range of strategies depending on the problem and their experience, including counting-on from the larger operand, number decomposition, and fact retrieval (Chang et al., 2016; Cho et al., 2011; Geary & Burlingham-Dubree, 1989; Lemaire & Siegler, 1995; Siegler, 1988, 1991). It is important to distinguish which of our findings reflect properties of counting in general versus this particular procedure. For instance, the model's continuous drift dynamics may be well suited for addition but may not capture the discrete, deterministic character of counting in other contexts, such as the qualitative developmental leap that occurs when children become Cardinal-Principle knowers (Gelman et al., 2009). Third, our model does not separately characterize contributions of evidence encoding, working-memory maintenance, and retrieval to the observed dynamics, dissociating these components in both model and brain data is an important direction for future work. Finally, understanding how strategy selection is implemented, when to count versus retrieve a memorized fact, and how the transition from slow counting to fast retrieval occurs with practice (Cho et al., 2011; Qin et al., 2014; Verguts & Fias, 2005) remain important open questions that the current model does not address.

Conclusion

In conclusion, our findings demonstrate that algorithmic generalization in recurrent neural networks can arise from structured internal representations, specifically, emergent number-line geometries and stable drift dynamics, rather than requiring explicit supervision of the generalization itself. An important next step is to test whether analogous low-dimensional, number-line-like trajectories and differential drift speeds between retrieval and counting phases are present in human neural recordings during sequential arithmetic, bridging the computational account developed here with the neural mechanisms of human mathematical cognition.

Acknowledgements

This work was supported by the National Institutes of Health (R01HD059205, R37HD094623), the National Science Foundation (NSF2024856), the Stanford Institute for Human-Centered Artificial Intelligence, and the Stanford Symbolic Systems Summer Internship Program.

References

Anil, C., Wu, Y., Andreassen, A., Lewkowycz, A., Misra, V., Ramasesh, V., Slone, A., Gur-Ari, G., Dyer, E., & Neyshabur, B. (2022). Exploring length generalization in large language models. *Advances in Neural Information Processing Systems*, *35*, 38546–38556. https://doi.org/10.48550/arXiv.2207.04901

Chang, T.-T., Metcalfe, A. W., Padmanabhan, A., Chen, T., & Menon, V. (2016). Heterogeneous and nonlinear development of human posterior parietal cortex function. *NeuroImage*, *126*, 184–195. https://doi.org/10.1016/j.neuroimage.2015.11.053

Cho, S., Ryali, S., Geary, D. C., & Menon, V. (2011). How does a child solve 7+ 8? decoding brain activity patterns associated with counting and retrieval strategies. *Developmental science*, *14*(5), 989–1001. https://doi.org/10.1111/j.1467-7687.2011.01055.x

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, *14*(2), 179–211. https://doi.org/10.1016/0364-0213(90)90002-E

Geary, D. C., & Burlingham-Dubree, M. (1989). External validation of the strategy choice model for addition. *Journal of Experimental Child Psychology*, *47*(2), 175–192. https://doi.org/10.1016/0022-0965(89)90028-3

Gelman, R., Gallistel, C. R., & Gelman, R. (2009). *The child's understanding of number*. Harvard University Press. https://doi.org/10.4159/9780674037533

Harvey, B. M., & Dumoulin, S. O. (2017). A network of topographic numerosity maps in human association cortex. *Nature Human Behaviour*, *1*(2), 0036. https://doi.org/10.1038/s41562-016-0036

Harvey, B. M., Klein, B. P., Petridou, N., & Dumoulin, S. O. (2013). Topographic representation of numerosity in the human parietal cortex. *Science*, *341*(6150), 1123–1126. https://doi.org/10.1126/science.1239052

Lake, B. M., & Baroni, M. (2023). Human-like systematic generalization through a meta-learning neural network. *Nature*, *623*(7985), 115–121. https://doi.org/10.1038/s41586-023-06668-3

Lemaire, P., & Siegler, R. S. (1995). Four aspects of strategic change: Contributions to children's learning of multiplication. *Journal of experimental psychology: General*, *124*(1), 83. https://doi.org/10.1037/0096-3445.124.1.83

Menon, V. (2016). Memory and cognitive control circuits in mathematical cognition and learning. *Progress in brain research*, *227*, 159–186. https://doi.org/10.1016/bs.pbr.2016.04.026

Mistry, P. K., Strock, A., Liu, R., Young, G., & Menon, V. (2023). Learning-induced reorganization of number neurons and emergence of numerical representations in a biologically inspired neural network. *Nature Communications*, *14*(1), 3843. https://doi.org/10.1038/s41467-023-39548-5

Moyer, R. S., & Landauer, T. K. (1967). Time required for judgements of numerical inequality. *Nature*, *215*(5109), 1519–1520. https://doi.org/10.1038/2151519a0

Nieder, A. (2016). The neuronal code for number. *Nature Reviews Neuroscience*, *17*(6), 366–382. https://doi.org/10.1038/nrn.2016.40

Nieder, A., & Dehaene, S. (2009). Representation of number in the brain. *Annual review of neuroscience*, *32*(1), 185–208. https://doi.org/10.1146/annurev.neuro.051508.135550

Orvieto, A., Smith, S. L., Gu, A., Fernando, A., Gulcehre, C., Pascanu, R., & De, S. (2023). Resurrecting recurrent neural networks for long sequences. *International conference on machine learning*, 26670–26698. https://doi.org/10.5555/3618408.3619518

Piazza, M., Pinel, P., Le Bihan, D., & Dehaene, S. (2007). A magnitude code common to numerosities and number symbols in human intraparietal cortex. *Neuron*, *53*(2), 293–305. https://doi.org/10.1016/j.neuron.2006.11.022

Qin, S., Cho, S., Chen, T., Rosenberg-Lee, M., Geary, D. C., & Menon, V. (2014). Hippocampal-neocortical functional reorganization underlies children's cognitive development. *Nature neuroscience*, *17*(9), 1263–1269. https://doi.org/10.1038/nn.3788

Rivera, S. M., Reiss, A. L., Eckert, M. A., & Menon, V. (2005). Developmental changes in mental arithmetic: Evidence for increased functional specialization in the left inferior parietal cortex. *Cerebral cortex*, *15*(11), 1779–1790. https://doi.org/10.1093/cercor/bhi055

Rosenberg-Lee, M., Barth, M., & Menon, V. (2011). What difference does a year of schooling make?: Maturation of brain response and connectivity between 2nd and 3rd grades during arithmetic problem solving. *Neuroimage*, *57*(3), 796–808. https://doi.org/10.1016/j.neuroimage.2011.05.013

Shrager, J., & Siegler, R. S. (1998). Scads: A model of children's strategy choices and strategy discoveries. *Psychological science*, *9*(5), 405–410. https://doi.org/10.1111/1467-9280.00076

Siegler, R. S. (1988). Strategy choice procedures and the development of multiplication skill. *Journal of experimental psychology: General*, *117*(3), 258. https://doi.org/10.1037/0096-3445.117.3.258

Siegler, R. S. (1991). Strategy choice and strategy discovery. *Learning and Instruction*, *1*(1), 89–102. https://doi.org/10.1016/0959-4752(91)90020-9

Song, H. F., Yang, G. R., & Wang, X.-J. (2016). Training excitatory-inhibitory recurrent neural networks for cognitive tasks: A simple and flexible framework. *PLoS computational biology*, *12*(2), e1004792. https://doi.org/10.1371/journal.pcbi.1004792

Strock, A., Hinaut, X., & Rougier, N. P. (2020). A robust model of gated working memory. *Neural Computation*, *32*(1), 153–181. https://doi.org/10.1162/neco_a_01249

Strock, A., Liu, R., Iyer, R., Mistry, P. K., & Menon, V. (2025). Symbolic numerical generalization through representational alignment. *CogSci... Annual Conference of the Cognitive Science Society. Cognitive Science Society (US). Conference*, *47*, 1882. https://pmc.ncbi.nlm.nih.gov/articles/PMC12445152

Strock, A., Mistry, P. K., & Menon, V. (2025). Personalized deep neural networks reveal mechanisms of math learning disabilities in children. *Science Advances*, *11*(23), eadq9990. https://doi.org/10.1126/sciadv.adq9990

Thompson, J. A., Sheahan, H., Dumbalska, T., Sandbrink, J. D., Piazza, M., & Summerfield, C. (2024). Zero-shot counting with a dual-stream neural network model. *Neuron*, *112*(24), 4147–4158. https://doi.org/10.1016/j.neuron.2024.10.008

Verguts, T., & Fias, W. (2005). Interacting neighbors: A connectionist model of retrieval in single-digit multiplication. *Memory & cognition*, *33*(1), 1–16. https://doi.org/10.3758/BF03195293

Zorzi, M., & Testolin, A. (2018). An emergentist perspective on the origin of number sense. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *373*(1740). https://doi.org/10.1098/rstb.2017.0043