PRETRAINING WITH MASKED BACKSTORIES IN A TOY WORLD

Anonymous authors
Paper under double-blind review

ABSTRACT

Context-enhanced learning (CEL) involves augmenting context in large language models (LLMs) with special masked context to accelerate learning. It has previously only been explored using LLMs with billions of parameters during fine-tuning because CEL requires in-context learning (ICL) abilities to work. Here, we leverage a toy world (symbolically labeled randomly interleaved vector time-series from linear deterministic dynamical systems) that admits LLM-style "next token" pretraining and has been shown to exhibit multiple emergences of different ICL/recall abilities in tiny transformer models with mere millions of parameters.

In this toy world, we can see a late transition from ICL to in-weights learning that also corresponds to a degradation of ICL performance. We enhance pretraining with additional masked context that allows the model to make near perfect predictions on the original training examples. Masking this additional context disincentivizes the model from memorizing it, and the capability of perfect prediction on the training example disincentivizes the model from memorizing the remaining portion of the training example. Not only does this enhancement suppress in-weights learning of the specific training systems, but we also show that it improves the model's performance in the seemingly unrelated task of associative recall.

The study of this enhancement led to a few counterintuitive findings. First, despite such a model during training only seeing losses (and hence gradients) for tokens that are perfectly predictable, it generalizes well at test time when predicting tokens that are not perfectly predictable, nearly matching the performance of the optimal solution for those cases. Second, we found that memorization of 0.025% of the training systems was enough to degrade ICL performance on test data. Finally, when the enhancement is applied only at a specific position in each training example, the model still memorizes the training data, but can only recall these memories at the position where the enhancement was applied and not where the data was exposed to be memorized.

1 INTRODUCTION

The machine learning community is used to the general idea that models get better with scale. We carve out an exception for "overfitting/overtraining" but reality is more complicated. The phenomenon of "inverse scaling" exists (Lin et al., 2022; Michaelov & Bergen, 2023) and is broad, with many instances of specific capabilities getting worse even as many other capabilities get better, including the 11 language tasks where models exhibit this behavior described by McKenzie et al. (2023), the worsening of fine-tunability with more pretraining in Springer et al. (2025), and the inverse scaling with respect to test-time compute found in Gema et al. (2025), among others (Zhou et al., 2024; Xu et al., 2026; Merrill et al., 2026).[1] An example of an inverse scaling task is prompting a model to always complete with the word "heavy" and then giving it "Absence makes the heart grow" — larger models are more likely to disregard the instructions and respond "fonder." This classic example illustrates a general pattern: inverse-scaling examples seem to manifest a tension

[1] Wei et al. (2023); Wu & Lo (2024) add further nuance to the inverse scaling discussion by showing that at even larger scales, there can be "double descent" (Nakkiran et al., 2021) improvement even after some worsening in the inverse-scaling capabilities.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

between what's been learned in the weights (IWL) and what should be learned from the context (ICL).

We believe that this diversity in the dynamics of LLM capability development necessitates a large repertoire of methods for evaluating, controlling, and modulating the progress of language models at a finer-grained, capability-by-capability level.[2] To provide a controlled testbed for the development of these techniques, in this paper, we study an algorithmic toy problem, where small-scale models (millions of parameters) exhibit both IWL and ICL *as well as* the phenomenon of some capabilities getting better while others get worse at some point in training. This toy ecosystem displays a particularly well-studied[3] form of inverse scaling termed *ICL transience* by Singh et al. (2023), where the ICL capability that emerges during training then gives way to the resurgence of IWL. This transience occurs in the toy model at the same time that it is improving in associative recall (Arora et al., 2024).

We investigate a technique for modulating the development of a capability during training inspired by context-enhanced learning (CEL) (Zhu et al., 2025) and gradient starvation (Tachet et al., 2020; Pezeshki et al., 2021). Two novel aspects of our work are: (a) the toy problem exhibits the "some abilities get better while others get worse" phenomenon during training — earlier works studying ICL transience did not have toys that showed both directions simultaneously; and (b) we engage with an intervention of "masked context" which is qualitatively and substantively distinct from the kinds of regularization or data-mix curation interventions studied in earlier toys. We show that the intervention works in blocking the transition to IWL and ends up strengthening ICL, including a seemingly unrelated aspect of associative recall.

The process of exploring the efficacy of this intervention then uncovered a string of novel and puzzling results:

1. We believe our technique works because it removes any loss incentive to perform well on the additional masked context indices, which in turn starves gradients, yet the model still learns to perform near-optimal in-context learning for the initial indices of each segment that never received any loss. (Section 4.2)
2. The current understanding in the literature is that rare, intermittent training examples encourage ICL solutions, while repetitive examples encourage IWL (Chan et al., 2022). When we apply the intervention on a vast majority of the training examples, and only leave a 0.025% of them exposed to be possibly memorized, not only does the model memorize only the exposed fraction, but this memorization of the subset still degrades ICL performance on held-out data. (Section 5)
3. When we apply the intervention only at the beginning of the training example, but in other positions of the example the data is exposed to be possibly memorized, the model memorizes the training data, but cannot recall its memories in the positions where the data was exposed. It can only recall its memories at the beginning of the training example, the very place where the intervention was administered. To our further surprise, the ICL capabilities of the model degrade less at the beginning of the sequence, and degrade more at the exposed positions, although the model can perform IWL better at the beginning and worse at the exposed positions. (Section 6)

2 SETUP

We build directly on Daniels et al. (2025), which was inspired by Garg et al. (2022); Sander et al. (2024), and focus on the orthogonally evolved system family, where the system is defined by $U \in \mathbb{R}^{5\times5}$, a random orthogonal matrix. Each U is generated to ensure a uniform sampling over all 5×5 orthogonal matrices (Mezzadri, 2006). The initial state is $\mathbf{x}_0 \sim \mathcal{N}\left(0, \frac{1}{5}I\right)$, with state updates: $\mathbf{x}_{i+1} = U\mathbf{x}_i = U^{i+1}\mathbf{x}_0$. The system state is in-principle perfectly predictable, but only after six positions are observed:

$$U = \begin{bmatrix}\mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5\end{bmatrix}\begin{bmatrix}\mathbf{x}_0 & \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4\end{bmatrix}^{-1}. \tag{1}$$

[2]Currently, model merging, data curation, context distillation, and post-training techniques are some methods that try to mitigate this problem. See Appendix A for more.

[3]See Table 1 in Appendix A.

2.1 Optimal Pseudoinverse Predictor

Following from (1), given the state observations $\{\mathbf{x}_0, \ldots, \mathbf{x}_i\}$, the MSE-optimal predictor computes $\widehat{\mathbf{x}}_{i+1} = \widehat{U}\mathbf{x}_i$, where

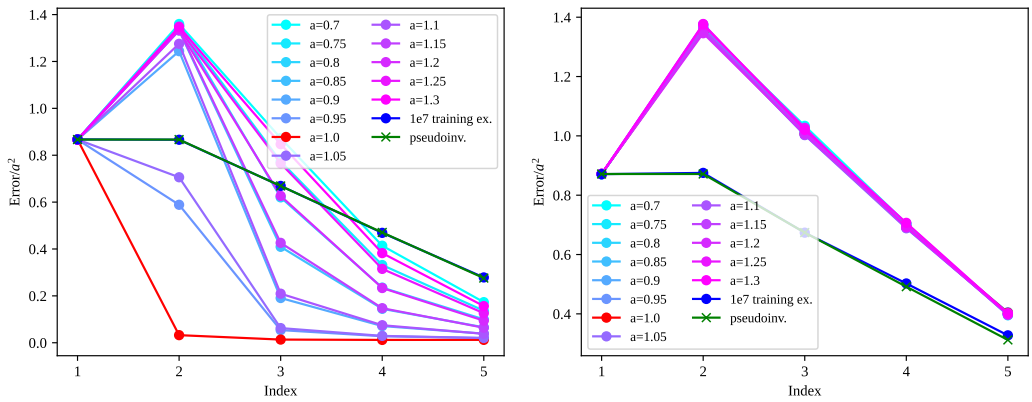
$$\widehat{U} = [\mathbf{x}_1 \quad \ldots \quad \mathbf{x}_i] [\mathbf{x}_0 \quad \ldots \quad \mathbf{x}_{i-1}]^\dagger, \tag{2}$$

and $X^\dagger$ denotes the Moore-Penrose pseudoinverse of X. Essentially, this baseline only makes non-zero errors on the first six entries of any sequence — it gets everything else perfectly correct.

Following Arora et al. (2024), an aspect of associative recall is added to the toy task. To form a training trace, we interleave segments of observation sequences from a library of 40,000 orthogonal systems into a length-251 context window. We follow the same interleaving procedure that is described in Daniels et al. (2025) and illustrated as the "original training example" in Fig. 3.

See Appendix B for details on how the model was evaluated.

3 ICL Transience



(a) Training traces.

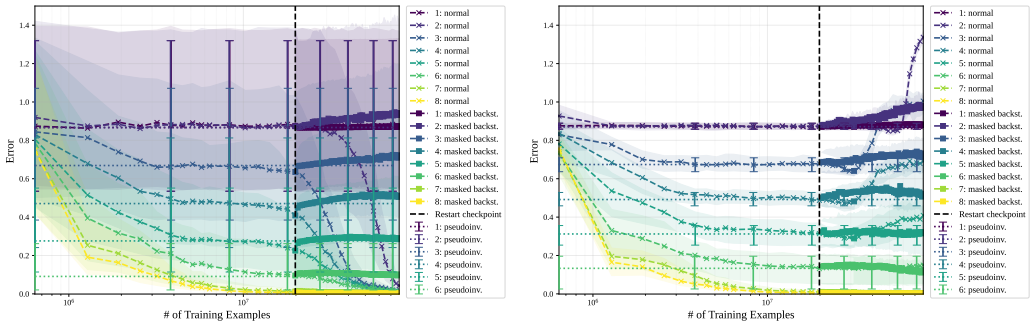
(b) Test traces.

Figure 1: Effect of scaling each initial state by a factor a on the model's median squared error (normalized by a^2) versus the index into the initial segment, evaluated on (a) training traces and (b) test traces. The green-crossed curve corresponds to the pseudoinverse predictor from (2).

ICL transience is a broadly observed phenomenon (see Table 1) where after in-context learning capabilities have already emerged, late in training, neural networks transition away from the in-context learning solution towards memorizing their training data. We see this phenomenon here in the setting of interleaved time-series prediction. Just as in Daniels et al. (2025), whenever any particular system appears in a training example for the first time, it begins with the distinct initial state that happened to be randomly sampled for that system when building the training library. These early indices (from the first to the sixth) are not perfectly predictable by pure ICL alone, since six state observations are required to predict with an MSE of zero as shown in (1).

In Fig. 1a, we plot the squared error of the model at the checkpoints corresponding to $\approx 2 \times 10^7$ (blue) and $\approx 8.6 \times 10^7$ training examples seen so far during training versus the index into the initial segment when tested on its training data. The green crossed pseudoinverse predictor curve corresponding to (2) shows the lowest squared error per index achievable without memorization. We see that the model's squared error after $\approx 2 \times 10^7$ training examples sits near the pseudoinverse predictor's error, while after $\approx 8.6 \times 10^7$ training examples the model's squared error is lower than the optimal pseudoinverse predictor's: a clear sign of training data memorization. Fig. 1b shows the same curves but when the model is tested on held-out data and we see that the transition to IWL has hurt ICL performance.

In Appendix D, Fig. 10 shows that the model's memorization is heavily dependent on observing the specific initial state that maps to each training system. For this reason, in Fig. 1a, we study how



(a) Training traces.

(b) Test traces.

Figure 2: The median squared error with quartile error bars of the normally trained model (crossed curves), the masked backstory trained model (square-marker curves), and the pseudoinverse predictor (dotted lines). The horizontal axes measure training progress in terms of examples seen. (Multiply by 250 to get tokens.)

robustly the model can recall the training data, by scaling each initial state by different values of $a \in [0.7, 1.3]$ and displaying the squared error divided by a^2. We see that the model is sensitive to this norm scaling, with normalized squared error increasing as the norm changes. When $a \in [0.95, 1.05]$, the error degrades but stays below pseudoinverse predictor error for each index. Otherwise the error degrades above pseudoinverse predictor levels for the second index, but then recovers below pseudoinverse levels for the subsequent indices.

Fig. 2 shows the same ICL transience story through looking at the entire training trajectory, where in Fig. 2a the error on training data (crossed curves) drops below pseudoinverse predictor levels after 1×10^7 training examples have been seen, and for held-out test data shown in Fig. 2b the degradation of ICL performance occurs after 3×10^7 training examples are seen by the model.

4 Training with Masked Backstories Suppresses ICL Transience

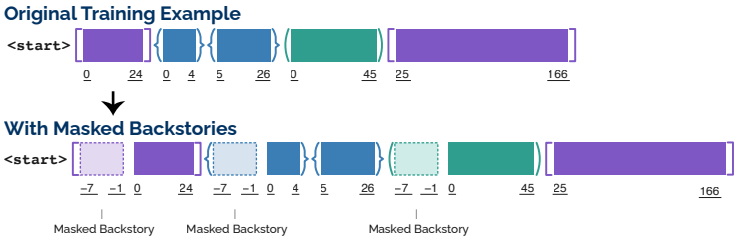


Figure 3: Training with masked backstories consists of reversing the evolution of a linear system when it appears in the training example for the first time, prepending these state observations to the corresponding segment, and masking any training loss on these backstory indices.

To suppress this in-weights learning (in the style of Lampinen et al. (2025); Park et al. (2025b)), if a system U appears for the first time in a training example, we evolve the system in reverse for seven time steps by successively applying U^T to the initial state $\mathbf{x}_0$ to obtain $\{\mathbf{x}_{-7}, \ldots, \mathbf{x}_{-1}\}$. We call these 7 state observations a "backstory". We then prepend this backstory to the system segment as seen in Fig. 3. Finally, when the training loss is computed, we mask any losses incurred on the backstory indices. This masking is qualitatively identical to the CEL approach in Zhu et al. (2025), and so is in the spirit of leaning on the model's ICL ability to starve the gradients that would improve the model's IWL capabilities (Tachet et al., 2020; Pezeshki et al., 2021).

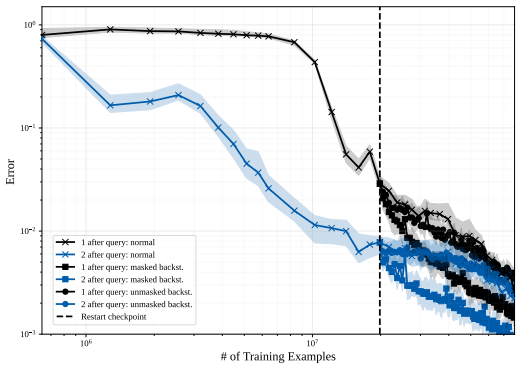


Figure 4: The median squared error with quartile error bars of the normally trained model (dotted curves), the masked backstory trained model (square curves), and the unmasked backstory trained model (circle curves) on indices one and two into the test segment of held-out "needle-in-a-haystack" traces. See that the masked backstory trained model's squared error is always the lowest for each index.

4.1 Restarting Training with Masked Backstories

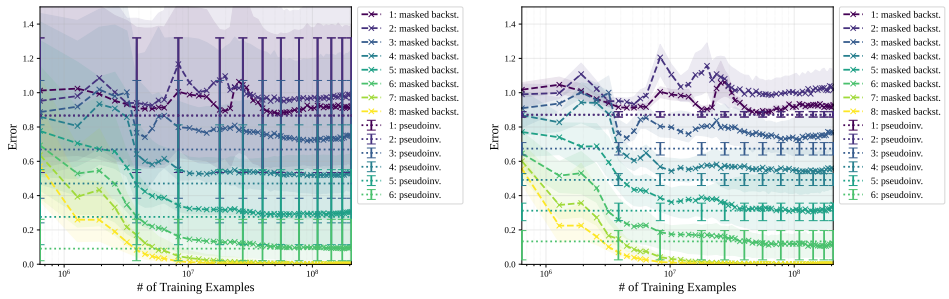
Our main experiment is to choose a checkpoint before the generalization performance has deteriorated and continue training with masked backstories from that point. The checkpoint we chose was when the model had seen 2×10^7 training examples, and we call this checkpoint the "restart checkpoint." It is denoted by a vertical dashed black line in Figs. 2 and 4.

The thick square-marker curves in Fig. 2a show that on the training data, the intervention by masked backstories completely blocks and even reverses the memorization of the training data. For the most part, the curves are near the level of the optimal pseudoinverse predictor, with some additional degradation visible for the earlier time steps. The transition to IWL appears blocked, and indeed Fig. 2b shows that on held-out test traces, the degradation in ICL performance seen in the crossed curves is largely not present in the square-marker curves with the masked-backstory intervention. Instead, we only see the same slight degradation in the purple and blue curves corresponding to earlier time steps that we see in the training data. There is no discrepancy between training and test performance. Fig. 4 shows a further benefit from the intervention. The black and blue curves associated with associative recall performance continue to improve, and actually improve even faster with the masked backstories than with either normal training or a control trained with *unmasked* backstories (Appendix E.1).

4.2 Backstories from the Start: Near-optimal predictions without specific gradients

Section 4.1 showed that when restarting training with masked backstories from a checkpoint taken before generalization performance deteriorates, the model's transition to in-weights learning is blocked, and the model's in-context associative recall abilities are improved. Now, we ask the question: what happens when we train a model with masked backstories from the very start? A detailed look at Fig. 2b from the earlier restart checkpoint experiment shows us that there is some degradation in the prediction of the second index, as this index is no longer receiving any gradient pressure to keep its prediction error low. This makes us hypothesize that a model trained from the beginning with masked backstories will not learn to do in-context learning for the initial indices, as it never receives any loss on them.

First, we check if the model trained using masked backstories from the very beginning exhibits a transition to in-weights learning like the normally trained model. We check this by testing this model on the training traces with prepended backstories, since it is the prepended backstories now that have irreducible loss for this model. In Fig. 5a, we plot the median squared error of the pseudoinverse predictor (dotted lines) and the transformer model (dashed curves with cross markers) for the first through the eighth index into the backstoried training traces. In this figure, we see that through $2 \times$



(a) Tested on backstoried training data.

(b) Tested on held-out data.

Figure 5: Median squared error of the masked backstory trained model (crossed curves) and the pseudoinverse predictor (dotted lines) tested on backstoried training data and held-out traces vs the number of training examples seen so far during training. There is no degradation in the model's generalization performance.

10^8 training examples seen during training the transformer model's median squared error converges down almost to the pseudoinverse predictor's median squared error without any sudden drop down to zero as we saw in Fig. 2a with the normally trained model at around 2×10^7 training examples. Additionally, we also test the model on held-out "needle-in-a-haystack" traces and Fig. 5b shows the corresponding plot for generalization performance. Notice that the crossed curves approach the pseudoinverse predictor error, remain horizontal, and do not sharply increase through 2×10^8 training examples seen unlike the crossed curves in Fig. 2b at 3×10^7 training examples seen for the normally trained model. These two results together show us that the model trained with masked backstories from the beginning does not transition to in-weights learning for the early indices up to this point in training.

> Since this model was trained with masked backstories from the beginning, it was never penalized during training for poor predictions on the first 7 indices of an initial segment. Nonetheless, it still developed the ability to make near-optimal predictions for these indices.

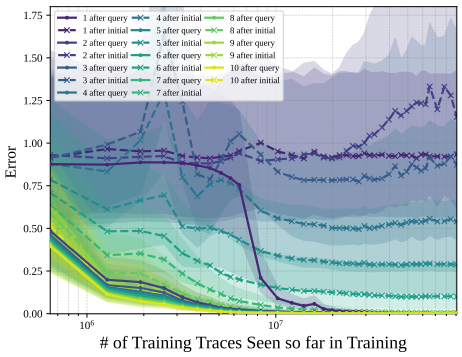
The model's median prediction error approaching the pseudoinverse predictor's median prediction error in Fig. 5b provides the evidence for this. Furthermore, one could think of the task of predicting the next vector in the segment when fewer than six observations have been seen as a more difficult task than predicting the next vector when there are enough observations to know the underlying system perfectly. This is because when there are not enough observations, the model must learn how to deal with the extra degrees of freedom caused by the underdetermined system of equations. Therefore, in this toy setting, the model is exhibiting a form of "easy-to-hard" generalization (Hase et al., 2024; Sun et al., 2024; Schwarzschild et al., 2021; Song et al., 2025). Somehow, its ability to predict the next vector perfectly when there are enough observations also allows it to reasonably predict the next vector when it faces uncertainty.

5 Exposing Just Ten Training Systems Degrades Performance

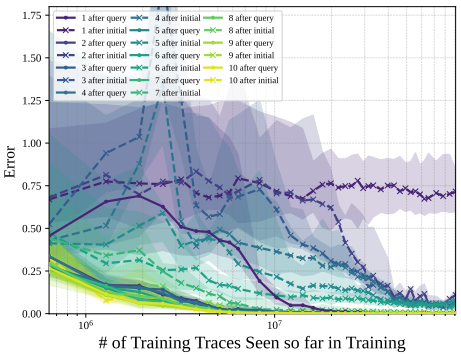
For all of the previous experiments, when applying masked backstories, they were applied to every training system. Now we can ask the question, if the masked backstories are only applied to a fraction of the training systems, can the ICL ability of the model be maintained while allowing for the memorization of a select group of systems? According to Chan et al. (2022), rare, intermittent, and bursty training examples encourage models to learn ICL. Because of this, our hypothesis for this experiment was that the model would memorize the ten exposed training systems, but since these systems appear so rarely during training, the model's ICL abilities would remain intact.

Here, we study what happens when only 10 training systems are unmasked and 39,990 are masked during training. We provide evaluations on held-out systems (Fig. 6c), the masked training systems (Fig. 6a), and the unmasked training systems (Fig. 6b). We find that the unmasked systems get

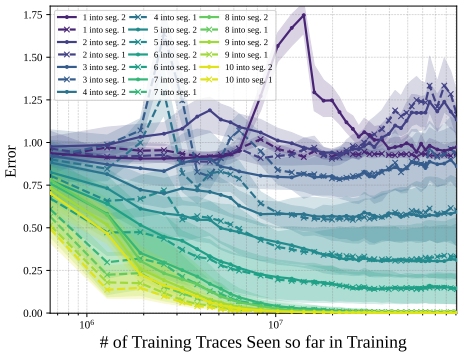
memorized after $\approx 2 \times 10^7$ training examples, the same point in training where memorization occurs in the normally trained model. In Fig. 6a, the masked systems are not memorized, but in Fig. 6c, the median squared error of the predictions in the initial haystack segment for held-out test data still appears to be slowly rising. This is especially visible for the third index, which hovers at a squared error of around 0.9 while the model that was trained with masked backstories from the beginning had an error below 0.8 for this index. This means that even when the unmasked subset takes up 0.025% of training systems, the memorization of these systems adversely affects the model's ability to perform ICL.



(a) Train, masked subset.



(b) Train, unmasked subset.



(c) Held-out test data.

Figure 6: The training dynamics of the model with 10 of the training systems without masked backstories when predicting needle-in-a-haystack style data. Each sub-caption shows which evaluation dataset was used. Fig. 6c shows that the model's ICL ability is still moderately affected by this memorization.

To underscore why this result surprising, we can calculate a rough estimate that an unmasked system will only appear in around two out of every three batches during training. Since the maximum number of distinct systems in a single trace is sampled from the $\text{Zipf}(1.5, 25)$ distribution (Daniels et al., 2025), which has a mean of $\approx$ 3.9, the average number of distinct systems in a trace is upper bounded by four. Then, knowing that the model was trained with a batch size of 640, we can calculate an approximate upper bound on the average appearance rate of an unmasked system in a batch as,

$$\text{Avg. appearance rate} \lesssim \frac{(4 \text{ total systems per trace})(640 \text{ traces per batch})(10 \text{ unmasked systems})}{40,000 \text{ total systems}} \tag{3}$$

$$= \frac{256}{400} = 0.64 \text{ unmasked systems per batch.} \tag{4}$$

This shows that an unmasked system appears in about two out of every three batches.

Even when less than 0.64 unmasked systems appear in each batch on average, these unmasked systems will be memorized by the model and will consequently degrade its generalization performance on held-out systems.

This same behavior also occurs to a greater degree when 10,000 of the training systems are left unmasked and the other 30,000 are masked corresponding to a higher average appearance rate. These results are given in Appendix G.1.

6 Backstories only for the Initial Segment

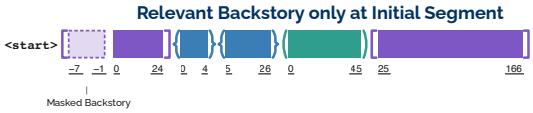
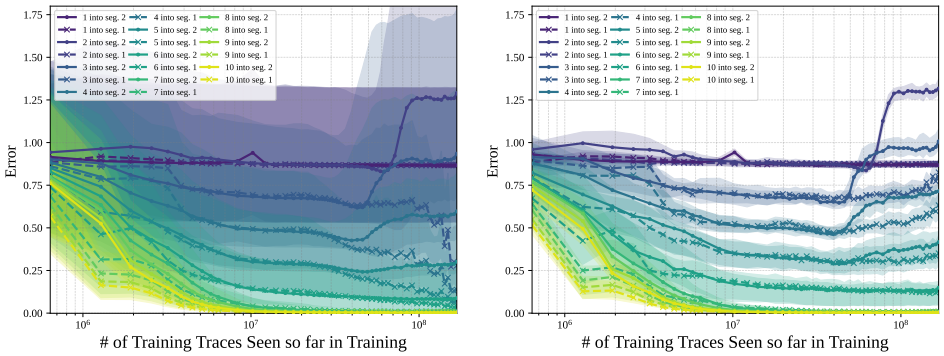


Figure 7: Masked backstories are prepended only to the initial segment of each training example.



(a) Train, no backstory.

(b) Test, no backstory.

Figure 8: Training dynamics for the model trained with masked backstories only for the initial segment. Each sub-caption shows which evaluation dataset was used. Notice that Fig. 8a shows memorization of the unbackstoried training data only at the initial segment. This memorization occurs after $\approx 10^8$ training examples for this experiment while it occurs earlier at $\approx 2 \times 10^7$ training examples for the normally trained model from Fig. 2a.

When the model is trained without any interventions, Fig. 18 in Appendix F.1 shows that the model only memorizes the beginning indices of the initial haystack segment, and gets worse at predicting the beginning indices of subsequent haystack segments. So a natural question is, if masked backstories are only inserted before the initial segment during training (see Fig. 7), will the memorization be suppressed? Fig. 8 shows that the answer is no. Training with masked backstories only at the initial segment leads to degraded ICL abilities (Fig. 8b).

Surprisingly, Fig. 8a shows that the model only memorizes indices in the initial segment, which the model had no gradient pressure to memorize during training, while losing performance on training data at indices in subsequent haystack segments that did have gradient pressure during training.

This counterintuitive result shows that providing supervision at one portion of a training example can lead to transferable abilities in other portions of the example and degraded abilities at the portion that received the supervision itself. By analogy, Cai et al. (2025) saw that length generalization can improve on one task when a model is trained on longer inputs from another similar task, but that work does not observe the puzzling phenomenon of the model simultaneously performing worse on long inputs for the task that was supervised with long inputs.

A closer look at Fig. 8b shows that the model's ICL capabilities degrade more on the subsequent haystack segments where the model cannot recall its memories and perform IWL than in the initial indices where the model can clearly perform IWL when prompted with training data. This last observation motivates a refined understanding of how the IWL and ICL capabilities interact with each other.

7 Discussion

Training models to perform next-token prediction on interleaved time-series generated from linear dynamical systems provides a controlled setting for studying training dynamics. The combination of associative recall (from interleaving) and least-squares-style ICL (from the system dynamics) allowed the toy to exhibit multiple capabilities, in which ICL transience hurt one of them while associative recall continued to improve during training. This simple data model also facilitated the conception of training with masked backstories, and devising experiments to validate this technique. Practically, since Qi et al. (2025) showed that it is possible to automatically generate additional context for language data using another LLM (in their case GPT-4), this opens the opportunity for larger-scale experiments for language model training with masked backstories.

The observation that models trained with masked backstories throughout the entirety of training (Section 4.2) are still able to make reasonable predictions for indices that were always masked opens up questions about the inductive biases of the transformer architecture. There are infinitely many possible predictors for this problem that would achieve zero MSE once six observations are seen, but would obtain worse MSE than the pseudoinverse predictor when predicting the first six indices. For example, randomly projecting each observation vector, performing the pseudoinverse prediction in the new subspace, and then projecting the predicted vector back to the original space. An interesting theoretical question is, why does the transformer architecture have the proclivity to find something close to the min-norm solution in original coordinates when the training objective does not require this and weight matrices are initialized randomly? Empirically, are there other tasks or scenarios where a model can incidentally learn a capability without any loss fueling its learning?

Furthermore, it is unclear exactly why the additional masked context leads to improved in-context associative recall and in-context learning on later indices in initial segments. This phenomenon closely aligns with the claims of Singh et al. (2025) where ICL and "context-constrained" IWL compete in the first attention layer of a two-layer transformer, but share circuitry in the second layer. Suppressing IWL with masked backstories could then be allowing the ICL circuit to develop more strongly in the early layers of our model without the interference of IWL circuit development. Still, our results show that the IWL and ICL capabilities interact with each other in a nontrivial way, as just memorizing 0.025% of the training systems degrades the ICL capability.

Our last finding, that memories can only be recalled in a specific position, even when the exposed data was never provided in that position, opens a question of whether this type of capability transfer at preferential positions is a broad phenomenon. We report these results with the hope that their resolution can lead to deeper insights within the field.

AI use statement

LLMs aided in finding relevant literature, proofreading the document for typos, and debugging code.

References

Suraj Anand, Michael Lepori, Jack Merullo, and Ellie Pavlick. Dual process learning: Controlling use of in-context vs. in-weights strategies with weight forgetting. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 50280–50306, 2025.

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Y Zou, Atri Rudra, and Christopher Re. Zoology: Measuring and improving recall in efficient language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 15664–15730, 2024.

Timothy J. Boerner, Stephen Deems, Thomas R. Furlani, Shelley L. Knuth, and John Towns. ACCESS: Advancing Innovation: NSF's Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support. In *Practice and Experience in Advanced Research Computing (PEARC '23)*, pp. 4, New York, NY, USA, July 2023. ACM. doi: 10.1145/3569951.3597559. URL https://doi.org/10.1145/3569951.3597559.

Ziyang Cai, Nayoung Lee, Avi Schwarzschild, Samet Oymak, and Dimitris Papailiopoulos. Extrapolation by Association: Length Generalization Transfer In Transformers. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 71042–71070. Curran Associates, Inc., 2025. doi: 10.52202/085713-2389.

Bryan Chan, Xinyi Chen, Andras Gyorgy, and Dale Schuurmans. Toward understanding in-context vs. in-weight learning. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 69836–69870, 2025.

Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, James McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. *Advances in neural information processing systems*, 35:18878–18891, 2022.

Yihong Chen, Kelly Marchisio, Roberta Raileanu, David Adelani, Pontus Lars Erik Saito Stenetorp, Sebastian Riedel, and Mikel Artetxe. Improving language plasticity via pretraining with active forgetting. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 31543–31557. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6450ea28ebbc8437bc38775157818172-Paper-Conference.pdf.

Sultan Daniels, Dylan Davis, Dhruv Gautam, Wentinn Liao, Gireeja Ranade, and Anant Sahai. Decomposing prediction mechanisms for in-context recall, 2025. URL https://arxiv.org/abs/2507.01414.

Jasper Dekoninck, Marc Fischer, Luca Beurer-Kellner, and Martin Vechev. Controlled text generation via language model arithmetic, 2024. URL https://arxiv.org/abs/2311.14479.

Zhe Du, Haldun Balim, Samet Oymak, and Necmiye Ozay. Can transformers learn optimal filtering for unknown systems? *IEEE Control Systems Letters*, 7:3525–3530, 2023. doi: 10.1109/LCSYS.2023.3335318.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

Aryo Pradipta Gema, Alexander Hägele, Runjin Chen, Andy Arditi, Jacob Goldman-Wetzler, Kit Fraser-Taliente, Henry Sleight, Linda Petrini, Julian Michael, Beatrice Alex, Pasquale Minervini, Yanda Chen, Joe Benton, and Ethan Perez. Inverse Scaling in Test-Time Compute. *Transactions on Machine Learning Research*, July 2025. ISSN 2835-8856.

Peter Hase, Mohit Bansal, Peter Clark, and Sarah Wiegreffe. The unreasonable effectiveness of easy training data for hard tasks, 2024. URL https://arxiv.org/abs/2401.06751.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation, 2026. URL https://arxiv.org/abs/2601.20802.

Alexander Y. Ku, Thomas L. Griffiths, and Stephanie C. Y. Chan. Predictability shapes adaptation: An evolutionary perspective on modes of learning in transformers, 2025. URL https://arxiv.org/abs/2505.09855.

Andrew K Lampinen, Arslan Chaudhry, Stephanie CY Chan, Cody Wild, Diane Wan, Alex Ku, Jörg Bornschein, Razvan Pascanu, Murray Shanahan, and James L McClelland. On the generalization of language models from in-context learning and finetuning: a controlled study. *arXiv preprint arXiv:2505.00661*, 2025.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 3214–3252, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Shalini Maiti, Amar Budhiraja, Bhavul Gauri, Gaurav Chaurasia, Anton Protopopov, Alexis Audran-Reiss, Michael Slater, Despoina Magka, Tatiana Shavrina, Roberta Raileanu, and Yoram Bachrach. Souper-model: How simple arithmetic unlocks state-of-the-art llm performance, 2025. URL https://arxiv.org/abs/2511.13254.

Mitch Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

Ian R. McKenzie, Alexander Lyzhov, Michael Martin Pieler, Alicia Parrish, Aaron Mueller, Ameya Prabhu, Euan McLean, Xudong Shen, Joe Cavanagh, Andrew George Gritsevskiy, Derik Kauffman, Aaron T. Kirtland, Zhengping Zhou, Yuhui Zhang, Sicong Huang, Daniel Wurgaft, Max Weiss, Alexis Ross, Gabriel Recchia, Alisa Liu, Jiacheng Liu, Tom Tseng, Tomasz Korbak, Najoung Kim, Samuel R. Bowman, and Ethan Perez. Inverse Scaling: When Bigger Isn't Better. *Transactions on Machine Learning Research*, June 2023. ISSN 2835-8856.

Nick Merrill, Jaeho Lee, and Ezra Karger. Is Capability a Liability? More Capable Language Models Make Worse Forecasts When It Matters Most, May 2026.

Francesco Mezzadri. How to generate random matrices from the classical compact groups. *arXiv preprint math-ph/0609050*, 2006.

James Michaelov and Ben Bergen. Emergent inabilities? inverse scaling over the course of pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14607–14615. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.973. URL http://dx.doi.org/10.18653/v1/2023.findings-emnlp.973.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

Alex Nguyen and Gautam Reddy. Differential learning kinetics govern the transition from memorization to generalization during in-context learning, 2024. URL https://arxiv.org/abs/2412.00104.

Core Francisco Park, Ekdeep Singh Lubana, and Hidenori Tanaka. Competition dynamics shape algorithmic phases of in-context learning. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 66381–66433, 2025a.

Core Francisco Park, Zechen Zhang, and Hidenori Tanaka. New news: System-2 fine-tuning for robust integration of new knowledge. *arXiv preprint arXiv:2505.01812*, 2025b.

Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks, 2021. URL https://arxiv.org/abs/2011.09468.

Siyuan Qi, Bangcheng Yang, Kailin Jiang, Xiaobo Wang, Jiaqi Li, Yifan Zhong, Yaodong Yang, and Zilong Zheng. In-context editing: Learning knowledge from self-induced distributions, 2025. URL https://arxiv.org/abs/2406.11194.

Neil Rathi and Alec Radford. Shaping capabilities with token-level data filtering, 2026. URL https://arxiv.org/abs/2601.21571.

Yeonju Ro, Cong Xu, Agnieszka Ciborowska, Suparna Bhattacharya, Frankie Li, and Martin Foltin. Dataset efficient training with model ensembling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 4700–4704, June 2023.

Michael Eli Sander, Raja Giryes, Taiji Suzuki, Mathieu Blondel, and Gabriel Peyré. How do transformers perform in-context autoregressive learning ? In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 43235–43254. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/sander24a.html.

Avi Schwarzschild, Eitan Borgnia, Arjun Gupta, Furong Huang, Uzi Vishkin, Micah Goldblum, and Tom Goldstein. Can you learn an algorithm? generalizing from easy to hard problems with recurrent networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 6695–6706. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/3501672ebc68a5524629080e3ef60aef-Paper.pdf.

Thibault Sellam, Steve Yadlowsky, Jason Wei, Naomi Saphra, Alexander D'Amour, Tal Linzen, Jasmijn Bastings, Iulia Turc, Jacob Eisenstein, Dipanjan Das, Ian Tenney, and Ellie Pavlick. The multiberts: Bert reproductions for robustness analysis, 2022. URL https://arxiv.org/abs/2106.16163.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning, 2026. URL https://arxiv.org/abs/2601.19897.

Aaditya Singh, Stephanie Chan, Ted Moskovitz, Erin Grant, Andrew Saxe, and Felix Hill. The transient nature of emergent in-context learning in transformers. *Advances in Neural Information Processing Systems*, 36, 2023.

Aaditya K. Singh, Ted Moskovitz, Sara Dragutinovic, Felix Hill, Stephanie C. Y. Chan, and Andrew M. Saxe. Strategy coopetition explains the emergence and transience of in-context learning, 2025. URL https://arxiv.org/abs/2503.05631.

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context, 2022. URL https://arxiv.org/abs/2209.15189.

Jiajun Song, Zhuoyan Xu, and Yiqiao Zhong. Out-of-distribution generalization via composition: A lens through induction heads in transformers. *Proceedings of the National Academy of Sciences*, 122(6):e2417182122, 2025. doi: 10.1073/pnas.2417182122. URL https://www.pnas.org/doi/abs/10.1073/pnas.2417182122.

Jacob Mitchell Springer, Sachin Goyal, Kaiyue Wen, Tanishq Kumar, Xiang Yue, Sadhika Malladi, Graham Neubig, and Aditi Raghunathan. Overtrained language models are harder to fine-tune, 2025. URL https://arxiv.org/abs/2503.19206.

Zhiqing Sun, Longhui Yu, Yikang Shen, Weiyang Liu, Yiming Yang, Sean Welleck, and Chuang Gan. Easy-to-hard generalization: Scalable alignment beyond human supervision. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 51118–51168. Curran Associates, Inc., 2024. doi: 10.52202/079017-1618. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/5b6346a05a537d4cdb2f50323452a9fe-Paper-Conference.pdf.

Marton Szep, Jorge Marin Ruiz, Georgios Kaissis, Paulina Seidl, Rüdiger von Eisenhart-Rothe, Florian Hinterwimmer, and Daniel Rueckert. Unintended memorization of sensitive information in fine-tuned language models, 2026. URL https://arxiv.org/abs/2601.17480.

Remi Tachet, Mohammad Pezeshki, Samira Shabanian, Aaron Courville, and Yoshua Bengio. On the learning dynamics of deep neural networks, 2020. URL https://arxiv.org/abs/1809.06848.

Jason Wei, Najoung Kim, Yi Tay, and Quoc Le. Inverse scaling can become U-shaped. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 15580–15591, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.963. URL https://aclanthology.org/2023.emnlp-main.963/.

Tung-Yu Wu and Pei-Yu Lo. U-shaped and inverted-u scaling behind emergent abilities of large language models. *ArXiv*, abs/2410.01692, 2024. URL https://api.semanticscholar.org/CorpusID:273025558.

Alexander Xiong, Xuandong Zhao, Aneesh Pappu, and Dawn Song. The landscape of memorization in llms: Mechanisms, measurement, and mitigation, 2025. URL https://arxiv.org/abs/2507.05578.

Manjie Xu, Isabella Yin, Xinyi Tu, Chi Zhang, and Yixin Zhu. Code over Words: Overcoming Semantic Inertia via Code-Grounded Reasoning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 16610–16632, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.819.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. Model merging in llms, mllms, and beyond: Methods, theories, applications, and opportunities. *ACM Computing Surveys*, 2024.

Lexin Zhou, Wout Schellaert, Fernando Martínez-Plumed, Yael Moros-Daval, Cèsar Ferri, and José Hernández-Orallo. Larger and more instructable language models become less reliable. *Nature*, 634(8032):61–68, October 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07930-y.

Xingyu Zhu, Abhishek Panigrahi, and Sanjeev Arora. On the Power of Context-Enhanced Learning in LLMs. In *Forty-second International Conference on Machine Learning*, 2025.

APPENDIX CONTENTS

A Related Work **15**

B Needle-in-a-haystack Testing **16**

C Model Details **17**

D Shifting the Initial Trace Index **17**

E More Figures for Restarting Training with Masked Backstories **17**

E.1 Unmasked Backstories: Teasing Apart the Effect of Masking vs. Segment Lengths 19

F Training with Masked Backstories from the Beginning **20**

F.1 Performance at Another Haystack Segment 22

F.2 Further Interventions 23

F.2.1 Backstories only for the Initial Segment 24

F.2.2 Backstories of Length One 25

F.2.3 Backstories of Length Two 27

F.2.4 Unmasked Backstories 28

F.2.5 Noise Backstories 29

G Masking only a Fraction of the Systems **29**

G.1 Masking Three-fourths of the Training Systems 30

G.2 Masking All but Ten Training Systems 31

H Multiple Initial State Vectors Corresponding to a Single Training System **32**

A Related Work

Memorization in LLMs It has been observed that modern large language models sometimes memorize training examples into their weights (Xiong et al., 2025; Szep et al., 2026). This memorization is a strong instance of in-weights learning (IWL), where models make predictions based on weight updates (Chan et al., 2022; Singh et al., 2023). Still, this memorization may help a language model generate factually correct text with good grammatical structure.

Table 1: Toy models of in-weights and in-context learning dynamics.

Paper	Task Dataset	Transformer Architecture	Claims
Chan et al. (2022)	Omniglot	12 layers with MLP	Data distributions with many rare classes encourage ICL, while common repeated classes encourage IWL. ICL and IWL can coexist when the data distribution has both of these properties, like a Zipf distribution.
Singh et al. (2023)	Omniglot	12 layers with MLPs	ICL can deteriorate during training. L2 regularization of MLP weights especially can mitigate ICL transience.
Nguyen & Reddy (2024)	Gaussian random vectors mapped to binary labels	One attention layer with a three-layer MLP	IWL and ICL circuits compete, with IWL being slower to develop. ICL transience appears when attention weights are L2 regularized more than MLP weights.
Anand et al. (2025)	Synthetic language data for syntax probing constructed from sampled words from the Penn Treebank 3 dataset (Marcus et al., 1993)	MultiBert checkpoints (Sellam et al., 2022)	Study "structural ICL" where the model cannot use any of the semantic properties of the context tokens to make predictions. This type of ICL is also transient. A variant of active forgetting (Chen et al., 2023) which scrambles the model's token embeddings is used just at the beginning of training to mitigate the transience of ICL.
Park et al. (2025a)	Finite mixtures of Markov chains	Two layers with MLPs.	IWL and ICL strategies compete with each other. ICL transience is due to IWL achieving better loss on training distribution data in the long run. When the model is performing the IWL strategy, Markov chain transition matrices from the training data can be recovered from MLP weights.
Singh et al. (2025)	Omniglot	Two layers, attention-only	"Context-constrained IWL" (CIWL) is asymptotically preferred, and in Omniglot classification the ICL and CIWL mechanisms share subcircuits. They found that ICL forms faster, but is not as favorable for the model to lower training loss asymptotically while the CIWL circuit takes a while to form. Furthermore, the embryonic development of the CIWL circuit allows for ICL to emerge. They used a trick specific to the Omniglot dataset to fix the in-context exemplar to be the same as the query exemplar and showed that this made ICL asymptotically preferred.
Ku et al. (2025)	Sinusoidal regression and Omniglot	Four layers with MLPs	ICL and IWL transience is task-specific and depends on the relative computational costs of each strategy. For example, the Omniglot classification task requires high memorization capacity but is simple for ICL, therefore IWL is slow to develop. The opposite is true for the sinusoid regression task.
Chan et al. (2025)	Random vector inputs mapped to random labels	Two layers with MLPs	Through the perspective of expected generalization error, ICL and IWL compete for which lowers the loss more. IWL is encouraged by data that repeats frequently and ICL is encouraged by rare data that can be reasoned about through context. ICL transience is explained by the rare data showing up enough times for it finally to be memorized.

Methods for controlling the development of different language model capabilities One practical way for modulating the capabilities of deep learning models is through model merging, where the weights of multiple models are combined in certain ways to produce a single model (Yang et al., 2024). Examples include Maiti et al. (2025) using benchmark performance of individual models to determine expert models in certain domains and computing weighted averages of their weights to achieve a high performing and robust single model, Dekoninck et al. (2024) using model merging to control the style of text generation in language models, and Ro et al. (2023) showing that training an ensemble of vision models on portions of a dataset with a variety of difficulty levels and then merging them was a more sample-efficient way to obtain a single model than training on the whole dataset.

A way of suppressing a certain capability from being learned during training is filtering out tokens in the dataset that correspond to that capability (Rathi & Radford, 2026). For post-training techniques, some are using "self-distillation" to mitigate catastrophic forgetting in models by leveraging their ICL capabilities (Shenfeld et al., 2026; Hübotter et al., 2026). To enhance a certain ability when finetuning a model, Snell et al. (2022) proposes to distill context. Noticing that models perform better when given instructions in context and a scratchpad for generation before predicting the final output, context distillation consists of providing a model additional context and a scratchpad along with the training examples, then recording the outputs of the model. These outputs are then used as labels for finetuning. They show that this method makes models better able to internalize the instructions than just supervised finetuning.

Context-enhanced learning Context-enhanced learning corresponds to using the ICL capabilities of models to modulate their training. Similar to Snell et al. (2022) for finetuning, Qi et al. (2025) provides the training examples along with GPT-4 generated context that contains full information of the target to models, then collects their outputs to compute an "in-context editing loss" between the output conditioned on the additional context and the unconditional output. The model is then finetuned on the normal dataset, but this "in-context editing loss" is a regularization term in the training objective. They find that this process mitigates overfitting and leads to more natural outputs. Zhu et al. (2025) provides partial completions or intermediate steps towards the target of a training example to a model during finetuning and shows that models can get to better performance with exponentially fewer training examples. They also show that the additional context does not get memorized by the models. Finally, Lampinen et al. (2025) shows that ICL usually generalizes better than finetuning, so they use a model's ICL capabilities to augment finetuning datasets that lead to better generalization after a finetune.

B NEEDLE-IN-A-HAYSTACK TESTING

To evaluate the model, we create a series of structured "needle-in-a-haystack" test traces by interleaving traces in the testing library of 50 held-out systems, each with 1000 held-out initial states, generated in the same way as the training library from Daniels et al. (2025). Each "needle-in-a-haystack" trace with N systems is generated by inserting a segment of 10 state observations starting from index 0 from the testing library into the test trace. Each of these segments is individually punctuated with a unique open and close symbol pair. We call this portion of the trace the "haystack." We then append a query open symbol to the test trace (corresponding to the needle), followed by 10 state observations that continue the system corresponding to the query open symbol (the query segment). Daniels et al. (2025) gives a more detailed explanation of this procedure. See Fig. 9 for a diagram of a test trace for $N = 2$ systems in the haystack and system U_1 as the needle.

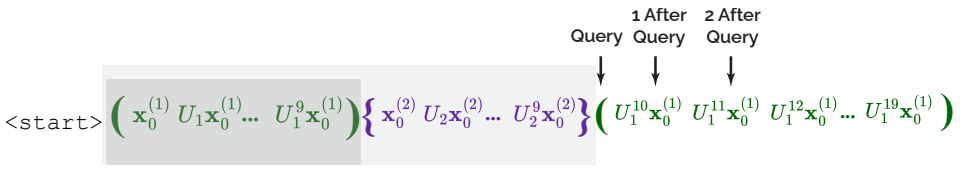


Figure 9: Needle-in-a-haystack test example. This shows a two-system haystack, where the dark gray region corresponds to the needle segment and the light gray region corresponds to the entire haystack.

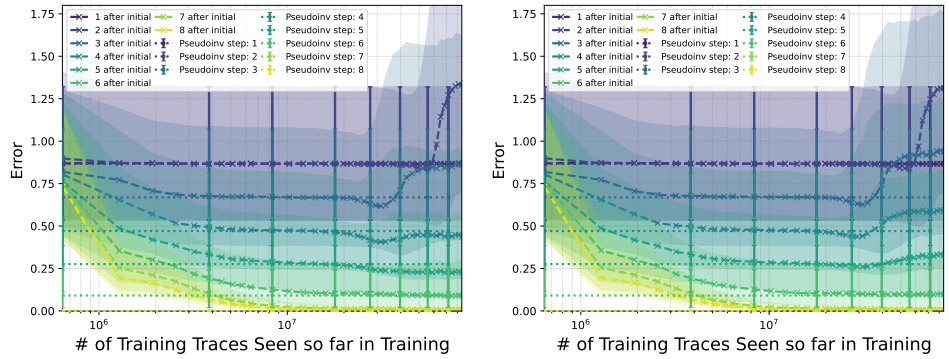
C Model Details

Building on the codebase of Du et al. (2023), which was influenced by Garg et al. (2022), we train a 10.7M-parameter GPT-2-style transformer to perform this task. Our model has hidden dimension 192, 24 layers, and 12 heads. Our model's input embedding is 192×57 dimensional. The model's output layer is 5×192 to ensure that the model makes 5-dimensional predictions. The input and output layers are untied (Du et al., 2023). Experiments have also been done on smaller models, and they exhibit the same phenomena after more training iterations. We focus on the largest model in this work to lower the amount of training time needed to see the relevant phenomena.

We used a batch size of 640, a learning rate of $\approx 1.58 \times 10^{-5}$, and most experiments were trained on a single NVIDIA H200 GPU (Boerner et al., 2023). A single training run takes around 5 days. We used the AdamW optimizer (Loshchilov & Hutter, 2019) and trained using mean-squared error loss.

D Shifting the Initial Trace Index

Here we study how dependent the IWL capabilities are on observing the initial state that corresponds to each system. To do this, we evaluate the model on the training data, but the first observation in each trace has been shifted over to a later index from the training example. The training system is unchanged, so if the model has memorized more than just the initial state, it could still have predictions that converge to zero error. However, in Fig. 10 we see that this does not occur, and that its predictions actually degrade for this shifted training data. This provides evidence that the model's memorization is heavily triggered by the observation of the specific initial state that corresponds to each training system.

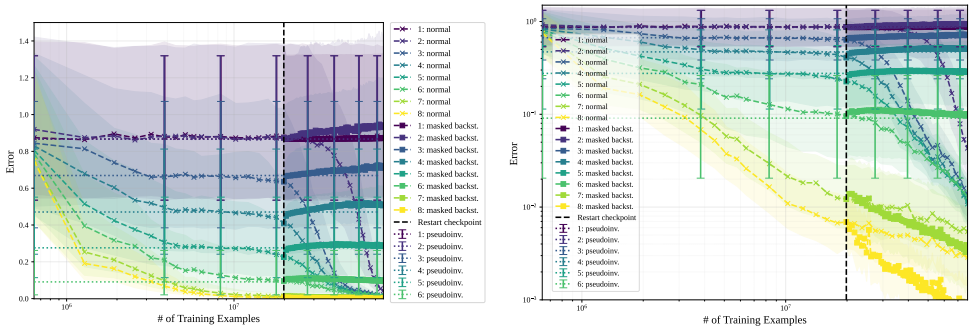


(a) Shifted by one.

(b) Shifted by two.

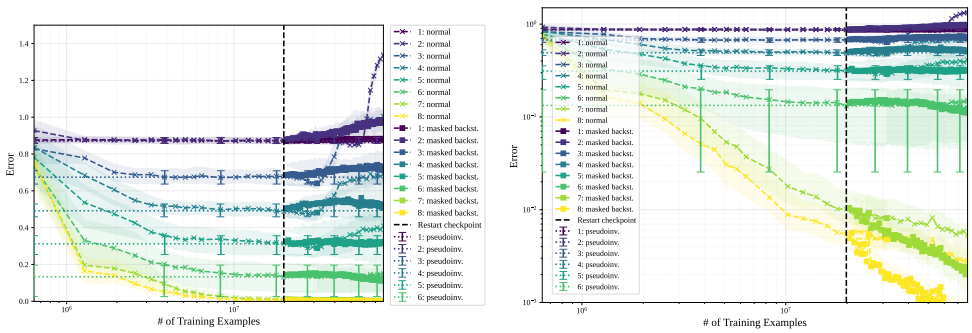
Figure 10: Median squared error when evaluated on training data where the starting index of the training example is shifted. For example, in Fig. 10a, the first observation is $\mathbf{x}_1$ as opposed to $\mathbf{x}_0$. Since none of these curves converge to zero, we see that the model's memorization is dependent on observing the initial state that corresponds to each training system.

E More Figures for Restarting Training with Masked Backstories



(a) Linear scale. (b) Log scale.

Figure 11: Median squared error of the normally trained model (crossed curves), the masked backstory trained model (square-marker curves), and the pseudoinverse predictor (dotted lines) on specific indices into training traces vs the number of training examples seen so far during training. The crossed curves eventually sharply decrease towards zero (signifying a transition to in-weights learning) while the square-marker curves do not.

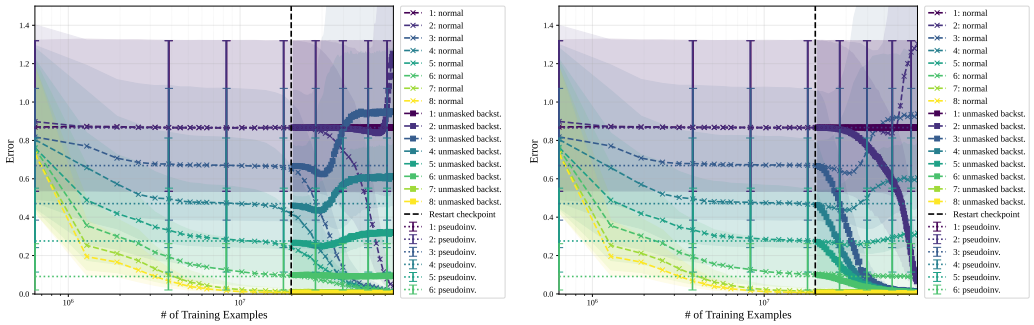


(a) Linear scale. (b) Log scale.

Figure 12: Median squared error of the normally trained model (crossed curves), the masked backstory trained model (square-marker curves), and the pseudoinverse predictor (dotted lines) on specific indices into held-out traces vs the number of training examples seen so far during training. Notice that the crossed curves sharply increase late in training while the square-marker curves remain closer to their corresponding dotted lines. Test traces use a five-system haystack.

E.1 Unmasked Backstories: Teasing Apart the Effect of Masking vs. Segment Lengths

Further analyzing the positive effect of masked backstories on associative recall performance (Section 4.1), we notice that augmenting the training traces with masked backstories means that the minimum length of a segment from a system that was previously unseen in the training example is 7, while the minimum length for the original interleaving procedure is zero (Daniels et al., 2025). A minimum length of 7 means that the full system is learnable by (1), and consequently, when queried for recall the model now more frequently recalls systems that it has more context about. To check if the improved performance is just due to this shift in the distribution of segment lengths in the training data, we trained a model with *unmasked* backstories. This model sees the same training data as the masked backstory trained model; the only difference is that the training loss on the backstory indices is not zeroed out. The performance of the unmasked backstory model shows us any ICL performance gains that can be made just by the shift in the segment length distribution. In Fig. 13 we see that the unmasked backstory trained model memorizes the backstories as opposed to the unbackstoried training data, and it still has a corresponding degradation in ICL on the initial haystack segment. Therefore, a shift in the length distribution alone will not remedy this effect.



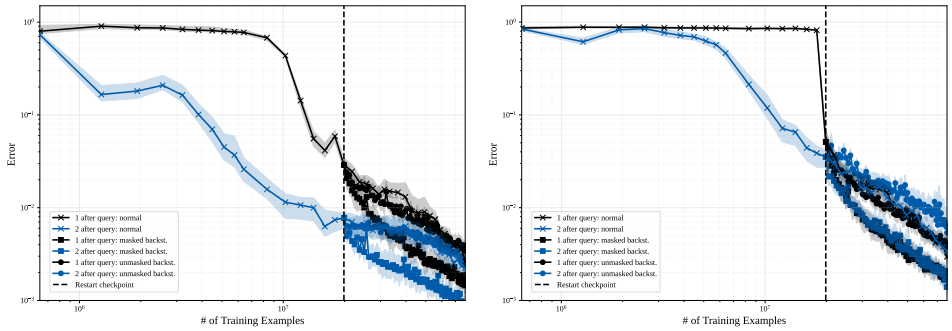
(a) Training traces.

(b) Backstoried training traces.

(c) Held-out test traces.

Figure 13: Median squared error of an unmasked-backstory trained model restarted from a mid-training checkpoint, evaluated on (a) training traces, (b) backstoried training traces, and (c) held-out test traces.

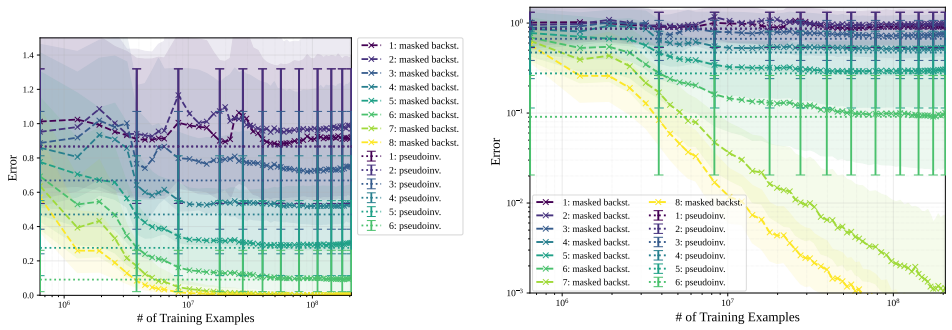
In Fig. 14, the unmasked backstory model's recall performance is shown by the solid curves with circle markers. Looking at the solid black curves for the first index into the test segment, the unmasked backstory trained model performs better than the normally trained model, but worse than the masked backstory trained model. Looking at the solid blue curves for the second index into the test segment, the unmasked backstory trained model performs at the same quality as the normally trained model, and worse than the masked backstory trained model. This shows that the improvements that we see from the masked backstory trained model cannot be solely attributed to the shift in the segment length distribution.



(a) 1 system haystack. (b) 5 system haystack.

Figure 14: Median squared error of the normally trained model (crossed curves), the masked backstory trained model (square curves), and the unmasked backstory trained model (circle curves) on indices one and two into the test segment of held-out "needle-in-a-haystack" traces vs the number of training examples seen so far during training. See that the masked backstory trained model's squared error (square curves) is always the lowest for each index.

F Training with Masked Backstories from the Beginning

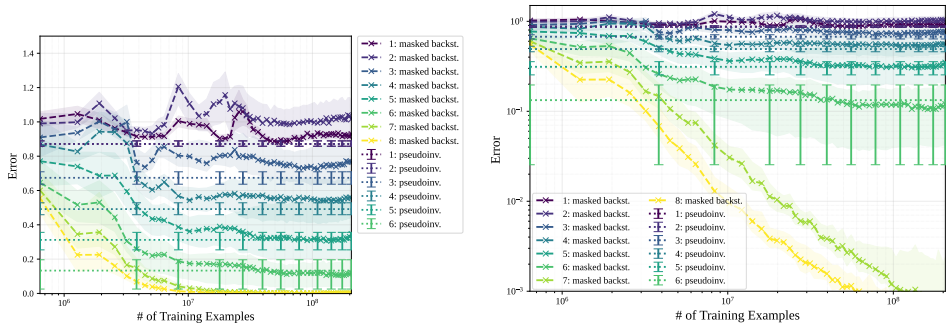


(a) Linear scale. (b) Log scale.

Figure 15: Median squared error of the masked backstory trained model (crossed curves) and the pseudoinverse predictor (dotted lines) tested on backstoried training traces vs the number of training examples seen so far during training. The model's prediction error converges to the pseudoinverse predictor's error without any sudden shifts to in-weights learning.

Section 4.1 showed that when restarting training with masked backstories from a checkpoint that was trained using the normal interleaving procedure before the model's generalization performance deteriorates, the model's transition to in-weights learning is blocked, and the model's in-context associative recall abilities are improved. Now, we ask the question: what happens when we train a model with masked backstories from the very start?

First, we check if the model trained using masked backstories from the very beginning exhibits a transition to in-weights learning like the normally trained model. We check this by testing this model on the training traces with prepended backstories, since it is the prepended backstories now that have irreducible loss for this model. In Fig. 15, we plot the median squared error of the pseudoinverse predictor (dotted lines) and the transformer model (dashed curves with cross markers) for the first through the eighth index into the backstoried training traces. In this figure, we see that through 2×10^8 training examples seen during training the transformer model's median squared error converges down almost to the pseudoinverse predictor's median squared error without any sudden drop down to zero median squared error as we saw with the normally trained model at around 2×10^7 training examples in Fig. 11. Additionally, we also test the model on held-out "needle-in-a-haystack" traces and Fig. 16 shows the corresponding plot for generalization performance. Notice that the crossed



(a) Linear scale. (b) Log scale.

Figure 16: Median squared error of the masked backstory trained model (crossed curves) and the pseudoinverse predictor (dotted lines) tested on held-out traces vs the number of training examples seen so far during training. There is no degradation in the model's generalization performance.

curves remain horizontal and do not sharply increase through 2×10^8 training examples seen unlike the crossed curves in Fig. 12 at 3×10^7 training examples seen for the normally trained model. These two results together show us that the model trained with masked backstories from the beginning does not transition to in-weights learning for the early indices up to this point in training.

> Since this model was trained with masked backstories from the beginning, it was never penalized during training for poor predictions on the first 7 indices of an initial segment. Nonetheless, it still developed the ability to make near-optimal predictions for these indices.

The model's median prediction error approaching the pseudoinverse predictor's median prediction error in Fig. 16 provides the evidence for this. Furthermore, one could think of the task of predicting the next vector in the segment when fewer than six observations have been seen as a more difficult task than predicting the next vector when there are enough observations to know the underlying system perfectly. This is because when there are not enough observations, the model must learn how to deal with the extra degrees of freedom caused by the underdetermined system of equations. Therefore, in this toy setting, the model is exhibiting a form of "easy-to-hard" generalization (Hase et al., 2024; Sun et al., 2024; Schwarzschild et al., 2021; Song et al., 2025), where its ability to predict the next vector perfectly when there are enough observations also allows it to reasonably predict the next vector when it faces uncertainty.

The solid curves in Fig. 17 show the recall performance of the normally trained model (dot markers), the masked backstory trained model (square markers), and the unmasked backstory trained model (circle markers). First, notice that the square marked curves stay below the dot marked curves, which means the masked backstory trained model's associative recall ability is superior to the normally trained model's. Next, looking at the circle marked curves, in the beginning of training they follow the improved performance of the masked backstory trained model, but later in training, around 3×10^7 training examples seen, the unmasked backstory trained model improves at a slower rate than the masked backstory trained model. Training a model from scratch with masked backstories seems to lead to a decisive advantage in associative recall performance.

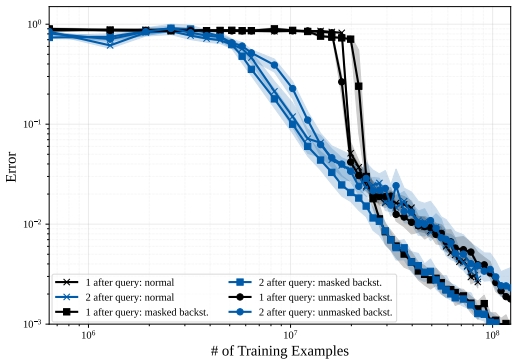
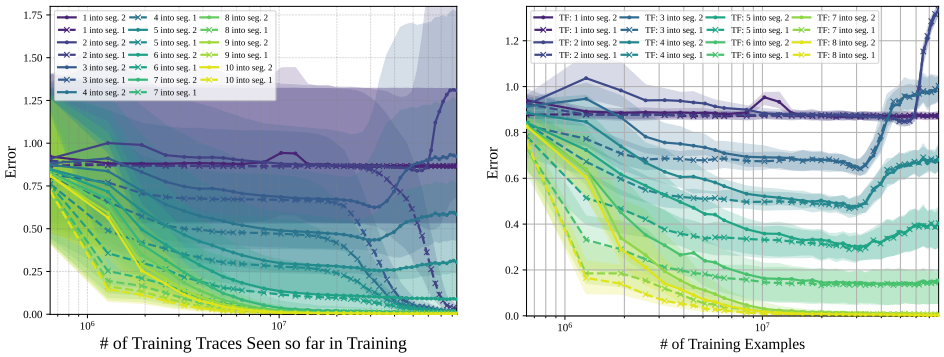


Figure 17: Median squared error of the normally trained model (dotted curves), the model trained with masked backstories from the beginning (square curves), and the model trained with unmasked backstories from the beginning (circle curves) on indices one and two into the test segment of held-out "needle-in-a-haystack" traces with 5 systems in the haystack vs the number of training examples seen so far during training. Again, the masked backstory trained model's squared error (square curves) is always the lowest for each index.

F.1 Performance at Another Haystack Segment

In the main paper, we focus on the first indices of a needle-in-a-haystack prompt, but in principle, memorization can occur at the beginning of any of the subsequent haystack segments as well. Here, we find that not only does the model not memorize at these later positions, its performance for predicting both training and test data degrades there as well.



(a) Train. (b) Held-out test data.

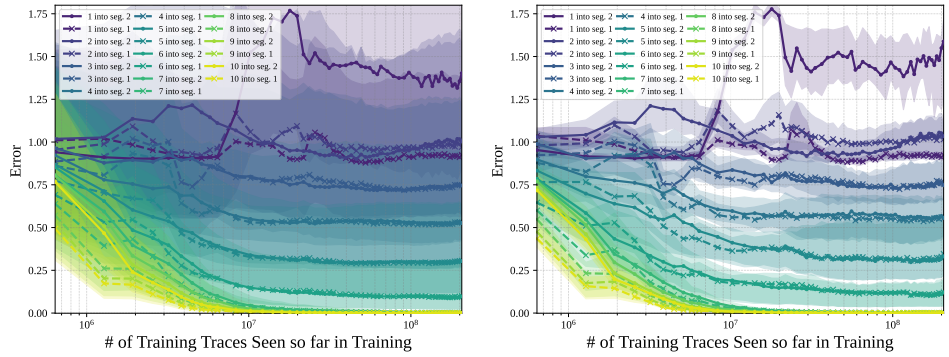
Figure 18: The median squared error of the normally trained model on the first system to appear in the haystack (dashed curves with crosses), and the second system to appear in the haystack (solid curves with dots). In Fig. 18a, memorization is visible in the first haystack segment, but this is not true in the second haystack segment. Notice the degradation in the error for most indices. In Fig. 18b there is degradation for both the initial haystack segment and the subsequent one.

In Fig. 18a, performance on the training data is shown. Memorization is clearly visible in the first haystack segment, as the dashed curves sharply drop towards zero. Contrarily, this is not true in the second haystack segment. Surprisingly, the solid curves actually rise late in training, showing that the development of memorization in the initial segment leads to the model becoming worse at predicting its training data in subsequent segments. In Fig. 18b, it is shown that memorization of the initial segment leads to degraded generalization performance for both segments.

This motivates our alternate intervention of only applying masked backstories to the initial haystack segment, with results shown in Section F.2.1. However, the results in that section show that back-

stories applied to every segment where a system appears for the first time are necessary to maintain held-out test performance on the haystack segments.

Fig. 19 provides the performance on the first and second haystack segments for the masked backstory model. Here, we see that all curves except the unpredictable first index of the second segment do not degrade and maintain their near-optimal performance.



(a) Train. (b) Held-out test data.

Figure 19: The median squared error of the masked backstory trained model on the first system to appear in the haystack (dashed curves with crosses), and the second system to appear in the haystack (solid curves with dots). Other than the unpredictable first index in the second haystack segment, no other indices show degraded performance.

F.2 Further Interventions

To better justify our design choices for pretraining with masked backstories, here we provide results for modified implementations of masked backstories. Some of these modifications are shown in Fig. 20. These results are summarized in Fig. 21 and help surface the separate effects that masked backstories pretraining has on a model's training dynamics.

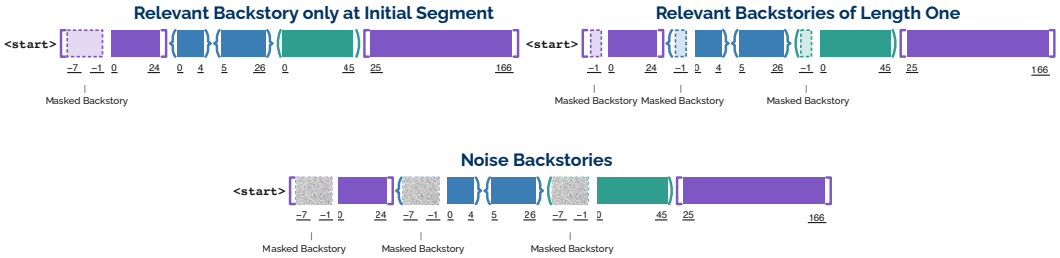


Figure 20: Diagrams for the various interventions studied in Section F.2. The backstory only at the initial segment experiment is studied in Section F.2.1, backstory of length one in Section F.2.2, and noise backstories in Section F.2.5.

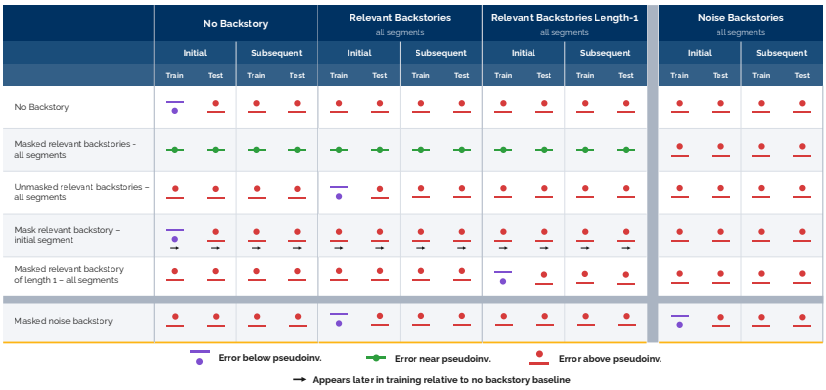
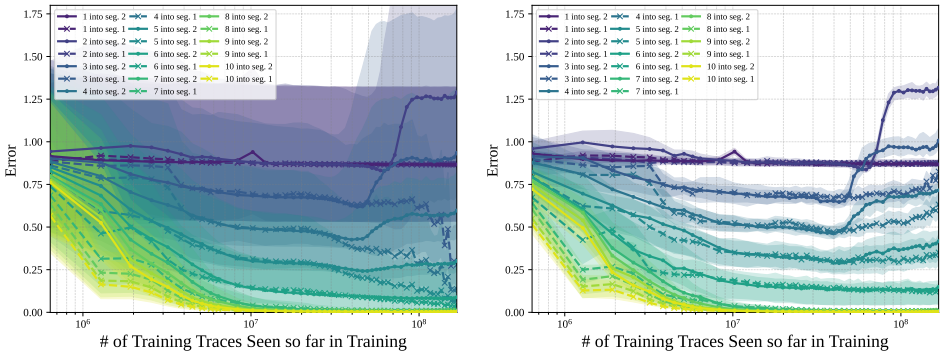


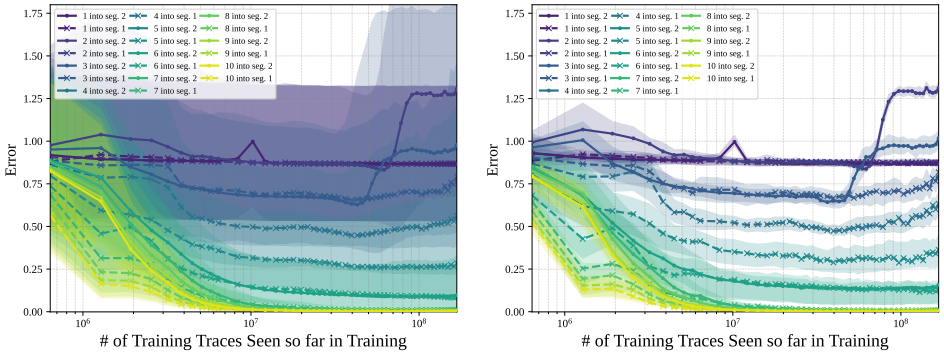
Figure 21: Results from various modifications of the masked backstories intervention. Each row corresponds to a trained model, and each column an evaluation dataset. For example, the entry in the first row and column shows that the model trained with no backstories performs preternaturally well on training data at the initial segment of the haystack.

F.2.1 Backstories only for the Initial Segment



(a) Train, no backstory.

(b) Test, no backstory.



(c) Train, backstory of length 7.

(d) Test, backstory of length 7.

Figure 22: Training dynamics for the model trained with masked backstories only for the initial segment. Each sub-caption shows which evaluation dataset was used. Notice that Fig. 22a shows memorization of the unbackstoried training data only at the initial segment. This memorization occurs after $\approx 10^8$ training examples for this experiment while it occurs after $\approx 2 \times 10^7$ training examples for the normally trained model.

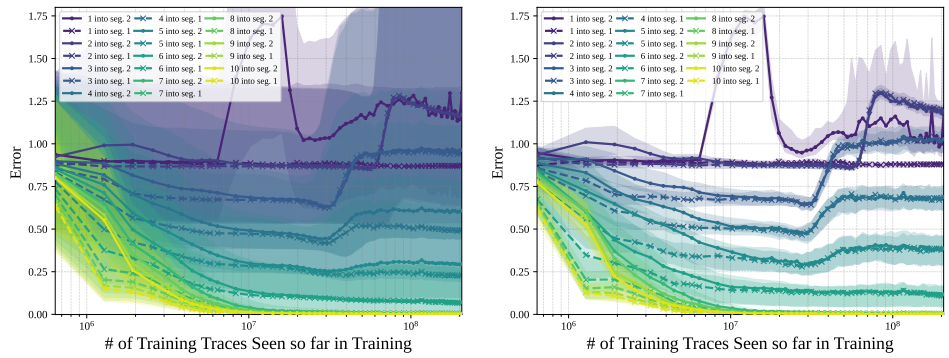
When the model is trained without any interventions, Fig. 18 shows that the model only memorizes the beginning indices of the initial haystack segment, and gets worse at predicting the beginning indices of subsequent haystack segments. So a natural question is, if masked backstories are only inserted before the initial segment during training, will the memorization be suppressed? Fig. 22 shows that the answer is no. Training with masked backstories only at the initial segment leads to degraded ICL abilities (Figs. 22b and 22d).

> Surprisingly, Fig. 22a shows that the model only memorizes indices in the initial segment, which the model had no gradient pressure to memorize during training, while losing performance on indices in subsequent haystack segments that did have gradient pressure during training.

This counterintuitive result shows that providing supervision at one portion of a training example can lead to transferable abilities in other portions of the example and degraded abilities at the portion that received the supervision itself.

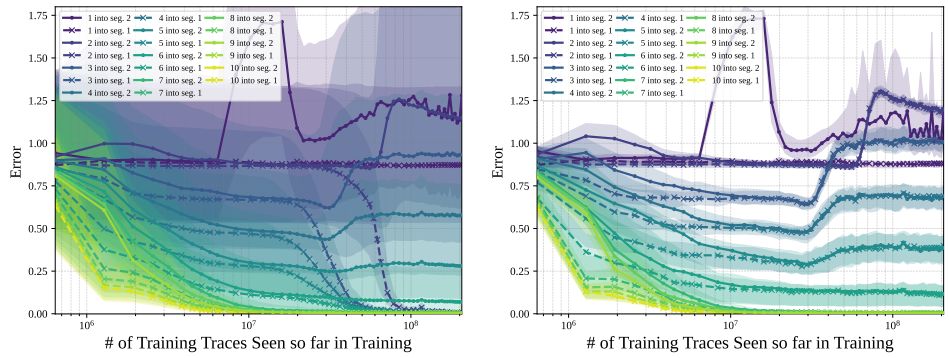
F.2.2 Backstories of Length One

One of our main hypotheses for why masked backstories pretraining is an effective intervention is that the additional context allows the model to use its in-context learning ability to predict subsequent indices that would have previously required memorization to get low pretraining loss. To test this hypothesis, we trained a model where, instead of providing a backstory of length 7, we provide a backstory of length one. One observation vector does not provide any information about the underlying system, so this backstory of length one is not additional context that can be exploited for better in-context prediction. Here, in Fig. 23a the model does not memorize the unbackstoried training data, but it does memorize the backstoried training data as shown in Fig. 23c. Again, this memorization causes degradation of performance for subsequent haystack segments and for held-out test data (Figs. 23b and 23d).



(a) Train, no backstory.

(b) Test, no backstory.



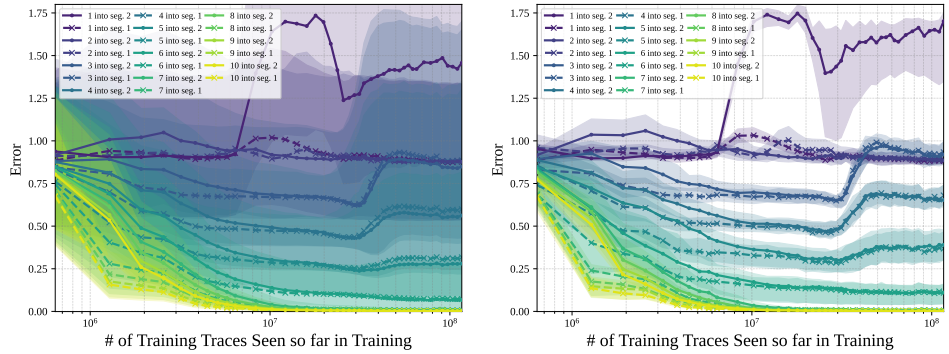
(c) Train, backstory of length one.

(d) Test, backstory of length one.

Figure 23: Training dynamics for the model trained with masked backstories of length one. Each sub-caption shows which evaluation dataset was used. Notice that Fig. 23c shows memorization of the backstoried training data, but Fig. 23a shows no memorization of the unbackstoried training data.

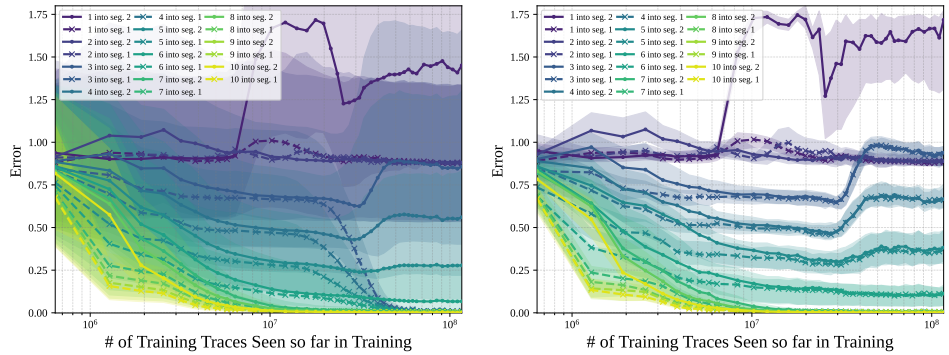
F.2.3 BACKSTORIES OF LENGTH TWO

This experiment is identical to the masked backstories of length one experiment from Section F.2.2, except each masked backstory is now two indices long. With two observations, some information about the underlying system is provided. Nonetheless, the results here are the same as the length-one case where the model does not memorize the unbackstoried training data (Fig. 24a), but it does memorize the backstoried training data (Fig. 24c).



(a) Train, no backstory.

(b) Test, no backstory.



(c) Train, backstory of length two.

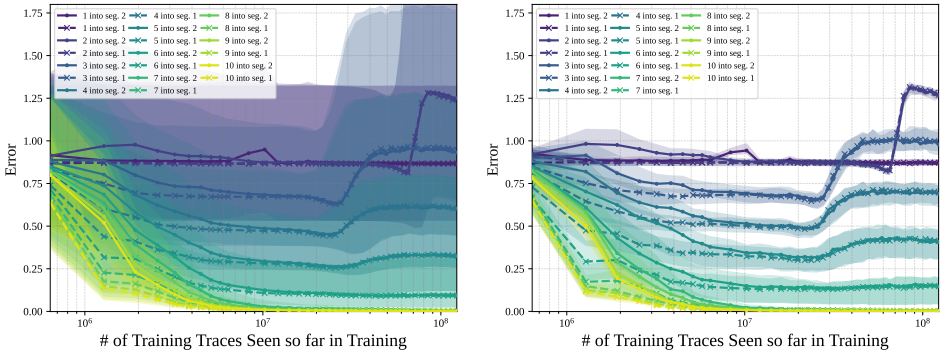
(d) Test, backstory of length two.

Figure 24: Training dynamics for the model trained with masked backstories of length two. Each sub-caption shows which evaluation dataset was used. Notice that Fig. 24c shows memorization of the backstoried training data, but Fig. 24a shows no memorization of the unbackstoried training data.

The results from masked backstories of length one and two experiments provide evidence that the additional context must provide significant predictive power to the model in order for masked backstories to be an effective pretraining intervention.

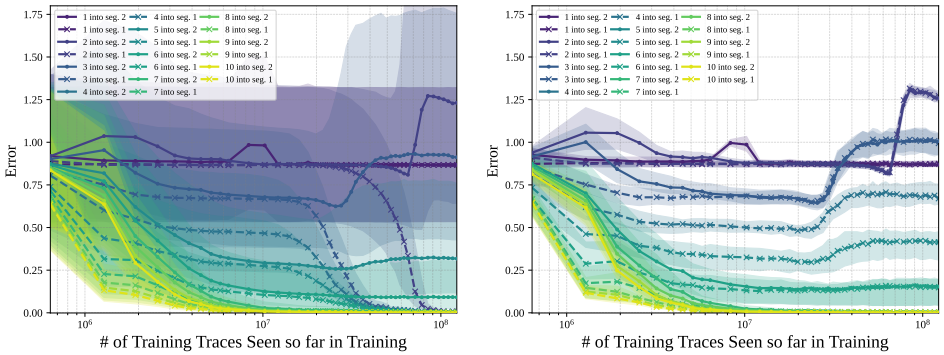
F.2.4 UNMASKED BACKSTORIES

Now, we study the counterpart to Section E.1 for training with backstories from the beginning. Again, here we show that masking the backstory is an important aspect of the intervention, as the unmasked backstory trained model no longer memorizes the unbackstoried training data (Fig. 25a), but it memorizes the backstoried training data (Fig. 25c). This memorization again leads to degraded generalization performance (Figs. 25b and 25d).



(a) Train, no backstory.

(b) Test, no backstory.



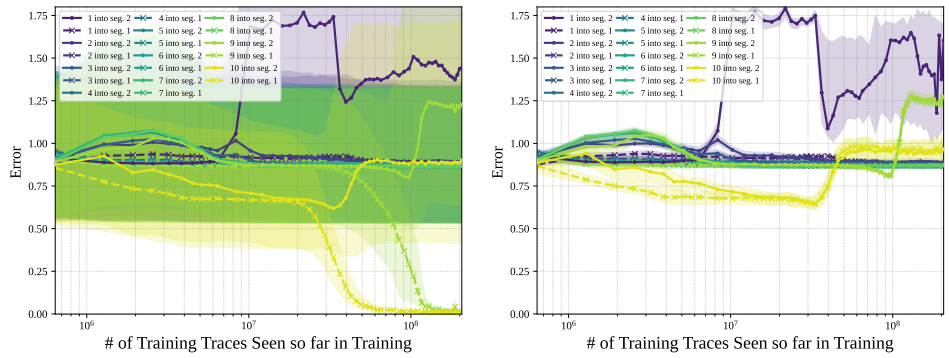
(c) Train, backstory of length 7.

(d) Test, backstory of length 7.

Figure 25: Training dynamics for the model trained with unmasked backstories of length 7. Each sub-caption shows which evaluation dataset was used. Fig. 25c shows memorization of the backstoried training data.

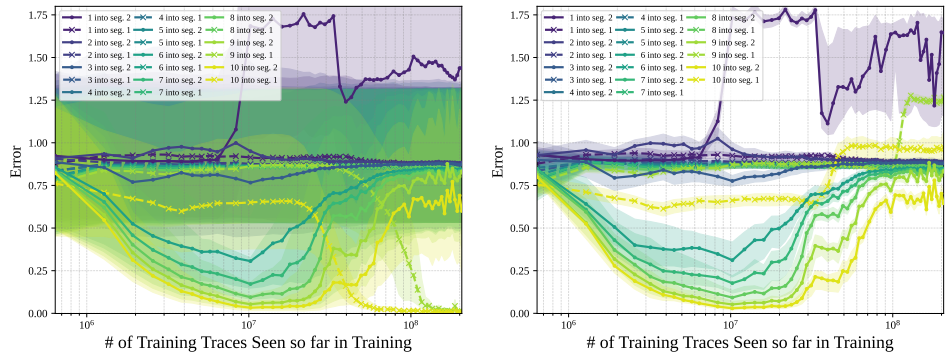
F.2.5 NOISE BACKSTORIES

To further test the claim that the additional context must provide significant predictive power for masked backstories to be an effective technique, we can provide 7 indices of IID Gaussian noise of the same expected squared norm as the linear dynamical system observations at the first occurrence of each system in a training example. Masking the loss of this noise then results in noise backstories. As this IID noise is unpredictable, the model must learn to ignore the noise, then make optimal predictions on the linear dynamical system observations afterward. Here, we see in Figs. 26a and 26c that the model again memorizes the first indices that are memorizable. In this case they are indices 9 and 10 in the initial haystack segment. This experiment again underscores the importance of additional context that provides information that can be used for prediction.



(a) Train, IID Gaussian backstories of length 7.

(b) Test, IID Gaussian backstories of length 7.



(c) Train, backstory of length 7.

(d) Test, backstory of length 7.

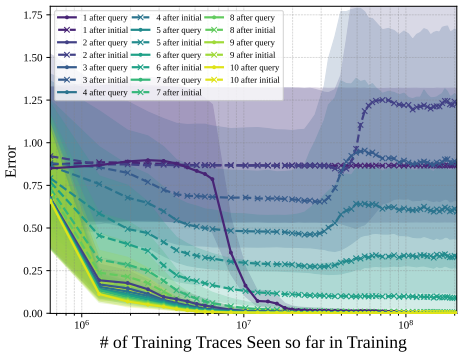
Figure 26: Training dynamics for the model trained with IID Gaussian backstories of length 7. Each sub-caption shows which evaluation dataset was used. Figs. 26a and 26c show memorization of indices 9 and 10 in the initial haystack segment.

Interestingly, in Figs. 26c and 26d, which show the model's performance on backstoried data from the actual linear dynamical systems, we can see that early in training the model is making accurate predictions for early indices in the second haystack segment before learning to predict near zero. This shows that the model uses in-context learning early in training before it learns that these indices in its training data consist of IID Gaussian noise.

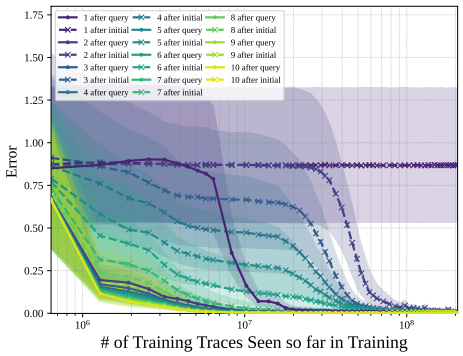
G MASKING ONLY A FRACTION OF THE SYSTEMS

For all of the previous experiments, when applying masked backstories, they were applied to every training system. Now we can ask the question, if the masked backstories are only applied to a fraction of the training systems, can the ICL ability of the model be maintained while allowing for the memorization of a select group of systems?

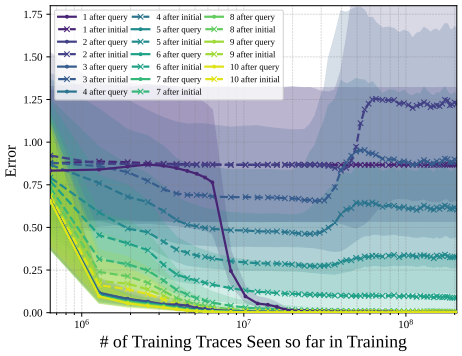
G.1 MASKING THREE-FOURTHS OF THE TRAINING SYSTEMS



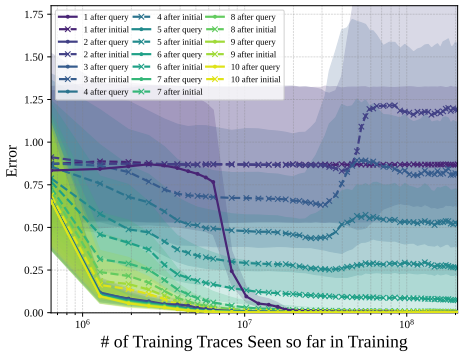
(a) Train, masked subset.



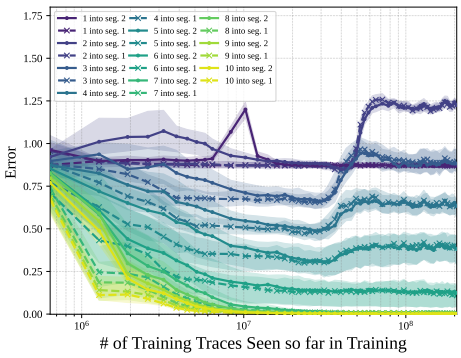
(b) Train, unmasked subset.



(c) Train, backstory of length 7, masked subset.



(d) Train, backstory of length 7, unmasked subset.



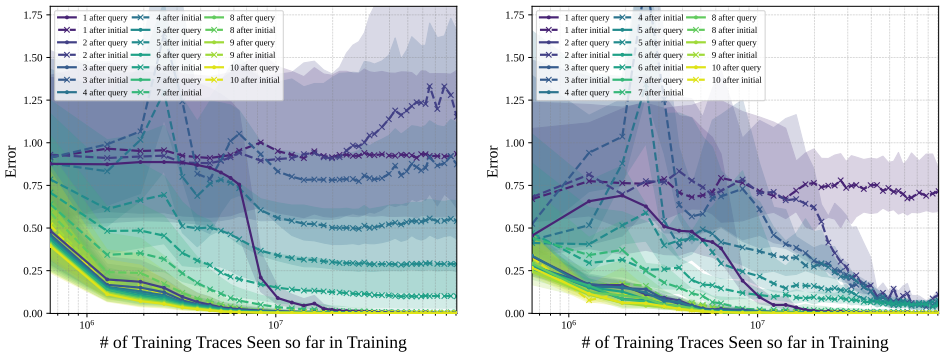
(e) Test.

Figure 27: The training dynamics of the model with 75% of the training systems masked when predicting needle-in-a-haystack style data. Each sub-caption shows which evaluation dataset was used. Fig. 27c shows that the masked systems are not memorized and Fig. 27d shows that the unmasked systems are memorized. Fig. 27e shows that the model's ICL ability is adversely affected by this memorization.

Here, we apply masked backstories to 30,000 of the training systems while allowing 10,000 of the training systems to appear in the training data without masked backstories. We provide evaluations on held-out systems, the masked training systems, and the unmasked training systems. We find that the unmasked systems get memorized after $\approx 2 \times 10^7$ training examples (Fig. 27b), the same point in training where memorization occurs in the normally trained model. For the masked systems, not only do they not get memorized, but the model's performance on this training data degrades (Fig. 27a). Overall, on held-out test data performance degrades once the 10,000 unmasked systems

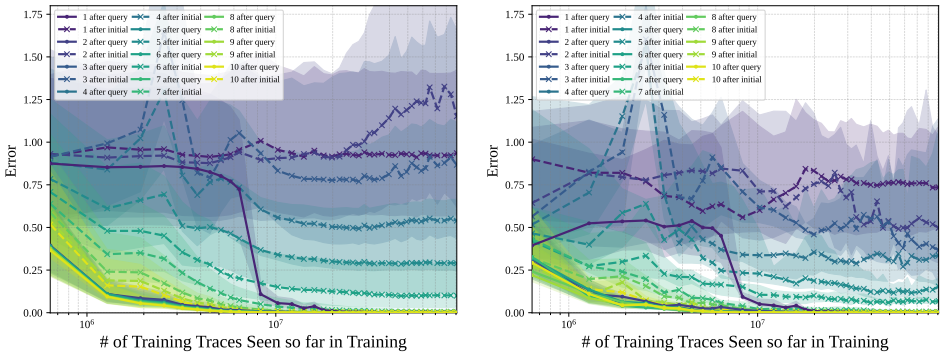
get memorized as well (Fig. 27e). This shows that the model's ICL ability is very sensitive to the development of in-weights memorization.

G.2 Masking All but Ten Training Systems



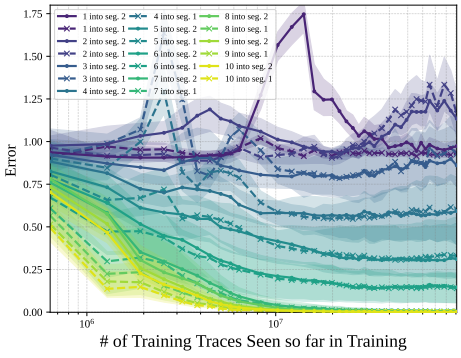
(a) Train, masked subset.

(b) Train, unmasked subset.



(c) Backstory train, masked subset.

(d) Backstory train, unmasked subset.



(e) Held-out test data.

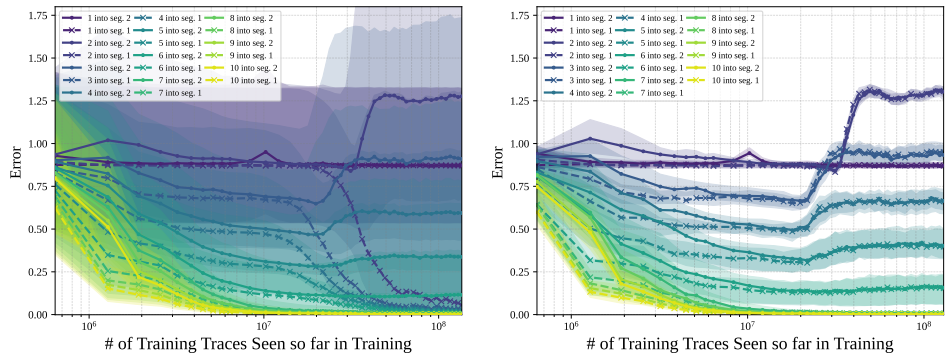
Figure 28: The training dynamics of the model with 99.975% of the training systems masked when predicting needle-in-a-haystack style data. Each sub-caption shows which evaluation dataset was used. Fig. 28c shows that the masked systems are not memorized and Fig. 28d shows that the unmasked systems are memorized. Fig. 28e shows that the model's ICL ability is still moderately affected by this memorization.

We can go to an extreme and see if the same effect from Section G.1 manifests when only 10 training systems are unmasked and 39,990 are masked. Again, in Fig. 28b we see that the unmasked systems are memorized. In Fig. 28a the masked systems are not memorized, but in Fig. 28e, the median squared error of the predictions in the initial haystack segment for held-out test data still appears

to be slowly rising. This means that even when the unmasked subset takes up 0.025% of training systems, the memorization of these systems adversely affects the model's ability to perform ICL.

H Multiple Initial State Vectors Corresponding to a Single Training System

For the previous experiments, each training system had a corresponding initial state vector. This is what allowed for the memorization of systems. A relevant question is whether increasing the number of corresponding initial state vectors to a single system will make the system harder to memorize. Here we present results for an experiment where the training library of sequences consisted of 20,000 training systems and each had two corresponding initial state vectors. This means that the training library still had 40,000 total sequences as was the case for the earlier experiments as well. Here we see that this mapping of two initial state vectors to one system had qualitatively no effect on the training dynamics compared to the one-to-one setting. Memorization still occurs after $\approx 2 \times 10^7$ training examples (Fig. 29a) and ICL ability still degrades (Fig. 29b).



(a) Train.

(b) Held-out test data.

Figure 29: Two initial state vectors per training system. Memorization still occurs after $\approx 2 \times 10^7$ training examples (Fig. 29a) and ICL ability still degrades (Fig. 29b).