How Truncating Weights Improves Reasoning in Language Models

author names withheld

Under Review for the Workshop on High-dimensional Learning Dynamics, 2024

Abstract

In addition to the ability to generate fluent text in various languages, large language models have been successful at tasks that involve basic forms of logical "reasoning" over their context. Recent work found that selectively removing certain components from weight matrices in pre-trained models can improve such reasoning capabilities. We investigate this phenomenon further by carefully studying how certain global associations tend to be stored in specific weight components or Transformer blocks, in particular feed-forward layers. Such associations may hurt predictions in reasoning tasks, and removing the corresponding components may then improve performance. We analyze how this arises during training, both empirically and theoretically, on a two-layer Transformer trained on a basic reasoning task with noise, a toy associative memory model, and on the Pythia family of pre-trained models tested on simple reasoning tasks.

1. Introduction

Large language models (LLMs) have shown impressive capabilities on a variety of tasks, from generating coherent and grammatically correct text, to language understanding and basic mathematical reasoning [8, 39]. At the heart of this success is the Transformer architecture [40], which relies on a sequence of self-attention and feed-forward layers to efficiently combine information from the input context and patterns learned from training data. Despite recent progress on interpreting the mechanisms learned by different layers [23, 42], these models remain largely black boxes. A better understanding of the role of Transformer layers and how they are affected by the training process could enable new monitoring and editing techniques, better training data, and ultimately more reliable LLMs.

The task of next-token prediction in language modeling inherently involves different subtasks that may be at odds with each other. For instance, given the context "John gave a book to", the word "the" is a natural and grammatically correct next word to predict, and relying on global bigram statistics might be enough to predict it given the last word "to". Nonetheless, if another character is present in the context, say Mary, then the name "Mary" may be a better prediction, and this would require a more involved form of "reasoning" over the context to retrieve this name. Previous work on interpretability has found that "circuits" of attention heads seem responsible for such in-context predictions [42], while feed-forward layers may be storing more global statistics such as the bigram "to the" or general factual knowledge [18, 23]. The recent work [33] found that selectively replacing certain layer weights to their low-rank approximation may improve performance on various reasoning benchmarks, and observed that the truncated components were often responsible for predicting "generic" words such as "the".

In this paper, we provide a finer understanding of these phenomena by studying how such mechanisms arise during training, in particular how global associations, such as the bigram "to the", can be localized to specific components or layers of the model weights. We first investigate this on pre-trained language models, namely the Pythia family, which has checkpoints available at different training steps [6]. We then provide a fine-grained study of dynamics on simple data models and architectures exhibiting similar properties:

- In a two layer transformer architecture trained on an in-context recall task similar to [7], but with additional noise on in-context tokens, we show that the noise is mainly learned in feed-forward layers, even for large noise levels. Removing those layers then leads to clean in-context predictions. We provide some theoretical justification through the first gradient step.
- In a linear associative memory model trained on data involving a common noise token, we show that the noise can be identified in a rank-one subspace of the weights. When the noise level is small, low-rank truncation can filter it out and predict clean outputs.

Overall, we provide a useful description of how global associations and in-context reasoning mechanisms are learned during training, and tend to be disentangled in different parts of the model, such that selectively removing certain components may lead to better predictions in reasoning tasks.

Related works are provided in Appendix A.

2. Background and Motivation

In this section, we provide some background and motivation on reasoning tasks and rank reduction, and conduct initial investigations on pre-trained language models.

2.1. Reasoning from Context

Recent LLMs have shown promising results in more complex "reasoning" tasks which may involve multiple steps of logical or computational processing from context or prompt [9, 14, 35, 43], as opposed to simple pattern matching or memorization of training data, for instance using learned n-gram predictions.

While it is difficult to clearly separate reasoning from memorization, in this work we will make the simplifying distinction that **reasoning** involves dependencies between *multiple tokens* potentially far away in the context, while we consider **global associations** as simpler predictions that only depend on the *last token*, e.g., through a global bigram model. Thus, reasoning will typically require using attention operations in Transformers over context, while feed-forward layers should suffice for learning global associations.

Under this definition, we list a few simple examples of reasoning that we will consider in the sequel:

- *In-context recall*: when the last token is `a`, we'd like to copy the token that follows previous occurrences of `a` in the context. This `[.. a b .. a]` $\to$ `b` pattern typically requires a two-layer attention mechanism known as an *induction head* [7, 17];
- *Indirect object identification (IOI)*: we consider contexts of the form "When Mary and John went to the store, John gave the ice cream to" where the prediction should be "Mary" (IO, the indirect object), instead of "John" (S, the subject). Wang et al. [42] found a circuit of several attention heads that perform this task by copying the name which only occurs once in the context;

- *Factual recall*: sentences of the form "Paul Citroen is a native speaker of" with target "Dutch" as in [33]. While this may be seen as retrieving a global association, we will treat it here as reasoning since it involves combining the subject and relation from the context, while a global bigram that only depends on the last token "of" might instead predict the word "the."

We note that our assumption of global associations depending only on the last token is mainly for convenience of our analysis. In practice, the last token's representation at intermediate layers of the Transformer may contain additional information from the context, and our arguments can easily extend to global associations that only depend on that representation. For instance, this could include previous tokens thanks to position-based attention heads [2, 17, 41], which allows global n-grams instead of just bigrams.

2.2. LASER: Layer-Selective Rank Reduction

[33] observed that reducing the rank of MLP matrices in certain layers of LLMs effectively brings better performance on several reasoning benchmarks. Their proposed method, Layer-Selective Rank Reduction (LASER), replaces any matrix in the full model by its low-rank approximation with fraction ρ, *i.e.*, a matrix $\mathbf{W} \in \mathbb{R}^{d_{\text{in}}, d_{\text{out}}}$ would be replaced by its rank-$\lfloor \rho \cdot \min\{d_{\text{in}}, d_{\text{out}}\} \rfloor$ approximation via Singular Value Decomposition (SVD). We refer to their Table 3 for more results of the parameters after searching.

Another observation from [33] is that, when LASER improves the model's prediction on some samples, the full model often predicts "generic" words while the improved model is able to predict the ground-truth answer. For instance, given an input "Madrid is located in", the full model predicts "the" while the truncated model predicts the target "Spain" in Table 1. Here, the generic word is consistent with our definition of global associations in Section 2.1, as it may naturally follow from a bigram distribution conditioned on "in", while the factual answer is more akin to reasoning from context. Thus, we would like to better understand how LASER improves the model from predicting generic words to inferring the answer from context, and how such a gap appears during training.

2.3. An Investigation on GPT-2 Small and Pythia Models

In this section, we empirically investigate how LLMs process in-context vs global associations, and how this evolves during training. We consider GPT-2 small and Pythia models on the indirect object identification (IOI) and factual recall tasks described in Section 2.1.

IOI on GPT2 Small. Different from [42], we would like to consider whether a model proposes an output beyond the input x. A quick demonstration is to consider the IOI task with input $x =$"When Mary and John went to a store, John gave a drink to"[1]. The top 4 predicted tokens for GPT-2 Small [30] on x are ["Mary", "them", "the", "John"]. Although GPT-2 Small successfully predicts Mary (the IO target) instead of John (S), the other two top candidate tokens, *i.e.*, "them" and "the", do not even appear in the context. This prominence of such "generic" words is similar to the factual recall example from Section 2.2, and plausibly follows from a global associative mechanism conditioned on the preposition "to".

Therefore, for the above input x, we naturally extend the candidate set as $\mathcal{C} = \{$"Mary", "them", "the", "John"$\}$. To verify whether or not the emergence of "the" is connected to the mechanism of LASER, we examine how the probability of each $c \in \mathcal{C}$ change after running LASER on different

1. Note that here we use "a" store instead of "the" store in the original example of [42]. The reason is to rule out the word "the" from the input context.

layers on GPT-2 Small in Figure 1. LASER on Layer 9, 10 and 11 turns out to significantly decrease the probability of predicting "the" and "them" compared with the full model.

The above demonstration on GPT-2 Small implies that, when a model introduces extra candidates beyond the input x, LASER may decrease the probability of predicting these extra candidates, which means LASER may enhance the model's performance on contextual tasks.

IOI on Pythia-1B. Now we would like to verify this observation on more models and, more comprehensively, track the behavior of these models along training. We choose to conduct the IOI experiments on Pythia [6], a family of models ranging in sizes from 14M to 12B trained on web data, with hundreds of training checkpoints for each size. We generate an IOI dataset of 100 sentences with random names for [IO] and [S] in each sample. Figure 2 reports the test results of Pythia-1B along training. Here LASER is conducted on MLP weights, with parameters given in Appendix D.2. LASER boosts the probability ratio of [IO] over "the" from 2.3× to 12.3× at 14K steps.

Factual recall on Pythia-1B. As in Table 1, we verify factual recall with input as "Madrid is located in". The full model of Pythia-1B generates "Madrid is located in the north of Spain", while the model after LASER generates "Madrid is located in Spain". We track the probability of predicting "Spain" and "the" along training in Figure 2. LASER turns out to boost the probability ratio of "Spain" over "the" from 0.16× to 11.3× at 14K steps.

Training dynamics on Pythia. The behavior of the Pythia models on the IOI and factual recall tasks during their pre-training process displays several phases, as shown in Figure 2. For IOI, we observe:

i. Initialization: all tokens have similar logits since the weights are random initialized.

ii. Between 10 and 1000 steps: the models consistently output "the". They cannot solve IOI task at all, as long as they have almost the same output for [IO] and [S]. After 500 steps, [IO] starts the growth towards one of the top predictions.

iii. After 2000 steps: Pythia starts to be able to solve IOI task by preferring [IO] than [S] and "the". Meanwhile, the benefit of LASER appears as enhancing the leading position of [IO].

Therefore, the training process reveals the capacity of predicting "the" is learnt much earlier than predicting [IO]. The reason might be that predicting "the" requires a simpler grammar structure, while predicting [IO] requires a complicated architecture of attention heads of different roles across layer [42]. Then we note that the IOI task always has "to" before the masked [IO], which means "to" may be an indicator for the model to predict "the" with non-negligible probability. Similarly, for factual recall we see early learning of the "generic" answer, while the factual answer is learned later. Conceptually, if LLMs are able to write natural text or have been trained sufficiently with natural texts, it is not surprising for the model to predict "the" with high probability after seeing "to". This is verified in Appendix D.1.

Implications from experiments. We summarize our main experimental observations of this section.

Observation 1 *Global associations may "distract" LLMs away from in-context predictions, hurting performance on reasoning tasks.*

Observation 2 *LASER on MLP weights in LLMs helps inhibit predictions of global associations, thus improving in-context predictions.*

Observation 3 *During pre-training, global associations are learned earlier than complex reasoning.*

These observations raise the following questions, which we investigate in the next sections.
Q1: *Why are global associations learned before than complex reasoning?*
Q2: *Are feed-forward layers responsible of learning global associations?*

3. Two-layer Transformer on Noisy In-context Recall

In this section, we consider two-layer transformers on an in-context recall task with added global noise, which allows us to study some key properties observed in Section 2 in a controlled setting. We empirically show how transformers solve this task by storing the noise in feed-forward layers, while attention implements the in-context mechanism. We then provide theory showing why feed-forward layers are more likely to store the global noise association, by studying gradients at initialization.

Setup. Details of data, task and models are provided in Appendix B.1.

Experimental observations are reported in Appendix B.2.

3.1. Theoretical analysis: how and why do feed-forward layers store the noise?

With more details presented in Appendix B.3, we conducted an analysis of a one-step update for the Feed-forward weight $\mathbf{W}_F$ and the value matrix $\mathbf{W}_V$ in attention on a simplified model.

Theorem 1 thus shows that in the initial phase of training, feed-forward layers are more likely to pick up the noise token, while attention will be slower due to additional noise and possibly smaller step-sizes. We may then expect the attention layers to focus instead on learning the induction head mechanism, as we observe empirically. Understanding this trade-off requires studying the dynamics of other attention parameters including key-query matrices, a much more involved endeavor which we leave to future work.

3.2. Theoretical insight: attention avoids attending to noise tokens

When the feed-forward weight learns to predict the noise as shown in Theorem 1, Figure 5 reveals that the second-layer attention in the two-layer model attends only towards the correct tokens. In contrast, a model pre-trained without noise has second-layer attention attend towards all tokens just after the triggers [7], as observed in the attention pattern at the first step in Figure 5(right). Then, after being fine-tuned on noise data, the attention becomes only focused on the correct tokens. Understanding this mechanism requires the analysis of the dynamics of $\mathbf{W}_{KQ}$.

Following the simplified model and data distribution in Section B.3, we take a step towards understanding how attention "avoids" the noise tokens, detailed in Appendix C.2.

4. Discussion and Limitations

In this paper, we studied the questions of how transformer language models learn to process global associations differently than in-context inputs, and how truncating specific weights or layers, particularly feed-forward layers, can help reasoning tasks. While our work provides some initial theoretical understanding of how this may arise on simple controlled settings, our analysis is heavily simplified, and many questions remain open: (i) what are the training dynamics and truncation behaviors in richer data models where there are many more places and ways to choose between in-context and global associations? (ii) in some architectures, with an example reported in Appendix C.4, it appears that global associations are not stored in MLPs, but rather in attention. We believe these are both interesting directions for future work.

References

[1] Jacob Abernethy, Alekh Agarwal, Teodor Vanislavov Marinov, and Manfred K Warmuth. A mechanism for sample-efficient in-context learning for sparse retrieval tasks. In *International Conference on Algorithmic Learning Theory*, 2024.

[2] Ekin Akyürek, Bailin Wang, Yoon Kim, and Jacob Andreas. In-context language learning: Arhitectures and algorithms. *arXiv preprint arXiv:2401.12973*, 2024.

[3] AI Anthropic. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 2024.

[4] Jimmy Ba, Murat A Erdogdu, Taiji Suzuki, Zhichao Wang, Denny Wu, and Greg Yang. High-dimensional asymptotics of feature learning: How one gradient step improves the representation. *Advances in Neural Information Processing Systems*, 2022.

[5] Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *Advances in neural information processing systems*, 2023.

[6] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.

[7] Alberto Bietti, Vivien Cabannes, Diane Bouchacourt, Herve Jegou, and Leon Bottou. Birth of a transformer: A memory viewpoint. *Advances in Neural Information Processing Systems*, 2023.

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020.

[9] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

[10] Vivien Cabannes, Berfin Simsek, and Alberto Bietti. Learning associative memories with gradient descent. *arXiv preprint arXiv:2402.18724*, 2024.

[11] Angelica Chen, Ravid Schwartz-Ziv, Kyunghyun Cho, Matthew L Leavitt, and Naomi Saphra. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in mlms. In *International Conference on Learning Representations*, 2024.

[12] Alexandru Damian, Jason Lee, and Mahdi Soltanolkotabi. Neural networks can learn representations with gradient descent. In *Conference on Learning Theory*, 2022.

[13] Yatin Dandi, Florent Krzakala, Bruno Loureiro, Luca Pesce, and Ludovic Stephan. Learning two-layer neural networks, one (giant) step at a time. *arXiv preprint arXiv:2305.18270*, 2023.

[14] Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, et al. Faith and fate: Limits of transformers on compositionality. *Advances in Neural Information Processing Systems*, 2024.

[15] Benjamin L Edelman, Surbhi Goel, Sham Kakade, and Cyril Zhang. Inductive biases and variable creation in self-attention mechanisms. In *International Conference on Machine Learning*, 2022.

[16] Benjamin L Edelman, Ezra Edelman, Surbhi Goel, Eran Malach, and Nikolaos Tsilivis. The evolution of statistical induction heads: In-context learning markov chains. *arXiv preprint arXiv:2402.11004*, 2024.

[17] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.

[18] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

[19] Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

[20] Samy Jelassi, Michael Sander, and Yuanzhi Li. Vision transformers provably learn spatial structure. In *Advances in Neural Information Processing Systems*, 2022.

[21] Yuchen Li, Yuanzhi Li, and Andrej Risteski. How do transformers learn topic structure: Towards a mechanistic understanding. In *International Conference on Machine Learning*, 2023.

[22] Bingbin Liu, Jordan T Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. In *International Conference on Learning Representations*, 2023.

[23] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 2022.

[24] William Merrill, Ashish Sabharwal, and Noah A Smith. Saturated transformers are constant-depth threshold circuits. *Transactions of the Association for Computational Linguistics*, 10: 843–856, 2022.

[25] Neel Nanda, Lawrence Chan, Tom Liberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *International Conference on Learning Representations*, 2023.

[26] Eshaan Nichani, Alex Damian, and Jason D Lee. How transformers learn causal structure with gradient descent. In *International Conference on Learning Representations*, 2024.

[27] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain,

Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022.

[28] Samet Oymak, Ankit Singh Rawat, Mahdi Soltanolkotabi, and Christos Thrampoulidis. On the role of attention in prompt-tuning. In *International Conference on Machine Learning*, 2023.

[29] Lucia Quirke, Lovis Heindrich, Wes Gurnee, and Neel Nanda. Training dynamics of contextual n-grams in language models. *arXiv preprint arXiv:2311.00863*, 2023.

[30] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *Technical report, OpenAI*, 2019.

[31] Gautam Reddy. The mechanistic basis of data dependence and abrupt learning in an in-context classification task. In *International Conference on Learning Representations*, 2024.

[32] Clayton Sanford, Daniel Hsu, and Matus Telgarsky. Transformers, parallel computation, and logarithmic depth. *arXiv preprint arXiv:2402.09268*, 2024.

[33] Pratyusha Sharma, Jordan T Ash, and Dipendra Misra. The truth is in there: Improving reasoning in language models with layer-selective rank reduction. *arXiv preprint arXiv:2312.13558*, 2023.

[34] Charlie Snell, Ruiqi Zhong, Dan Klein, and Jacob Steinhardt. Approximating how single head attention learns. *arXiv preprint arXiv:2103.07601*, 2021.

[35] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

[36] Sainbayar Sukhbaatar, Edouard Grave, Guillaume Lample, Herve Jegou, and Armand Joulin. Augmenting self-attention with persistent memory. *arXiv preprint arXiv:1907.01470*, 2019.

[37] Yuandong Tian, Yiping Wang, Beidi Chen, and Simon S Du. Scan and snap: Understanding training dynamics and token composition in 1-layer transformer. In *Advances in Neural Information Processing Systems*, 2023.

[38] Yuandong Tian, Yiping Wang, Zhenyu Zhang, Beidi Chen, and Simon Du. Joma: Demystifying multilayer transformers via joint dynamics of mlp and attention. 2024.

[39] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[41] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[42] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

[43] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 2022.

[44] Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. Trained transformers learn linear models in-context. *Journal of Machine Learning Research*, 25(49):1–55, 2024.

[45] Yi Zhang, Arturs Backurs, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, and Tal Wagner. Unveiling transformers with lego: a synthetic reasoning task. *arXiv preprint arXiv:2206.04301*, 2022.

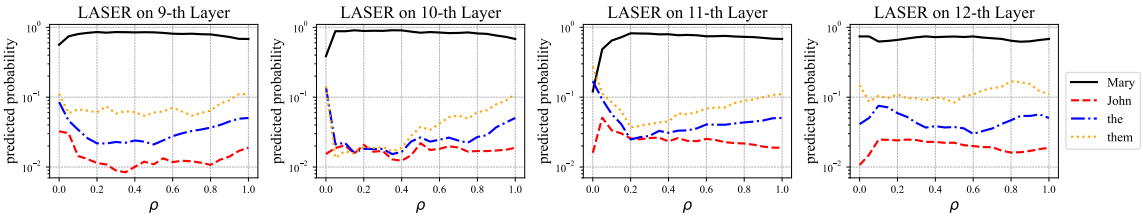

Figure 1: Predicted probability for $c \in \{$"Mary", "them", "the", "John"$\}$. LASER is conducted on input matrices of MLP layers on the layer $l = 9, 10, 11, 12$ of GPT-2 Small. The input is "When Mary and John went to a store, John gave a drink to". The horizontal is the fraction of perserved rank, $\rho \in [0, 1]$, where $\rho = 1$ stands for the full model. It turns out LASER clearly decreases probability of "the" and "them" when $\rho \in [0.1, 0.8]$ for layer $l = 9, 10, 11$, compared with the full model.

Table 1: Probabilities of the top-5 next-tokens in Pythia-1B before and after LASER. The input prompt is "Madrid is located in". Probabilities of two generic words, *i.e.*, "the" and "a", drop sharply after LASER, while probabilities of meaningful words increase, especially the target "Spain".

	"the"	"Spain"	"a"	"southern"	"northern"
Full	**0.499**	0.079	0.069	0.023	0.021
LASER	0.027	**0.300**	0.002	0.044	0.046

Appendix A. Related Work

[33] recently empirically observed that a low-rank approximation of some weights in some pre-trained LLMs can improve reasoning capabilities. Several interpretability works have looked at the role of attention versus feed-forward layers for different tasks. The prominence of feed-forward/MLP layers for storing "global" or "persistent" associations or facts has been observed in [18, 19, 23, 36]. In contrast, several works have investigated the role of attention heads for "reasoning" or computation over the context, *e.g.*, for simple copying mechanisms with so-called induction heads [7, 17, 27], or for more complex tasks [22, 24, 32, 42, 45].

Training dynamics of transformers and attention have been studied in various works [7, 16, 20, 21, 26, 28, 31, 34, 37, 38, 44]. In particular, the two-layer model and copy task we consider are similar to Bietti et al. [7], yet their data model does not involve noise on in-context predictions, and they do not study learning of global associations. Reddy [31] study in-context vs. in-weights learning empirically, on a different task than ours. Cabannes et al. [10] study training dynamics of linear associative memories, but focuses on deterministic data while our setup has noise. Training dynamics were also studied empirically for interpretability [11, 25, 27, 29]. Abernethy et al. [1], Bai et al. [5], Edelman et al. [15] studied sample complexity of self-attention and in-context learning operations, but did not consider training dynamics.

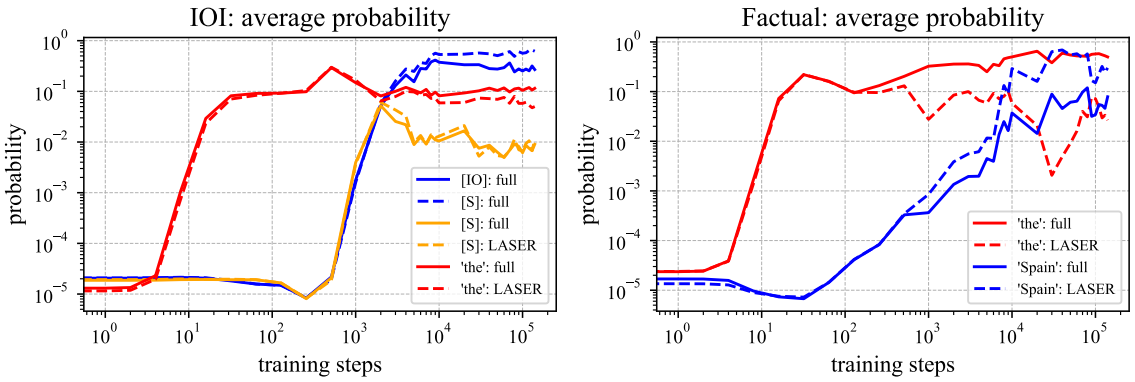

Figure 2: **Left**: average probability of tokens [IO], [S] and "the" in IOI task in the prediction by Pythia-1B along training. **Right**: average probability of tokens "Spain" and "the" in a factual task predicted by Pythia-1B along training, with input as "Madrid is located in". In both tasks, the full model learns to predict "the" with high probability starting from $\sim$10 steps, and then learns to solve the tasks. LASER boosts the probability of correct answers against "the" in both tasks: the average probability ratio of correct answers against "the" improves from 2.3$\times$ to 12.3$\times$ (in IOI) and from 0.16$\times$ to 11.3$\times$ (in factual) at 14K steps.

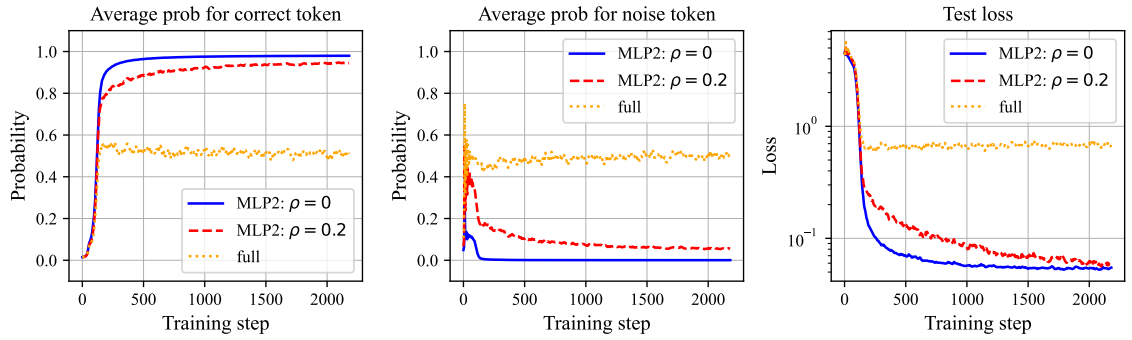

Figure 3: Average probability of predicting correct and noise tokens, and test loss on clean data ($\alpha = 0$), with different fractions ρ of preserved rank in U_{in} of the second-layer MLP F_2. The full model learns to predict noise with probability around $\alpha = 0.5$, as expected from training data. When F_2 is dropped ($\rho = 0$), the model predicts the correct token $\bar{y}$ with probability ≈ 0.98.

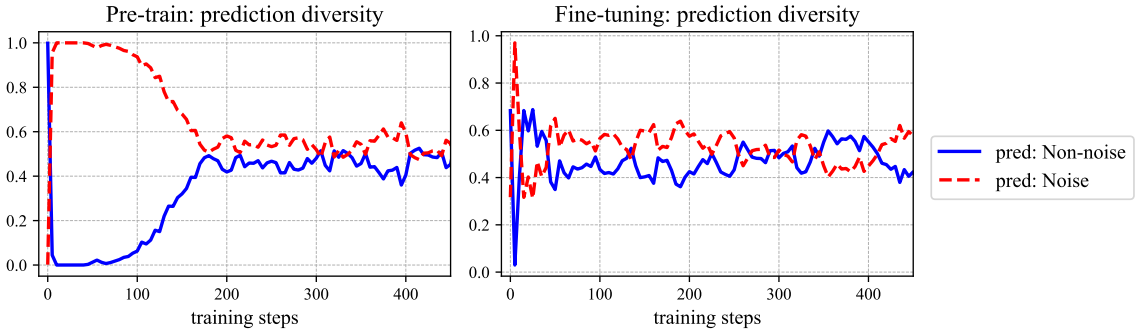

Figure 4: Fractions of predicting the noise token and the other non-noise tokens with $\alpha = 0.5$. (Left) pretraining steps on noisy data; (right) finetuning steps on noisy data, after pretraining on clean data with $\alpha = 1$. In both cases, the models learn to predict noise with probability nearly 0.5. In the first few (~ 5) steps, the models quickly learn to predict noise with probability close to 1.

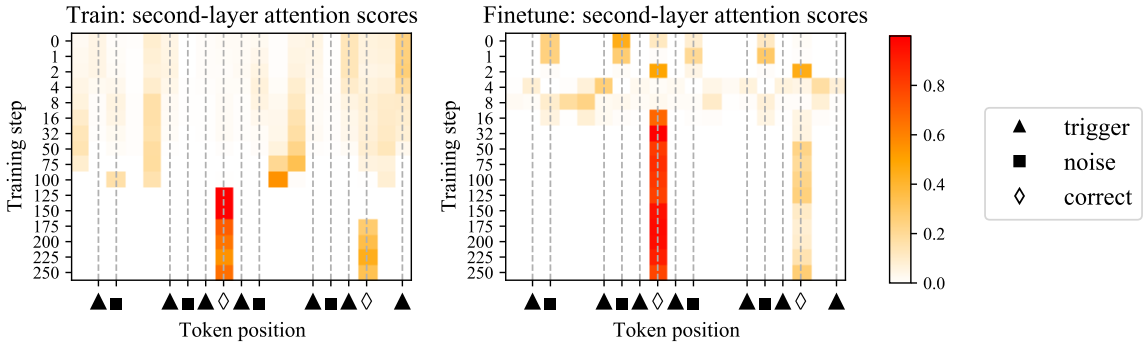

Figure 5: The second-layer attention scores of models trained with noise (left), fine-tuned with noise (right, initialized as a model pre-trained without noise), given the same input. It turns out both models learn to attend to the informative structure "[trigger]+$\bar{y}$" instead of "[trigger]+noise". This implies that the attention in these models is only responsible to predict $\bar{y}$, although the training input and output have noise with probability $\alpha = \Theta(1)$.

Appendix B. Two-layer Transformer solving noisy in-context recall: setting and results

B.1. Setup

Data and task. The data model we consider is similar to Bietti et al. [7], with additional noise. Consider a vocabulary $\mathcal{V} = \{1, 2, \ldots, N, N+1\}$. The token $N+1$ is the noise token. We fix a *trigger* token $q \in [N]$, which governs in-context recall, and a context length T. Each sequence of tokens $z_{1:T} = [z_1, z_2, \ldots, z_T]$ is generated as follows:

i. Sample a correct *output* token $\bar{y}$ uniformly in $[N]$.

ii. Sample $z_{1:T-1}$ according to the following Markov process (π_u, π_b are distributions on $[N]$ defined later): $z_1 \sim \pi_u(\cdot)$, and

$$z_{t+1}|z_t \sim \begin{cases} \pi_b(\cdot|z_t), & \text{if } z_t \neq q, \\ p_{\alpha,\bar{y}}(\cdot), & \text{otherwise,} \end{cases}$$

$$p_{\alpha,\bar{y}}(x) = \begin{cases} 1-\alpha, & \text{if } x = \bar{y}, \\ \alpha, & \text{if } x = N+1, \\ 0, & \text{otherwise.} \end{cases}$$

iii. Set $z_T = q$, and sample the final output $y = z_{T+1} \sim p_{\alpha,\bar{y}}(\cdot)$.

Note that the true $\bar{y}$ varies across sequences, so that the model needs to infer it from context, e.g., using an induction head as in [7]. Predicting $\bar{y}$ may thus be seen as a basic "reasoning" task, yet when training with $\alpha > 0$, the noisy output also requires the model to learn a global trigger-noise association, similar to the "to the" bigram discussed in Section 2. We also consider using multiple trigger tokens in Appendix C.3 and Figure 7.

Following [7], we take π_u and π_b to be the unigram and brigram character-level distributions estimated from the tiny Shakespeare dataset with $N = 65$.

Two-layer transformer. We consider a simplified two-layer transformer formulated on the right. The input is a sequence of tokens $z_{1:T} = [z_1, \ldots, z_T] \in [N+1]^T$, and the output is ξ. The embedding matrix $\mathbf{W}_E \in \mathbb{R}^{(N+1)\times d}$ and un-embedding matrix $\mathbf{W}_E \in \mathbb{R}^{(N+1)\times d}$ are fixed at random initialization. The two attention layers have learnable weights $\mathbf{W}^1_{KQ}, \mathbf{W}^1_V, \mathbf{W}^2_{KQ}, \mathbf{W}^2_V \in \mathbb{R}^{d\times d}$ with $\sigma(\cdot)$ the softmax on a vector. The two feed-forward layers F_1, F_2 are also learnable, and typically we set them as two-layer MLPs with ReLU activation. We will discuss different architectural choices of F_1, F_2 in Appendix C.4. We use the cross-entropy loss to predict $y = z_{T+1}$ from the logits $\xi_T \in \mathbb{R}^{N+1}$.

$$\begin{aligned} x_t &\triangleq \mathbf{W}_E(z_t) + p_t, \\ h^1_t &\triangleq \sum_{s\le t} \left[\sigma(x_t^\top \mathbf{W}^1_{KQ} x_{1:t})\right]_s \cdot \mathbf{W}^1_V x_s, \\ x^1_t &\triangleq x_t + h^1_t + F_1(x_t + h^1_t), \\ h^2_t &\triangleq \sum_{s\le t} \left[\sigma({x^1_t}^\top \mathbf{W}^2_{KQ} x^1_{1:t})\right]_s \cdot \mathbf{W}^2_V x^1_s, \\ x^2_t &\triangleq x^1_t + h^2_t + F_2(x^1_t + h^2_t), \\ \xi_t &\triangleq \mathbf{W}_U x^2_t. \end{aligned}$$

The model setup includes $d = 256$ and two-layer MLPs with ReLU for both F_1, F_2. The training setup includes batch size as 512 and the context length $T = 256$. When evaluating trained models, we consider LASER on the input weight U_{in} of F_2.

B.2. Experimental observations

We consider a noise level $\alpha = 0.5$ for training data (though any other constant value would lead to similar observations). During test time, we set $\alpha = 0$ to compute the test loss, aiming to measure how likely the (full or after-LASER) model predicts the ground-truth $\bar{y}$.

Experimental results are reported in Figure 3 and 4. The full model predicts noise with probability close to α, which is expected since it is trained to predict the noise token w.p. α. However, when dropping the second-layer MLP F_2, the truncated model predicts the ground-truth $\bar{y}$ with an almost perfect probability ≈ 0.98. This suggests that F_2 is responsible for storing the global association "[trigger] + [noise]". Another observation is that the full model first learns to predict the noise with high probability in very early steps, after which it starts learning to predict the correct $\bar{y}$, which resembles the dynamics observed for learning the "to the" bigram in Pythia models in Figure 2. This suggests that learning the (global) trigger-noise association is easier than predicting $\bar{y}$, and we will study this theoretically in Section B.3.

After the global noise association is learned, we observe a slower learning of an induction head mechanism, with similar dynamics to Bietti et al. [7]. Compared to Bietti et al. [7], we notice that the induction head (i.e., the second layer attention head) filters out the noise tokens and only attends to non-noisy output tokens following the trigger, corresponding to the correct $\bar{y}$, as shown in Figure 5. We present primitive exploration into this mechanism in Section B.4. Appendix C.1 summarizes roles of all components in the two-layer transformer in this task.

B.3. Theoretical analysis: how and why do feed-forward layers store the noise?

As we saw in Figure 3 and 4, the model very quickly learns to predict the noise token after a few steps. Then the gap between $\rho = 0$ and 1 in Figure 3 suggests that the feed-forward layer F_2 is responsible for storing the global association about noise, which is verified in Figure 6 (middle). We now provide theoretical justification for this behavior. Understanding the full dynamics of the model used in our experiments is out of the scope of the present paper, due to the many moving parts and the complexity of non-linear MLPs. Instead, we focus on a simpler model involving one linear feed-forward layer and one attention layer, and look at the gradient dynamics near initialization. In particular, we will show that the gradients over the feed-forward parameters are much more informative than the attention gradient, which is dominated by noise unless the sample size is very large. This shows that the feed-forward layer is much more likely to capture the global association.

Simplified architecture and data. Consider the input $x_t \in \mathbb{R}^d$ at position t defined as $x_t \triangleq \mathbf{W}_E(z_t)$, where $z_t \in [N+1]$ is the token at position t and $\mathbf{W}_E(\cdot)$ returns its (untrained) embedding. Here we ignore positional encoding for simplicity as it carries little signal at initialization, noting it could be easily incorporated. For data generation, π_u and π_b are uniform distributions on $[N]$.

Given a sequence of inputs, $x_{1:T} \in \mathbb{R}^{T\times d}$, the output of model is $\xi \triangleq \xi_{\text{attn}} + \xi_{\text{ff}}$ as

$$\begin{aligned} \xi_{\text{attn}}(x_{1:T}) &\triangleq \mathbf{W}_U \phi(x_T, x_{1:T}) \in \mathbb{R}^{N+1}, \\ \xi_{\text{ff}}(x_{1:T}) &\triangleq \mathbf{W}_U F(x_T) = \mathbf{W}_U \mathbf{W}_F x_T \in \mathbb{R}^{N+1}, \\ \phi(x_T, x_{1:T}) &\triangleq \sum_{t\le T} \left[\sigma\left(x_T^\top \mathbf{W}_{KQ} x_{1:T}\right)\right]_t \cdot \mathbf{W}_V x_t \in \mathbb{R}^d, \end{aligned} \tag{1}$$

where $\mathbf{W}_U \in \mathbb{R}^{(N+1)\times d}$ is the unembedding matrix, $\phi(s,t)$ is the attention module with query s and context t, and $F(\cdot)$ is a linear feed-forward layer. This architecture is similar to a one-layer transformer, but already highlights the difference between feed-forward and attention layers in a way that we expect to still hold for more layers. In the above parametrization, the learnable matrices are $\mathbf{W}_{KQ}, \mathbf{W}_F, \mathbf{W}_V \in \mathbb{R}^{d\times d}$. At initialization, we set $\mathbf{W}_{KQ}, \mathbf{W}_F, \mathbf{W}_V = 0$, noting that random initialization in high dimension would lead to similar behaviors thanks to near-orthogonality. Hence

we assume all embeddings follow Assumption F.1. We now look at the first gradient step from initialization, which has commonly been used to understand feature learning and sample complexity in neural networks [4, 7, 12, 13, 28]. Note that $\mathbf{W}_{KQ}$ has no gradient at initialization, so that the gradient of W_V is most relevant initially [see also 7, 21, 28, 34].

Theorem 1 (Logits after one gradient step) *Assume $N, T \gg 1, \alpha = \Theta(1)$. Consider a one gradient step update from zero-initialization on m i.i.d. samples of $z_{1:T}$ with separate learning rates η_f for $\mathbf{W}_F$ and η_v for $\mathbf{W}_V$ (note that the gradient on $\mathbf{W}_{KQ}$ is zero). With probability $1-\delta$, the resulting logits for the feed-forward and attention blocks satisfy, for any test sequence $z_{1:T}$,*

$$\left|\Delta(\xi_{\mathit{ff}}(x_{1:T})) - \eta_f \cdot \alpha\right| \le \eta_f \cdot O\left(\sqrt{\frac{\ln \frac{2(N+1)}{\delta}}{m}}\right),$$

$$\left|\Delta(\xi_{\mathit{atn}}(x_{1:T})) - \frac{\eta_v}{N} \cdot \hat{\alpha}\right| \le \eta_v \cdot O\left(\sqrt{\frac{(\frac{1}{TN} + \frac{1}{N^2}) \ln \frac{2(N+1)}{\delta}}{m}} + \frac{\ln \frac{2(N+1)}{\delta}}{m}\right),$$

where $\Delta(\xi) = \xi_{N+1} - \max_{j \in [N]} \xi_j$ is the margin of predicting the noise token and $\hat{\alpha} = (\alpha^2 \hat{q} + \alpha(1-\hat{q}))$, where $\hat{q} = \frac{1}{T}\sum_{t \le T} \mathbb{1}\{z_t = N+1\}$ is the fraction of noise tokens in $z_{1:T}$.

The margin $\Delta(\xi)$ reflects how much signal there is in the logits for predicting the noise token, and the theorem provides concentration bounds on the contributions of the updates on $\mathbf{W}_F$ and $\mathbf{W}_V$ to the margin. Note that $\hat{q} \ll 1$ w.h.p. for large N, T, so $\hat{\alpha} \approx \alpha$. We make the following observations:

i. When $m = \tilde{\Omega}(1)$, there is enough signal in $\mathbf{W}_F$ to predict the noise, say with $\eta_f = 1$, and a choice of $\eta_v = O(1)$ will lead to a small but controlled contribution to the prediction from $\mathbf{W}_V$.

ii. When $m = \tilde{\Omega}(N)$, $\mathbf{W}_V$ can also reliably predict the noise by setting $\eta_v = \Theta(N)$ (i.e., with small deviation on the r.h.s.), at the cost of many more samples.

Our result thus shows that in the initial phase of training, feed-forward layers are more likely to pick up the noise token, while attention will be slower due to additional noise and possibly smaller step-sizes. We may then expect the attention layers to focus instead on learning the induction head mechanism, as we observe empirically. Understanding this trade-off requires studying the dynamics of other attention parameters including key-query matrices, a much more involved endeavor which we leave to future work.

B.4. Theoretical insight: attention avoids attending to noise tokens

When the feed-forward weight learns to predict the noise as shown in Theorem 1, Figure 5 reveals that the second-layer attention in the two-layer model attends only towards the correct tokens. In contrast, a model pre-trained without noise has second-layer attention attend towards all tokens just after the triggers [7], as observed in the attention pattern at the first step in Figure 5(right). Then, after being fine-tuned on noise data, the attention becomes only focused on the correct tokens. Understanding this mechanism requires the analysis of the dynamics of $\mathbf{W}_{KQ}$.

Following the simplified model and data distribution in Section B.3, we take a step towards understanding how attention "avoids" the noise tokens, detailed in Appendix C.2. Concretely, this mechanism appears because, after the initial training phase, $\mathbf{W}_V$ has a minor structure that has

a smaller projection onto $\mathbf{W}_U(N+1)\mathbf{W}_E(N+1)^\top$ as in Table 2, which makes $\mathbf{W}_{KQ}$ move negative in the direction of $\mathbf{W}_E(N+1)\mathbf{W}_E(q)^\top$. A more detailed analysis of the dynamics of $\mathbf{W}_{KQ}$ throughout the training process would be an interesting avenue for future work.

Appendix C. How Does the Two-layer Model Solve Noisy In-context Recall?

C.1. Summarizing: roles of key components in the two-layer transformer

Recall the architecture of two-layer transformers in Section 3 as

$$\begin{aligned}
x_t &\triangleq \mathbf{W}_E(z_t) + p_t, \\
h_t^1 &\triangleq \sum_{s\le t} \left[\sigma(x_t^\top \mathbf{W}_{KQ}^1 x_{1:t})\right]_s \cdot \mathbf{W}_V^1 x_s, \\
x_t^1 &\triangleq x_t + h_t^1 + F_1(x_t + h_t^1), \\
h_t^2 &\triangleq \sum_{s\le t} \left[\sigma({x_t^1}^\top \mathbf{W}_{KQ}^2 x_{1:t}^1)\right]_s \cdot \mathbf{W}_V^2 x_s^1, \\
x_t^2 &\triangleq x_t^1 + h_t^2 + F_2(x_t^1 + h_t^2), \\
\xi_t &\triangleq \mathbf{W}_U x_t^2.
\end{aligned}$$

When the task is without noise, *i.e.*, $\alpha = 0$, [7] point out the first-layer attention attends to the previous token through $\mathbf{W}_{KQ}^1 = \sum_{t=2}^T p_{t-1}p_t^\top$. Therefore, when $z_t = \bar{y}$ with $z_{t-1} = q$, the output of the first layer is $x_t^1 \approx \mathbf{W}_E(\bar{y}) + \mathbf{W}_V^1\mathbf{W}_E(q)$. Then they show that the second-layer attention matches such x_t^1 with $z_T = q$ by $\mathbf{W}_{KQ}^2 = (\mathbf{W}_V\mathbf{W}_E(q))\mathbf{W}_E(q)^\top$, through which the information of $\bar{y}$ in x_t^1 is copied to last token as $h_T^2 \approx \mathbf{W}_V^2\mathbf{W}_E(\bar{y})$. Finally $\mathbf{W}_V^2 = \sum_{z\in[N]} \mathbf{W}_U(z)\mathbf{W}_E(z)^\top$ helps output the correct label of $\bar{y}$.

In our work with noise $\alpha > 0$, the key difference is that there is a fixed probability α for a noise token $N+1$ to appear after each trigger q. This requires $\mathbf{W}_{KQ}^2$ to not only match the trigger but also avoid the noise token after trigger. Let's first summarize the whole pipeline of this model for our task.

Roles of key components. The first layer will be basically the same as [7], where $\mathbf{W}_{KQ}^1 = \sum_{t=2}^T p_{t-1}p_t^\top$ attends to the previous token. Consider two positions t_1, t_2 with $z_{t_1-1} = z_{t_2-1} = q, z_{t_1} = \bar{y}, z_{t_2} = N+1$, then outputs of the first layer at these two positions are $x_{t_1}^1 \approx \mathbf{W}_E(\bar{y}) + \mathbf{W}_V^1\mathbf{W}_E(q)$, $x_{t_2}^1 \approx \mathbf{W}_E(N+1) + \mathbf{W}_V^1\mathbf{W}_E(q)$. Then the second-layer attention $\mathbf{W}_{KQ} = (\mathbf{W}_V\mathbf{W}_E(q) - c\cdot\mathbf{W}_E(N+1))\mathbf{W}_E(q)^\top$ with some positive c makes the attention attend to t_1 and avoid t_2 simultaneously, matching with the last token $z_T = q$. Therefore, the output of the second-layer attention at T is basically $h_T^2 \approx \mathbf{W}_V^2\mathbf{W}_E(\bar{y})$. Similar to the noiseless case, $\mathbf{W}_V^2 = \sum_{z\in[N]} \mathbf{W}_U(z)\mathbf{W}_E(z)^\top$ helps output the correct label of $\bar{y}$. Meanwhile, note that x_T^1 actually contains $\mathbf{W}_E(q)$ through x_T, so F_2 is able to predict the noise $N+1$ when seeing a fixed $\mathbf{W}_E(q)$. As a result, combining the two streams from h_T^2 and $F_2(x_T^1)$, the full model is able to predict any $\bar{y}$ w.p. $1-\alpha$ and predict the noise $N+1$ w.p. α.

Evidence. Figure 5 illustrates that the second-layer attention learns to attend to $z_{t_1} = \bar{y}$ and avoid $z_{t_2} = N+1$, with Appendix C.2 presenting a primitive exploration on how the avoidance is learnt in a simplified setting. Figure 6 (left) shows the attention pattern from $\mathbf{W}_{KQ}^1$ of attending to the previous token. Figure 6 (middle) shows the memory recall of $\mathbf{W}_U(N+1)^\top F_2(\mathbf{W}_E(q))$ to

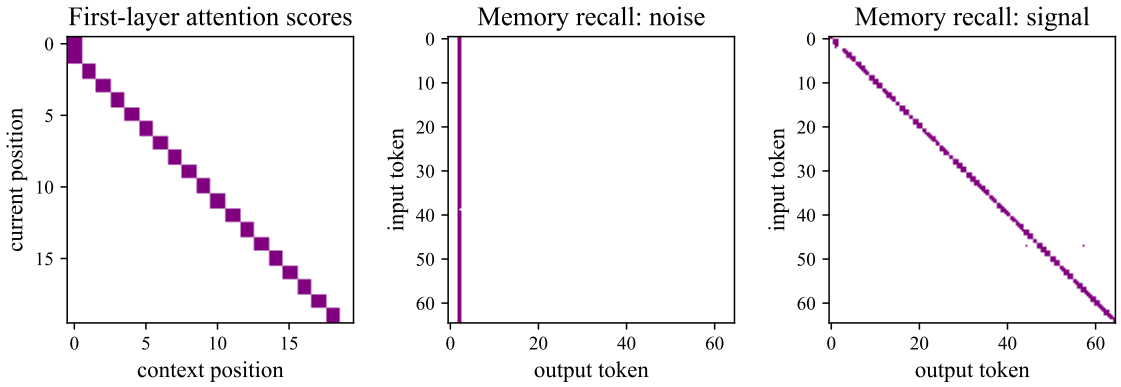

Figure 6: **Left**: first-layer attention attending to the previous token from the current token. **Middle**: logits to predict noise from $\langle F_2(\mathbf{W}_E(i)), \mathbf{W}_U(j)\rangle$ with input $i \in [N+1]$ and output $j \in [N+1]$, where the output channel 2 is set as the noise channel. It turns out, for all input i, the logits on output 2 are large, which matches our construction that, at least for trigger q as input, the output 2 has large logits. **Right**: logits to predict singal from $\langle \mathbf{W}_V^2 \mathbf{W}_E(i), \mathbf{W}_U(j)\rangle$ for input $i \in [N+1]$ and output $j \in [N+1]$. It matches our construction that $i = j$ has large logits. Meanwhile, $i = j = 2$ does not have large logits since 2 is the noise channel.

predict the noise. Figure 6 (right) illustrates the memory recall of $\mathbf{W}_U(i)^\top \mathbf{W}_V^2 \mathbf{W}_E(i)$ to predict the correct token.

C.2. How does attention attend less towards the noise token?

We use the same simplified model as in Section B.3 to understand how the second-layer attention learns to avoid the noise. When using the same learning rate $\eta = \eta_v = \eta_f$, Theorem 1 implies that the feed-forward $\mathbf{W}_F$ makes the most contribution for predicting the noise after the first-step update. Denote the logits for the noise of the model at time t as ξ_t. The arguments in this section make the following assumptions, which hold at least after the first-step update:

i. $\mathbf{W}_F$ dominates the logits ξ_t of predicting the noise token, compared with $\mathbf{W}_V$.

ii. Logits for predicting any $k \le N$ is close to 0, which means the predicted probability p_t is approximately $p_t \approx \frac{\exp(\xi_t)}{N+\exp(\xi_t)}$.

iii. The predicted probability $p_t < \alpha$.

iv. The attention matrix $\mathbf{W}_{KQ}$ is approximately 0, inducing a uniform attention.

v. The dataset has $T, N \gg 1$ and $m \to \infty$, so the gradient is from population loss.

The first assumption holds after the first step from Theorem 1 with $\eta_f = \eta_v$.

The second assumption holds because $|\mathbf{W}_U(k)^\top(\nabla_{\mathbf{W}_F} L)\mathbf{W}_E(q)| = O(\frac{1}{N})\cdot|\mathbf{W}_U(N+1)^\top(\nabla_{\mathbf{W}_F} L)\mathbf{W}_E(q)|$ for any $k \le N$ in Lemma 5. Meanwhile, the projection of $\nabla_{\mathbf{W}_V} L$ onto any direction in Lemma 6 is also smaller than $\mathbf{W}_U(N+1)^\top(\nabla_{\mathbf{W}_F} L)\mathbf{W}_E(q)$ by a factor of $O(1/N)$.

Let's check the condition of the third assumption. In the proof of Lemma 5, the gradient of $\mathbf{W}_F$ has the form of

$$\mathbf{W}_U(N+1)^\top(-\nabla_{\mathbf{W}_F}L)\mathbf{W}_E(q) = \alpha - p_t.$$

This update induces ξ_t to increase by $\eta(\alpha - p_t)$. This implies

$$\xi_t \approx \xi_{t-1} + \eta\left(\alpha - \frac{\exp(\xi_t)}{N + \exp(\xi_t)}\right), \ \ \forall\, t \geq 1.$$

This sequence $\{\xi_t\}_{t\geq 1}$ has stationary point $\xi^* = \log N + \log(\frac{\alpha}{1-\alpha})$. Denoting $\hat{\xi}_t \triangleq \xi_t - \xi^*$ with $\hat{\xi}_1 = -\xi^* < 0$, the iteration becomes

$$\hat{\xi}_{t+1} \approx \hat{\xi}_t + \eta\left(\alpha - \frac{\exp(\hat{\xi}_t)}{\frac{1-\alpha}{\alpha} + \exp(\hat{\xi}_t)}\right).$$

If we would like to have $\hat{\xi}_t$ not hit the positive region by controlling η, it suffices to bound η with any $\hat{\xi} < 0$,

$$\eta \leq \frac{\hat{\xi}}{\frac{\exp(\hat{\xi})}{\frac{1-\alpha}{\alpha}+\exp(\hat{\xi})} - \alpha},$$

where RHS is continuous and decreasing on $\xi < 0$ when $\alpha < 0.5$. Hence, we have $\eta \leq \frac{1}{\alpha(1-\alpha)}$ evaluated at $\hat{\xi} = 0$ by L'Hospital rule. This bound of η is very strong, since $\eta = O(\log N)$ can still have $\hat{\xi} < 0$ after one step.

The fourth assumption is basically from what we will show at the end of this section, as the second observation.

Then consider the dynamics of $\mathbf{W}_V$, which is much slower than $\mathbf{W}_F$. From the proof of Lemma 6, the gradient of $\mathbf{W}_V$ satisfies

$$\begin{aligned}
\nabla_{\mathbf{W}_V}L &= \mathbb{E}_x\left[\sum_{k=1}^{N+1}(p_{\mathbf{W}}(k|x) - \mathbb{1}\{y=k\})\mathbf{W}_U(k)\left(\frac{1}{T}\sum_{t=1}^{t}x_t\right)^\top\right], \\
\mathbf{W}_U(N+1)^\top(-\nabla_{\mathbf{W}_V}L)\mathbf{W}_E(k) &\approx \frac{1}{N}\sum_{t\geq 1}(\alpha - p_t)(\mathbb{1}\{k\leq N\} + \alpha\cdot\mathbb{1}\{k = N+1\}) \\
&\triangleq c\cdot\mathbb{1}\{k\leq N\} + c\cdot\alpha\cdot\mathbb{1}\{k=N+1\} = \Theta(\frac{1}{N}),
\end{aligned} \tag{2}$$

where the projection on $W_E(N+1)$ is always positive and smaller than that on other directions when $p_t < \alpha$. Projections onto other directions $\mathbf{W}_U(j)\mathbf{W}_E(k)^\top, \forall\, j \leq N$, are smaller as $\Theta(\frac{1}{N^2})$.

Finally, let's consider the dynamics of $\mathbf{W}_{KQ}$. At initialization, $\mathbf{W}_{KQ} = 0$ and $\nabla_{\mathbf{W}_{KQ}}L = 0$ due to zero initialization of $\mathbf{W}_V$. After one-step, $\mathbf{W}_V$ has such a structure in Eq.(2). Then, with

$\bar{x}_{1:T} \triangleq \frac{1}{T}\sum_{1\le t\le T} x_t$ from uniform attention, the gradient of $\mathbf{W}_{KQ}$ satisfies

$$\begin{aligned}
-\nabla_{\mathbf{W}_{KQ}} L &= \mathbb{E}_x\left[\sum_{k=1}^{N}(\mathbb{1}\{y=k\}-p_{\mathbf{W}}(k|x))\frac{1}{T}\sum_{t=1}^{T}(\mathbf{W}_U(k)^\top \mathbf{W}_V x_t)\cdot(x_t-\bar{x}_{1:T})\mathbf{W}_E(q)^\top\right] \\
&\approx \sum_{k=1}^{N}\left(\frac{1-\alpha}{N}-\frac{1-p_t}{N}\right)\underbrace{\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\mathbf{W}_U(k)^\top\mathbf{W}_V x_t\cdot(x_t-\bar{x}_{1:T})\mathbf{W}_E(q)^\top\right]}_{\triangleq A} \\
&\quad+(\alpha-p_t)\underbrace{\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}(\mathbf{W}_U(N+1)^\top\mathbf{W}_V x_t)\cdot(x_t-\bar{x}_{1:T})\mathbf{W}_E(q)^\top\right]}_{\triangleq B}.
\end{aligned} \tag{3}$$

Then, we have

$$\begin{aligned}
\mathbf{W}_E(N+1)^\top B\mathbf{W}_E(q) &= \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}(\mathbf{W}_U(N+1)^\top\mathbf{W}_V x_t)\cdot\mathbf{W}_E(N+1)^\top(x_t-\bar{x}_{1:T})\right] \\
&\overset{(a)}{=} \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}(c+c(\alpha-1)\cdot\mathbb{1}\{z_t=N+1\})\cdot\mathbf{W}_E(N+1)^\top(x_t-\bar{x}_{1:T})\right] \\
&\overset{(b)}{=} \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}(c(\alpha-1)\cdot\mathbb{1}\{z_t=N+1\})\cdot\mathbf{W}_E(N+1)^\top(x_t-\bar{x}_{1:T})\right] \\
&= \frac{\alpha}{N}\cdot c(\alpha-1)(1-\frac{\alpha}{N}) = \Theta(\frac{1}{N^2}) < 0.
\end{aligned}$$

where (a) is from Eq.(2), (b) is due to $\bar{x}_{1:T} = \frac{1}{T}\sum_t x_t$ and note that $c = \Theta(\frac{1}{N})$.

Similarly, we also have

$$\begin{aligned}
\mathbf{W}_E(N+1)^\top A\mathbf{W}_E(q) &= \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}(\mathbf{W}_U(k)^\top\mathbf{W}_V x_t)\mathbf{W}_E(N+1)^\top\cdot(x_t-\bar{x}_{1:T})\right] \\
&= \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\Theta(\frac{1}{N^2})\cdot\mathbb{1}\{z_t=N+1\}\mathbf{W}_E(N+1)^\top\cdot(x_t-\bar{x}_{1:T})\right] = \Theta(\frac{1}{N^3}).
\end{aligned}$$

For any $k \le N$, we have

$$\begin{aligned}
\mathbf{W}_E(k)^\top B\mathbf{W}_E(q) &= \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}(\mathbf{W}_U(N+1)^\top\mathbf{W}_V x_t)\cdot\mathbf{W}_E(k)^\top(x_t-\bar{x}_{1:T})\right] \\
&= \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}(c(\alpha-1)\cdot\mathbb{1}\{z_t=k\})\cdot\mathbf{W}_E(N+1)^\top(x_t-\bar{x}_{1:T})\right] \\
&= \frac{\alpha}{N}\cdot c(\alpha-1)(-\frac{1}{N}) = \Theta(\frac{1}{N^3}) > 0,
\end{aligned}$$

and

$$\mathbf{W}_E(k)^\top A \mathbf{W}_E(q) = \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}(\mathbf{W}_U(k)^\top \mathbf{W}_V x_t)\mathbf{W}_E(k)^\top \cdot (x_t - \bar{x}_{1:T})\right]$$
$$= \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\Theta(\frac{1}{N^2})\cdot \mathbb{1}\{z_t = N+1\}\mathbf{W}_E(k)^\top \cdot (x_t - \bar{x}_{1:T})\right] = \Theta(\frac{1}{N^4}).$$

Combining the above four esimation of projections of A and B with Eq.(3), we have

$$\mathbf{W}_E(N+1)^\top(-\nabla_{\mathbf{W}_{KQ}}L)\mathbf{W}_E(q) = \Theta(\frac{1}{N^2}) < 0,$$
$$\forall\, k \le N,\ \ \mathbf{W}_E(k)^\top(-\nabla_{\mathbf{W}_{KQ}}L)\mathbf{W}_E(q) = \Theta(\frac{1}{N^3}) > 0.$$

Then we have three observations

i. $\mathbf{W}_{KQ}$ in this phase avoids the noise token $N+1$ and uniformly attends to all tokens $k \le N$.

ii. The update of $\mathbf{W}_{KQ}$ is in $\Theta(\frac{1}{N^2})$, while the update of $\mathbf{W}_F$ is $\Theta(1)$ in Lemma 5 and that of $\mathbf{W}_V$ is $\Theta(\frac{1}{N})$ in Lemma 6. These three levels of updating speed also coincide with the assumptions that $\mathbf{W}_F$ dominates first and then $\mathbf{W}_V$ has a micro structure that induces the evolving of $\mathbf{W}_{KQ}$.

iii. The current proof for $\mathbf{W}_{KQ}$ strongly depends on the fact that the noise token appears less than other token by a factor α in expectation. The proof will have the opposite result if the noise token is made to appear more by manipulating the data distribution. Therefore, we leave a new proof that is robust to such an assumption in data distribution as future work.

C.3. Multiple Triggers

In Section 3, we assume there is only one fixed trigger $q \in [N]$ for simplicity. Actually the case of multiple triggers has the same mechanism. As discussed by [7] and Appendix C.1, for one trigger, the second-layer attention has large logits in $\langle \mathbf{W}_V^1 \mathbf{W}_E(i)^\top, \mathbf{W}_{KQ}^2 \mathbf{W}_E(j)\rangle$ only for $i = j = q$. For multiple triggers, basically $\langle \mathbf{W}_V^1 \mathbf{W}_E(i)^\top, \mathbf{W}_{KQ}^2 \mathbf{W}_E(j)\rangle$ only have large values when $q \in Q$. This is verified in Figure 7.

C.4. Architectural Choices

In Section 3 and Appendix C.1, we were focused on experiments with both F_1, F_2 being two-layer ReLU MLPs. Meanwhile, we have also tried other choices of F_1, F_2 and then search for the best truncation method for each architecture. In this section, we would like to summarize our experimental results for better understanding of all modules in the two-layer transformer.

Generally, the feed-forward layer can be two-layer ReLU MLPs, one-layer Linear or "None", where None stands for there is no feed-forward layer so that the value matrices in attention layers are the only weight matrices that transform features.

Both F_1, F_2 are two-layer MLPs. This is our main setting. The best truncation method is to *fully* drop F_2. We also try to fully drop F_1, as reported in Figure 8. It turns out fully dropping F_1 makes the model predict the noise with high probability.

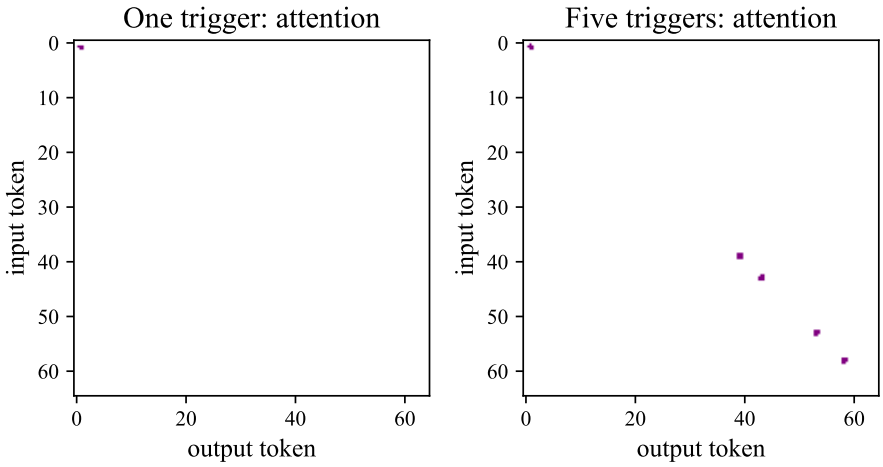

Figure 7: Logits of $\langle \mathbf{W}_V^1 \mathbf{W}_E(i)^\top, \mathbf{W}_{KQ}^2 \mathbf{W}_E(j) \rangle$ for input i and output j when there is one trigger (left, $q = 1$) and five triggers (right, $q \in Q = \{1, 39, 43, 53, 58\}$). In both cases, the logits only have large values when $i = j = q$, verifies the matching mechanism in Appendix C.1.

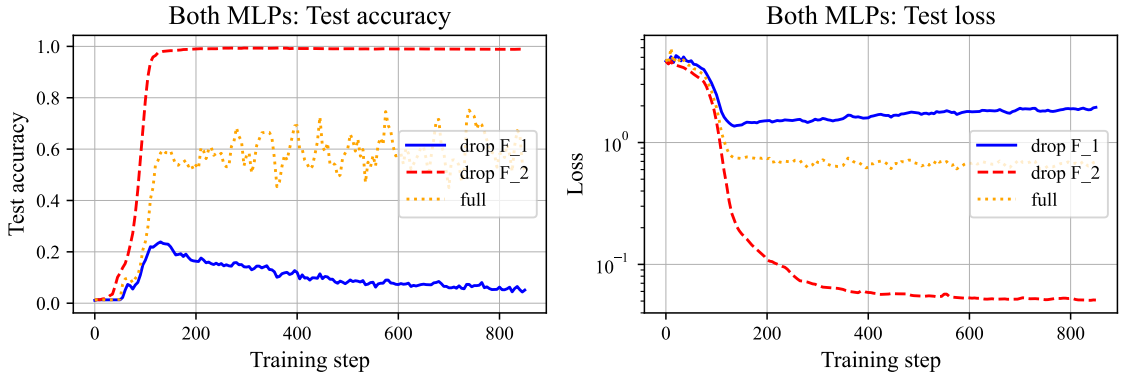

Figure 8: Test performance of fully dropping F_1, F_2 when both F_1, F_2 are two-layer MLPs. It turns out, while dropping F_2 makes the model predict correctly w.p. near 1, dropping F_1 has the model predict noise with high probability.

F_1 is MLPs and F_2 is Linear. Figure 9 reports the results. Dropping F_1 and F_2 both improve the correct prediction, and dropping F_1 is better with lower test loss. Note that, when test accuracies are near 100%, lower test loss is a better measurement of the prediction quality, because accuracies are taken by argmax over the output logits while test loss are about the exactly predicted probability.

F_1 is Linear and F_2 is MLPs. Figure 10 reports the results. Dropping F_2 improves the correct prediction while dropping F_1 makes the model predict noise more.

Both F_1 and F_2 are None. Figure 11 reports the results. While there is no feed-forward layer any more, low-rank truncating a part $\mathbf{W}_O^1$ of the first-layer matrix improves the model's prediction a

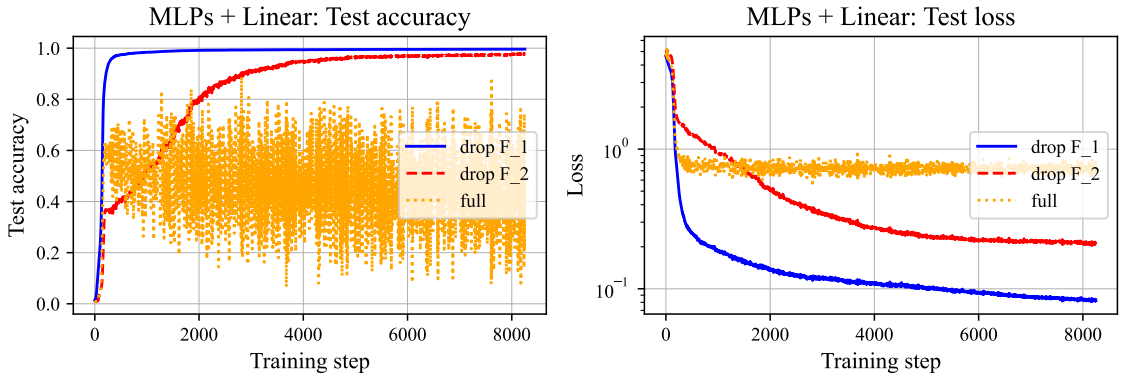

Figure 9: Test performance of fully dropping F_1, F_2 when both F_1 is MLPs and F_2 Linear. Both dropping methods turn out to help predict more correctly than the full model. Meanwhile, dropping the MLP F_1 is better with lower test loss.

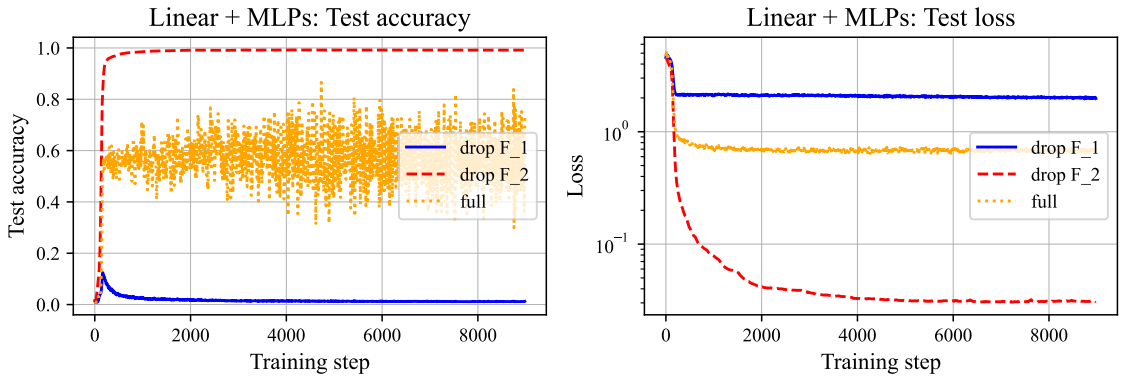

Figure 10: Test performance of fully dropping F_1, F_2 when both F_1 is Linear and F_2 MLPs. Only dropping F_2 helps predict more correctly. Dropping F_1 makes the model predicting noise more.

little. This implies that, when there is not feed-forward layers, the noise association is possible stored in the first-layer value matrix of attention. Note that the improvement of such low-rank truncation is clearly smaller than *fully* dropping one of feed-forward layers in the previous cases. Meanwhile, a smaller $\rho = 0.01$ destroys the model's performance. This implies fully dropping is not the optimal choice for low-rank truncation of the value matrix, and there is low-rank subspace in it that is useful for predicting the correct tokens. Our discussion of the role of $\mathbf{W}_V^1$ in Appendix C.1 is a possible answer to this phenomena.

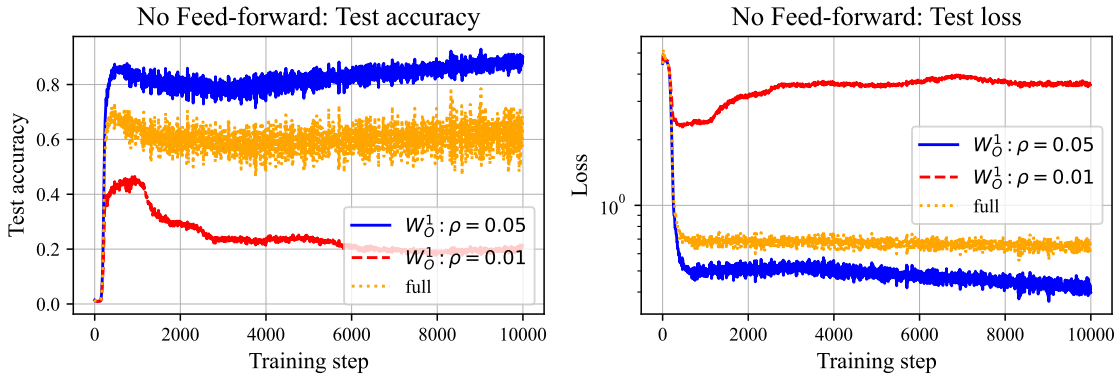

Figure 11: Test performance of low-rank truncating of $\mathbf{W}_O^1$ when there is no F_1, F_2. Here ρ is the fraction of preserved rank of $\mathbf{W}_O^1$, where actually we re-parametrize the first-layer value matrix in attention as $\mathbf{W}_O^1\mathbf{W}_V^1 \in \mathbb{R}^{d\times d}$. It turns out the best $\rho = 0.05$ improves the model's prediction a little. Meanwhile, a smaller ρ destroys the model's performance.

C.5. Training Details about Experiments

All of the training is with SGD optimization with learning rate in $\{0.001, 0.03\}$. The batch size is 512. The dimension is 256. The context length is 256. All results in the experiments are stable for any learning rate between 0.001 and 0.03. Each run of experiments is on a single Nvidia Tesla V100 GPU. It takes 3 hours to finish each run for 2K steps, which probably can be optimized a lot since we are tracking a lot of measurement along training, not limited to hundreds of possible truncations at each test time.

Appendix D. More Experiments on Pythia

D.1. Learning Association with Prepositions

We would like to verify our guess about the structure of "to + the" in Pythia in Section 2.3. To make the argument generalizable than IOI dataset, we consider a structure of "[preposition] + the", where [preposition] has a pool of 30 prepositions in English, including "to". The input is a raw "[preposition]" or a random sentence ending with "[preposition]", with some examples in Appendix H.1. For both kinds of inputs, Pythia-160M/410M/1B turns out to learn the structure of "[preposition] + the" around 10 steps, as shown in Figure 12.

D.2. LASER Parameters for Evaluated LLMs

Following the definition of LASER in Section 2.2, we search for the optimal layer, ρ and target weights in Pythia models and GPT-2 Small for each dataset.

IOI on Pythia-1B. The model has 16 layers. The truncation is on the input matrix of MLPs on the 11-th layer with $\rho = 0.008$.

Factual recall on Pythia-1B. The truncation is on the input matrix of MLPs on the 16-th layer with $\rho = 0.0125$.

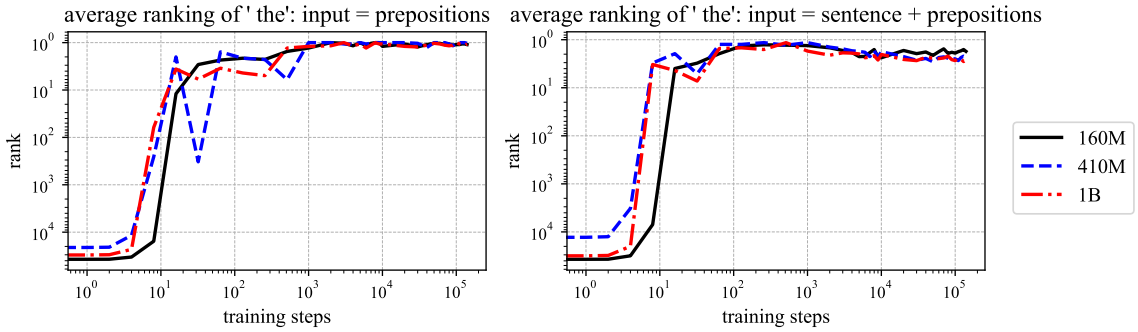

Figure 12: Average ranking of tokens "the" in the prediction by Pythia-160M/410M/1B along training. The inputs are 30 preposition words (left) and 40 sentences ending with prepositions. It turns out "the" becomes one of top predictions around 10 steps.

IOI on GPT2 Small. Related parameters have been contained in Section 2.3.

Appendix E. Linear Associative Memory

E.1. Experiments and Discussions

In Section 3, we showed that *fully* truncating a feed-forward layer can be helpful for reasoning. We now present a setting where noisy associations are stored in a rank-one subspace of a layer, so that *intermediate* levels of truncation are more useful to remove noise.

Model and data. We consider a simple associative memory setting where the goal is learn an fixed permutation from input tokens to output tokens (w.l.o.g. taken to be the identity), with a linear model similar to Cabannes et al. [10]. Consider a learnable weight matrix $\mathbf{W} \in \mathbb{R}^{d\times d}$. Consider embeddings for n input tokens as $\{e_i\}_{i=1}^n \subset \mathbb{R}^d$ and embeddings for c output tokens as $\{u_i\}_{i=1}^c \subset \mathbb{R}^d$. In contrast to Cabannes et al. [10], we consider an additional "common noise" output token $c = n + 1$, which is chosen for any input with probability $\alpha \in (0, 1)$. For any input $x \in [n]$, the target distribution $p_\alpha(\cdot|x)$ is defined by

$$p_\alpha(y|x) = (1-\alpha)\cdot \mathbb{1}\{y = x\} + \alpha \cdot \mathbb{1}\{y = c\}. \tag{4}$$

In other words, the last channel (c) for output is the **common noise** with probability α for any input. The training dataset $\mathcal{D}_\alpha$ consists of uniformly distributed inputs $x \in [n]$, and outputs conditionally sampled as $y|x \sim p_\alpha(\cdot|x)$.

Given any pair of input and output tokens, the associative memory model takes the form

$$f(i, j; \mathbf{W}) \triangleq \langle u_j, \mathbf{W}e_i\rangle, \quad \forall\, i, j \in [n] \times [c], \tag{5}$$

When $k \le d$, we denote the rank-k approximation of f as $f^{(k)}$ by replacing $\mathbf{W}$ with $\mathbf{W}^{(k)}$, where $\mathbf{W}^{(k)}$ is the rank-k approximation of $\mathbf{W}$.

Training. During training, the dataset $\mathcal{D}_\alpha$ is generated with non-zero noise probability $\alpha > 0$. At test time, the dataset $\mathcal{D}_0$ is without noise as $\alpha = 0$, so the computed loss is called **pure-label** loss. The model is trained with Gradient Descent (GD) subjected to cross-entropy loss.

Experiments with randomness. Assume both $\{e_i\}_{i=1}^n$ and $\{u_i\}_{i=1}^c$ are i.i.d. uniformly drawn from sphere $\mathbb{S}^{d-1}$. Also assume the model is initialized as $\mathbf{W}_{i,j} \sim \mathcal{N}(0, \frac{1}{d})$. Due to randomness from embeddings and model initialization, let's first conduct 20 runs of experiments to obtain significant factors before moving the theoretical argument.

Note that *only full models are trained*, and we track loss for low-rank models by conducting SVD in each step without manipulating training. In Figure 5, we illustrate the pure-label loss *v.s.* training steps for models of different ranks, where $n = 3$, $\alpha = 0.03$ and $d = 8$ or 12. It turns out, while the full model (rank≥ 3) has a constant pure-label loss (~ 0.03, dependent on α), the rank-2 model is very likely to have a significant loss than the full model. Meanwhile, the larger d has more stable results than small d.

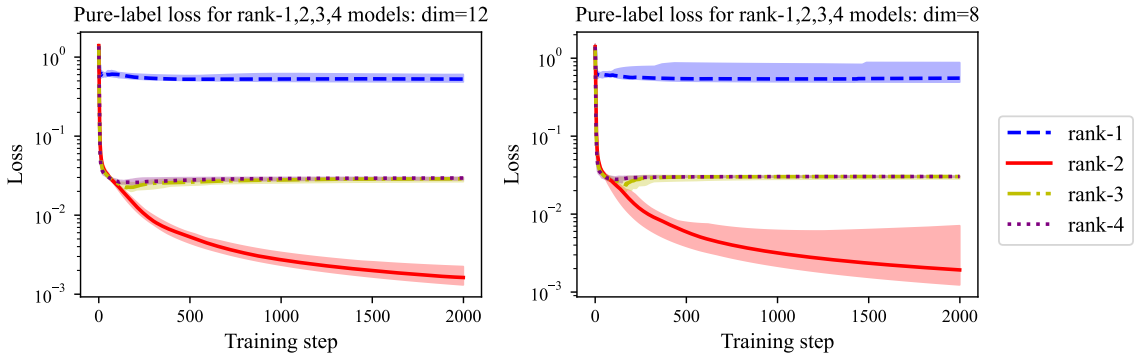

Figure 13: Pure-label loss for rank-1,2,3,4 models with $n = 3, \alpha = 0.03$ and $d = 12$ (left) or 8 (right). *Only full models are trained*, and we report low-rank results by conducting SVD in each step without manipulating the training. In both figures, the experiments are run for 20 times to examine the randomness. For each rank, we plot curves of the median, 25% and 75% out of 20 runs. It turns out: i) rank-2 models are very likely to have significantly lower pure-label loss thant full models (rank≥ 3), and ii) the larger dimension d has more stable results.

Therefore, we can qualify the following important factors for this model:

i. d *v.s.* n, c: when $d \gg n, c$, random drawn embeddings tend to be orthogonal to each other, with inner product in $O(1/\sqrt{d})$. If $n, c = \Omega(d)$, embeddings will be in strong correlations, making the problem extremely difficult to understand. [10] also discussed about such particle interaction in associative memory.

ii. Low-rank subspace storing the noise. In Figure 13, the rank-1 subspace between the full and rank-2 models is responsible to store the noise, removing which will induce a model ideally predicting the ground-truth without noise. This is understandable if the embeddings are orthogonal, as shown in Theorem **??**.

iii. α *v.s.* n. When n is large, orthogonal embeddings still induces a low-rank subspace storing the noise, but α decides whether the low-rank subspace corresponds to the smallest singular values of $\mathbf{W}$. If not, it requires more careful manipulation of the spectrum instead of low-rank approximation of $\mathbf{W}$.

Now we present a theoretical analysis of this problem with some assumptions.

Assumption E.1 (Orthonormality) *Embeddings of input and output tokens are orthonormal, i.e.,* $e_i^\top e_j = \mathbb{1}\{i = j\}, \forall\, i, j$ *and* $u_i^\top u_j = \mathbb{1}\{i = j\}, \forall\, i, j$.

Assumption E.2 (Initialization) *The learnable matrix* $\mathbf{W}$ *is initialized from* $\mathbf{0}$ *when* $t = 0$.

Theorem 2 (Restatement of Theorem ??) *Assume Assumptions E.1 and E.2 hold, considering* $n = 2, c = 3$ *and* $\alpha \in (0.2, 0.4)$*, we train the full model* $f(\cdot, \cdot; \mathbf{W})$ *with gradient flow. Denote* $P(i, j; \mathbf{W})$ *as the model's predicted probability for output* j *conditioned on input* i*. Then, for* $t \to \infty$ *and* $i \in \{1, 2\}$*, we have*

$$P(i, j; \mathbf{W}) = (1 - \alpha) \cdot \mathbb{1}\{j = i\} + \alpha \cdot \mathbb{1}\{j = c\},$$
$$P(i, j; \mathbf{W}^{(1)}) = (1 - \Theta(t^{-1/2})) \cdot \mathbb{1}\{j = i\} + \Theta(t^{-1/2}) \cdot \mathbb{1}\{j = c\}.$$

Remark 3 *Note that here the assumption* $\alpha \in (0.2, 0.4)$ *is a technical choice. In experiments, any value* $\alpha \in (0, 0.4)$ *still has the same result.*

Proof W.l.o.g., we assume the embeddings are standard basis in $\mathbb{R}^d$. For any $\mathbf{W}$, the gradient $\nabla_{\mathbf{W}} L$ can be decomposed as

$$\nabla_{\mathbf{W}} L = \gamma_1 \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 \end{bmatrix} + \gamma_2 \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}. \tag{6}$$

Since $\mathbf{W}$ initializes from zero, this implies $\mathbf{W}$ can always be decomposed with the same basis

$$\mathbf{W} = \beta_1 \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 \end{bmatrix} + \beta_2 \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}. \tag{7}$$

Then gradient flow gives the following ODE

$$\begin{aligned} \dot{\beta}_1 = -\gamma_1 &= \frac{\exp(-\beta_1 + \beta_2) - \exp(\beta_1 + \beta_2)}{\exp(-\beta_1 + \beta_2) + \exp(\beta_1 + \beta_2) + \exp(-2\beta_2)} + 1 - \alpha \\ &= \frac{\exp(-2\beta_1) - 1}{\exp(-2\beta_1) + \exp(-\beta_1 - 3\beta_2) + 1} + 1 - \alpha, \\ \dot{\beta}_2 = -\gamma_2 &= \frac{3\exp(-2\beta_2)}{\exp(-\beta_1 + \beta_2) + \exp(\beta_1 + \beta_2) + \exp(-2\beta_2)} - 3\alpha \\ &= \frac{3\exp(-\beta_1 - 3\beta_2)}{\exp(-2\beta_1) + \exp(-\beta_1 - 3\beta_2) + 1} - 3\alpha. \end{aligned} \tag{8}$$

Denoting $a = -2\beta_1, b = -\beta_1 - 3\beta_2$, the ODE becomes

$$\begin{aligned} \dot{a} &= \frac{2 - 2\exp(a)}{\exp(a) + \exp(b) + 1} - 2 + 2\alpha, \\ \dot{b} &= \frac{2 - 8\exp(b)}{\exp(a) + \exp(b) + 1} - 2 + 10\alpha. \end{aligned} \tag{9}$$

Lemma 10 gives the solution as, when $t \to \infty$,

$$a \to -\log(t) - \log(1-\alpha)(4-2\alpha), \quad b \to \log\frac{\alpha}{1-\alpha}.$$

For the full model, taking the scores $\mathbf{W}_{1,:}$ of the first input token as an example, we have $\mathbf{W}_{11} = \beta_1 + \beta_2, \mathbf{W}_{12} = -\beta_1 + \beta_2, \mathbf{W}_{13} = -2\beta_2$, so the margins are

$$\mathbf{W}_{11} - \mathbf{W}_{12} = 2\beta_1 = -a, \mathbf{W}_{11} - \mathbf{W}_{13} = \beta_1 + 3\beta_2 = -b.$$

For the rank-1 model (assuming $\beta_1 > \beta_2$), the margins are

$$\mathbf{W}^{(1)}_{11} - \mathbf{W}^{(1)}_{12} = 2\beta_1, \mathbf{W}^{(1)}_{11} - \mathbf{W}^{(1)}_{13} = \beta_1.$$

The proof finishes by computing softmax on the margins. ■

Appendix F. Proof for Theorem 1

Assumption F.1 (Orthonormal embeddings) *The embeddings $u_k \in \mathbb{R}^d$ are assumed to be orthonormal, i.e., $u_i^\top u_j = \mathbb{1}\{i = j\}$.*

Theorem 4 (Restatement of Theorem 1) *Assume $N, T \gg 1, \alpha = \Theta(1)$. Consider a one gradient step update from zero-initialization on m i.i.d. samples of $z_{1:T}$ with separate learning rates η_f for $\mathbf{W}_F$ and η_v for $\mathbf{W}_V$ (note that the gradient on $\mathbf{W}_{KQ}$ is zero). For a test sequence $z_{1:T}$, the resulting logits for the feed-forward and attention blocks satisfy, with probability $1-\delta$*

$$\left|\Delta(\xi_{ff}(x_{1:T})) - \eta_f \cdot \alpha\right| \le \eta_f \cdot O\left(\sqrt{\frac{\ln\frac{2(N+1)}{\delta}}{m}}\right),$$

$$\left|\Delta(\xi_{attn}(x_{1:T})) - \frac{\eta_v}{N}\cdot(\alpha^2\hat{q} + \alpha(1-\hat{q}))\right| \le \eta_v \cdot O\left(\sqrt{\frac{(\frac{1}{TN}+\frac{1}{N^2})\ln\frac{2(N+1)}{\delta}}{m}} + \frac{\ln\frac{2(N+1)}{\delta}}{m}\right),$$

where $\Delta(\xi) = \xi_{N+1} - \max_{j\in[N]}\xi_j$ is the margin of predicting the noise token and $\hat{q} = \frac{1}{T}\sum_{t\le T}\mathbb{1}\{z_t = N+1\}$.

Proof For $\mathbf{W}_F$, since the input is always $z_T = q$, the logits will be $[\xi_{\text{ff}}]_k = \mathbf{W}_U(k)^\top \mathbf{W}_F \mathbf{W}_E(q)$, $\forall\, k \in [N+1]$. As $\mathbf{W}_F$ is initialized from 0 and updated by GD with learning rate η_f, after one-step update, we have

$$\xi_{\text{ff}} = \mathbf{W}_U(k)^\top\left(-\eta_f \nabla_{\mathbf{W}_F}\hat{L}\Big|_{\mathbf{W}_F=0}\right)\mathbf{W}_E(q) \in \mathbb{R}^{N+1}.$$

By Lemma 5, with probability $1 - \frac{1}{2}\delta$, we have

$$|[\xi_{\text{ff}}]_{N+1} - \eta_f\cdot\alpha| \le \eta_f \cdot O\left(\sqrt{\frac{\ln\frac{2(N+1)}{\delta}}{m}}\right),$$

$$\forall\, k \le N, \quad \left|[\xi_{\text{ff}}]_k - \eta_f\cdot\left(\frac{1-\alpha}{N} - \frac{1}{N+1}\right)\right| \le \eta_f \cdot O\left(\sqrt{\frac{\ln\frac{2(N+1)}{\delta}}{Nm}} + \frac{\ln\frac{2(N+1)}{\delta}}{m}\right),$$

and then triangle inequality finishes the proof for ξ_{ff}.

For $\mathbf{W}_V$, since the gradient on $\mathbf{W}_{KQ}$ at initialization is zero, $\mathbf{W}_{KQ}$ being zero after the first step induces a uniform attention over the input sequence. Consider the input sequence $\{z_i\}_{i=1}^T$, then the logits will be $[\xi_{\text{attn}}]_j = \mathbf{W}_U(j)^\top \mathbf{W}_V \frac{1}{T}\sum_{t=1}^T \mathbf{W}_E(z_t),\ \forall\, j \in [N+1]$.

Then considering the concentration bound of $\mathbf{W}_V$ after one-step update in Lemma 6, denoting $\Gamma(j,k) = \mathbf{W}_U(j)^\top \mathbf{W}_V \mathbf{W}_E(k)$, we have

$$[\xi_{\text{attn}}]_j = \frac{1}{T}\sum_{t\le T}\Gamma(j,z_t) = \frac{1}{T}\sum_{k\le N+1} n_k \cdot \Gamma(j,k),$$

with concentration bound for each $\Gamma(\cdot,\cdot)$ in Lemma 6. From Table 2, note that for all $j = N+1, k \le N$, the expectation and variances are the same, while $k = N+1$ has slightly different expectation and variance (but still in the same order of the others). Hence, denoting $\hat{q} = \frac{1}{T}\sum_{t\le T}\mathbb{1}\{z_t = N+1\}$ dependent of the test sample $z_{1:T}$, we have

$$\left|[\xi_{\text{attn}}(x_{1:T})]_{N+1} - \frac{\eta_v}{N}\cdot(\alpha^2\hat{q} + \alpha(1-\hat{q}))\right| \le \eta_v \cdot O\left(\sqrt{\frac{(\frac{1}{TN}+\frac{1}{N^2})\ln\frac{2(N+1)}{\delta}}{m}} + \frac{\ln\frac{2(N+1)}{\delta}}{m}\right).$$

Meanwhile, as the terms in Table 2 for $j \ne N+1$ always have much smaller mean and variance by a factor $1/N$, using the Bernstein's inequalites for these terms in Lemma 6 finishes the proof for $\mathbf{W}_V$. ■

In this section, we will present the expectations and variances of $\nabla_{\mathbf{W}_V}\hat{L}$ and $\nabla_{\mathbf{W}_F}\hat{L}$ with $\mathbf{W}_V = \mathbf{W}_F = 0$ at initialization. The targets are to show:

1. a gap between $\lim_{m\to\infty}\nabla_{\mathbf{W}_V}\hat{L}$ and $\lim_{m\to\infty}\nabla_{\mathbf{W}_F}\hat{L}$ so that a step of GD with large learning rates is enough to learn the noise in $\mathbf{W}_F$, and
2. sample complexity of $\nabla_{\mathbf{W}_V}\hat{L}$ and $\nabla_{\mathbf{W}_F}\hat{L}$ based on expectations and variances.

F.1. Gradient for the Feed-forward Matrix $\mathbf{W}_F$

Lemma 5 *Consider zero initialization, $\mathbf{W}_V = \mathbf{W}_F = \mathbf{W}_{KQ} = 0$ and $N \gg 1$. Then with probability $1-\delta$, for any $j,k \in [N+1]$, it holds*

$$\begin{aligned}&\left|\mathbf{W}_U(k)^\top(\nabla_{\mathbf{W}_F}\hat{L})\mathbf{W}_E(q) - \mu(k)\right| \\ &\le \sqrt{\frac{4\sigma^2(k)\left(\ln(N+1)+\ln(\frac{2}{\delta})\right)}{m}} + \frac{4R(k)\left(\ln(N+1)+\ln(\frac{2}{\delta})\right)}{m},\end{aligned} \tag{10}$$

where $\mu(k), \sigma^2(k), R(k)$ are expectation, variance and range for different choices of $k \in [N]$ as follows:

$$\begin{array}{lccc} & \mu(N+1) = -\alpha, & \sigma^2(N+1) = \alpha(1-\alpha), & R(N+1) = \max\{\alpha, 1-\alpha\}, \\ \forall\, k \le N: & \mu(k) = \frac{1}{N+1} - \frac{1-\alpha}{N}, & \sigma^2(k) = \frac{1-\alpha}{N}, & R(k) = 1. \end{array}$$

Proof Due to zero initialization, *i.e.*, $\mathbf{W}_V = \mathbf{W}_F = 0$, the current predicted probability is $\hat{p}_{\mathbf{W}}(k|x_i) \equiv \frac{1}{N+1}$ for all $i \in [m]$ and $k \in [N+1]$. Therefore, from Lemma 8, we have

$$\nabla_{\mathbf{W}_F} \hat{L} = \frac{1}{m} \sum_{i=1}^{m} \left[\sum_{k=1}^{N+1} \left(\frac{1}{N+1} - \mathbb{1}\{y_i = k\} \right) \mathbf{W}_U(k) x_{i,T}^\top \right],$$

where $x_{i,T} \in \mathbb{R}^d = \mathbf{W}_E(z_{i,T}) + p_T$ is the input embedding with input token $z_{i,T}$ at position T in sequence i, together with positional encoding p_T for position T. Since $z_{i,T}$ is set to be the trigger q in the data generation process and p_T is assumed to orthogonal to any other vector in $\mathbf{W}_E$ in Assumption F.1, we have the following projections for $\nabla_{\mathbf{W}_F} \hat{L}$: $\forall\, k \in [N+1]$,

$$\mathbf{W}_U(k)^\top (\nabla_{\mathbf{W}_F} \hat{L}) \mathbf{W}_E(q) = \frac{1}{m} \sum_{i=1}^{m} \left(\frac{1}{N+1} - \mathbb{1}\{y_i = k\} \right).$$

From the data generation process, it is obvious to get

$$\mathbb{E}_{(x,y)} \left[\frac{1}{N+1} - \mathbb{1}\{y = k\} \right] = \frac{1}{N+1} - \alpha \cdot \mathbb{1}\{k = N+1\} - \frac{1-\alpha}{N} \cdot \mathbb{1}\{k \le N\}. \tag{11}$$

Since $\alpha = \Theta(1)$ is much larger than $\frac{1}{N+1}$ when $N \gg 1$, due to law of large numbers, we have the population gradient $\nabla_{\mathbf{W}_F} L$ satisfying

$$\begin{aligned} &\mathbf{W}_U(N+1)^\top (-\nabla_{\mathbf{W}_F} L) \mathbf{W}_E(q) \approx \alpha = \Theta(1), \\ \forall\, k \le N: \quad &\mathbf{W}_U(k)^\top (-\nabla_{\mathbf{W}_F} L) \mathbf{W}_E(q) < 0, \text{ with absolute value in } O(1/N). \end{aligned}$$

The variance of the gradient projection onto $\mathbf{W}_U(N+1)\mathbf{W}_E(q)^\top$ of a single data point follows that of Bernoulli distribution with parameter α, which means

$$\mathrm{Var}\left[\frac{1}{N+1} - \mathbb{1}\{y = N+1\} \right] = \alpha(1-\alpha). \tag{12}$$

Similarly, for any $k \le N$, the variance of the gradient projection onto $\mathbf{W}_U(N+1)\mathbf{W}_E(q)^\top$ of a single data point follows that of Bernoulli distribution with parameter $\frac{1-\alpha}{N}$, which means

$$\mathrm{Var}\left[\frac{1}{N+1} - \mathbb{1}\{y = k\} \right] = \frac{1-\alpha}{N} \left(1 - \frac{1-\alpha}{N} \right) = \Theta(1/N). \tag{13}$$

The ranges of the gradient projections' deviation from the expectation are

$$\begin{aligned} &\left| \frac{1}{N+1} - \mathbb{1}\{y = N+1\} - \left(\frac{1}{N+1} - \alpha \right) \right| \le \max\{\alpha, 1-\alpha\}, \\ \forall\, k \le N: \quad &\left| \frac{1}{N+1} - \mathbb{1}\{y = k\} - \left(\frac{1}{N+1} - \frac{1-\alpha}{N} \right) \right| \lessapprox 1. \end{aligned} \tag{14}$$

For each choice of $k \in [N+1]$ *individually*, after having the expectation $\mu(k)$, variance $\sigma^2(k)$ and range $R(k)$, by applying Bernstein's inequality, then: for each $k \in [N+1]$, with probability $1-\delta$, it holds

$$\left| \mathbf{W}_U(k)^\top (\nabla_{\mathbf{W}_F} \hat{L}) \mathbf{W}_E(q) - \mu(k) \right| \le \sqrt{\frac{4\sigma^2(k) \ln(\frac{2}{\delta})}{m}} + \frac{4R(k) \ln(\frac{2}{\delta})}{m}.$$

Then by the union bound in probability, we need $(N+1)$ events above to hold at the same time, so we can substitute δ with $\frac{\delta}{N+1}$ to have: with probability $1-\delta$, for any $k \in [N+1]$, it holds

$$\left|\mathbf{W}_U(k)^\top(\nabla_{\mathbf{W}_F}\hat{L})\mathbf{W}_E(q) - \mu(k)\right| \leq \sqrt{\frac{4\sigma^2(k)\left(\ln(N+1)+\ln(\frac{2}{\delta})\right)}{m}} + \frac{4R(k)\left(\ln(N+1)+\ln(\frac{2}{\delta})\right)}{m}. \tag{15}$$

■

F.2. Gradient for the Value Matrix $\mathbf{W}_V$

Lemma 6 *Consider zero initialization, $\mathbf{W}_V = \mathbf{W}_F = \mathbf{W}_{KQ} = 0$. Then with probability $1-\delta$, for any $j,k \in [N+1]$, it holds*

$$\begin{aligned}&\left|\mathbf{W}_U(j)^\top(\nabla_{\mathbf{W}_V}\hat{L})\mathbf{W}_E(k) - \mu(j,k)\right| \\ &\leq \sqrt{\frac{4\sigma^2(j,k)\left(2\ln(N+1)+\ln(\frac{2}{\delta})\right)}{m}} + \frac{4R(j,k)\left(2\ln(N+1)+\ln(\frac{2}{\delta})\right)}{m},\end{aligned} \tag{16}$$

where $\mu(j,k), \sigma^2(j,k), R(j,k)$ are expectation, variance and range for different choices of (j,k) at listed in Table 2.

Table 2: $\mu(j,k), \sigma^2(j,k), R(j,k)$ for different choices of (j,k) in Lemma 6.

j	k	μ	σ^2	R
$N+1$	$N+1$	$-\frac{\alpha^2}{N}$	$\frac{\alpha^2}{TN} + \frac{\alpha^3-\alpha^4}{N^2}$	$\frac{1}{2}$
$N+1$	q	$-\frac{\alpha}{N}$	$\frac{\alpha}{TN} + \frac{\alpha-\alpha^2}{N^2}$	1
$N+1$	$[N]\setminus\{q\}$	$-\frac{\alpha}{N}$	$\frac{\alpha}{TN} + \frac{\alpha-\alpha^2}{N^2}$	1
q	$N+1$	$\frac{2\alpha-1}{N^2}$	$\frac{1}{TN^2} + \frac{\alpha^2-\alpha+1}{N^3}$	$\frac{1}{2}$
q	q	$\frac{2\alpha-1}{\alpha N^2}$	$\frac{\alpha^3-\alpha^2-\alpha+2}{\alpha^3 TN^2} + \frac{\alpha^2-\alpha+1}{\alpha^2 N^3}$	1
q	$[N]\setminus\{q\}$	$\frac{\alpha}{N^2}$	$(2-\alpha)\cdot\left(\frac{1}{TN^2}+\frac{1}{N^3}\right)$	1
$[N]\setminus\{q\}$	$N+1$	$\frac{\alpha^2}{N^2}$	$(2-\alpha)\left(\frac{\alpha}{TN^2}+\frac{\alpha^2}{N^3}\right)$	$\frac{1}{3}$
$[N]\setminus\{q\}$	q	$\frac{\alpha}{N^2}$	$(2-\alpha)\left(\frac{1}{TN^2}+\frac{1}{N^3}\right)$	$\frac{1}{2}$
$[N]\setminus\{q\}$	j	$\frac{-\alpha^2+3\alpha-1}{N^2}$	$\frac{1+(1-\alpha)(2-\alpha)}{TN^2} + \frac{1+(1-\alpha)(2-\alpha)^2}{N^3}$	1
$[N]\setminus\{q\}$	$[N]\setminus\{q,j\}$	$\frac{\alpha}{N^2}$	$(2-\alpha)\left(\frac{1}{TN^2}+\frac{1}{N^3}\right)$	1

Remark 7 *The full proof is available in the full version of the paper.*

Appendix G. Useful Lemmas

Lemma 8 *Let p be a data distribution on $(x, y) \in \mathbb{R}^d \times [N]$. Consider training data as m i.i.d. samples $\mathcal{D} \triangleq \{(x_i, y_i)\}_{i=1}^m \subset \mathbb{R}^d \times [N+1]$ from p. Consider the following classification problem, with fixed output embeddings $\mathbf{W}_U$:*

$$\hat{L}(\mathbf{W}) = \frac{1}{m} \sum_{i=1}^{m} [l(y_i, \mathbf{W}_U \mathbf{W} x_i)].$$

The gradients take the following form: denoting $\hat{p}_{\mathbf{W}}(k|x_i)$ as the current predicted probability of class k in $[N+1]$ classes for input x_i,

$$\nabla_{\mathbf{W}} \hat{L}(\mathbf{W}) = \frac{1}{m} \sum_{i=1}^{m} \left[\sum_{k=1}^{N+1} (\hat{p}_{\mathbf{W}}(k|x_i) - \mathbb{1}\{y_i = k\}) \mathbf{W}_U(k) x_i^\top \right].$$

Proof Recall the form of the cross-entropy loss for classification with K classes:

$$l(y, \epsilon) = -\sum_{k=1}^{K} \mathbb{1}\{y = k\} \log \frac{e^{\xi_k}}{\sum_j e^{\xi_j}}.$$

Its derivatives take the form

$$\frac{\partial l}{\partial \xi_k}(y, \xi) = s(\xi)_k - \mathbb{1}\{y = k\},$$

where $s(\xi)_k = \frac{e^{\xi_k}}{\sum_j e^{\xi_j}}$.

The gradient of L is then given by

$$\begin{aligned} \nabla_{\mathbf{W}} \hat{L}(\mathbf{W}) &= \frac{1}{m} \sum_{i=1}^{m} \left[\sum_{k=1}^{N+1} \frac{\partial l}{\partial \xi_k}(y_i, \mathbf{W}_U \mathbf{W} x_i) \nabla_{\mathbf{W}} (\mathbf{W}_U(k)^\top \mathbf{W} x_i) \right] \\ &= \frac{1}{m} \sum_{i=1}^{m} \left[\sum_{k=1}^{N+1} (\hat{p}_{\mathbf{W}}(k|x_i) - \mathbb{1}\{y_i = k\}) \mathbf{W}_U(k) x_i^\top \right]. \end{aligned}$$

■

Lemma 9 *Consider a sequence $\{S_t\}_{t \geq 1}$ with $S_t = a^t \cdot t$ where $a \neq 1$. Then $\sum_{1 \leq t \leq T} S_t = \frac{a(1-a^T)}{(a-1)^2} + \frac{a^{T+1} \cdot T}{a-1}$.*

Proof Denote $X_t \triangleq \sum_{1 \leq t \leq T} S_t$. Then we have $a \cdot X_t = \sum_{2 \leq t \leq T+1} a^t \cdot (t-1)$. Hence, it holds $(a-1)X_t = -\sum_{2 \leq t \leq T} a^t - a + a^{T+1} \cdot T = -\frac{a(1-a^T)}{1-a} + a^{T+1} \cdot T$. Therefore, we have

$$X_t = \frac{a(1-a^T)}{(a-1)^2} + \frac{a^{T+1} \cdot T}{a-1}.$$

■

Lemma 10 *Consider the following ODE with with $a(0) = b(0) = 0$ and $\alpha \in (0.2, 0.4)$,*

$$\dot{a} = \frac{2 - 2\exp(a)}{\exp(a) + \exp(b) + 1} - 2 + 2\alpha,$$
$$\dot{b} = \frac{2 - 8\exp(b)}{\exp(a) + \exp(b) + 1} - 2 + 10\alpha.$$

Then, when $t \to \infty$, we have

$$a \to -\log(t) - \log(1-\alpha)(4-2\alpha), \quad b \to \log\frac{\alpha}{1-\alpha}.$$

Proof The ODE can be re-written as

$$\dot{a} = 2 \cdot \frac{(\alpha-2)\exp(a) + (\alpha-1)\exp(b) + \alpha}{\exp(a) + \exp(b) + 1} \triangleq \frac{2D}{\exp(a) + \exp(b) + 1},$$
$$\dot{b} = 10 \cdot \frac{(\alpha-\frac{1}{5})\exp(a) + (\alpha-1)\exp(b) + \alpha}{\exp(a) + \exp(b) + 1} \triangleq \frac{10E}{\exp(a) + \exp(b) + 1}.$$

At $t = 0$, it holds $\dot{a}(0) < 0, \dot{b}(0) < 0$ since $D = 3\alpha - 3 < 0, E = 3\alpha - \frac{6}{5} < 0$. Hence, a and b start to decrease from $t = 0$. The ending of the decreasing happens when one of D and E gets positive. Let's show D and E will never be positive when $\alpha \in (0.2, 0.4)$ by contradiction.

Assume time T_1 is when one of E and E equals to 0 for the first time. This means $E = 0$, because, for any time t, it always holds $D < E$ since $\exp(a) > 0$ for any $a \in \mathbb{R}$. Then at T_1, we have $\dot{a} < 0, \dot{b} = 0$, which means $\exp(a)$ will decrease for any small time window $\Delta t > 0$ and $\exp(b)$ stays unchanged. Together with $\alpha > 0.2$, this means it has $E < 0$ again at time $T_1 + \Delta t$. Therefore, it is possible for E to be 0, but E will never be positive. Meanwhile, this also guarantees D will always be negative because $D < E$.

Then, we make an observation that when D is always negative and E is always non-positive, the decreasing nature of a will have $D \approx E$ when $t \to \infty$ by $\exp(a) \approx 0$. This implies $b = \log\frac{\alpha}{1-\alpha}$. Then, by taking $\exp(a) = \beta \cdot t^{-\gamma}$, the ODE gives

$$-\gamma\frac{1}{t} = \frac{(2\alpha-4)\beta \cdot t^{-\gamma}}{\beta \cdot t^{-\gamma} + \frac{1}{1-\alpha}},$$

which gives $\gamma = 1, \beta = \frac{1}{(1-\alpha)(4-2\alpha)}$.

Therefore, when $t \to \infty$, we have

$$a \to \log\left(\frac{1}{(1-\alpha)(4-2\alpha)}t^{-1}\right), \quad b \to \log\frac{\alpha}{1-\alpha}.$$

■

Appendix H. Input Examples for LLMs

H.1. Examples for Prepositions

For experiments in Appendix D.1, we use two synthetic datasets: inputs are 30 prepositions, and inputs are 40 incomplete sentences ending with a preposition.

The 30 prepositions are:

”about”, ”above”, ”across”, ”after”, ”against”, ”along”, ”around”, ”at”, ”before”, ”behind”, ”below”, ”beneath”, ”beside”, ”between”, ”by”, ”during”, ”for”, ”from”, ”in”, ”inside”, ”into”, ”near”, ”of”, ”on”, ”over”, ”through”, ”to”, ”under”, ”with”, ”without”.

Generated by Claude 3 [3], the 40 incomplete sentences are:

[”Inspired painter gazed at pristine canvas, envisioning next creation about”, ”Children’s delighted squeals filled yard as they frolicked, stumbling across”, ”Singer inhaled deeply, calming nerves before gracing stage before”, ”Ominous storm clouds amassed, promising downpour that would soon roll in”, ”Awestruck trekker admired breathtaking summit vista, looking over”, ”Rich aroma of freshly roasted beans permeated cozy cafe, enticing during”, ”With deft sleight of hand, illusionist made coin vanish, leaving spectators in awe without”, ”Majestic oak stood tall, branches reaching skyward above”, ”Gentle waves caressed shoreline, soothing rhythm lulling along”, ”Meticulous investigator scoured crime scene, searching for any evidence left behind”, ”Radiant sunbeams filtered through sheer curtains, warming hardwood floor beneath”, ”Concert pianist’s nimble fingers glided across ivory keys, room resonating with melody around”, ”Crickets’ evening chorus filled silent field from nearby meadow during”, ”Jubilant laughter resounded down corridor as jovial group headed towards celebration without”, ”Struggling poet tapped pen restlessly, seeking words to capture elusive emotion beneath”, ”Soothing patter of raindrops danced on windowpane, inviting serene relaxation with”, ”Mouthwatering scent of fresh bread beckoned passersby into cozy bakery without”, ”Mighty waves thundered against jagged cliffs, echoing roar along rugged shoreline around”, ”Seasoned trekker carefully navigated winding trail, cautiously avoiding exposed roots and rocks beneath”, ”Graceful ballerina flowed across stage, movements blending seamlessly with melody during”, ”Crackling campfire cast dancing shadows across gathered faces around”, ”Vibrant brush strokes danced across canvas, bold hues bursting into life before”, ”Photographer framed breathtaking sunset, capturing fleeting beauty over glistening ocean without”, ”Stern librarian hushed raucous group, reminding them to stay quiet inside”, ”Ink flowed from author’s pen, words brimming with raw passion as page filled during”, ”Earthy aroma of freshly steeped tea perfumed air, inviting moment of serenity along”, ”Masterful guitarist’s fingers danced nimbly across strings, room alive with haunting melody around”, ”Meticulous chef artfully garnished plate, adding delicate finishing touches over”, ”Indomitable marathoner pushed through punishing final stretch, fortitude driving every stride before”, ”Engrossed scientist examined specimen’s intricate structures through microscope beneath”, ”Nervous thespian steadied breathing, striding into dazzling spotlight, delivering flawless performance with”, ”Skilled artist’s pencil glided gracefully, deftly capturing subject’s essence without”, ”Weary hiker paused to catch breath, marveling at sweeping panorama from lofty peak above”, ”Deep in thought, writer drummed fingers, seeking perfect phrasing to convey profound emotion without”, ”Lost in reverie, violinist swayed gently, fingers dancing across delicate strings during”, ”Painter’s brushstrokes burst into radiant life, canvas ablaze with vivid sunset hues over”, ”Adept photographer framed picturesque scene, preserving landscape’s beauty without”, ”World-renowned chef meticulously garnished plate, each component strategically placed around”,

”Dedicated researcher scrutinized specimen under microscope, documenting minute details beneath”, ”Seasoned actor inhaled deeply, embodying character as bright lights engulfed stage with”,].