On the Role of Unstructured Training Data in Transformers' In-Context Learning Capabilities

Kevin Christian Wibisono
University of Michigan, Statistics
kwib@umich.edu

Yixin Wang
University of Michigan, Statistics
yixinw@umich.edu

Abstract

Transformers have exhibited impressive in-context learning (ICL) capabilities: they can generate predictions for new query inputs based on sequences of inputs and outputs (i.e., prompts) without parameter updates. Efforts to provide theoretical explanations for the emergence of these abilities have primarily focused on the structured data setting, where input-output pairings in the training data are known. This scenario can enable simplified transformers (e.g., ones comprising a single attention layer without the softmax activation) to achieve notable ICL performance. However, transformers are primarily trained on unstructured data that rarely include such input-output pairings. To better understand how ICL emerges, we propose to study transformers that are trained on unstructured data, namely data that lack prior knowledge of input-output pairings. This new setting elucidates the pivotal role of softmax attention in the robust ICL abilities of transformers, particularly those with a single attention layer. We posit that the significance of the softmax activation partially stems from the equivalence of softmax-based attention models with mixtures of experts, facilitating the implicit inference of input-output pairings in the test prompts. Additionally, a probing analysis reveals where these pairings are learned within the model. While subsequent layers predictably encode more information about these pairings, we find that even the first attention layer contains a significant amount of pairing information.

1 Introduction

Transformers, like other attention-based architectures, have shown remarkable in-context learning (ICL) abilities [Brown et al., 2020]. For instance, given the prompt *"FJD: Fiji; CAD: Canada; JPY: Japan; KRW: ?"*, a well-trained transformer should produce *South Korea* as a response. As a step towards theoretically understanding how and why transformers excel at ICL, Garg et al. [2022] viewed ICL as learning a specific function class $\mathcal{F}$ from training data of the form $(x_1, f(x_1), \ldots, x_n, f(x_n), x_{n+1})$, where $f \in \mathcal{F}$, and their corresponding responses $f(x_{n+1})$. Extending this ICL formulation, subsequent studies have delved into the ICL capabilities of transformers; see Appendix A for a summary of recent works on ICL. These works often assume prior knowledge of input-output pairings in the training data, either through (trained) positional encodings or concatenated tokens comprising both x_i and $f(x_i)$ for each $i \in \{1, 2, \ldots, n\}$.

In practice, however, transformers are most often trained on unstructured natural language data. Instead of structured prompts like *"FJD: Fiji; CAD: Canada, ..."*, transformers' training data might involve sentences like *"In Canada, octane-95 gasoline costs CAD 7.00 (USD 5.20) per gallon, while the same quantity is priced at FJD 10.86 (USD 4.72) in Fiji..."* Apart from being impractical, structured training data can facilitate remarkable ICL performance even in simplified transformers, including single-layer transformers with linear or ReLU (instead of softmax) activation [e.g., Zhang

Mathematics of Modern Machine Learning Workshop at NeurIPS 2023.

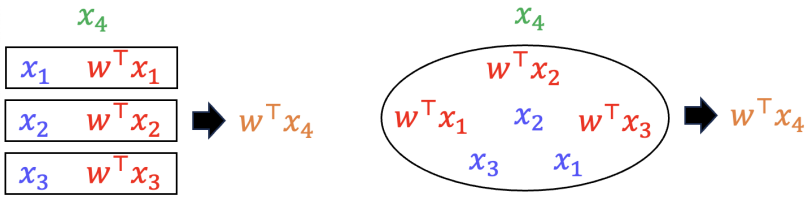

Figure 1: In-context learning (ICL) with structured (*left*) versus unstructured (*right*) training data on a test prompt with the same inputs and outputs. In the former, known $(x_i, f(x_i))$ pairings are exploited to predict $w^\top x_4$ given x_4. In contrast, such pairings are inferred in the latter. We argue that unstructured training data are more ideal for studying ICL as they better mirror how transformers are trained on natural language data. In this case, the softmax activation enables tokens to act as a mixture of experts. This allows for automatic inference of input-output pairs in the test prompt, especially with a single attention layer.

et al., 2023a, Von Oswald et al., 2023, Bai et al., 2023]. Moreover, they can lead to over-simplified analyses of ICL where the attention mechanism is not fully utilized, as we demonstrate in Section 2.

How does ICL emerge from training on unstructured data then? In this work, we introduce a novel ICL training setup that does not assume prior knowledge of input-output pairings, mimicking transformer's unstructured training data (see illustration in Figure 1). Our training prompts largely follow Garg et al.'s [2022] formulation, i.e., $(x_1, f(x_1), \ldots, x_n, f(x_n), x_{n+1})$, with positional encodings removed from the architecture. Given that transformers are inherently position-invariant [Vaswani et al., 2017], the absence of positional encodings implies that the model lacks access to any positional information.

Our empirical findings in this new setting elucidate the critical role of the softmax activation in the robust ICL abilities of transformers, especially those with a single attention layer. We argue that the significance of the softmax activation can be partially explained via the equivalence of softmax-based attention models with mixtures of experts [Jacobs et al., 1991], where each token position serves as an expert. Furthermore, we demonstrate through a probing analysis that intermediate representations from the softmax activation layers implicitly learn input-output pairings in the test prompts. While it is expected that subsequent layers contain more information about these pairings, we demonstrate that even the first attention layer encapsulates a significant amount of pairing information.

2 In-context learning with structured training data

The predominant focus of investigations into in-context learning (ICL) has been on the structured data scenario, where training prompts contain information regarding input-output pairings. Two prevalent approaches include combining x_i and $f(x_i)$ into a single token [e.g., Zhang et al., 2023a] or employing distinct tokens for x_i and $f(x_i)$ alongside positional encodings [e.g., Garg et al., 2022]. This section focuses on the former approach. In Section 2.1 and 2.2, we present theoretical and empirical evidence that ICL in this case still works well even with a single attention layer or the removal of the softmax activation. In Section 2.3, we argue that this particular form of ICL may not only be impractical but also fail to fully capture the essence of the attention mechanism.

2.1 With structured training data, ICL works well even without the softmax activation

We consider prompts of the form $P = (x_1, w^\top x_1, \ldots, x_n, w^\top x_n, x_{n+1})$ and the corresponding responses $w^\top x_{n+1}$, where $x_i's$ and w are independently sampled from the k-variate standard Gaussian distribution. To simulate structured training data, we convert P into the following matrix:

$$S = S(P) = \begin{bmatrix} x_1 & x_2 & \cdots & x_n & x_{n+1} \\ w^\top x_1 & w^\top x_2 & \cdots & w^\top x_n & 0 \end{bmatrix} \in \mathbb{R}^{(k+1)\times(n+1)}.$$

Drawing inspiration from Garg et al. [2022], we first project each column of S into a d-dimensional vector via a trainable linear transformation. This operation is then followed by ℓ consecutive softmax or linear attention layers (without positional encodings), with each attention layer containing h heads. Finally, a trainable linear transformation is applied to the last column, resulting in a scalar intended to estimate $w^\top x_{n+1}$.

n	d	ℓ	Softmax attn.	Linear attn.
10	8	1	0.3027	0.2522
		3	0.0017	0.0005
	32	1	0.2841	0.2510
		3	0.0007	0.0002
20	8	1	0.1722	0.1454
		3	0.0004	0.0002
	32	1	0.1449	0.1405
		3	0.0002	0.0001
30	8	1	0.1212	0.0611
		3	0.0001	0.0001
	32	1	0.0750	0.0982
		3	0.0001	0.0000

Table 1: With structured training data, trained transformers using both softmax and linear activations perform ICL well across various parameter combinations (n: number of input-output pairs within each prompt; d: dimensionality of the projected tokens; ℓ: number of attention layers). Here, each number represents the average mean squared error (MSE) between predicted and actual responses over the last 5,000 steps.

n	d	ℓ	Softmax attn.	Linear attn.
10	8	1	1.2521	1.8668
		3	1.0022	0.8325
	32	1	1.2293	1.8631
		3	0.6996	0.6843
20	8	1	1.2726	1.8648
		3	0.9678	1.0307
	32	1	1.2308	1.8635
		3	0.7372	0.6743
30	8	1	1.3079	1.8577
		3	1.0039	1.0174
	32	1	1.2424	1.8538
		3	0.6116	0.6751

Table 2: ICL with unstructured training data is more challenging across various parameter combinations (n: number of input-output pairs within each prompt; d: dimensionality of the projected tokens; ℓ: number of attention layers), and requires the softmax activation to perform reasonably well when $\ell = 1$. Here, each number represents the average MSE between predicted and actual responses over the last 5,000 steps.

The model is trained by minimizing the mean squared error loss over 300,000 steps using the Adam optimizer [Kingma and Ba, 2015] with a learning rate of 10^{-3}. We set $k = 2$, $h = 4$, $\ell \in \{1, 3\}$, $n \in \{10, 20, 30\}$, and $d \in \{8, 32\}$. To avoid overfitting, we generate a new prompt-response batch of size 256 at each step following Garg et al. [2022]. The result is in Table 1: the trained transformers demonstrate effective ICL performance even without the softmax activation in the attention layers.

2.2 In some architectures, ICL performance is notable with a single linear attention layer

To illustrate ICL performance with a single linear attention layer, we consider the architecture in Zhang et al. [2023a]: given any prompt S, it outputs the bottom-right entry of $f(S) = S + \frac{1}{n} W^{PV} S \cdot S^\top W^{KQ} S$ (denoted by $f_{BR}(S)$). Here, $W^{PV}, W^{KQ} \in \mathbb{R}^{(k+1)\times(k+1)}$ are trainable parameters. We follow the same training procedure as in Section 2.1. Table 3 in Appendix C shows that the trained transformers exhibit excellent ICL abilities even when W^{PV} and W^{KQ} are set to identity. This result is not surprising due to the following lemma, whose proof is provided in Appendix B.

Lemma 1. *Let $\Lambda \in \mathbb{R}^{k\times k}$ be a positive definite matrix with eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_k > 0$. Consider a prompt S where $x_i's$ are independently generated from $\mathcal{N}_k(0, \Lambda)$ and w is generated from $\mathcal{N}_k(0, I)$. Let $f_{BR}(S)$ denote the prediction for $w^\top x_{n+1}$ when $W^{PV} = W^{KQ} = I$. We have*

$$\operatorname{corr}\left(w^\top x_{n+1}, f_{BR}(S)\right) \to \frac{\sum_{i=1}^k \lambda_i^2}{\sqrt{\sum_{i=1}^k \lambda_i}\sqrt{\sum_{i=1}^k \lambda_i^3}}$$

as $n \to \infty$. When $k = 2$, the limiting correlation is lower bounded by $2\sqrt{2}/3$.

Lemma 1 suggests that even if the weight matrices are set to identity (i.e., no parameters are learned), there is a significant correlation between the predicted and actual responses assuming a sufficiently large number of input-output pairs. Specifically, when $k = 2$, the correlation converges to a near perfect value of around 0.95.

2.3 ICL with structured training data may under-utilize core attention mechanism features

A common theme in the theoretical arguments for ICL is the notion that transformers are capable of implementing gradient descent [e.g., Von Oswald et al., 2023, Akyürek et al., 2022, Dai et al., 2023,

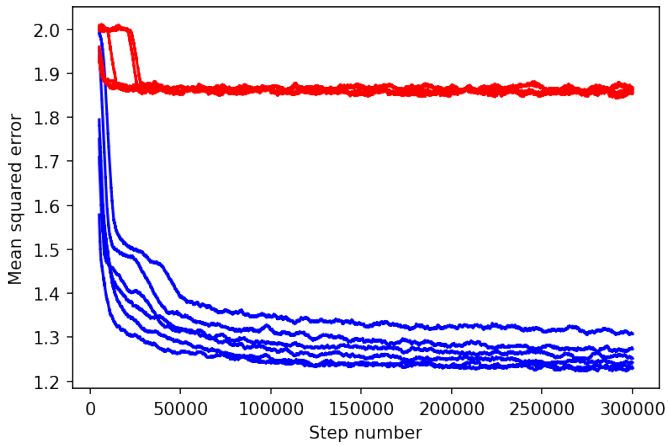

Figure 2: The importance of the softmax attention layer in facilitating ICL with structured training data when $\ell = 1$ is evident from the observed training dynamics when utilizing softmax and linear attention layers. Here, each line represents one combination of parameters.

Bai et al., 2023]. This is typically demonstrated through the construction of a transformer specifically designed to perform this function. It turns out that in this structured training data setting, the essence of the attention mechanism may not be fully leveraged. For example, consider a simplified version of Bai et al.'s [2023] construction. Specifically, they analyzed prompts of the form

$$H = \begin{bmatrix} x_1 & x_2 & \cdots & x_n & x_{n+1} \\ y_1 & y_2 & \cdots & y_n & 0 \\ p_1 & p_2 & \cdots & p_n & p_{n+1} \end{bmatrix} \in \mathbb{R}^{K \times (n+1)},$$

where $p_i = [\mathbf{0}, 1, \mathbb{1}(i \leq n)]^\top \in \mathbb{R}^{K-k-1}$ for some $K = \Theta(k)$. Their objective is to minimize the mean squared error loss as given by $L(w) = \frac{1}{2n} \sum_{i=1}^{n} (w^\top x_i - y_i)^2$. A step of gradient descent with step size η thus transforms w into $\tilde{w} = w - \frac{\eta}{n} \sum_{i=1}^{n} x_i (w^\top x_i - y_i)$. Bai et al. [2023] showed that it is possible to construct a one-layer, two-head transformer with the ReLU activation that transforms each column of H of the form $h_i = [x_i, y_i', w, \mathbf{0}, 1, t_i]^\top$, where $y_i' = y_i \mathbb{1}(i \leq n)$ and $t_i = \mathbb{1}(i \leq n)$, into $\tilde{h}_i = [x_i, y_i', \tilde{w}, \mathbf{0}, 1, t_i]^\top$. This transformation mimics a gradient descent step.

Despite this connection, their attention mechanism does not involve comparisons across different tokens to determine their relative importance. In particular, in their construction, h_i is related to $\tilde{h}_i$ via the equation

$$\tilde{h}_i = h_i + \frac{1}{n+1} \sum_{m=1}^{2} \sum_{j=1}^{n+1} \sigma(\langle Q_m h_i, K_m h_j \rangle) V_m h_j$$

for particular choices of Q_m, K_m and V_m ($m \in \{1, 2\}$), where $\sigma(\cdot)$ denotes the ReLU activation. These choices yield the *attention weights* $\sigma(\langle Q_1 h_i, K_1 h_j \rangle) = \frac{1}{2} \sigma(w^\top x_j - y_j) \mathbb{1}(j \leq n)$ and $\sigma(\langle Q_2 h_i, K_2 h_j \rangle) = \frac{1}{2} \sigma(-w^\top x_j + y_j) \mathbb{1}(j \leq n)$, which are independent of h_z for any $z \neq j$. As a remedy, we introduce a new ICL training setting that does not assume known input-output pairings in Section 3. In this setting, the softmax attention proves essential for ensuring robust ICL performance, especially in the case of a single attention layer.

3 In-context learning with unstructured training data

In Section 2, we showed that in-context learning (ICL) with structured training data may not only be impractical but also fall short in capturing the fundamental aspects of the attention mechanism. We now introduce a novel ICL training setup that does not assume known input-output pairings in each training prompt. Specifically, each training prompt $P = \left(x_1, w^\top x_1, \ldots, x_n, w^\top x_n, x_{n+1}\right)$ is now transformed into

$$T = T(P) = \begin{bmatrix} x_1 & \mathbf{0} & \cdots & x_n & \mathbf{0} & x_{n+1} \\ 0 & w^\top x_1 & \cdots & 0 & w^\top x_n & 0 \end{bmatrix} \in \mathbb{R}^{(k+1) \times (2n+1)}.$$

We adopt the transformer architecture and training specifics outlined in Section 2.1. Table 2 and Figure 2 highlight the challenging nature of this new scheme and the crucial role of the softmax activation, especially with a single attention layer. In Section 3.1, we offer a potential explanation for the importance of the softmax activation, drawing insights from mixtures of experts [Jacobs et al., 1991]. In Section 3.2, we present a probing result that offers empirical evidence that transformers can infer input-output pairings in the test prompts. This finding sheds light on how transformers can excel in ICL despite being trained on unstructured text data.

3.1 Softmax attention layers serve as mixtures of experts

To illustrate our argument, we consider two test prompts with $k = 1$ and $n = 3$: $[1, 10, 4, 40, 3, 30, 6]$ (target response: $6 \times 10 = 60$) and $[1, 4, 10, 40, 3, 12, 5]$ (target response: $5 \times 4 = 20$). With a single attention layer, we expect that token 1 pay more attention to 10 (4) as compared to 4 (10) in the first (second) prompt. As discussed in Section 2.3, it is impossible to achieve this behavior using linear attention where the attention weight from one token to another can only depend on these two tokens alone. However, Proposition 2 highlights how softmax attention overcomes this limitation by acting as a mixture of experts, with each token position being an expert. The details and proof are deferred to Appendix D.

Proposition 2. *For any prompt T, a one-layer, h-head, softmax transformer with no bias terms following the structure in Section 2.1 outputs a stacked mixture-of-experts prediction $\hat{y} = \hat{y}(T) = \sum_{i=1}^{h} \left(\sum_{j=1}^{2n+1} \pi_j^i(T) \beta_j^i(T) \right)$ (see Appendix D for detailed definitions of $\pi_j^i(T)$ and $\beta_j^i(T)$).*

With multiple attention layers, ICL performance is not compromised when using non-softmax activation, as shown in Table 2. We hypothesize that multiple attention layers could potentially function as mixtures of experts, and leave the detailed analysis for future work.

3.2 ICL with unstructured training data learns input-output pairings in the test prompts

We finally perform a probing analysis to study whether ICL with unstructured training data indeed learns input-output pairings in the test prompts. Specifically, we fix the weights of a four-head, six-layer transformer with $n = 20$ and $d = 32$ that has been trained for 1.6 million steps following the procedure detailed in Section 2.1. For each $0 \leq i \leq 6$, denote the intermediate representation of x_1 in the i-th attention layer by $r_i(x_1) \in \mathbb{R}^d$ (here, $i = 0$ refers to the representation right before the first attention layer). Subsequently, a neural network is trained to predict $w^\top x_1$ given $r_i(x_1)$ on a newly created training set of size $10,000$. Evaluating $\mathrm{corr}(r_i(x_1), y_1)$ for each i on a test set of the same size yields correlations of 0.004, 0.631, 0.639, 0.663, 0.747, 0.811, and 0.780, respectively. In contrast, if we replicate the same experiment except that we predict $w^\top x_2$ given $r_i(x_1)$, we observe correlations of at most 0.2 that do not increase with i. These results indicate that transformers can infer input-output pairings in the test prompts, even as early as in the first hidden layer. Moreover, increasingly robust pairing information is encoded as we go deeper into the network.

4 Discussion

This paper explores the role of structured and unstructured training data in the in-context learning (ICL) capabilities of transformers. We show that structured training data, i.e., those with known input-output pairings, yield robust ICL performance even without essential attention features like the softmax activation. To better understand the role of unstructured training data in ICL, we introduce a novel ICL training setup, revealing the crucial role of the softmax attention layer particularly within a single-layer transformer. We posit that this phenomenon occurs partly due to the resemblance between softmax-based attention models and mixtures of experts, allowing for informed inferences about input-output pairings in the test prompts. A probing analysis shows that transformers start learning these pairings as early as in the first attention layer. Formal theoretical arguments explaining the emergence of transformers' ICL abilities from unstructured training data are interesting avenues for future work.

Acknowledgements. This work is supported in part by the Office of Naval Research under grant number N00014-23-1-2590 and the National Science Foundation under grant number 2231174 and number 2310831.

References

K. Ahn, X. Cheng, H. Daneshmand, and S. Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. *arXiv preprint arXiv:2306.00297*, 2023.

K. Ahuja, M. Panwar, and N. Goyal. In-context learning through the Bayesian prism. *arXiv preprint arXiv:2306.04891*, 2023.

E. Akyürek, D. Schuurmans, J. Andreas, T. Ma, and D. Zhou. What learning algorithm is in-context learning? Investigations with linear models. In *International Conference on Learning Representations*, 2022.

Y. Bai, F. Chen, H. Wang, C. Xiong, and S. Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. In *Neural Information Processing Systems*, 2023.

A. Bietti, V. Cabannes, D. Bouchacourt, H. Jegou, and L. Bottou. Birth of a transformer: A memory viewpoint. In *Neural Information Processing Systems*, 2023.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *Neural Information Processing Systems*, 2020.

S. C. Chan, A. Santoro, A. K. Lampinen, J. X. Wang, A. K. Singh, P. H. Richemond, J. McClelland, and F. Hill. Data distributional properties drive emergent in-context learning in transformers. In *Neural Information Processing Systems*, 2022.

D. Dai, Y. Sun, L. Dong, Y. Hao, Z. Sui, and F. Wei. Why can GPT learn in-context? Language models secretly perform gradient descent as meta optimizers. In *Association for Computational Linguistics*, 2023.

N. Ding, T. Levinboim, J. Wu, S. Goodman, and R. Soricut. CausalLM is not optimal for in-context learning. *arXiv preprint arXiv:2308.06912*, 2023.

M. L. Eaton. Multivariate statistics: A vector space approach. 1983.

D. Fu, T.-Q. Chen, R. Jia, and V. Sharan. Transformers learn higher-order optimization methods for in-context learning: A study with linear models. In *Workshop on Mathematics of Modern Machine Learning at NeurIPS*, 2023.

S. Garg, D. Tsipras, P. S. Liang, and G. Valiant. What can transformers learn in-context? A case study of simple function classes. In *Neural Information Processing Systems*, 2022.

T. Guo, W. Hu, S. Mei, H. Wang, C. Xiong, S. Savarese, and Y. Bai. How do transformers learn in-context beyond simple functions? A case study on learning with representations. In *Workshop on Mathematics of Modern Machine Learning at NeurIPS*, 2023.

M. Hahn and N. Goyal. A theory of emergent in-context learning as implicit structure induction. *arXiv preprint arXiv:2303.07971*, 2023.

C. Han, Z. Wang, H. Zhao, and H. Ji. Explaining emergent in-context learning as kernel regression. *arXiv preprint arXiv:2305.12766*, 2023.

Y. Huang, Y. Cheng, and Y. Liang. In-context convergence of transformers. In *Workshop on Mathematics of Modern Machine Learning at NeurIPS*, 2023.

R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 1991.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

J. Kossen, T. Rainforth, and Y. Gal. In-context learning in large language models learns label relationships but is not conventional learning. *arXiv preprint arXiv:2307.12375*, 2023.

S. Li, Z. Song, Y. Xia, T. Yu, and T. Zhou. The closeness of in-context learning and weight shifting for softmax regression. *arXiv preprint arXiv:2304.13276*, 2023a.

Y. Li, M. E. Ildiz, D. Papailiopoulos, and S. Oymak. Transformers as algorithms: Generalization and stability in in-context learning. In *International Conference on Machine Learning*, 2023b.

S. Lu, I. Bigoulaeva, R. Sachdeva, H. T. Madabushi, and I. Gurevych. Are emergent abilities in large language models just in-context learning? *arXiv preprint arXiv:2309.01809*, 2023.

A. Mahankali, T. B. Hashimoto, and T. Ma. One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention. *arXiv preprint arXiv:2307.03576*, 2023.

S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, and L. Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Empirical Methods in Natural Language Processing*, 2022.

C. Olsson, N. Elhage, N. Nanda, N. Joseph, N. DasSarma, T. Henighan, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, D. Drain, D. Ganguli, Z. Hatfield-Dodds, D. Hernandez, S. Johnston, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. Brown, J. Clark, J. Kaplan, S. McCandlish, and C. Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022.

A. Raventós, M. Paul, F. Chen, and S. Ganguli. Pretraining task diversity and the emergence of non-Bayesian in-context learning for regression. In *Neural Information Processing Systems*, 2023.

R. Ren and Y. Liu. In-context learning with transformer is really equivalent to a contrastive learning pattern. *arXiv preprint arXiv:2310.13220*, 2023.

L. Shen, A. Mishra, and D. Khashabi. Do pretrained transformers really learn in-context by gradient descent? *arXiv preprint arXiv:2310.08540*, 2023.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Neural Information Processing Systems*, 2017.

J. Von Oswald, E. Niklasson, E. Randazzo, J. Sacramento, A. Mordvintsev, A. Zhmoginov, and M. Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, 2023.

X. Wang, W. Zhu, M. Saxon, M. Steyvers, and W. Y. Wang. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning. In *Neural Information Processing Systems*, 2023.

N. Wies, Y. Levine, and A. Shashua. The learnability of in-context learning. In *Neural Information Processing Systems*, 2023.

J. Wu, D. Zou, Z. Chen, V. Braverman, Q. Gu, and P. L. Bartlett. How many pretraining tasks are needed for in-context learning of linear regression? *arXiv preprint arXiv:2310.08391*, 2023.

S. M. Xie, A. Raghunathan, P. Liang, and T. Ma. An explanation of in-context learning as implicit Bayesian inference. In *International Conference on Learning Representations*, 2021.

S. Yadlowsky, L. Doshi, and N. Tripuraneni. Pretraining data mixtures enable narrow model selection capabilities in transformer models. *arXiv preprint arXiv:2311.00871*, 2023.

R. Zhang, S. Frei, and P. L. Bartlett. Trained transformers learn linear models in-context. In *Workshop on Robustness of Zero/Few-Shot Learning in Foundation Models at NeurIPS*, 2023a.

Y. Zhang, F. Zhang, Z. Yang, and Z. Wang. What and how does in-context learning learn? Bayesian model averaging, parameterization, and generalization. *arXiv preprint arXiv:2305.19420*, 2023b.

Supplementary Material

A Related work on in-context learning

Since the discovery that transformers demonstrate exceptional performance at in-context learning (ICL) [Brown et al., 2020], numerous studies have been devoted to understanding this intriguing phenomenon from diverse theoretical and empirical perspectives. Many of them adopt Garg et al.'s 2022 formulation of ICL as learning a specific function class $\mathcal{F}$ from prompts of the form $(x_1, f(x_1), \ldots, x_n, f(x_n), x_{n+1})$, where $f \in \mathcal{F}$, and their corresponding responses $f(x_{n+1})$. Here, ICL refers to the transformer's ability to produce a response close to $g(y_{n+1})$ when supplied with a prompt $(y_1, g(y_1), \ldots, y_n, g(x_n), y_{n+1})$ for any $g \in \mathcal{F}$.

In pursuit of a deeper understanding of ICL, some studies take a *Bayesian* perspective. Xie et al. [2021] conceptualized ICL as implicit Bayesian inference, wherein language models infer a latent document-level concept to generate coherent next tokens in the pre-training phase and a shared latent concept among examples in a prompt during testing. Wang et al. [2023] argued that large language models operate as latent variable models with latent variables encompassing task-related information being implicitly derived, thereby playing a crucial role in their remarkable ICL performance. Ahuja et al. [2023] provided empirical evidence of transformers displaying characteristics akin to the Bayesian predictor when tackling ICL across linear and non-linear function classes. Zhang et al. [2023b] demonstrated that without updating the neural network parameters, ICL is equivalent to Bayesian model averaging parameterized by the attention mechanism.

Other studies argue that transformers can learn in-context by *gradient descent*. Akyürek et al. [2022], Von Oswald et al. [2023], and Dai et al. [2023] showed that transformers can implement gradient descent, providing a mechanistic framework for comprehending ICL on regression problems. Bai et al. [2023] employed an efficient implementation of in-context gradient descent to establish generalization bounds and argued that transformers can perform algorithm selection in a manner analogous to statisticians. Zhang et al. [2023a] established that the gradient flow dynamics of transformers converge to a global minimum capable of performing ICL. Ahn et al. [2023] scrutinized the optimization landscape of transfomers, revealing that the optimal parameters align with an iteration of preconditioned gradient descent. Huang et al. [2023] studied the learning dynamics of single-layer transformers with softmax attention trained via gradient descent to perform ICL on linear functions. Li et al. [2023a] investigated ICL using a softmax regression formulation and demonstrated that the models learned through gradient descent exhibit a high degree of similarity to transformers.

In a related exploration, Mahankali et al. [2023] proved that in single-layer linear transformers, minimizing the pretraining loss is analogous to performing a step of gradient descent when the covariates are sampled from the standard Gaussian distribution. Ren and Liu [2023] established a link between ICL with softmax attention and contrastive learning, interpreting the inference process of ICL as a form of gradient descent within a contrastive learning framework. However, some studies present evidence suggesting that transformers may not exclusively rely on gradient descent in performing ICL. Fu et al. [2023] revealed that for linear regression, ICL is acquired through higher-order optimization techniques such as iterative Newton's method rather than gradient descent. While numerous studies have developed transformers capable of emulating gradient descent, Shen et al. [2023] argued that the direct equivalence between gradient descent and ICL might not necessarily apply in real-world scenarios.

The *pretraining aspects*, e.g., data quantity and distribution, task diversity, and algorithm, are also common themes in numerous works on ICL. Min et al. [2022] discovered that the precise input-label mapping in the demonstrations used for ICL does not affect performance, whereas factors such as independent specification of the input and label spaces have a more substantial influence. Chan et al. [2022] revealed that the impressive ICL capabilities observed in transformers are influenced by both the characteristics of the training data distributions and the inherent architectural features of the models. Kossen et al. [2023] examined the influence of the conditional label distribution in in-context examples on ICL predictions, revealing that ICL takes into account in-context label information and can even acquire the capability to learn entirely new tasks in-context.

In a similar spirit, Wu et al. [2023] demonstrated that pretraining single-layer linear attention models for performing ICL on linear regression with a Gaussian prior can be achieved effectively with a minimal number of independent tasks, regardless of the task dimension. Raventós et al. [2023]

highlighted the presence of a task diversity threshold that differentiates between the regimes in which transformers can or cannot successfully tackle previously unseen tasks. Yadlowsky et al. [2023] argued the impressive ICL capabilities of transformers could be attributed to the range and diversity of the data mixtures in their pretraining rather than relying solely on their inductive biases for generalizing to new tasks. Ding et al. [2023] compared transformers' ICL performance when trained with prefixLM (allowing in-context samples to attend to all tokens) and causalLM (preventing in-context samples from attending to subsequent tokens), and concluded that the latter led to inferior ICL performance.

Other studies look into ICL from a *learning theory* perspective. Wies et al. [2023] presented the first PAC-type framework for ICL and provided finite-sample complexity results. Hahn and Goyal [2023] derived an information-theoretic bound showing how ICL emerges from the general task of predicting the next token. Several other investigations approach ICL with an emphasis on *mechanistic interpretability*. Olsson et al. [2022] attributed ICL in large transformers to the development of attention heads with the ability to complete token sequences such as [A][B] $\cdots$ [A] $\rightarrow$ [B], which they referred to as "induction heads." Bietti et al. [2023] analyzed a setup where tokens are generated from either global or context-specific bigram distributions to differentiate global from in-context learning, showing that the former occurs rapidly and the latter is achieved gradually through the development of an induction head.

Finally, several works delve into other aspects of ICL. Li et al. [2023b] viewed ICL as an algorithm learning problem in which a transformer model implicitly constructs a hypothesis function at inference-time, and presented generalization bounds through the perspective of multi-task learning. Han et al. [2023] contended that transformers' capacity to execute ICL following training on a general language corpus can be attributed to their ability to simulate kernel regression. Guo et al. [2023] examined ICL within a more realistic framework where the label is influenced by the input through a potentially complex yet constant representation function, combined with a distinct linear function for each instance. Lu et al. [2023] asserted that emergent abilities in transformers can be primarily attributed to ICL.

B Proof of Lemma 1

Proof. A straightforward calculation yields $f_{BR}(S) = w^\top L_n x_{n+1}$, where $L_n = \frac{1}{n}\sum_{i=1}^n x_i x_i^\top$. Let $X = (x_1, x_2, \ldots, x_n)$. Observe that

$$\begin{aligned}
\operatorname{cov}\left(w^\top x_{n+1}, f_{BR}(S)\right) &= \operatorname{cov}\left(w^\top x_{n+1}, w^\top L_n x_{n+1}\right) \\
&= \mathbb{E}\left(\operatorname{cov}\left(w^\top x_{n+1}, w^\top L_n x_{n+1} \mid X, w\right)\right) \\
&= \mathbb{E}\left(w^\top \Lambda L_n w\right) \\
&= \mathbb{E}\left(\mathbb{E}\left(w^\top \Lambda L_n w \mid X\right)\right) \\
&= \mathbb{E}\left(\operatorname{tr}\left(\Lambda L_n\right)\right) \\
&= \operatorname{tr}\left(\Lambda \mathbb{E}\left(L_n\right)\right) \\
&= \operatorname{tr}\left(\Lambda^2\right),
\end{aligned}$$

where we used the law of total covariance, the law of total expectation, and the linearity of $\operatorname{tr}(\cdot)$. Similarly, we have $\operatorname{var}(w^\top x_{n+1}) = \mathbb{E}(w^\top \Lambda w) = \operatorname{tr}(\Lambda)$ and

$$\begin{aligned}
\operatorname{var}\left(f_{BR}(S)\right) &= \operatorname{var}\left(w^\top L_n x_{n+1}\right) \\
&= \mathbb{E}\left(\operatorname{var}\left(w^\top L_n x_{n+1} \mid X, w\right)\right) \\
&= \mathbb{E}\left(w^\top L_n \Lambda L_n w\right) \\
&= \mathbb{E}\left(\mathbb{E}\left(w^\top L_n \Lambda L_n w \mid X\right)\right) \\
&= \mathbb{E}\left(\operatorname{tr}\left(L_n \Lambda L_n\right)\right) \\
&= \mathbb{E}\left(\operatorname{tr}\left(\Lambda L_n^2\right)\right) \\
&= \operatorname{tr}\left(\Lambda \mathbb{E}\left(L_n^2\right)\right) \\
&\to \operatorname{tr}\left(\Lambda^3\right),
\end{aligned}$$

since

$$\begin{aligned}
\mathbb{E}\left(L_n^2\right) &= \operatorname{var}\left(L_n\right) + \left(\mathbb{E}\left(L_n\right)\right)^2 \\
&= \frac{1}{n^2}\operatorname{var}\left(\mathcal{W}_k(\Lambda, n)\right) + \Lambda^2 \\
&\to \Lambda^2
\end{aligned}$$

as $n \to \infty$ by Proposition 8.3 of Eaton [1983]. Here, $\mathcal{W}$ refers to the Wishart distribution.

The first result then follows from the well-known fact that the trace of a matrix is the same as the sum of its eigenvalues. The second result follows from the inequality $9(a^2+b^2)^2 \geq 8(a+b)(a^3+b^3)$ for every $a, b > 0$, which is equivalent to $(a^2 - 4ab + b^2)^2 \geq 0$. Equality is attained if and only if $b = (2-\sqrt{3})a$ or $b = (2+\sqrt{3})a$. □

C ICL performance under the architecture described in Section 2.2

Λ	n	Trainable	Fixed
$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	10	0.8800	0.8801
	30	0.9538	0.9540
$\begin{bmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & 1 \end{bmatrix}$	10	0.8783	0.8486
	30	0.9544	0.9034
$\begin{bmatrix} 1 & -\frac{1}{2} \\ -\frac{1}{2} & 1 \end{bmatrix}$	10	0.8675	0.8304
	30	0.9528	0.9090

Table 3: In the architecture described in Section 2.2, trained transformers successfully perform ICL across various parameter combinations (n: number of input-output pairs within each prompt; Λ: covariance matrix of the Gaussian distribution that generates x_i's), even when the weights W^{PV} and W^{KQ} are set to identity (*Fixed*). Here, each number represents the Pearson correlation between predicted and actual responses on a test set of size 5,000.

D Details and proof of Proposition 2

We begin by examining how a transformer adhering to the structure outlined in Section 2.1 with $\ell = 1$ generates a predicted response for a given prompt of the form

$$T = T(P) = \begin{bmatrix} x_1 & \mathbf{0} & \cdots & x_n & \mathbf{0} & x_{n+1} \\ 0 & w^\top x_1 & \cdots & 0 & w^\top x_n & 0 \end{bmatrix} \in \mathbb{R}^{(k+1)\times(2n+1)}.$$

The following steps provide a comprehensive overview of this process, with bias matrices removed to enhance clarity in explanation.

1. **Linear projection of input.** We first project each token into a d-dimensional vector. In matrix form, this can be written as $T^{\mathrm{proj}} = CT \in \mathbb{R}^{d\times(2n+1)}$, where $C \in \mathbb{R}^{d\times(k+1)}$.
2. **Attention mechanism for each head.** For each attention head $i \in \{1, 2, \ldots, h\}$, we introduce query, key and value mappings $Q_i, K_i, V_i \in \mathbb{R}^{d'\times d}$ and compute

   $$T_i^{\mathrm{attn}} = V_i T^{\mathrm{proj}} \mathrm{softmax}\left(\left(Q_i T^{\mathrm{proj}}\right)^\top \left(K_i T^{\mathrm{proj}}\right)\right) \in \mathbb{R}^{d'\times(2n+1)},$$

   where the softmax is applied column-wise.
3. **Concatenation of heads.** We then concatenate the outputs from all h heads and apply a linear transformation to restore the dimensionality to $d \times (2n + 1)$. In matrix form, this can be written as

   $$T' = \sum_{i=1}^{h} O_i T_i^{\mathrm{attn}} \in \mathbb{R}^{d\times(2n+1)},$$

   where $O_i \in \mathbb{R}^{d\times d'}$ for each $i \in \{1, 2, \ldots, h\}$.
4. **Residual connection.** We apply a residual connection, yielding

   $$T'' = T^{\mathrm{proj}} + T' \in \mathbb{R}^{d\times(2n+1)}.$$

5. **Linear projection of last column.** A linear transformation is applied to the last column, resulting in the predicted response

   $$\hat{y} = \alpha^\top T'' e_{2n+1} \in \mathbb{R}.$$

   Here, $\alpha \in \mathbb{R}^d$ and $e_j \in \{0, 1\}^{2n+1}$ denotes a zero vector with 1 on the j-th entry.

We now state the proof of Proposition 2.

Proof. For each $i \in \{1, 2, \ldots, h\}$, let $O_i V_i = OV_i \in \mathbb{R}^{d\times d}$ and $Q_i^\top K_i = QK_i \in \mathbb{R}^{d\times d}$. Observe that

$$\begin{aligned}
\hat{y} &= \alpha^\top \left(CT + \sum_{i=1}^{h} (OV_i)\, CT \mathrm{softmax}\left(T^\top C^\top (QK_i)\, CT\right)\right) e_{2n+1} \\
&= \sum_{i=1}^{h}\sum_{j=1}^{2n+1} \left(\frac{\exp\left(e_j^\top T^\top C^\top (QK_i) CT e_{2n+1}\right)}{\sum_{j=1}^{2n+1} \exp\left(e_j^\top T^\top C^\top (QK_i) CT e_{2n+1}\right)} \right) \left(\alpha^\top (OV_i) CT e_j + \frac{1}{h}\left(\alpha^\top CT e_{2n+1}\right)\right) \\
&= \sum_{i=1}^{h} \left(\sum_{j=1}^{2n+1} \pi_j^i(T)\beta_j^i(T)\right),
\end{aligned}$$

where

$$\beta_j^i(T) = \alpha^\top (OV_i) CT e_j + \frac{1}{h}\left(\alpha^\top CT e_{2n+1}\right)$$

and

$$\pi_j^i(T) = \frac{\exp\left(e_j^\top T^\top C^\top (QK_i) CT e_{2n+1}\right)}{\sum_{j=1}^{2n+1} \exp\left(e_j^\top T^\top C^\top (QK_i) CT e_{2n+1}\right)},$$

completing the proof. □

Note that the resulting predictor can be interpreted as a *stacked mixture of experts* [Jacobs et al., 1991]. Indeed, for each head $i \in \{1, 2, \ldots, h\}$ and token $j \in \{1, 2, \ldots, 2n+1\}$, $\beta_j^i(T)$ represents the prediction of the j-th expert in the i-th head and $\pi_j^i(T)$ represents its corresponding expert weight.

It is important to emphasize that for a fixed head i, $\pi_j^i(T)$ depends on *all* columns of T, as indicated by the presence of the terms $e_j^\top T^\top$ for all $j \in \{1, 2, \ldots, 2n+1\}$. This dependence is facilitated by the softmax attention, enabling it to achieve the desired behavior illustrated through the example in Section 3.1.

In contrast, if we substitute the softmax attention with linear attention, we have

$$\pi_j^i(T) = e_j^\top T^\top C^\top (QK_i) CT e_{2n+1},$$

resulting in $\pi_j^i(T)$ being dependent solely on the j-th and $(2n+1)$-th columns of T. Additionally, we no longer have $\sum_{j=1}^{2n+1} \pi_j^i(T) \neq 1$ for each head $i \in \{1, 2, \ldots, h\}$.