A Simple Generalisation of the Implicit Dynamics of In-Context Learning

Francesco Innocenti[1, 2, 3] **El Mehdi Achour**[4]

[1]MRC Brain Network Dynamics Unit, University of Oxford, UK
[2]MRC CoRE in Restorative Neural Dynamics, UK
[3]University of Sussex, Brighton, UK
[4]University Mohammed VI Polytechnic, College of Computing, Rabat, Morocco
Correspondence to: `francesco.innocenti@ndcn.ox.ac.uk`

Abstract

In-context learning (ICL) refers to the ability of a model to learn new tasks from examples in its input without any parameter updates. In contrast to previous theories of ICL relying on toy models and data settings, recently it has been shown that an abstraction of a transformer block can be seen as implicitly updating the weights of its feedforward network according to the context [4]. Here, we provide a simple generalisation of this result for (i) all sequence positions beyond the last, (ii) any transformer block beyond the first, and (iii) more realistic residual blocks including layer normalisation. We empirically verify our theory on simple in-context linear regression tasks and investigate the relationship between the implicit updates related to different tokens within and between blocks. These results help to bring the theory of [4] even closer to practice, with potential for validation on large-scale models.

1 Motivation and main result

Large-scale pretrained models show a remarkable emergent ability to learn new tasks from examples in their input without any fine-tuning or parameter updates. This "in-context learning" (ICL) capability was first noted for GPT-3 [3] and has more recently also been shown by large vision models [2]. For this reason, there has been increasing interest in understanding the mechanisms behind ICL [5, 12], using both empirical and theoretical approaches. While previous theoretical analyses of ICL have relied on simplified models and data settings, recently [4] showed that an abstraction of a transformer block—consisting of a "context-aware" layer such as self-attention [8] and a multi-layer perceptron (MLP)—has the implicit effect of modifying the MLP weights according to the context.

However, among other limitations, the analysis of [4] applies only to the last token and the first transformer block, and their extension to blocks with skip connections does not exactly correspond to the standard Pre-LayerNorm (LN) transformer architecture used in practice [9, 10]. Our main contribution is to generalise the main result of [4] in all these respects, namely for any token, block and more accurate residual blocks including layer normalisation.

Following their setup, we define a *contextual layer* $\mathbf{A} : \mathbb{R}^{d\times N} \to \mathbb{R}^{d\times N}$ as any layer such as self-attention that can process a d-dimensional input sequence of any length N. We ignore multiple sequence batches for simplicity. The contextual layer can take as input either a single query vector $\mathbf{A}(\mathbf{x})$ with $\mathbf{x} \in \mathbb{R}^d$, or also a context sequence in addition to the query $\mathbf{A}(\mathbf{C}, \mathbf{x})$, with $\mathbf{C} \in \mathbb{R}^{d\times(N-1)}$. A *contextual block* is then defined as the stacking of a contextual layer with an MLP, $\mathbf{T}_{\mathbf{W}}(\cdot) = \mathbf{W}'\big(\sigma(\mathbf{W}\mathbf{A}(\cdot)+\mathbf{b})\big) + \mathbf{b}' \in \mathbb{R}^{d\times N}$ with weights $(\mathbf{W}, \mathbf{W}')$, biases $(\mathbf{b}, \mathbf{b}')$ and activation function σ.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

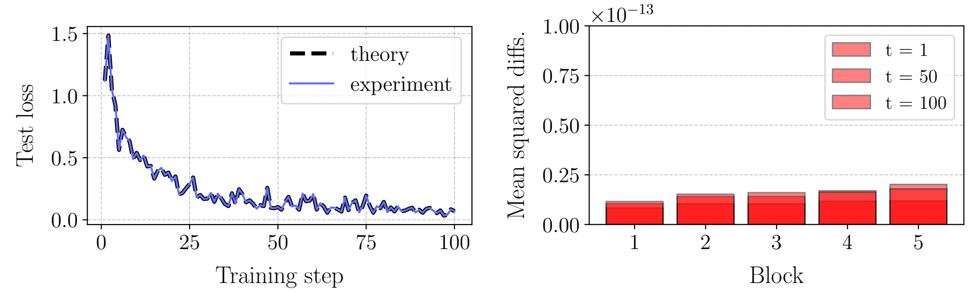

Figure 1: **Empirical verification of Theorem 1 for in-context linear regression.** (*Left*) Test losses of a 5-layer transformer trained to solve linear regression tasks in context (see §A.2 for details). The empirical and theoretical losses were computed using the left- and right-hand side of Eq. 2, respectively, for the last block $\ell = 5$ and token $i = N$. (*Right*) Mean squared differences between the theoretical and empirical predictions (see Eq. 35) of every block at different training steps t. Results were consistent across different random seeds.

Based on these definitions, a simplified version of the main result of [4] can be stated as follows:[1]

$$\mathbf{T_W}(\mathbf{C}, \mathbf{x})_{(N)} = \mathbf{T}_{\mathbf{W}+\Delta \mathbf{W}_N(\mathbf{C})}(\mathbf{x}), \quad \Delta \mathbf{W}_N(\mathbf{C}) = \frac{(\mathbf{W}\Delta \mathbf{A}_{(N)})\mathbf{A}(\mathbf{x})^T}{||\mathbf{A}(\mathbf{x})||^2} \tag{1}$$

where $\Delta \mathbf{A}_{(N)} = \mathbf{A}(\mathbf{C}, \mathbf{x})_{(N)} - \mathbf{A}(\mathbf{x})$ is the difference in the contextual layer's prediction of the last token with and without context, and N indexes the last sequence element, which is left implicit in [4]. Eq. 1 shows that the last-token prediction of a contextual (e.g. transformer) block taking some context and query as input (LHS) is equivalent to that of the same block with only the query as input and the first weight matrix of the MLP updated by the context (RHS). Notably, the implicit weight update $\Delta \mathbf{W}_N(\mathbf{C})$ is of rank one, as the outer product of a column vector and a row vector.

Our generalisation of Eq. 1, given in the following theorem, shows that the prediction of *any* token i by *any* contextual block ℓ with more realistic skip connections including Pre-LN (see §A.1.4 for details) is equivalent to that of the same block with only the previous query as input and specific MLP parameters updated by the context. For any block other than the first, we can think of the inputs $(\mathbf{C}_\ell, \mathbf{x}_\ell)$ as "refined" versions of the original context and query.

Theorem 1. *Consider a contextual block* $\mathbf{T}^\ell_{\mathbf{W},\mathbf{b}'}$ *with skip connections, Pre-LN (as in Eq. 25) and input* $(\mathbf{C}_\ell, \mathbf{x}_\ell)$. *Then, the following equality holds (see §A.1.4 for proof):*

$$\mathbf{T}^\ell_{\mathbf{W},\mathbf{b}'}(\mathbf{C}_\ell, \mathbf{x}_\ell)_{(i)} = \mathbf{T}^\ell_{\mathbf{W}+\Delta\mathbf{W}_i(\mathbf{C}),\mathbf{b}'+\Delta\mathbf{b}'_i(\mathbf{C})}(\mathbf{x}_\ell), \tag{2}$$

where the MLP updates of the first weight matrix and the last layer's biases are given in Eqs. 29 and 30, respectively. The weight update (Eq. 29) is of rank one as in [4].

Following [4] and other previous works [6, 11], we verified our theory on the well-defined problem of in-context linear regression by testing multi-layer transformers to predict sequences of linear functions that were not previously seen during training (see §A.2 for details). Figure 1 shows an excellent match between the theory and experiment. Additional analyses of the implicit weight updates related to different token positions within and between blocks are included in Appendix A.

To conclude, our results help to bring the theory of [4] even closer to practice, potentially allowing for validation on large-scale models. In particular, it would be interesting to analyse the implicit weight updates of models trained on language, which our generalisation enables. Our work is still limited by considering one step of token generation, and it could be important to study ICL settings where the answer is itself a sequence of tokens.

[1] The more general version of the theorem considers any subset of the context $\mathbf{Y} \subset \mathbf{C}$ that may modify $\Delta \mathbf{W}_N(\mathbf{Y})$, which we will ignore for simplicity.

References

[1] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] A. Bar, Y. Gandelsman, T. Darrell, A. Globerson, and A. Efros. Visual prompting via image inpainting. *Advances in Neural Information Processing Systems*, 35:25005–25017, 2022.

[3] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[4] B. Dherin, M. Munn, H. Mazzawi, M. Wunder, and J. Gonzalvo. Learning without training: The implicit dynamics of in-context learning. *arXiv preprint arXiv:2507.16003*, 2025.

[5] Q. Dong, L. Li, D. Dai, C. Zheng, J. Ma, R. Li, H. Xia, J. Xu, Z. Wu, T. Liu, et al. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

[6] S. Garg, D. Tsipras, P. S. Liang, and G. Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in neural information processing systems*, 35:30583–30598, 2022.

[7] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[9] Q. Wang, B. Li, T. Xiao, J. Zhu, C. Li, D. F. Wong, and L. S. Chao. Learning deep transformer models for machine translation. *arXiv preprint arXiv:1906.01787*, 2019.

[10] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T. Liu. On layer normalization in the transformer architecture. In *International conference on machine learning*, pages 10524–10533. PMLR, 2020.

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

[12] Y. Zhou, J. Li, Y. Xiang, H. Yan, L. Gui, and Y. He. The mystery of in-context learning: A comprehensive survey on interpretation and analysis. *arXiv preprint arXiv:2311.00237*, 2023.

A Appendix

Contents

A.1 Proofs and derivations

A.1.1 Extension to all sequence positions

Generalising the main result of [4] (Theorem 2.2) to all output sequence positions (including the last) is simply a matter of indexing. As made explicit by our indexing in Eq. 1, [4] focus only on the last-token prediction of the contextual block $\mathbf{T_W}(\mathbf{C}, \mathbf{x})_{(N)}$. We can therefore relax the result by simply considering any token position i

$$\mathbf{T_W}(\mathbf{C}, \mathbf{x})_{(i)} = \mathbf{T}_{\mathbf{W}+\Delta\mathbf{W}_i(\mathbf{C})}(\mathbf{x}), \quad \Delta\mathbf{W}_i(\mathbf{C}) = \frac{(\mathbf{W}\Delta\mathbf{A}_{(i)})\mathbf{A}(\mathbf{x})^T}{||\mathbf{A}(\mathbf{x})||^2}. \tag{3}$$

where one only needs to index the output of the contextual layer $\Delta\mathbf{A}_{(i)} = \mathbf{A}(\mathbf{C}, \mathbf{x})_{(i)} - \mathbf{A}(\mathbf{x})$. Note that different sequence positions will be associated with different weight updates $\Delta\mathbf{W}_i(\mathbf{C})$ and that each update retains rank 1. This result can also be rewritten in matrix form by stacking all the weight updates related to different positions into a single matrix $\mathbf{B}$

$$\mathbf{T_W}(\mathbf{C}, \mathbf{x}) = \mathbf{T}_{\mathbf{B}+\Delta\mathbf{B}(\mathbf{C})}(\mathbf{x}) \tag{4}$$

where the new matrix and its update are

$$\mathbf{B} = \begin{pmatrix} \mathbf{W} \\ \mathbf{W} \\ \vdots \\ \mathbf{W} \end{pmatrix} \in \mathbb{R}^{(hN)\times d} \quad \text{and} \quad \Delta\mathbf{B}(\mathbf{C}) = \begin{pmatrix} \Delta\mathbf{W}_1(\mathbf{C}) \\ \Delta\mathbf{W}_2(\mathbf{C}) \\ \vdots \\ \Delta\mathbf{W}_N(\mathbf{C}) \end{pmatrix} \in \mathbb{R}^{(hN)\times d} \tag{5}$$

with $\mathbf{W} \in \mathbb{R}^{h\times d}$. It is straightforward to show that the rank of this update matrix $\Delta\mathbf{B}(\mathbf{C})$ is also 1. In particular, $\operatorname{rank}\begin{pmatrix} \Delta\mathbf{W}_1(\mathbf{C}) \\ \Delta\mathbf{W}_2(\mathbf{C}) \\ \vdots \\ \Delta\mathbf{W}_N(\mathbf{C}) \end{pmatrix} \leq N$. However, all the $\Delta\mathbf{W}_i(\mathbf{C})$ can be written by definition as the outer product of a column vector and a row vector $\mathbf{u}_i\mathbf{v}^T$, where $\mathbf{u}_i = \mathbf{W}\Delta\mathbf{A}_{(i)} \in \mathbb{R}^h$ and the same $\mathbf{v} = \mathbf{A}(\mathbf{x}) \in \mathbb{R}^d$ for all i. Hence

$$\Delta\mathbf{B}(\mathbf{C}) = \begin{pmatrix} \Delta\mathbf{W}_1(\mathbf{C}) \\ \Delta\mathbf{W}_2(\mathbf{C}) \\ \vdots \\ \Delta\mathbf{W}_N(\mathbf{C}) \end{pmatrix} = \begin{pmatrix} \mathbf{u}_1\mathbf{v}^T \\ \mathbf{u}_2\mathbf{v}^T \\ \vdots \\ \mathbf{u}_N\mathbf{v}^T \end{pmatrix} = \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_N \end{pmatrix} \mathbf{v}^T,$$

which shows that $\operatorname{rank}(\Delta\mathbf{B}(\mathbf{C})) = 1$.

A.1.2 Extension to any contextual block

Similar to the previous extension to all sequence positions (§A.1.1), the main result of [4] can be generalised to any contextual block beyond the first one by simple iterative application. For any block ℓ other than the first, we can do this by thinking of their inputs $(\mathbf{C}_\ell, \mathbf{x}_\ell)$ as refined versions of the original (unprocessed) input context and query $(\mathbf{C}_1, \mathbf{x}_1)$. Hence

$$\mathbf{T}^\ell_{\mathbf{W}}(\mathbf{C}_\ell, \mathbf{x}_\ell)_{(i)} = \mathbf{T}^\ell_{\mathbf{W}+\Delta\mathbf{W}_i(\mathbf{C}_\ell)}(\mathbf{x}_\ell), \quad \Delta\mathbf{W}_i(\mathbf{C}_\ell) = \frac{(\mathbf{W}\Delta\mathbf{A}^\ell_{(i)})\mathbf{A}^\ell(\mathbf{x}_\ell)^T}{||\mathbf{A}^\ell(\mathbf{x}_\ell)||^2}. \tag{6}$$

where $\mathbf{T}^\ell_{\mathbf{W}}$ and $\mathbf{A}^\ell$ indicate the ℓth contextual block and layer, respectively. Note that, as expected, Eq. 6 simplifies to Eq. 1 for $i = N$ and $\ell = 1$.

A.1.3 Extension to any block with more accurate skip connections

Motivated by the Pre-LN architecture [9, 10], [4] consider blocks with the following skip connections:

$$\mathbf{T}(\mathbf{C}, \mathbf{x})_{(N)} = \mathbf{x} + \mathbf{A}(\mathbf{C}, \mathbf{x})_{(N)} + \mathbf{W}'\sigma\big(\mathbf{W}\mathbf{A}(\mathbf{C}, \mathbf{x})_{(N)} + \mathbf{b}\big) + \mathbf{b}', \tag{7}$$

However, this fails to input the full contextual layer's output into the MLP, specifically the input skip $\mathbf{x}$. The more exact block structure would be

$$\mathbf{T}(\mathbf{C}, \mathbf{x})_{(N)} = \mathbf{x} + \mathbf{A}(\mathbf{C}, \mathbf{x})_{(N)} + \mathbf{W}'\sigma\Big(\mathbf{W}\big(\mathbf{A}(\mathbf{C}, \mathbf{x})_{(N)} + \mathbf{x}\big) + \mathbf{b}\Big) + \mathbf{b}' \tag{8}$$

where now the MLP is also fed the input skip $\mathbf{x}$. To lay the groundwork for the proof of Theorem 1, we proceed in 3 main steps. First, using the same logic as in §A.1.1, we extend Eq. 8 to any token position i

$$\mathbf{T}(\mathbf{C}, \mathbf{x})_{(i)} = (\mathbf{C}, \mathbf{x})_{(i)} + \mathbf{A}(\mathbf{C}, \mathbf{x})_{(i)} + \mathbf{W}'\sigma\Big(\mathbf{W}\big(\mathbf{A}(\mathbf{C}, \mathbf{x})_{(i)} + (\mathbf{C}, \mathbf{x})_{(i)}\big) + \mathbf{b}\Big) + \mathbf{b}' \tag{9}$$

where note that the input skip is equal to the query vector $(\mathbf{C}, \mathbf{x})_{(i)} = \mathbf{x}$ for the last position $i = N$. Second, we prove Theorem 1 without layer normalisation for the first block, which can be stated as follows:

$$\mathbf{T}_{\mathbf{W},\mathbf{b}'}(\mathbf{C}, \mathbf{x})_{(i)} = \mathbf{T}_{\mathbf{W}_i(\mathbf{C}),\mathbf{b}'_i(\mathbf{C})}(\mathbf{x}) \tag{10}$$

where the updates of the first MLP weight matrix and the biases of the last layer are given by

$$\Delta\mathbf{W}_i(\mathbf{C}) = \frac{\big(\mathbf{W}(\Delta\mathbf{A}_{(i)} + \Delta\mathbf{z}_{(i)})\big)\big(\mathbf{A}(\mathbf{x}) + \mathbf{x}\big)^T}{||\mathbf{A}(\mathbf{x}) + \mathbf{x}||^2} \quad \text{and} \tag{11}$$

$$\Delta\mathbf{b}'_i(\mathbf{C}) = \Delta\mathbf{A}_{(i)} + \Delta\mathbf{z}_{(i)}, \tag{12}$$

with $\Delta\mathbf{z}_{(i)} = (\mathbf{C}, \mathbf{x})_{(i)} - \mathbf{x}$ as the difference between any input element and the query. The result now follows by direct computation as in [4]. Let $\mathbf{W}_i(\mathbf{C}) = \mathbf{W} + \Delta\mathbf{W}_i(\mathbf{C})$ and $\mathbf{b}'_i(\mathbf{C}) = \mathbf{b}' + \Delta\mathbf{b}'_i(\mathbf{C})$. Then by definition, the right-hand side of Eq. 10 is

$$\mathbf{T}_{\mathbf{W}_i(\mathbf{C}),\mathbf{b}'_i(\mathbf{C})}(\mathbf{x}) = \mathbf{x} + \mathbf{A}(\mathbf{x}) + \mathbf{W}'\sigma\Big(\big(\mathbf{W} + \Delta\mathbf{W}_i(\mathbf{C})\big)\big(\mathbf{A}(\mathbf{x}) + \mathbf{x}\big) + \mathbf{b}\Big) + \mathbf{b}' + \Delta\mathbf{b}'_i(\mathbf{C}) \tag{13}$$

$$= \mathbf{x} + \mathbf{A}(\mathbf{x}) + \Delta\mathbf{b}'_i(\mathbf{C}) + \mathbf{W}'\sigma\Big(\big(\mathbf{W} + \Delta\mathbf{W}_i(\mathbf{C})\big)\big(\mathbf{A}(\mathbf{x}) + \mathbf{x}\big) + \mathbf{b}\Big) + \mathbf{b}', \tag{14}$$

where the second line simply moves the update of the last layer's biases for later convenience. Substituting $\Delta\mathbf{W}_i(\mathbf{C})$ (Eq. 11) and using the fact that $\frac{\mathbf{x}^T}{||\mathbf{x}||^2}\mathbf{x} = 1$, we obtain

$$\Delta\mathbf{W}_i(\mathbf{C})\big(\mathbf{A}(\mathbf{x}) + \mathbf{x}\big) = \frac{\big(\mathbf{W}(\Delta\mathbf{A}_{(i)} + \Delta\mathbf{z}_{(i)})\big)\big(\mathbf{A}(\mathbf{x}) + \mathbf{x}\big)^T}{||\big(\mathbf{A}(\mathbf{x}) + \mathbf{x}\big)||^2}\big(\mathbf{A}(\mathbf{x}) + \mathbf{x}\big) \tag{15}$$

$$= \mathbf{W}(\Delta\mathbf{A}_{(i)} + \Delta\mathbf{z}_{(i)}), \tag{16}$$

which gives

$$\mathbf{T}_{\mathbf{W}_i(\mathbf{C}),\mathbf{b}'_i(\mathbf{C})}(\mathbf{x}) = \mathbf{x} + \mathbf{A}(\mathbf{x}) + \Delta\mathbf{b}'_i(\mathbf{C}) + \mathbf{W}'\sigma\Big(\mathbf{W}\big(\mathbf{A}(\mathbf{x}) + \mathbf{x} + \Delta\mathbf{A}_{(i)} + \Delta\mathbf{z}_{(i)}\big) + \mathbf{b}\Big) + \mathbf{b}'. \tag{17}$$

By the above definitions, we have that $\mathbf{A}(\mathbf{x}) + \Delta\mathbf{A}_{(i)} = \mathbf{A}(\mathbf{C}, \mathbf{x})_{(i)}$ and $\mathbf{x} + \Delta\mathbf{z}_{(i)} = (\mathbf{C}, \mathbf{x})_{(i)}$. Hence,

$$\mathbf{T}_{\mathbf{W}_i(\mathbf{C}), \mathbf{b}'_i(\mathbf{C})}(\mathbf{x}) = \mathbf{x} + \mathbf{A}(\mathbf{x}) + \Delta\mathbf{b}'_i(\mathbf{C}) + \mathbf{W}'\sigma\Big(\mathbf{W}\big(\mathbf{A}(\mathbf{C}, \mathbf{x})_{(i)} + (\mathbf{C}, \mathbf{x})_{(i)}\big) + \mathbf{b}\Big) + \mathbf{b}'. \tag{18}$$

Finally, by definition of $\Delta\mathbf{b}'_i(\mathbf{C})$, we obtain

$$\mathbf{T}_{\mathbf{W}_i(\mathbf{C}), \mathbf{b}'_i(\mathbf{C})}(\mathbf{x}) = (\mathbf{C}, \mathbf{x})_{(i)} + \mathbf{A}(\mathbf{C}, \mathbf{x})_{(i)} + \mathbf{W}'\sigma\Big(\mathbf{W}\big(\mathbf{A}(\mathbf{C}, \mathbf{x})_{(i)} + (\mathbf{C}, \mathbf{x})_{(i)}\big) + \mathbf{b}\Big) + \mathbf{b}' \tag{19}$$

$$= \mathbf{T}_{\mathbf{W}, \mathbf{b}'}(\mathbf{C}, \mathbf{x})_{(i)} \tag{20}$$

The last step is to extend, following §A.1.2, the result to any contextual block ℓ

$$\mathbf{T}^\ell(\mathbf{C}_\ell, \mathbf{x}_\ell)_{(i)} = (\mathbf{C}_\ell, \mathbf{x}_\ell)_{(i)} + \mathbf{A}^\ell(\mathbf{C}_\ell, \mathbf{x}_\ell)_{(i)} + \mathbf{W}'\sigma\Big(\mathbf{W}\big(\mathbf{A}^\ell(\mathbf{C}_\ell, \mathbf{x}_\ell)_{(i)} + (\mathbf{C}_\ell, \mathbf{x}_\ell)_{(i)}\big) + \mathbf{b}\Big) + \mathbf{b}' \tag{21}$$

where we simply index the block $\mathbf{T}^\ell$, layer $\mathbf{A}^\ell$ and input $(\mathbf{C}_\ell, \mathbf{x}_\ell)$, with $(\mathbf{C}_1, \mathbf{x}_1)$ as the original context and query. The parameter updates now become

$$\Delta\mathbf{W}_i(\mathbf{C}_\ell) = \frac{\big(\mathbf{W}(\Delta\mathbf{A}^\ell_{(i)} + \Delta\mathbf{z}^\ell_{(i)})\big)\big(\mathbf{A}^\ell(\mathbf{x}_\ell) + \mathbf{x}_\ell\big)^T}{||\mathbf{A}^\ell(\mathbf{x}_\ell) + \mathbf{x}_\ell||^2} \quad \text{and} \tag{22}$$

$$\Delta\mathbf{b}'_i(\mathbf{C}_\ell) = \Delta\mathbf{A}^\ell_{(i)} + \Delta\mathbf{z}^\ell_{(i)}, \tag{23}$$

with $\Delta\mathbf{z}^\ell_{(i)} = (\mathbf{C}_\ell, \mathbf{x}_\ell)_{(i)} - \mathbf{x}_\ell$. By exactly the same computation as above, this leads to

$$\mathbf{T}^\ell_{\mathbf{W}_i(\mathbf{C}_\ell), \mathbf{b}'_i(\mathbf{C}_\ell)}(\mathbf{x}_\ell) = \mathbf{T}^\ell_{\mathbf{W}, \mathbf{b}'}(\mathbf{C}_\ell, \mathbf{x}_\ell)_{(i)} \tag{24}$$

which proves Theorem 1 without layer normalisation.

A.1.4 Generalisation to any block with skips and layer normalisations (Theorem 1)

Building on the previous results in §A.1.1-A.1.3, here we prove Theorem 1 in its most general form, namely for any token and block including both skip connections and Pre-LN. Omitting the layer index ℓ for simplicity, the Pre-LN contextual block is given by

$$\begin{aligned}\mathbf{T}(\mathbf{C}, \mathbf{x})_{(i)} = (\mathbf{C}, \mathbf{x})_{(i)} &+ \mathbf{A}\big(\operatorname{LN}(\mathbf{C}, \mathbf{x})\big)_{(i)} \\ &+ \mathbf{W}'\sigma\Big(\mathbf{W}\operatorname{LN}'\big[\mathbf{A}\big(\operatorname{LN}(\mathbf{C}, \mathbf{x})\big)_{(i)} + (\mathbf{C}, \mathbf{x})_{(i)}\big] + \mathbf{b}\Big) + \mathbf{b}'\end{aligned} \tag{25}$$

where recall that layer normalisation (LN) [1] of some input vector $\mathbf{x}$ is given by $\operatorname{LN}(\mathbf{x}) = \boldsymbol{\gamma} \odot \frac{\mathbf{x} - \mathbb{E}[\mathbf{x}]}{\sqrt{\operatorname{Var}[\mathbf{x}] + \epsilon}} + \boldsymbol{\beta}$, with some optional learnable factors $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$. Similar to the MLP weight matrices and biases, $\operatorname{LN}(\cdot)$ and $\operatorname{LN}'(\cdot)$ indicate the pre-attention and pre-MLP layer normalisations, respectively. As in §A.1.3, we want to prove that there exist some MLP parameter updates such that

$$\mathbf{T}^\ell_{\mathbf{W}, \mathbf{b}'}(\mathbf{C}_\ell, \mathbf{x}_\ell)_{(i)} = \mathbf{T}^\ell_{\mathbf{W} + \Delta\mathbf{W}_i(\mathbf{C}_\ell), \mathbf{b}' + \Delta\mathbf{b}'_i(\mathbf{C}_\ell)}(\mathbf{x}_\ell) \tag{26}$$

for any token position i and block ℓ. To derive the updates, we first show some more general expressions that recover all the previous cases. Specifically, the general weight update is given by

$$(\mathbf{W} + \Delta\mathbf{W}_i)\mathbf{f} = \mathbf{W}\mathbf{g}_i \implies \Delta\mathbf{W}_i = \frac{\mathbf{W}(\mathbf{g}_i - \mathbf{f})\mathbf{f}^T}{||\mathbf{f}||^2} \tag{27}$$

where $\mathbf{g}_i$ and $\mathbf{f}$ are the *full input to MLP with and without context*, respectively. For example, in the case of no skip connections, Eq. 27 reduces to the result of [4] (Eq. 1 for $i = N$) where $\mathbf{g}_i = \mathbf{A}(\mathbf{C}, \mathbf{x})_{(i)}$ and $\mathbf{f} = \mathbf{A}(\mathbf{x})$, hence $\mathbf{g}_i - \mathbf{f} = \Delta\mathbf{A}_{(i)}$. In this case, the difference in the MLP input with and without context coincides with that of the contextual layer's output (with and without context). Note also that Eq. 27 shows that the implicit weight update has rank 1 for any $\mathbf{g}_i$ and $\mathbf{f}$.

If we add skip connections as in §A.1.3 (Eq. 9), Eq. 27 reduces to the derived update of Eq. 22, where $\mathbf{g}_i = \mathbf{A}(\mathbf{C}, \mathbf{x})_{(i)} + (\mathbf{C}, \mathbf{x})_{(i)}$ and $\mathbf{f} = \mathbf{A}(\mathbf{x}) + \mathbf{x}$, hence $\mathbf{g}_i - \mathbf{f} = \Delta\mathbf{A}_{(i)} + \Delta\mathbf{z}_{(i)}$. Note that now the input to the MLP with and without context no longer coincides with that of the contextual layer's output and also includes a skip connection delta $\Delta \mathbf{z}_{(i)} = (\mathbf{C}, \mathbf{x})_{(i)} - \mathbf{x}$. In this case, as shown in §A.1.3, we also need an implicit update for the biases of the last MLP layer

$$\Delta \mathbf{b}'_i = \mathbf{q}_i - \mathbf{p} \tag{28}$$

where $\mathbf{q}_i$ and $\mathbf{p}$ are the *full output of the contextual layer* (including the skip) *with and without context*, respectively. In this case, they reduce to the input to the MLP (with and without context) and the derived update of Eq. 23, namely $\mathbf{q}_i - \mathbf{p} = \mathbf{g}_i - \mathbf{f} = \Delta \mathbf{A}_{(i)} + \Delta \mathbf{z}_{(i)}$.

Finally, if we consider a Pre-LN contextual block as in Eq. 25, the general weight update of Eq. 27 leads to

$$\Delta \mathbf{W}_i(\mathbf{C}) = \frac{\Big(\mathbf{W}\big(\overbrace{\mathrm{LN}'\left[\mathbf{A}\big(\mathrm{LN}(\mathbf{C},\mathbf{x})\big)_{(i)} + (\mathbf{C},\mathbf{x})_{(i)}\right]}^{\mathbf{g}_i} - \overbrace{\mathrm{LN}'\left[\mathbf{A}\big(\mathrm{LN}(\mathbf{x})\big) + \mathbf{x}\right]}^{\mathbf{f}}\big)\Big)\big(\overbrace{\mathrm{LN}'\left[\mathbf{A}\big(\mathrm{LN}(\mathbf{x})\big) + \mathbf{x}\right]}^{\mathbf{f}}\big)^T}{\Big|\Big|\underbrace{\mathrm{LN}'\left[\mathbf{A}\big(\mathrm{LN}(\mathbf{x})\big) + \mathbf{x}\right]}_{\mathbf{f}}\Big|\Big|^2} \tag{29}$$

where note that now the difference in the full input to the MLP $\mathbf{g}_i - \mathbf{f}$ (including the input skip and LNs) does not simplify because of the nonlinear, nested LNs. The general update for the last layer's biases of Eq. 28 gives

$$\begin{aligned}\Delta \mathbf{b}'_i(\mathbf{C}) &= \big[\underbrace{\mathbf{A}\big(\mathrm{LN}(\mathbf{C},\mathbf{x})\big)_{(i)} + (\mathbf{C},\mathbf{x})_{(i)}}_{\mathbf{q}_i}\big] - \big[\underbrace{\mathbf{A}\big(\mathrm{LN}(\mathbf{x})\big) + \mathbf{x}}_{\mathbf{p}}\big] \\ &= \Delta \mathbf{A}_{(i)} + \Delta \mathbf{z}_{(i)}.\end{aligned} \tag{30}$$

Note (i) that now $\mathbf{q}_i \neq \mathbf{g}_i$ and $\mathbf{p} \neq \mathbf{f}$ because of the second (pre-MLP) LN, and (ii) that the update is the same as that without LN except that the difference in the contextual layer's output now includes the first (pre-attention) LN, i.e. $\Delta \mathbf{A}_{(i)} = \mathbf{A}\big(\mathrm{LN}(\mathbf{C},\mathbf{x})\big)_{(i)} - \mathbf{A}\big(\mathrm{LN}(\mathbf{x})\big)$. Using these updates (Eqs. 29-30), it can be shown by direct computation as in §A.1.3 that

$$\mathbf{T}^{\ell}_{\mathbf{W},\mathbf{b}'}(\mathbf{C}_\ell, \mathbf{x}_\ell)_{(i)} = \mathbf{T}^{\ell}_{\mathbf{W}+\Delta\mathbf{W}_i(\mathbf{C}_\ell),\mathbf{b}'+\Delta\mathbf{b}'_i(\mathbf{C}_\ell)}(\mathbf{x}_\ell) \tag{31}$$

for any token position i and block ℓ, which concludes the proof. This particular equality for Pre-LN blocks is empirically verified in Figure A.3. Note that, as in §A.1.1, we can rewrite the result more compactly in matrix-vector form:

$$\mathbf{T}^{\ell}_{\mathbf{W},\mathbf{b}'}(\mathbf{C}_\ell, \mathbf{x}_\ell) = \mathbf{T}^{\ell}_{\mathbf{B}+\Delta\mathbf{B}(\mathbf{C}_\ell),\mathbf{e}+\Delta\mathbf{e}(\mathbf{C}_\ell)}(\mathbf{x}_\ell) \tag{32}$$

where the stacked weight matrices $\mathbf{B}$ and their updates $\Delta\mathbf{B}(\mathbf{C}_\ell)$ are given in Eq. 5 for the first block but can be similarly extended to any block, while all the biases $\mathbf{e}$ and their updates $\Delta\mathbf{e}(\mathbf{C}_\ell)$ are concatenated as follows

$$\mathbf{e} = \begin{pmatrix} \mathbf{b}' \\ \mathbf{b}' \\ \vdots \\ \mathbf{b}' \end{pmatrix} \in \mathbb{R}^{hN} \quad \text{and} \quad \Delta\mathbf{e}(\mathbf{C}_\ell) = \begin{pmatrix} \Delta\mathbf{b}_1(\mathbf{C}_\ell) \\ \Delta\mathbf{b}_2(\mathbf{C}_\ell) \\ \vdots \\ \Delta\mathbf{b}_N(\mathbf{C}_\ell) \end{pmatrix} \in \mathbb{R}^{hN}. \tag{33}$$

Note that $\Delta\mathbf{B}(\mathbf{C}_\ell)$ retains rank 1.

A.2 Experimental details

We trained transformers to learn linear regression tasks in context, following [4, 6, 11]. This involved exposing the model to a sequence of input-output pairs from linear functions at training time, and testing it on previously unseen linear functions at inference time. An example sequence consists of $(\mathbf{x}_1, h_b(\mathbf{x}_1), \ldots, \mathbf{x}_{N-1}, h_b(\mathbf{x}_{N-1}), \mathbf{x}_{\text{query}})$, where $\mathbf{x}_i \sim \mathcal{N}(0, \mathbf{I}_{d_x}) \in \mathbb{R}^{d_x}$ and $h_b(\mathbf{x}_i) = \langle \mathbf{w}, \mathbf{x}_i \rangle$ with $\mathbf{w} \sim \mathcal{N}(0, \mathbf{I}_{d_x}) \in \mathbb{R}^{d_x}$. Note that one function (or task) b is sampled for every N inputs. For input to the model, all the input-output pairs are concatenated along the d_x dimension as in [4], such that the input or embedding matrix is $(\mathbf{C}, \mathbf{x}) \in \mathbb{R}^{(d_x+1)\times N}$, with $(\mathbf{C}, \mathbf{x})_{(N)} = [\mathbf{x}_{\text{query}}, 0]^T$.[2]

[2] As a small side note, [4] write the context as having length N, leading to a $N+1$ sequence. We use an $N-1$ context to keep the notation compact when indexing the last token.

Transformers were trained to minimise the last-token prediction error over a batch of tasks

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2B}\sum_{b=1}^{B} ||y_b - f_{\boldsymbol{\theta}}(\mathbf{C}, \mathbf{x})_{(b,d,N)}||^2 \tag{34}$$

where $y_b = h_b(\mathbf{x}_{\text{query}})$ and $\hat{y}_b = f_{\boldsymbol{\theta}}(\mathbf{C}, \mathbf{x})_{(b,d,N)}$ indicates the model prediction of the last token over the last input (target) dimension.

For the results of Figure 1, we used batch size $B = 128$, sequence length $N = 51$ and input dimension $d_x = 2$. The transformers had $L = 5$ residual blocks, each composed of a causal attention layer with 3 heads followed by a standard 2-layer MLP with GeLU as activation function. All models were trained for 100 steps using Adam [7] with learning rate $\eta = 5e^{-2}$. The mean squared differences (MSDs) reported in Figure 1 were computed using

$$\text{MSD} = \frac{1}{BNd}\sum_{b=1}^{B}\sum_{i=1}^{N} ||\mathbf{T}^{\ell}_{\mathbf{W},\mathbf{b}'}(\mathbf{C}_\ell, \mathbf{x}_\ell)_{(b,i)} - \mathbf{T}^{\ell}_{\mathbf{W}+\Delta\mathbf{W}_i(\mathbf{C}),\mathbf{b}'+\Delta\mathbf{b}'_i(\mathbf{C})}(\mathbf{x}_\ell)_{(b)}||^2 \tag{35}$$

for every block $\ell = 1, \dots, L$. This is simply a measure of the deviation of the theoretical predictions from the empirical ones averaged over B batches, N sequence positions and d input dimensions. Every run was repeated for different random seeds to ensure consistency. Code to reproduce all the results will be made publicly available upon publication of this work. All experiments were run on a CPU.

A.3 Alignment of implicit weight updates

Given our result that different token positions i (as well as blocks ℓ) are associated with different implicit weight updates (Eq. 2), we investigated their relationship. The experimental setup was the same as in Figure 1. As a metric of the "directional alignment" (DA) between any two weight updates $\Delta\mathbf{W}_i(\mathbf{C})$ and $\Delta\mathbf{W}_j(\mathbf{C})$, we computed their normalised Frobenius inner product

$$\text{DA}(\Delta\mathbf{W}_i, \Delta\mathbf{W}_j) = \frac{\langle \Delta\mathbf{W}_i, \Delta\mathbf{W}_j\rangle_F}{||\Delta\mathbf{W}_i||_F||\Delta\mathbf{W}_j||_F}, \tag{36}$$

where $\langle \mathbf{A}, \mathbf{B}\rangle_F = \text{Tr}(\mathbf{A}^T\mathbf{B})$. We first investigated the alignment between the updates related to different tokens across blocks. We find that, for a given task sequence b, the structure of the tokens' alignment appears qualitatively consistent across blocks (Figures A.1, A.7 & A.8).

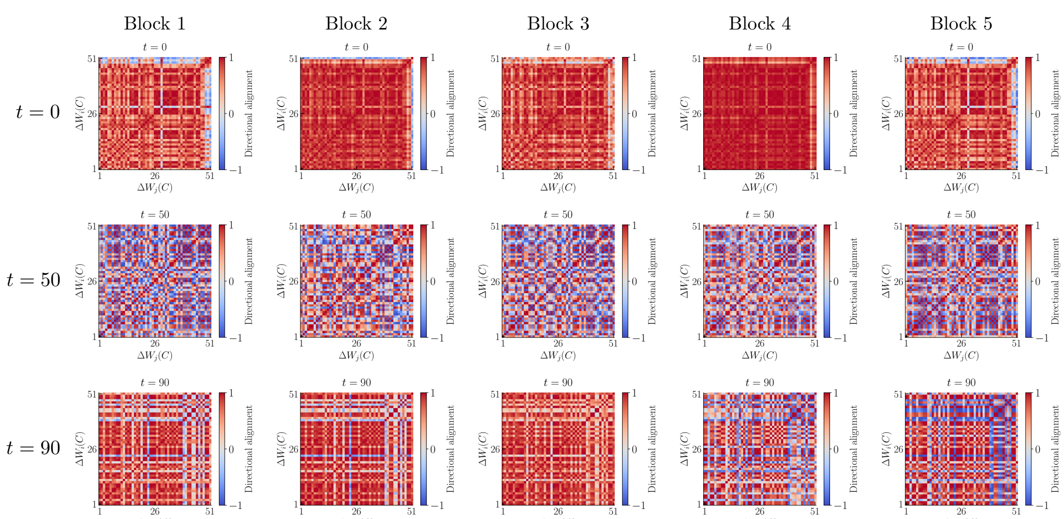

Figure A.1: **The alignment of the implicit weight updates related to different tokens has a qualitatively consistent structure across blocks.** Directional alignment (Eq. 36) between weight updates associated with different sequence positions $\text{DA}(\Delta\mathbf{W}_i, \Delta\mathbf{W}_j)$ for $i = j = 1, \dots, N$ for each block, at different steps in training. See also Figures A.7 & A.8 for other example tasks.

However, the alignment of the weight update related to only the last token of different blocks showed no particular structure at any point during training (Figure A.2).

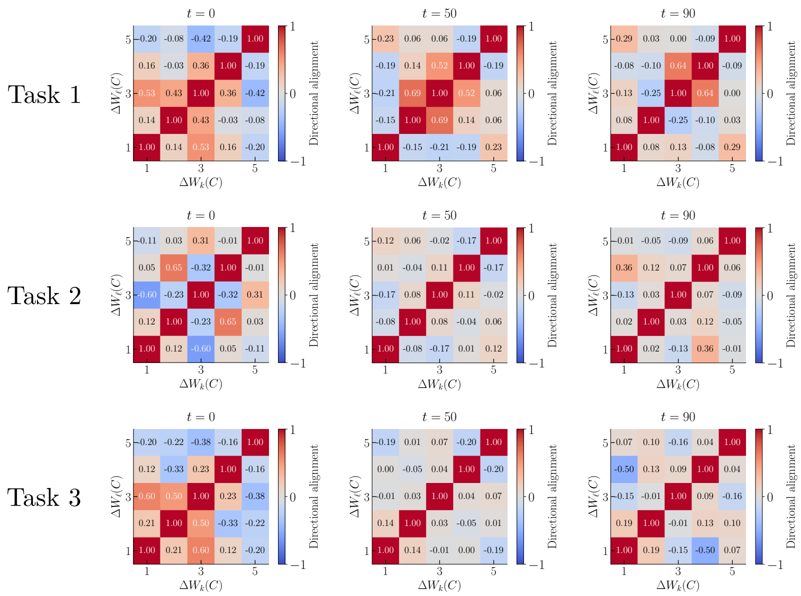

Figure A.2: **The alignment of the implicit weight update related to the last token does not share a consistent structure between blocks.** Directional alignment (Eq. 36) between weight updates associated with the last sequence element of different blocks $\text{DA}(\Delta \mathbf{W}_N^\ell, \Delta \mathbf{W}_N^l)$ for $\ell = k = 1, \ldots, L$ for different tasks b, at different training steps t. Results were consistent across different random seeds.

A.4 Supplementary figures

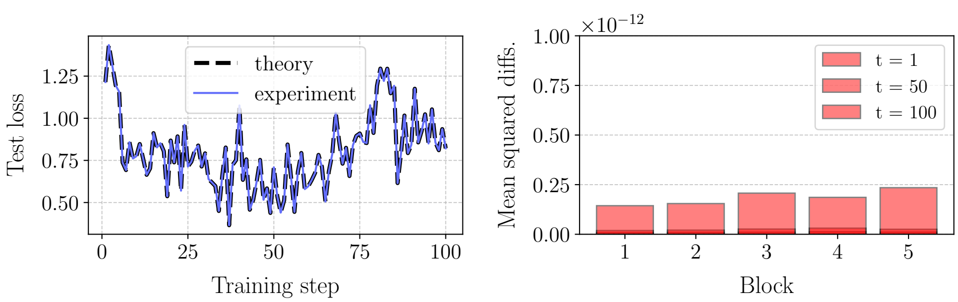

Figure A.3: **Empirical verification of Theorem 1 for Pre-LN transformer blocks.** We plot the same metrics as in Figure 1 for a transformer with layer normalisation (Pre-LN), with all other hyperparameters held constant. Strangely, we found that it was more challenging to obtain good generalisation performance on in-context linear regression tasks with LN for many different hyperparameters. However, it should be noted that the Pre-LN architecture remains the standard for most large language models.

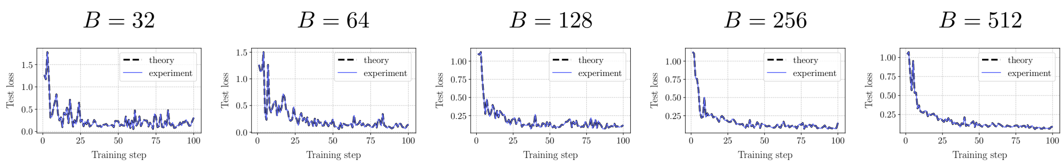

Figure A.4: **Increasing the number of tasks B makes learning easier.** Empirical vs theoretical test losses on the same task as in Figure 1, varying the number of tasks B (i.e. number of sequences of linear functions), while holding all other hyperparameters constant.

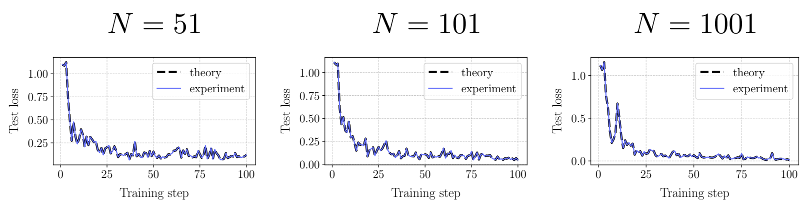

Figure A.5: **Increasing the input sequence length N facilitates learning.** Empirical vs theoretical test losses on the same task as in Figure 1, varying the data sequence length N, while holding all other hyperparameters constant.

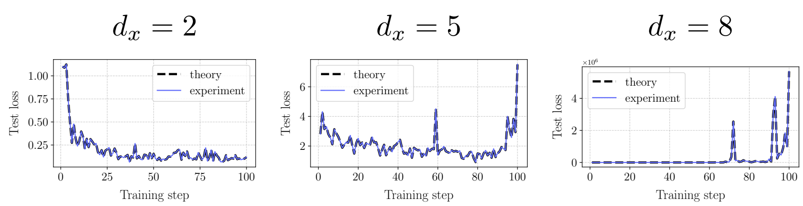

Figure A.6: **Increasing the input dimensionality d_x makes learning more challenging.** Empirical vs theoretical test losses on the same task as in Figure 1, varying the input dimension d_x (i.e. number of regression coefficients), while holding all other hyperparameters constant.

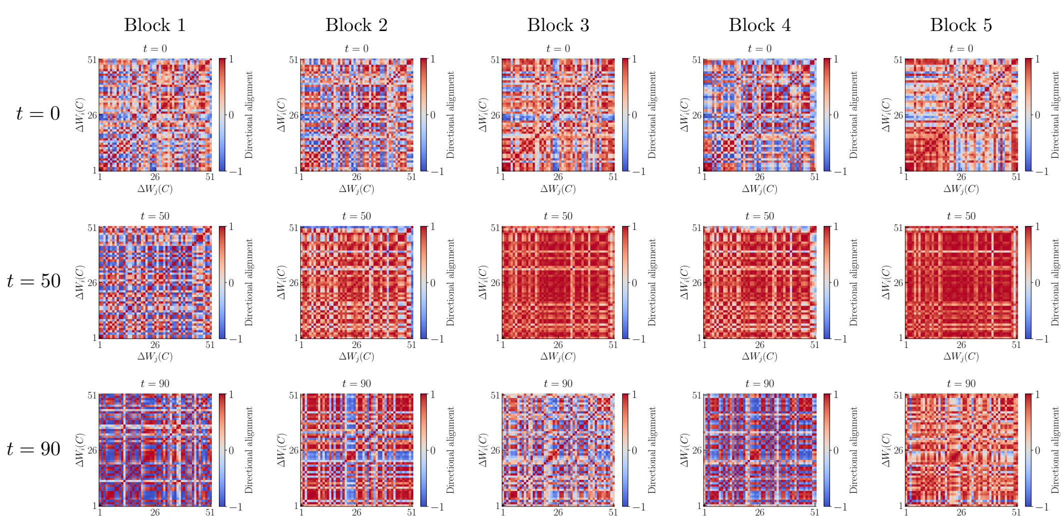

Figure A.7: **Same results as Figure A.1 for a different example task or input sequence.**

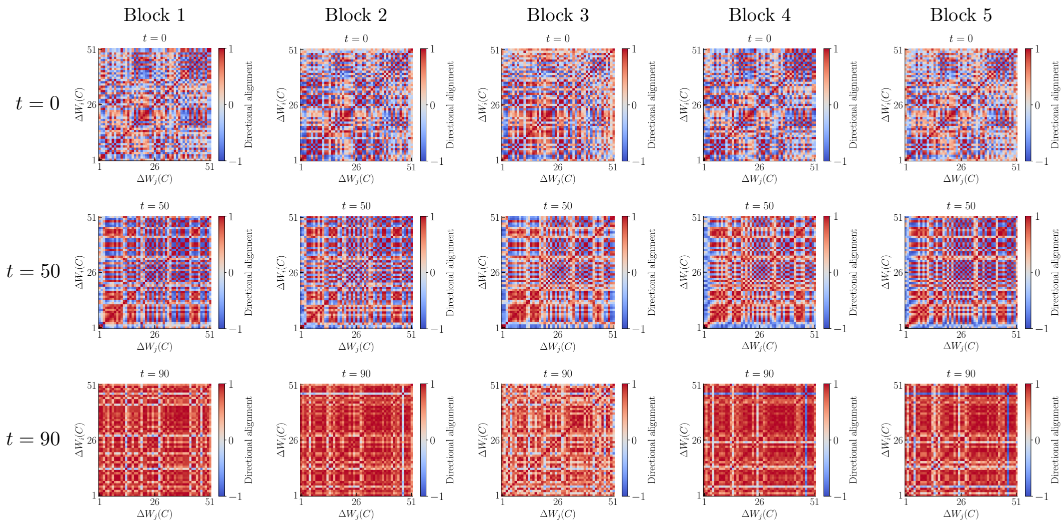

Figure A.8: **Same results as Figures A.1 and A.7 for yet another example task.**

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our claims are clearly stated in the abstract and introduction and are verified by experiments.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The main limitation of our work is stated in the conclusion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Complete proofs are provided in Appendix A.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide details needed to reproduce all the experimental results in §A.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code used to reproduce all the experimental results will be released upon publication of this work.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify important details needed to reproduce and understand the experiments in §A.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Where relevant, we do not report error bars because results did not significantly vary across different random seeds or runs, as we state in the captions of relevant figures.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All of our experiments were run on a single CPU, as stated in §A.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: We see no potential positive or negative societal impact of the work since the models tested are too simple for modern AI applications.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.