Incremental Learning in Transformers for In-Context Associative Recall

Anonymous Author(s)
Affiliation
Address
email

Abstract

Transformers acquire in-context learning abilities in abrupt phases during training, often unfolding over multiple stages, during which certain keys circuits like induction heads emerge. In this work, we characterize the training dynamics behind the emergence of such circuits during these stages. We focus on a synthetic in-context associative recall task, where sequences are drawn from random maps between a permutation group and a vocabulary range and the model is required to complete the mapping of a permutation by retrieving it from the context. On this task, we study the trajectories of gradient flow of a simplified two-layer, attention-only transformer. Leveraging symmetries in both the transformer architecture and the data, we derive conservation laws that guide the dynamics of the parameters. These conservation laws crucially reveal how initialization —both in shape and scale— determines the order of learning as well as the timescales over which such circuits emerge revealing the implicit curriculum. Furthermore, at the limit of vanishing scale of initialization, we characterize the trajectory of the gradient flow revealing how the training jumps from one saddle to another.

1 Introduction

In-context learning (ICL) [11], the ability of a model to perform new tasks from examples provided in its prompt without parameter updates, is a characteristic ability of language models. Beyond what these models can do in context, how these abilities emerge during training remains poorly understood. Empirical works [30, 12] report long plateaus in the training loss followed by abrupt transitions, after which specific circuits, such as induction heads, become functional. Understanding these training dynamics is essential both for theory (which optimization biases make ICL learnable by gradient descent) and for practice (how hyperparameters impact convergence speed and training stability).

While recent analyses [28, 13] have advanced understanding, they fall short of a complete explanation. Approaches based on layerwise training [28] or highly simplified architectures [41] (e.g., linear attentions) have crucially clarified isolated aspects of the phenomenon, but struggle to account for the sequential acquisition of partial solutions and the duration of plateau phases observed in full models. In particular, we lack a theory that predicts the order in which partial circuits appear, and that explains what controls the length of each phase, including sensitivity to initialization scale.

In this paper, we propose combining optimization dynamics with mechanistic interpretability to study how circuits emerge during training. Our analysis is purely dynamical, we study the training trajectories induced by gradient-based optimization, yet our conclusions are mechanistic: we identify which circuit is implemented, which sub-circuits appear first, and how the full circuit crystallizes.

To render the problem analytically tractable while preserving its essential structure, we introduce a simplified recall task that retains the induction mechanism underlying in-context n-gram learning but in a form that is more amenable to analysis. Crucially, we couple this task with a series of principled simplifications, that isolate the essential components of the transformer responsible for incremental learning, while removing spurious elements that obscure analysis. Our analysis reveals a staged learning process: training trajectories encounter intermediate, partially correct solutions where gradients nearly vanish (training plateaus) before transitioning to higher-order solutions.

Contributions. Our results (i) formalize a in-context recall task that preserves the induction structure while enabling tractable analysis, (ii) derive training dynamics that exhibit plateaus aligned with sub-circuits, and (iii) provide quantitative predictions for phase ordering and lengths as functions of model and optimization parameters, with empirical validation on small transformers trained end-to-end.

2 Problem Setting

2.1 In-Context Associative Recall Task

In-context learning abilities of transformers are driven by certain key circuits, such as induction heads. A basic induction head circuit [30] learns to complete simple repeating patterns, e.g., $[A][B][C]\dots[A]$. When the model encounters the second $[A]$, the circuit attends to the first $[A]$ and predicts the subsequent token, $[B]$. This paper investigates how models handle a more general version of this pattern: $[A][B][C][X]...[A][B][C]$. In this setting, the model must recognize the entire sequence $[A][B][C]$, locate its previous occurrence in the context, and then use it to predict the next token $[X]$.

Formally, the task is defined as follows. The model must complete a sequence by matching the last k tokens, where $k > 1$ is the *task order*. Let $\mathcal{P}_k$ denote the set of all permutations of $\{0, 1, \dots, k-1\}$, indexed as $\pi_1, \pi_2, \dots, \pi_{k!}$, where each π_i is a string of k numbers. Let $\mathcal{R}$ be the set of possible responses. The task is defined by a function $f : \mathcal{P}_k \to \mathcal{R}$, sampled from a uniform distribution $\mathcal{D}(\mathcal{F})$ over the set of all such functions $\mathcal{F} = \{f \mid f : \mathcal{P}_k \to \mathcal{R}\}$. An input sequence is then generated by first sampling $q \in \mathcal{P}_k$ uniformly at random, and independently sampling a function f_τ from $\mathcal{D}(\mathcal{F})$. The final input sequence takes the form:

$$\underbrace{\pi_{(1,0)}, \pi_{(1,1)}, \dots, \pi_{(1,k-1)}}_{\pi_1},\ f_\tau(\pi_1),\ \underbrace{\pi_{(2,0)}, \dots, \pi_{(2,k-1)}}_{\pi_2},\ f_\tau(\pi_2),\ \dots, \underbrace{q_0, \dots, q_{k-1}}_{q_{\mathcal{M}-1}},\ ?.$$

Note that $\pi_1, \pi_2, \dots$ each represent a sequence of k tokens, rather than a single token. For example, $\pi_1 = 0, 1, \dots, k-1$. Figure **??** illustrates the task for $k = 2$ with a response set $\{A, B\}$. We define the vocabulary as $\mathcal{S} = [k] \cup \mathcal{R}$. Each sequence has a fixed length of $l = (k+1)! + k$. To solve this task, the transformer must learn to identify the part of the context that matches the final k tokens and recall the subsequent token. For completeness, the context contains all possible query permutations, ensuring that the model can always retrieve the correct response, which enables exact learning.

2.2 Multi-headed Attention-Only Transformer

We analyze a specific attention-only transformer with a two-layer structure. The first layer contains k attention heads, and the second layer contains a single head. The architecture is based on the disentangled transformer [15, 28] and incorporates several simplifications from prior work [28, 14].

Token encodings. We represent the input sequence using one-hot encodings. A sequence of length l is mapped into $\mathbb{R}^{|\mathcal{S}|}$ by the embedding function $E : \mathcal{S} \to \mathbb{R}^{|\mathcal{S}|}$, defined as $E(i) = \mathsf{e}_i^{|\mathcal{S}|}$ for $i \in [k]$ and $E(r_i) = \mathsf{e}_{k+i}^{|\mathcal{S}|}$. For convenience, we omit the superscript $|\mathcal{S}|$ in what follows. After the encoding layer, the input sequence $x_0, x_1, \dots, x_{l-1}$ is given by

$$\mathsf{X} = \begin{bmatrix} e_{x_0} & e_{x_1} & \dots & e_{x_{l-1}} \end{bmatrix}^\top \in \mathbb{R}^{l \times |\mathcal{S}|}.$$

First attention layer. The first attention layer has k heads and considers only positional information, which is a convenient choice for this task. We use relative positional encodings with a causal mask. Each head i is parameterized by a vector $\mathbf{w}^i \in \mathbb{R}^l$. The pre-softmax attention scores of head i form a lower-triangular matrix with entries given by

$$\mathbf{A}^i[g, h] = \begin{cases} \mathbf{w}^i_{g-h} & \text{if } 0 \leqslant h \leqslant g \leqslant l-1 \\ -\infty & \text{otherwise} \end{cases}. \tag{1}$$

The output of attention head i $(1 \leqslant i \leqslant k)$ is $\mathbf{R}^i = \boldsymbol{\sigma}(\mathbf{A}^i)\,\mathsf{X} \in \mathbb{R}^{l \times |\mathcal{S}|}$, where $\boldsymbol{\sigma}$ denotes the row-wise softmax operation with causal masking. The output of the first attention layer is the concatenation of the outputs from all heads together with a skip connection $\mathbf{R}^0 = X$ (which differs from the standard architecture): $\mathbf{R} = [\mathbf{R}^0 \quad \mathbf{R}^1 \quad \dots \quad \mathbf{R}^k] \in \mathbb{R}^{l\times(k+1)\mathcal{S}} = \sum_{i=0}^{k} \left(\mathsf{e}_i^{k+1}\right)^\top \otimes \mathbf{R}^i$.

Second attention layer. The second attention layer consists of a single head, parameterized by matrices $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{(k+1)|\mathcal{S}|\times(k+1)|\mathcal{S}|}$. The attention scores are $\boldsymbol{\sigma}\left(X\mathbf{Q}^\top\mathbf{K}X^\top\right)$, where the softmax is applied row-wise with causal masking. The output of the second layer is $\mathbf{R}_+ = \boldsymbol{\sigma}\left(\mathbf{R}\mathbf{Q}^\top\mathbf{K}\mathbf{R}^\top\right)\mathbf{R}V$ We further simplify it by introducing a parameter $\boldsymbol{\beta} \in \mathbb{R}^k$ and parameterize $\mathbf{Q}^\top\mathbf{K}$ as

$$\mathbf{Q}^\top\mathbf{K} = \operatorname{diag}_{-1}\left(\boldsymbol{\beta}^2\right) \otimes \widetilde{\mathrm{I}}_k, \quad \mathbf{R}_+ = \boldsymbol{\sigma}\left(\mathbf{R}\left[\operatorname{diag}_{-1}\left(\boldsymbol{\beta}^2\right) \otimes \widetilde{\mathrm{I}}_k\right]\mathbf{R}^\top\right)\mathbf{R}V$$

where $\operatorname{diag}_{-1}(u) \in \mathbb{R}^{(k+1)\times(k+1)}$ denotes a matrix with u on its first sub-diagonal and zeros elsewhere. Squaring of $\boldsymbol{\beta}$ serves two purposes, (i) it ensures positivity of the entries and (ii) it preserves the 2-homogeneity of $\mathbf{Q}^\top\mathbf{K}$. We choose the value matrix $V = \mathsf{e}_0^{k+1} \otimes \mathrm{I}_{|\mathcal{S}|} \in \mathbb{R}^{(k+1)|\mathcal{S}|\times|\mathcal{S}|}$. This matrix consists of a column of blocks, with the identity matrix as the first block and zeros elsewhere. By construction, V extracts the skip connection from the concatenated output of the first layer, i.e., $\mathbf{R}V = \mathbf{R}^0 = X$, using the mixed-product property of the Kronecker product.

The model output. As the loss is computed only on the *last token*, the model's output depends only on the embedding of the final token after the second layer, i.e.,

$$\mathbf{p} = (\mathbf{R}_+)_{l-1} = \left(\boldsymbol{\sigma}\left(\mathbf{R}\mathbf{Q}^\top\mathbf{K}\mathbf{R}^\top\right)X\right)_{l-1} = \mathbf{R}^\top\left(\boldsymbol{\sigma}\left(\mathbf{R}\mathbf{Q}^\top\mathbf{K}\mathbf{R}^\top\right)_{l-1}\right) = \mathbf{R}^\top\boldsymbol{\sigma}\left(\left(\mathbf{R}\mathbf{Q}^\top\mathbf{K}\mathbf{R}^\top\right)_{l-1}\right).$$

We denote the attention scores by $\mathbf{s} = \boldsymbol{\sigma}\left(\mathbf{R}\mathbf{Q}^\top\mathbf{K}\mathbf{R}^\top\right)_{l-1}$ and the corresponding pre-softmax scores by $\widetilde{\mathbf{s}} = \left(\mathbf{R}\mathbf{Q}^\top\mathbf{K}\mathbf{R}^\top\right)_{l-1}$. The choice of V together with the orthogonal embeddings ensures that $\mathbf{p}$ is a valid probability distribution over the vocabulary, i.e., $\mathbf{p} \in \Delta^{|\mathcal{S}|}$, and requires no further normalization. Hence, the output of the model is given by $\mathbf{p} = X^\top\mathbf{s} = X^\top\boldsymbol{\sigma}(\widetilde{\mathbf{s}})$. Finally, we denote the parameters of the simplified model by $\boldsymbol{\theta} = (\mathbf{w}^1, \mathbf{w}^2, \dots, \mathbf{w}^k, \boldsymbol{\beta})$ here $\mathbf{w}^i \in \mathbb{R}^k$ for all i and $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \dots, \boldsymbol{\beta}_k) \in \mathbb{R}^k$. We use $\mathbf{p}(\boldsymbol{\theta})$ to denote the output of the model for parameters $\boldsymbol{\theta}$. We refer to section A.2 for a discussion on the simplifications and their implications.

2.3 The final problem setup

Training Objective. We replace the cross-entropy (CE) loss with the dot-product (DP) loss

$$\ell(\mathbf{p}, \mathbf{p}_*) = 1 - \langle\mathbf{p}, \mathbf{p}_*\rangle \qquad \ell_{\mathrm{CE}}(\mathbf{p}, \mathbf{p}_*) = -\langle\mathbf{p}_*, \log\mathbf{p}\rangle.$$

See App. A.3 for a detailed comparison of the two loss functions. Finally, the population DP loss is

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{f_\tau\sim\mathcal{D}(\mathcal{F}), q\sim\mathcal{P}_k}\,\ell(\mathbf{p}(\boldsymbol{\theta}), e_{f_\tau(q)}) = 1 - \mathbb{E}_{f_\tau, q}\left\langle\mathbf{p}(\boldsymbol{\theta}), e_{f_\tau(q)}\right\rangle. \tag{2}$$

Gradient Flow. To analyse the training dynamics, we consider the continuous-time limit of gradient descent, known as gradient flow. The parameters evolve according to the negative gradient of the population loss $\mathcal{L}$ with respect to the parameters. This approach does not account for the stochasticity or adaptive features of the optimizers used in practice. Nevertheless, it captures key aspects of training. The gradient flow is given by

$$\dot{\boldsymbol{\beta}}_h = -\frac{\partial\mathcal{L}}{\partial\boldsymbol{\beta}_h}, \quad \text{and} \quad \dot{\mathbf{w}}_i^h = -\frac{\partial\mathcal{L}}{\partial\mathbf{w}_i^h}, \quad \text{for all } i, h \in [k].$$

3 Technical Results

A stagewise learning process. We train the simplified transformer model with SGD and momentum on the DP loss over the in-context learning task of order $k+1$. A crucial observation is that the training dynamics are stage-wise. The model plateaus for an extended period before abruptly transitioning to another plateau with lower loss, and eventually converges to zero loss. This behavior becomes more pronounced as the initialization scale decreases, a phenomenon reminiscent of the saddle-to-saddle dynamics observed in deep networks under small-scale initialization [20, 31].

To mechanistically interpret these intermediate stages, we analyze what the model represents on each plateau, see Figure 1. At the first plateau, the model learns to match a single token in the context:

one attention head h_1 and its associated coefficient $\boldsymbol{\beta}_{h_1}$ are activated. At the second plateau, an additional head h_2 with coefficient $\boldsymbol{\beta}_{h_2}$ is activated, enabling the model to match two tokens of the query, q_{k+1-h_1}, q_{k+1-h_2} in the context. This process continues incrementally: at each stage, a new head–coefficient pair is activated, allowing the model to match one additional token. After k such stages, all tokens in the query are matched and the model achieves zero loss. We now turn to a detailed study of the stage-wise dynamics. We begin by analyzing a stylized initialization that isolates the transitions between plateaus. We then combine these analyses to obtain a complete picture.

The first jump. We study the dynamics of the first jump, when the model escapes from the initial plateau. We consider a stylized initialization, denoted $\mathcal{I}_1$, where $\boldsymbol{\beta}_1 = \epsilon$ and $\boldsymbol{\beta}_j = 0$ for all $j \neq 1$. The heads are initialized as $\mathbf{w}^i = 0$ for all i. Note that at initialization, all heads are symmetric.

Theorem 3.1. *Consider the simplified transformer model $\boldsymbol{\theta} = (\mathbf{w}^1, \mathbf{w}^2, \dots \mathbf{w}^k, \boldsymbol{\beta})$ with k heads and initialization $\mathcal{I}_1$, evolving under gradient flow on the DP loss. Then:*

(a) ***Directional bias:*** *For all time $t \geqslant 0$, $\mathbf{w}^1(t) = \alpha_1(t)\mathrm{e}_1^k + \delta_1(t)\mathbf{1}$ for some $\alpha(t), \delta(t) \in \mathbb{R}$.*

(b) ***Sparse attention:*** *$\dot{\mathbf{w}}_1^1 > 0$ and $\dot{\mathbf{w}}_i^1 < 0, \forall i \neq 1$, i.e., the head attends to the 1^{st} token from end.*

(c) ***A Sufficient ODE:*** *The learning dynamics can be fully described by the evolution of $\alpha_1(t), \boldsymbol{\beta}_1(t)$*

$$\dot{\alpha}_1 = \frac{\boldsymbol{\beta}_1^2 e^{\alpha_1}}{(e^{\alpha_1} + k - 1)^2} \frac{k^2}{k-1} \Xi, \qquad \dot{\boldsymbol{\beta}}_1 = 2\boldsymbol{\beta}_1 \frac{e^{\alpha_1} - 1}{e^{\alpha_1} + k - 1} \Xi$$

where Ξ is defined in equation (4) in Appendix.

(d) ***Conservation law:*** *The quantity $f(\alpha_1) - \boldsymbol{\beta}_1^2/4$ is conserved along the trajectory, i.e., the time derivative $\mathrm{d}(f(\alpha_1) - \boldsymbol{\beta}_1^2/4) = 0$ where*

$$f(\alpha_1) = 2\frac{k-1}{k^2}\left(\sinh(\alpha_1) - \alpha_1\right) - \frac{k-1}{k}\left[e^{-\alpha_1} + \alpha_1 - 1\right].$$

Some comments are in order. The parameters of the heads except head 1 are stationary. All relative position encodings except $\mathbf{w}_1^1$ evolves together. This follows from the inherent symmetry of the task: since token positions an be permuted within a sequence without leaving the distribution, the dynamics must preserve this symmetry. Combined with the symmetry of the initialization, this leads to the directional bias described above. The transformer rapidly learns to attend to the first token and results in sprase attention. There is a clear dichotomy: the embedding corresponding to this token grows, while the others decay at proportional rates.

From the ODE description, on any compact set, the time derivatives are bounded away from zero, ensuring that both α_1 and $\boldsymbol{\beta}_1$ diverge to infinity, where the gradient is zero. The conservation law show a coupled evolution of the parameters, note that sub conservation laws are common in dynamical systems, and recent works have identified conservation principles in transformers as well [27]. However, prior results are typically restricted to a single attention layer. In contrast, our result shows how parameters across multiple layers, separated by the softmax, jointly obey a conservation law.

This conservation law allows us to derive the timescale of the jump, i.e., how long training remains in the plateau. Suppose $\boldsymbol{\beta}_1(0) = \epsilon$ with $\epsilon \approx 0$. For convenience define $s = e^{\alpha_1} - 1$ with $s(0) = 0$, for small s, a Taylor expansion gives $\boldsymbol{\beta}_1^2 \approx 4s^2 + \epsilon^2$, and the local dynamics reduce to $\mathrm{d}s \sim c(s^2 + \epsilon^2)$. Thus the growth of parameters in self-attention has an information exponent of 2 [5]. Solving the ODE yields $s \approx \epsilon \tan(\epsilon T)$, implying that s requires $O(1/\epsilon)$ time to reach $O(1)$. When s has sufficiently grown, the dynamics switch regimes. Now $\boldsymbol{\beta}_h \sim \sqrt{s}$, Ξ which decays in s kicks in and the ODE simplifies to

$$\mathrm{d}s \sim bse^{-s} \implies \int \frac{e^s}{s}\mathrm{d}s = bT \tag{3}$$

The integral function on the right-hand side is the exponential integral, implying that s grows at rate $\log T$ and hence α_1 grows only at rate $\log\log T$.

Full stagewise dynamics. We refer to Appendix A.6 for the analysis of subsequent jumps, which is very similar to the first jump but of reduced order. Combining these analyses, we paint a complete picture of the stagewise dynamics.

References

[1] E. Abbe, E. B. Adsera, and T. Misiakiewicz. Sgd learning on neural networks: leap complexity and saddle-to-saddle dynamics. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 2552–2623. PMLR, 2023.

[2] K. Ahn, X. Cheng, H. Daneshmand, and S. Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[3] E. Akyürek, B. Wang, Y. Kim, and J. Andreas. In-context language learning: Arhitectures and algorithms. *arXiv preprint arXiv:2401.12973*, 2024.

[4] S. Arora, N. Cohen, W. Hu, and Y. Luo. Implicit regularization in deep matrix factorization. *Advances in Neural Information Processing Systems*, 32, 2019.

[5] G. B. Arous, R. Gheissari, and A. Jagannath. Online stochastic gradient descent on non-convex losses from high-dimensional inference. *Journal of Machine Learning Research*, 22(106):1–51, 2021. URL `http://jmlr.org/papers/v22/20-1288.html`.

[6] R. Berthier. Incremental learning in diagonal linear networks. *arXiv preprint arXiv:2208.14673*, 2022.

[7] S. Bhattamishra, A. Patel, P. Blunsom, and V. Kanade. Understanding in-context learning in transformers and llms by learning to learn discrete functions. *arXiv preprint arXiv:2310.03016*, 2023.

[8] A. Bietti, V. Cabannes, D. Bouchacourt, H. Jegou, and L. Bottou. Birth of a transformer: A memory viewpoint. *Advances in Neural Information Processing Systems*, 36, 2024.

[9] E. Boix-Adsera, E. Littwin, E. Abbe, S. Bengio, and J. Susskind. Transformers learn through gradual rank increase. *arXiv preprint arXiv:2306.07042*, 2023.

[10] E. Boursier, L. Pillaud-Vivien, and N. Flammarion. Gradient flow dynamics of shallow reLU networks for square loss and orthogonal inputs. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=L74c-iUxQ1I`.

[11] T. B. Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[12] A. Chen, R. Shwartz-Ziv, K. Cho, M. L. Leavitt, and N. Saphra. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in MLMs. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=MO5PiKHELW`.

[13] S. Chen, H. Sheen, T. Wang, and Z. Yang. Unveiling induction heads: Provable training dynamics and feature learning in transformers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[14] B. L. Edelman, E. Edelman, S. Goel, E. Malach, and N. Tsilivis. The evolution of statistical induction heads: In-context learning markov chains. *arXiv preprint arXiv:2402.11004*, 2024.

[15] D. Friedman, A. Wettig, and D. Chen. Learning transformer programs. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[16] K. Fukumizu and S. Amari. Local minima and plateaus in hierarchical structures of multilayer perceptrons. *Neural Networks*, 13(3):317–327, 2000. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(00)00009-5. URL `https://www.sciencedirect.com/science/article/pii/S0893608000000095`.

[17] 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.

[18] G. Gidel, F. Bach, and S. Lacoste-Julien. Implicit regularization of discrete gradient dynamics in linear neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[19] D. Gissin, S. Shalev-Shwartz, and A. Daniely. The implicit bias of depth: How incremental learning drives generalization. In *International Conference on Learning Representations*, 2020.

[20] A. Jacot, F. Ged, B. Şimşek, C. Hongler, and F. Gabriel. Saddle-to-saddle dynamics in deep linear networks: Small initialization training, symmetry, and sparsity. *arXiv preprint arXiv:2106.15933*, 2021.

[21] L. Jiang, Y. Chen, and L. Ding. Algorithmic regularization in model-free overparametrized asymmetric matrix factorization. *arXiv preprint arXiv:2203.02839*, 2022.

[22] J. Jin, Z. Li, K. Lyu, S. S. Du, and J. D. Lee. Understanding incremental learning of gradient descent: A fine-grained analysis of matrix sensing. *arXiv preprint arXiv:2301.11500*, 2023.

[23] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, Upper Saddle River, NJ, 2nd edition, 2009.

[24] J. Kim, S. Kwon, J. Y. Choi, J. Park, J. Cho, J. D. Lee, and E. K. Ryu. Task diversity shortens the icl plateau, 2024. URL `https://arxiv.org/abs/2410.05448`.

[25] Z. Li, Y. Luo, and K. Lyu. Towards resolving the implicit bias of gradient descent for matrix factorization: Greedy low-rank learning. In *International Conference on Learning Representations*, 2021.

[26] A. V. Makkuva, M. Bondaschi, A. Girish, A. Nagle, M. Jaggi, H. Kim, and M. Gastpar. Attention with markov: A framework for principled analysis of transformers via markov chains. *arXiv preprint arXiv:2402.04161*, 2024.

[27] S. Marcotte, R. Gribonval, and G. Peyré. Transformative or conservative? conservation laws for resnets and transformers. *arXiv preprint arXiv:2506.06194*, 2025.

[28] E. Nichani, A. Damian, and J. D. Lee. How transformers learn causal structure with gradient descent, 2024. URL `https://arxiv.org/abs/2402.14735`.

[29] E. Nichani, J. D. Lee, and A. Bietti. Understanding factual recall in transformers via associative memories. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=hwSmPOAmhk`.

[30] C. Olsson, N. Elhage, N. Nanda, N. Joseph, N. DasSarma, T. Henighan, B. Mann, A. Askell, Y. Bai, A. Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

[31] S. Pesme and N. Flammarion. Saddle-to-saddle dynamics in diagonal linear networks. *Advances in Neural Information Processing Systems*, 36:7475–7505, 2023.

[32] N. Rajaraman, M. Bondaschi, K. Ramchandran, M. Gastpar, and A. V. Makkuva. Transformers on markov data: Constant depth suffices. *arXiv preprint arXiv:2407.17686*, 2024.

[33] N. Razin, A. Maman, and N. Cohen. Implicit regularization in tensor factorization. In *International Conference on Machine Learning*, pages 8913–8924. PMLR, 2021.

[34] A. M. Saxe, J. L. McClelland, and S. Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23):11537–11546, 2019.

[35] C. E. Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[36] A. Svete and R. Cotterell. Transformers can represent n-gram language models. *arXiv preprint arXiv:2404.14994*, 2024.

[37] A. Varre, G. Yüce, and N. Flammarion. Learning in-context n-grams with transformers: Sub-n-grams are near-stationary points. In *International Conference on Machine Learning*, 2025.

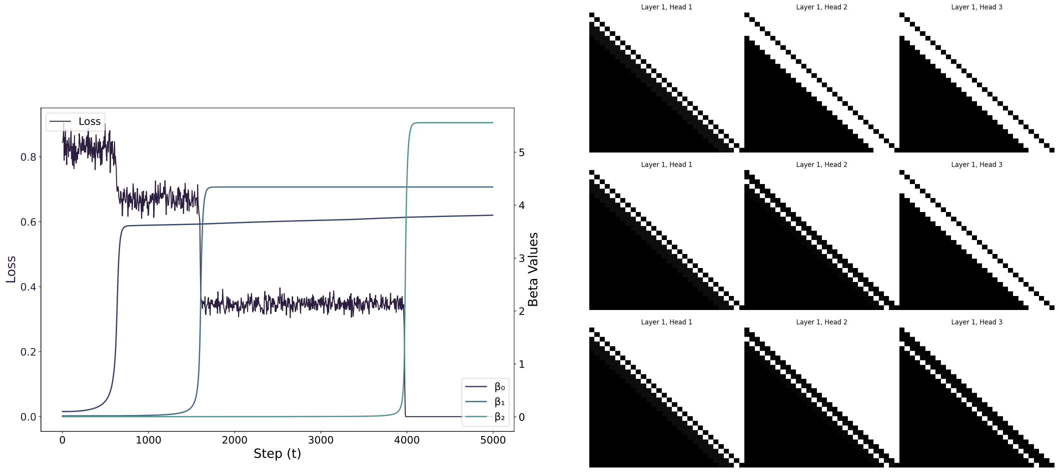
[38] A. V. Varre, M.-L. Vladarean, L. Pillaud-Vivien, and N. Flammarion. On the spectral bias of two-layer linear networks. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=FFdrXkm3Cz`.

[39] A. V. Varre, M. Sagitova, and N. Flammarion. Sgd vs gd: Rank deficiency in linear networks. *Advances in Neural Information Processing Systems*, 37:60133–60161, 2024.

[40] 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*, pages 35151–35174. PMLR, 2023.

[41] Y. Zhang, A. K. Singh, P. E. Latham, and A. Saxe. Training dynamics of in-context learning in linear attention, 2025. URL `https://arxiv.org/abs/2501.16265`.

A Appendix



(a)

(b)

Figure 1: The left panel is shown in a and the right panel in b. Note that the β values rise simultaneously with the loss, and the attention patterns at times 800, 1750, and 4500 demonstrate incremental learning.

Notation. For any positive integers a,b,s, $[s]$ denotes the set $\{0, 1, \ldots, s-1\}$, and $[a, b]$ represents $\{a, \ldots, b\}$. For a vector v, its i-th coordinate is v_i and e_i^s is the i-th standard basis vector in $\mathbb{R}^s$. For a matrix $A \in \mathbb{R}^{m \times n}$, its entry at row i and column j is A_{ij}, and its r-th row is $A_r \in \mathbb{R}^n$. For any set $\mathcal{S}$, $|\mathcal{S}|$ denotes the cardinality of the vocabulary. Δ^N denotes the probability simplex in $\mathbb{R}^N$. The Kronecker product is denoted by $\otimes$. We use $\mathbf{1}$ to denote all vector of all ones.

Definition A.1 (Jacobian of a function)**.** Let $f : \mathbb{R}^m \to \mathbb{R}^n$ be a C_1-function defined on a variable $X \in \mathbb{R}^m$. $\frac{\partial f}{\partial X}$ denotes the Jacobian which is a function from $\mathbb{R}^m \to \mathbb{R}^{n \times m}$.

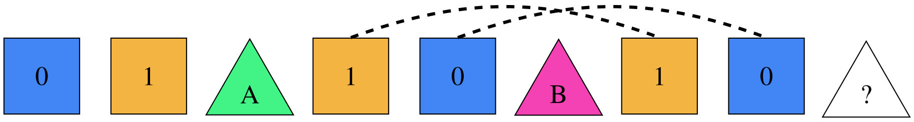


Figure 2: Illustration of the in-context associative recall task. The sequence shows mappings between query permutation elements (rectangles labeled 0, 1) and response tokens (triangles labeled A, B). The model must predict the missing association, marked with "?".

A.1 Related Work

In-context learning. The phenomenon of in-context learning (ICL) [11] has been investigated from several perspectives. Mechanistic interpretability has identified induction heads as key circuits supporting ICL [30]. A complementary direction examines restricted hypothesis classes, providing controlled settings to analyze how transformers develop in-context capabilities. A recurring observation across these studies is the emergence of training plateaus followed by sudden capability gains [12, 24]. These dynamics have been observed in regression tasks [17, 40, 2], boolean and formal language recognition [7, 3], and n-gram prediction.

n-gram models. n-grams models are related our work as the transformer circuit that solves our task also solves the taks of in-context learning n-grams. n-gram language models [35, 23] provide a natural testbed for analyzing transformer behavior. Several recent works adopt this viewpoint: the optimization landscape has been analyzed in [26], expressivity over n-gram distributions in [36], and in-context generalization in [32]. Other studies connect ICL to the emergence of induction heads [8] and their acquisition through gradient descent [28]. Edelman et al. [14] identify stage-wise dynamics in transformer training on in-context n-gram prediction, where intermediate solutions resemble sub-n-grams, while Varre et al. [37] formalize these sub-n-grams as near-stationary points. Finally, Chen et al. [13] investigate the same task with a modified architecture and initialization scheme that enforces head specialization from the start, thereby eliminating the stage-wise dynamics central to our analysis.

Incremental learning. Plateau-shaped learning curves arise broadly in neural network training, beyond ICL. Early work by Fukumizu and Amari [16] linked such phenomena to critical points in supervised learning. Related characterizations appear in simplified models such as matrix and tensor factorization [33, 21], matrix sensing [4, 25, 22], diagonal and linear networks [19, 34, 18, 20, 6, 31, 38, 39], ReLU networks [10, 1], and simplified transformer architectures [9]. Nichani et al. [29] studied the stage wise dynamics in Factual recall with linear attention. These results provide theoretical tools that we build on to characterize plateaus in in-context learning.

A.2 A simplified model

The goal of this paper is to study the training dynamics of transformers on the in-context associative recall task. The simplified architecture described above, although easier than a full transformer, remains too complex for a complete study of training dynamics. To address this intractability, we introduce additional simplifications that preserve the qualitative behavior of the full model while making analysis feasible. Before detailing these simplifications, we first present the construction of the solution implemented by the transformer for this task, which clarifies the rationale behind our design choices.

The transformer's solution. To solve the task, the transformer must learn to attend to the portion of the context that matches the final k tokens. This mechanism is implemented through a multi-head construction, variants of which have appeared in prior work [14, 32]. Its parameters are defined as

$$\mathbf{w}^i = c \cdot \mathsf{e}_i^l \text{ where } i \in [1, k], \quad \mathbf{Q}^\top \mathbf{K} = c \left(B \otimes \widetilde{\mathrm{I}}_k \right).$$

Here c is a positive constant, $\widetilde{\mathrm{I}}_k \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ where the first $k \times k$ block is given by $\widetilde{\mathrm{I}}_{:k,:k} = \mathrm{I}_k - \gamma \mathbf{1}_k \mathbf{1}_k^\top$ and 0's elsewhere. The matrix $B \in \mathbb{R}^{(k+1)\times(k+1)}$ is defined as

$$\text{For } 0 \leqslant g, h \leqslant k \;\; B_{gh} = \begin{cases} 1 & \text{if } g + 1 = h \\ 0 & \text{otherwise} \end{cases}.$$

As $c \to \infty$, the relative positional encoding ensures that head h outputs the embedding of x_{i-h} for token i. With this choice of B, the presoftmax attention scores are $\widetilde{\mathbf{s}}_i \approx c \sum_{h=1}^{k} \mathbb{1}\{x_{l-h} = x_{i-h}\}$ which is maximized when the histories of i and l match. In the limit $c \to \infty$, the softmax approaches hardmax attention, i.e, $\mathbf{s}_i \to 1$ (where i is the token that matches the history). The model output is then

$$\mathbf{p} = \sum_j \mathbf{s}_j e_{x_j} \approx e_{x_i}.$$

This construction is illustrated in the Figure 3.

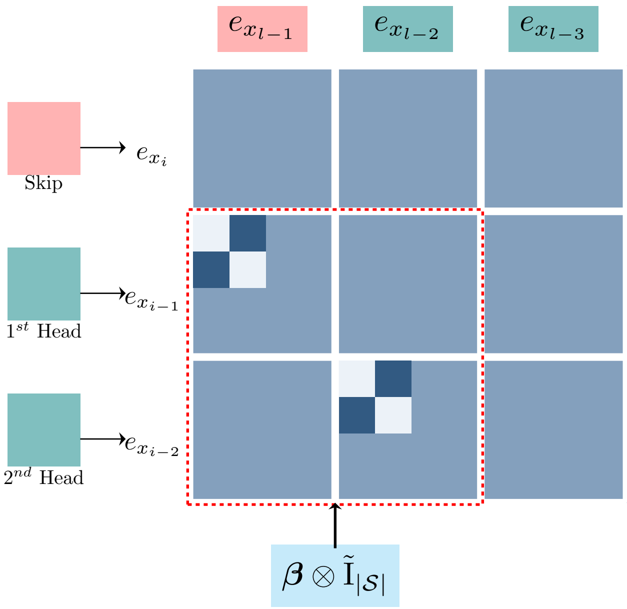


Figure 3: Layer-2 representation structure of the optimal solution constructed by the transfomer We use it to simplify this structure by a diagonal block matrix with trainable scales.

Simplifying the model. Our simplifications are motivated by the transformer's solution described above and preserve its overall structure, especially in the second layer. We emphasize that, despite these simplifications, the analysis remains intricate, as shown in the following sections and the training dynamics of the simplified model closely mirror those of the original model. For $i \in [k, l-1], t \in [l], h \in [1, k]$, the modification of the parametric model are given by:

Original model

$$\mathbf{R}_i^h = \sum_{j=0}^{i} \boldsymbol{\sigma}(\mathbf{A}^h)_{ij}\, e_{x_j},$$

$$\widetilde{\mathbf{s}}_t = \sum_{g,h=0}^{k} \left[\mathbf{R}_{l-1}^h\right]^\top \left[\mathbf{Q}^\top \mathbf{K}\right]_{hg} \left[\mathbf{R}_t^g\right],$$

$$\mathbf{s} = \boldsymbol{\sigma}(\widetilde{\mathbf{s}}), \quad \mathbf{p} = \sum_{t=0}^{l-1} \mathbf{s}_t\, e_{x_t}.$$

Simplified model

$$\mathbf{R}_i^h = \sum_{j=i-k}^{i-1} \boldsymbol{\sigma}(\mathbf{A}^h)_{ij}\, e_{x_j},$$

$$\widetilde{\mathbf{s}}_t = \sum_{h=1}^{k} \left[e_{x_{l-h}}\right]^\top \boldsymbol{\beta}_h^2 \left[\widetilde{\mathrm{I}}_k\right] \left[\mathbf{R}_t^h\right],,$$

$$\mathbf{s} = \boldsymbol{\sigma}\left(\widetilde{\mathbf{s}}_{\mathcal{R}}\right), \quad \mathbf{p} = \sum_{t: x_t \in \mathcal{R}} \mathbf{s}_t\, e_{x_t}.$$

In words, we make the following modifications:

(A0) We fix the attention window in the first layer to k for all heads. For each head h we train only the weights $\mathbf{w}_j^h$ for $1 \leqslant j \leqslant k$ while the remaining entries are masked out and set to $-\infty$.

(A1) We configure the second-layer attention parameters to match the structure of the optimal solution shown in Figure 3. This configuration is held fixed, and we train only the scalar multipliers that scale these parameters. In particular, we introduce a parameter $\boldsymbol{\beta} \in \mathbb{R}^k$ and parameterize $\mathbf{Q}^\top \mathbf{K}$ as

$$\mathbf{Q}^\top \mathbf{K} = \mathrm{diag}_{-1}\left(\boldsymbol{\beta}^2\right) \otimes \widetilde{\mathrm{I}}_k,$$

where $\mathrm{diag}_{-1}(u) \in \mathbb{R}^{(k+1)\times(k+1)}$ denotes a matrix with u on its first sub-diagonal and zeros elsewhere. Squaring of $\boldsymbol{\beta}$ serves two purposes, (i) it ensures positivity of the entries and (ii) it preserves the 2-homogeneity of $\mathbf{Q}^\top \mathbf{K}$.

(A2) As a further simplification, we replace the embeddings of the last with its embeddings at the solution. This assumption is mild and does not affect the training dynamics. It is mainly a convenience, as it avoids the bilinearity of the first layer outputs and leads to simpler gradient computations.

(A3) Since the output always lies in $\mathcal{R}$, we trim $\widetilde{\mathbf{s}}$ and apply the softmax only to the coordinates corresponding to responses. This ensures the output is always a probability vector over $\mathcal{R}$.

Other than (A1) which simplifies the second attention layer from matrix parameters to vector parameters, the other simplifications are mild and do not affect the essence of the analysis. We justify these choices both analytically and empirically in the next section.

Finally, we denote the parameters of the simplified model by $\boldsymbol{\theta} = (\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^k, \boldsymbol{\beta})$ here $\mathbf{w}^i \in \mathbb{R}^k$ for all i and $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_k) \in \mathbb{R}^k$. We use $\mathbf{p}(\boldsymbol{\theta})$ to denote the output of the model for parameters $\boldsymbol{\theta}$.

A.3 Cross Entropy and Dot-Product losses.

The minimum of the DP loss is

$$\underset{\mathbf{p} \in \Delta^{|\mathcal{S}|}}{\arg\min}\, 1 - \langle \mathbf{p}, \mathbf{p}_* \rangle = e_i \text{ where } i = \underset{j}{\arg\max}\ (\mathbf{p}_*)_j,$$

while the minimum of the CE loss is $\mathbf{p}_*$. These minima coincide whenever $\mathbf{p}_*$ is a one-hot vector. Since in our task each input sequence has a unique correct response, the target distribution $\mathbf{p}_*$ is always one-hot and the two losses are therefore equivalent. Their gradients also align when p_* is one-hot, differing only by a scaling factor: $\nabla_{\mathbf{p}} \ell(\mathbf{p}, \mathbf{p}_*) = -\mathbf{p}_*$, $\nabla_{\mathbf{p}} \ell_{\mathrm{CE}}(\mathbf{p}, \mathbf{p}_*) = -\langle \mathbf{p}, \mathbf{p}_* \rangle^{-1} \mathbf{p}_*$. Thus, the training dynamics under DP and CE losses are qualitatively identical, making the DP loss a perfect proxy for CE loss in our analysis.

A.4 Stage and Order of Learning

Formal description of stages Formally, if $h_1, h_2, \ldots h_k$ denote the sequence of heads activated across training, then on an input sequence f_τ the model incrementally learns the functions:

$$f_\tau^{\emptyset} \longrightarrow f_\tau^{\{h_1\}} \longrightarrow f_\tau^{\{h_1, h_2\}} \longrightarrow \ldots \longrightarrow f_\tau^{[1,k]},$$

where $\emptyset$ is the empty set and $f_\tau^{\mathcal{N}}$ for $\mathcal{N} \subseteq [1, k]$ is a function from $\mathcal{P}_{k+1}$ to $\Delta^{|\mathcal{R}|}$ and gives the frequency of the set $\{f_\tau(\pi) : \forall\ i \in \mathcal{N},\ \pi_{k+1-i} = q_{k+1-i}\}$, i.e., count the frequency of output of the permutations that match q at positions in $\mathcal{N}$ from the right.

Order of learning. A key observation concerns the order in which heads are activated. At small initialization scales, this order is determined by the relative magnitudes of the $\boldsymbol{\beta}$ coefficients at initialization. For example, if $\boldsymbol{\beta}_{(h_1)} > \boldsymbol{\beta}_{(h_2)} > \ldots > \boldsymbol{\beta}_{(h_k)}$, the heads are activated sequentially in the order $h_1, h_2, \ldots h_k$, see Fig. 4 in App.. Thus the implicit regularization induced by the scale and shape of initialization provides a natural curriculum, guiding the model to acquire the task in a stage-wise manner. For the remainder of the analysis, we assume without loss of generality that the coefficients are ordered $\boldsymbol{\beta}_1 > \boldsymbol{\beta}_2 > \ldots > \boldsymbol{\beta}_k$, so that heads are activated in order $1, 2, \ldots, k$. By re-indexing the heads, the analysis for arbitrary initial orderings reduces to this canonical case.

A.5 Supporting material for theoritical results

$$\Xi = \frac{2(1+\gamma)}{(e^{\gamma_1} + k - 1)^2} \left[\frac{e^{\gamma_1}(|\mathcal{R}| - 1)}{(k-2)!|\mathcal{R}|} \right] \text{ where } \gamma_1 = (1+\gamma)\boldsymbol{\beta}_1^2 \frac{e^{\alpha_1} - 1}{e^{\alpha_1} + k - 1} \tag{4}$$

A.6 Subsequent Jumps and their analysis

The subsequent jumps. Similar to the first jump, we can analyze the subsequent ones. We consider a stylized initialization, denoted $\mathcal{I}_h$, where $\boldsymbol{\beta}_i = c$ for all $i \in [1, h-1]$, $\boldsymbol{\beta}_h = \epsilon$, and $\boldsymbol{\beta}_j = 0$ for all $j > h$, with c taken to be very large. Likewise, we set $\mathbf{w}^i = c_1 \mathrm{e}_i^k + c_2 \mathbf{1}$ for $i \in [1, h-1]$, and $\mathbf{w}^i = 0$ otherwise. Under this initialization, we study the dynamics of the h^{th} jump as the model escapes the plateau where it has learned to match $h-1$ tokens in the context. A key detail is the interplay between macroscopic parameters (the large c) and microscopic parameters (the small ϵ). In this setting, the striking feature is that the macroscopic parameters remain stationary while the microscopic ones evolve.

Proposition A.2. *Consider the simplified transformer model $\boldsymbol{\theta} = (\mathbf{w}^1, \mathbf{w}^2, \dots \mathbf{w}^k, \boldsymbol{\beta})$ with k heads and initialization $\mathcal{I}_h$, evolving under gradient flow on the DP loss. Then:*

- *(a)* ***Stationarity of the macroscopic variables:*** *The gradients of parameters in the first $h-1$ heads vanish at scale $\nabla_{\mathbf{w}^i}\mathcal{L}, \nabla_{\boldsymbol{\beta}_i}\mathcal{L} = O(e^{-c})$ for $i \in [1, h-1]$.*
- *(b)* ***Dynamics of the microscopic variables:*** *At $c \to \infty$, the dynamics of the remaining parameters corresponds to the those of the first jump on a task of reduced order.*

Stitching the jumps. Without loss of generality, assume the initialization $\boldsymbol{\beta} = (c_1\epsilon, c_2\epsilon, c_3\epsilon, \dots)$, for $c_1 > c_2 > c_3 > \cdots$, with ϵ very small. Using the first-jump computations, we obtain an time T such that $\boldsymbol{\beta}_1(T) > C$ for some large constant C. During this time, the gradients of the other heads remain $O(\epsilon)$, so their parameters stay close to the origin. Next, applying the subsequent-jump analysis, we can compute a time T_2 such that $\boldsymbol{\beta}_2 > C$. Proceeding in this manner, we can stitch the jumps together to describe the full trajectory. This shows that, in principle, the entire evolution can be characterized by chaining together successive jumps, though we do not pursue the full analysis here, as it does not reveal qualitatively new phenomena beyond perturbation analysis.

Generalizations of the simple model. We discuss possible relaxations of the perturbed model. In particular, we highlight three illustrative generalizations. For simplification **(A0)**, the attention window of size k provides a convenient way to compute closed-form expressions. The conservation law and the time scale can also be derived without this assumption, though in that case we lose the directional bias and the ability to obtain closed-form formulas. For simplification (A2), replacing the embeddings of the last token does not pose difficulties for the analysis. The argument still holds for the ordering $\boldsymbol{\beta}_1 > \boldsymbol{\beta}_2 > \boldsymbol{\beta}_3 > \cdots$, since the skip connection supplies the embedding of the last token for the first jump, and the head learned at jump i provides the embedding of the last token for jump $i+1$. Simplification (A3) can also be avoided by choosing a value matrix V that directly outputs the response. However, the output is not guaranteed to be a probability vector. Normalizing by the sum restores this property, making it equivalent to considering the pre-softmax scores of the responses. These generalizations indicate that the phenomena we study are robust to modest relaxations of the simplified setup, even if the algebraic convenience of the original model is lost.

Experiments. We use task of order 4 and the response vocabulary of also size 4. Overall, the results confirm our theoretical predictions: the model exhibits stage-wise plateaus followed by sharp jumps, with attention heads activating sequentially to implement recall.

B Proofs of Main Results

B.1 Proof of Theorem 3.1

The gradient flow of the parameters is given by

$$\dot{\boldsymbol{\beta}}_h = -\mathbb{E}\frac{\partial \ell}{\partial \boldsymbol{\beta}_h},$$
$$\dot{\mathbf{w}}_i^h = -\mathbb{E}\frac{\partial \ell}{\partial \mathbf{w}_i^h}.$$

First to show a directional bias, we will show that the trajectory always move along the manifold $\mathcal{M}_1 = \{\boldsymbol{\theta} : \mathbf{w}^1 = \alpha_1 \mathsf{e}_h^1 + \delta_1 \mathbf{1},\ \alpha_1, \delta_1, \in \mathbb{R}$ and $\boldsymbol{\beta}_h, \mathbf{w}^h = 0$ for $h \neq 1\}$.

Consider any point on the manifold $\mathcal{M}_1$ say $\boldsymbol{\theta} = (\mathbf{w}^1, \mathbf{w}^2, \dots \mathbf{w}^k, \boldsymbol{\beta})$ where $\mathbf{w}^1 = \alpha_1 \mathsf{e}_h^1 + \delta_1 \mathbf{1}$ and $\boldsymbol{\beta}_h = 0$ for $h \neq 1$. We will show that the gradient has no component along the normal space of the manifold $\mathcal{M}_1$ at the point $\boldsymbol{\theta}$. Hence the trajectory will always move along the manifold $\mathcal{M}_1$.

As the parameters satisfy the form prescribed by Lemma C.2, we can invoke the Lemma C.1 to compute the gradients. The population gradients are given by

$$\mathbb{E}\frac{\partial \ell}{\partial \boldsymbol{\beta}_h} = 2\boldsymbol{\beta}_{_h}\frac{e^{\alpha_h}-1}{e^{\alpha_h}+k-1}(1+\gamma)\mathbb{E}\Delta_{h,h}, \tag{5}$$

$$\mathbb{E}\frac{\partial \ell}{\partial \mathbf{w}^h_i} = (1+\gamma)\frac{\boldsymbol{\beta}^2_{_h}\left(\left(e^{\alpha_h}-1\right)\mathbb{1}\{i=h\}+1\right)}{e^{\alpha_h}+k-1}\left(\mathbb{E}\Delta_{h,i} - \frac{e^{\alpha_h}-1}{e^{\alpha_h}+k-1}\mathbb{E}\Delta_{h,h}\right) \tag{6}$$

where

$$\Delta_{h,i} = -\left[\sum_{t=1}^{k!}\mathbf{s}_t\mathbb{1}\{r_t = r \wedge q_{k-h} = \pi_{(t,k-i)}\} - \left(\sum_{j=1}^{k!}\mathbf{s}_j\mathbb{1}\{r_j = r\}\right)\left(\sum_{t=1}^{k!}\mathbf{s}_t\left(\mathbb{1}\{q_{k-h} = \pi_{(t,k-i)}\}\right)\right)\right],$$

$$s = \boldsymbol{\sigma}(\mathbf{s}^{'}), \quad \mathbf{s}^{'}_t = \sum_{h=1}^{k}\gamma_h\mathbb{1}\{q_{k-h} = \pi_{(t,k-h)}\}$$

$$\gamma_h = (1+\gamma)\boldsymbol{\beta}^2_{_h}\frac{e^{\alpha_h}-1}{e^{\alpha_h}+k-1}$$

Note that the gradients of $\boldsymbol{\beta}_h$ for $h \neq 1$ are zero as $\boldsymbol{\beta}_{_h} = 0$ for $h \neq 1$. Also note that the gradients of $\mathbf{w}^h$ for $h \neq 1$ are zero as $\boldsymbol{\beta}^2_{_h} = 0$ for $h \neq 1$. The only thing that is left to show is that the gradients of $\mathbf{w}^1$ are of the form $\frac{\partial \ell}{\partial \mathbf{w}^1} = \xi_1 \mathsf{e}^1_1 + \zeta_1\mathbf{1}$ for some $\xi_1, \zeta_1 \in \mathbb{R}$.

Note that $\mathbf{s}^{'}_t = \gamma_1\mathbb{1}\{q_{k-1} = \pi_{(t,k-1)}\}$ as $\gamma_h = 0$ for $h \neq 1$ and the softmax score is given by

$$\mathbf{s}_t = \frac{e^{\gamma_1\mathbb{1}\{q_{k-1}=\pi_{(t,k-1)}\}}}{\sum_{j=1}^{k!}e^{\gamma_1\mathbb{1}\{q_{k-1}=\pi_{(j,k-1)}\}}} = \frac{\left(e^{\gamma_1}-1\right)\mathbb{1}\{q_{k-1} = \pi_{(t,k-1)}\}+1}{e^{\gamma_1}+k-1}\frac{1}{(k-1)!}.$$

From Lemma C.3 we have

$$\mathbb{E}\Delta_{1,i} = \frac{1}{(e^{\gamma_1}+k-1)^2}\left[\frac{e^{\gamma_1}(|\mathcal{R}|-1)}{(k-1)!|\mathcal{R}|}\right], \quad \text{for } i \neq 1.$$

$$\mathbb{E}\Delta_{1,1} = -(k-1)\mathbb{E}\Delta_{1,i} = -\frac{1}{(e^{\gamma_1}+k-1)^2}\left[\frac{e^{\gamma_1}(|\mathcal{R}|-1)}{(k-2)!|\mathcal{R}|}\right].$$

So the gradient of $\mathbf{w}^1_i$ given by $\frac{\partial \ell}{\partial \mathbf{w}^1_i}$ is independent of i for $i \neq 1$ since $\mathbb{E}\Delta_{1,i}$ is independent of i. Hence the gradient of $\mathbf{w}^1$ is of the form $\frac{\partial \ell}{\partial \mathbf{w}^1} = \xi_1\mathsf{e}^1_1 + \zeta_1\mathbf{1}$ for some $\xi_1, \zeta_1 \in \mathbb{R}$. This proves the directional bias of the trajectory.

For the next signal propagation part of the result, we will show that there exists note that $\mathbb{E}\Delta_{1,1} < 0$ for all $\gamma_1 > 0$ and $\mathbb{E}\Delta_{1,i} > 0$ for all $\gamma_1 > 0$ and $i \neq 1$. Using these expressions, the population gradient of $\mathbf{w}^1_1$ is given by

$$\begin{aligned}\mathbb{E}\frac{\partial \ell}{\partial \mathbf{w}^1_1} &= (1+\gamma)\frac{\boldsymbol{\beta}^2_{_1}e^{\alpha_1}}{e^{\alpha_1}+k-1}\left(\mathbb{E}\Delta_{1,1} - \frac{e^{\alpha_1}-1}{e^{\alpha_1}+k-1}\mathbb{E}\Delta_{1,1}\right)\\ &= (1+\gamma)\frac{\boldsymbol{\beta}^2_{_1}e^{\alpha_1}}{e^{\alpha_1}+k-1}\left(\frac{k-1}{e^{\alpha_1}+k-1}\mathbb{E}\Delta_{1,1}\right) < 0\end{aligned}$$

Similarly the population gradient of $\mathbf{w}^1_i$ for $i \neq 1$ is given by

$$\begin{aligned}\mathbb{E}\frac{\partial \ell}{\partial \mathbf{w}^1_i} &= (1+\gamma)\frac{\boldsymbol{\beta}^2_{_1}}{e^{\alpha_1}+k-1}\left(\mathbb{E}\Delta_{1,i} - \frac{e^{\alpha_1}-1}{e^{\alpha_1}+k-1}\mathbb{E}\Delta_{1,1}\right),\\ &= (1+\gamma)\frac{\boldsymbol{\beta}^2_{_1}}{e^{\alpha_1}+k-1}\left(-\frac{\mathbb{E}\Delta_{1,1}}{k-1} - \frac{e^{\alpha_1}-1}{e^{\alpha_1}+k-1}\mathbb{E}\Delta_{1,1}\right),\\ &> 0\end{aligned}$$

The dynamics of gradient descent is given by

$$\begin{aligned}
\dot{\mathbf{w}}_{1}^{1} &= -\mathbb{E}\frac{\partial \ell}{\partial \mathbf{w}_{1}^{1}} = -(1+\gamma)\frac{\boldsymbol{\beta}_{1}^{2}e^{\alpha_1}}{e^{\alpha_1}+k-1}\left(\frac{k}{e^{\alpha_1}+k-1}\mathbb{E}\Delta_{1,1}\right) > 0,\\
\dot{\mathbf{w}}_{i}^{1} &= -\mathbb{E}\frac{\partial \ell}{\partial \mathbf{w}_{i}^{1}} = (1+\gamma)\frac{\boldsymbol{\beta}_{1}^{2}}{e^{\alpha_1}+k-1}\left(-\frac{\mathbb{E}\Delta_{1,1}}{k-1} - \frac{e^{\alpha_1}-1}{e^{\alpha_1}+k-1}\mathbb{E}\Delta_{1,1}\right) < 0, \quad i \neq 1\\
\dot{\boldsymbol{\beta}}_{1} &= -\mathbb{E}\frac{\partial \ell}{\partial \boldsymbol{\beta}_1} = -2\boldsymbol{\beta}_{1}\frac{e^{\alpha_1}-1}{e^{\alpha_1}+k-1}(1+\gamma)\mathbb{E}\Delta_{1,1} > 0
\end{aligned}$$

The complete dynamics of the system is given by $\dot{\alpha}_1, \dot{\boldsymbol{\beta}}_{1}$ and they can be written as

$$\begin{aligned}
\dot{\alpha}_1 &= \dot{\mathbf{w}}_{1}^{1} - \dot{\mathbf{w}}_{2}^{1} = (1+\gamma)\frac{\boldsymbol{\beta}_{1}^{2}}{e^{\alpha_1}+k-1}\left(\frac{(k)e^{\alpha_1}}{e^{\alpha_1}+k-1}\mathbb{E}\Delta_{1,1} + \frac{\mathbb{E}\Delta_{1,1}}{k-1} + \frac{e^{\alpha_1}-1}{e^{\alpha_1}+k-1}\mathbb{E}\Delta_{1,1}\right) < 0,\\
\dot{\boldsymbol{\beta}}_{1} &= -2\boldsymbol{\beta}_{1}\frac{e^{\alpha_1}-1}{e^{\alpha_1}+k-1}(1+\gamma)\mathbb{E}\Delta_{1,1} > 0
\end{aligned}$$

The final system of ODE is given by '

$$\begin{aligned}
\dot{\alpha}_1 &= \frac{\boldsymbol{\beta}_{1}^{2}e^{\alpha_1}}{e^{\alpha_1}+k-1}^{2}\frac{k^2}{k-1} - \mathbb{E}(1+\gamma)\Delta_{1,1},\\
\dot{\boldsymbol{\beta}}_{1} &= 2\boldsymbol{\beta}_{1}\frac{e^{\alpha_1}-1}{e^{\alpha_1}+k-1} - \mathbb{E}(1+\gamma)\Delta_{1,1}
\end{aligned}$$

Lets denote $\Xi = -2(1+\gamma)\mathbb{E}\Delta_{1,1}$ which is positive and given by

$$\Xi = \frac{2(1+\gamma)}{(e^{\gamma_1}+k-1)^2}\left[\frac{e^{\gamma_1}(|\mathcal{R}|-1)}{(k-2)!|\mathcal{R}|}\right] \text{ where } \gamma_1 = (1+\gamma)\boldsymbol{\beta}_{1}^{2}\frac{e^{\alpha_1}-1}{e^{\alpha_1}+k-1}$$

Using this the system of ODE can be written as

$$\begin{aligned}
\dot{\alpha}_1 &= \frac{\boldsymbol{\beta}_{1}^{2}e^{\alpha_1}}{(e^{\alpha_1}+k-1)^2}\frac{k^2}{k-1}\Xi,\\
\dot{\boldsymbol{\beta}}_{1} &= 2\boldsymbol{\beta}_{1}\frac{e^{\alpha_1}-1}{e^{\alpha_1}+k-1}\Xi
\end{aligned}$$

Note that

$$\frac{\dot{\boldsymbol{\beta}}_{1}^{2}}{4} = \boldsymbol{\beta}_{1}^{2}\frac{e^{\alpha_1}-1}{e^{\alpha_1}+k-1}\Xi$$

$$f(\dot{\alpha}_1) = f^{'}(\alpha_1)\dot{\alpha}_1 = f^{'}(\alpha_1)\frac{\boldsymbol{\beta}_{1}^{2}e^{\alpha_1}}{(e^{\alpha_1}+k-1)^2}\frac{k^2}{k-1}\Xi$$

If we choose $f(\alpha_1)$ such that

$$\begin{aligned}
f^{'}(\alpha_1)\frac{\boldsymbol{\beta}_{1}^{2}e^{\alpha_1}}{(e^{\alpha_1}+k-1)^2}\frac{k^2}{k-1}\Xi = \boldsymbol{\beta}_{1}^{2}\frac{e^{\alpha_1}-1}{e^{\alpha_1}+k-1}\Xi,&\\
\implies f^{'}(\alpha_1) &= \frac{k-1}{k^2}\frac{(e^{\alpha_1}-1)(e^{\alpha_1}-1+k)}{e^{\alpha_1}}\\
&= \frac{k-1}{k^2}\left(e^{\alpha_1} - 2 + e^{-\alpha_1} - ke^{-\alpha_1} + k\right)\\
\implies f(\alpha_1) &= \frac{k-1}{k^2}\left(e^{\alpha_1} - 2\alpha_1 - e^{-\alpha_1} + ke^{-\alpha_1} + k\alpha_1\right),\\
f(\alpha_1) &= 2\frac{k-1}{k^2}\left(\sinh(\alpha_1) - \alpha_1\right) + \frac{k-1}{k}\left[e^{-\alpha_1} + \alpha_1 - 1\right]
\end{aligned}$$

Now $f(\alpha_1) - {\beta_1^2}/{4}$ is conserved along the trajectory. As $\mathrm{d}(f(\alpha_1) - {\beta_1^2}/{4}) = 0$.

This completes the proof of the theorem.

B.2 Proof of Theorem A.2

The gradient flow is given by The population gradients are given by

$$\mathbb{E}\frac{\partial \ell}{\partial \boldsymbol{\beta}_h} = 2\boldsymbol{\beta}_h \frac{e^{\alpha_h}-1}{e^{\alpha_h}+k-1}(1+\gamma)\mathbb{E}\Delta_{h,h}, \tag{7}$$

$$\mathbb{E}\frac{\partial \ell}{\partial \mathbf{w}_i^h} = (1+\gamma)\frac{\boldsymbol{\beta}_h^2\left((e^{\alpha_h}-1)\,\mathbb{1}\{i=h\}+1\right)}{e^{\alpha_h}+k-1}\left(\mathbb{E}\Delta_{h,i} - \frac{e^{\alpha_h}-1}{e^{\alpha_h}+k-1}\mathbb{E}\Delta_{h,h}\right) \tag{8}$$

where

$$\Delta_{h,i} = -\left[\sum_{t=1}^{k!}\mathbf{s}_t\mathbb{1}\{r_t = r \wedge q_{k-h} = \pi_{(t,k-i)}\} - \left(\sum_{j=1}^{k!}\mathbf{s}_j\mathbb{1}\{r_j = r\}\right)\left(\sum_{t=1}^{k!}\mathbf{s}_t\left(\mathbb{1}\{q_{k-h} = \pi_{(t,k-i)}\}\right)\right)\right],$$

$$s = \boldsymbol{\sigma}(\mathbf{s}^{'}), \quad \mathbf{s}_t^{'} = \sum_{h=1}^{k}\gamma_h\mathbb{1}\{q_{k-h} = \pi_{(t,k-h)}\}$$

$$\gamma_h = (1+\gamma)\boldsymbol{\beta}_h^2\frac{e^{\alpha_h}-1}{e^{\alpha_h}+k-1}$$

Without loss of generality, assume that $\gamma_i = c$ for $i \in [1, h-1]$. Now split the set of permutations into two partitions $\mathcal{P}_S$ and $\mathcal{P}_S^c$ where for query q, $\mathcal{P}_S$ is the set that matches the last $h-1$ tokens and others do not. Now $\mathbf{s}_t = e^{-c}$ for $t \in \mathcal{P}_S^c$. Now $\Delta_{i,j}$ for $i \in [1, h-1]$ and $j \neq i$, it can seen that $\mathbf{s}_t\left(\mathbb{1}\{q_{k-i} = \pi_{(t,k-j)}\}\right) = e^{-c}$ for any t, hence $\mathbb{E}\Delta_{i,j} = e^{-c}$. The cases left are $\mathbb{E}\Delta_{i,i}$, but it is also of e^{-c} as the $\Delta_{i,:}$'s sum to 0.

Now the case of $i = h$, now for $t \in \mathcal{P}_S$

$$s_t = \frac{\exp\{c\}\exp\{\gamma_h\mathbb{1}\{q_{k-h} = \pi_{(t,k-h)}\}\}}{\sum_{\pi_t\in\mathcal{P}_S}\exp\{c\}\exp\{\gamma_h\mathbb{1}\{q_{k-h} = \pi_{(t,k-h)}\}\} + O(1)},$$
$$= \frac{\exp\{\gamma_h\mathbb{1}\{q_{k-h} = \pi_{(t,k-h)}\}\}}{\sum_{\pi_t\in\mathcal{P}_S}\exp\{\gamma_h\mathbb{1}\{q_{k-h} = \pi_{(t,k-h)}\}\}} + O(e^{-c})$$

This expression is exactly equivalent to the first jump however now restricted on the set P_S of reduced order.

C Computations of the derivatives of the simplified transformer model

The forward pass of the simplified transformer model on a single input sequence is given by, let r be the response of the query.

$$\mathbf{R}_i^h = \sum_{j=i-k}^{i-1}\boldsymbol{\sigma}(\mathbf{A}^h)_{ij}\,e_{x_j}, \tag{9a}$$

$$\widetilde{\mathbf{s}}_t = \sum_{h=1}^{k}\boldsymbol{\beta}_h^2\left\langle e_{x_{l-h}}, \widetilde{\mathrm{I}}_k\,\mathbf{R}_t^h\right\rangle, \tag{9b}$$

$$\mathbf{s} = \boldsymbol{\sigma}\left(\widetilde{\mathbf{s}}_{\mathcal{R}}\right) \tag{9c}$$

$$\mathbf{p} = \sum_{t:x_t\in\mathcal{R}}\mathbf{s}_t\,e_{x_t} \tag{9d}$$

$$\ell = 1 - \langle\mathbf{p}, e_r\rangle \tag{9e}$$

For the convenience of the analysis, we will drop $\mathcal{R}$ in the subscript of $\widetilde{\mathbf{s}}$ and now $\widetilde{\mathbf{s}} \in \mathbb{R}^{k!}$ where $\widetilde{\mathbf{s}}_t$ denotes the score of the response corresponding to the t^{th} permutation in the sequence.

Derivatives wrt to the $\widetilde{\mathbf{s}}$. The derivatives of the loss with respect to the predicted scores can be computed using the chain rule:

$$\frac{\partial \ell}{\partial \widetilde{\mathbf{s}}} = \frac{\partial \ell}{\partial \mathbf{p}}\frac{\partial \mathbf{p}}{\partial \mathbf{s}}\frac{\partial \mathbf{s}}{\partial \widetilde{\mathbf{s}}} = -e_r^\top X_{\mathcal{R}}^\top\left(\mathrm{diag}(\mathbf{s}) - \mathbf{s}\,\mathbf{s}^\top\right).$$

The co-ordinate wise derivative is

$$\frac{\partial \ell}{\partial \widetilde{\mathbf{s}}_t} = -\mathbf{s}_t \left(\mathbb{1}\{r_t = r\} - \sum_{j=1}^{k!} \mathbf{s}_j \mathbb{1}\{r_j = r\} \right) = \ell_t^{'}.$$

where r_t is the response corresponding to the t^{th} permutation in the sequence. Using the property of the softmax we have $\ell(\widetilde{\mathbf{s}} + c\mathbf{1}) = \ell(\widetilde{\mathbf{s}})$ as the softmax is invariant when even co-ordinate is shifted by a constant. Now taking the derivative with c at $c = 0$ we get

$$\sum_{t=1}^{k!} \ell_t^{'} = 0 \tag{10}$$

Derivative of $\widetilde{\mathrm{s}}$ wrt to $\boldsymbol{\beta}$ and $\mathbf{w}^h$'s The derivative of $\widetilde{\mathrm{s}}$ with respect to $\boldsymbol{\beta}$ can be computed as follows:

$$\frac{\partial \widetilde{\mathbf{s}}_t}{\partial \boldsymbol{\beta}_h} = 2\boldsymbol{\beta}_h \left\langle e_{x_{l-h}}, \widetilde{\mathrm{I}}_k \, \mathbf{R}_t^h \right\rangle$$

The derivative of $\widetilde{\mathrm{s}}$ with respect to $\mathbf{w}_i^h$ can be computed as follows:

$$\frac{\partial \widetilde{\mathbf{s}}_t}{\partial \mathbf{w}_i^h} = \boldsymbol{\beta}_h^2 \left\langle e_{x_{l-h}}, \widetilde{\mathrm{I}}_k \, \frac{\partial \mathbf{R}_t^h}{\partial \mathbf{w}_i^h} \right\rangle$$

Using Lemma D.2 we have

$$\begin{aligned}
\frac{\partial \widetilde{\mathbf{s}}_t}{\partial \mathbf{w}_i^h} &= \boldsymbol{\beta}_h^2 \left\langle e_{x_{l-h}}, \widetilde{\mathrm{I}}_k \, \frac{e^{\mathbf{w}_i^h}}{\sum_{j=1}^k e^{\mathbf{w}_j^h}} \left(e_{\pi_{(t,k-i)}} - \mathbf{R}_t^h \right) \right\rangle \\
&= \boldsymbol{\beta}_h^2 \frac{e^{\mathbf{w}_i^h}}{\sum_{j=1}^k e^{\mathbf{w}_j^h}} \left(\left\langle e_{x_{l-h}}, \widetilde{\mathrm{I}}_k \, e_{\pi_{(t,k-i)}} \right\rangle - \left\langle e_{x_{l-h}}, \widetilde{\mathrm{I}}_k \, \mathbf{R}_t^h \right\rangle \right)
\end{aligned}$$

Note that $x_{l-h} = q_{k-h}$ and $\left\langle e_{x_{l-h}}, \widetilde{\mathrm{I}}_k \, e_{\pi_{(t,k-i)}} \right\rangle = \mathbb{1}\{q_{k-h} = \pi_{(t,k-i)}\} - \gamma \mathbb{1}\{q_{k-h} \neq \pi_{(t,k-i)}\} \mathbb{1}\{q_{k-h} = \pi_{(t,k-i)}\} - \gamma$. Using this computation, we have,

$$\frac{\partial \widetilde{\mathbf{s}}_t}{\partial \mathbf{w}_i^h} = \frac{\boldsymbol{\beta}_h^2 e^{\mathbf{w}_i^h}}{\sum_{j=1}^k e^{\mathbf{w}_j^h}} \left((1+\gamma)\mathbb{1}\{q_{k-h} = \pi_{(t,k-i)}\} - \gamma - \left\langle e_{x_{l-h}}, \widetilde{\mathrm{I}}_k \, \mathbf{R}_t^h \right\rangle \right)$$

Computing the derivative of the loss with respect to $\boldsymbol{\beta}$ and $\mathbf{w}_i^h$ we have

$$\begin{aligned}
\frac{\partial \ell}{\partial \boldsymbol{\beta}_h} &= \sum_{t=1}^{k!} \ell_t^{'} \, 2\boldsymbol{\beta}_h \left\langle e_{x_{l-h}}, \widetilde{\mathrm{I}}_k \, \mathbf{R}_t^h \right\rangle = 2\boldsymbol{\beta}_h \sum_{t=1}^{k!} \ell_t^{'} \left\langle e_{x_{l-h}}, \widetilde{\mathrm{I}}_k \, \mathbf{R}_t^h \right\rangle \\
\frac{\partial \ell}{\partial \mathbf{w}_i^h} &= \sum_{t=1}^{k!} \ell_t^{'} \, \frac{\boldsymbol{\beta}_h^2 e^{\mathbf{w}_i^h}}{\sum_{j=1}^k e^{\mathbf{w}_j^h}} \left((1+\gamma)\mathbb{1}\{q_{k-h} = \pi_{(t,k-i)}\} - \gamma - \left\langle e_{x_{l-h}}, \widetilde{\mathrm{I}}_k \, \mathbf{R}_t^h \right\rangle \right), \\
&= \frac{\boldsymbol{\beta}_h^2 e^{\mathbf{w}_i^h}}{\sum_{j=1}^k e^{\mathbf{w}_j^h}} \sum_{t=1}^{k!} \ell_t^{'} \left((1+\gamma)\mathbb{1}\{q_{k-h} = \pi_{(t,k-i)}\} - \left\langle e_{x_{l-h}}, \widetilde{\mathrm{I}}_k \, \mathbf{R}_t^h \right\rangle \right),
\end{aligned}$$

Now collecting the final expressions we have the following lemma.

Lemma C.1. *The derivatives of the loss ℓ with respect to the parameters $\boldsymbol{\beta}$ and $\mathbf{w}^h$'s are given by*

$$\begin{aligned}
\frac{\partial \ell}{\partial \boldsymbol{\beta}_h} &= 2\boldsymbol{\beta}_h \sum_{t=1}^{k!} \ell_t^{'} \left\langle e_{x_{l-h}}, \widetilde{\mathrm{I}}_k \, \mathbf{R}_t^h \right\rangle \\
\frac{\partial \ell}{\partial \mathbf{w}_i^h} &= \frac{\boldsymbol{\beta}_h^2 e^{\mathbf{w}_i^h}}{\sum_{j=1}^k e^{\mathbf{w}_j^h}} \sum_{t=1}^{k!} \ell_t^{'} \left((1+\gamma)\mathbb{1}\{q_{k-h} = \pi_{(t,k-i)}\} - \left\langle e_{x_{l-h}}, \widetilde{\mathrm{I}}_k \, \mathbf{R}_t^h \right\rangle \right),
\end{aligned}$$

where $\ell_t^{'} = -\mathbf{s}_t \left(\mathbb{1}\{r_t = r\} - \sum_{j=1}^{k!} \mathbf{s}_j \mathbb{1}\{r_j = r\} \right)$.

The proof of the lemma is given above.

Now we will use the above lemma to compute the gradients at a particular parameter configuration, we choose a general parameter configuration so that we can invoke this lemma whenever gradient computations are needed.

Lemma C.2. *Consider a parameter configuration $\mathcal{I}_G$ defined such that for all h and i,*

$$\mathbf{w}^h = \alpha_h \mathsf{e}_h^k + \delta_h \mathbf{1}$$

$\boldsymbol{\beta}_h$ is used as is. Then the gradients at this parameter configuration are given by

$$\frac{\partial \ell}{\partial \boldsymbol{\beta}_h} = 2\boldsymbol{\beta}_h \frac{e^{\alpha_h} - 1}{e^{\alpha_h} + k - 1}(1+\gamma)\Delta_{h,h}, \tag{11}$$

$$\frac{\partial \ell}{\partial \mathbf{w}_i^h} = (1+\gamma)\frac{\boldsymbol{\beta}_h^2 e^{\alpha_h + \delta_h}}{e^{\alpha_h} + k - 1}\left(\Delta_{h,i} - \frac{e^{\alpha_h} - 1}{e^{\alpha_h} + k - 1}\Delta_{h,h}\right) \tag{12}$$

where

$$\Delta_{h,i} = -\left[\sum_{t=1}^{k!} \mathbf{s}_t \mathbb{1}\{r_t = r \wedge q_{k-h} = \pi_{(t,k-i)}\} - \left(\sum_{j=1}^{k!} \mathbf{s}_j \mathbb{1}\{r_j = r\}\right)\left(\sum_{t=1}^{k!} \mathbf{s}_t \left(\mathbb{1}\{q_{k-h} = \pi_{(t,k-i)}\}\right)\right)\right]$$

where $s = \boldsymbol{\sigma}(\mathbf{s}^{'})$ and $\mathbf{s}_t^{'} = \sum_{h=1}^{k} \gamma_h \mathbb{1}\{q_{k-h} = \pi_{(t,k-h)}\}$ with $\gamma_h = (1+\gamma)\boldsymbol{\beta}_h^2 \frac{e^{\alpha_h}-1}{e^{\alpha_h}+k-1}$.

Proof. First let us compute the forward pass

$$\begin{aligned}
\mathbf{R}_t^h &= \sum_{i=1}^{k} \boldsymbol{\sigma}(\mathbf{A}^h)_{t,t-i}\, e_{\pi_{(t,k-i)}} = \sum_{i=1}^{k} \frac{e^{\mathbf{w}_i^h}}{\sum_{j=1}^{k} e^{\mathbf{w}_j^h}}\, e_{\pi_{(t,k-i)}}, \\
&= \frac{e^{\alpha_h + \delta_h}}{(e^{\alpha_h} + k - 1)\, e^{\delta_h}}\, e_{\pi_{(t,k-h)}} + \sum_{i \neq h} \frac{e^{\delta_h}}{(e^{\alpha_h} + k - 1)\, e^{\delta_h}}\, e_{\pi_{(t,k-i)}}, \\
&= \frac{e^{\alpha_h} - 1}{e^{\alpha_h} + k - 1} e_{\pi_{(t,k-h)}} + \frac{1}{e^{\alpha_h} + k - 1}\mu
\end{aligned}$$

Note that $\mu = \sum_{i=1}^{k} e_{\pi_{(t,k-i)}}$ is the same for all t. Now the presoftmax score is given by

$$\begin{aligned}
\widetilde{\mathbf{s}}_t &= \sum_{h=1}^{k} \boldsymbol{\beta}_h^2 \left\langle e_{x_{l-h}}, \widetilde{\mathrm{I}}_k\, \mathbf{R}_t^h \right\rangle, \\
&= \sum_{h=1}^{k} \boldsymbol{\beta}_h^2 \left(\frac{e^{\alpha_h} - 1}{e^{\alpha_h} + k - 1}(1+\gamma)\mathbb{1}\{q_{k-h} = \pi_{(t,k-h)}\} - \gamma - \frac{1}{e^{\alpha_h} + k - 1}(1+\gamma) \right), \\
&= (1+\gamma)\sum_{h=1}^{k} \boldsymbol{\beta}_h^2 \frac{e^{\alpha_h} - 1}{e^{\alpha_h} + k - 1}\mathbb{1}\{q_{k-h} = \pi_{(t,k-h)}\} - (1+\gamma)\sum_{h=1}^{k} \frac{\boldsymbol{\beta}_h^2}{e^{\alpha_h} + k - 1} - \gamma \sum_{h=1}^{k} \boldsymbol{\beta}_h^2.
\end{aligned}$$

Define

$$\gamma_h = (1+\gamma)\boldsymbol{\beta}_h^2 \frac{e^{\alpha_h} - 1}{e^{\alpha_h} + k - 1}. \tag{13}$$

Denote

$$\mathbf{s}_t^{'} = \sum_{h=1}^{k} \gamma_h \mathbb{1}\{q_{k-h} = \pi_{(t,k-h)}\}.$$

Note that $\widetilde{\mathbf{s}}_t = \mathbf{s}_t^{'} + c$ where c is a constant independent of t. Using the property of the softmax we have $\mathbf{s} = \ell(\mathbf{s}^{'})$. Using the lemma C.1 we have

$$\begin{aligned}
\frac{\partial \ell}{\partial \boldsymbol{\beta}_h} &= 2\boldsymbol{\beta}_h \sum_{t=1}^{k!} \ell_t^{'} \left\langle e_{x_{l-h}}, \widetilde{\mathrm{I}}_k \, \mathbf{R}_t^h \right\rangle, \\
&= 2\boldsymbol{\beta}_h \sum_{t=1}^{k!} \ell_t^{'} \left(\frac{e^{\alpha_h} - 1}{e^{\alpha_h} + k - 1}(1+\gamma)\mathbb{1}\{q_{k-h} = \pi_{(t,k-h)}\} - \gamma - \frac{1}{e^{\alpha_h} + k - 1}(1+\gamma) \right), \\
&= 2\boldsymbol{\beta}_h \frac{e^{\alpha_h} - 1}{e^{\alpha_h} + k - 1}(1+\gamma) \sum_{t=1}^{k!} \ell_t^{'} \mathbb{1}\{q_{k-h} = \pi_{(t,k-h)}\} - 2\boldsymbol{\beta}_h \left(\gamma + \frac{1}{e^{\alpha_h} + k - 1}(1+\gamma) \right) \sum_{t=1}^{k!} \ell_t^{'}, \\
&= 2\boldsymbol{\beta}_h \frac{e^{\alpha_h} - 1}{e^{\alpha_h} + k - 1}(1+\gamma) \sum_{t=1}^{k!} \ell_t^{'} \mathbb{1}\{q_{k-h} = \pi_{(t,k-h)}\},
\end{aligned}$$

Now the derivative with respect to $\mathbf{w}_i^h$ is given by

$$\begin{aligned}
\frac{\partial \ell}{\partial \mathbf{w}_i^h} &= \frac{\boldsymbol{\beta}_h^2 e^{\mathbf{w}_i^h}}{\sum_{j=1}^{k} e^{\mathbf{w}_j^h}} \sum_{t=1}^{k!} \ell_t^{'} \left((1+\gamma)\mathbb{1}\{q_{k-h} = \pi_{(t,k-i)}\} - \left\langle e_{x_{l-h}}, \widetilde{\mathrm{I}}_k \, \mathbf{R}_t^h \right\rangle \right), \\
&= \frac{\boldsymbol{\beta}_h^2 e^{\alpha_h + \delta_h}}{e^{\alpha_h} + k - 1} \sum_{t=1}^{k!} \ell_t^{'} \left((1+\gamma)\mathbb{1}\{q_{k-h} = \pi_{(t,k-i)}\} - \left\langle e_{x_{l-h}}, \widetilde{\mathrm{I}}_k \, \mathbf{R}_t^h \right\rangle \right), \\
&= \frac{\boldsymbol{\beta}_h^2 e^{\alpha_h + \delta_h}}{e^{\alpha_h} + k - 1} \sum_{t=1}^{k!} \ell_t^{'} \left((1+\gamma)\mathbb{1}\{q_{k-h} = \pi_{(t,k-i)}\} - \frac{e^{\alpha_h} - 1}{e^{\alpha_h} + k - 1}(1+\gamma)\ell_t^{'}\mathbb{1}\{q_{k-h} = \pi_{(t,k-h)}\} \right), \\
&= (1+\gamma)\frac{\boldsymbol{\beta}_h^2 e^{\alpha_h + \delta_h}}{e^{\alpha_h} + k - 1} \sum_{t=1}^{k!} \ell_t^{'} \left(\mathbb{1}\{q_{k-h} = \pi_{(t,k-i)}\} - \frac{e^{\alpha_h} - 1}{e^{\alpha_h} + k - 1}\mathbb{1}\{q_{k-h} = \pi_{(t,k-h)}\} \right),
\end{aligned}$$

We introduce a notation where

$$\Delta_{h,i} = \sum_{t=1}^{k!} \ell_t^{'} \left(\mathbb{1}\{q_{k-h} = \pi_{(t,k-i)}\} \right)$$

Using this notation, the gradients can be written as

$$\begin{aligned}
\frac{\partial \ell}{\partial \boldsymbol{\beta}_h} &= 2\boldsymbol{\beta}_h \frac{e^{\alpha_h} - 1}{e^{\alpha_h} + k - 1}(1+\gamma)\Delta_{h,h}, \\
\frac{\partial \ell}{\partial \mathbf{w}_i^h} &= (1+\gamma)\frac{\boldsymbol{\beta}_h^2 e^{\alpha_h + \delta_h}}{e^{\alpha_h} + k - 1} \left(\Delta_{h,i} - \frac{e^{\alpha_h} - 1}{e^{\alpha_h} + k - 1}\Delta_{h,h} \right),
\end{aligned}$$

where $\Delta_{h,i} = \sum_{t=1}^{k!} \ell_t^{'} \left(\mathbb{1}\{q_{k-h} = \pi_{(t,k-i)}\} \right)$. Substituting the expression of $\ell_t^{'}$ we have

$$\begin{aligned}
\Delta_{h,i} &= \sum_{t=1}^{k!} -\mathbf{s}_t \left(\mathbb{1}\{r_t = r\} - \sum_{j=1}^{k!} \mathbf{s}_j \mathbb{1}\{r_j = r\} \right) \left(\mathbb{1}\{q_{k-h} = \pi_{(t,k-i)}\} \right), \\
&= -\left[\sum_{t=1}^{k!} \mathbf{s}_t \mathbb{1}\{r_t = r \wedge q_{k-h} = \pi_{(t,k-i)}\} - \left(\sum_{j=1}^{k!} \mathbf{s}_j \mathbb{1}\{r_j = r\} \right) \left(\sum_{t=1}^{k!} \mathbf{s}_t \left(\mathbb{1}\{q_{k-h} = \pi_{(t,k-i)}\} \right) \right) \right],
\end{aligned}$$

$$\Delta_{h,i} = -\left[\sum_{t=1}^{k!} \mathbf{s}_t \mathbb{1}\{r_t = r \wedge q_{k-h} = \pi_{(t,k-i)}\} - \left(\sum_{j=1}^{k!} \mathbf{s}_j \mathbb{1}\{r_j = r\}\right)\left(\sum_{t=1}^{k!} \mathbf{s}_t \left(\mathbb{1}\{q_{k-h} = \pi_{(t,k-i)}\}\right)\right)\right]. \tag{14}$$

where $s = \boldsymbol{\sigma}(\mathrm{s}^{'})$. This completes the proof of the lemma.

□

Lemma C.3. *For the parameter configuration $\boldsymbol{\theta} = (\mathbf{w}^1, \mathbf{w}^2, \dots \mathbf{w}^k, \boldsymbol{\beta})$ where $\mathbf{w}^1 = \alpha_1 \mathrm{e}_h^1 + \delta_1 \mathbf{1}$ and $\boldsymbol{\beta}_h = 0$ for $h \neq 1$. The following quantities are*

$$\mathbb{E}\Delta_{1,i} = \frac{1}{(e^{\gamma_1} + k - 1)^2}\left[\frac{e^{\gamma_1}(|\mathcal{R}| - 1)}{(k-1)!|\mathcal{R}|}\right], \quad \textit{for } i \neq 1.$$

$$\mathbb{E}\Delta_{1,1} = -(k-1)\mathbb{E}\Delta_{1,i} = -\frac{1}{(e^{\gamma_1} + k - 1)^2}\left[\frac{e^{\gamma_1}(|\mathcal{R}| - 1)}{(k-2)!|\mathcal{R}|}\right].$$

where $\gamma_1 = (1+\gamma)\boldsymbol{\beta}_1^2 \frac{e^{\alpha_1} - 1}{e^{\alpha_1} + k - 1}$.

Proof. Recall that $\mathrm{s}_t^{'} = \gamma_1 \mathbb{1}\{q_{k-1} = \pi_{(t,k-1)}\}$ as $\gamma_h = 0$ for $h \neq 1$ and the softmax score is given by

$$\mathbf{s}_t = \frac{e^{\gamma_1 \mathbb{1}\{q_{k-1} = \pi_{(t,k-1)}\}}}{\sum_{j=1}^{k!} e^{\gamma_1 \mathbb{1}\{q_{k-1} = \pi_{(j,k-1)}\}}} = \frac{(e^{\gamma_1} - 1)\,\mathbb{1}\{q_{k-1} = \pi_{(t,k-1)}\} + 1}{e^{\gamma_1} + k - 1} \frac{1}{(k-1)!}.$$

First lets compute $\mathbb{E}\Delta_{1,i}$ for $i \neq 1$, Note that

$$\begin{aligned}\sum_{t=1}^{k!} \mathbf{s}_t \left(\mathbb{1}\{q_{k-1} = \pi_{(t,k-i)}\}\right) &= \sum_{t=1}^{k!} \frac{(e^{\gamma_1} - 1)\,\mathbb{1}\{q_{k-1} = \pi_{(t,k-i)}\} + 1}{e^{\gamma_1} + k - 1} \frac{1}{(k-1)!} \left(\mathbb{1}\{q_{k-1} = \pi_{(t,k-i)}\}\right), \\ &= \frac{1}{e^{\gamma_1} + k - 1}.\end{aligned}$$

For $i = 1$, we have,

$$\begin{aligned}\sum_{t=1}^{k!} \mathbf{s}_t \left(\mathbb{1}\{q_{k-1} = \pi_{(t,k-i)}\}\right) &= \sum_{t=1}^{k!} \frac{(e^{\gamma_1} - 1)\,\mathbb{1}\{q_{k-1} = \pi_{(t,k-i)}\} + 1}{e^{\gamma_1} + k - 1} \frac{1}{(k-1)!} \left(\mathbb{1}\{q_{k-1} = \pi_{(t,k-i)}\}\right), \\ &= \frac{e^{\gamma_1}}{e^{\gamma_1} + k - 1}.\end{aligned}$$

Now we can compute $\mathbb{E}\Delta_{1,i}$ for $i \neq 1$ as follows:

$$\begin{aligned}\mathbb{E}\Delta_{1,i} &= -\mathbb{E}\left[\sum_{t=1}^{k!} \mathbf{s}_t \mathbb{1}\{r_t = r \wedge q_{k-1} = \pi_{(t,k-i)}\} - \left(\sum_{j=1}^{k!} \mathbf{s}_j \mathbb{1}\{r_j = r\}\right)\left(\sum_{t=1}^{k!} \mathbf{s}_t \left(\mathbb{1}\{q_{k-1} = \pi_{(t,k-i)}\}\right)\right)\right], \\ &= -\mathbb{E}\left[\sum_{t=1}^{k!} \mathbf{s}_t \mathbb{1}\{r_t = r \wedge q_{k-1} = \pi_{(t,k-i)}\}\right] + \mathbb{E}\left[\left(\sum_{j=1}^{k!} \mathbf{s}_j \mathbb{1}\{r_j = r\}\right)\frac{1}{e^{\gamma_1} + k - 1}\right],\end{aligned}$$

For $i \neq 1$,

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{k!} \mathbf{s}_t \mathbb{1}\{r_t = r \wedge q_{k-1} = \pi_{(t,k-i)}\}\right] &= \frac{1}{e^{\gamma_1} + k - 1}\frac{1}{(k-1)!}\mathbb{E}\left[\sum_{t=1}^{k!} \mathbb{1}\{r_t = r \wedge q_{k-1} = \pi_{(t,k-i)}\}\right], \\ &= \frac{1}{e^{\gamma_1} + k - 1}\frac{1}{(k-1)!}\sum_{t=1}^{k!} \mathbb{P}(r_t = r \wedge q_{k-1} = \pi_{(t,k-i)}),\end{aligned}$$

Note that for $i \neq 1$, $\mathbb{P}(r_t = r \wedge q_{k-1} = \pi_{(t,k-i)}) = \mathbb{P}(r_t = r)\mathbb{P}(q_{k-1} = \pi_{(t,k-i)})$ due to the independence as q is not same as π_t. $\mathbb{P}(q_{k-1} = \pi_{(t,k-i)}) = \frac{1}{k}$ as q_{k-1} is uniformly distributed over $[k]$. $\mathbb{P}(r_t = r) = \frac{1}{|\mathcal{R}|}$. So,

$$\mathbb{E}\left[\sum_{t=1}^{k!} \mathbf{s}_t \mathbb{1}\{r_t = r \wedge q_{k-1} = \pi_{(t,k-i)}\}\right] = \frac{1}{e^{\gamma_1}+k-1}\frac{1}{(k-1)!}\sum_{t=1}^{k!}\frac{1}{k|\mathcal{R}|} = \frac{1}{e^{\gamma_1}+k-1}\frac{1}{|\mathcal{R}|}.$$

Now for $i = 1$,

$$E\left[\sum_{t=1}^{k!} \mathbf{s}_t \mathbb{1}\{r_t = r \wedge q_{k-1} = \pi_{(t,k-1)}\}\right] = \frac{e^{\gamma_1}}{e^{\gamma_1}+k-1}\frac{1}{(k-1)!}\mathbb{E}\left[\sum_{t=1}^{k!} \mathbb{1}\{r_t = r \wedge q_{k-1} = \pi_{(t,k-1)}\}\right],$$

Coming to the expectation term, now they are no longer independent.

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{k!} \mathbb{1}\{r_t = r \wedge q_{k-1} = \pi_{(t,k-1)}\}\right] &= \sum_{t=1}^{k!} \mathbb{P}(r_t = r \wedge q_{k-1} = \pi_{(t,k-1)}), \\
&= \sum_{t=1}^{k!} \mathbb{P}(r_t = r | q_{k-1} = \pi_{(t,k-1)})\mathbb{P}(q_{k-1} = \pi_{(t,k-1)}), \\
&= \sum_{t=1}^{k!} \left[\mathbb{P}(r_t = r, q = \pi_t \big| q_{k-1} = \pi_{(t,k-1)}) + \mathbb{P}(r_t = r, q \neq \pi_t \big| q_{k-1} = \pi_{(t,k-1)})\right] \mathbb{P}(q_{k-1} = \pi_{(t,k-1)}),
\end{aligned}$$

$$\begin{aligned}
\mathbb{P}(r_t = r, q = \pi_t \big| q_{k-1} = \pi_{(t,k-1)}) &= \mathbb{P}(r_t = r | q = \pi_t, q_{k-1} = \pi_{(t,k-1)})\mathbb{P}(q = \pi_t | q_{k-1} = \pi_{(t,k-1)}), \\
&= 1 \cdot \frac{1}{(k-1)!} = \frac{1}{(k-1)!}. \\
\mathbb{P}(r_t = r, q \neq \pi_t \big| q_{k-1} = \pi_{(t,k-1)}) &= \mathbb{P}(r_t = r | q \neq \pi_t, q_{k-1} = \pi_{(t,k-1)})\mathbb{P}(q \neq \pi_t | q_{k-1} = \pi_{(t,k-1)}), \\
&= \frac{1}{|\mathcal{R}|} \cdot \left[1 - \frac{1}{(k-1)!}\right]
\end{aligned}$$

Summming them up and substituting in the previous equation, we get,

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{k!} \mathbb{1}\{r_t = r \wedge q_{k-1} = \pi_{(t,k-1)}\}\right] &= \sum_{t=1}^{k!}\left[\frac{1}{(k-1)!} + \frac{1}{|\mathcal{R}|}\cdot\left(1 - \frac{1}{(k-1)!}\right)\right]\frac{1}{k}, \\
&= k!\left[\frac{1}{(k-1)!} + \frac{1}{|\mathcal{R}|}\cdot\left(1 - \frac{1}{(k-1)!}\right)\right]\frac{1}{k}, \\
&= \frac{|\mathcal{R}|-1}{|\mathcal{R}|} + \frac{(k-1)!}{|\mathcal{R}|}.
\end{aligned}$$

Substituting this back in the expectation term, we get,

$$\begin{aligned}
E\left[\sum_{t=1}^{k!} \mathbf{s}_t \mathbb{1}\{r_t = r \wedge q_{k-1} = \pi_{(t,k-1)}\}\right] &= \frac{e^{\gamma_1}}{e^{\gamma_1}+k-1}\frac{1}{(k-1)!}\mathbb{E}\left[\sum_{t=1}^{k!} \mathbb{1}\{r_t = r \wedge q_{k-1} = \pi_{(t,k-1)}\}\right] \\
&= \frac{e^{\gamma_1}}{e^{\gamma_1}+k-1}\frac{1}{(k-1)!}\left[\frac{|\mathcal{R}|-1}{|\mathcal{R}|} + \frac{(k-1)!}{|\mathcal{R}|}\right], \\
&= \frac{e^{\gamma_1}}{e^{\gamma_1}+k-1}\left[\frac{|\mathcal{R}|-1}{(k-1)!|\mathcal{R}|}\right] + \frac{e^{\gamma_1}}{e^{\gamma_1}+k-1}\frac{1}{|\mathcal{R}|}.
\end{aligned}$$

Recall the term for $i \neq i$, we have,

$$\mathbb{E}\left[\sum_{t=1}^{k!} \mathbf{s}_t \mathbb{1}\{r_t = r \wedge q_{k-1} = \pi_{(t,k-i)}\}\right] = \frac{1}{e^{\gamma_1}+k-1}\frac{1}{(k-1)!}\sum_{t=1}^{k!}\frac{1}{k|\mathcal{R}|} = \frac{1}{e^{\gamma_1}+k-1}\frac{1}{|\mathcal{R}|}.$$

Note that

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{k!}\mathbf{s}_t\mathbb{1}\{r_t = r\}\right] &= \sum_{i=1}^{k}\mathbb{E}\left[\sum_{t=1}^{k!}\mathbf{s}_t\mathbb{1}\{r_t = r \wedge q_{k-1} = \pi_{(t,k-i)}\}\right],\\ &= \frac{e^{\gamma_1}}{e^{\gamma_1}+k-1}\left[\frac{|\mathcal{R}|-1}{(k-1)!|\mathcal{R}|}\right] + \frac{1}{|\mathcal{R}|}.\end{aligned}$$

Now for $i \neq 1$, we have,

$$\begin{aligned}\mathbb{E}\Delta_{1,i} &= -\mathbb{E}\left[\sum_{t=1}^{k!}\mathbf{s}_t\mathbb{1}\{r_t = r \wedge q_{k-1} = \pi_{(t,k-i)}\} - \left(\sum_{j=1}^{k!}\mathbf{s}_j\mathbb{1}\{r_j = r\}\right)\frac{1}{e^{\gamma_1}+k-1}\right],\\ &= -\left[\frac{1}{e^{\gamma_1}+k-1}\frac{1}{|\mathcal{R}|} - \left(\frac{e^{\gamma_1}}{e^{\gamma_1}+k-1}\left[\frac{|\mathcal{R}|-1}{(k-1)!|\mathcal{R}|}\right] + \frac{1}{|\mathcal{R}|}\right)\frac{1}{e^{\gamma_1}+k-1}\right],\\ &= \frac{1}{(e^{\gamma_1}+k-1)^2}\left[\frac{e^{\gamma_1}(|\mathcal{R}|-1)}{(k-1)!|\mathcal{R}|}\right].\end{aligned}$$

Now for $i = 1$, we have,

$$\begin{aligned}\mathbb{E}\Delta_{1,1} &= -\mathbb{E}\left[\sum_{t=1}^{k!}\mathbf{s}_t\mathbb{1}\{r_t = r \wedge q_{k-1} = \pi_{(t,k-1)}\} - \left(\sum_{j=1}^{k!}\mathbf{s}_j\mathbb{1}\{r_j = r\}\right)\frac{e^{\gamma_1}}{e^{\gamma_1}+k-1}\right],\\ &= -\left[\frac{e^{\gamma_1}}{e^{\gamma_1}+k-1}\left[\frac{|\mathcal{R}|-1}{(k-1)!|\mathcal{R}|}\right] + \frac{e^{\gamma_1}}{e^{\gamma_1}+k-1}\frac{1}{|\mathcal{R}|} - \left(\frac{e^{\gamma_1}}{e^{\gamma_1}+k-1}\left[\frac{|\mathcal{R}|-1}{(k-1)!|\mathcal{R}|}\right] + \frac{1}{|\mathcal{R}|}\right)\frac{e^{\gamma_1}}{e^{\gamma_1}+k-1}\right],\\ &= -\left[\frac{e^{\gamma_1}}{e^{\gamma_1}+k-1}\left[\frac{|\mathcal{R}|-1}{(k-1)!|\mathcal{R}|}\right] - \frac{\left(e^{\gamma_1}\right)^2}{(e^{\gamma_1}+k-1)^2}\left[\frac{|\mathcal{R}|-1}{(k-1)!|\mathcal{R}|}\right]\right],\\ &= -\left[\frac{e^{\gamma_1}(k-1)}{(e^{\gamma_1}+k-1)^2}\left[\frac{|\mathcal{R}|-1}{(k-1)!|\mathcal{R}|}\right]\right] = -(k-1)\mathbb{E}\Delta_{1,i} \text{ for } i \neq 1.\end{aligned}$$

□

D Technical Lemmas

Gradient of the first layer. For both relative positional encoding, the gradient of the first layer can be computed as follows.

Lemma D.1. *Denote the forward pass of the first layer as follows,*

$$\mathbf{R}^h = \boldsymbol{\sigma}(\mathbf{A}^h)\,X, \quad \mathbf{R}^h_t = X^\top\boldsymbol{\sigma}(\mathbf{A}^h)_t = \sum_{i=0}^{t}\boldsymbol{\sigma}(\mathbf{A}^h)_{t,i}\,\mathsf{e}^{\mathcal{S}}_{x_i},$$

The gradient of the embeddings with respect to the parameters $\mathbf{w}^h_i$ is given by

$$\frac{\partial\mathbf{R}^h_t}{\partial\mathbf{w}^h_i} = \begin{cases}\frac{e^{\mathbf{w}^h_i}}{\sum_{j=0}^{t}e^{\mathbf{w}^h_j}}\left(\mathsf{e}^{\mathcal{S}}_{x_{t-i}} - \mathbf{R}^h_t\right) & \text{if } i \leqslant t,\\ 0 & \text{if } i > t,\end{cases}$$

Proof.

$$\begin{aligned}\mathbf{R}^h &= \boldsymbol{\sigma}(\mathbf{A}^h)\,X,\\ \mathbf{R}^h_t &= X^\top\boldsymbol{\sigma}(\mathbf{A}^h)_t = \sum_{i=0}^{t}\boldsymbol{\sigma}(\mathbf{A}^h)_{t,i}\,\mathsf{e}^{\mathcal{S}}_{x_i}\end{aligned}$$

Recall that the matrix $\mathbf{A}^h$ is defined as follows,

$$\mathbf{A}^h = \begin{bmatrix} \mathbf{w}_0^h & 0 & 0 & \dots & 0 \\ \mathbf{w}_1^h & \mathbf{w}_0^h & 0 & \dots & 0 \\ \mathbf{w}_2^h & \mathbf{w}_1^h & \mathbf{w}_0^h & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{w}_{l-2}^h & \mathbf{w}_{l-3}^h & \dots & \mathbf{w}_0^h & 0 \\ \mathbf{w}_{l-1}^h & \mathbf{w}_{l-2}^h & \mathbf{w}_{l-3}^h & \dots & \mathbf{w}_0^h \end{bmatrix}$$

Now,

$$\begin{aligned} \boldsymbol{\sigma}(\mathbf{A}^h)_{t,i} &= \frac{e^{\mathbf{w}_{t-i}^h}}{\sum\limits_{j=0}^{t} e^{\mathbf{w}_j^h}}, \\ \mathbf{R}_t^h &= \sum_{i=0}^{t} \frac{e^{\mathbf{w}_{t-i}^h}}{\sum_{j=0}^{t} e^{\mathbf{w}_j^h}} \, \mathsf{e}_{x_i}^{\mathcal{S}} = \sum_{i=0}^{t} \frac{e^{\mathbf{w}_i^h}}{\sum_{j=0}^{t} e^{\mathbf{w}_j^h}} \, \mathsf{e}_{x_{t-i}}^{\mathcal{S}}, \\ \text{For } i \leqslant t, \quad \frac{\partial \mathbf{R}_t^h}{\partial \mathbf{w}_i^h} &= \frac{e^{\mathbf{w}_i^h}}{\sum_{j=0}^{t} e^{\mathbf{w}_j^h}} \, \mathsf{e}_{x_{t-i}}^{\mathcal{S}} - \frac{e^{\mathbf{w}_i^h}}{\left(\sum_{j=0}^{t} e^{\mathbf{w}_j^h}\right)^2} \sum_{j=1}^{t} e^{\mathbf{w}_j^h} \mathsf{e}_{x_{t-j}}^{\mathcal{S}}, \\ &= \frac{e^{\mathbf{w}_i^h}}{\sum_{j=0}^{t} e^{\mathbf{w}_j^h}} \left(\mathsf{e}_{x_{t-i}}^{\mathcal{S}} - \mathbf{R}_t^h\right) \end{aligned}$$

So, we have the result as follows,

$$\frac{\partial \mathbf{R}_t^h}{\partial \mathbf{w}_i^h} = \begin{cases} \frac{e^{\mathbf{w}_i^h}}{\sum_{j=0}^{t} e^{\mathbf{w}_j^h}} \left(\mathsf{e}_{x_{t-i}}^{\mathcal{S}} - \mathbf{R}_t^h\right) & \text{if } i \leqslant t, \\ 0 & \text{if } i > t, \end{cases}$$

□

Now in the context of simplified transformer model with a fixed window we have the following

Lemma D.2. *Denote the forward pass of the first layer as follows,*

$$\mathbf{R}^h = \boldsymbol{\sigma}(\mathbf{A}^h)\, \mathsf{X}, \quad \mathbf{R}_t^h = \mathsf{X}^\top \boldsymbol{\sigma}(\mathbf{A}^h)_t = \sum_{i=1}^{k} \boldsymbol{\sigma}(\mathbf{A}^h)_{t,t-i}\, e_{\pi_{(t,k-i)}},$$

The gradient of the embeddings with respect to the parameters $\mathbf{w}_i^h$ is given by

$$\frac{\partial \mathbf{R}_t^h}{\partial \mathbf{w}_i^h} = \frac{e^{\mathbf{w}_i^h}}{\sum\limits_{j=1}^{k} e^{\mathbf{w}_j^h}} \left(e_{\pi_{(t,k-i)}} - \mathbf{R}_t^h\right)$$

Proof.

$$\mathbf{R}^h = \boldsymbol{\sigma}(\mathbf{A}^h)\, \mathsf{X}, \quad \mathbf{R}_t^h = \mathsf{X}^\top \boldsymbol{\sigma}(\mathbf{A}^h)_t = \sum_{i=1}^{k} \boldsymbol{\sigma}(\mathbf{A}^h)_{t,t-i}\, e_{\pi_{(t,k-i)}},$$

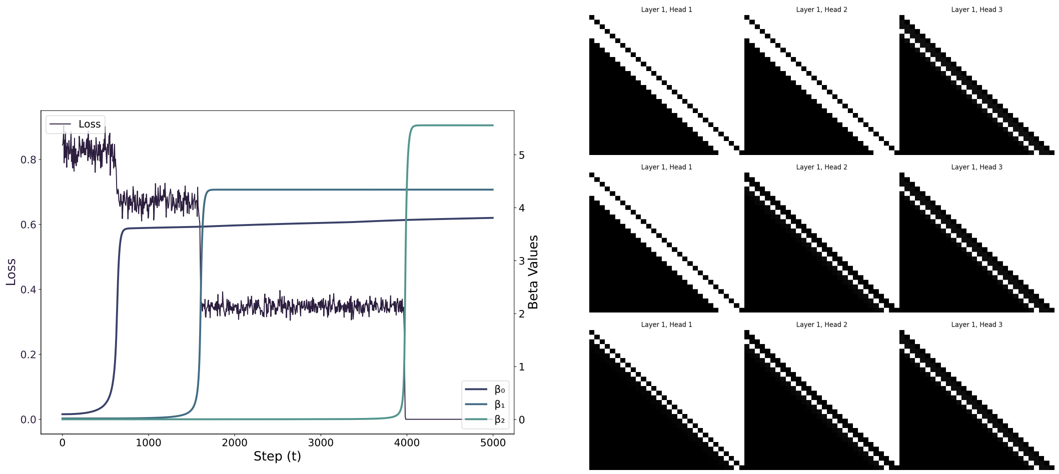
$$\boldsymbol{\sigma}(\mathbf{A}^h)_{t,t-i} = \frac{e^{\mathbf{w}_i^h}}{\sum\limits_{j=1}^{k} e^{\mathbf{w}_j^h}},$$

$$\mathbf{R}_t^h = \sum_{i=1}^{k} \frac{e^{\mathbf{w}_i^h}}{\sum\limits_{j=1}^{k} e^{\mathbf{w}_j^h}}\, e_{\pi_{(t,k-i)}},$$

$$\text{For } 1 \leqslant i \leqslant k, \quad \frac{\partial \mathbf{R}_t^h}{\partial \mathbf{w}_i^h} = \frac{e^{\mathbf{w}_i^h}}{\sum\limits_{j=1}^{k} e^{\mathbf{w}_j^h}} \left(e_{\pi_{(t,k-i)}} - \mathbf{R}_t^h\right)$$

□

E Additional Experiments



(a) (b)

Figure 4: The left panel is shown in a and the right panel in b. Note that the β values rise as the loss drops, and the attention patterns at times 800, 1750, and 4500 demonstrate incremental learning. This is in the opposite order as they are initialized in the opposite order $\beta_3 > \beta_2 > \beta_1$