Incremental Learning in Transformers for In-Context Associative Recall

Aditya Varre*
Inria, École Normale Supérieure, PSL Research University, Paris
aditya.varre@inria.fr

Nicolas Flammarion
Theory of Machine Learning Lab, EPFL, Lausanne
nicolas.flammarion@epfl.ch

Abstract

Transformers acquire in-context learning abilities through abrupt phases during training, often unfolding over multiple stages in which key circuits, such as induction heads, emerge. In this work, we characterize the dynamics underlying the emergence of such circuits across these stages. We focus on a synthetic associative recall task, where sequences are drawn from random maps between a permutation group and a vocabulary, and the model is required to complete the mapping of a permutation by retrieving the corresponding association from context. For this task, we study the gradient-flow trajectories of a simplified two-layer attention-only transformer. Leveraging symmetries in both the transformer architecture and the data distribution, we derive *closed-form parameter dynamics* for both attention layers. We identify a conservation law that couples parameters across layers and governs their joint learning. This law reveals how initialization controls the *timescales* over which such circuits emerge. In the vanishing-initialization limit, we characterize the gradient-flow trajectory, showing how training jumps from saddle to saddle. Finally, we provide empirical evidence across different architectural choices, validating our simplifications and extending the insights from our analysis beyond the simplified setting.

1 Introduction

In-context learning (ICL) [Brown, 2020], the ability of a model to perform new tasks from examples provided in its prompt without parameter updates, is characteristic of transformers. Beyond what these models can do in context, how these abilities emerge during training remains poorly understood. Empirical works [Olsson et al., 2022, Chen et al., 2024a] 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 (identifying which architectural and optimization biases make ICL learnable by gradient descent) and for practice (how hyperparameters govern convergence speed and training stability).

While recent analyses [Nichani et al., 2024, Chen et al., 2024b] have advanced understanding, they fall short of a complete explanation. Approaches based on layerwise training [Nichani et al., 2024] or highly simplified architectures [Zhang et al., 2025] (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

*Work done while at EPFL.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

that predicts the order in which partial circuits appear, and that explains what controls the length of each phase, including sensitivity to initialization scale and other hyperparameters.

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 over time. This interface viewpoint goes beyond static descriptions of trained models by explaining why training lingers on particular plateaus and when it departs them.

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. Formally, the task is factual recall: the model sees a sequence containing a length-n permutation that reappears later and must locate the earlier copy of its query and recall the associated response (e.g., $[A][B][C][X]\cdots[A][B][C]$). In our construction $k = n - 1$ matched positions already determine the permutation. The same induction-style circuit solves the task, while the permutation structure makes its dynamics easier to analyze.

Crucially, we couple this task with a series of principled simplifications, each motivated by the circuit that actually solves the task and by typical training behavior. These simplifications isolate the essential components of the transformer responsible for incremental learning, while removing spurious elements that obscure analysis but do not qualitatively affect circuit formation. The resulting model remains rich enough to display the multi-phase, non-convex dynamics observed in practice.

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. We show that these intermediate solutions align with sub-circuits of the full recall mechanism. This perspective explains both the order in which partial solutions appear and what controls the duration of each phase, providing predictions for the effects of initialization scale and related hyperparameters. Beyond clarifying that induction-like behavior emerges, we characterize how it emerges under gradient-based training.

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

1.1 Related works

In-context learning. The phenomenon of in-context learning (ICL) [Brown, 2020] has been investigated from several perspectives. Mechanistic interpretability has identified induction heads as key circuits supporting ICL [Olsson et al., 2022]. 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 [Chen et al., 2024a, Kim et al., 2024]. These dynamics have been observed in regression tasks [Garg et al., 2022, Von Oswald et al., 2023, Ahn et al., 2024], formal language recognition [Bhattamishra et al., 2023, Akyürek et al., 2024], and n-gram prediction.

n-grams and induction heads. The circuit solving our task is the same multi-head induction circuit that solves in-context n-gram learning. n-gram language models [Shannon, 1948, Jurafsky and Martin, 2009] provide a natural testbed: the optimization landscape has been analyzed by Makkuva et al. [2024], expressivity over n-gram distributions by Svete and Cotterell [2024], and in-context generalization by Rajaraman et al. [2024]. Other studies connect ICL to induction heads [Bietti et al., 2024] and their acquisition via gradient descent [Nichani et al., 2024]. Edelman et al. [2024] identify stage-wise dynamics where intermediate solutions resemble sub-n-grams, while Varre et al. [2025] formalize sub-n-grams as near-stationary points. Chen et al. [2024b] study the same task with an architecture that enforces head specialization from the start, eliminating the stage-wise dynamics central to our analysis. Mușat [2025] prove that attention heads always emerge in a fixed sequence governed by an implicit curriculum. Musat et al. [2026] studied the recall task with single attention head and proved a context-length dependent emergence time.

Incremental learning and saddle-to-saddle dynamics. Plateau-shaped learning curves arise broadly in neural networks. Related characterizations appear in matrix and tensor factorization [Razin et al., 2021, Jiang et al., 2022], matrix sensing [Arora et al., 2019, Li et al., 2021, Jin et al., 2023], diagonal and linear networks [Saxe et al., 2019, Jacot et al., 2021, Berthier, 2022, Pesme and Flammarion, 2023, Varre et al., 2023, 2024], ReLU networks [Boursier et al., 2022, Abbe et al., 2023], and simplified transformer architectures [Boix-Adsera et al., 2023]. Nichani et al. [2025] study stage-wise dynamics in factual recall with linear attention. Zhang et al. [2026] unify simplicity bias across architectures via saddle-to-saddle dynamics, extending results to self-attention models. Kunin et al. [2025] introduce alternating gradient flows as a unifying framework for feature learning from small initialization, producing the same staircase loss structure we observe. Gopalani and Hu [2025] provide a mechanistic account of what happens during loss plateaus in transformers, directly complementing our plateau-duration analysis. Yüksel et al. [2026] study a standalone (not in-context) high-order Markov chain task with a single layer softmax attention transformer and show stage-wise behaviour. Incremental dynamics are closely coupled with invariants of the training dynamics. Conservation laws constrain gradient-flow trajectories in neural networks. Marcotte et al. [2025] derive conservation laws for single attention layers. In our case, we establish a cross-layer conservation law that governs the formation of induction heads.

2 Problem Setting

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 the vector of all ones.

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 [Olsson et al., 2022] learns to complete simple repeating patterns, e.g., $[A][B][C]\ldots[A]$. 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 sequence $[A][B][C]$, locate its previous occurrence in the context, and then use it to predict the next token $[X]$.

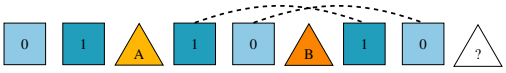
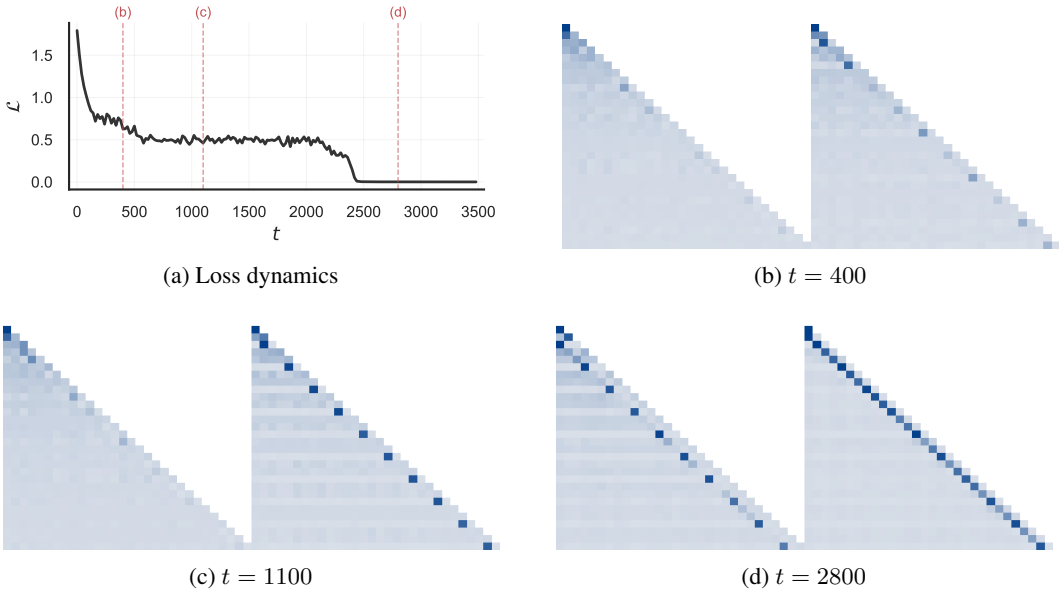


Figure 1: **In-context associative recall task.** Rectangles (0, 1) are query permutation elements and triangles (A, B) are response tokens. The model must predict the missing association "?".

Formally, let $k > 1$ be the number of first-layer heads and $n = k + 1$ the *task order*. Let $\mathcal{P}_n$ be the set of all $n!$ permutations of $[n] = \{0, 1, \ldots, n-1\}$, indexed by $\pi_1, \ldots, \pi_{n!}$. Each permutation is a string of n tokens. Let $\mathcal{R}$ be a response set with $|\mathcal{R}| > 1$, $\mathcal{F} = \{f : \mathcal{P}_n \to \mathcal{R}\}$, and $\mathcal{D}(\mathcal{F})$ the uniform distribution on $\mathcal{F}$. Draw $f_\tau \sim \mathcal{D}(\mathcal{F})$ and, independently, $q \in \mathcal{P}_n$ uniformly. The input contains every permutation and its response, followed by the query:

$$\underbrace{\pi_{(1,0)}, \ldots, \pi_{(1,n-1)}}_{\pi_1},\ f_\tau(\pi_1),\ \underbrace{\pi_{(2,0)}, \ldots, \pi_{(2,n-1)}}_{\pi_2},\ f_\tau(\pi_2),\ \ldots,\ \underbrace{q_0, \ldots, q_{n-1}}_{q},\ ?.$$

Figure 1 illustrates the construction with two symbols and response set $\{A, B\}$. The vocabulary is $\mathcal{S} = [n] \cup \mathcal{R}$, and every input has length $l = (n+1)n! + n$. Matching $k = n-1$ positions of a permutation already determines its final symbol, so k heads suffice. 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.



(a) Loss dynamics

(b) $t = 400$

(c) $t = 1100$

(d) $t = 2800$

Figure 2: **Dynamics of the full transformer** ($n = 2$)**. (a)** Loss curve, exhibiting two abrupt drops. **(b–d)** Attention patterns of Heads 1 and 2 across the three stages. At t=400 the head attends nowhere in particular; in the second stage (t=1100) Head 2 learns to attend to the (-1)-token at every response position; in the final stage (t=2800) Head 1 attends to the (-2)-token, solving the task.

2.2 From the full transformer to a simplified model

In this section, we inspect how a transformer solves this task. Taking a close look at the trajectory of the gradient flow, we make a sequence of simplifications and arrive at a simplified model which we analyze going forward.

Multi-head transformer. We analyze a two-layer transformer where the first layer contains k attention heads, and the second layer contains a single head. 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 $E(i) = \mathrm{e}_i^{|\mathcal{S}|}$ for $i \in [n]$ and $E(r_i) = \mathrm{e}_{n+i}^{|\mathcal{S}|}$. For convenience, we omit the superscript $|\mathcal{S}|$ in what follows. After the encoding layer, the input sequence $x_0, x_1, \ldots, x_{l-1}$ is given by $X = [e_{x_0}\ e_{x_1}\ \ldots\ e_{x_{l-1}}]^\top \in \mathbb{R}^{l \times |\mathcal{S}|}$.

Stage-wise dynamics in the full transformer. With the orthogonal token embeddings, we first train a full transformer (relative positional encoding, MLP layers, LayerNorm, skip connections; $n = 2$, vocabulary size 6, two heads in the first layer, one in the second) on the standard cross-entropy loss. Figure 2 shows a striking property of the training process: it proceeds in distinct stages, with clear transitions in the loss curve and incremental specialization of attention heads. At each stage, a head specializes to a particular relative position, and the loss plateaus until the next head specializes to a different position. First the transformer learns to attend to the previous token (Head 2 attends at response positions, i.e., $x_i \in \mathcal{R}$), then to the token before that, and so on until it has fully learned the task. This stage-wise specialization is consistent with prior observations on n-grams [Edelman et al., 2024, Varre et al., 2025] and motivates our subsequent simplifications.

First attention layer (parameterization). The attention patterns in Figure 2(b–d) carry no semantic content beyond positional matching: each head h just learns a single relative position. Motivated by this observation, we parameterize the first attention layer by a per-head relative-position vector $\mathbf{w}^i \in \mathbb{R}^l$, with no Q/K/V matrices on this layer. The first layer has k heads; the pre-softmax score of head i forms a lower-triangular matrix with entries

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

The head outputs $\mathbf{R}^i = \boldsymbol{\sigma}(\mathbf{A}^i)\, X \in \mathbb{R}^{l \times |\mathcal{S}|}$, where $\boldsymbol{\sigma}$ is the row-wise softmax with causal masking.

Table 1: The disentangled and simplified model. The simplifications are highlighted in color and they match the tags of assumptions. Here $i, t \in [n, l-1]$.

Original model	**Simplified model**
$\mathbf{R}_i^h = \sum_{j=0}^{i} \boldsymbol{\sigma}(\mathbf{A}^h)_{ij} e_{x_j},$	$\mathbf{R}_i^h = \sum_{j=i-n}^{i-1} \boldsymbol{\sigma}(\mathbf{A}^h)_{ij} e_{x_j},$
$\widetilde{\mathbf{s}}_t = \sum_{g,h=0}^{k} [\mathbf{R}_{l-1}^h]^\top [\mathbf{Q}^\top \mathbf{K}]_{hg} [\mathbf{R}_t^g],$	$\widetilde{\mathbf{s}}_t = \sum_{h=1}^{k} [e_{x_{l-h}}]^\top \boldsymbol{\beta}_h^2 [\widetilde{\mathrm{I}}_n] [\mathbf{R}_t^h],$
$\mathbf{s} = \boldsymbol{\sigma}(\widetilde{\mathbf{s}}),\ \mathbf{p} = \sum_{t=0}^{l-1} \mathbf{s}_t e_{x_t}.$	$\mathbf{s} = \boldsymbol{\sigma}(\widetilde{\mathbf{s}}_{\mathcal{R}}),\ \mathbf{p} = \sum_{t: x_t \in \mathcal{R}} \mathbf{s}_t e_{x_t}.$

Disentangled model. We further apply simplifications standard in the theoretical literature. We work with a disentangled attention-only model [Edelman et al., 2024, Rajaraman et al., 2024] that excludes MLP and normalization layers, and disentangles the skip connection: the first-layer output $\mathbf{R}$ is the concatenation of the skip connection ($\mathbf{R}^0 = X$) with the head outputs. The second layer is a single head with parameters $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{(k+1)|\mathcal{S}| \times (k+1)|\mathcal{S}|}$:

$$\mathbf{R} = [\mathbf{R}^0 \quad \mathbf{R}^1 \quad \dots \quad \mathbf{R}^k] \in \mathbb{R}^{l \times (k+1)|\mathcal{S}|}, \qquad \mathbf{R}_+ = \boldsymbol{\sigma}\left(\mathbf{R}\mathbf{Q}^\top \mathbf{K} \mathbf{R}^\top\right) \mathbf{R} V.$$

We fix the value matrix $V = \mathrm{e}_0^{k+1} \otimes \mathrm{I}_{|\mathcal{S}|} \in \mathbb{R}^{(k+1)|\mathcal{S}| \times |\mathcal{S}|}$, which extracts the skip connection: $\mathbf{R}V = \mathbf{R}^0 = X$. Fixing V is benign — Bietti et al. [2024] observe that V is learned quickly during training. The last-token output is then

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

where $\widetilde{\mathbf{s}}, \mathbf{s}$ are the pre- and post-softmax scores; the orthogonal embeddings ensure $\mathbf{p} \in \Delta^{|\mathcal{S}|}$.

Stage-wise dynamics persist; structure of layer-2 $\mathbf{Q}^\top \mathbf{K}$. Figures 3 and 7(a) confirm that, when trained with this disentangled architecture, the loss still drops in distinct stages. The disentangled nature of the transformer makes the second layer interpretable. The second layer also incrementally learns specific structures. $Q^\top K$ has a block structure, each block is responsible for comparing the outputs of head g and head h. The shape of these blocks is along $\widetilde{\mathrm{I}}_n$ where the first $n \times n$ block (query component) is given by $\widetilde{\mathrm{I}}_{:n,:n} = \mathrm{I}_n - \gamma \mathbf{1}_n \mathbf{1}_n^\top$, for some $\gamma > 0$, with zeros elsewhere (response components). Intuitively $\mathbf{Q}^\top \mathbf{K} = B \otimes \widetilde{\mathrm{I}}_n$ where $B \in \mathbb{R}^{(k+1)\times(k+1)}$ gives the block scales. Notably, the blocks above the principal block-diagonal dominate the learning. Motivated by this structure, we define the *simplified model* by parameterizing the second-layer $\mathbf{Q}^\top \mathbf{K}$ directly on the super-diagonal block manifold,

$$\mathbf{Q}^\top \mathbf{K} = \mathrm{diag}_1\left(\boldsymbol{\beta}^2\right) \otimes \widetilde{\mathrm{I}}_n, \qquad \boldsymbol{\beta} \in \mathbb{R}^k.$$

Here $\boldsymbol{\beta}$ is the trainable parameters and we do not have to analyze the complete matrices. We list this together with further simplifications enabling closed-form analysis in the next subsection.

The transformer's solution. At the end of training, the relative positional encoding ensures that head h outputs the embedding of x_{i-h} for token i. With this choice of B with $\boldsymbol{\beta} = c\mathbf{1}$ for large c, the pre-softmax attention scores are $\widetilde{\mathbf{s}}_i \approx c \sum_{h=1}^{k} \mathbb{1}\{x_{l-h} = x_{i-h}\}$ (see Figure 8 in Appendix A for intuition) which is maximized when the histories (queries) of i and the last token match. For large c 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 the correct response $\mathbf{p} = \sum_j \mathbf{s}_j e_{x_j} \approx e_{x_i}$.

2.3 The simplified model: assumptions for closed-form analysis

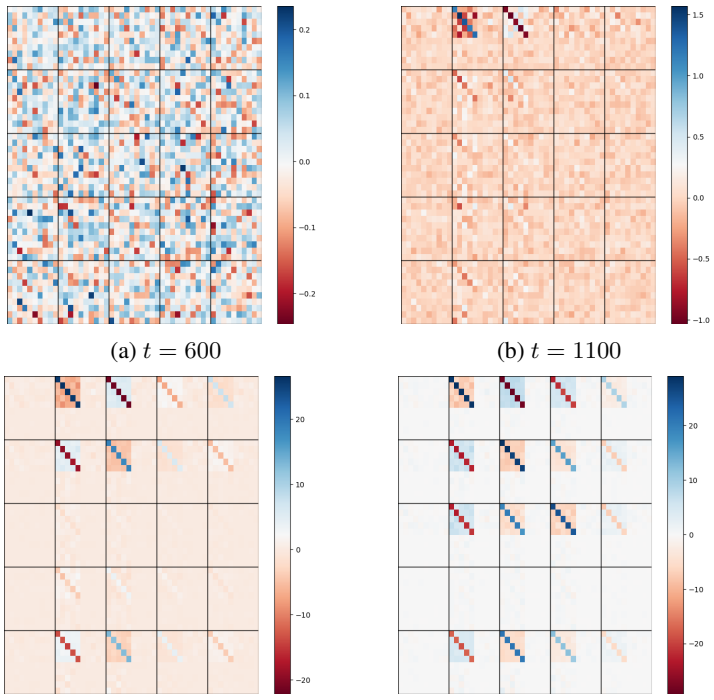
We now state the simplified model formally as four assumptions on the architecture of Section 2.2: A1 fixes the second-layer block structure motivated above, while A0, A2, A3 are technical conveniences enabling closed-form dynamics.

Assumption 2.1 (Attention window). ***(A0)*** *We fix the attention window in the first layer to $n = k + 1$ for all heads. For each head h, we train only the weights $\mathbf{w}_j^h$ for $1 \leqslant j \leqslant n$; all remaining entries are masked to $-\infty$.*

Assumption 2.2 (Second-layer). ***(A1)*** *We parameterize the second layer by introducing $\boldsymbol{\beta} \in \mathbb{R}^k$,*

$$\mathbf{Q}^\top \mathbf{K} = \mathrm{diag}_1\left(\boldsymbol{\beta}^2\right) \otimes \widetilde{\mathrm{I}}_n,$$

where $\mathrm{diag}_1(u) \in \mathbb{R}^{(k+1)\times(k+1)}$ places u on its first super-diagonal.



(a) $t = 600$ (b) $t = 1100$

(c) $t = 1600$ (d) $t = 2300$

Figure 3: **Structure of $Q^\top K$ (n=5).** Starting from a random Gaussian initialization (panel (a), $t = 600$, unstructured), the sub-diagonal block structure is maintained along the trajectory (panels (b)–(d)). A cleaner picture is given in Figure 7 in Appendix A.

Assumption 2.3 (Fixed last-token embedding)**.** ***(A2)*** *We replace the embedding of the last sequence token with its value at the solution.*

Assumption 2.4 (Response-only softmax)**.** ***(A3)*** *Since the correct output always lies in $\mathcal{R}$, we trim $\widetilde{\mathbf{s}}$ to the response coordinates and apply the softmax only to the coordinates corresponding to responses. This ensures the output is a probability vector over $\mathcal{R}$.*

We will comment on these assumptions now. The first-layer attention patterns observed in Section 2.2 show that entries outside the window are redundant, so this fix removes only a redundant degree of freedom while making closed-form ODEs available. Squaring $\boldsymbol{\beta}$ enforces positivity and preserves 2-homogeneity of $Q^\top K$. Assumption 2.3 removes a bilinearity in the first-layer outputs and simplifies gradient computations without qualitatively affecting the dynamics. Their limitations are discussed analytically in Section 3; the relaxations of A0 and A3 are validated empirically in Appendix F.0.2, and the relaxation of A1 in Figure 3 (random Q, K).

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

2.4 The final problem setup

Assumption 2.5 (DP loss)**.** ***(A4)*** *The cross-entropy (CE) is replaced with the dot-product (DP) loss:*

$$\ell(\mathbf{p}, \mathbf{p}_*) = 1 - \langle \mathbf{p}, \mathbf{p}_* \rangle , \qquad \ell_{\mathrm{CE}}(\mathbf{p}, \mathbf{p}_*) = - \langle \mathbf{p}_*, \log \mathbf{p} \rangle . \tag{2}$$

Since every input has a unique correct response, $\mathbf{p}_*$ is one-hot, and both sample losses decrease as the probability of that response increases. Their gradients satisfy $\nabla_{\mathbf{p}} \ell = -\mathbf{p}_*$ and $\nabla_{\mathbf{p}} \ell_{\mathrm{CE}} = - \langle \mathbf{p}, \mathbf{p}_* \rangle^{-1} \mathbf{p}_*$. This positive factor varies across samples, so it need not produce the same population gradient flow. We use DP for the exact dynamics below; the conservation law also holds for CE (Remark A.2), while our CE timescale evidence is empirical (Remark A.1). The population DP loss is

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{f_\tau \sim \mathcal{D}(\mathcal{F}), q \sim \mathcal{P}_n} \, \ell(\mathbf{p}(\boldsymbol{\theta}), e_{f_\tau(q)}) = 1 - \mathbb{E}_{f_\tau, q} \left\langle \mathbf{p}(\boldsymbol{\theta}), e_{f_\tau(q)} \right\rangle . \tag{3}$$

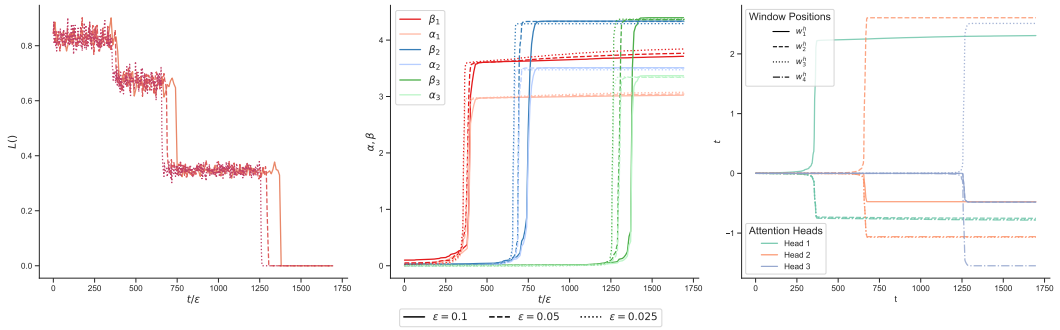


Figure 4: **Simplified two-layer transformer trained with cross-entropy loss under assumptions (A0–A3).** We initialize the parameter vector $\boldsymbol{\beta}$ as $\boldsymbol{\beta} = \epsilon(\beta_1, \beta_2, \beta_3)$ where $\beta_1 > \beta_2 > \beta_3$, for different values of ϵ. The dynamics are plotted with respect to the rescaled time t/ϵ.

Gradient flow. To analyze 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

$$\dot{\boldsymbol{\beta}}_h = -\frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}_h}, \quad \text{and} \quad \dot{\mathbf{w}}_i^h = -\frac{\partial \mathcal{L}}{\partial \mathbf{w}_i^h}, \quad h \in \{1, \ldots, k\},\ i \in \{1, \ldots, n\}.$$

3 Technical Results

We study how signals propagate through the layers of a transformer and how the associated training dynamics unfold. In this section k denotes the number of heads and $n = k + 1$ the task order. We apply A0–A4 with an n-position first-layer window and the n-symbol block $\widetilde{\mathrm{I}}_n$; the unused offset has no dedicated head.

Stage-wise dynamics in the small-ϵ limit. With the simplifications in the previous section, we now analyze the stage-wise dynamics already previewed there. We train the simplified transformer model with SGD and momentum on the DP loss over the in-context learning task of order n — a k-head transformer with $k = n - 1$ can solve this task exactly, since matching k tokens in a permutation determines the remaining one. With $n = 4$ (so $k = 3$ heads) and initialization scale ϵ (time rescaled as t/ϵ), training proceeds in distinct stages: the model lingers at plateaus of decreasing loss and then transitions abruptly (Figure 4). This stage-wise behavior becomes more pronounced as $\epsilon \to 0$ and is reminiscent of the saddle-to-saddle dynamics observed in deep networks under small-scale initialization [Jacot et al., 2021, Pesme and Flammarion, 2023].

To mechanistically interpret these intermediate stages, we analyze what the model represents on each plateau, see Figure 4 and Figure 5 in Appendix A. 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, the matched tokens determine the remaining symbol and the model achieves zero loss.

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 $\boldsymbol{\beta}_{(h_1)} > \boldsymbol{\beta}_{(h_2)} > \cdots > \boldsymbol{\beta}_{(h_k)}$, the heads are activated sequentially in the order $h_1, h_2, \ldots, h_k$; see Figure 6 in Appendix A. 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.

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 of the training trajectory.

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, the first-layer attention parameters $\mathbf{w}^i$ are symmetric across heads (all zero), so no positional signal breaks the symmetry.

Theorem 3.1. *Under Assumptions 2.1, 2.2, 2.4, and 2.5, consider the simplified transformer model $\boldsymbol{\theta} = (\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^k, \boldsymbol{\beta})$ and initialization $\mathcal{I}_1$, evolving under gradient flow. Assumption 2.3 is not needed. Then:*

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

(b) ***Sparse attention:*** *$\dot{\mathbf{w}}_1^1 > 0$ and $\dot{\mathbf{w}}_i^1 < 0$ for all $i \neq 1$, i.e., the head learns to attend to the previous token (relative offset 1).*

(c) ***Closed-form ODE:*** *The dynamics can be fully described by the evolution of $\alpha_1(t), \boldsymbol{\beta}_1(t)$ as*

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

$$\text{where} \quad \Xi = \frac{(1+\gamma)}{(e^{\gamma_1} + n - 1)^2} \left[\frac{e^{\gamma_1}(|\mathcal{R}| - 1)}{(n-2)!|\mathcal{R}|} \right], \qquad \gamma_1 = (1+\gamma)\boldsymbol{\beta}_1^2 \frac{e^{\alpha_1} - 1}{e^{\alpha_1} + n - 1}.$$

(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*

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

Symmetries and directional bias. All relative-position weights of the first head except the previous-token weight $\mathbf{w}_1^1$ evolve together. Indeed, the task is unchanged when the other token positions are permuted, and the head starts with uniform window weights. Those coordinates therefore have equal gradients throughout the first jump. This is the directional bias in Theorem 3.1(a); see the window-weight plot in Figure 4.

Sparsification of attention. The transformer rapidly learns to attend to the previous token (relative offset 1): $\mathbf{w}_1^1$ grows, while the other relative-position weights decrease together. The reason is a covariance between two simple events. For a candidate permutation π, ask whether its token at offset m matches the last query token, and whether its response agrees with the answer $f_\tau(q)$. Regard the second-layer attention $\mathbf{s}$ as a distribution over candidates. At the first jump it is independent of the random response map, and averaging over that map gives

$$\mathbb{E}_{f_\tau} \operatorname{Cov}_{\pi \sim \mathbf{s}}\Big(\mathbb{1}\{q_{n-1} = \pi_{n-m}\}, \mathbb{1}\{f_\tau(q) = f_\tau(\pi)\}\Big)$$
$$= \left(1 - \frac{1}{|\mathcal{R}|}\right) \mathbf{s}(q) \left[\mathbb{1}\{m = 1\} - \mathbb{E}_{\pi \sim \mathbf{s}} \mathbb{1}\{q_{n-1} = \pi_{n-m}\}\right].$$

Indeed, the exact copy $\pi = q$ always has the correct response, whereas every other permutation agrees with it only with probability $1/|\mathcal{R}|$. When $m = 1$, the bracket is positive: the exact copy matches at the previous-token offset. When $m \neq 1$, its symbols are distinct, so the bracket is negative. Applying the first-layer softmax chain rule yields $\dot{\mathbf{w}}_1^1 > 0$ and $\dot{\mathbf{w}}_m^1 < 0$ for $m \neq 1$. The full calculation is in Lemma D.3 and the proof of Theorem 3.1 in Appendix B. The active query-side input is the exact skip connection, so this first-jump argument does not require A2.

Cross-layer coupling and plateau length. Write $\rho(\alpha) = (e^\alpha - 1)/(e^\alpha + n - 1)$. Since the first-layer weights have zero-sum velocity, $\alpha_1 = \mathbf{w}_1^1 - \mathbf{w}_2^1$. The ODE in Theorem 3.1(c) can then be written as

$$\dot{\alpha}_1 = \frac{n}{n-1} \boldsymbol{\beta}_1^2 \rho'(\alpha_1) \Xi, \qquad \dot{\boldsymbol{\beta}}_1 = 2\boldsymbol{\beta}_1 \rho(\alpha_1) \Xi.$$

The common factor Ξ determines the speed, but cancels when the two equations are combined. Indeed, the function f in Theorem 3.1(d) satisfies $f'(\alpha) = \frac{n-1}{n}\rho(\alpha)/\rho'(\alpha)$, and hence

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(f(\alpha_1) - \frac{\boldsymbol{\beta}_1^2}{4}\right) = 0.$$

This exact invariant binds the first-layer sharpness to the second-layer scale across the softmax; compare the single-layer conservation laws of Marcotte et al. [2025].

The invariant also quantifies the initial plateau. With $\boldsymbol{\beta}_1(0) = \epsilon$ and $s = e^{\alpha_1} - 1$, the explicit f in Theorem 3.1(d) yields

$$\boldsymbol{\beta}_1^2 = \epsilon^2 + 4f(\alpha_1) = \epsilon^2 + \frac{2(n-1)}{n}s^2 + O(s^3) \qquad (s \downarrow 0).$$

Thus $\dot{s}$ has the Riccati form $\dot{s} \asymp \epsilon^2 + s^2$ near the initial saddle, whose comparison solutions have tangent profiles. More precisely, for any fixed $a > 0$, the exact ODE gives constants $A_-, A_+ > 0$, independent of ϵ, such that $A_-(\epsilon^2 + \alpha_1^2) \leqslant \dot{\alpha}_1 \leqslant A_+(\epsilon^2 + \alpha_1^2)$ until $\alpha_1 = a$. If $\tau_a = \inf\{t : \alpha_1(t) = a\}$, integration gives

$$\frac{\arctan(a/\epsilon)}{A_+\epsilon} \leqslant \tau_a \leqslant \frac{\arctan(a/\epsilon)}{A_-\epsilon}, \qquad \text{so} \qquad \tau_a = \Theta(1/\epsilon).$$

This is the fixed-threshold meaning of the plateau duration seen in Figure 4; the comparison is proved in Appendix B. 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

$$\mathrm{d}s \sim bse^{-s} \implies \int \frac{e^s}{s}\mathrm{d}s = bT \tag{4}$$

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

The subsequent jumps. Similar to the first jump, we analyze the subsequent jumps. 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 $\mathbf{w}^i = c_1\mathsf{e}_i^n + c_2\mathbf{1}$ for $i \in [1, h-1]$, and $\mathbf{w}^i = 0$ otherwise. Under this initialization, we study the dynamics of the hth jump as the model escapes the plateau where it has learned to match $h-1$ tokens in the context.

Proposition 3.2. *Under Assumptions 2.1–2.5, consider the simplified transformer model $\boldsymbol{\theta} = (\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^k, \boldsymbol{\beta})$ with k heads at the stylized initialization $\mathcal{I}_h$, where $c_1 > 0$ is fixed and $c \to \infty$. Then:*

(a) ***Near-stationarity of the learned heads:*** *At $\mathcal{I}_h$, $\nabla_{\mathbf{w}^i}\mathcal{L}, \nabla_{\boldsymbol{\beta}_i}\mathcal{L} = O(e^{-c})$ for $i < h$.*

(b) ***Initial direction of the next head:*** *Conditional on matching the first $h-1$ offsets, the candidate softmax is the first-jump match-score softmax on $m = n - h + 1$ free symbols. At $\mathcal{I}_h$, the population match-feature covariances converge to its first-jump values as $c \to \infty$. Consequently, the initial velocity of $\mathbf{w}^h$ is positive at offset h, negative at the other free offsets, and vanishes at the fixed offsets in this limit. Its chain-rule factor uses the original n-position window.*

The reduction behind the later jumps is explicit. Once heads $1, \ldots, h-1$ have saturated, their scores confine second-layer attention, up to exponentially small error, to

$$\mathcal{S}_{h-1}(q) = \{\pi : \pi_{n-i} = q_{n-i} \text{ for } 1 \leqslant i < h\}, \qquad |\mathcal{S}_{h-1}(q)| = (n-h+1)!.$$

On this set the learned heads contribute the same score to every candidate, so their terms cancel in the softmax. The remaining $m = n - h + 1$ free symbols therefore give head h the same match-score covariance as a first jump on permutations of m symbols. This proves the initial directional bias while the old heads have $O(e^{-c})$ gradients. The reduction concerns the candidate distribution and its gradient at $\mathcal{I}_h$: head h still attends over the original n offsets, so its full trajectory is not the first-jump ODE with n replaced by m. Appendix B gives the computation.

3.1 A complete picture of the dynamics

We stitch the per-jump analyses into a full trajectory using a simplified (α, β) model with $\mathbf{w}^i = \alpha_i\mathsf{e}_i^n$.

Theorem 3.3. *Under Assumptions 2.1–2.5, consider the (α, β) model on the order-$n = k + 1$ task, with $\boldsymbol{\beta}(0) = \epsilon(c_1, \ldots, c_k)$ and $\alpha_i(0) = 0$. For any fixed thresholds $C_1, C_2 > 0$, there is an $\eta > 0$ such that, if $c_1 > \cdots > c_k > 0$ and $c_{i+1}/c_i \leqslant \eta$, there are times $0 < T_1 < \cdots < T_k$, each $O(1/\epsilon)$, for which $\boldsymbol{\beta}_j(T_i) > C_1$ and $\alpha_j(T_i) > C_2$ for $j \leqslant i$, while $\boldsymbol{\beta}_{j'}(T_i), \alpha_{j'}(T_i) = O(\epsilon)$ for $j' > i$.*

Sketch of proof. Each head has an exact conservation law in the (α, β) model. Before its activation, its sharpness satisfies a Riccati comparison, with escape time proportional to $1/(c_h\epsilon)$. Swapping that head's offset with the single offset having no head keeps its gradient uniformly positive, irrespective of how large earlier heads become. Sufficient separation of the c_h orders the escape times and keeps later heads at scale $O(\epsilon)$; see Appendix E. Directional bias is built into this model, rather than derived from random initialization [Kumar and Haupt, 2024].

3.2 Experiments and conclusion

Here we summarize the global picture: the dynamics predicted by Theorem 3.1 and Proposition 3.2 persist across each relaxation of the simplified setup. Table 2 indexes the corresponding figures.

Experiments. Figure 4 shows training of the simplified $n = 4$ ($k = 3$ heads) model under A0–A4. The loss panel ($\mathcal{L}$) collapses under the time-rescaling t/ϵ at the predicted $O(1/\epsilon)$ rate; the $\boldsymbol{\beta}$–α panel shows the conservation law holding across stages; the window-weights panel (w) shows the stage-wise specialization — $\mathbf{w}_1^1$ grows while the rest decay together (Theorem 3.1), then becomes near-stationary as $\mathbf{w}_1^2$ takes over (Proposition 3.2). The minor decay of $\mathbf{w}_1^2$ after its own jump is a finite-scale effect not captured by gradient flow. See Table 2 for the corresponding ablations.

Table 2: Quick reference of experimental settings and locations.

Setting	Location
Simplified model	Fig. 4, App. F.0.1
Relaxing (A0) and (A3)	Fig. 9, App. F.0.2
Relaxing (A0–A4), structured Q, K init	Fig. 10, App. F.0.3
Relaxing (A1): random Q, K init	Fig. 3
CE loss + incomplete context	Fig. 13 & Fig. 14
Transformer with MLP	Fig. 2, App. F.0.4

Limitations. Our analysis relies on Assumptions 2.1–2.5; we discuss which results extend beyond them. The fixed window in A0 is a convenience for closed-form ODEs — without it we lose the directional bias and explicit formulas. A1 (block structure) is the most substantive assumption: Figures 7 and 10 show the structure is approximately maintained by gradient flow for random and structured initializations, a partial proof is given in Lemma A.3 in the appendix, but a full proof for arbitrary initializations remains open. A2 is unnecessary for the first jump because its query-side input is the exact skip connection. At later jumps, a specialized head approximates the required last-token embedding, but controlling the resulting error along the coupled trajectory remains open. A3 can be avoided by choosing a V that directly outputs the response and normalizing by the sum (equivalent to a softmax over response coordinates). For A4, the conservation law extends exactly to CE by Remark A.2; the CE timescale is supported empirically, while the timescale proof here is for DP (Remark A.1). Beyond the assumptions: our theory uses continuous-time gradient flow while experiments use mini-batch SGD with momentum and show the phenomena persist [Corlouer et al., 2026]; the analysis runs at $k \leqslant 5$ since sequence length is factorial in k; we assume all permutations are present in-context, with partial robustness to incomplete permutations shown in Figure 13.

Conclusion. We have studied the training dynamics of transformers on an in-context associative recall task, identifying a stage-wise learning process in which the model incrementally activates attention circuits. For the simplified model we derive exact first-jump dynamics, a cross-layer conservation law, and a fixed-threshold escape time of order $\Theta(1/\varepsilon)$. The constrained (α, β) model yields a complete k-stage ordering under separated initialization scales. Experiments show analogous stages in full transformers with MLP layers; the conservation principle also extends empirically to the full Q, K parameterization without the block-structure assumption (Figure 7(a)).

Acknowledgments and Disclosure of Funding

This work was partially funded by an unrestricted gift from Coefficient Giving, and the grant number 212111 from the Swiss National Science Foundation. AV was funded by Swiss Data Science Fellowship. AV thanks Gizem Yuce for some helpful discussions and proof reading.

References

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.

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.

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

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

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

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.

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.

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

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`.

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

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*, 2024a. URL `https://openreview.net/forum?id=MO5PiKHELW`.

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*, 2024b.

G. Corlouer, J. Semler, A. Strang, and A. Gietelink Oldenziel. Stochastic gradient descent in the saddle-to-saddle regime of deep linear networks, 2026.

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.

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.

P. Gopalani and Y. Hu. What happens during the loss plateau? understanding abrupt learning in transformers. In *Advances in Neural Information Processing Systems*, 2025.

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.

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

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.

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.

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`.

A. Kumar and J. Haupt. Directional convergence near small initializations and saddles in two-homogeneous neural networks. *Transactions on Machine Learning Research*, 2024.

D. Kunin, G. L. Marchetti, J. Chen, D. Karkada, J. Simon, M. DeWeese, S. Ganguli, and N. Miolane. Alternating gradient flows: A theory of feature learning in two-layer neural networks. In *Advances in Neural Information Processing Systems*, 2025.

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.

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.

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

T. Musat, T. Pimentel, L. Noci, A. Stolfo, M. Sachan, and T. Hofmann. On the emergence of induction heads for in-context learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

T. Muṣat. Mechanism and emergence of stacked attention heads in multi-layer transformers. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=rUC7tHecSQ`.

E. Nichani, A. Damian, and J. D. Lee. How transformers learn causal structure with gradient descent. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

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`.

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.

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

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.

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

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.

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

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

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.

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`.

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.

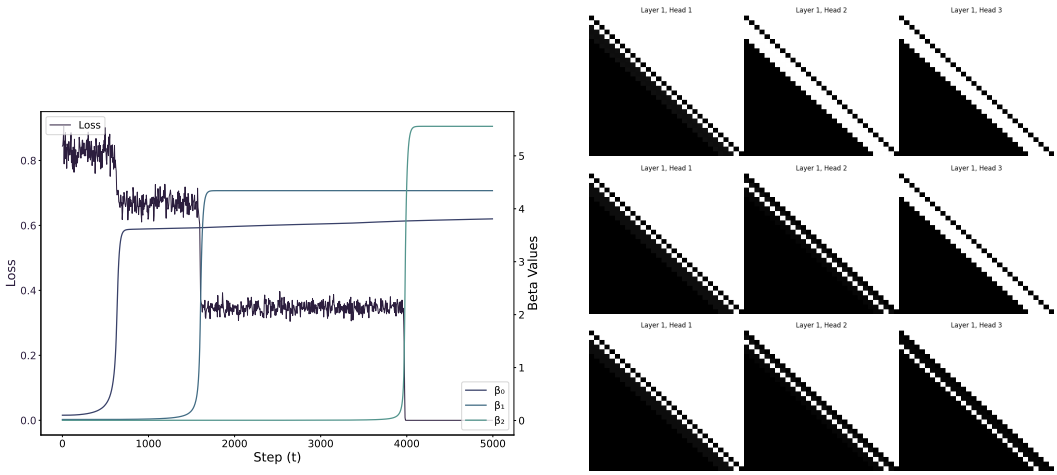
J. Von Oswald, E. Niklasson, E. Randazzo, J. Sacramento, A. Mordvintsev, A. Zhmoginov, and M. Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR, 2023.

O. K. Yüksel, M. Alvarez Lucendo, and N. Flammarion. Incremental learning of sparse attention patterns in transformers, 2026.

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`.

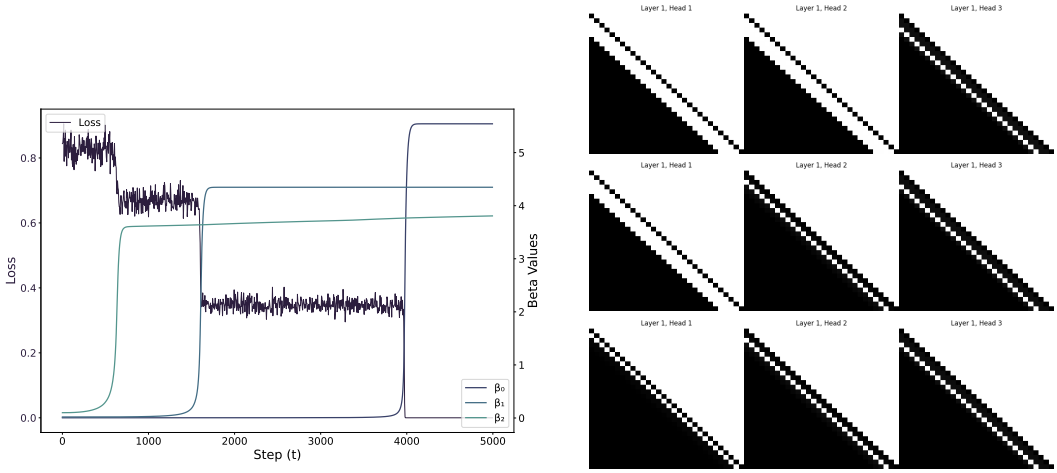
Y. Zhang, A. M. Saxe, and P. E. Latham. Saddle-to-saddle dynamics explains a simplicity bias across neural network architectures. In *The Fourteenth International Conference on Learning Representations*, 2026.

A Supporting Figures



(a) (b)

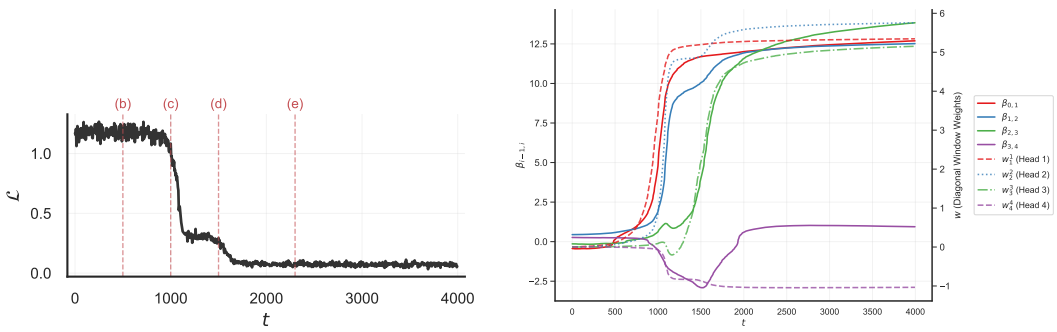
Figure 5: **Simplified model: loss, β values, and attention.** **(a)** Loss and β values over training. **(b)** First-layer attention patterns. Note that the β values rise simultaneously with the loss, and the attention patterns at times 800, 1750, and 4500 demonstrate incremental learning.



(a) (b)

Figure 6: **Simplified model, reversed initialization order.** **(a)** Loss and β values over training. **(b)** First-layer attention patterns. Note that the β values rise as the loss drops, and the attention patterns at times 800, 1750, and 4500 demonstrate incremental learning. Compared to Figure 5, the heads activate in the reverse order, induced by the reversed initialization $\beta_3 > \beta_2 > \beta_1$.

Notation. Throughout the appendix, n is the task order, i.e. the permutation length and the first-layer window size (so $n!$ permutations appear in context), and $k = n - 1$ is the number of first-layer heads: matching k tokens determines the remaining one. Heads are indexed by $h \in [1, k]$ and window offsets by $i \in [1, n]$.



(a) Loss

(b) β-equivalent values and window weights

(c) $Q^\top K, t = 600$

(d) $Q^\top K, t = 1100$

(e) $Q^\top K, t = 1600$

(f) $Q^\top K, t = 2300$

Figure 7: **Relaxing Assumption 2.2: full Q, K with random Gaussian initialization ($n = 5$, $l = 120$). (a)** Loss curve over training; dashed verticals mark the snapshot times in (c)–(f). **(b)** β-equivalent values (computed as $\beta_{h,g}^2 = |\langle B_{h,g}, \widetilde{\mathrm{I}}_k \rangle|$ where $B_{h,g}$ is the (h, g)-block of $Q^\top K$) and attention window weights over training. Stage-wise loss drops, the coupling between β-equivalent values and window weights, and their tied evolution are all consistent with the theoretical predictions, showing that the conservation law and stage-wise dynamics persist without the block-diagonal parameterization of Assumption 2.2. **(c)–(f)** Snapshots of $Q^\top K$ in the second layer. Starting from a random Gaussian initialization ($Q^\top K$ is unstructured at $t = 600$), the sub-diagonal block structure predicted by our theory emerges progressively during training.

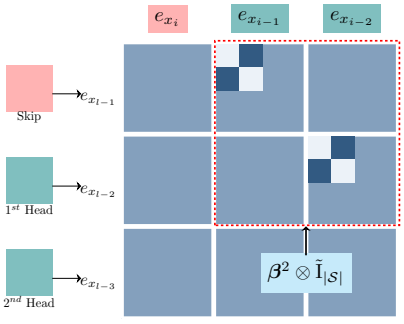


Figure 8: **Layer-2 representation $B^\top$.** The block-diagonal target structure observed empirically. The simplified model parameterizes $\mathbf{Q}^\top \mathbf{K}$ directly on this manifold (Assumption 2.2).

A.1 CE vs. DP loss

Remark A.1 (CE vs. DP timescales)**.** On each saddle plateau, the model output concentrates on a finite set of responses, so $\langle \mathbf{p}, \mathbf{p}_* \rangle$ is approximately constant throughout the plateau. Therefore, CE and DP gradient flows are mutual time-rescalings on each stage, and all timescale results (e.g., the $O(1/\varepsilon)$ plateau duration in Theorem 3.1) carry over from DP to CE training without modification. This is confirmed empirically in Figure 9.

Remark A.2 (Conservation law under CE)**.** The conservation law derived for the DP loss extends to CE training as follows. The law holds for the per-sample loss ℓ, hence in expectation over the data $\mathbb{E}[\ell]$. For CE, the per-sample gradient is scaled by a factor $1/\langle \mathbf{p}, \mathbf{p}_* \rangle$ relative to DP; this scalar rescaling preserves the conservation law samplewise, and therefore also for the population CE loss. Consequently, the timescales of the dynamics remain identical under CE and DP — confirmed empirically in Fig. 4 (CE training) and proved analytically for DP. In particular, the conservation law implies an information exponent of 2 for training with CE, which governs the timescale of each jump.

A.2 Block structure of $\mathbf{Q}^\top\mathbf{K}$ under partial specialization

The next lemma justifies Assumption 2.2 (block-diagonal $\mathbf{Q}^\top\mathbf{K}$) at the first jump: when only the first first-layer head has begun to specialize, the population DP-loss gradient with respect to the full second-layer matrix $B = \mathbf{Q}^\top\mathbf{K}$ is concentrated on the first super-diagonal block, matching the shape $\mathrm{diag}_1(1, 0, \ldots, 0) \otimes \widetilde{\mathrm{I}}_n$ that A1 parameterizes.

Lemma A.3 (Block gradient at the first jump)**.** *Consider the disentangled attention-only model with full second-layer parameterization $B = \mathbf{Q}^\top\mathbf{K} \in \mathbb{R}^{(k+1)|\mathcal{S}|\times(k+1)|\mathcal{S}|}$ and per-block decomposition $B_{h,g}$, $h, g \in \{0, 1, \ldots, k\}$. Suppose the first first-layer head has begun to specialize while the others remain at the symmetric initialization, i.e.,*

$$\mathbf{R}_t^1 = c_1\, e_{x_{t-1}} + c_2\, \mu, \qquad \mathbf{R}_t^h = c_3\, \mu \quad \textit{for } h > 1,$$

where μ is the mean token embedding and c_1, c_2, c_3 are scalars reflecting the current degree of specialization. Then the population DP-loss gradient with respect to B is approximately

$$\nabla_B \mathcal{L} \;\approx\; c_1^2 \left(\mathrm{diag}_1(1, 0, \ldots, 0) \otimes \widetilde{\mathrm{I}}_n\right),$$

i.e., concentrated on the first super-diagonal block $B_{0,1}$ with the $\widetilde{\mathrm{I}}_n$ shape parameterized by Assumption 2.2.

Proof. Recall ℓ'_t — the gradient of the loss with respect to the pre-softmax score $\tilde{\mathbf{p}}_t$ at token t — defined later in the proof of Theorem 3.1. The key property used below is $\sum_t \ell'_t = 0$, which follows from shift-invariance of the softmax.

Block structure of the gradient. The pre-activation at token t in the second layer is $\tilde{\mathbf{p}}_t = \mathbf{R}_l^\top B\, \mathbf{R}_t$ where $\mathbf{R}_t = [(\mathbf{R}_t^0)^\top, (\mathbf{R}_t^1)^\top, \ldots, (\mathbf{R}_t^n)^\top]^\top$ stacks the head outputs (with $\mathbf{R}_t^0 = e_{x_t}$ from the skip connection). Hence

$$\frac{\partial \ell}{\partial B} = \sum_t \ell'_t\, \mathbf{R}_l\, \mathbf{R}_t^\top, \qquad \frac{\partial \ell}{\partial B_{h,g}} = \sum_t \ell'_t\, \mathbf{R}_l^h\, (\mathbf{R}_t^g)^\top.$$

Interactions between heads h and g in the first layer thus directly shape how $B_{h,g}$ is updated.

Substituting the single-head specialization. Under the lemma's assumption, $\mathbf{R}_l^1 = c_1\, e_{x_{l-1}} + c_2\, \mu$, $\mathbf{R}_l^h = c_3\, \mu$ for $h > 1$, and $\mathbf{R}_l^0 = e_{x_l}$ (skip connection). We examine each block.

(i) $h \geqslant 2$**.** The factor $\mathbf{R}_l^h = c_3\, \mu$ is constant in t, so

$$\frac{\partial \ell}{\partial B_{h,g}} = c_3\, \mu \left(\sum_t \ell'_t\, \mathbf{R}_t^g\right)^\top.$$

The mean-only part of $\mathbf{R}_t^g$ is constant in t and vanishes by $\sum_t \ell'_t = 0$; the only token-dependent piece is $c_1\, e_{x_{t-1}}$ when $g = 1$, scaled by the small factor c_3 — sub-leading in the partial-specialization regime (c_3 small). All other blocks with $h \geqslant 2$ vanish to leading order.

(ii) $h \in \{0, 1\}$**,** $g \neq 1$**.** If $g \geqslant 2$, then $\mathbf{R}_t^g = c_3\, \mu$ is constant in t and $\sum_t \ell'_t = 0$ kills the leading contribution. If $g = 0$, both $\mathbf{R}_l^h$ and $\mathbf{R}_t^0 = e_{x_t}$ are token-dependent, but the resulting outer product

$\mathbf{R}_l^h (e_{x_t})^\top$ has the wrong index alignment: the symmetry argument behind Theorem 3.1(b) shows the relevant covariance vanishes (the predictively-relevant token in column position is at $t-1$, not t).

(iii) $h=0, g=1$**.** Both factors are token-dependent: $\mathbf{R}_l^0 = e_{x_l}$ and $\mathbf{R}_t^1 = c_1 e_{x_{t-1}} + c_2 \mu$. Expanding,

$$\dot{B}_{0,1} = \sum_t \ell'_t \, e_{x_l} \big(c_1 \, e_{x_{t-1}} + c_2 \, \mu\big)^\top \;=\; c_1 \sum_t \ell'_t \, e_{x_l} \, (e_{x_{t-1}})^\top + c_2 \Big(\sum_t \ell'_t\Big) e_{x_l} \, \mu^\top .$$

The second term vanishes by $\sum_t \ell'_t = 0$. The leading c_1 term gives

$$\dot{B}_{0,1} \;\propto\; c_1 \, \mathbb{E}_{f_\tau, q}\Big[\sum_t \ell'_t \, e_{x_l} \, (e_{x_{t-1}})^\top\Big] \;\propto\; \mathrm{Cov}_{\pi_t \sim \mathbf{s}}\big(\mathbb{1}\{q_{n-1} = \pi_{t,n-1}\},\; \mathbb{1}\{f(q) = f(\pi_t)\}\big) \cdot \widetilde{\mathrm{I}}_n,$$

following the same covariance argument as Theorem 3.1(b): when $\pi_t = q$ both indicators are 1; otherwise they are independent, giving a positive correlation. Permutation symmetry of the task collapses the resulting matrix to a positive scalar multiple of $\widetilde{\mathrm{I}}_n$.

Conclusion. At the first jump the only block with non-vanishing leading-order gradient is $B_{0,1}$, with shape $\propto \widetilde{\mathrm{I}}_n$. Therefore

$$\nabla_B \mathcal{L} \;\approx\; c_1^2 \, \mathrm{diag}_1(1, 0, \ldots, 0) \otimes \widetilde{\mathrm{I}}_n,$$

which matches the leading entry of the manifold parameterized by Assumption 2.2. As further heads specialize at later stages, an analogous co-evolution argument yields non-zero $B_{h,h+1}$ blocks for $h \geqslant 1$, populating the full super-diagonal predicted by A1. □

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 moves along the manifold $\mathcal{M}_1 = \{\boldsymbol{\theta} : \mathbf{w}^1 = \alpha_1 \mathrm{e}_1^n + \delta_1 \mathbf{1},\ \alpha_1, \delta_1 \in \mathbb{R} \text{ and } \boldsymbol{\beta}_h, \mathbf{w}^h = 0 \text{ for } h \neq 1\}$.

Consider any point on the manifold $\mathcal{M}_1$ say $\boldsymbol{\theta} = (\mathbf{w}^1, \mathbf{w}^2, \ldots \mathbf{w}^k, \boldsymbol{\beta})$ where $\mathbf{w}^1 = \alpha_1 \mathrm{e}_1^n + \delta_1 \mathbf{1}$ 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

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

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

where

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

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

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

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 $\boldsymbol{\beta}_h^2 = 0$ 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 $\mathbf{s}_t^{'} = \gamma_1 \mathbb{1}\{q_{n-1} = \pi_{(t,n-1)}\}$ as $\gamma_h = 0$ for $h \neq 1$ and the softmax score is given by

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

From Lemma C.3 we have

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

So the gradient of $\mathbf{w}_i^1$ given by $\frac{\partial \ell}{\partial \mathbf{w}_i^1}$ 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

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

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

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

The dynamics of gradient descent is given by

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

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

$$\dot{\alpha}_1 = \dot{\mathbf{w}}_1^1 - \dot{\mathbf{w}}_2^1 = (1+\gamma)\frac{\boldsymbol{\beta}_1^2}{e^{\alpha_1}+n-1}\left(\frac{(n)e^{\alpha_1}}{e^{\alpha_1}+n-1}\mathbb{E}\Delta_{1,1} + \frac{\mathbb{E}\Delta_{1,1}}{n-1} + \frac{e^{\alpha_1}-1}{e^{\alpha_1}+n-1}\mathbb{E}\Delta_{1,1}\right) < 0,$$
$$\dot{\boldsymbol{\beta}}_1 = -2\boldsymbol{\beta}_1\frac{e^{\alpha_1}-1}{e^{\alpha_1}+n-1}(1+\gamma)\mathbb{E}\Delta_{1,1} > 0$$

The final system of ODE is given by

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

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

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

Using this the system of ODE can be written as

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

Note that

$$\frac{\dot{\boldsymbol{\beta}_1^2}}{4} = \boldsymbol{\beta}_1^2\frac{e^{\alpha_1}-1}{e^{\alpha_1}+n-1}\Xi$$

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

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

$$\begin{aligned}
f^{'}(\alpha_1)\frac{\boldsymbol{\beta}_1^2 e^{\alpha_1}}{(e^{\alpha_1}+n-1)^2}\frac{n^2}{n-1}\Xi &= \boldsymbol{\beta}_1^2\frac{e^{\alpha_1}-1}{e^{\alpha_1}+n-1}\Xi, \\
\implies f^{'}(\alpha_1) &= \frac{n-1}{n^2}\frac{(e^{\alpha_1}-1)(e^{\alpha_1}-1+n)}{e^{\alpha_1}} \\
&= \frac{n-1}{n^2}\left(e^{\alpha_1}-2+e^{-\alpha_1}-ne^{-\alpha_1}+n\right) \\
\implies f(\alpha_1) &= \frac{n-1}{n^2}\left(e^{\alpha_1}-2\alpha_1-e^{-\alpha_1}+ne^{-\alpha_1}+n\alpha_1\right), \\
f(\alpha_1) &= 2\frac{n-1}{n^2}\left(\sinh(\alpha_1)-\alpha_1\right)+\frac{n-1}{n}\left[e^{-\alpha_1}+\alpha_1-1\right]
\end{aligned}$$

Now $f(\alpha_1) - \boldsymbol{\beta}_1^2/4$ is conserved along the trajectory. As $\mathrm{d}(f(\alpha_1) - \boldsymbol{\beta}_1^2/4) = 0$.

This completes the proof of the theorem.

B.2 Proof of Theorem 3.2

The population gradients are given by

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

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

where

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

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

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

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 the others do not. Now $\mathbf{s}_t = e^{-c}$ for $t \in \mathcal{P}_S^c$. Now for $\Delta_{i,j}$ with $i \in [1, h-1]$ and $j \neq i$, it can be seen that $\mathbf{s}_t \left(\mathbb{1}\{q_{n-i} = \pi_{(t,n-j)}\}\right) = e^{-c}$ 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 $t \in \mathcal{P}_S$

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

This expression is exactly equivalent to the first jump, however now restricted to the set $\mathcal{P}_S$ 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.

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

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

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

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

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

For the convenience of the analysis, we will drop $\mathcal{R}$ in the subscript of $\widetilde{\mathbf{s}}$ and now $\widetilde{\mathbf{s}} \in \mathbb{R}^{n!}$ where $\widetilde{\mathbf{s}}_t$ denotes the score of the response corresponding to the tth 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:

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

The coordinate-wise derivative is

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

where r_t is the response corresponding to the tth 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 every coordinate is shifted by a constant. Now taking the derivative with c at $c = 0$ we get

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

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

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

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

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

Using Lemma D.2 we have

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

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

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

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

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

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*

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

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

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 $\mathcal{I}_{\mathcal{G}}$ defined such that for all h and i,*

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

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

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

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

where

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

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

Proof. First let us compute the forward pass

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

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

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

Define

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

Denote

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

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 $\mathbf{s} = \boldsymbol{\sigma}(\mathbf{s}^{'})$. Using the lemma C.1 we have

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

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

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

Using this notation, the gradients can be written as

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

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

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

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

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

□

Lemma C.3. *For the parameter configuration $\boldsymbol{\theta} = (\mathbf{w}^1, \mathbf{w}^2, \ldots \mathbf{w}^k, \boldsymbol{\beta})$ where $\mathbf{w}^1 = \alpha_1 \mathsf{e}_1^n + \delta_1 \mathbf{1}$ and $\boldsymbol{\beta}_h = 0$ for $h \neq 1$. The following quantities are*

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

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

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

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

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

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

For $i = 1$, we have,

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

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

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

For $i \neq 1$,

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

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

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

Now for $i = 1$,

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

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

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

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

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

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

Substituting this back in the expectation term, we get,

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

Recall the term for $i \neq 1$, we have,

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

Note that

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

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

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

Now for $i = 1$, we have,

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

□

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,

$$\mathbf{A}^h = \begin{bmatrix} \mathbf{w}^h_0 & 0 & 0 & \dots & 0 \\ \mathbf{w}^h_1 & \mathbf{w}^h_0 & 0 & \dots & 0 \\ \mathbf{w}^h_2 & \mathbf{w}^h_1 & \mathbf{w}^h_0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{w}^h_{l-2} & \mathbf{w}^h_{l-3} & \dots & \mathbf{w}^h_0 & 0 \\ \mathbf{w}^h_{l-1} & \mathbf{w}^h_{l-2} & \mathbf{w}^h_{l-3} & \dots & \mathbf{w}^h_0 \end{bmatrix}$$

Now,

$$\boldsymbol{\sigma}(\mathbf{A}^h)_{t,i} = \frac{e^{\mathbf{w}^h_{t-i}}}{\sum\limits_{j=0}^{t} e^{\mathbf{w}^h_j}},$$

$$\mathbf{R}^h_t = \sum_{i=0}^{t} \frac{e^{\mathbf{w}^h_{t-i}}}{\sum_{j=0}^{t} e^{\mathbf{w}^h_j}}\, \mathsf{e}^{\mathcal{S}}_{x_i} = \sum_{i=0}^{t} \frac{e^{\mathbf{w}^h_i}}{\sum_{j=0}^{t} e^{\mathbf{w}^h_j}}\, \mathsf{e}^{\mathcal{S}}_{x_{t-i}},$$

$$\text{For } i \leqslant t, \quad \frac{\partial \mathbf{R}^h_t}{\partial \mathbf{w}^h_i} = \frac{e^{\mathbf{w}^h_i}}{\sum_{j=0}^{t} e^{\mathbf{w}^h_j}}\, \mathsf{e}^{\mathcal{S}}_{x_{t-i}} - \frac{e^{\mathbf{w}^h_i}}{\left(\sum_{j=0}^{t} e^{\mathbf{w}^h_j}\right)^2} \sum_{j=1}^{t} e^{\mathbf{w}^h_j}\, \mathsf{e}^{\mathcal{S}}_{x_{t-j}},$$

$$= \frac{e^{\mathbf{w}^h_i}}{\sum_{j=0}^{t} e^{\mathbf{w}^h_j}} \left(\mathsf{e}^{\mathcal{S}}_{x_{t-i}} - \mathbf{R}^h_t\right)$$

So, we have the result as follows,

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

□

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,*

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

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

$$\frac{\partial \mathbf{R}^h_t}{\partial \mathbf{w}^h_i} = \frac{e^{\mathbf{w}^h_i}}{\sum\limits_{j=1}^{n} e^{\mathbf{w}^h_j}} \left(e_{\pi_{(t,n-i)}} - \mathbf{R}^h_t\right)$$

Proof.

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

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

$$\mathbf{R}^h_t = \sum_{i=1}^{n} \frac{e^{\mathbf{w}^h_i}}{\sum\limits_{j=1}^{n} e^{\mathbf{w}^h_j}}\, e_{\pi_{(t,n-i)}},$$

$$\text{For } 1 \leqslant i \leqslant n, \quad \frac{\partial \mathbf{R}^h_t}{\partial \mathbf{w}^h_i} = \frac{e^{\mathbf{w}^h_i}}{\sum\limits_{j=1}^{n} e^{\mathbf{w}^h_j}} \left(e_{\pi_{(t,n-i)}} - \mathbf{R}^h_t\right)$$

□

Covariances with respect to parameters.

Lemma D.3 (A positive covariance)**.** *Let $s \in \mathbb{R}^{n!}$ be the softmax scores over a permutation group, f_τ represents the mappings of the permutation and q is the query.*

Consider the covariance between the indicator that the hth element of the permutation is equal to the hth element of the query (q) and the indicator function that the responses of the permutation is equal to response of the query $f_\tau(q)$, given by

$$\operatorname{Cov}\left(\mathbb{1}\{q_{n-h} = \pi_{(t,n-h)}\}, \mathbb{1}\{r_t = r\}\right) = \left[\sum_t \mathbf{s}_t \mathbb{1}\{r_t = r, q_{n-h} = \pi_{(t,n-h)}\} - \left(\sum_{t'} \mathbf{s}_{t'} \mathbb{1}\{r_{t'} = r\}\right)\left(\sum_t \mathbf{s}_t \mathbb{1}\{q_{n-h} = \pi_{(t,n-h)}\}\right)\right].$$

then the expectation of the covariance over the population of tasks and queries is given by

$$\mathop{\mathbb{E}}_{f_\tau, q} \operatorname{Cov}\left(\mathbb{1}\{q_{n-h} = \pi_{(t,n-h)}\}, \mathbb{1}\{r_t = r\}\right) = \mathop{\mathbb{E}}_{q}\left[\left(1 - \frac{1}{|\mathcal{R}|}\right) \mathop{\mathbb{E}}_{\pi \sim s(q)} \mathbb{1}\{\pi = q\}\left(1 - \mathop{\mathbb{E}}_{\pi \sim s(q)} \mathbb{1}\{q_{n-h} = \pi_{(t,n-h)}\}\right)\right].$$

Proof. First, note that the covariance can be succinctly written as follows,

$$\operatorname{Cov}\left(\mathbb{1}\{q_{n-h} = \pi_{(t,n-h)}\}, \mathbb{1}\{r_t = r\}\right) = \mathbb{E}_s \mathbb{1}\{r_t = r, q_{n-h} = \pi_{(t,n-h)}\} - \mathbb{E}_s \mathbb{1}\{r_t = r\}\, \mathbb{E}_s \mathbb{1}\{q_{n-h} = \pi_{(t,n-h)}\}.$$

Now taking the expectation over the population of tasks and queries, we have,

$$\begin{aligned}
\mathcal{C} &= \mathop{\mathbb{E}}_{f_\tau, q}\left[\mathop{\mathbb{E}}_{s} \mathbb{1}\{r_t = r, q_{n-h} = \pi_{(t,n-h)}\} - \mathbb{E}_s \mathbb{1}\{r_t = r\}\, \mathbb{E}_s \mathbb{1}\{q_{n-h} = \pi_{(t,n-h)}\}\right], \\
&= \mathop{\mathbb{E}}_{f_\tau, q}\left[\mathop{\mathbb{E}}_{\pi \sim s(q)} \mathbb{1}\{f(\pi) = f(q)\}\mathbb{1}\{q_{n-h} = \pi_{(t,n-h)}\} - \mathop{\mathbb{E}}_{\pi \sim s(q)} \mathbb{1}\{f(\pi) = f(q)\} \mathop{\mathbb{E}}_{\pi \sim s(q)} \mathbb{1}\{q_{n-h} = \pi_{(t,n-h)}\}\right],
\end{aligned}$$

Pushing the f inside the expectation over f_τ gives

$$\mathop{\mathbb{E}}_{q}\left[\mathop{\mathbb{E}}_{\pi \sim s(q)} \mathop{\mathbb{E}}_{f_\tau} \mathbb{1}\{f(\pi) = f(q)\}\mathbb{1}\{q_{n-h} = \pi_{(t,n-h)}\} - \mathop{\mathbb{E}}_{\pi \sim s(q)} \mathop{\mathbb{E}}_{f_\tau} \mathbb{1}\{f(\pi) = f(q)\} \mathop{\mathbb{E}}_{\pi \sim s(q)} \mathbb{1}\{q_{n-h} = \pi_{(t,n-h)}\}\right].$$

Now, note that $\mathop{\mathbb{E}}_{f_\tau} \mathbb{1}\{f(\pi) = f(q)\} = \frac{1}{|\mathcal{R}|}$ for any $\pi \neq q$ and 1 when $\pi = q$. Therefore, we have,

$$\begin{aligned}
\mathop{\mathbb{E}}_{\pi \sim s(q)} \mathop{\mathbb{E}}_{f_\tau} \mathbb{1}\{f(\pi) = f(q)\}\mathbb{1}\{q_{n-h} = \pi_{(t,n-h)}\} &= \mathop{\mathbb{E}}_{\pi \sim s(q)}\left(\mathbb{1}\{\pi = q\} + \frac{1}{|\mathcal{R}|}\mathbb{1}\{\pi \neq q\}\right)\mathbb{1}\{q_{n-h} = \pi_{(t,n-h)}\}, \\
&= \mathop{\mathbb{E}}_{\pi \sim s(q)}\left[\left(1 - \frac{1}{|\mathcal{R}|}\right)\mathbb{1}\{\pi = q\}\mathbb{1}\{q_{n-h} = \pi_{(t,n-h)}\} + \frac{1}{|\mathcal{R}|}\mathbb{1}\{q_{n-h} = \pi_{(t,n-h)}\right. \\
&= \mathop{\mathbb{E}}_{\pi \sim s(q)}\left[\left(1 - \frac{1}{|\mathcal{R}|}\right)\mathbb{1}\{\pi = q\} + \frac{1}{|\mathcal{R}|}\mathbb{1}\{q_{n-h} = \pi_{(t,n-h)}\}\right], \\
&= \left[\left(1 - \frac{1}{|\mathcal{R}|}\right)\mathop{\mathbb{E}}_{\pi \sim s(q)}\mathbb{1}\{\pi = q\} + \frac{1}{|\mathcal{R}|}\mathop{\mathbb{E}}_{\pi \sim s(q)}\mathbb{1}\{q_{n-h} = \pi_{(t,n-h)}\}\right].
\end{aligned}$$

$$\mathop{\mathbb{E}}_{\pi \sim s(q)} \mathop{\mathbb{E}}_{f_\tau} \mathbb{1}\{f(\pi) = f(q)\} \mathop{\mathbb{E}}_{\pi \sim s(q)} \mathbb{1}\{q_{n-h} = \pi_{(t,n-h)}\}, = \left[\left(1 - \frac{1}{|\mathcal{R}|}\right)\mathop{\mathbb{E}}_{\pi \sim s(q)}\mathbb{1}\{\pi = q\} + \frac{1}{|\mathcal{R}|}\right]\mathop{\mathbb{E}}_{\pi \sim s(q)}\mathbb{1}\{q_{n-h} = \pi_{(t,n-h)}\}.$$

Subtracting them, we get,

$$\begin{aligned}
\mathcal{C} &= \mathop{\mathbb{E}}_{\pi \sim s(q)} \mathop{\mathbb{E}}_{f_\tau} \mathbb{1}\{f(\pi) = f(q)\}\mathbb{1}\{q_{n-h} = \pi_{(t,n-h)}\} - \mathop{\mathbb{E}}_{\pi \sim s(q)} \mathop{\mathbb{E}}_{f_\tau} \mathbb{1}\{f(\pi) = f(q)\} \mathop{\mathbb{E}}_{\pi \sim s(q)} \mathbb{1}\{q_{n-h} = \pi_{(t,n-h)}\}, \\
&= \left(1 - \frac{1}{|\mathcal{R}|}\right)\mathop{\mathbb{E}}_{\pi \sim s(q)}\mathbb{1}\{\pi = q\}\left(1 - \mathop{\mathbb{E}}_{\pi \sim s(q)}\mathbb{1}\{q_{n-h} = \pi_{(t,n-h)}\}\right).
\end{aligned}$$

Note taking the expectation over q, we have,

$$\mathcal{C} = \mathop{\mathbb{E}}_{q}\left[\left(1 - \frac{1}{|\mathcal{R}|}\right)\mathop{\mathbb{E}}_{\pi \sim s(q)}\mathbb{1}\{\pi = q\}\left(1 - \mathop{\mathbb{E}}_{\pi \sim s(q)}\mathbb{1}\{q_{n-h} = \pi_{(t,n-h)}\}\right)\right].$$

□

In the same lines as the above lemma, we can also show the following lemma for a negative covariance.

Lemma D.4. *Let $s \in \mathbb{R}^{n!}$ be the softmax scores over a permutation group, f_τ represents the mappings of the permutation and q is the query. Consider the covariance between the indicator that the ith element of the permutation is equal to the hth $(h \neq i)$ element of the query (q) and the indicator function that the responses of the permutation is equal to response of the query $f_\tau(q)$,*

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

then the expectation of the covariance over the population of tasks and queries is given by

$$\mathop{\mathbb{E}}_{f_\tau,q} \operatorname{Cov}\left(\mathbb{1}\{q_{n-h} = \pi_{(t,n-i)}\}, \mathbb{1}\{r_t = r\}\right) = -\mathop{\mathbb{E}}_{q}\left[\left(1 - \frac{1}{|\mathcal{R}|}\right) \mathop{\mathbb{E}}_{\pi\sim s(q)} \mathbb{1}\{\pi = q\} \left(\mathop{\mathbb{E}}_{\pi\sim s(q)} \mathbb{1}\{q_{n-h} = \pi_{(t,n-i)}\}\right)\right].$$

Proof. Rewriting the covariance as,

$$\operatorname{Cov}\left(\mathbb{1}\{q_{n-h} = \pi_{(t,n-i)}\}, \mathbb{1}\{r_t = r\}\right) = \mathbb{E}_s \mathbb{1}\{r_t = r, q_{n-h} = \pi_{(t,n-i)}\} - \mathbb{E}_s \mathbb{1}\{r_t = r\}\, \mathbb{E}_s \mathbb{1}\{q_{n-h} = \pi_{(t,n-i)}\}.$$

Now taking the expectation over the population of tasks and queries, we have,

$$\begin{aligned}
\mathcal{C} &= \mathop{\mathbb{E}}_{f_\tau,q}\left[\mathop{\mathbb{E}}_{s} \mathbb{1}\{r_t = r, q_{n-h} = \pi_{(t,n-i)}\} - \mathbb{E}_s \mathbb{1}\{r_t = r\}\, \mathbb{E}_s \mathbb{1}\{q_{n-h} = \pi_{(t,n-i)}\}\right], \\
&= \mathop{\mathbb{E}}_{f_\tau,q}\left[\mathop{\mathbb{E}}_{\pi\sim s(q)} \mathbb{1}\{f(\pi) = f(q)\}\mathbb{1}\{q_{n-h} = \pi_{(t,n-i)}\} - \mathop{\mathbb{E}}_{\pi\sim s(q)} \mathbb{1}\{f(\pi) = f(q)\} \mathop{\mathbb{E}}_{\pi\sim s(q)} \mathbb{1}\{q_{n-h} = \pi_{(t,n-i)}\}\right],
\end{aligned}$$

We are going to push the expectation over f_τ inside,

$$\mathop{\mathbb{E}}_{q}\left[\mathop{\mathbb{E}}_{\pi\sim s(q)} \mathop{\mathbb{E}}_{f_\tau} \mathbb{1}\{f(\pi) = f(q)\}\mathbb{1}\{q_{n-h} = \pi_{(t,n-i)}\} - \mathop{\mathbb{E}}_{\pi\sim s(q)} \mathop{\mathbb{E}}_{f_\tau} \mathbb{1}\{f(\pi) = f(q)\} \mathop{\mathbb{E}}_{\pi\sim s(q)} \mathbb{1}\{q_{n-h} = \pi_{(t,n-i)}\}\right],$$

Now, note that $\mathop{\mathbb{E}}_{f_\tau} \mathbb{1}\{f(\pi) = f(q)\} = \frac{1}{|\mathcal{R}|}$ for any $\pi \neq q$ and 1 when $\pi = q$. Therefore, we have,

$$\begin{aligned}
\mathop{\mathbb{E}}_{\pi\sim s(q)} \mathop{\mathbb{E}}_{f_\tau} \mathbb{1}\{f(\pi) = f(q)\}\mathbb{1}\{q_{n-h} = \pi_{(t,n-i)}\} &= \mathop{\mathbb{E}}_{\pi\sim s(q)}\left(\mathbb{1}\{\pi = q\} + \frac{1}{|\mathcal{R}|}\mathbb{1}\{\pi \neq q\}\right)\mathbb{1}\{q_{n-h} = \pi_{(t,n-i)}\}, \\
&= \mathop{\mathbb{E}}_{\pi\sim s(q)}\left[\left(1 - \frac{1}{|\mathcal{R}|}\right)\mathbb{1}\{\pi = q\}\mathbb{1}\{q_{n-h} = \pi_{(t,n-i)}\} + \frac{1}{|\mathcal{R}|}\mathbb{1}\{q_{n-h} = \pi_{(t,n-i)}\}\right. \\
&= \mathop{\mathbb{E}}_{\pi\sim s(q)}\left[\frac{1}{|\mathcal{R}|}\mathbb{1}\{q_{n-h} = \pi_{(t,n-i)}\}\right].
\end{aligned}$$

Since $i \neq h$, the product of two indicators is always zero and this was used to simplify the last step.

$$\mathop{\mathbb{E}}_{\pi\sim s(q)} \mathop{\mathbb{E}}_{f_\tau} \mathbb{1}\{f(\pi) = f(q)\} \mathop{\mathbb{E}}_{\pi\sim s(q)} \mathbb{1}\{q_{n-h} = \pi_{(t,n-i)}\} = \left[\left(1 - \frac{1}{|\mathcal{R}|}\right) \mathop{\mathbb{E}}_{\pi\sim s(q)} \mathbb{1}\{\pi = q\} + \frac{1}{|\mathcal{R}|}\right] \mathop{\mathbb{E}}_{\pi\sim s(q)} \mathbb{1}\{q_{n-h} = \pi_{(t,n-i)}\},$$

Subtracting them, we get,

$$\begin{aligned}
\mathcal{C} &= \mathop{\mathbb{E}}_{\pi\sim s(q)} \mathop{\mathbb{E}}_{f_\tau} \mathbb{1}\{f(\pi) = f(q)\}\mathbb{1}\{q_{n-h} = \pi_{(t,n-i)}\} - \mathop{\mathbb{E}}_{\pi\sim s(q)} \mathop{\mathbb{E}}_{f_\tau} \mathbb{1}\{f(\pi) = f(q)\} \mathop{\mathbb{E}}_{\pi\sim s(q)} \mathbb{1}\{q_{n-h} = \pi_{(t,n-i)}\} \\
&= -\left(1 - \frac{1}{|\mathcal{R}|}\right) \mathop{\mathbb{E}}_{\pi\sim s(q)} \mathbb{1}\{\pi = q\} \left(\mathop{\mathbb{E}}_{\pi\sim s(q)} \mathbb{1}\{q_{n-h} = \pi_{(t,n-i)}\}\right).
\end{aligned}$$

Note taking the expectation over q, we have,

$$\mathcal{C} = -\mathop{\mathbb{E}}_{q}\left[\left(1 - \frac{1}{|\mathcal{R}|}\right) \mathop{\mathbb{E}}_{\pi\sim s(q)} \mathbb{1}\{\pi = q\} \left(\mathop{\mathbb{E}}_{\pi\sim s(q)} \mathbb{1}\{q_{n-h} = \pi_{(t,n-i)}\}\right)\right].$$

□

E A simple α, β system

In this section, we consider a simple parameterization from Lemma C.2. Consider a parameter configuration $\mathcal{I}_{\mathcal{G}}$ defined such that, for all h and i,

$$\mathbf{w}^h = \alpha_h \mathsf{e}_h^n$$

and $\boldsymbol{\beta}_h$ is used as is.

Forward Pass of the Model. Using this, the presoftmax scores of the token at position t can be written as

$$\widetilde{\mathbf{s}}_t = \sum_{h=1}^{k} \gamma_h \mathbb{1}\{q_{n-h} = \pi_{(t,n-h)}\}, \text{ with } \gamma_h = (1+\gamma)\boldsymbol{\beta}_h^2 \frac{e^{\alpha_h}-1}{e^{\alpha_h}+n-1}.$$

The attention scores

$$\mathbf{s}_t = \boldsymbol{\sigma}(\widetilde{\mathbf{s}})_t.$$

Now the output probabilities can be written as

$$\mathbf{p} = \sum_t \mathbf{s}_t e_{r_t}.$$

The loss function on a single example

$$\ell_\tau(\alpha, \boldsymbol{\beta}) = 1 - \langle \mathbf{p}, e_r \rangle = 1 - \sum_t \mathbf{s}_t \mathbb{1}\{r_t = r\},$$

Let u be the vector such that $u_t = \mathbb{1}\{r_t = r\}$. Then, the loss can be written as

$$\ell_\tau(\alpha, \boldsymbol{\beta}) = 1 - \langle \mathbf{s}, u \rangle$$

Gradient Computations. Let us take the gradient wrt to the $\widetilde{\mathbf{s}}$,

$$\begin{aligned}
\nabla_{\widetilde{\mathbf{s}}} \ell_\tau(\alpha, \boldsymbol{\beta}) &= -\left(u - \langle \mathbf{s}, u \rangle \mathbf{1}\right) \odot \mathbf{s}, \\
\nabla_{\widetilde{\mathbf{s}}_t} \ell_\tau(\alpha, \boldsymbol{\beta})_t &= -\left[\mathbf{s}_t u_t - \mathbf{s}_t \langle \mathbf{s}, u \rangle\right].
\end{aligned}$$

Now let us take a gradient wrt to γ_h,

$$\begin{aligned}
\langle \nabla_{\widetilde{\mathbf{s}}} \ell_\tau, \nabla_{\gamma_h} \widetilde{\mathbf{s}} \rangle &= \sum_t \nabla_{\widetilde{\mathbf{s}}_t} \ell_\tau \cdot \mathbb{1}\{q_{n-h} = \pi_{(t,n-h)}\} = \sum_t \nabla_{\widetilde{\mathbf{s}}_t} \ell_\tau \cdot \mathbb{1}\{q_{n-h} = \pi_{(n-h)}\}, \\
&= -\sum_t \left[\mathbf{s}_t \mathbb{1}\{r_t = r\} - \mathbf{s}_t \sum_{t'} \mathbf{s}_{t'} \mathbb{1}\{r_{t'} = r\} \right] \cdot \mathbb{1}\{q_{n-h} = \pi_{(n-h)}\}, \\
&= -\left[\sum_t \mathbf{s}_t \mathbb{1}\{r_t = r, q_{n-h} = \pi_{(t,n-h)}\} - \left(\sum_{t'} \mathbf{s}_{t'} \mathbb{1}\{r_{t'} = r\} \right) \left(\sum_t \mathbf{s}_t \mathbb{1}\{q_{n-h} = \pi_{(n-h)}\} \right) \right].
\end{aligned}$$

Consolidating the gradient of the loss, we get,

$$-\nabla_{\gamma_h} \ell = \left[\sum_t \mathbf{s}_t \mathbb{1}\{r_t = r, q_{n-h} = \pi_{(t,n-h)}\} - \left(\sum_{t'} \mathbf{s}_{t'} \mathbb{1}\{r_{t'} = r\} \right) \left(\sum_t \mathbf{s}_t \mathbb{1}\{q_{n-h} = \pi_{(n-h)}\} \right) \right]$$

Taking the derivative wrt to $\alpha_h, \boldsymbol{\beta}_h$, we get

$$\begin{aligned}
\frac{\partial \gamma_h}{\partial \boldsymbol{\beta}_h} &= 2(1+\gamma)\boldsymbol{\beta}_h \frac{e^{\alpha_h}-1}{e^{\alpha_h}+n-1}, \\
\frac{\partial \gamma_h}{\partial \alpha_h} &= (1+\gamma)\boldsymbol{\beta}_h^2 \frac{n}{(e^{\alpha_h}+n-1)^2} e^{\alpha_h}.
\end{aligned}$$

Assumption E.1. *Now consider the initialization where $\alpha_h = 0$ and $\boldsymbol{\beta}_h = (c_1\epsilon, c_2\epsilon, \ldots, c_n\epsilon)$ for a very small ϵ and some constants $c_1 \gg c_2 \gg \ldots \gg c_n$.*

Note that at the initialization, the scores are approximately equal $\mathbf{s}_t \sim 1/(n)!$. Using these to compute the gradient of γ_h at initialization, we get,

$$-\nabla_{\gamma_h}\ell \approx \frac{1}{(n)!}\mathbb{E}\left[\sum_t \mathbb{1}\{r_t = r, q_{n-h} = \pi_{(t,n-h)}\} - \left(\sum_{t'} \mathbb{1}\{r_{t'} = r\}\right)\left(\sum_t \mathbb{1}\{q_{n-h} = \pi_{(n-h)}\}\right)\right],$$

Note that all the γ_h have the approximately same gradient at $t = 0$ and therefore the gradients are clearly ordered by the $\boldsymbol{\beta}_h$ values. So, the largest $\boldsymbol{\beta}_h$ will grow the fastest. As it grows, the corresponding γ_h will grow faster and faster, leading to a positive feedback loop.

The gradient with respect to γ_h is given by

$$\Delta_h = -\mathbb{E}\nabla_{\gamma_h}\ell = \mathbb{E}\left[\sum_t \mathbf{s}_t \mathbb{1}\{r_t = r, q_{n-h} = \pi_{(t,n-h)}\} - \left(\sum_{t'} \mathbf{s}_{t'} \mathbb{1}\{r_{t'} = r\}\right)\left(\sum_t \mathbf{s}_t \mathbb{1}\{q_{n-h} = \pi_{(n-h)}\}\right)\right].$$

The gradient flow dynamics of γ_h is given by

$$\dot{\boldsymbol{\beta}}_h = 2(1+\gamma)\boldsymbol{\beta}_h \frac{e^{\alpha_h} - 1}{e^{\alpha_h} + n - 1}\Delta_h,$$
$$\dot{\alpha}_h = (1+\gamma)\boldsymbol{\beta}_h^2 \frac{n}{(e^{\alpha_h} + n - 1)^2} e^{\alpha_h}\Delta_h.$$

We will drop the $(1+\gamma)$ term for simplicity and scale the time accordingly later. Note that the time derivatives of $\alpha_h, \boldsymbol{\beta}_h$ are always positive hence they are increasing functions of time.

The conservation law for this system is given by

$$\frac{\beta_h^2}{4} - \frac{c_h^2\epsilon^2}{4} = \frac{1}{n}\left(e^{\alpha} - e^{-\alpha} - 2\alpha\right) + \left[e^{-\alpha} + \alpha - 1\right].$$

Using Lemma D.3, we can write Δ_h as,

$$\Delta_h = \left(1 - \frac{1}{|\mathcal{R}|}\right) \mathop{\mathbb{E}}_{\pi\sim s(q)} \mathbb{1}\{\pi = q\}\left(1 - \mathop{\mathbb{E}}_{\pi\sim s(q)} \mathbb{1}\{q_{n-h} = \pi_{(n-h)}\}\right).$$

the expectation at the start can be dropped as the term is independent of q.

To begin with, we will see the first head grows faster. Assume that till time t, $m_0 \leqslant \Delta_1 \leqslant 1$ and $\Delta_h \leqslant 1$ for $h > 1$. Then, we can bound the dynamics as following,

$$\dot{\beta}_1 \geqslant 2\beta_1 \frac{e^{\alpha_1} - 1}{e^{\alpha_1} + n - 1}\, c, \quad \dot{\alpha}_1 \geqslant \beta_1^2 \frac{n e^{\alpha_1}}{(e^{\alpha_1} + n - 1)^2}\, c,$$
$$\dot{\beta}_h \leqslant 2\beta_h \frac{e^{\alpha_h} - 1}{e^{\alpha_h} + n - 1}, \quad \dot{\alpha}_h \leqslant \beta_h^2 \frac{n e^{\alpha_h}}{(e^{\alpha_h} + n - 1)^2}, \quad h > 1.$$

Invoking the following lemma, we get that,

$$\frac{c_1\epsilon}{\sqrt{m_1}} \tan\left(\frac{m_0\sqrt{m_1}}{n} c_1\epsilon t\right) \leqslant \alpha_1(t),$$
$$\alpha_h(t) \leqslant \frac{c_h\epsilon}{\sqrt{m_2}} \tan\left(c_h\epsilon\sqrt{m_2}\, t\right), \quad h > 1.$$

Let α_* be the maximal value of α such that $\delta_1 \geqslant c_0$. Let t_* be such that

$$\frac{c_1\epsilon}{\sqrt{m_1}} \tan\left(\frac{m_0\sqrt{m_1}}{n} c_1\epsilon t_*\right) = \alpha_*,$$
$$t_* = \frac{n}{m_0\sqrt{m_1}c_1\epsilon} \tan^{-1}\left(\frac{\sqrt{m_1}\alpha_*}{c_1\epsilon}\right) \leqslant \frac{\pi n}{2m_0\sqrt{m_1}c_1\epsilon}.$$

Now at t_* the $\alpha_1 = O(1)$, and one can derive that $\Delta_1 > c_3 \frac{e^{-\gamma_1}}{n!}$. Now the evolution of

$$\dot{\gamma}_1 = \left(\frac{\partial\gamma_1}{\partial\alpha_1}\right)^2 \Delta_1 + \left(\frac{\partial\gamma_1}{\partial\boldsymbol{\beta}_1}\right)^2 \Delta_1,$$
$$= \boldsymbol{\beta}_1^4 \frac{n^2 e^{2\alpha_1}}{(e^{\alpha_1} + n - 1)^4}\Delta_1 + 4\boldsymbol{\beta}_1^2 \frac{(e^{\alpha_1} - 1)^2}{(e^{\alpha_1} + n - 1)^2}\Delta_1.$$

Note that

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

We can write the dynamics of γ_1 as,

$$\begin{aligned}
\dot{\gamma_1} &\geqslant m_3 \gamma_1^2 e^{-\gamma_1} \\
\frac{e^{\gamma_1}}{\gamma_1^2} \mathrm{d}\gamma_1 &\geqslant m_3 \mathrm{d}t \\
\mathrm{dEi} - e^{\gamma_1}/\gamma_1 &\geqslant m_3 \mathrm{d}t \\
\mathrm{Ei}(\gamma) - e^{\gamma_1}/\gamma_1 &\geqslant m_3 t
\end{aligned}$$

In time $\mathrm{Ei}(C^2) < 2\frac{e^{C^2}}{C^2} = t_{**}$, γ_1 will reach C^2 and $\boldsymbol{\beta}_1$ will reach C. So in time

$$T_1 = t_* + t_{**} = O(1/\epsilon) + 2\frac{e^{C^2}}{C^2},$$

we have $\boldsymbol{\beta}_1 > C$.

For such T_1, we can choose c_2 such that c_2/c_1 is sufficiently small such that

$$\alpha_h \leqslant \frac{c_h \epsilon}{\sqrt{m_2}} \tan\left(c_h \epsilon \sqrt{m_2}\, T_1\right) \leqslant \frac{c_h \epsilon}{10}.$$

So for sufficiently separate initializations, the order is preserved.

Induction. Now assume that till time T_{i-1}, we have shown that $\boldsymbol{\beta}_j > C$ and $\alpha_j > \log C$ for all $j < i$ and for $j > i$, $\boldsymbol{\beta}_j \sim c_j^{'}\epsilon$ and $\alpha_j = \frac{c_j^{'}\epsilon}{10}$.

The dynamics of the rest of the heads will be seen through their Δ_h's. For the second jump, we use similar arguments. Now at time T_1, we have $\boldsymbol{\beta}_1 > C$ and $\alpha_1 > \log C$, and for $h > 1$, $\boldsymbol{\beta}_h \sim 1.1\, c_h\epsilon$ and $\alpha_h = \frac{c_h\epsilon}{10}$.

Now to calculate the dynamics of Δ_h for $h \geqslant i$, note that

$$\Delta_h = \left(1 - \frac{1}{|\mathcal{R}|}\right) \mathop{\mathbb{E}}_{\pi\sim s(q)} \mathbb{1}\{\pi = q\} \left(1 - \mathop{\mathbb{E}}_{\pi\sim s(q)} \mathbb{1}\{q_{n-h} = \pi_{(n-h)}\}\right).$$

Consider the matched $\mathcal{S}_{i-1} = \{\pi : \pi_{n-i+1:} = q_{n-i+1:}\}$ and its complement. For $\pi \in \mathcal{S}_{i-1}$,

$$\begin{aligned}
s(\pi) &= \frac{\exp\sum_h \gamma_h \mathbb{1}\{q_{n-h} = \pi_{(n-h)}\}}{\sum_{\pi'} \exp\sum_h \gamma_h \mathbb{1}\{q_{n-h} = \pi'(n-h)\}}, \\
&= \frac{\exp\sum_{h<i}\gamma_h + \sum_{h\geqslant i}\gamma_h \mathbb{1}\{q_{n-h} = \pi'(n-h)\}}{\sum_{\pi'\in\mathcal{S}_{i-1}} \exp\sum_h \gamma_h \mathbb{1}\{q_{n-h} = \pi'(n-h)\} + \sum_{\pi'\notin\mathcal{S}_{i-1}} \exp\sum_h \gamma_h \mathbb{1}\{q_{n-h} = \pi'(n-h)\}}, \\
&= \frac{\exp\sum_{h<i}\gamma_h \exp\sum_{h\geqslant i}\gamma_h \mathbb{1}\{q_{n-h} = \pi_{(n-h)}\}}{\exp\sum_{h<i}\gamma_h \sum_{\pi'\in\mathcal{S}_{i-1}} \exp\sum_{h\geqslant i}\gamma_h \mathbb{1}\{q_{n-h} = \pi'(n-h)\} + \sum_{\pi'\notin\mathcal{S}_{i-1}} \exp\sum_h \gamma_h \mathbb{1}\{q_{n-h} = \pi'(n-h)\}}, \\
&= \frac{\exp\sum_{h\geqslant i}\gamma_h \mathbb{1}\{q_{n-h} = \pi_{(n-h)}\}}{\sum_{\pi'\in\mathcal{S}_{i-1}} \exp\sum_{h\geqslant i}\gamma_h \mathbb{1}\{q_{n-h} = \pi'(n-h)\} + \exp\{-\sum_{h<i}\gamma_h\}\sum_{\pi'\notin\mathcal{S}_{i-1}} \exp\sum_h \gamma_h \mathbb{1}\{q_{n-h} = \pi'(n-h)\}}, \\
&= \frac{\exp\sum_{h\geqslant i}\gamma_h \mathbb{1}\{q_{n-h} = \pi_{(n-h)}\}}{\sum_{\pi'\in\mathcal{S}_{i-1}} \exp\sum_{h\geqslant i}\gamma_h \mathbb{1}\{q_{n-h} = \pi'(n-h)\}} - O(e^{-C^2}).
\end{aligned}$$

So restricting the expectation to $\mathcal{S}_{i-1}$ only, we get similar dynamics just with redefined constants.

Therefore, we can repeat the same argument as earlier to show that in time $T_i = T_{i-1} + O(1/\epsilon) + 2\frac{e^{C^2}}{C^2}$, $\boldsymbol{\beta}_i$ will reach C and α_i will reach $\log C$. Repeating this for all n heads, we get the total time.

Lemma E.2. *For some constant $m_0 < 1$, consider the ODE,*

$$\dot{\beta} = 2\beta \frac{e^{\alpha}-1}{e^{\alpha}+n-1}\, m_0, \quad \dot{\alpha} = \beta^2 \frac{n e^{\alpha}}{(e^{\alpha}+n-1)^2}\, m_0.$$

with the initial conditions $\alpha(0) = 0$ and $\beta(0) = \epsilon$. Then, for time t such that $\alpha(t) < k$, we have the following bound on

$$\frac{\epsilon}{\sqrt{m_1}} \tan\left(\frac{m_0\sqrt{m_1}}{n}\epsilon t\right) \leqslant \alpha(t) \leqslant \frac{\epsilon}{\sqrt{m_2}} \tan\left(\epsilon\sqrt{m_2}\, t\right),$$

for some constants $m_1, m_2 > 0$.

Proof. Consider the following ODE,

$$\dot{\beta} = 2\beta \frac{e^{\alpha} - 1}{e^{\alpha} + n - 1} m_0,$$
$$\dot{\alpha} = \beta^2 \frac{n e^{\alpha}}{(e^{\alpha} + n - 1)^2} m_0.$$

with the initial conditions $\alpha(0) = 0$ and $\beta(0) = \epsilon$. Here $C > 0$ is a constant. For this system, we will have the following conservation law,

$$\frac{\beta^2}{4} - \frac{\epsilon^2}{4} = \frac{1}{n}\left(e^{\alpha} - e^{-\alpha} - 2\alpha\right) + \left[e^{-\alpha} + \alpha - 1\right].$$

Still the dynamics cannot be precisely controlled, but we can show that α and β will grow to large values in finite time. Till the time $\alpha < k$, we have the following lower bound on the dynamics, the conservation law gives us that

$$\beta^2 \approx \epsilon^2 + m_1\alpha^2,$$

Using this, we get,

$$\left[\epsilon^2 + m_2\alpha^2\right] \geqslant \dot{\alpha} \geqslant \left[\epsilon^2 + m_1\alpha^2\right] \frac{1}{n} m_0,$$

This gives the following bound for the α till it reaches k,

$$\frac{\epsilon}{\sqrt{m_1}} \tan\left(\frac{m_0\sqrt{m_1}}{n}\epsilon t\right) \leqslant \alpha(t) \leqslant \frac{\epsilon}{\sqrt{m_2}} \tan\left(\epsilon\sqrt{m_2}\, t\right)$$

□

F Additional Experiments with ablation studies.

We train a two–layer simplified transformer model and perform a series of ablation studies to examine the influence of different modeling assumptions. In particular, we evaluate the impact of assumptions (A0), (A1), (A2), and (A3) by selectively relaxing them during training. Results under relaxed assumptions are shown in the accompanying plots; refer to the glossary below for a concise description of each assumption and its role in the analysis.

Glossary. To quickly find the plots corresponding to different assumptions, refer to the following glossary:

Table 3: Glossary of experimental settings and corresponding figures/sections.

Setting	Description	Reference
Simplified model	The baseline two-layer transformer model trained under all assumptions **(A0–A3)**.	Figure 4, § F.0.1
Relaxing (A0) and (A3)	Experiments where the fixed window size and certain attention constraints only on the response tokens are removed.	Figure 5, § F.0.2
Attention only transformer	Further relaxations involving attention-only mechanisms without assumptions **A0–A4**.	Figure 10, § F.0.3
Full transformer architecture	Full transformer with multiple heads in the first layer and a single head in the second layer on a task of order 2.	Figure 2, § F.0.4
Without β^2 term	Experiments investigating the role of the β^2 parameterization.	Figure 11, Figure 12, § F.0.5

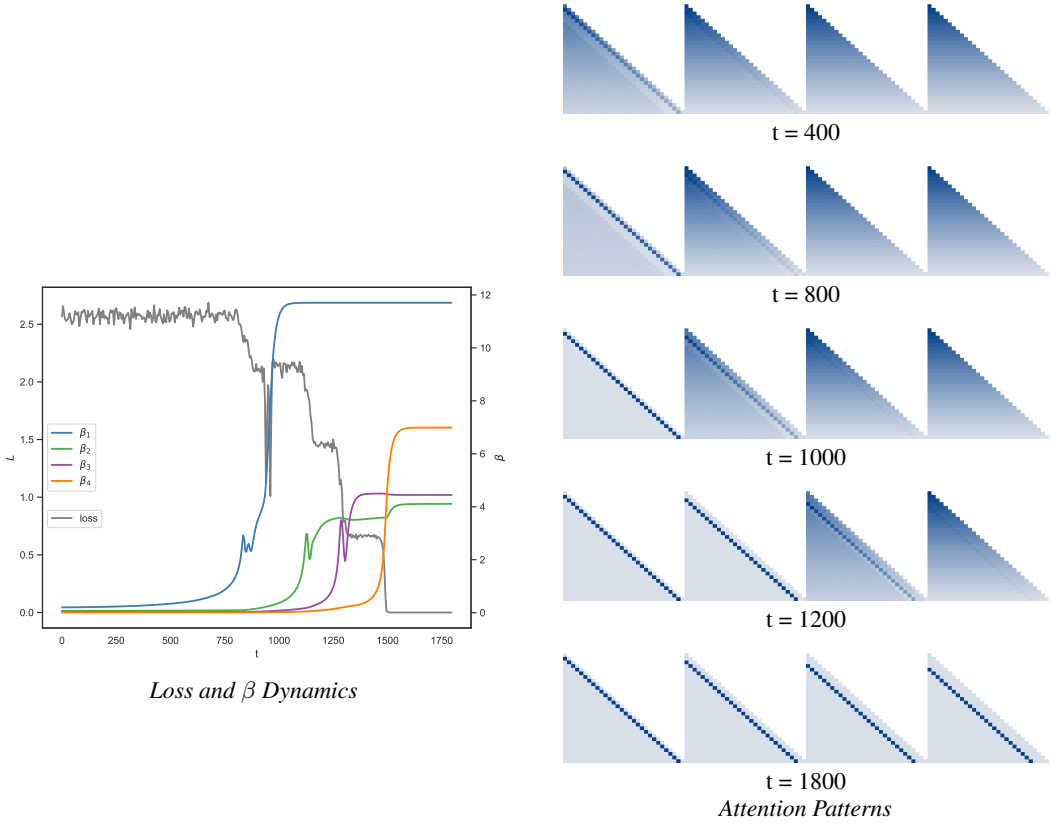
F.0.1 Training Dynamics Across Initialization Scales with the simple transformer

First, we will provide empirical evidence for the training dynamics of simplified two-layer transformer model under assumptions **A0-4** verifying the theoretical claims made in Theorem 3.1 and Theorem 3.2. Here $n = 4$, so we need a transformer with $k = n - 1 = 3$ heads in the first layer. See Figure 4 for the training dynamics and attention patterns. We rescale the time axis by the initialization scale ϵ, i.e., $t \to t/\epsilon$ to highlight the characteristic timescale of parameter jumps. The trajectories converge to a limiting trajectory showing that the timescale of the jumps is $O(1/\epsilon)$.

Furthermore, we observe the conservation law between $\boldsymbol{\beta}$ and α parameters in the right plot of Figure 4a. At the start of evolution $\beta \approx \alpha$, as the conservation states at the small scales and they grow in scale together as the loss abruptly decreases at the end of each stage. Finally, we observe the emergence of sparse attention and directional bias in Figure 4b. Note that first $\mathbf{w}_1^1$ of the head grows and all the remaining relative position parameters decay together confirming the theoretical prediction of Theorem 3.1. After the jump they remain stationary as shown in Theorem 3.2. In the next stage, $\mathbf{w}_1^2$ grows and the remaining relative position parameters $(\mathbf{w}_3^2, \mathbf{w}_4^2)$ decay together. Note that $\mathbf{w}_1^2$ also decays — a behavior our theory does not capture, attributable to finite-scale effects.

F.0.2 Beyond assumptions A0 and A3

In this section, we will present experimental results that relax assumptions (A0) and (A3). Specifically, we will explore the effects of removing the fixed window size constraint and allowing for more general attention mechanisms. The results demonstrate that even without these assumptions, the model still exhibits key behaviors of incremental learning. In Figure 9, we experiment with $n = 4$ and, due to removing assumption (A3), we use $n = 4$ heads in the first layer (one more than the $k = n - 1 = 3$ heads needed in the simplified model). The attention patterns in Figure 9b show that the model still learns to focus on the relevant positions incrementally over time, similar to the



(a) (b)

Figure 9: **Without Assumptions A0 and A3.** The left panel shows the loss and β parameter dynamics. The right panel displays attention patterns at times 400, 800, 1000, 1200, and 1800 demonstrating incremental learning of the model.

behavior observed under the original assumptions. The loss and $\boldsymbol{\beta}$ dynamics in Figure 9a further confirm that the training process remains stage-wise and the abrupt change in loss is coupled with an increase in the $\boldsymbol{\beta}$ parameters.

F.0.3 Attention only transformer

In this section, we will present the empirical evidence without the assumptions **A0-4** of the attention only transformer model with multi-headed disentangled attention. The experiments demonstrate that without these assumptions, the model exhibits the incremental learning behaviour. See Figure 10 for a comprehensive analysis of multiple attention heads and their dynamics.

More importantly, in Figure 10c, we analyze the $\mathbf{Q}^\top\mathbf{K}$ matrices for the parameters in the second layer. We observe that the $\mathbf{Q}^\top\mathbf{K}$ matrices maintain a block diagonal structure. A block seems to have a form where the diagonal elements are positive and the off-diagonal elements are negative, so it is a scaled version of $\widetilde{\mathrm{I}}_n$ with a positive scaling factor. Note that it is the same $\widetilde{\mathrm{I}}_n \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ where the first $n \times n$ block is given by $\widetilde{\mathrm{I}}_{:n,:n} = \mathrm{I}_n - \gamma \mathbf{1}_n \mathbf{1}_n^\top$ defined in the subsection 2.3. Let $\mathrm{r}_g^{(h)}$ be the output of head h for the gth token. Then, with the block structure the presoftmax attention scores

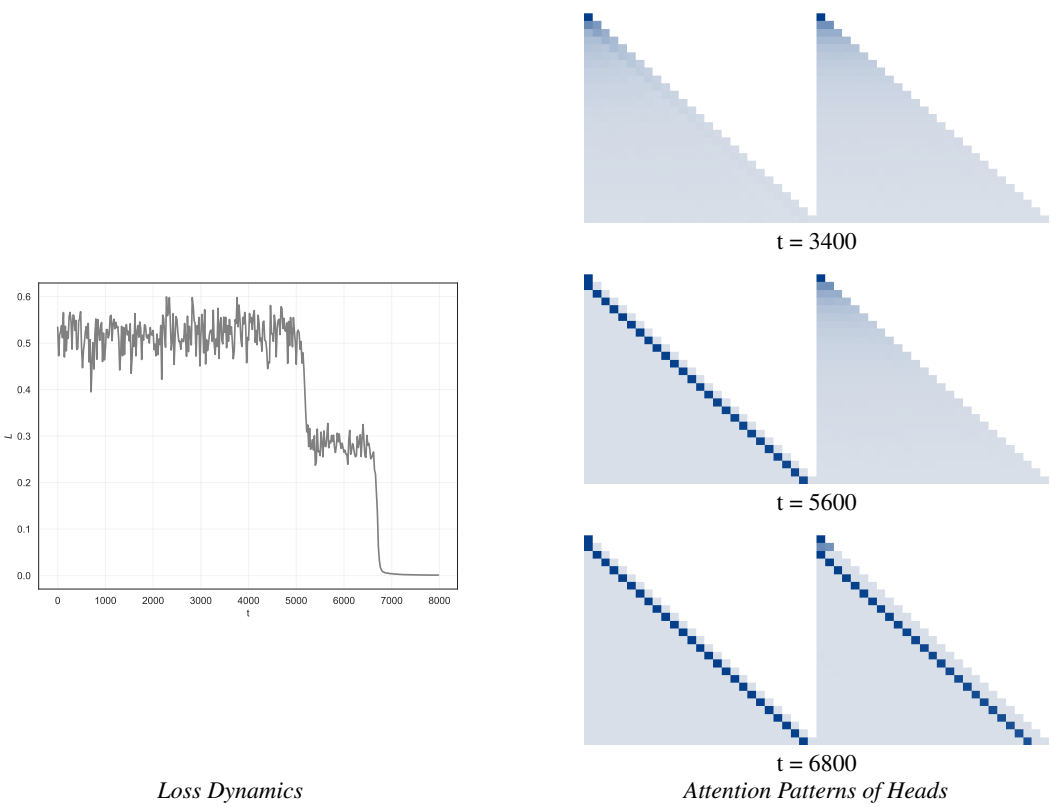


Figure 10: **Dynamics of attention-only transformer model.** **(a)** Loss dynamics during training. **(b)** Attention patterns for Head Attention 0 at times 3400, 5600, and 6800 exhibiting incremental learning. **(c)** We consider the dynamics of $\mathbf{Q}^\top\mathbf{K}$ for the parameters in layer 2; we see that it maintains a block-diagonal structure very close to the parameterization with assumption (A2).

after the second layer are given by

$$\begin{aligned}
\widetilde{\mathbf{s}}_t &= \sum_{g,h=0}^{k} \left[\mathbf{R}_{l-1}^{g}\right]^\top \left[\mathbf{Q}^\top\mathbf{K}\right]_{gh} \left[\mathbf{R}_t^h\right], \\
&\approx \sum_{h=1}^{k} \left[\beta_{h,h} \left[\mathbf{R}_{l-1}^{h}\right]^\top \widetilde{\mathrm{I}}_n \left[\mathbf{R}_t^h\right] - \sum_{g=0,g\neq h}^{k-1} \beta_{g,h} \left[\mathbf{R}_{l-1}^{h}\right]^\top \widetilde{\mathrm{I}}_n \left[\mathbf{R}_t^g\right]\right], \\
&= \sum_{h=1}^{k} \left[2\beta_{h,h} \left[\mathbf{R}_{l-1}^{h}\right]^\top \widetilde{\mathrm{I}}_n \left[\mathbf{R}_t^h\right] - \sum_{g=0}^{k-1} \beta_{g,h} \left[\mathbf{R}_{l-1}^{g}\right]^\top \widetilde{\mathrm{I}}_n \left[\mathbf{R}_t^h\right]\right],
\end{aligned}$$

where $\beta_{h,g}$ are some positive constants as seen in the Figure 10c. As $\sum_{g=0}^{k-1} \beta_{g,h} \left[\mathbf{R}_{l-1}^{g}\right]^{\top} \widetilde{\mathrm{I}}_n \left[\mathbf{R}_{t}^{h}\right]$ is approximately constant addition for all t and hence diagonally dominated just by the diagonal terms, this justifies the assumption **(A2)**.

F.0.4 Full transformer model

Experimental Setting. In this section, we present experimental results for the full transformer model without the simplifying assumptions used in our theoretical analysis. We consider the task with $n = 2$ and vocabulary size of 6. For this task, we train a transformer with 2 heads in the first layer and 1 head in the second layer. The transformer uses relative positional encoding, has MLP layers, uses layernorm and skip connection. We train on the standard cross-entropy loss using Adam and a constant learning rate of 0.003. We initialize all the parameters with a standard Kaiming initialization.

Despite the additional complexity compared to the simplified model, Figure 2 demonstrates that the model still exhibits stage-wise learning behavior with clear transitions in the loss curves. The attention patterns of the heads show the incremental learning behaviour. Note that $n = 2$ and the response tokens appear with a periodicity of $n + 1 = 3$. At time $t = 400$, amidst a small plateau, the heads do not attend to any particular tokens. At $t = 1100$, the second head becomes active and at the response tokens (every 3rd token) learns to attend to the *previous* token. At $t = 2800$, the first head learns to attend to the (-2)-token, and each response token attends to its corresponding (-2)-token.

F.0.5 Positivity and parameterization of Second layer

In this subsection, we investigate the role of the β^2 parameterization by training the model without this term, using only the standard β parameters. This investigation helps validate the robustness of the incremental learning phenomenon and provides additional insights into the role of parameter positivity constraints. See Figure 12 for the training dynamics with this alternative parameterization and a negative initialization.

F.0.6 Incomplete Permutations

In this section, we investigate the model's behavior when trained on incomplete permutations, where not all possible permutations are present in the training data. This variant tests the robustness of the incremental learning phenomenon to variations in the task structure. Figure 13 shows the loss and β dynamics alongside the evolving attention patterns at different training stages. The results demonstrate that even with incomplete permutations, the model exhibits stage-wise learning behavior with clear phase transitions in loss and parameter values. Here we choose a task of order 4 and the context contains 12 permutations instead of the full set of 24 permutations. We train a simplified two-layer transformer with all the simplifying assumptions.

F.1 Compute Resources

The experiments with the value-softmax model were conducted on CPU, the experiments with toy transformers used one A100 node with different experiments running in parallel on separate GPUs. All experiments combined take less than 1 day of walltime.

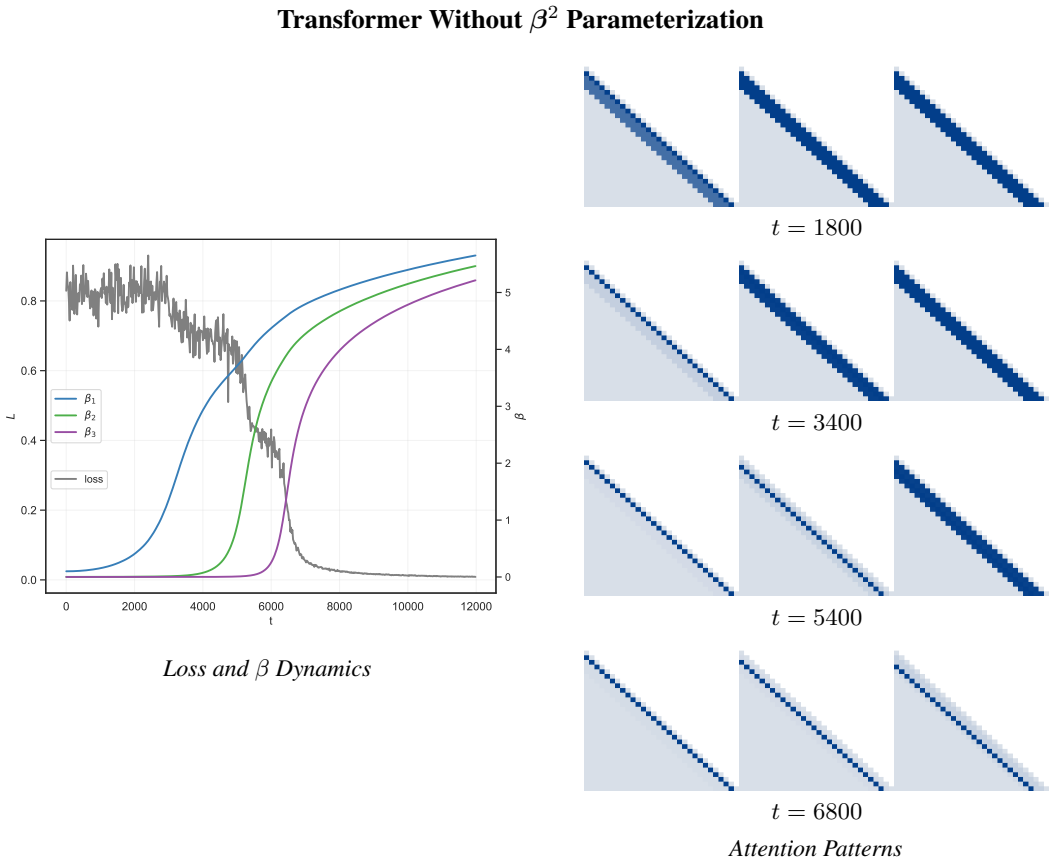
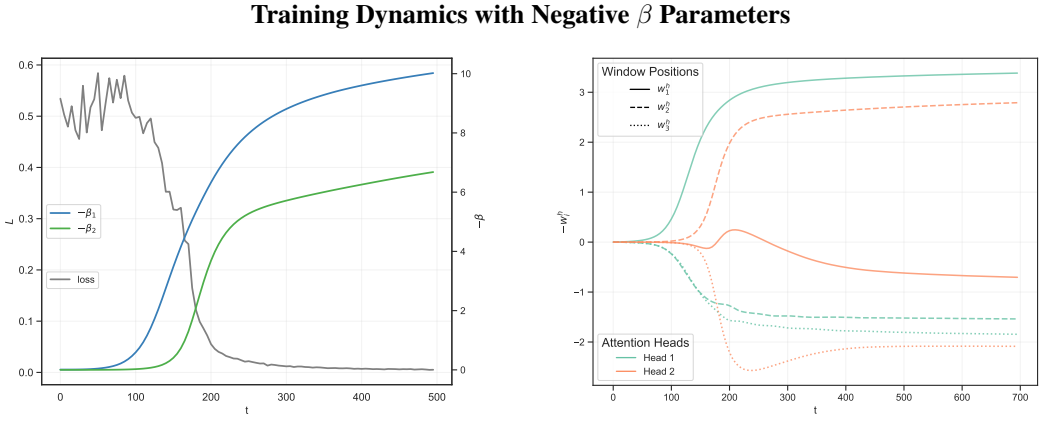


Figure 11: **Without the β^2 parameterization.** The left panel shows the loss and β parameter dynamics. The right panel displays attention patterns at times 1800, 3400, 5400, and 6800 demonstrating incremental learning without the β^2 term.



(a) Loss and negative β parameters over time

(b) Weights of the window (attention head parameters) over time

Figure 12: **Training dynamics with negative β parameterization. (a)** Loss dynamics (gray) alongside the negative β parameters ($-\beta_i$, colored lines) showing their evolution during training. **(b)** The negative values of the fixed window weights ($-\mathbf{w}_i^h$) for each attention head and position. The plot shows convergence in the case of initialization with negative $\boldsymbol{\beta}$'s.

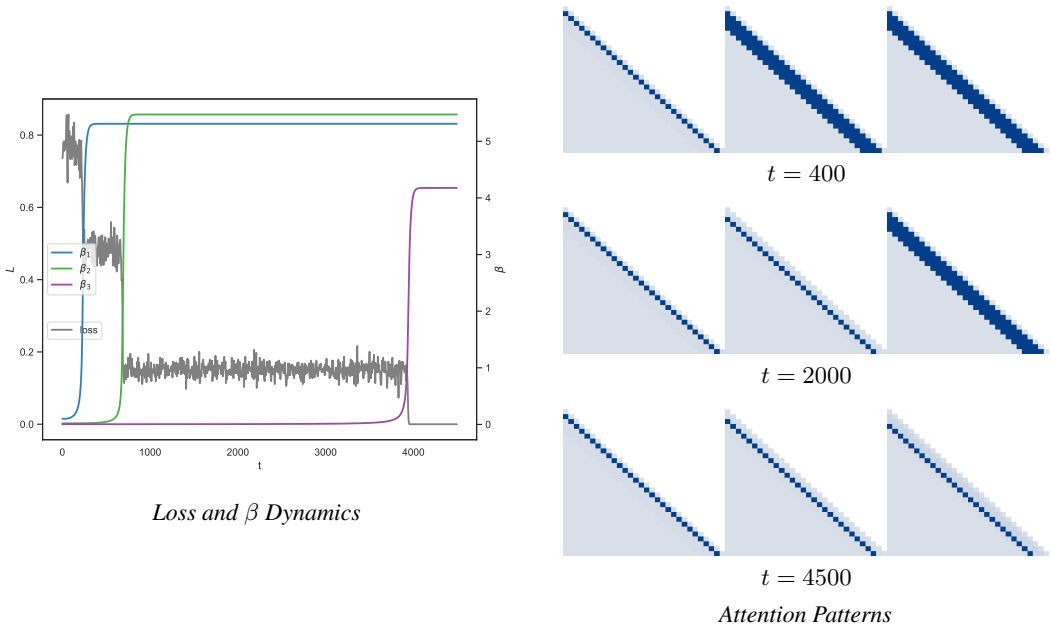


Figure 13: **Incomplete Permutations.** The left panel shows the loss and β parameter dynamics. The right panel displays attention patterns at times 400, 2000, and 4500 demonstrating how the model learns without all permutations present in the prompt.

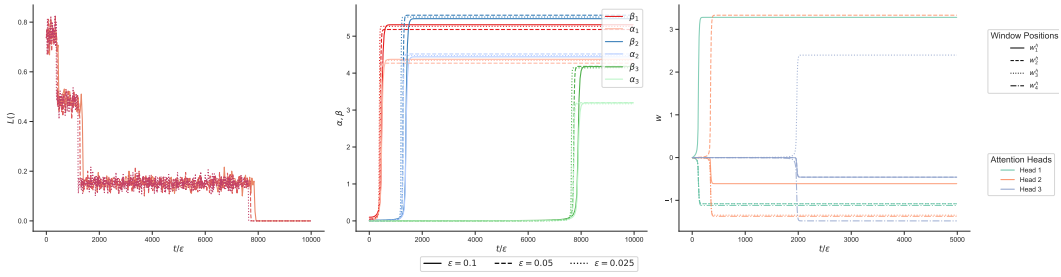


Figure 14: **Relaxing Assumptions 2.4 and 2.5: cross-entropy loss with incomplete context ($n = 4$, $l = 60$).** Runs with various initialization scales ε trained with CE loss, where not all permutations appear in the context. Time is rescaled by ε to expose the $O(1/\varepsilon)$ timescale predicted by Theorem 3.1. The trajectories collapse under this rescaling and the conservation law between β and α parameters holds, confirming that the $O(1/\varepsilon)$ timescale and conservation law are not artifacts of the DP loss (Assumption 2.5) or the complete-context setup.

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: the abstract and introduction list our theoretical contributions and experimental results, which are all described in the main sections.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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 paper is that the theoretical contributions are made for a simplified model that doesn't fully reflect the architecture of modern LLMs. We acknowledge this in the main paper.

   Guidelines:

   - The answer [N/A] 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: we did our best to make sure that all theoretical claims are carefully framed and proven.

Guidelines:

- The answer [N/A] 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 the experimental details needed to reproduce our experiments in Appendix F.

Guidelines:

- The answer [N/A] 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: we submit the code with appropriate instructions for all experiments excluding the comparison of pretrained softmax and sigmoid LLM.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: we provide our experimental setting in Appendix F.

   Guidelines:

   - The answer [N/A] 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: the value-softmax runs are single-run and show the dynamics, so including error bars is not useful. For the toy-transformer experiments, we omit error bars for readability. All of our results are consistent across different settings; besides, the experimental results are not the core contribution of the paper.

   Guidelines:

   - The answer [N/A] 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: we provide the details on compute used in Appendix F.1.

   Guidelines:

   - The answer [N/A] 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: our paper conforms with the Code of ethics.

   Guidelines:

   - The answer [N/A] 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: [N/A]

Justification: our paper is a theoretical work on Machine Learning. We do not see any direct societal impacts of our work.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: we do not release anything with a risk of misuse.

Guidelines:

- The answer [N/A] 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: [Yes]

Justification: the only existing asset we use are the models and the codebase for the LLM experiment; we cite the authors and provide a link to their codebase.

Guidelines:

- The answer [N/A] 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: [N/A]

    Justification: we do not release any assets besides the code submitted as the supplementary material.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: our work does not involve crowdsourcing and human subjects.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: our work does not involve crowdsourcing and human subjects.

Guidelines:

- The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were only use for minor assistance with text editing and code.

    Guidelines:

    - The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.