Incremental Learning in Transformers for In-Context Associative Recall

Anonymous Authors[1]

Abstract

Transformers acquire in-context learning abilities in abrupt phases during training, often unfolding over multiple stages, during which certain keys circuits like induction heads emerge. In this work, we characterize the training dynamics behind the emergence of such circuits during these stages. We focus on a synthetic in-context associative recall task, where sequences are drawn from random maps between a permutation group and a vocabulary range and the model is required to complete the mapping of a permutation by retrieving it from the context. On this task, we study the trajectories of gradient flow of a simplified two-layer, attention-only transformer. Leveraging symmetries in both the transformer architecture and the data, we derive conservation laws that guide the dynamics of the parameters. These conservation laws crucially reveal how initialization —both in shape and scale— determines the order of learning as well as the timescales over which such circuits emerge revealing the implicit curriculum. Furthermore, at the limit of vanishing scale of initialization, we characterize the trajectory of the gradient flow revealing how the training jumps from one saddle to another. Finally, we provide empirical evidence across different architectural choices, validating our simplifications and generalising the insights from our analysis beyond the simple 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 a characteristic ability of transformer models. 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 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 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 a factual recall: the model sees a sequence containing a length-k pattern that reappears later, and must match the last k tokens, locate the previous occurrence of that pattern in the context, and predict the next token by recalling the associated response (e.g., contexts of the form $[A][B][C][X]\cdots[A][B][C]$). This task is implemented by the same induction-style circuit as in-context n-grams, but admits cleaner analysis.

Crucially, we couple this task with a series of principled

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

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

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), boolean and formal language recognition (Bhattamishra et al., 2023; Akyürek et al., 2024), and n-gram prediction.

n-gram models. n-grams models are related our work as the transformer circuit that solves our task also solves the taks of in-context learning n-grams. n-gram language models (Shannon, 1948; Jurafsky & Martin, 2009) provide a natural testbed for analyzing transformer behavior. Several recent works adopt this viewpoint: the optimization landscape has been analyzed in (Makkuva et al., 2024), expressivity over n-gram distributions in (Svete & Cotterell, 2024), and in-context generalization in (Rajaraman et al., 2024). Other studies connect ICL to the emergence of induction heads (Bietti et al., 2024) and their acquisition through gradient descent (Nichani et al., 2024). Edelman et al. (2024) identify stage-wise dynamics in transformer training on in-context n-gram prediction, where intermediate solutions resemble sub-n-grams, while Varre et al. (2025) formalize these sub-n-grams as near-stationary points. Finally, Chen et al. (2024b) investigate the same task with a modified architecture and initialization scheme that enforces head specialization from the start, thereby eliminating the stage-wise dynamics central to our analysis.

Incremental learning. Plateau-shaped learning curves arise broadly in neural network training, beyond ICL. Early work by Fukumizu & Amari (2000) linked such phenomena to critical points in supervised learning. Related characterizations appear in simplified models such as 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 (Gissin et al., 2020; Saxe et al., 2019; Gidel et al., 2019; Jacot et al., 2021; Berthier, 2022; Pesme & 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) studied the stage wise dynamics in Factual recall with linear attention. These results provide theoretical tools that we build on to characterize plateaus in in-context learning.

2. Problem Setting

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

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., $[A][B][C]\dots[A]$. When the model encounters the second $[A]$, the circuit attends to the first $[A]$ and predicts the subsequent token, $[B]$. This paper investigates how models handle a more general version of this pattern: $[A][B][C][X]...[A][B][C]$. In this setting, the model must recognize the entire sequence $[A][B][C]$, locate its previous occurrence in the context, and then use it to predict the next token $[X]$.

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

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

Note that $\pi_1, \pi_2, \ldots$ each represent a sequence of k tokens, rather than a single token. For example, $\pi_1 = 0, 1, \ldots, k-1$. Figure 1 illustrates the task for $k = 2$ with a response set $\{A, B\}$. We define the vocabulary as $\mathcal{S} = [k] \cup \mathcal{R}$. Each sequence has a fixed length of $l = (k+1)! + k$.

To solve this task, the transformer must learn to identify the part of the context that matches the final k tokens and recall the subsequent token. For completeness, the context contains all possible query permutations, ensuring that the model can always retrieve the correct response, which enables exact learning.

2.2. Multi-headed Attention-Only Transformer

We analyze a specific attention-only transformer with a two-layer structure. The first layer contains k attention heads, and the second layer contains a single head. The architecture is based on the disentangled transformer (Friedman et al., 2023; Nichani et al., 2024) and incorporates several simplifications from prior work (Nichani et al., 2024; Edelman et al., 2024; Varre et al., 2025). In particular, tokens are represented with orthogonal encodings, and the only trainable parameters are in the attention layers. We exclude both trainable value matrices and MLP layers. Below, we provide a formal description of this simplified transformer.

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

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

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

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

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

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}^0 & \mathbf{R}^1 & \ldots & \mathbf{R}^k \end{bmatrix} \in \mathbb{R}^{l \times (k+1)\mathcal{S}} = \sum_{i=0}^{k} \left(\mathsf{e}_i^{k+1}\right)^\top \otimes \mathbf{R}^i.$$

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

$$\mathbf{R}_+ = \boldsymbol{\sigma}\left(\mathbf{R}\mathbf{Q}^\top \mathbf{K}\mathbf{R}^\top\right) \mathbf{R}V.$$

We choose the value matrix $V = \mathsf{e}_0^{k+1} \otimes \mathrm{I}_{|\mathcal{S}|} \in \mathbb{R}^{(k+1)|\mathcal{S}| \times |\mathcal{S}|}$. This matrix consists of a column of blocks, with the identity matrix as the first block and zeros elsewhere. By construction, V extracts the skip connection from the concatenated output of the first layer, i.e., $\mathbf{R}V = \mathbf{R}^0 = X$, using the mixed-product property of the Kronecker product.

The model output. Note that the loss is computed only on the *last token* of the sequence. Consequently, the model's output depends only on the embedding of the final token after the second layer, i.e., $\mathbf{p} = (\mathbf{R}_+)_{l-1} \in \mathbb{R}^{|\mathcal{S}|}$.

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

We denote the attention scores by $\mathbf{s} = \boldsymbol{\sigma}\left(\mathbf{R}\mathbf{Q}^\top \mathbf{K}\mathbf{R}^\top\right)_{l-1}$ and the corresponding pre-softmax scores by $\widetilde{\mathbf{s}} = \left(\mathbf{R}\mathbf{Q}^\top \mathbf{K}\mathbf{R}^\top\right)_{l-1}$. The choice of V together with the orthogonal embeddings ensures that $\mathbf{p}$ is a valid probability distribution over the vocabulary, i.e., $\mathbf{p} \in \Delta^{|\mathcal{S}|}$, and requires no further normalization. Hence, the output of the model is given by

$$\mathbf{p} = X^\top \mathbf{s} = X^\top \boldsymbol{\sigma}(\widetilde{\mathbf{s}}).$$

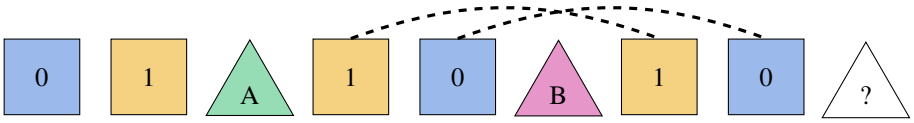


Figure 1. Illustration of the in-context associative recall task. The sequence shows mappings between query permutation elements (rectangles labeled 0, 1) and response tokens (triangles labeled A, B). The model must predict the missing association, marked with "?".

2.3. A simplified model

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

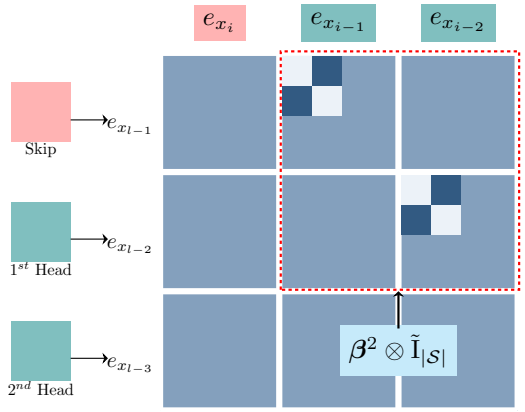


Figure 2. Layer-2 representation $\boldsymbol{B}^\top$ of the structure of the optimal solution constructed by the transfomer We use it to simplify this structure by a diagonal block matrix with trainable scales.

The transformer's solution. To solve the task, the transformer must learn to attend to the portion of the context that matches the final k tokens. This mechanism is implemented through a multi-head construction, variants of which have appeared in prior work (Edelman et al., 2024; Rajaraman et al., 2024). Its parameters are defined as

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

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

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

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

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

This construction is illustrated in the Figure 2. One of the remarkable aspects of this solution is if it is initialized with the same shape of the block diagonal structure, it seems to maintain almost the same shape throughout the training process, see Figure 7c in the appendix.

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

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

In words, we make the following modifications:

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

(A1) We configure the second-layer attention parameters to match the structure of the optimal solution shown in Figure 2. Note that Figure 7c in the appendix shows that during training, the parameters of the second attention layer remain close to this structure, hence this assumption is not very restrictive and preserves the essence of the analysis with the full model. This configuration is held fixed, and we train only the scalar

multipliers that scale these parameters. In particular, we introduce a parameter $\boldsymbol{\beta} \in \mathbb{R}^k$ and parameterize $\mathbf{Q}^\top \mathbf{K}$ as

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

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

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

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

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

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

2.4. The final problem setup

Training Objective (A4). For our analysis, we replace the conventional cross-entropy (CE) loss with the dot-product (DP) loss

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

The minimum of the DP loss is

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

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

Finally, given the per-sequence DP loss, the population DP loss is

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

$$= 1 - \mathbb{E}_{f_\tau, q} \left\langle \mathbf{p}(\boldsymbol{\theta}), e_{f_\tau(q)} \right\rangle . \tag{3}$$

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

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

3. Technical Results

We study how signals propagate through the layers of a transformer and how the associated training dynamics unfold.

A stagewise learning process. We train the simplified transformer model with SGD and momentum on the DP loss over the in-context learning task of order $k+1$. Note that a k-head transformer can solve this task exactly: once the model matches the k tokens in a permutation, the remaining token is uniquely determined. We train the model with different scales of initialization of $\boldsymbol{\beta}$ and for $k = 4$, see Figure 3 and plot with the time rescaled as t/ϵ as ϵ is the scale of initialization. We observe that training proceeds in distinct stages, with the model lingering at plateaus of decreasing loss before abruptly transitioning to lower loss levels. This stagewise behavior becomes more pronounced as the initialization scale decreases, there should exist a limiting dynamics as the scale goes to zero. This phenomenon is reminiscent of the saddle-to-saddle dynamics observed in deep networks under small-scale initialization (Jacot et al., 2021; Pesme & Flammarion, 2023).

To mechanistically interpret these intermediate stages, we analyze what the model represents on each plateau, see Figure 3, Figure 4 in the Appendix. At the first plateau, the model learns to match a single token in the context: one attention head h_1 and its associated coefficient $\boldsymbol{\beta}_{h_1}$ are activated. At the second plateau, an additional head h_2 with coefficient $\boldsymbol{\beta}_{h_2}$ is activated, enabling the model to match two tokens of the query, q_{k+1-h_1}, q_{k+1-h_2} in the context. This process continues incrementally: at each stage, a new head–coefficient pair is activated, allowing the model to match one additional token. After k such stages, all tokens in the query are matched and the model achieves zero loss.

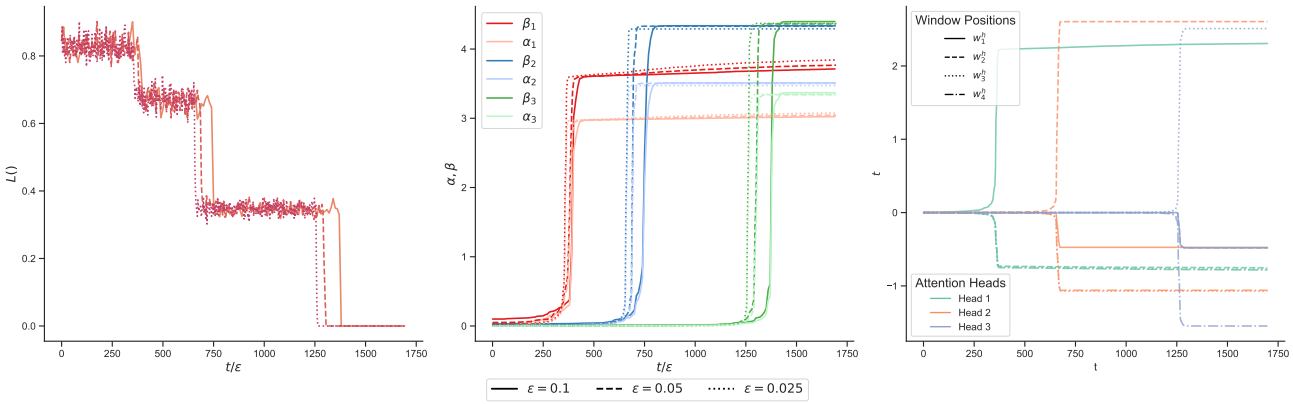


Figure 3. We train a **simplified two-layer transformer** using **cross-entropy loss** under the assumptions **(A0–A3)**. We initialize the parameter vector $\boldsymbol{\beta}$ as $\boldsymbol{\beta} = \epsilon(\beta_1, \beta_2, \beta_3)$ where $\beta_1 > \beta_2 > \beta_3$, for different values of ϵ. The dynamics are plotted with respect to a **rescaled time** t/ϵ

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

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

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 $\boldsymbol{\beta}_{(h_1)} > \boldsymbol{\beta}_{(h_2)} > \dots > \boldsymbol{\beta}_{(h_k)}$, the heads are activated sequentially in the order $h_1, h_2, \dots h_k$, see Fig. 5 in App.. Thus the implicit regularization induced by the scale and shape of initialization provides a natural curriculum, guiding the model to acquire the task in a stage-wise manner. For the remainder of the analysis, we assume without loss of generality that the coefficients are ordered $\boldsymbol{\beta}_1 > \boldsymbol{\beta}_2 > \dots > \boldsymbol{\beta}_k$, so that heads are activated in order $1, 2, \dots, k$. By re-indexing the heads, the analysis for arbitrary initial orderings reduces to this canonical case.

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, all heads are symmetric, and there is no signal that breaks this symmetry.

.

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

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

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

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

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

(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*

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

Symmetries and directional bias. All relative position encodings except $\mathbf{w}_1^1$ (which corresponds to the previous token) evolves together. This follows from the inherent symmetry of the task: since token positions can be permuted within a sequence without leaving the distribution, the dynamics must preserve this symmetry. Combined with the symmetry of the initialization, this leads to the directional bias described above, see rightmost plot in Fig. 3.

Sparsification of attention. The transformer rapidly learns to attend to the first token. There is a clear dichotomy: the embedding corresponding to this token grows, while the others decay at proportional rates, leading to rapid sparsification of attention.

We will give a brief sketch of how gradient flow shapes the attention parameters in the case of the first jump in Theorem 3.1(b). The gradient with respect to $\mathbf{w}^1_m$ is proportional to covariance between two random variables over the choice of permutation $\pi \sim \mathbf{s}$ and the input function f_τ. The first random variable is the indicator that the token at position $k-m$ in the permutation matches the query token at position $k-1$, i.e., $\mathbb{1}\{q_{k-1} = \pi_{k-m}\}$. The second random variable is the indicator that the output of the permutation matches the output of the input function, i.e., $\mathbb{1}\{f_\tau(q) = f_\tau(\pi)\}$.

$$\frac{\mathrm{d}\mathbf{w}^1_m}{\mathrm{d}t} \propto \mathop{\mathbb{E}}_{f_\tau, q} \mathop{\mathrm{Cov}}_{\pi\sim\mathbf{s}} \left(\mathbb{1}\{q_{k-1} = \pi_{k-m}\}, \mathbb{1}\{f_\tau(q) = f_\tau(\pi)\}\right)$$

We first consider the case $m = 1$. When $\pi = q$, both indicator variables equal 1; when $\pi \neq q$, the two variables are independent. Using this observation, one can show that the resulting covariance is positive (see App. D.3 for a formal proof). Consequently, $\mathbf{w}^1_1$ increases under gradient flow. In contrast, for $m \neq 1$, the two variables are uncorrelated when $\pi = q$ and independent when $\pi \neq q$. In this case, the covariance can be shown to be negative (see App. D.4 for the formal proof), implying a decrease in $\mathbf{w}^1_m$.

Conservation laws. For this initialization, we obtain an exact ODE description of the dynamics. On any compact set, the time derivatives are bounded away from zero, ensuring that both α_1 and $\boldsymbol{\beta}_1$ diverge to infinity, where the gradient is zero. Their evolution is coupled, and a conserved quantity is derived. Conservation laws are common in dynamical systems, and recent work has identified conservation principles in transformers as well (Marcotte et al., 2025). However, prior results are typically restricted to a single attention layer. In contrast, our result shows how parameters across multiple layers, separated by the softmax, jointly obey a conservation law.

Coupled dynamics and timescales. These conservation laws allow us to derive the timescale of the jump, i.e., how long training remains in the plateau. Suppose $\boldsymbol{\beta}_1(0) = \epsilon$ with $\epsilon \approx 0$. Define $s = e^{\alpha_1} - 1$ with $s(0) = 0$. The conservation law becomes

$$\frac{k-1}{k^2}\left[s + 1 + \frac{k-1}{s+1} + (k-2)\log(1+s) - k\right] = \frac{\boldsymbol{\beta}_1^2 - \epsilon^2}{4}.$$

For small s, a Taylor expansion gives $\boldsymbol{\beta}_1^2 \approx 4s^2 + \epsilon^2$, and the local dynamics reduce to

$$\mathrm{d}s \sim c(s^2 + \epsilon^2). \tag{4}$$

Thus the growth of parameters in self-attention has an information exponent of 2 (Arous et al., 2021). Solving the ODE yields $s = \epsilon \tan(\epsilon T)$, implying that s requires $O(1/\epsilon)$ time to reach $O(1)$. The timescale has matched the observation in Figure 3. Thus, the dynamics spend this amount of time lingering at the plateau. 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

$$\mathrm{d}s \sim bse^{-s} \implies \int \frac{e^s}{s}\mathrm{d}s = bT \tag{5}$$

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

The subsequent jumps. Similar to the first jump, we can analyze the subsequent ones. We consider a stylized initialization, denoted $\mathcal{I}_h$, where $\boldsymbol{\beta}_i = c$ for all $i \in [1, h-1]$, $\boldsymbol{\beta}_h = \epsilon$, and $\boldsymbol{\beta}_j = 0$ for all $j > h$, with c taken to be very large. Likewise, we set $\mathbf{w}^i = c_1\mathbf{e}^k_i + c_2\mathbf{1}$ 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. *Consider the simplified transformer model $\boldsymbol{\theta} = (\mathbf{w}^1, \mathbf{w}^2, \ldots \mathbf{w}^k, \boldsymbol{\beta})$ with k heads and initialization $\mathcal{I}_h$, evolving under gradient flow on the DP loss. Then:*

(a) ***Near-Stationarity of the macroscopic variables:*** *The gradients of parameters in the first $h-1$ heads vanish at scale $\nabla_{\mathbf{w}^i}\mathcal{L}, \nabla_{\boldsymbol{\beta}_i}\mathcal{L} = O(e^{-c})$ for $i \in [1, h-1]$.*

(b) ***Dynamics of the microscopic variables:*** *At $c \to \infty$, the dynamics of the remaining parameters corresponds to the those of the first jump on a task of reduced order.*

A key detail is the interplay between macroscopic parameters (the large c) and microscopic parameters (the small ϵ). In this setting, the striking feature is that the macroscopic parameters remain stationary while the microscopic ones evolve. The near-stationarity of the macroscopic parameters can be seen as a consequence of (Varre et al., 2025, Proposition 3.2), where one can show that the gradient of logits, i.e., the score only depend on the last $(h-1)$ tokens.

3.1. A complete picture of the dynamics

An α, β model. To give the full picture of the dynamics, we consider the case where $\mathbf{w}^i$'s are parameterized as the following: $\mathbf{w}^i = \alpha_i\mathbf{e}^k_i$, This model serves the purpose of showing how to stitch the jumps together and give an idea of the relevant dynamics.

Theorem 3.3. *Consider the simplified (α, β) model with parameters $\boldsymbol{\theta} = (\alpha_1, \alpha_2, \ldots, \alpha_k, \boldsymbol{\beta})$, consisting of k heads*

and initialized such that

$$\boldsymbol{\beta} = \big(c_1\epsilon, , c_2\epsilon, , c_3\epsilon, , \ldots\big),$$

with $c_1 \geqslant c_2 \geqslant c_3 \geqslant \cdots$, *and* $\alpha_i = 0$ *for all* i. *Let the parameters evolve under gradient flow with respect to the DP loss. Then there exists an increasing sequence of times*

$$0 < T_1 < T_2 < \cdots < T_k,$$

with $T_i = O(1/\epsilon)$ *for each* $i \in [k]$, *such that the following holds at time* T_i, *we enter stage* $i+1$ *where the parameter configurations satisfy for* $j \leqslant i$ *and* $k > i$,

$$\boldsymbol{\beta}_j(T_i) > C_1, \alpha_i(T_i) > C_2, \quad \boldsymbol{\beta}_k(T_i), \alpha_k(T_i) = O(\epsilon),$$

for some large $C_1, C_2 > 0$ *that are independent of* ϵ. *Moreover, the ordering of the remaining parameters is preserved, in the sense that* $\boldsymbol{\beta}_i(T_i) > \boldsymbol{\beta}_j(T_i) \quad$ *for all* $j > i$.

A sketch of proof. Without loss of generality, assume the initialization $\boldsymbol{\beta} = (c_1\epsilon, c_2\epsilon, c_3\epsilon, \ldots)$, where ϵ is a very small positive number and $c_1 \gg c_2 \gg c_3 \gg \cdots$ are constants independent of ϵ. Using the first-jump computations, we obtain an time T such that $\boldsymbol{\beta}_1(T) > C$ for some large constant C. During this time, the gradients of the other heads remain $O(\epsilon)$, so their parameters stay close to the origin. Next, applying the subsequent-jump analysis, we can compute a time T_2 such that $\boldsymbol{\beta}_2 > C$. Proceeding in this manner, we can stitch the jumps together to describe the full trajectory where at time T_i, the first i heads have large parameters while the remaining heads remain small. Note that in this section, we have assumed the directional bias and proving directional bias for random initializations is a hard open problem.

3.2. Generalization and experiments

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

Table 1. Quick reference of experimental settings and locations.

Setting	Location
Simplified model	Fig. 3, App. F.0.1
Relaxing (A0) and (A3)	Fig. 6, App. F.0.2
Relaxing (A0-A4)	Fig. 7, App. F.0.3
Transformer with MLP	Fig. 8, App. F.0.4

Experiments. First, we will provide empirical evidence for the training dynamics of simplified two-layer transformer model under assumptions **A0-4** verifying the theoritical claims made in Theorem 3.1 and Theorem 3.2. Here $k = 4$ and we need a transformer with $k - 1 = 3$ heads in the first layer. See Figure 3 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.

Furthermore, we observe the conservation law between $\boldsymbol{\beta}$ and α parameters in the middle plot of Figure 3. 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 3's right most plot. 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 are near-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 which our theory does not capture this is due to finite scale effects. Please refer to Table 1 for an abalation study.

Conclusion. In this work, we have studied the training dynamics of transformers on a simple in-context learning task. We identified a stage-wise learning process, where the model incrementally acquires the ability to match tokens in the context. Each stage is characterized by the activation of a new attention head learning a sparse activation pattern, leading to abrupt decreases in loss. These findings provide a understanding of how transformers learn in-context, shedding light on their training dynamics.

References

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

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

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

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

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

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

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

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

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

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

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

Chen, A., Shwartz-Ziv, R., Cho, K., Leavitt, M. L., and Saphra, N. 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.

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

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

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

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

Garg, S., Tsipras, D., Liang, P. S., and Valiant, G. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

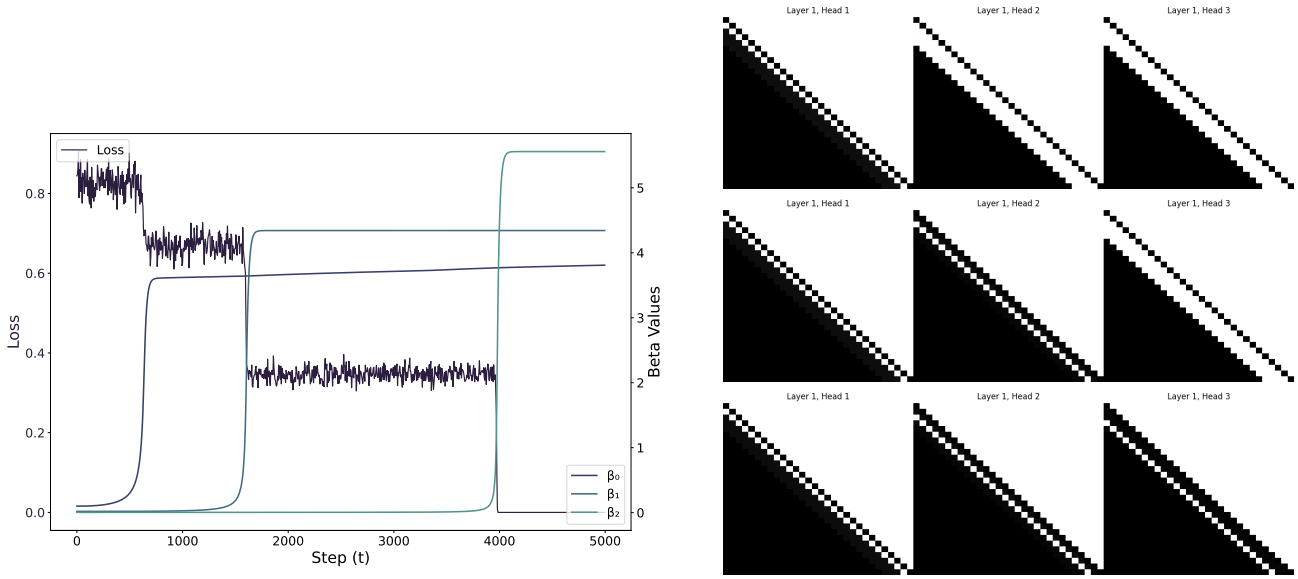
Varre, A. V., Vladarean, M.-L., Pillaud-Vivien, L., and Flammarion, N. On the spectral bias of two-layer linear networks. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=FFdrXkm3Cz.

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

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

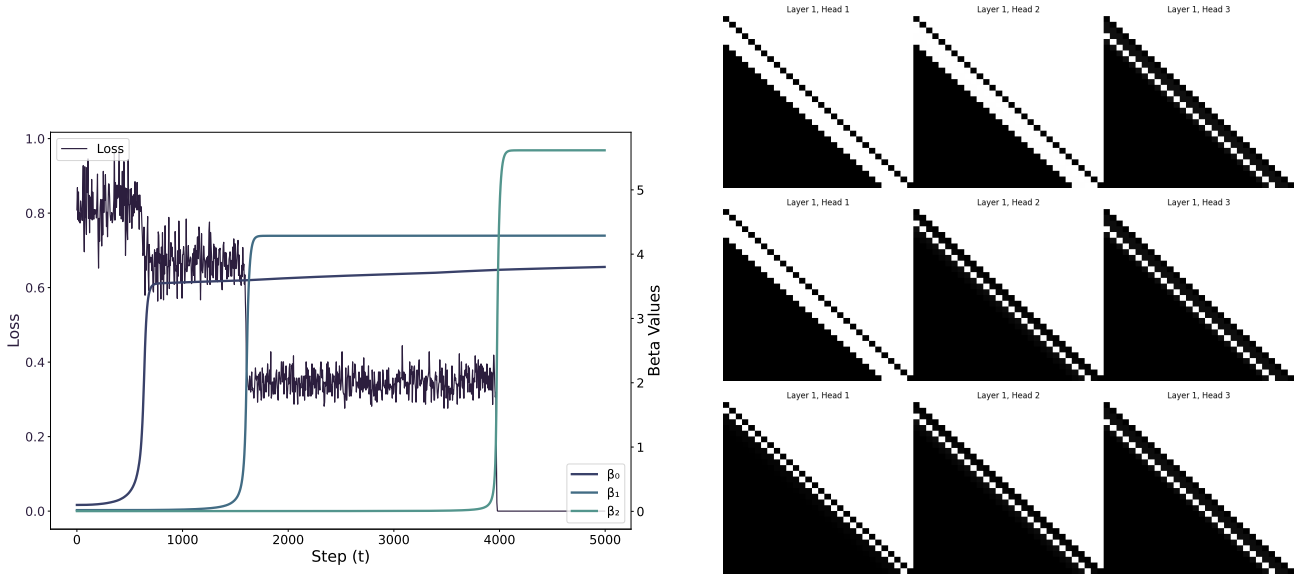
Zhang, Y., Singh, A. K., Latham, P. E., and Saxe, A. Training dynamics of in-context learning in linear attention, 2025. URL https://arxiv.org/abs/2501.16265.

A. Supporting Figures



(a) *(b)*

Figure 4. The left panel is shown in a and the right panel in b. Note that the β values rise simultaneously with the loss, and the attention patterns at times 800, 1750, and 4500 demonstrate incremental learning.



(a) *(b)*

Figure 5. The left panel is shown in a and the right panel in b. Note that the β values rise as the loss drops, and the attention patterns at times 800, 1750, and 4500 demonstrate incremental learning. This is in the opposite order as they are initialized in the opposite order $\beta_3 > \beta_2 > \beta_1$

B. Proofs of Main Results

B.1. Proof of Theorem 3.1

The gradient flow of the parameters is given by

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

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

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

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

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

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

where

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

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

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

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

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

From Lemma C.3 we have

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

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

So the gradient of $\mathbf{w}_i^1$ 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

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

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

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

The dynamics of gradient descent is given by

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

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

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

The final system of ODE is given by

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

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

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

Using this the system of ODE can be written as

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

Note that

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

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

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

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

Now $f(\alpha_1)-\boldsymbol{\beta}_1^2/4$ 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

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

$$
\mathbb{E}\frac{\partial \ell}{\partial \mathbf{w}_i^h} = (1+\gamma)\frac{\boldsymbol{\beta}_h^2\left((e^{\alpha_h}-1)\,\mathbb{1}\{i=h\}+1\right)}{e^{\alpha_h}+k-1}\left(\mathbb{E}\Delta_{h,i}-\frac{e^{\alpha_h}-1}{e^{\alpha_h}+k-1}\mathbb{E}\Delta_{h,h}\right). \tag{9}
$$

where

$$
\begin{aligned}
\Delta_{h,i} &= -\left[\sum_{t=1}^{k!}\mathbf{s}_t\mathbb{1}\{r_t=r\wedge q_{k-h}=\pi_{(t,k-i)}\}-\left(\sum_{j=1}^{k!}\mathbf{s}_j\mathbb{1}\{r_j=r\}\right)\left(\sum_{t=1}^{k!}\mathbf{s}_t\left(\mathbb{1}\{q_{k-h}=\pi_{(t,k-i)}\}\right)\right)\right], \\
s &= \boldsymbol{\sigma}(\mathbf{s}^{'}), \quad \mathbf{s}_t^{'} = \sum_{h=1}^{k}\gamma_h\mathbb{1}\{q_{k-h}=\pi_{(t,k-h)}\}, \\
\gamma_h &= (1+\gamma)\boldsymbol{\beta}_h^2\frac{e^{\alpha_h}-1}{e^{\alpha_h}+k-1}.
\end{aligned}
$$

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

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

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

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

C. Computations of the derivatives of the simplified transformer model

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

$$\mathbf{R}_i^h = \sum_{j=i-k}^{i-1} \boldsymbol{\sigma}(\mathbf{A}^h)_{ij}\, e_{x_j}, \tag{10a}$$

$$\widetilde{\mathbf{s}}_t = \sum_{h=1}^{k} \boldsymbol{\beta}_h^2 \left\langle e_{x_{l-h}}, \widetilde{\mathrm{I}}_k\, \mathbf{R}_t^h \right\rangle, \tag{10b}$$

$$\mathbf{s} = \boldsymbol{\sigma}\left(\widetilde{\mathbf{s}}_{\mathcal{R}}\right) \tag{10c}$$

$$\mathbf{p} = \sum_{t:x_t \in \mathcal{R}} \mathbf{s}_t\, e_{x_t} \tag{10d}$$

$$\ell = 1 - \langle \mathbf{p}, e_r \rangle \tag{10e}$$

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

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

The co-ordinate wise derivative is

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

where r_t is the response corresponding to the 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 even co-ordinate is shifted by a constant. Now taking the derivative with c at $c = 0$ we get

$$\sum_{t=1}^{k!} \ell_t^{'} = 0 \tag{11}$$

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:

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

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

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

Using Lemma D.2 we have

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

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

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

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

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

Now collecting the final expressions we have the following lemma.

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

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

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

The proof of the lemma is given above.

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

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

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

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

$$\frac{\partial \ell}{\partial \boldsymbol{\beta}_h} = 2\boldsymbol{\beta}_h \frac{e^{\alpha_h} - 1}{e^{\alpha_h} + k - 1}(1+\gamma)\Delta_{h,h}, \tag{12}$$

$$\frac{\partial \ell}{\partial \mathbf{w}_i^h} = (1+\gamma)\frac{\boldsymbol{\beta}_h^2 e^{\alpha_h + \delta_h}}{e^{\alpha_h} + k - 1} \left(\Delta_{h,i} - \frac{e^{\alpha_h} - 1}{e^{\alpha_h} + k - 1}\Delta_{h,h} \right) \tag{13}$$

where

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

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

Proof. First let us compute the forward pass

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

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

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

Define

$$\gamma_h = (1+\gamma)\boldsymbol{\beta}_h^2 \frac{e^{\alpha_h}-1}{e^{\alpha_h}+k-1}. \tag{14}$$

Denote

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

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

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

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

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

Using this notation, the gradients can be written as

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

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

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

$$\Delta_{h,i} = - \left[\sum_{t=1}^{k!} \mathbf{s}_t \mathbb{1}\{r_t = r \wedge q_{k-h} = \pi_{(t,k-i)}\} - \left(\sum_{j=1}^{k!} \mathbf{s}_j \mathbb{1}\{r_j = r\} \right) \left(\sum_{t=1}^{k!} \mathbf{s}_t \left(\mathbb{1}\{q_{k-h} = \pi_{(t,k-i)}\} \right) \right) \right]. \tag{15}$$

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 $\mathbf{w}^1 = \alpha_1 \mathrm{e}_h^1 + \delta_1 \mathbf{1}$ and $\boldsymbol{\beta}_h = 0$ for $h \neq 1$. The following quantities are*

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

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

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

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

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

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

For $i = 1$, we have,

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

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

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

For $i \neq 1$,

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

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

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

Now for $i = 1$,

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

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

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

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

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

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

Substituting this back in the expectation term, we get,

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

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

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

Note that

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

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

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

Now for $i = 1$, we have,

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

□

D. Technical Lemmas

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

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

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

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

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

Proof.

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

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

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

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,*

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

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

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

Proof.

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

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

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

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

□

Covariances with respect to parameters.

Lemma D.3 (A positive covariance)**.** *Let $s \in \mathbb{R}^{k!}$ 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

$$\begin{aligned}&\mathrm{Cov}\left(\mathbb{1}\{q_{k-h} = \pi_{(t,k-h)}\}, \mathbb{1}\{r_t = r\}\right)\\ &\qquad = \left[\sum_t \mathbf{s}_t \mathbb{1}\{r_t = r, q_{k-h} = \pi_{(t,k-h)}\} - \left(\sum_{t'} \mathbf{s}_{t'} \mathbb{1}\{r_{t'} = r\}\right)\left(\sum_t \mathbf{s}_t \mathbb{1}\{q_{k-h} = \pi_{(t,k-h)}\}\right)\right].\end{aligned}$$

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

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

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

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

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

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

Pushing the f inside the expectation over f_τ gives

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

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,

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

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

Subtracting them, we get,

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

Note taking the expectation over q, we have,

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

□

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

Lemma D.4. *Let $s \in \mathbb{R}^{k!}$ 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 $f_\tau(q)$,*

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

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

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

Proof. Rewriting the covariance as,

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

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

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

We are going to push the expectation over f_τ inside,

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

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,

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

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

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

Subtracting them, we get,

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

Note taking the expectation over q, we have,

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

□

E. A simple α, β system

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

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

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

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

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

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. Lets take the gradient wrt to the $\widetilde{\mathbf{s}}$,

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

Now lets take a gradient wrt to γ_h,

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

Consolidating the gradient of the loss, we get,

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

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

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

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

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

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

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

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

The gradient flow dynamics of γ_h is given by

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

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

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

Using Lemma D.3, we can write Δ_h as,

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

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,

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

Invoking the following lemma, we get that,

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

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

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

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

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

Note that

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

We can write the dynamics of γ_1 as,

$$\dot{\gamma_1} \geqslant m_3\gamma_1^2e^{-\gamma_1}$$
$$\frac{e^{\gamma_1}}{\gamma_1^2}\mathrm{d}\gamma_1 \geqslant m_3\mathrm{d}t$$
$$\mathrm{dEi} - e^{\gamma_1}/\gamma_1 \geqslant m_3\mathrm{d}t$$
$$\mathrm{Ei}(\gamma) - e^{\gamma_1}/\gamma_1 \geqslant m_3t$$

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 seperate initalizations, 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 $\Delta_h^{'}s$. For For the second jump, we use the similar arguments. Now at time T_1, we have $\boldsymbol{\beta}_1 > C$ and $\alpha_1 > logC$, and for $h > 1$, $\boldsymbol{\beta}_h \sim 1.1c_h\epsilon$ and $\alpha_h = \frac{c_h\epsilon}{10}$.

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

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

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

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

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 k heads, we get the total time.

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

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

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

$$
\frac{\epsilon}{\sqrt{m_1}} \tan\left(\frac{m_0\sqrt{m_1}}{k+1}\epsilon t\right) \leqslant \alpha(t) \leqslant \frac{\epsilon}{\sqrt{m_2}} \tan\left(\epsilon\sqrt{m_2}\, t\right),
$$

for some constants $m_1, m_2 > 0$.

Proof. Consider the following ODE,

$$
\begin{aligned}
\dot{\beta} &= 2\beta \frac{e^{\alpha} - 1}{e^{\alpha} + k} m_0, \\
\dot{\alpha} &= \beta^2 \frac{(k+1)e^{\alpha}}{(e^{\alpha} + k)^2} m_0.
\end{aligned}
$$

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,

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

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,

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

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

$$
\frac{\epsilon}{\sqrt{m_1}} \tan\left(\frac{m_0\sqrt{m_1}}{k+1}\epsilon t\right) \leqslant \alpha(t) \leqslant \frac{\epsilon}{\sqrt{m_2}} \tan\left(\epsilon\sqrt{m_2}\, t\right)
$$

□

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 2. 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 **??**, § 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 4, § F.0.2
Attention only transformer	Further relaxations involving attention-only mechanisms without assumptions **A0–A4**.	Figure 7, § 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 8, § F.0.4
Without β^2 term	Experiments investigating the role of the β^2 parameterization.	Figure 9, Figure 10, § F.0.5

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

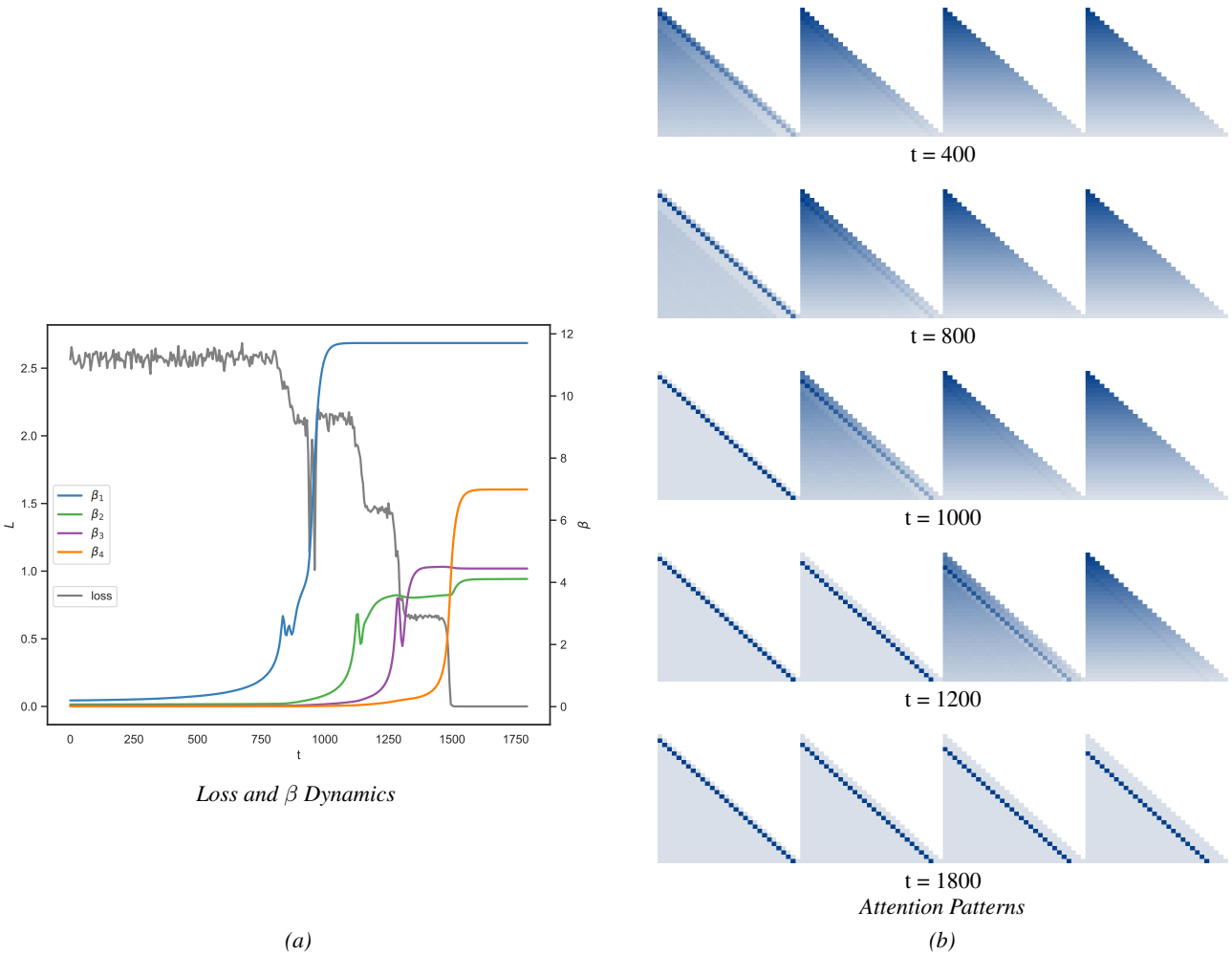
First, we will provide empirical evidence for the training dynamics of simplified two-layer transformer model under assumptions **A0-4** verifying the theoritical claims made in Theorem 3.1 and Theorem 3.2. Here $k = 4$ and we need a transformer with $k - 1 = 3$ heads in the first layer. See Figure **??** 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 trajectory 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 **??**a. 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 **??**b. 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 do are 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 which our theory does not capture this is due 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 4, we experiment with $k = 4$ and due to removing assumption **(A3)**, we use $k = 4$ heads in the first layer. The attention patterns in Figure 6b show that the model still learns to focus on the relevant positions incrementally over time, similar to the behavior observed under the original assumptions. The loss and $\boldsymbol{\beta}$ dynamics in Figure 6a further confirm that the training process remains stagewise and the abrupt change in loss is coupled with increase in the $\boldsymbol{\beta}$ parameters.

Beyond Assumptions A0 and A3



Loss and β Dynamics *Attention Patterns*

(a) *(b)*

Figure 6. **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.

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 defined in Section **??**. The experiments demonstrate that without these assumptions, the model exhibits the incremental learning behaviour. See Figure 7 for a comprehensive analysis of multiple attention heads and their dynamics.

More importantly, in Figure 7c, 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 $\widetilde{\mathrm{I}}_k$ with a positive scaling factor. Note that it is the same $\widetilde{\mathrm{I}}_k \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ where the first $k \times k$ block is given by $\widetilde{\mathrm{I}}_{:k,:k} = \mathrm{I}_k - \gamma \mathbf{1}_k \mathbf{1}_k^\top$ 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 after the second layer

are given by

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

where $\beta_{h,g}$ are some positive constant as seen in the Figure 7c.As $\sum_{g=0}^{k-1} \beta_{g,h} \left[\mathbf{R}_{l-1}^{g}\right]^\top \widetilde{\mathrm{I}}_k \left[\mathbf{R}_t^h\right]$ is approximately constant addition for all t and hence diagonally dominated just by the diagonal terms and this justify 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 $k = 2$ 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 8 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 $k = 2$ and the response tokens appears with a periodicity of 3. At time $t = 400$ amdist a small plateau, the heads do not attend to any particular tokens. At $t = 1100$, second head becomes active and at response token (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 their 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 10 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 11 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.

Attention-Only Transformer Model

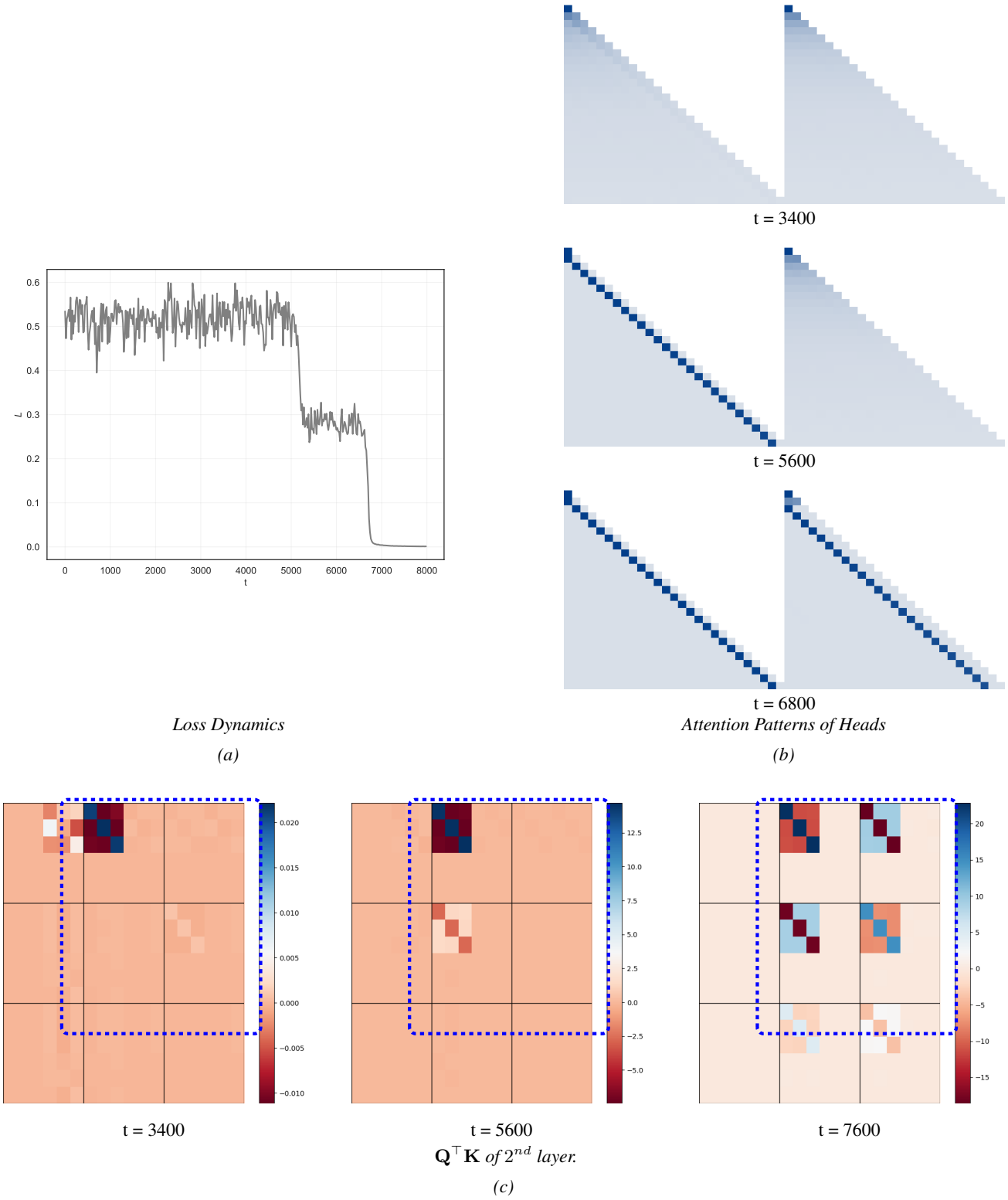
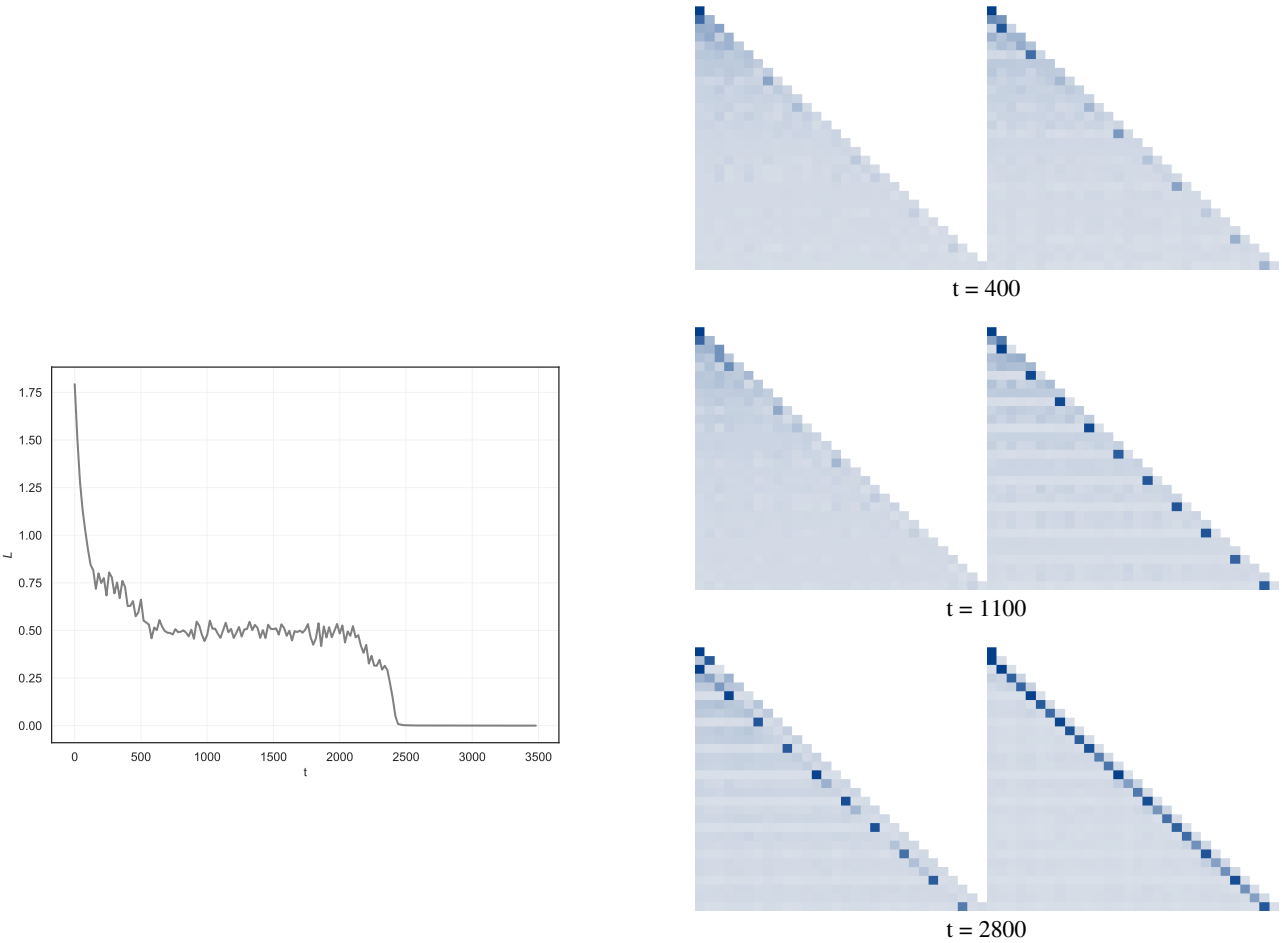


Figure 7. **Dynamics of attention-only transformer model. (a)** Loss dynamics during training. **(b)** Attention patterns for Head Attention 0 at times 3400, 5600, and 6800 exhibiting incremental learning **(c)** We consider the dynamics of $\mathbf{Q}^\top\mathbf{K}$ for the parameters in layer 2, we see that it maintains a block diagonal structure very close to the parameterization with assumption (A2).

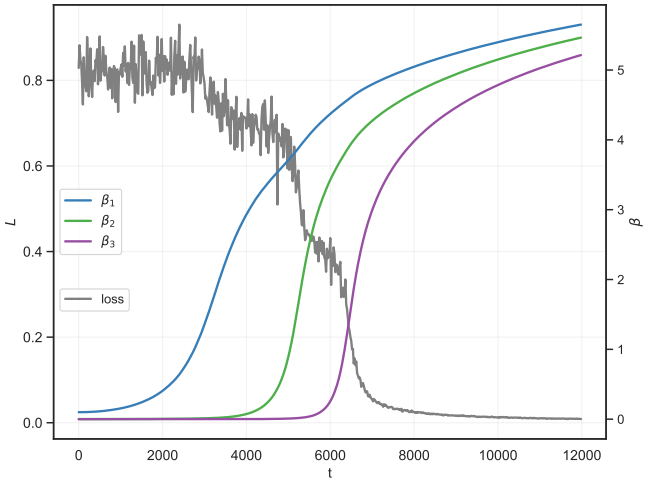
Full Transformer Model



Loss Dynamics *(a)* *Attention Patterns of Heads* *(b)*

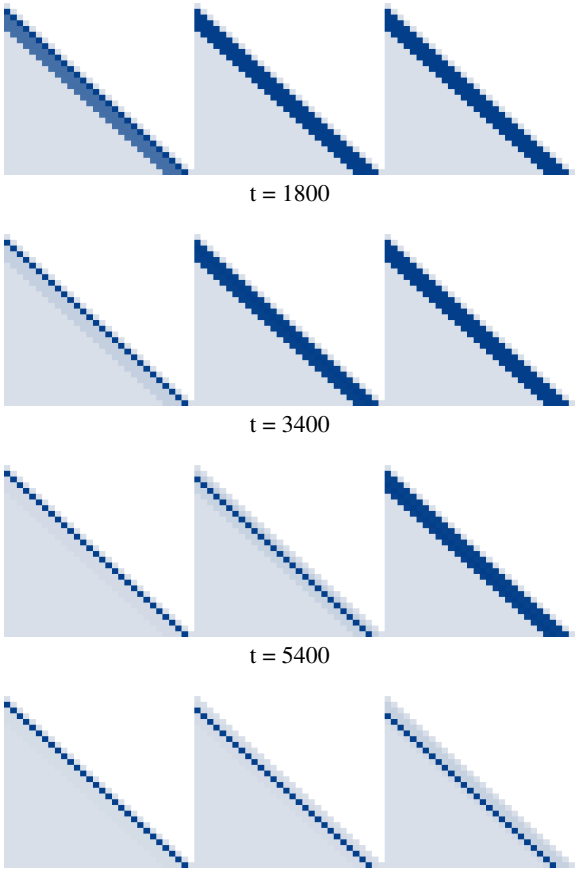
Figure 8. **Dynamics of full transformer model. (a)** Loss dynamics during training on task with order 2, i.e., $k = 2$. **(b)** Attention patterns for Head Attention 0 at times 400, 1100, and 2800 exhibiting incremental learning. The heads at $t = 400$ do not attend to any particular tokens, while in the **second stage** the second head at time $t = 1100$ learns to attend to the previous token. Note that $k = 2$ and every 3rd token is a response token and it learns to attend its previous token. In the **final stage**, the first head learns to attend (-2)-token (at the response tokens) and solves the task.

Transformer Without β^2 parameterization



Loss and β Dynamics

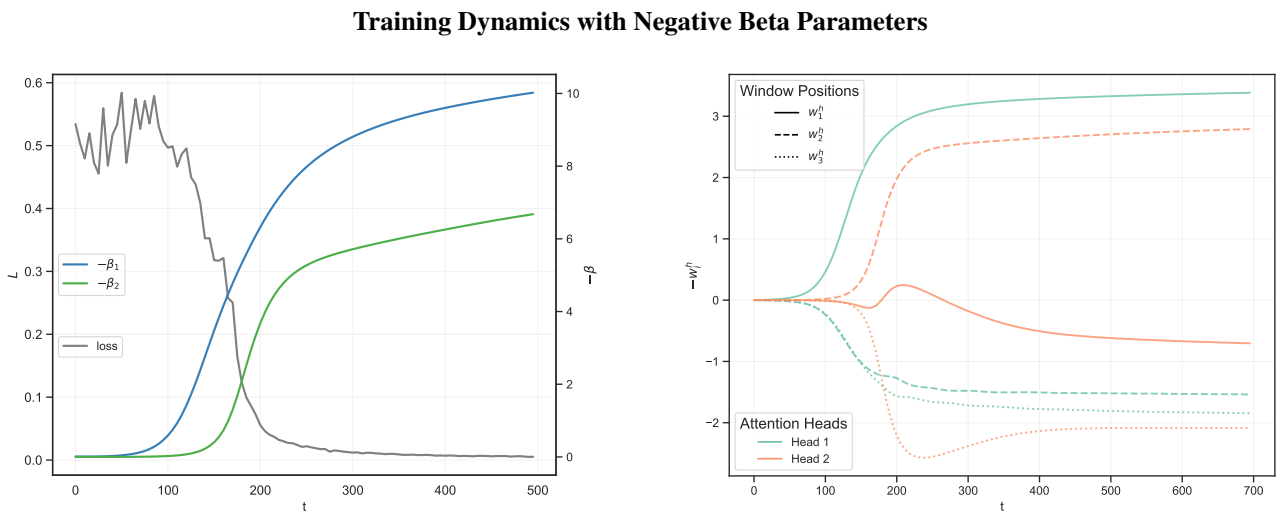
(a)



Attention Patterns

(b)

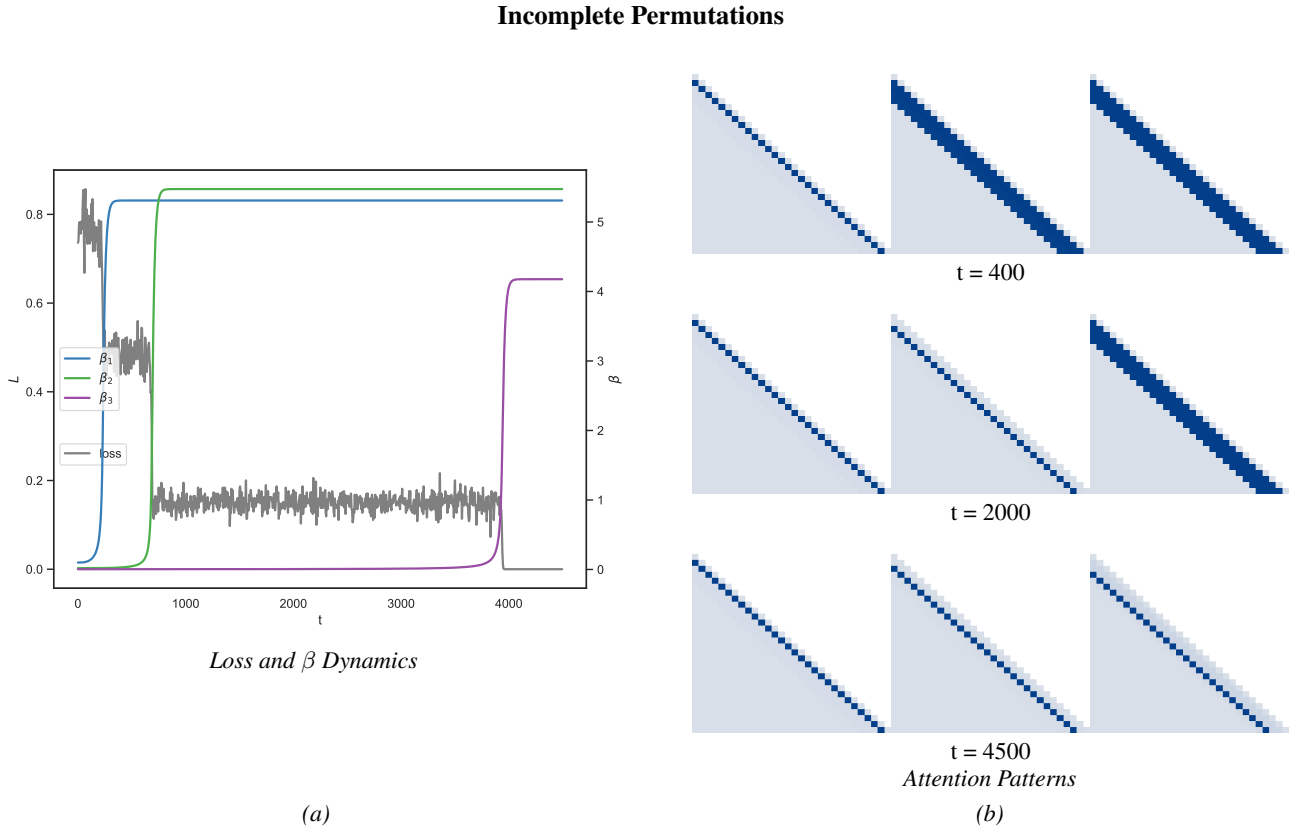
Figure 9. The left panel shows the loss and β parameter dynamics. The right panel displays attention patterns at times 1800, 3400, 5000, and 6800 demonstrating incremental learning without the β^2 term.



(a) Loss and negative beta parameters over time

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

Figure 10. **Training dynamics with negative β parameterization. (a)** Loss dynamics (gray) alongside the negative beta 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.



(a)

(b)

Figure 11. **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 permutation present in the prompt.