Learning In-context n-grams with Transformers: Sub-n-grams Are Near-stationary Points

Aditya Varre [*1] **Gizem Yüce** [*1] **Nicolas Flammarion** [1]

Abstract

In this article, we explore the loss landscape of next-token prediction with transformers. Specifically, we focus on learning in-context n-gram language models with cross-entropy loss using a simplified two-layer transformer. We design a series of transformers that represent k-grams (for $k \leqslant n$) for which the gradient of the population loss approaches zero in the limit of both infinite sequence length and infinite parameter norm. This construction reveals a key property of the loss landscape: *k-grams are stationary points of the population cross-entropy loss*, offering theoretical insights for widely observed empirical phenomena such as stage-wise learning dynamics and emergent phase transitions. These insights are further supported by numerical experiments that illustrate the dynamics of learning n-grams, characterized by jumps between stationary points.

1. Introduction

Transformers (Vaswani et al., 2017) have become central to modern machine learning, due to their capabilities such as in-context learning (ICL) (Brown, 2020)—the ability of models to perform new tasks by leveraging a few examples provided within the context, without the need for parameter updates or retraining.

Recent empirical studies have revealed that the dynamics that result in in-context learning abilities often deviate from a simple monotonic decrease in loss, exhibiting complex behaviors with plateaus such as grokking (Power et al., 2022) and stage-wise transitions (Olsson et al., 2022) where the training process frequently lingers in regions of slow progress before going through a sudden phase transition after which ICL abilities are acquired. Mechanistic interpretability studies suggest that after these plateaus, specific circuits, such as induction heads (Olsson et al., 2022), or syntactic structures (Chen et al., 2024a), are gradually learned. This raises a natural question:

Why does training linger at plateaus before developing such abilities?

The setting of Edelman et al. (2024) provides an avenue to take a closer look at this question with the specialized task of learning in-context n-grams (Shannon, 1948; Chomsky, 1956; Brown et al., 1992). Here, the training unfolds in a hierarchical, phase-wise manner. It begins with the transformer making uniform predictions, progresses through unigram and bigram predictions, and potentially generalizes to higher-order n-grams. Figure 1 illustrates how these training phases overlap with the losses of the sub-n-gram estimators.

Building on this observation, we provide a theoretical foundation for why training lingers at long plateaus—the loss landscape of transformers trained on in-context sequential data exhibit stationary points aligned with sub-hierarchical or sub-syntactic solutions. These points act as intermediate solutions where gradients vanish, causing training to stagnate before transitioning to the next hierarchical level.

Concretely, this paper explores the loss landscape of transformer models trained on in-context n-gram language models to predict the next token. We show a sufficient stationarity condition for the cross-entropy loss that the sub-n-gram constructions satisfy, shedding light on the incremental and phase-wise learning phenomena observed during training for in-context n-grams. Our main contributions can be summarized as follows:

- In Section 3, we provide a sufficient condition for the solutions to be the stationary points of the cross-entropy loss in the next-token prediction task when the derivatives of the model's logits depend solely on a sub-sequence of the input history. This characterization provides a powerful tool for analyzing the structure of the loss landscape for various tasks, including but not limited to n-grams.
- In Section 4, we demonstrate that a set of solutions repre-

[*]Equal contribution [1]Theory of Machine Learning Lab, EPFL, Switzerland. Correspondence to: Aditya Varre <aditya.varre@epfl.ch>, Gizem Yüce <gizem.yuce@epfl.ch>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

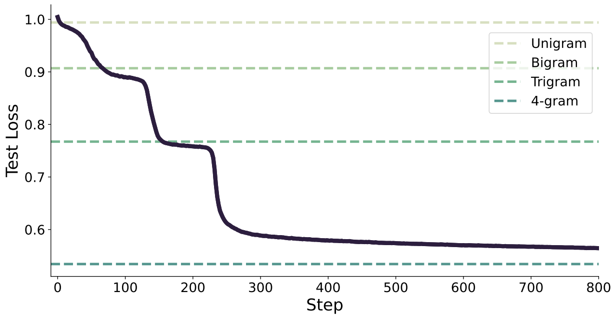


Figure 1: The stage-wise behavior of the test loss during training for the 4-gram language model. The dashed lines represent the cross entropy loss of the k-gram estimators for $k = 1, 2, 3, 4$. The plateaus of the test loss overlap with the losses of k-gram estimators.

senting sub-n-gram estimators are near-stationary points, by verifying that they satisfy the conditions sufficient for the gradient of the population loss to converge to zero presented in Section 3 in the infinite weight and sequence length limit.

- In Section 5, we present empirical evidence that illustrates the structural evolution of these models during training and how transitions between training phases are in alignment with the predictions of our theory.

1.1. Related Work

In-context Learning. To understand in-context learning (ICL) (Brown, 2020), previous works have explored various approaches. One approach is mechanistic interpretability, which has revealed the emergence of circuits called induction heads during ICL (Olsson et al., 2022). Another approach involves studying ICL on specific hypothesis classes to understand how transformers solve these tasks in context. A common feature of the training dynamics in these studies is the presence of plateaus (Chen et al., 2024a; Kim et al., 2024), after which the models acquire certain capabilities. Examples include studying ICL on regression tasks (Garg et al., 2022; Von Oswald et al., 2023; Ahn et al., 2024), boolean functions (Bhattamishra et al., 2023), regular languages (Akyürek et al., 2024), and n-grams.

(In-context) Learning n-gram language models. n-gram language models, or higher-order Markov chains, are effective mathematical models for generating sequential data and capturing certain aspects of natural language (Shannon, 1948; Jelinek, 1998; Jurafsky & Martin, 2024). As a result, many studies have focused on analyzing transformers through the lens of Markov sequences. Svete & Cotterell (2024) examines the representational limits of transformers on n-grams, while Rajaraman et al. (2024) focuses on the in-context counterpart. Makkuva et al. (2024) explores the landscape of transformers on data from binary first-order Markov chains (bigrams) but does not consider the in-context setting. Bietti et al. (2024) investigates the formation of induction heads needed to learn bigrams with specific trigger tokens. Additionally, Nichani et al. (2024) studies the formation of induction heads through gradient descent in learning causal structures. A closely related work to ours is Edelman et al. (2024). They report the stage-wise dynamics of transformers while learning in-context n-grams, and identify that these stages correspond to sub-n-grams. However, their theoretical analysis is limited to one step of gradient descent on binary bigrams, while we characterize the sub-n-grams as near-stationary points for unrestricted n and vocabulary size, explaining the long plateaus once the training reaches a state expressing these solutions. Chen et al. (2024b) study the same in-context n-gram prediction task, but with an architecture that involves a feed-forward network layer that does the token selection. However, they have a threee-stage training procedure and an initialization scheme that ensures different heads attend to different tokens from the start, eliminating the stage-wise learning dynamics we study.

Sequential learning. Fukumizu & Amari (2000) analyzes the plateaus in the loss curve and their relationship to critical points for supervised learning with neural networks. The characterization of dynamics, including jumps between these stationary points, has also been studied for simpler 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 networks (Gissin et al., 2020; Berthier, 2022; Pesme & Flammarion, 2023), linear networks (Saxe et al., 2019; Gidel et al., 2019; Jacot et al., 2021; Varre et al., 2023), ReLU networks (Boursier et al., 2022; Abbe et al., 2023) and transformers with diagonal weight matrices (Boix-Adsera et al., 2023).

Large Language Models(LLMs) as n-grams Nguyen (2024) investigate whether LLM predictions can be approximated by simpler, interpretable statistical rules based on n-gram frequencies, and show that LLMs exhibit curriculum learning during training—starting with simpler n-gram patterns and progressively capturing more complex ones. In a related vein, Zekri et al. (2025) leverages the equivalence between auto-regressive models and n-gram models with long contexts to derive generalization bounds.

2. In-context n-grams and Transformers

In this section, we define the in-context next token prediction loss, the landscape of which we aim to understand for transformers. Next, we introduce the n-gram language models and formally describe the disentangled attention-only transformer architecture.

Notation. Let $[\mathcal{S}] = \{1, 2, \ldots \mathcal{S}\}$ be a finite alphabet for $\mathcal{S} \in \mathbb{N}$. The Kleene star $[\mathcal{S}]^*$ denotes the set of all finite-length sequences whose elements are in $[\mathcal{S}]$. For $l \leqslant k$, $x_l^k = \{x_l, x_{l+1} \dots x_{k-1}, x_k\} \in [\mathcal{S}]^{k-l+1}$ denotes a subsequence. When not specified, $l = 1$. Δ^{k-1} denotes the probability simplex over $\mathbb{R}^k$. For an element x_t in a sequence, its l-history refers to x_{t-l}^{t-1} and its $(-l)$-token refers to x_{t-l}. A language model p is a distribution over $[\mathcal{S}]^*$. For a matrix $r \in \mathbb{R}^{T\times d}$, $r[t] \in \mathbb{R}^d$ denotes the vector representation of it's t^{th} row.

2.1. In-context Next Token Predictions

Consider a language model p and a parametric model $p(\theta, .) : [\mathcal{S}]^* \to \Delta^{\mathcal{S}-1}$, which predicts the probability for the next token. Given a sequence $x^T \in [\mathcal{S}]^T$, the model's performance in predicting the next token is evaluated using the cross-entropy loss (CE):

$$\ell\left(\theta, x^T\right) = \sum_{s\in\mathcal{S}} p(x_{T+1}{=}s|x^T) \log\left(p(\theta, x^T)[s]\right).$$

In the in-context setting, we assume that the ground truth language model is a mixture of multiple language models. Specifically, the language model p_τ is sampled from a prior distribution $\mathcal{P}$. Then, given p_τ, the sequence x^T is sampled accordingly. The in-context population loss is defined as

$$\mathcal{L}(\theta) \coloneqq \mathop{\mathbb{E}}_{p_\tau\sim\mathcal{P}} \mathop{\mathbb{E}}_{x^T\sim p_\tau} \ell\left(\theta, x^T\right). \tag{1}$$

We focus on the in-context n-gram task case where p_τ's are modeled as n-gram language models.

2.2. In-context n-grams task

We start with the definition of n-gram language model and discuss some estimators for this setting.

The n-gram Language Model. A language model p_τ is an n-gram language model if it satisfies the two key assumptions:

(a) Markov property: The conditional probability of a token depends only on the $(n{-}1)$-history rather than the entire history, i.e.,

$$p_\tau\left(x_l|x^{l-1}\right) = p_\tau\left(x_l|x_{l-n+1}^{l-1}\right), \text{ for } l \geqslant n.$$

(b) Time Homogeneity: The transition probabilities are independent of the position in the sequence. Formally, for all $t, l \in [T]$ and sequences $s^n \in [\mathcal{S}]^n$,

$$p_\tau\left(x_l{=}s_n|x_{l-n+1}^{l-1}{=}s^{n-1}\right) = p_\tau\left(x_t{=}s_n|x_{t-n+1}^{t-1}{=}s^{n-1}\right).$$

The assumptions above distinguish n-grams from general language models by limiting the dependency range and assuming uniform transition dynamics over time. Although these assumptions restrict their expressive power, they retain certain key characteristics of natural language, such as causal dependence on the prior context. Under these assumptions, the probability of the sequence x^T can be expressed using the chain rule as

$$p_\tau(x^T) = \prod_{l\in[T]} p_\tau(x_l|x^{l-1}) = \prod_{l\in[T]} p_\tau\left(x_l|x_{l-n+1}^{l-1}\right).$$

Estimators. For the next token prediction task with the in-context n-gram language model, the k-gram estimator is of particular interest.

Definition 2.1 (k-gram estimator). Given a sequence x^T and $i \in [\mathcal{S}]$, the k-gram estimator is defined as

$$\widehat{p}_k\left(x_{T+1}{=}i\middle|x^T\right) = \frac{\sum_{l=k}^{T} \mathbb{1}\{x_{l-k+1}^{l-1}{=}x_{T-k+2}^{T}\}\mathbb{1}\{x_l{=}i\}}{\sum_{l=k}^{T} \mathbb{1}\{x_{l-k+1}^{l-1}{=}x_{T-k+2}^{T}\}}.$$

Intuitively, this estimator checks whether the $(k{-}1)$-histories match and counts the tokens that follow. For $k = 1$ (unigram), it computes the empirical frequency of each token in the sequence. For $k = 2$ (bigram), the estimator computes the empirical frequency of tokens that follow those matching the T^{th} token in the sequence. The n-gram estimator is the "in-context" maximum likelihood estimator (MLE) for the n-gram language model. Moreover, Han et al. (2021) shows that the smoothed version of this estimator achieves the minimax optimal rate for the next-token probability estimation. We include all the k-gram estimators where $k < n$, in the definition of sub-n-grams.

We note that our choice of the n-gram language model and k-gram estimator is primarily for ease of presentation. Our results naturally extend to more general time homogeneous causal dependencies in the sequence beyond n-grams as in Nichani et al. (2024). For example, our results also apply to causal graphs where a token x_t depends only on specific parent tokens, such as x_{t-2} and x_{t-4}. Similarly, while the k-gram estimator matches a contiguous $(k{-}1)$-history by definition, a similar estimator can match non-contiguous histories and count what follows. For instance, let the $(\{1, 3\})$-history of a token x_t refer to (x_{t-1}, x_{t-3}); an estimator could leverage this pattern to predict subsequent tokens in the same way the 3-gram estimator does for the contiguous 2-history. We use the term sub-n-grams also to include such cases, which we discuss further in Appendix F.2. Next, we introduce the attention-only transformer architecture used for this in-context learning task.

2.3. Disentangled Transformer Architecture

Given an input sequence $x^T{\in}[\mathcal{S}]^*$, the transformer first maps each token to a d-dimensional embedding using token-wise semantic and positional embeddings, defined as $E : [\mathcal{S}] \to \mathbb{R}^d$ and $P : [T] \to \mathbb{R}^d$. The core of the transformer is the self-attention mechanism, which allows the model to weigh different parts of the input sequence based on learned similarity scores. Given a sequence embedding $r \in \mathbb{R}^{T\times d}$, self-attention computes a weighted sum of token embeddings as

$$\mathrm{SA}_{(Q,K,V)}(r) = \boldsymbol{\sigma}\left(\left(r\mathbf{Q}^\top \mathbf{K} r^\top\right)\right) r\mathbf{V},$$

where $(\mathbf{Q}, \mathbf{K}, \mathbf{V})$, are the query, key and value matrices of the attention head, and the masked soft-max $\boldsymbol{\sigma}(\cdot)$ is defined as

$$\boldsymbol{\sigma}(x)[i][j] = \begin{cases} \frac{exp(x_{ij})}{\sum_{k\leqslant i} exp(x_{ik})}, & j \leqslant i \\ 0, & otherwise \end{cases}.$$

Traditionally, the outputs of self-attention layers are added to the residual stream. For ease of interpretation and analysis, we instead consider a disentangled architecture in which the outputs of individual heads and layers are concatenated rather than summed. This framework was proposed by Friedman et al. (2023) and formalized by Nichani et al. (2024). The disentangled architecture retains the same representational power as the standard formulation (see Theorem 3 of Nichani et al. (2024)) and is formally defined below.

Definition 2.2 (Disentangled Attention-only Transformer). Let L be the depth, $\{h_\ell\}_{\ell\in[L]}$ be the number of heads per layer, d be the embedding dimension, d_ℓ be the dimension of layer ℓ, d_h be the hidden dimension of the model parameters, and d_{out} be the output dimension. For the h^{th} head in the ℓ^{th} layer, let $\mathbf{Q}_\ell^{(h)}, \mathbf{K}_\ell^{(h)} \in \mathbb{R}^{d_h\times d_\ell}$, $\mathbf{V}_\ell^{(h)} \in \mathbb{R}^{d_\ell\times d_\ell}$, be the query, key, and value matrices of the respective head and layer, let $\mathrm{SA}_\ell^{(h)}(\cdot) = \mathrm{SA}_{\{\mathbf{Q}_\ell^{(h)},\mathbf{K}_\ell^{(h)},\mathbf{V}_\ell^{(h)}\}}(\cdot)$ and let $\mathbf{U} \in \mathbb{R}^{d_{out}\times d_L}$ be the unembedding matrix. Given an input sequence x^T, the disentangled transformer outputs the logits $\mathrm{TF}(\theta)$ for $\theta = \left\{\{\mathbf{Q}_\ell^{(h)}, \mathbf{K}_\ell^{(h)}, \mathbf{V}_\ell^{(h)}\}_{\ell\in[L],h\in[h_\ell]} \cup \mathbf{U}\right\}$, given by,

$$\begin{aligned} r_0 &= [E(x^T),\, P(x^T)] \in \mathbb{R}^{T\times d}, \\ r_\ell &= [r_{\ell-1},\, \mathrm{SA}_\ell^{(1)}(r_{\ell-1}), \dots, \mathrm{SA}_\ell^{(h)}(r_{\ell-1})] \\ \mathrm{TF}(\theta) &= r_L \mathbf{U}^\top. \end{aligned}$$

To simplify the presentation of our theoretical results, we consider a two-layer simplified disentangled transformer with specific design choices. These modifications, detailed below, include orthogonal token embeddings and a fixed value matrix in the second layer.

Embeddings. The token embedding $S : [\mathcal{S}] \to \mathbb{R}^d$ is orthogonal, which means that the set $(s_i)_{i\in[\mathcal{S}]}$ forms an orthogonal family in $\mathbb{R}^d$. Here, s_i denotes the embedding of the token i. For such an embedding to exist, $d \geqslant \mathcal{S}$. For any sequence x^T, the input is encoded as,

$$r_0 = \begin{bmatrix} s_{x_1} & s_{x_2} & \dots & s_{x_T} \end{bmatrix}^\top \in \mathbb{R}^{T\times d}.$$

No explicit positional embeddings are used[1].

First Attention Layer. The first attention layer contains m attention heads, and each attention matrix is parameterized by a single learned matrix $\mathbf{A}_1^{(h)} \in \mathbb{R}^{T\times T}$ and is independent of the input embeddings. The output of each head is therefore given by

$$r_1^{(h)} = \boldsymbol{\sigma}\left(\mathbf{A}_1^{(h)}\right) r_0 \left(\mathbf{V}_1^{(h)}\right)^\top \in \mathbb{R}^{T\times d}. \tag{2}$$

This construction is equivalent to a standard attention head where token embeddings are concatenated with one-hot positional embeddings and the attention only relies on the latter. Thus, the output of the first layer is given by

$$r_1 = \left[r_1^{(0)} \quad r_1^{(1)} \quad \dots \quad r_1^{(m)}\right] \in \mathbb{R}^{T\times(m+1)d}, \tag{3}$$

where $r_1^{(0)} = r_0$ is the skip connection.

Second Attention Layer. The second attention layer contains a single attention head, where the value matrix is fixed to $\mathbf{V}_2^{(1)} = [I_d; 0_d; \dots; 0_d]_{(m+1)d\times d}$ that reads the first block. Consequently , $r_1\mathbf{V}_2^{(1)} = r_1^{(1)} = r_0$ and the output of the second layer is given by

$$r_2 = \boldsymbol{\sigma}\left(r_1\mathbf{Q}_2^\top \mathbf{K}_2 r_1^\top\right) r_0 \in \mathbb{R}^{T\times d}. \tag{4}$$

We note that there is no concatenation to the residual stream in the second layer, which corresponds to not using a residual connection in the standard transformer.

Finally, the unembedding matrix is given by

$$U = \sum_{j=1}^{\mathcal{S}} e_j s_j^\top, \tag{5}$$

where (e_j)'s are canonical basis of $\mathbb{R}^{\mathcal{S}}$. Note that if the token embedding S is the one-hot encoding, then U is the identity matrix. Finally, given a sequence x^T, the probability of the next token estimated by the model, denoted by $p_\theta(x^T)$ is

$$p_\theta(x^T) = U\, r_2[T].$$

$\theta = \{\mathbf{A}_1^{(h)}, \mathbf{V}_1^{(h)}\}_{h=1}^{n-1} \cup \{\mathbf{K}_2, \mathbf{Q}_2\}$ denotes the set of parameters of our simplified disentangled model.

[1]Positional information is implicitly used through the attention matrix in the first layer.

3. A Sufficient Stationary Condition for Population CE on Sequences

In this section, we examine the properties of the gradient of the population next-token cross-entropy loss for sequential data, which is crucial for understanding the training dynamics. The following lemma provides an expression for the partial derivatives of the cross-entropy loss.

Lemma 3.1. *Consider any parametric model $p_\theta(.) : [\mathcal{S}]^T \to \Delta^{\mathcal{S}-1}$ that maps a sequence of states to a probability vector on the states. The derivative of the cross entropy loss function $\mathcal{L}(\theta)$ with respect to a parameter $\theta_i \in \mathbb{R}$ is*

$$\partial_{\theta_i}\mathcal{L}(\theta) = \mathop{\mathbb{E}}_{p_\tau \sim \mathcal{P}} \mathop{\mathbb{E}}_{x^T \sim p_\tau} \left\langle p_\theta(x^T) - p_\tau\left(.\middle|\, x^T\right), \partial_{\theta_i} \log p_\theta(x^T) \right\rangle,$$

where ∂_{θ_i} is the partial derivative with respect to θ_i and $\log(\cdot)$ denotes component-wise logarithm.

Although presented for an input sequence of length T, the above lemma generalizes to sequences of arbitrary length. This result can be viewed as a generalization of Bietti et al. (2024, Lemma 1) and Makkuva et al. (2024).

In Lemma 3.1, two key terms arise in the inner product: (a) the *residue* $p_\theta(x^T) - p_\tau(.|\, x^T)$, which represents the difference between the model's probability estimate and the true probability for the next token, and (b) the *derivative* of the logits of the model's prediction. We show in the next proposition that if this derivative depends only on a certain sub-part of the input sequence, the gradient of the population loss $\mathcal{L}$ can be further simplified.

Proposition 3.2. *For any $\theta_* \in \mathbb{R}^p$ such that $\partial_{\theta=\theta_*} \log p_\theta(x^T) = g\left(p_\tau, x_t^T\right)$, i.e., the derivative is solely a function of the context p_τ and the last $T-t+1$ elements of the sequence x^T, the gradient of the population loss $\mathcal{L}$ can be written as*

$$\nabla\mathcal{L}(\theta_*) = \mathop{\mathbb{E}}_{p_\tau \sim \mathcal{P}} \mathop{\mathbb{E}}_{x^T \sim p_\tau} \left\langle p_{\theta_*}(x^T) - p_\tau\left(.\middle|\, x_t^T\right), g\left(p_\tau, x_t^T\right) \right\rangle.$$

Furthermore, if for such $\theta_* \in \mathbb{R}^p$, the model estimates the conditional probability of the next token $p_\tau(.|\, x_t^T)$, i.e., $p_{\theta_*}(x^T) = p_\tau(.|\, x_t^T)$ almost surely for $p_\tau \sim \mathcal{P}$, then θ_* is a stationary point.

The proposition presents a sufficient condition for a point in the parameter space that computes the true conditional probability based only on the suffix, rather than the entire sequence, to be a stationary point. The blue is used to highlight the difference in history compared to the partial derivative in Lemma 3.1 (x_t^T vs x^T). The result of Proposition 3.2 is not specific to the setting of n-grams and generally holds for the cross-entropy loss in next-token prediction tasks. We leverage this proposition to show how the gradient vanishes for the k-gram estimators, which primarily depend on the k-history. A generalization of this proposition, applying to any subset of tokens rather than just contiguous history, is provided in App. C.2.

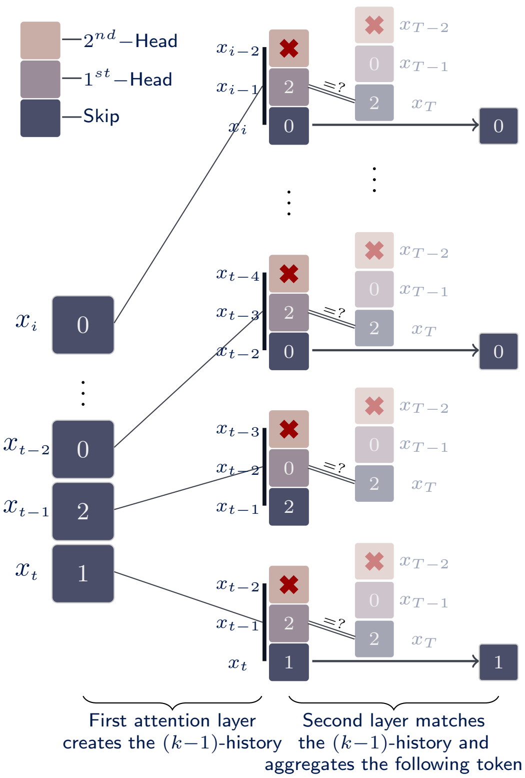


Figure 2: **Transformer defined at θ_*^k is a k-gram estimator.** ✖ represents that 2^{nd} head is deactivated. Hence, the first layer creates the $(1)-$history and the second layer matches with the $(1)-$history of token $T+1$, i.e, $x_T=2$ and attends and averages the tokens at $t, t-2, i$. See the caption of Figure 4 for more details.

4. Theoretical Insights into Stage-wise Dynamics Through the Loss Landscape

This section begins by providing the representation of the sub-n-grams with the simplified disentangled transformer architecture discussed in Subsection 2.3. We then prove that the sub-n-gram constructions are near-stationary points of the loss.

4.1. Representing Sub-n-grams with Simplified Transformer

We construct a disentangled transformer to represent the n-gram estimator in Definition 2.1. We then extend this construction to represent k-gram estimators for any $k \in [n]$. Parameters are often expressed as sums of outer products of orthogonal vectors, emphasizing their role as associative memories in the spirit of Bietti et al. (2024).

Transformer representing n-gram. Consider the simplified transformer model presented in Section 2.3 with $(n-1)$ attention heads.[2] For the h^{th} head in the first layer, we make the following parameter assignments:

$$\mathbf{A}_1^{(h)} = c\sum_{l=h}^{T-1} e_l e_{l-h}^\top + c\sum_{l=0}^{h-1} e_l e_0^\top\,, \qquad \mathbf{V}_1^{(h)} = \sum_{j=1}^{\mathcal{S}} s_j s_j^\top\,.$$

The second layer query and key matrices are assigned as

$$(\mathbf{Q}_2)^\top \mathbf{K}_2 = c\sum_{j=1}^{\mathcal{S}}\sum_{h=1}^{n-1} s_j^{h-1}(s_j^h)^\top, \tag{6}$$

where $c > 0$ is a constant scaling factor and $s_n^h \in \mathbb{R}^{nd}$ defined as $(s_n^h)^\top = \big[\underbrace{0_d^\top\ 0_d^\top\ \dots}_{h \text{ times}}\ s_n^\top\ \underbrace{0_d^\top\ \dots}_{n-h-1 \text{ times}}\big]$.

Intuitively, in the first layer, the h^{th} head attends to the $(-h)$-token. The value matrix acts as an identity map, copying the embedding of the $(-h)$-token into the h^{th} block of the first layer's output, $r_1^{(h)}$. As illustrated in Fig. 4, for any position t, the first layer's output retains the embeddings of the previous $n-1$ tokens (in the limit $c \to \infty$), given by:

$$r_1[t] = \begin{bmatrix} s_{x_t}^\top & s_{x_{t-1}}^\top & \dots & s_{x_{t-n+1}}^\top \end{bmatrix}^\top \in \mathbb{R}^{nd}. \tag{7}$$

Next, the second attention layer compares these histories using the specific structure of the query and key matrices defined in Equation (6). The pre-softmax attention score between the first-layer embeddings of the i^{th} and j^{th} tokens $r_1[i]$ and $r_1[j]$ is computed as:

$$\langle \mathbf{K}_2 r_1[j], (\mathbf{Q}_2) r_1[i]\rangle = c\sum_{l=1}^{n-1} \mathbb{1}\{x_{j-l} = x_{i+1-l}\}.$$

In summary, the second head compares the $(n-1)$-history of the j^{th} and $(i+1)^{\text{th}}$ tokens as detailed in Fig. 4. Using this attention matrix, in the limit as $c \to \infty$, where the softmax converges to the hardmax, p_θ computes the $n-$gram MLE estimator since the i^{th} token attends exclusively to the preceding tokens that exactly match $(n-1)-$history of the $(i+1)^{\text{th}}$ token (see Appendix B.1 for proof).

Transformer representing sub-n-gram. The construction representing an n-gram can be directly adapted to represent a k-gram, where $k < n$. This adaptation relies on the observation that, in the n-gram construction, the h^{th} head is responsible for computing the $(-h)$-token. Therefore, to compute a k-gram counting estimator, it suffices to deactivate the heads responsible for computing the $(-h)$-token for $h \geqslant k$ as illustrated in Fig. 2. The heads in the first layer are thus divided into two categories:

a) **Activated Heads.** The heads which compute $(-h)$-token for $h \leqslant k-1$ are *activated*.

b) **Deactivated Heads.** The heads which compute $(-h)$-token for $h \geqslant k$, are *not activated*, outputting a zero vector to the first layer embedding.

We implement this approach using the following assignments.

$$\mathbf{A}_1^{(h)} = \begin{cases} c\left(\sum_{l=h}^{T-1} e_l e_{l-h}^\top + \sum_{l=0}^{h-1} e_l e_0^\top\right), & \text{for } h \in [k-1], \\ \text{arbitrary} & \text{otherwise}, \end{cases} \tag{8a}$$

$$\mathbf{V}_1^{(h)} = \begin{cases} \sum_{j=1}^{\mathcal{S}} s_j s_j^\top \text{ for } h \in [k-1], \\ 0 \quad \text{otherwise}, \end{cases} \tag{8b}$$

$$(\mathbf{Q}_2)^\top \mathbf{K}_2 = c\sum_{j=1}^{\mathcal{S}}\sum_{h=1}^{k-1} s_j^{h-1}(s_j^h)^\top. \tag{8c}$$

We denote the point given by Equations (8a), (8b), (8c) by $\boldsymbol{\theta}_*^k = \left(\{\mathbf{A}_1^{(h)}, \mathbf{V}_1^{(h)}\}_{h=1}^{n-1} \cup \mathbf{K}_2 \cup \mathbf{Q}_2\right)$ defined precisely in App. (12). The following lemma demonstrates how $\boldsymbol{\theta}_*^k$ implements the k-gram estimator in Definition 2.1 The proof of the lemma is provided in App. B.1, B.2.

Lemma 4.1. *Let $a_{(j,i)}^{(1,h)}$ denote the attention score of the key and query element (i,j) in the h^{th} head of the first layer and let $a_{(i,t)}^{(2)}$ denote the attention score between elements i,t in the second layer. Let $\mathsf{M}_t^k = \left\{i \in [k,t] : \left(x_{i-k+1}^{i-1} = x_{t-k+2}^t\right)\right\}$ be the set of tokens which match the k-history of the next token.*

For the parameters $\boldsymbol{\theta}_*^k$ defined in Eq. (8), in the limit $c \to \infty$, the attention scores of the activated heads in the first layer, i.e., for heads h where $h \leqslant k-1$, are given by,

$$a_{(j,i)}^{(1,h)} = \begin{cases} 1 & \textit{when } i > h \textit{ and } j = i-h, \\ 1 & \textit{when } i \leqslant h \textit{ and } j = 1, \\ 0 & \textit{otherwise} \end{cases}, \tag{9}$$

and the attention scores in the second layer are given by

$$a_{(i,t)}^2 = \begin{cases} \frac{1}{|M_t^k|} & \textit{for } i \in \mathsf{M}_t^k, \\ 0 & \textit{otherwise} \end{cases}. \tag{10}$$

In the first layer, the heads h, for $h \leqslant k-1$ are *activated* and attend to the $(-h)$-token Eq.(9), while the heads h, for $h \geqslant k$ are deactivated through the value matrix and their corresponding attention matrices are chosen arbitrarily. In the second layer, the key-query matrix is similar to that used for n-gram in Equation (6), but with components related to the deactivated heads set to zero. This mechanism ensures that

[2] The heads can be more than $n-1$.

only the $k-$histories of the j^{th} token and the $(i+1)^{\text{th}}$ token are compared for query i and key j. The last token precisely attends to the tokens where the k-history matches Eq.(10) and uniformly averages them, computing the k-gram MLE estimator.

Other possible constructions. We emphasize that the construction presented above is not unique. Multiple heads may compute the $(-l)$-token for some $l \leqslant k-1$, meaning there can be several active heads performing this computation. However, for $l \geqslant k$, no heads should compute this history. The second layer compares these histories. The complete construction is given in Appendix F.

4.2. Sub-n-grams Are Stationary Points

In this subsection, we establish that the k-grams constructed in the previous subsection are first-order stationary points of the cross-entropy loss in the large context asymptotic. This result follows from the general characterization of stationary points for the cross-entropy loss, provided in Section 3.

Theorem 4.1. For the disentangled transformer p_θ, the gradients at θ_*^k are given by

$$\|\partial_{\theta=\theta_*^k} L(\theta)\| \\ = \sqrt{c} \mathop{\mathbb{E}}_{p_\tau\sim\mathcal{P}} \mathop{\mathbb{E}}_{x^T} \mathcal{O}(\left\|p_\tau\left(.\middle|x_{T-k+2}^T\right) - \widehat{p}_k\right\|^2) + \mathcal{O}(t\sqrt{c}e^{-c})$$

The theorem presents the norm of the gradient of the population loss at a point given by Eq. 8, which implements a k-gram estimator in Definition 2.1. The first term $\left\|p_\tau\left(.\middle|x_{T-k+2}^T\right) - \widehat{p}_k\right\|^2$ is the gap between the MLE estimator and the ground truth and decays as $e^{-\Theta(T)}$ (Penev, 1991). In the limit of $c = \Theta(T) \to \infty$, the gradient vanishes. As the result holds asymptotically as $T, c \to \infty$, leading us to term the k-gram estimator as *near*-stationary. Therefore, the theorem reveals a striking property of the landscape of learning transformers with n-gram language models: the points in the parameter space that compute the k-gram estimator are first-order stationary points.

Note that we present the result for any sequence length T, so it also applies to the commonly used loss function, which is averaged over sequences of varying lengths, i.e.,

$$\mathcal{L}_*(\theta) = \frac{1}{T-t'+1}\sum_{t=t'}^{T}\mathcal{L}_t(\theta),$$

where, with a slight abuse of notation, $\mathcal{L}_t$ refers to $\mathcal{L}$ in Equation (1) for sequences of length t. In the averaged loss, the gradients from the longer sequences suffer from the vanishing gradient problem as a result of the above theorem.

4.3. Proof Sketch

The proof consists of two components: (a) demonstrating that the k-gram estimator converges to the conditional probability $p_\tau(\cdot \mid x^{k-1})$, and (b) proving that the gradient of the logits at time step T depends *solely* on the $(k-1)$-history. Once these components are established, Proposition 3.2 can be applied to demonstrate the stationarity of the sub-n-grams. The first component holds asymptotically as $T \to \infty$, leveraging the properties of n-gram language models (see Lemma H.6 in App. for the formal argument).

However, it is not immediately obvious how the gradient of the logits depends exclusively on the $(k-1)$-history of token t. This dependence needs to account for the parameters of the transformer layers, which include the attention matrices $\mathbf{A}_1^{(h)}$ and value matrices $\mathbf{V}_1^{(h)}$ for each head h in the first layer, as well as the key and query matrices $\mathbf{K}_2, \mathbf{Q}_2$ in the second layer. The model output writes

$$p_{\boldsymbol{\theta}}(x^t) = \sum_{i=1}^{t} a_{(i,t)}^{(2)} e_{x_i}, \tag{11}$$

where $a_{(i,t)}^{(2)}$ are the attention scores in the second layer $a_{(i,t)}^{(2)}$ for key i and query t are given by Eq. (13) in Appendix. The derivative with respect to any parameter $\theta_{(1)}, \theta_{(2)}$ in the first and second layer can be written as

$$\frac{\partial p_{\boldsymbol{\theta}}(x^t)}{\partial\theta_{(1)}}, \frac{\partial p_{\boldsymbol{\theta}}(x^t)}{\partial\theta_{(2)}} = \sum_{i=1}^{t}\frac{\partial a_{(i,t)}^{(2)}}{\partial r_1[i]}\frac{\partial r_1[i]}{\partial\theta_{(1)}}e_{x_i}, \sum_{i=1}^{t}\frac{\partial a_{(i,t)}^{(2)}}{\partial\theta_{(2)}}e_{x_i}.$$

Since the softmax function exhibits a *self-bounding* property, i.e., its derivative is bounded by the softmax score itself, we obtain $\left(a_{(i,t)}^{(2)}\right)' \propto a_{(i,t)}^{(2)}$. As we move forward, this fact plays a key role in our analysis.

Next, we define a subset of tokens, $\mathsf{M}_t \subseteq [t]$, consisting of tokens whose k-history matches that of the $(t+1)^{\text{th}}$ token. Formally, $\mathsf{M}_t = \left\{i \in [t] : \mathbb{1}\left(x_{i-k+1}^{i-1} = x_{t-k+2}^{t}\right)\right\}$. At the k-gram estimator parameterized by $\boldsymbol{\theta}_*^k$, the attention scores are given by

$$a_{(i,t)}^{(2)} \approx \begin{cases} \frac{1}{|\mathsf{M}_t^k|}, & \text{for } i \in \mathsf{M}_t^k, \\ 0 & \text{o.w.} \end{cases}.$$

Using this approximation along with the self-bounding property of the softmax function, we simplify the derivative as

$$\left.\frac{\partial p_{\boldsymbol{\theta}}(x^t)}{\partial\theta_{(2)}}, \frac{\partial p_{\boldsymbol{\theta}}(x^t)}{\partial\theta_{(1)}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_*^k} \approx \sum_{i\in\mathsf{M}_t}\frac{\partial a_{(i,t)}^{(2)}}{\partial\theta_{(2)}}e_{x_i}, \sum_{i\in\mathsf{M}_t}\frac{\partial a_{(i,t)}^{(2)}}{\partial r_1[i]}\dots$$

This expression reveals that the derivative is supported only on tokens in M_t. A key result we establish next is that the derivatives of $a_{(i,t)}^{(2)}, r_1[i]$ depend exclusively on the $(k-1)-$history of token i. Several reinforcing factors contribute to this result. The structure of the *key* and *value* matrices in the second layer ensures that the deactivated heads remain deactivated. This deactivation, in turn, ensures that the embeddings after the first layer only contain the embeddings of the $(k-1)$-history. Finally, for $i \in \mathsf{M}_t$, its $(k-1)$-history is identical to that of token $t+1$. Together, these factors imply that the derivatives of the second layer are solely a function of $(k-1)$-history of token $t+1$. Refer to App. B.2 for the complete proof.

4.4. Extensions and Perspectives

Beyond contiguous history. As discussed previously in section 2.2, an n-gram language model allows for estimators beyond k-gram estimators. For example, consider the bigram estimator which matches the (-1)-token; alternative estimators can instead match the $(-i)$-token for $1 < i \leqslant n-1$ (i.e., $\{x_{j-i} = x_{T+1-i}\}$). Our construction and results thus far have focused only on histories that are contiguous suffixes. While our construction is homogeneous for any T, relaxing the model to allow parameters to explicitly depend on T enables us to extend our framework. Specifically, we can show that there are stationary points for a specific T, corresponding to estimators that match arbitrary subset of the $(n-1)$-history, rather than only contiguous subsets. For details, refer to Appendix F.2.

Towards general transformer architecture. In this section, we have designed stationary points for a simplified transformer architecture. However, the same techniques and methodology extend naturally to a general transformer architecture. First, positional encodings, which were not explicitly used in the simplified model, can be incorporated with a simple extension using one-hot positional encoding. The concatenation of attention head output and residual connections can be replaced with a simple addition of embeddings. Additionally, the value matrix in the second layer can be incorporated. However, in this case, the transformer's output would no longer be in $\Delta^{\mathcal{S}-1}$, requiring normalization via softmax at the end. This restriction on the value matrix can also be alleviated by using an MLP layer to approximate the logarithm (see Appendix F.1).

Multiple Heads and Causal Structures. Transformers learns specialized attention heads that attend to syntactic neighbors of tokens (Voita et al., 2019). In our case, each head specifically learns to attend to the $(-k)-$token. Consider the behavior of a gradient-based method at a sub-n-gram, say $\boldsymbol{\theta}_*^k$. As it is a stationary point, training remains at this point for a prolonged period, leading to a plateau in the training curve. However, as training progresses, the model eventually escapes due to landscape curvature or stochastic noise, allowing it to learn a new syntactic structure— attending to $(-k)$-token—before reaching the next stationary point. This phenomenon is general and has been empirically reported—emergence of a syntactic structure (Chen et al., 2024a; Wei et al., 2022), phase transitions (Olsson et al., 2022; Edelman et al., 2024). By carefully analyzing the loss landscape of a relevant yet simple in-context task, we demonstrate that the stationary points correspond to underlying syntactic structures. Consequently, our work provides insights into why these syntactic structures emerge following extended plateaus.

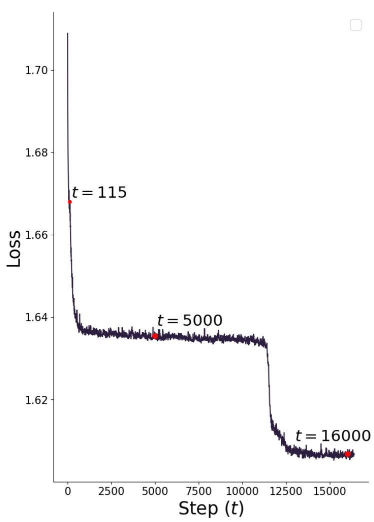
Limitations. Note that our results only hold at the asymptotic limit of norm(c) and sequence length(T). The dependence on the norm is mild, as the gradient decays exponentially as e^{-c} for finite c. However, there is an inherent difficulty in moving beyond the assumption of infinite sequence length. Existing works analyzing how transformers *represent* in-context n-gram like sequences, such as Rajaraman et al. (2024) and Nichani et al. (2024), also rely on the infinite sequence limit. Consequently, it remains unclear which estimator transformers learn for Markov chains at finite sequence lengths, even at the end of training. This unknown makes it particularly challenging to determine what transformers learn during intermediate stages of training

5. Experimental Evaluation

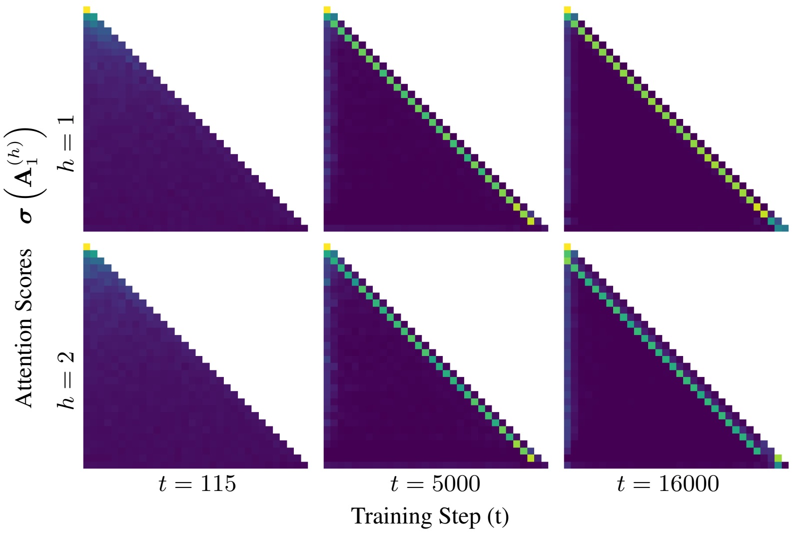
In this section, we perform experiments on the disentangled transformer introduced in the previous section to examine the stage-wise learning behavior and analyze the different solutions the transformer learns during different stages of training. The code is available at `https://github.com/tml-epfl/sub-n-grams-are-stationary`.

Experimental Setup. We select a vocabulary of size $\mathcal{S} = 5$. The input sequences have a length of $T = 32$ and are sampled in-context from a tri-gram language model, i.e., $n = 3$. The transition matrix is sampled from a uniform Dirichlet prior Dir($\alpha\mathbf{1}$) with $\alpha = .5$. We train a two-layer simplified transformer with 2 heads in the first layer. The token embedding dimension is set to $d = 5$, and we use one-hot embeddings for the input tokens. The transformer is trained with Adam without weight decay for 2^{14} iterations, with a constant learning rate of 0.01 and a batchsize of 128. The test loss is evaluated over 2^{16} test sequences.

Discussion. To accurately predict the next token, the model needs to attend to the previous $n - 1 = 2$ tokens in the sequence. Figure 3a shows that learning occurs in distinct phases, where the model remains in a plateau for an extended period before quickly jumping to the next one. In Figure 3b, we illustrate the evolution of the attention maps from both heads in the first layer at various plateaus during training, providing a fine-grained view of the structure of the model at these stages. Initially, the attention maps are uniform, and the model does not consider token history in its predictions. Upon reaching the first plateau, both attention heads focus on the previous token (the (-1)-token), and



(a)



(b)

Figure 3: The evolution of the attention heads in the first layer during training. (a) Progression of the test loss during training. The highlighted points are the iterations on the plateaus for which we demonstrate the attention matrices. (b) The evolution of attention scores of the heads of the simplified transformer architecture during training representing the tokens it is attending. First, both of the attention heads attend to all the previous tokens uniformly. At the second plateau, they both attend to the previous token. Finally, as the model escapes this plateau, the second attention head learns to attend to (-2)-token at the end of training.

the model behaves like a bigram estimator at this stationary point. In the later stages of training, the second attention head learns to shift its focus to the (-2)-token, transforming the model into a trigram estimator. The same phenomenon holds for the general attention-only transformers, see Fig 5.

6. Conclusion

In this work, we investigate the problem of learning in-context n-grams with transformers, specifically focusing on a simplified yet insightful setting to gain a deeper understanding of complex large language models. We have constructed a set of solutions for the parameters of the simplified disentangled transformer architecture that represents sub-n-gram solutions. Then, proved that these solutions correspond to near-stationary points of the population cross-entropy loss. This analysis sheds light on why training lingers at long plateaus corresponding to sub-syntactical solutions before new abilities emerge, i.e., transformer predictions shift to more complex solutions.

However, our analysis explicitly relies on the loss over the entire population, which leaves open the question of what occurs with finite sample complexities. The results presented in this paper are asymptotic in nature, applying to large sequence lengths, raising the natural question of how transformers behave with sequences of finite length. Additionally, it remains to be seen whether other estimators implemented by transformers are optimal in these cases. Another open question is the impact of explicit weight regularization, such as weight decay—-commonly used in training—-on the model's behavior. Lastly, since this paper focuses on the loss landscape rather than the dynamics of gradient-based methods, a characterization of the time evolution of (stochastic) gradient descent, as explored by Nichani et al. (2024), presents an exciting direction for future work.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

Acknowledgements

This work was supported by the Swiss National Science Foundation (Grant No. 212111) and an unrestricted gift from Google. A.V. acknowledges funding from a Swiss Data Science Center Fellowship. The authors extend their gratitude to Adway Girish for his valuable feedback on the manuscript and to anonymous reviewers for their insightful comments, which significantly improved the final version.

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.

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, P., Dellapietra, V., Souza, P., Lai, J., and Mercer, R. Class-based n-gram models of natural language. *Computational Linguistics*, 18:467–479, 01 1992.

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.

Chomsky, N. Three models for the description of language. *Transactions on Information Theory*, 2(3):113–124, 1956. doi: 10.1109/TIT.1956.1056813.

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.

Han, Y., Jana, S., and Wu, Y. Optimal prediction of markov chains with and without spectral gap. *Advances in Neural Information Processing Systems*, 34:11233–11246, 2021.

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.

Jelinek, F. *Statistical methods for speech recognition*. MIT Press, Cambridge, MA, USA, 1998. ISBN 0262100665.

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 with Language Models*. 3rd edition, 2024. URL https://web.stanford.edu/~jurafsky/slp3/. Online manuscript released August 20, 2024.

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.

Nguyen, T. Understanding transformers via n-gram statistics, 2024. URL https://arxiv.org/abs/2407.12034.

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

Odonnat, A., Bouaziz, W., and Cabannes, V. Clustering head: A visual case study of the training dynamics in transformers, 2025. URL https://arxiv.org/abs/2410.24050.

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.

Penev, S. Efficient estimation of the stationary distribution for exponentially ergodic markov chains. *Journal of Statistical Planning and Inference*, 27(1):105–123, 1991.

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

Power, A., Burda, Y., Edwards, H., Babuschkin, I., and Misra, V. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

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. 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.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., and Gomez, A. N. L. u. kaiser, and i. polosukhin,"attention is all you need,". *Advances in neural information processing systems*, 30:5998–6008, 2017.

Voita, E., Talbot, D., Moiseev, F., Sennrich, R., and Titov, I. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*, 2019.

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.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

Zekri, O., Odonnat, A., Benechehab, A., Bleistein, L., Boullé, N., and Redko, I. Large language models as markov chains, 2025. URL https://arxiv.org/abs/2410.02724.

A. Organization of the Supplementary Material

A.1. Links to Materials Referenced in The Main Text.

First, we present an index of the supporting material referenced in the main text.

- The proofs of Lemma 3.1 and Proposition 3.2 are provided in Section C and the discussion on the stationarity condition for subsequences beyond suffixes is provided at Remark C.2.
- The proof of the construction of k-grams is given in Subsection B.1 and the proof of Theorem 4.1 is provided in Subsection B.2.
- The discussion on the possible alternate representation for the stationary distribution is provided in Section F, and the extension for a general transformer architecture is given in Subsection F.1. Stationary points conditioned on subsequences that are not suffixes for a fixed T are further discussed in the Subsection F.2

A.2. Outline of the Supplementary Material.

- Section B, provides the construction of the k-grams and the proof of the stationarity of the construction (in Subsection B.1 and Subsection B.2 respectively).
- Section C provides the proofs of the results on the gradient of the cross-entropy and sufficient stationary conditions, i.e., of Lemma 3.1 and Proposition 3.2.
- Section D provides a technical lemma related to computing the gradient for two-layer simplified transformer.
- In Section E, a technical lemma related to the k-gram representations is provided.
- In Section F, we discuss the extensions of the results to a general transformer architecture in Subsection F.1 and the stationary points conditioned on subsequences that are not suffixes in Subsection F.2.
- In Section G, we present the derivatives of a single layer self-attention map.

A.3. Notation and Definitions

Notations. We use $\otimes$ to denote the Kronecker product. We use vec operator for flattening the matrix to a vector.

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

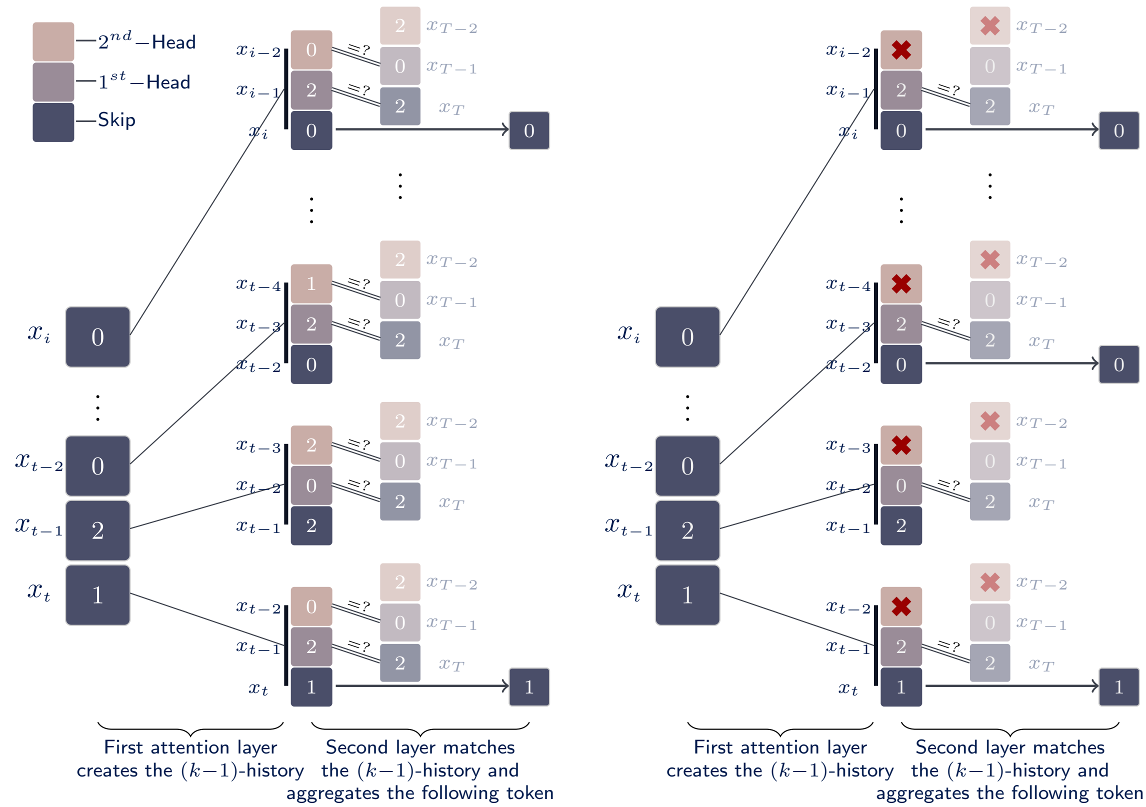
Definition A.2. Define $\boldsymbol{\theta}_*^k = \left\{\mathbf{A}_1^{(h)}, \mathbf{V}_1^{(h)}\right\}_{h\in[n-1]} \cup \{\mathbf{K}_2, \mathbf{Q}_2\}$ as the set of parameters given by the following expressions

$$(\mathbf{Q}_2)^\top = \sqrt{c}\sum_{j=1}^{\mathcal{S}}\sum_{h=1}^{k-1} s_j^{h-1}(s_j^h)^\top, \tag{12a}$$

$$\mathbf{K}_2 = \sqrt{c}\sum_{j=1}^{\mathcal{S}}\sum_{h=1}^{k-1} s_j^{h}(s_j^h)^\top, \tag{12b}$$

$$\mathbf{A}_1^{(h)} = c\sum_{l=h}^{T-1} e_l e_{l-h}^\top + c\sum_{l=0}^{h-1} e_l e_0^\top, \tag{12c}$$

$$\mathbf{V}_1^{(h)} = \begin{cases} \sum_{j=1}^{\mathcal{S}} s_j s_j^\top \text{ for } h\in[k-1], \\ 0 \quad \text{o.w.} \end{cases}. \tag{12d}$$



(a) (b)

Figure 4: **Transformer representing a n-gram and k-gram estimator.** (a) The task we consider task is learning in-context tri-grams (n=3). Here, we illustrate how the transformer given by $\boldsymbol{\theta}_*^n$ constructed in section 4.1 operates on the sequence $(\ldots, 2, x_i{=}0, \ldots, 2, 0, 2, x_t{=}1, \ldots 0, x_T{=}2)$ and computes a tri-gram(k=3) estimator. Skip, 1^{st}-Head, 2^{nd}-Head denotes the outputs of the skip connection, 1^{st} and the 2^{nd} head. The first layer creates the $(2)-$history and the second layer compares it with the $(2)-$history of token $T{+}1$, i.e, $x_T = 2$. The second layer compares these histories and attends to tokens at t, i and averages the tokens at that position. (b) ✖ represents that 2^{nd} head is deactivated. Hence, the first layer creates the $(1)-$history and the second layer matches with the $(1)-$history of token $T{+}1$, i.e, $x_T{=}2$ and attends and averages the tokens at $t, t{-}2, i$.

B. Construction and the Stationarity of the $k-$gram solutions

B.1. Representing k-grams with Simplified Transformer

Before presenting the proofs, we give an alternate form of the forward pass of the simplified transformer. First, we express the embeddings after the first layer as follows:

$$r_1[i] = \mathcal{W}_o^{(0)} r_0[i] + \sum_{h=1}^{n-1} \mathcal{W}_o^{(h)} \sum_{j=1}^{i} a_{(j,i)}^{(1,h)} \, \mathbf{V}_1^{(h)} r_0[j],$$

where the attention scores in the first layer $a_{(j,i)}^{(1,h)}$ for key j and query i for head h and layer 1 and the matrices $\mathcal{W}_o^{(h)}$ which are used for concatenation are given by,

$$a_{(j,i)}^{(1,h)} = \frac{\exp \mathbf{A}_1^{(h)}[i,j]}{\sum_{l=1}^{i} \exp \mathbf{A}_1^{(h)}[i,l]},$$

$$\mathcal{W}_o^{(h)} = \sum_{j=1}^{\mathcal{S}} s_j^h s_j^\top.$$

For the t^{th} token, the output after the second layer of transformer writes,

$$r_2[t] = \sum_{i=1}^{t} a^{(2)}_{(i,t)} r_0[i]$$

where the attention scores in the second layer $a^{(2)}_{(i,t)}$ for key i and query t are given by,

$$a^{(2)}_{(i,t)} = \frac{\exp\langle \mathbf{K}_2 r_1[i], \mathbf{Q}_2 r_1[t]\rangle}{\sum_{j=1}^{t} \exp\langle \mathbf{K}_2 r_1[j], \mathbf{Q}_2 r_1[t]\rangle}. \tag{13}$$

The final output probabilities are given by

$$p_{\boldsymbol{\theta}}(x^t) = U r_2[t] = \sum_{i=1}^{t} a^{(2)}_{(i,t)} U r_0[i] = \sum_{i=1}^{t} a^{(2)}_{(i,t)} e_{x_i}.$$

We now provide the proof for the k-gram MLE constructions. To support the proofs, we define a subset of tokens, $\mathsf{M}_t \subseteq [t]$. This subset consists of tokens whose k-history matches the k-history of the $(t+1)^{\text{th}}$ token. Formally, it is defined as follows:

$$\mathsf{M}^k_t = \left\{ i : \mathbb{1}\left(x^{i-1}_{i-k+1} = x^{t}_{t-k+2}\right)\right\}$$

Now we provide two lemmas one for each layer of the transformer, which denotes what tokens they attend to. Intuitively, the h^{th} head in the first layer attends to the $(-h)$-th token, and the second layer attends to the tokens whose k-history matches the k-history of the $(t+1)^{\text{th}}$ token. The $\mathcal{O}$ notation hides the terms polynomial in c and the sequence length T.

Lemma B.1. *[First-Layer] For the first layer given by the parameter defined in Eq. (12),*

(a) Let $a^{(1,h)}_{(j,i)}$ denote the attention score of head h of layer 1 where (i,j) denote the key and query,

$$a^{(1,h)}_{(j,i)} = \begin{cases} 1 - \mathcal{O}(ie^{-c}) \text{ when } i \geqslant h \text{ and } j = i-h, \\ 1 - \mathcal{O}(ie^{-c}) \text{ when } i < h \text{ and } j = 0, \\ \mathcal{O}(e^{-c}) \text{ o.w.}. \end{cases} \tag{14}$$

(b) The embeddings of the first layer,

$$r_1[i] = \begin{cases} s^0_{x_i} + \sum_{h=1}^{k-1} s^h_{x_{i-h}} + \mathcal{O}(ie^{-c}) \cdot \mathbf{1} \quad \text{for } i \geqslant k-1, \\ s^0_{x_i} + \sum_{h=1}^{i} s^h_{x_{i-h}} + \sum_{h=i+1}^{k-1} s^h_{x_0} + \mathcal{O}(ie^{-c}) \cdot \mathbf{1} \quad \text{for } i < k-1. \end{cases} \tag{15}$$

Lemma B.2. *[Second-Layer] With the construction given in Def. A.2, the attention scores after the second layer,*

$$a^{(2)}_{(i,t)} = \begin{cases} \frac{1}{|\mathsf{M}^k_t|} - \frac{\mathcal{O}(te^{-c})}{|\mathsf{M}^k_t|^2} \text{ for } i \in \mathsf{M}_t, \\ \frac{\mathcal{O}(e^{-c})}{|\mathsf{M}^k_t|} \text{ o.w.} \end{cases} \tag{16}$$

.

For the proof of the above lemmas, see Section E.

Proof of the construction for k-gram MLE. We give the proof of construction for any k and when $k = n$ we get the n-gram estimator. The final output probabilities writes

$$p_{\boldsymbol{\theta}}(x^t) = \sum_{i=1}^{t} a^{(2)}_{(i,t)} e_{x_i}.$$

In the limit $c \to \infty$, the attention scores from the second layer from Lemma B.2 is given by,

$$a^{(2)}_{(i,t)} = \begin{cases} \frac{1}{|\mathsf{M}_t^k|}, & \text{for } i \in \mathsf{M}_t^k, \\ 0 & \text{o.w.} \end{cases} .$$

Using this the output probabilities are only supported by the tokens in M_t^k and is given by,

$$p_{\boldsymbol{\theta}}(x^t) = \frac{1}{|\mathsf{M}_t^k|} \sum_{i \in \mathsf{M}_t^k} e_{x_i}.$$

Now

$$p_{\boldsymbol{\theta}}(x^t)[s] = \frac{1}{|\mathsf{M}_t^k|} \sum_i \mathbb{1}\{i \in \mathsf{M}_t^k\} \mathbb{1}\{x_i = s\}.$$

which exactly matches the k-gram MLE estimator in Definition 2.1.

B.2. Proof of Stationary Points with Simplified Transformer

Theorem 4.1. For the disentangled transformer p_θ, the gradients at θ_*^k are given by

$$\begin{aligned} &\|\partial_{\theta=\theta_*^k} L(\theta)\| \\ &= \sqrt{c} \mathop{\mathbb{E}}_{p_\tau \sim \mathcal{P}} \mathop{\mathbb{E}}_{x^T} \mathcal{O}(\left\|p_\tau\left(.|x^T_{T-k+2}\right) - \widehat{p}_k\right\|^2) + \mathcal{O}(t\sqrt{c}e^{-c}) \end{aligned}$$

Proof The proof is an application of Lemma D.1 for the transformer parameters that compute the k-gram MLE estimator $\boldsymbol{\theta}_*^k$ defined in A.2.

Parameters of the second layer. From Lemma D.1, the derivatives of $p_{\boldsymbol{\theta}}$ with respect to the second layer are

$$\begin{aligned} \frac{\partial p_{\boldsymbol{\theta}}}{\partial \mathbf{K}_2} &= \sum_{i=1}^t a^{(2)}_{(i,t)} (e_{x_i}) \otimes \operatorname{vec}\left(\mathbf{Q}_2 r_1[t](r_1[i] - \bar{r}_t^{(1)})^\top\right)^\top, \\ \frac{\partial p_{\boldsymbol{\theta}}}{\partial \mathbf{Q}_2} &= \sum_{i=1}^t a^{(2)}_{(i,t)} (e_{x_i}) \otimes \operatorname{vec}\left(\mathbf{K}_2 (r_1[i] - \bar{r}_t^{(1)})(r_1[t])^\top\right)^\top. \end{aligned}$$

Before computing these quantities at $\boldsymbol{\theta}_*^k$, we gather the attention scores and weighted embeddings from Lemma B.1, B.2.

$$r_1[i] = \begin{cases} s^0_{x_i} + \sum_{h=1}^{k-1} s^h_{x_{i-h}} + \mathcal{O}(ie^{-c}) \cdot \mathbf{1} & \text{for } i \geqslant k-1, \\ s^0_{x_i} + \sum_{h=1}^{i} s^h_{x_{i-h}} + \sum_{h=i+1}^{k-1} s^h_{x_0} + \mathcal{O}(ie^{-c}) \cdot \mathbf{1} & \text{for } i < k-1. \end{cases} , \tag{17}$$

$$a^{(2)}_{(i,t)} = \begin{cases} \frac{1}{|\mathsf{M}_t^k|} - \frac{\mathcal{O}(te^{-c})}{|\mathsf{M}_t^k|^2} & \text{for } i \in \mathsf{M}_t^k, \\ \frac{\mathcal{O}(e^{-c})}{|\mathsf{M}_t^k|} & \text{o.w.} \end{cases} \tag{18}$$

The average embedding,

$$\bar{r}_t^{(1)} = \sum_{i=1}^t a^{(2)}_{(i,t)} r_1[i] = \sum_{i \in \mathsf{M}_t^k} a^{(2)}_{(i,t)} r_1[i] + \sum_{i \notin \mathsf{M}_t^k} a^{(2)}_{(i,t)} r_1,$$

The deviation due to finite weights can be controlled as the following,

$$\left\| \sum_{i \notin M_t} a^{(2)}_{(i,t)} r_1[i] \right\|_\infty = \sum_{i \notin M_t} a^{(2)}_{(i,t)} \sup_{i \in [T]} \|r_1[i]\|_\infty = \frac{\mathcal{O}(te^{-c})}{|\mathsf{M}_t^k|},$$

as $\|r_1[i]\|_\infty \leqslant 1$ for all i and $a^{(2)}_{(i,t)}$ from Eq. (18) for $i \notin \mathsf{M}_t^k$. Now, we consider the summation $\sum_{i\in\mathsf{M}_t^k} a^{(2)}_{(i,t)} r_1[i]$.

$$\sum_{i\in\mathsf{M}_t^k} a^{(2)}_{(i,t)} r_1[i] = \sum_{i\in\mathsf{M}_t^k} \left[\frac{1}{\mathsf{M}_t^k} - \frac{\mathcal{O}(te^{-c})}{|\mathsf{M}_t^k|^2}\right] r_1[i] = \frac{1}{|\mathsf{M}_t^k|}\sum_{i\in\mathsf{M}_t^k} r_1[i] - \frac{\mathcal{O}(te^{-c})}{|\mathsf{M}_t^k|}\mathbf{1}.$$

Recall that for $i \in \mathsf{M}_t^k$, $r_1[i]$ from Eq. (17) gives

$$r_1[i] = s^0_{x_i} + \sum_{h=1}^{k-1} s^h_{x_{i-h}} + \mathcal{O}(ie^{-c})\cdot\mathbf{1}.$$

Now we use the definition of the set M_t^k to simplify the above expressions of $r_1[i]$ and $\bar{r}_t^{(1)}$. Note that $x_{i-h} = x_{t+1-h}$ for $h \in [k-1], i \in \mathsf{M}_t^k$. Using this, we have,

$$r_1[i] = s^0_{x_i} + \sum_{h=1}^{k-1} s^h_{x_{t+1-h}} + \mathcal{O}(ie^{-c})\cdot\mathbf{1},$$
$$\frac{1}{|\mathsf{M}_t^k|}\sum_{i\in\mathsf{M}_t^k} r_1[i] = \frac{1}{|\mathsf{M}_t^k|}\sum_{i\in\mathsf{M}_t^k} s^0_{x_i} + \sum_{h=1}^{k-1} s^h_{x_{t+1-h}} + \mathcal{O}(te^{-c})\cdot\mathbf{1}.$$

Combining them, we get

$$\left\|\sum_{i\in\mathsf{M}_t^k} a^{(2)}_{(i,t)} r_1[i] - \frac{1}{|\mathsf{M}_t^k|}\sum_{i\in\mathsf{M}_t^k} s^0_{x_i} - \sum_{h=1}^{k-1} s^h_{x_{t+1-k}}\right\|_\infty = \mathcal{O}(te^{-c}),$$
$$\bar{r}_t^{(1)} = \frac{1}{|\mathsf{M}_t^k|}\sum_{i\in\mathsf{M}_t^k} s^0_{x_i} + \sum_{h=1}^{k-1} s^h_{x_{t+1-k}} + \mathcal{O}(te^{-c})\cdot\mathbf{1}$$

For $i \in \mathsf{M}_t^k$,

$$r_1[i] - \bar{r}_t^{(1)} = s^0_{x_i} - \frac{\sum_{i\in\mathsf{M}_t^k} s^0_{x_i}}{\mathsf{M}_t^k} + \mathcal{O}(te^{-c})\cdot\mathbf{1},$$
$$\mathbf{K}_2(r_1[i] - \bar{r}_t^{(1)}) = \mathcal{O}(t\sqrt{c}e^{-c})\cdot\mathbf{1},$$
$$\mathbf{Q}_2(r_1[t]) = \sqrt{c}\sum_{h=1}^{k-1} s^h_{x_{t+1-h}} + \mathcal{O}(t\sqrt{c}e^{-c})\cdot\mathbf{1}.$$

For all i,

$$\left\|\mathbf{Q}_2 r_1[t](r_1[i] - \bar{r}_t^{(1)})^\top\right\|_\infty \leqslant \sqrt{c},$$
$$\left\|\mathbf{K}_2(r_1[i] - \bar{r}_t^{(1)})(r_1[t])^\top\right\|_\infty \leqslant \sqrt{c}.$$

Recalling the gradients,

$$\frac{\partial p_{\boldsymbol{\theta}}}{\partial \mathbf{K}_2} = \sum_{i=1}^{t} a^{(2)}_{(i,t)}\,(e_{x_i}) \otimes \mathrm{vec}\left(\mathbf{Q}_2 r_1[t](r_1[i] - \bar{r}_t^{(1)})^\top\right)^\top,$$
$$\frac{\partial p_{\boldsymbol{\theta}}}{\partial \mathbf{Q}_2} = \sum_{i=1}^{t} a^{(2)}_{(i,t)}\,(e_{x_i}) \otimes \mathrm{vec}\left(\mathbf{K}_2(r_1[i] - \bar{r}_t^{(1)})(r_1[t])^\top\right)^\top.$$

We split the gradients supported on M_t^k and its complement,

$$\begin{aligned}\frac{\partial p_{\boldsymbol{\theta}}}{\partial \mathbf{Q}_2} &= \sum_{i \in \mathsf{M}_t^k} a^{(2)}_{(i,t)}\left(e_{x_i}\right) \otimes \operatorname{vec}\left(\mathbf{K}_2 (r_1[i] - \bar{r}_t^{(1)})(r_1[t])^\top\right)^\top + \sum_{i \notin \mathsf{M}_t^k} a^{(2)}_{(i,t)}\left(e_{x_i}\right) \otimes \operatorname{vec}\left(\mathbf{K}_2 (r_1[i] - \bar{r}_t^{(1)})(r_1[t])^\top\right)^\top, \\ &= \mathcal{O}(t\sqrt{c}e^{-c}) \cdot \mathbf{1} + \frac{\mathcal{O}(t\sqrt{c}e^{-c}) \cdot \mathbf{1}}{|\mathsf{M}_t^k|}.\end{aligned}$$

The final gradient,

$$\frac{\partial p_{\boldsymbol{\theta}}}{\partial \mathbf{Q}_2} = \mathcal{O}(t\sqrt{c}e^{-c}) \cdot \mathbf{1} + \frac{\mathcal{O}(t\sqrt{c}e^{-c}) \cdot \mathbf{1}}{|\mathsf{M}_t^k|}. \tag{19}$$

On the similar lines,

$$\frac{\partial p_{\boldsymbol{\theta}}}{\partial \mathbf{K}_2} = \sqrt{c}\frac{1}{|\mathsf{M}_t^k|}\sum_{i \in \mathsf{M}_t^k} e_{x_i} \otimes \operatorname{vec}\left(\left[\sum_{h=1}^{k-1} s^h_{x_{t+1-h}}\right]\left[s^0_{x_i} - \frac{\sum_{i \in \mathsf{M}_t^k} s^0_{x_i}}{|\mathsf{M}_t^k|}\right]^\top\right) + \mathcal{O}(t\sqrt{c}e^{-c}) \cdot \mathbf{1}.$$

Denote

$$\begin{aligned}\phi(x^t_{t-k+2}) &\coloneqq \sum_{h=1}^{k-1} s^h_{x_{t+1-h}}, \\ \bar{s}^0_t &\coloneqq \frac{1}{|\mathsf{M}_t^k|}\sum_{i \in \mathsf{M}_t^k} s^0_{x_i}.\end{aligned}$$

Note that the first term in the above expression is only a function of k-history x^t_{t-k+2}. Using this,

$$\begin{aligned}\frac{\partial p_{\boldsymbol{\theta}}}{\partial \mathbf{K}_2} &= \sqrt{c}\frac{1}{|\mathsf{M}_t^k|}\sum_{i \in \mathsf{M}_t^k} e_{x_i} \otimes \operatorname{vec}\left(\phi(x^t_{t-k+2})\left[s^0_{x_i} - \bar{s}^0_t\right]^\top\right) + \mathcal{O}(t\sqrt{c}e^{-c}) \cdot \mathbf{1}, \\ &= c^{1/2}\frac{1}{|\mathsf{M}_t^k|}\sum_{a \in \mathcal{S}} \#\{a \in \mathsf{M}_t^k\}\left(e_a\right) \otimes \operatorname{vec}\left(\phi(x^t_{t-k+2})\left(s^0_a - \bar{s}^0_t\right)\right)^\top + \mathcal{O}(t\sqrt{c}e^{-c}) \cdot \mathbf{1}.\end{aligned}$$

Note that the second term $\bar{s}^0_t$ is a function of the k-gram MLE $\widehat{p}_k$, precisely,

$$\widehat{p}_k[a] = \frac{\#\{a \in \mathsf{M}_t^k\}}{|\mathsf{M}_t^k|}.$$

Using this $\bar{s}^0_t = S^0 \widehat{p}_k$ where S^0 is the $\mathbb{R}^{nd \times nd}$ matrix where the first block is the embedding matrix and 0's everywhere else. Using this

$$\frac{\partial p_{\boldsymbol{\theta}}}{\partial \mathbf{K}_2} = \sqrt{c}\sum_{a \in \mathcal{S}} \widehat{p}_k[a]\, e_a \otimes \operatorname{vec}\left(\phi(x^t_{t-k+2})\left[s^0_a - S^0\widehat{p}_k\right]^\top\right) + \mathcal{O}(t\sqrt{c}e^{-c}) \cdot \mathbf{1}. \tag{20}$$

Parameters of the first layer. The derivatives with respect to the first layer parameters are given by,

$$\begin{aligned}\frac{\partial p_{\boldsymbol{\theta}}}{\partial \mathbf{V}_1^{(h)}} &= \sum_{i=1}^{t} a^{(2)}_{(i,t)}\left((e_{x_i} - \bar{e}_t^{(1)})\,(\bar{r}_i^{(0)})^\top\right) \otimes \left(r_1[t]^\top \mathbf{Q}_2^\top \mathbf{K}_2 \mathcal{W}_o^{(h)} + (r_1[i] - \bar{r}_t^{(1)})^\top \mathbf{K}_2^\top \mathbf{Q}_2 \mathcal{W}_o^{(h)}\right), \\ \frac{\partial p_{\boldsymbol{\theta}}}{\partial \mathbf{A}_1^{(h)}[i,j]} &= a^{(1,h)}_{(j,i)} a^{(2)}_{(i,t)}\left(e_{x_i} - \bar{e}_t^{(1)}\right) \otimes \\ &\quad \left(r_1[t]^\top \mathbf{Q}_2^\top \mathbf{K}_2 \mathcal{W}_o^{(h)} \mathbf{V}_1^{(h)}\left(r_0[j] - \bar{r}_i^{(0,h)}\right) + (r_1[i] - \bar{r}_t^{(1)})^\top \mathbf{K}_2^\top \mathbf{Q}_2 \mathcal{W}_o^{(h)} \mathbf{V}_1^{(h)}\left(r_0[j] - \bar{r}_i^{(0,h)}\right)\right),\end{aligned}$$

where the averages of embeddings weighted with attention scores are given by,

$$\bar{e}_t^{(1)} = \sum_{i=1}^{t} a_{(i,t)}^{(2)} e_{x_i},$$
$$\bar{r}_i^{(0,h)} = \sum_{j=1}^{i} a_{(j,i)}^{(1,h)} r_0[j].$$

For the heads that are not activated, the derivatives with $\mathbf{A}_1^{(h)}[i,j]$ are 0, since the $\mathbf{V}_1^{(h)}$ is 0. For the activated heads, the averaged embedding is given by,

$$\bar{r}_i^{(0,h)} = \sum_{j=1}^{i} a_{(j,i)}^{(1,h)} r_0[j] = r_0[i-h] + \mathcal{O}(i\exp\{-c\}) \cdot \mathbf{1}.$$

Now, consider two cases for the derivatives.

- For $j = i-h$, $r_0[j] - \bar{r}_i^{(0,h)} = \mathcal{O}(\exp\{-c\})$, hence the derivative is $\mathcal{O}(ic\exp\{-c\})$.
- For $j \neq i-h$, due to the property of the softmax function, the attention scores are $\mathcal{O}(\exp\{-c\})$, hence the derivative is again $\mathcal{O}(c\exp\{-c\})$.

For the derivative with respect to $\mathbf{V}_1^{(h)}$, we again have two cases,

- For the non-activated heads $h \geqslant k$, $\mathbf{K}_2\mathcal{W}_o^{(h)} = \sum_{h'=1}^{k-1}\sum_{j=1}^{\mathcal{S}} s_j^{h'}(s_j^{h'})^\top \cdot \sum_{j=1}^{\mathcal{S}} s_j^h s_j^\top = 0$. Similarly for the other term, $\mathbf{Q}_2\mathcal{W}_o^{(h)} = 0$. Hence the derivative is 0.
- For the activated heads $h \leqslant k-1$, using the previous computations we have,

$$\mathbf{K}_2(r_1[i] - \bar{r}_t^{(1)}) = \mathcal{O}(t\sqrt{c}e^{-c}) \cdot \mathbf{1},$$
$$\mathbf{Q}_2(r_1[t]) = \sqrt{c}\sum_{h=1}^{k-1} s_{x_{t+1-h}}^h + \mathcal{O}(t\sqrt{c}e^{-c}) \cdot \mathbf{1},$$
$$\mathbf{K}_2\mathcal{W}_o^{(h)} = \sqrt{c}\sum_{h'=1}^{k-1}\sum_{j=1}^{\mathcal{S}} s_j^{h'}(s_j^{h'})^\top \cdot \sum_{j=1}^{\mathcal{S}} s_j^h s_j^\top$$

The product is solely a function of k-history x_{t-k+2}^t and does not depend on i. Computing the multiplicative factor in the front,

$$\sum_{i=1}^{t} a_{(i,t)}^{(2)} \left((e_{x_i} - \bar{e}_t^{(1)})\,(\bar{r}_i^{(0)})^\top\right) = \sum_{i \in \mathsf{M}_t^k} a_{(i,t)}^{(2)} \left((e_{x_i} - \bar{e}_t^{(1)})\,(\bar{r}_i^{(0)})^\top\right) + \mathcal{O}(\exp\{-c\})$$

Note that $\bar{r}_i^{(0)} = s_{x_{i-h}} + \mathcal{O}(i\exp\{-c\})$ and for $i \in \mathsf{M}_t^k$, we have, $s_{x_{i-h}} = s_{x_{t+1-h}}$. Using this,

$$\begin{aligned}\sum_{i=1}^{t} a_{(i,t)}^{(2)} \left((e_{x_i} - \bar{e}_t^{(1)})\,(\bar{r}_i^{(0)})^\top\right) &= \sum_{i \in \mathsf{M}_t^k} a_{(i,t)}^{(2)} \left((e_{x_i} - \bar{e}_t^{(1)})\,(s_{x_{t+1-h}})^\top\right) + \mathcal{O}(\exp\{-c\}),\\ &= \left[\sum_{i \in \mathsf{M}_t^k} a_{(i,t)}^{(2)}\,(e_{x_i}) - \left(\sum_{i \in \mathsf{M}_t^k} a_{(i,t)}^{(2)}\right)\bar{e}_t^{(1)}\right](s_{x_{t+1-h}})^\top + \mathcal{O}(\exp\{-c\})\end{aligned}$$

The term in the square brackets is $\mathcal{O}(\exp\{-c\})$ (for hard attention it is zero). Hence the derivative with respect to $\mathbf{V}_1^{(h)}$ is $\mathcal{O}(\exp\{-c\})$.

$$\frac{\partial p_{\boldsymbol{\theta}}}{\partial \mathbf{V}_1^{(h)}} = \mathcal{O}(\exp\{-c\})\left[\sqrt{c}\sum_{h=1}^{k-1} s_{x_{t+1-h}}^h + \mathcal{O}(t\sqrt{c}e^{-c}) \cdot \mathbf{1}\right]. \tag{21}$$

Final Gradients Bringing things together, from Equations (19), (20), (21),

$$\begin{aligned}
\frac{\partial p_{\boldsymbol{\theta}}}{\partial \mathbf{Q}_2} &= \mathcal{O}(t\sqrt{c}e^{-c}) \cdot \mathbf{1} + \frac{\mathcal{O}(t\sqrt{c}e^{-c}) \cdot \mathbf{1}}{|\mathsf{M}_t^k|}, \\
\frac{\partial p_{\boldsymbol{\theta}}}{\partial \mathbf{V}_1^{(h)}} &= \mathcal{O}(\exp\{-c\}) \left[\sqrt{c} \sum_{h=1}^{k-1} s^h_{x_{t+1-h}} + \mathcal{O}(t\sqrt{c}e^{-c}) \cdot \mathbf{1} \right], \\
\frac{\partial p_{\boldsymbol{\theta}}}{\partial \mathbf{K}_2} &= \sqrt{c} \sum_{a \in \mathcal{S}} \widehat{p}_k[a]\, e_a \otimes \operatorname{vec}\left(\phi(x^t_{t-k+2}) \left[s^0_a - S^0 \widehat{p}_k \right]^\top \right) + \mathcal{O}(t\sqrt{c}e^{-c}) \cdot \mathbf{1}.
\end{aligned}$$

Note that the only non-vanishing gradient is the gradient with respect to $\mathbf{K}_2$. Now, the gradient can be written as

$$\frac{\partial p_{\boldsymbol{\theta}}}{\partial \mathbf{K}_2} = \sqrt{c} \sum_{a \in \mathcal{S}} \widehat{p}_k[a]\, e_a \otimes \operatorname{vec}\left(\phi(x^t_{t-k+2}) \left[s^0_a - S^0 \widehat{p}_k \right]^\top \right) + \mathcal{O}(t\sqrt{c}e^{-c}) \cdot \mathbf{1}.$$

We use p_τ to denote the vector $p_\tau\left(.|\, x^T_{T-k+2}\right)$, the gradient using the difference as,

$$\begin{aligned}
\frac{\partial p_{\boldsymbol{\theta}}}{\partial \mathbf{K}_2} &= \sqrt{c} \sum_{a \in \mathcal{S}} p_\tau[a]\, e_a \otimes \operatorname{vec}\left(\phi(x^t_{t-k+2}) \left[s^0_a - S^0 p_\tau \right]^\top \right) + \mathcal{O}(\|p_\tau - \widehat{p}_k\|^2) + \mathcal{O}(t\sqrt{c}e^{-c}) \cdot \mathbf{1}, \\
&= \psi(x^t_{t-k+2}) + \mathcal{O}(\|p_\tau - \widehat{p}_k\|^2) + \mathcal{O}(t\sqrt{c}e^{-c}) \cdot \mathbf{1},
\end{aligned}$$

where psi is appropriately defined. The first term here only depends on the (k)-history and we can use Proposition 3.2, to show that it does not contribute to the final gradient. To give the final computation,

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mathbf{K}_2} &= \mathop{\mathbb{E}}_{p_\tau \sim \mathcal{P}} \mathop{\mathbb{E}}_{x^T \sim p_\tau} \left\langle p_\theta(x^T) - p_\tau\left(.|\, x^T\right), \frac{\partial \log p_\theta(x^T)}{\partial \mathbf{K}_2} \right\rangle, \\
&= \mathop{\mathbb{E}}_{p_\tau \sim \mathcal{P}} \mathop{\mathbb{E}}_{x^T \sim p_\tau} \left\langle \widehat{p}_k - p_\tau\left(.|\, x^T\right), \frac{\partial \log p_\theta(x^T)}{\partial \mathbf{K}_2} \right\rangle, \\
&= \mathop{\mathbb{E}}_{p_\tau \sim \mathcal{P}} \mathop{\mathbb{E}}_{x^T \sim p_\tau} \left\langle p_\tau\left(.|\, x^T_{T-k+2}\right) - p_\tau\left(.|\, x^T\right), \frac{\partial \log p_\theta(x^T)}{\partial \mathbf{K}_2} \right\rangle + \mathop{\mathbb{E}}_{p_\tau \sim \mathcal{P}} \mathop{\mathbb{E}}_{x^T \sim p_\tau} \|\widehat{p}_k - p_\tau\left(.|\, x^T_{T-k+2}\right)\| \|\frac{\partial \log p_\theta(x^T)}{\partial \mathbf{K}_2}\|, \\
&= \mathop{\mathbb{E}}_{p_\tau \sim \mathcal{P}} \mathop{\mathbb{E}}_{x^T \sim p_\tau} \left\langle p_\tau\left(.|\, x^T_{T-k+2}\right) - p_\tau\left(.|\, x^T\right), \psi(x^t_{t-k+2}) \right\rangle + \mathop{\mathbb{E}}_{p_\tau \sim \mathcal{P}} \mathop{\mathbb{E}}_{x^T} \mathcal{O}(\|p_\tau\left(.|\, x^T_{T-k+2}\right) - \widehat{p}_k\|^2) \cdot \mathbf{1} + \mathcal{O}(t\sqrt{c}e^{-c}) \cdot \mathbf{1},
\end{aligned}$$

■

The first term vanishes due to Proposition 3.2 and this finishes the proof.

C. Supporting Lemmas for The Derivatives of Cross-entropy Loss

In this section, we provide the proofs of the lemmas presented in Section 3 of the main text. In the end, we remark about extending Proposition 3.2.

Lemma 3.1. *Consider any parametric model $p_\theta(.) : [\mathcal{S}]^T \to \Delta^{\mathcal{S}-1}$ that maps a sequence of states to a probability vector on the states. The derivative of the cross entropy loss function $\mathcal{L}(\theta)$ with respect to a parameter $\theta_i \in \mathbb{R}$ is*

$$\partial_{\theta_i} \mathcal{L}(\theta) = \mathop{\mathbb{E}}_{p_\tau \sim \mathcal{P}} \mathop{\mathbb{E}}_{x^T \sim p_\tau} \left\langle p_\theta(x^T) - p_\tau\left(.|\, x^T\right), \partial_{\theta_i} \log p_\theta(x^T) \right\rangle,$$

where ∂_{θ_i} is the partial derivative with respect to θ_i and $\log(\cdot)$ denotes component-wise logarithm.

Proof Recalling the definition of the population cross-entropy loss from Eq. (1), we have,

$$\mathcal{L}(\theta) = \underset{p_\tau \sim \mathcal{P}}{\mathbb{E}} \underset{x^T \sim p_\tau}{\mathbb{E}} \ell\left(\theta, x^T\right),$$

$$\ell\left(\theta, x^T\right) = -\sum_{s=1}^{\mathcal{S}} p_\tau(x_{T+1}{=}s|x^T) \log\left(p_\theta(\theta, x^T)[s]\right),$$

$$\partial_{\theta_i} \ell\left(\theta, x^T\right) = -\sum_{s=1}^{\mathcal{S}} p_\tau(x_{T+1}{=}s|x^T) \frac{\partial_{\theta_i} p_\theta(x^T)[s]}{p_\theta(x^T)[s]}$$

Using the fact that $p_\theta(x^T) \in \Delta^{\mathcal{S}-1}$,

$$\sum_{s=1}^{\mathcal{S}} p_\theta(x^T)[s] = 1.$$

Taking the derivative of the above expression, we have,

$$\sum_{s=1}^{\mathcal{S}} \partial_{\theta_i} p_\theta(x^T)[s] = 0.$$

Using this gives,

$$\begin{aligned}\partial_{\theta_i} \ell\left(\theta, x^T\right) &= -\sum_{s=1}^{\mathcal{S}} p_\tau(x_{T+1}{=}s|x^T) \frac{\partial_{\theta_i} p_\theta(x^T)[s]}{p_\theta(x^T)[s]} + \sum_{s=1}^{\mathcal{S}} \partial_{\theta_i} p_\theta(x^T)[s], \\ &= \left\langle p_\theta(x^T) {-} p_\tau\left(.|\, x^T\right), \partial_{\theta_i} \log p_\theta(x^T) \right\rangle\end{aligned}$$

■

Remark C.1. In general, a parametric model computes a function $f_\theta : [\mathcal{S}]^* \to \mathbb{R}^{\mathcal{S}}$ after which a normalizing function like soft-max is used to project it onto the simplex $\Delta^{\mathcal{S}-1}$. The gradient in this case simplifies to

$$\partial_{\theta_i} \mathcal{L}(\theta) {=} \underset{p_\tau \sim \mathcal{P}}{\mathbb{E}} \underset{x^T \sim p_\tau}{\mathbb{E}} \left\langle p_\theta(x^T) {-} p_\tau\left(.|\, x^T\right), \partial_{\theta_i} f_\theta(x^T) \right\rangle.$$

Proposition 3.2. *For any $\theta_* \in \mathbb{R}^p$ such that $\partial_{\theta=\theta_*} \log p_\theta(x^T) = g\left(p_\tau, x_t^T\right)$, i.e., the derivative is solely a function of the context p_τ and the last $T-t+1$ elements of the sequence x^T, the gradient of the population loss $\mathcal{L}$ can be written as*

$$\nabla \mathcal{L}(\theta_*) {=} \underset{p_\tau \sim \mathcal{P}}{\mathbb{E}} \underset{x^T \sim p_\tau}{\mathbb{E}} \left\langle p_{\theta_*}(x^T) {-} p_\tau\left(.|\, x_t^T\right), g\left(p_\tau, x_t^T\right) \right\rangle.$$

Furthermore, if for such $\theta_* \in \mathbb{R}^p$, the model estimates the conditional probability of the next token $p_\tau\left(.|\, x_t^T\right)$, i.e., $p_{\theta_*}(x^T) {=} p_\tau\left(.|\, x_t^T\right)$ almost surely for $p_\tau \sim \mathcal{P}$, then θ_* is a stationary point.

Proof From the above lemma, the partial derivative of the population loss is given by,

$$\partial_{\theta_i} \mathcal{L}(\theta) {=} \underset{p_\tau \sim \mathcal{P}}{\mathbb{E}} \underset{x^T \sim p_\tau}{\mathbb{E}} \left\langle p_\theta(x^T) {-} p_\tau\left(.|\, x^T\right), \partial_{\theta_i} \log p_\theta(x^T) \right\rangle,$$

Using our assumption in (a), we can rewrite the above expression as,

$$\underset{x^T \sim p_\tau}{\mathbb{E}} \left\langle p_\tau\left(.|\, x^T\right), \partial_{\theta_i} \log p_\theta(x^T) \right\rangle = \underset{x^T \sim p_\tau}{\mathbb{E}} \left\langle p_\tau\left(.|\, x^T\right), g\left(p_\tau, x_t^T\right) \right\rangle,$$

now, we split the sequence x^T into two parts, $(x^{t-1},\, x_t^T)$, we can be slightly simplified as,

$$\underset{x^T \sim p_\tau}{\mathbb{E}} \left\langle p_\tau\left(.|\, x^T\right), \partial_{\theta_i} \log p_\theta(x^T) \right\rangle = \underset{\left(x^{t-1},\, x_t^T\right) \sim p_\tau}{\mathbb{E}} \left\langle p_\tau\left(.|\, (x^{t-1},\, x_t^T)\right), g\left(p_\tau, x_t^T\right) \right\rangle, \tag{22}$$

We use the following property on factorization of the expectation, for any two random variables, A, B. Let $\sigma(A), \sigma(B)$ be the support of the respective random variable, we have the following fact,

$$\begin{aligned}
\mathop{\mathbb{E}}_{A,B} Pr\Big(\cdot \Big|(A,B)\Big)\phi(B) &= \sum_{b\in\sigma(B)} \sum_{a\in\sigma(A)} Pr(A=a, B=b) Pr\Big(\cdot \Big|(A=a,B=b)\Big)\phi(B), \\
&= \sum_{b\in\sigma(B)} Pr(B=b)\phi(B) \sum_{a\in\sigma(A)} Pr(A=a|B=b)\, Pr\Big(\cdot \Big|(A=a,B=b)\Big), \\
&= \sum_{b\in\sigma(B)} Pr(B=b)\phi(B) Pr\Big(\cdot \Big|B=b\Big), \\
&= \sum_{b\in\sigma(B)} Pr(B=b)\phi(B) Pr\Big(\cdot \Big|B=b\Big) \sum_{a\in\sigma(A)} Pr(A=a|B=b), \\
&= \sum_{b\in\sigma(B)} \sum_{a\in\sigma(A)} Pr(A=a,B=b)\phi(B) Pr\Big(\cdot \Big|B=b\Big) \\
&= \mathop{\mathbb{E}}_{A,B} Pr\Big(\cdot \Big|B\Big)\phi(B).
\end{aligned}$$

Using the above property, we can rewrite the expression Eq.(22) using $A = x^{t-1}$ and $B = x_t^T$, the per task loss can be written as,

$$\mathop{\mathbb{E}}_{x^T\sim p_\tau} \Big\langle p_\tau\left(.|\, x^T\right), \partial_{\theta_i} \log p_\theta(x^T)\Big\rangle = \mathop{\mathbb{E}}_{x^T\sim p_\tau} \Big\langle p_\tau\left(.|\, x_t^T\right), g\left(p_\tau, x_t^T\right)\Big\rangle.$$

This gives us the desired result for (a). The proof of (b) is straight forward as the estimation of probability matches and the residue vanishes. ■

Remark C.2. The proof of the proposition splits the sequence into a prefix and suffix as shown in Eq. (22). This is a choice we made for the ease of presentation and discussion. The proof works similarly after splitting the sequence into any two disjoint subsequences that do not have to be a prefix and suffix. Therefore, the result can be extended to conditioning on any subsequence that is not necessarily a suffix.

C.1. Conditional Probabilities: Definition and Proper Asymptotics

In the main paper, we have used a somewhat informal treatment of conditional probabilities. Here, we provide formal definitions to supplement our discussion. First, we define them for any general sequences of length T for any generic probability distribution on $[\mathcal{S}]^*$.

Definition C.3. Given the propability distribution $p_\tau(x_1, \ldots, x_T, x_{T+1})$ of a sequence of random variables $x_1, \ldots, x_T$ taking their values in $[\mathcal{S}]$. We define the following conditional probabilities of the next token given a $(T-t)-$history is:

$$p_\tau\left(x_{T+1} = i_{T+1} \;\middle|\; x_{t+1}^T = i_{t+1}^T\right) = \frac{\sum_{i^t} p_\tau\left(x^{T+1} = i^{T+1}\right)}{\sum_{i^t} p_\tau\left(x^T = i^T\right)}.$$

where the marginal distribution for sequence length t is given by

$$p_\tau(x^T = i^T) = \sum_{i_T\in[\mathcal{S}]} p_\tau(x^{T+1} = i^{T+1}).$$

Now, under the assumption that p_τ is a $n-$gram language model, it remains to show that we have a time-homogeneous definition of conditional probabilities and that the $k-$gram estimators asymptotically converge towards them. This is handled in detail in the section H.

D. Derivatives of The Simplified Transformer

In this section, we focus on the derivatives of the simplified two-layer transformer. Using the derivative of the masked self-attention map Lemma G.1, the derivatives for the two layers are computed here.

Lemma D.1. *Using the Lemma G.1, define the following quantities which can be seen as embeddings weighted with attention scores for both the layers,*

$$\bar{r}_t^{(1)} = \sum_{i=1}^{t} a_{(i,t)}^{(2)} r_1[i], \tag{23}$$

$$\bar{e}_t^{(1)} = \sum_{i=1}^{t} a_{(i,t)}^{(2)} e_{x_i}, \tag{24}$$

$$\bar{r}_i^{(0,h)} = \sum_{j=1}^{i} a_{(j,i)}^{(1,h)} r_0[j]. \tag{25}$$

With the above notations the derivatives of the output probabilities after t tokens with respect to the parameters of the model are given by,

$$\begin{aligned}
\frac{\partial p_{\boldsymbol{\theta}}}{\partial \mathbf{K}_2} &= \sum_{i=1}^{t} a_{(i,t)}^{(2)} \, (e_{x_i}) \otimes \operatorname{vec}\left(\mathbf{Q}_2 r_1[t](r_1[i] - \bar{r}_t^{(1)})^\top\right)^\top, \\
\frac{\partial p_{\boldsymbol{\theta}}}{\partial \mathbf{Q}_2} &= \sum_{i=1}^{t} a_{(i,t)}^{(2)} \, (e_{x_i}) \otimes \operatorname{vec}\left(\mathbf{K}_2 (r_1[t] - \bar{r}_t^{(1)})(r_1[t])^\top\right)^\top, \\
\frac{\partial p_{\boldsymbol{\theta}}}{\partial \mathbf{V}_1^{(h)}} &= \sum_{i=1}^{t} a_{(i,t)}^{(2)} \left((e_{x_i} - \bar{e}_t^{(1)}) \, (\bar{r}_i^{(0)})^\top\right) \otimes \left(r_1[t]^\top \mathbf{Q}_2^\top \mathbf{K}_2 \mathcal{W}_o^{(h)} + (r_1[i] - \bar{r}_t^{(1)})^\top \mathbf{K}_2^\top \mathbf{Q}_2 \mathcal{W}_o^{(h)}\right) \\
\frac{\partial p_{\boldsymbol{\theta}}}{\partial \mathbf{A}_1^{(h)}[i,j]} &= a_{(j,i)}^{(1,h)} a_{(i,t)}^{(2)} \, (e_{x_i} - \bar{e}_t^{(1)}) \otimes \\
&\quad \left(r_1[t]^\top \mathbf{Q}_2^\top \mathbf{K}_2 \mathcal{W}_o^{(h)} \mathbf{V}_1^{(h)} \, (r_0[j] - \bar{r}_i^{(0,h)}) + (r_1[i] - \bar{r}_t^{(1)})^\top \mathbf{K}_2^\top \mathbf{Q}_2 \mathcal{W}_o^{(h)} \mathbf{V}_1^{(h)} \, (r_0[j] - \bar{r}_i^{(0,h)})\right)
\end{aligned}$$

Proof

The final output probabilities are given by

$$p_{\boldsymbol{\theta}}(x^t) = U r_2[t] = \sum_{i=1}^{t} a_{(i,t)}^{(2)} U r_0[i] = \sum_{i=1}^{t} a_{(i,t)}^{(2)} e_{x_i}.$$

where the attention scores write

$$a_{(i,t)}^{(2)} = \frac{\exp \langle \mathbf{K}_2 r_1[i], \mathbf{Q}_2 r_1[t] \rangle}{\sum_{j=1}^{t} \exp \langle \mathbf{K}_2 r_1[j], \mathbf{Q}_2 r_1[t] \rangle}.$$

Using the Lemma G.1, the derivatives of the output probabilities after t tokens with respect to the parameters in the second layer and the embedding of the first layer are given by,

$$\begin{aligned}
\frac{\partial p_{\boldsymbol{\theta}}}{\partial \mathbf{K}_2} &= \sum_{i=1}^{t} a_{(i,t)}^{(2)} \, (e_{x_i}) \otimes \operatorname{vec}\left(\mathbf{Q}_2 r_1[t](r_1[i] - \bar{r}_t^{(1)})^\top\right)^\top, \\
\frac{\partial p_{\boldsymbol{\theta}}}{\partial \mathbf{Q}_2} &= \sum_{i=1}^{t} a_{(i,t)}^{(2)} \, (e_{x_i}) \otimes \operatorname{vec}\left(\mathbf{K}_2 (r_1[t] - \bar{r}_t^{(1)})(r_1[t])^\top\right)^\top,
\end{aligned}$$

For $i \neq t$,

$$
\begin{aligned}
\frac{\partial r_2[t]}{\partial r_1[i]} &= a^{(2)}_{(i,t)} \left(e_{x_i} - \bar{e}^{(1)}_t\right) \otimes \left(\mathbf{K}_2^\top \mathbf{Q}_2 r_1[t]\right)^\top, \\
\frac{\partial r_2[t]}{\partial r_1[t]} &= a^{(2)}_{(t,t)} \left(e_{x_t} - \bar{e}^{(1)}_t\right) \otimes \left(\mathbf{K}_2^\top \mathbf{Q}_2 r_1[t]\right)^\top + \sum_{i=1}^{t} a^{(1)}_{(i,t)} \left(e_{x_i} - \bar{e}^{(1)}_t\right) \otimes \left(\mathbf{Q}_2^\top \mathbf{K}_2 (r_1[i] - \bar{r}^{(1)}_t)\right)^\top.
\end{aligned}
$$

For the first attention layer,

$$
r_1[i] = \mathcal{W}_o^{(0)} r_0[0] + \sum_{h=1}^{n-1} \sum_{j=1}^{i} a^{(1,h)}_{(j,i)} \mathcal{W}_o^{(h)} \mathbf{V}_1^{(h)} r_0[j],
$$

where

$$
\begin{aligned}
a^{(1,h)}_{(j,i)} &= \frac{\exp \mathbf{A}_1^{(h)}[i,j]}{\sum_{l=1}^{i} \exp \mathbf{A}_1^{(h)}[i,l]}, \\
\mathcal{W}_o^{(h)} &= \sum_{j=1}^{\mathcal{S}-1} s_j^h s_j^\top.
\end{aligned}
$$

The derivatives wrt to

$$
\begin{aligned}
\frac{\partial r_i[i]}{\partial \mathbf{V}_1^{(h)}} &= (\bar{r}^{(0)}_i)^\top \otimes \mathcal{W}_o^{(h)}, \\
\frac{\partial r_i[i]}{\partial \mathbf{A}_1^{(h)}[i,j]} &= a^{(1,h)}_{(j,i)} \mathcal{W}_o^{(h)} \mathbf{V}_1^{(h)} \left(r_0[j] - \bar{r}^{(0,h)}_i\right)
\end{aligned}
$$

Combining them, we get,

$$
\begin{aligned}
\frac{\partial p_{\boldsymbol{\theta}}}{\partial \mathbf{V}_1^{(h)}} &= \sum_{i=1}^{t} \frac{\partial p_{\boldsymbol{\theta}}}{\partial r_1[i]} \frac{\partial r_1[i]}{\partial \mathbf{V}_1^{(h)}} = \sum_{i=1}^{t} a^{(1)}_{(i,t)} \left[\left(e_{x_i} - \bar{e}^{(1)}_t\right) \otimes \left(\mathbf{K}_2^\top \mathbf{Q}_2 r_1[t]\right)^\top\right] (\bar{r}^{(0)}_i)^\top \otimes \mathcal{W}_o^{(h)}, \\
&= \sum_{i=1}^{t} a^{(1)}_{(i,t)} \left(\left(e_{x_i} - \bar{e}^{(1)}_t\right) (\bar{r}^{(0)}_i)^\top\right) \otimes \left(r_1[t]^\top \mathbf{Q}_2^\top \mathbf{K}_2 \mathcal{W}_o^{(h)} + (r_1[i] - \bar{r}^{(1)}_t)^\top \mathbf{K}_2^\top \mathbf{Q}_2 \mathcal{W}_o^{(h)}\right),
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial p_{\boldsymbol{\theta}}}{\partial \mathbf{A}_1^{(h)}[i,j]} &= \frac{\partial p_{\boldsymbol{\theta}}}{\partial r_1[i]} \frac{\partial r_1[i]}{\partial \mathbf{A}_1^{(h)}[i,j]}, \\
&= a^{(1,h)}_{(j,i)} a^{(1)}_{(i,t)} \left(e_{x_i} - \bar{e}^{(1)}_t\right) \otimes \\
&\quad \left(r_1[t]^\top \mathbf{Q}_2^\top \mathbf{K}_2 \mathcal{W}_o^{(h)} \mathbf{V}_1^{(h)} \left(r_0[j] - \bar{r}^{(0,h)}_i\right) + (r_1[i] - \bar{r}^{(1)}_t)^\top \mathbf{K}_2^\top \mathbf{Q}_2 \mathcal{W}_o^{(h)} \mathbf{V}_1^{(h)} \left(r_0[j] - \bar{r}^{(0,h)}_i\right)\right).
\end{aligned}
$$

In the above simplifications, we have used the following property of Kronecker product,

$$
(A \otimes B)(C \otimes D) = (AC) \otimes (BD),
$$

for matrices A, B, C, D of appropriate dimensions.

■

E. Proofs of Representation with Simplified Transformers

Lemma B.1. *[First-Layer] For the first layer given by the parameter defined in Eq. (12),*

(a) Let $a^{(1,h)}_{(j,i)}$ denote the attention score of head h of layer 1 where (i, j) denote the key and query,

$$a^{(1,h)}_{(j,i)} = \begin{cases} 1 - \mathcal{O}(ie^{-c}) \text{ when } i \geqslant h \text{ and } j = i - h, \\ 1 - \mathcal{O}(ie^{-c}) \text{ when } i < h \text{ and } j = 0, \\ \mathcal{O}(e^{-c}) \text{ o.w.}. \end{cases} . \tag{14}$$

(b) The embeddings of the first layer,

$$r_1[i] = \begin{cases} s^0_{x_i} + \sum_{h=1}^{k-1} s^h_{x_{i-h}} + \mathcal{O}(ie^{-c}) \cdot \mathbf{1} \quad \text{for } i \geqslant k-1, \\ s^0_{x_i} + \sum_{h=1}^{i} s^h_{x_{i-h}} + \sum_{h=i+1}^{k-1} s^h_{x_0} + \mathcal{O}(ie^{-c}) \cdot \mathbf{1} \quad \text{for } i < k-1. \end{cases} . \tag{15}$$

Proof Note that the attention scores in the first layer for any head h and query i and key j are given by,

$$a^{(1,h)}_{(j,i)} = \frac{\exp \mathbf{A}^{(h)}_1[i,j]}{\sum_{l=1}^{i} \exp \mathbf{A}^{(h)}_1[i,l]},$$

Now for the activated head h, the $\mathbf{A}^{(h)}_1$ is given by Eq. (8a),

$$\mathbf{A}^{(h)}_1 = c \sum_{l=h}^{T-1} e_l e_{l-h}^\top + c \sum_{l=0}^{h} e_l e_0^\top$$

Using the above two equations for $i \geqslant k-1$, we can write the attention scores as,

$$\begin{aligned} \mathbf{A}^{(h)}_1[i,j] &= c\, \mathbb{1}\{j = i-h\}, \\ \sum_{j=1}^{i} \exp \mathbf{A}^{(h)}_1[i,j] &= (i-1) + e^c, \\ \text{For } j \neq i-h, \quad a^{(1,h)}_{(j,i)} &= \frac{1}{(i-1)+e^c} = \mathcal{O}(\exp\{-c\}), \\ \text{and, for } j = i-h, \quad a^{(1,h)}_{(j,i)} &= \frac{e^c}{(i-1)+e^c} = 1 - \mathcal{O}(i \exp\{-c\}). \end{aligned}$$

Coming to the embeddings after the first layer,

$$r_1[i] = \mathcal{W}^{(0)}_o r_0[0] + \sum_{h=1}^{n-1} \sum_{j=1}^{i} a^{(1,h)}_{(j,i)} \mathcal{W}^{(h)}_o \mathbf{V}^{(h)}_1 r_0[j],$$

where,

$$\mathcal{W}^{(h)}_o = \sum_{j=1}^{\mathcal{S}} s^h_j s_j^\top.$$

Note that for $h \geqslant k$ the term $\mathbf{V}^{(h)}_1 = 0$, and for $h < k$, the term $\mathbf{V}^{(h)}_1$ is given by Eq. (8b). Using the above two equations and $r_0[i] = s_{x_i}$, we can write the embeddings after the first layer as,

$$r_1[i] = \mathcal{W}^{(0)}_o s_{x_i} + \sum_{h=1}^{n-1} \sum_{j=1}^{i} a^{(1,h)}_{(j,i)} \mathcal{W}^{(h)}_o \mathbf{V}^{(h)}_1 s_{x_j}.$$

Note that $\mathbf{V}_1^{(h)} s_{x_j} = s_{x_j}$ for $h < k$ and $\mathcal{W}_o^{(h)} s_{x_j} = s_{x_j}^h$. Furthermore, for $j \in [\mathcal{S}]$,

$$\|s_j\|_\infty \leqslant 1, \text{ and } \|\mathbf{V}_1^{(h)} s_j\|_\infty \leqslant 1.$$

Using this, the above equation can be written as,

$$r_1[i] = s_{x_i}^0 + \sum_{h=1}^{k-1} s_{x_{i-h}}^h + \mathcal{O}(ie^{-c}) \cdot \mathbf{1}.$$

A similar computation for $i < h$ gives the required result. ■

Lemma B.2. *[Second-Layer] With the construction given in Def. A.2, the attention scores after the second layer,*

$$a_{(i,t)}^{(2)} = \begin{cases} \frac{1}{|\mathsf{M}_t^k|} - \frac{\mathcal{O}(te^{-c})}{|\mathsf{M}_t^k|^2} \text{ for } i \in \mathsf{M}_t, \\ \frac{\mathcal{O}(e^{-c})}{|\mathsf{M}_t^k|} \text{ o.w.} \end{cases} \tag{16}$$

.

Proof The product of the query and key in the second layer from Eq. 8c is given by,

$$(\mathbf{Q}_2)^\top \mathbf{K}_2 = c \sum_{h=1}^{k-1} \sum_{j=1}^{\mathcal{S}} s_j^{h-1} (s_j^h)^\top.$$

Using this the attention scores in the second layer for any key i and query t are given by,

$$a_{(i,t)}^{(2)} = \frac{\exp \langle \mathbf{K}_2 r_1[i], \mathbf{Q}_2 r_1[t] \rangle}{\sum_{j=1}^{t} \exp \langle \mathbf{K}_2 r_1[j], \mathbf{Q}_2 r_1[t] \rangle}.$$

To compute the inner product,

$$\langle \mathbf{K}_2 r_1[i], \mathbf{Q}_2 r_1[t] \rangle = \left\langle r_1[t], \mathbf{Q}_2^\top \mathbf{K}_2 r_1[i] \right\rangle$$

We know from Lemma B.1 that the embeddings after the first layer for $i \geqslant k$ are given by,

$$r_1[i] = \sum_{h=0}^{k-1} s_{x_{i-h}}^h + \mathcal{O}(ie^{-c}) \cdot \mathbf{1}.$$

Now for $t \geqslant i$,

$$\begin{aligned} (\mathbf{Q}_2)^\top \mathbf{K}_2 r_1[i] &= c \sum_{h=1}^{k-1} \sum_{j=1}^{\mathcal{S}} s_j^{h-1} (s_j^h)^\top \left[\sum_{h=0}^{k-1} s_{x_{i-h}}^h + \mathcal{O}(\exp\{-c\}). \right], \\ &= c \sum_{h=1}^{k-1} s_{x_{i-h}}^{h-1} + \mathcal{O}(\exp\{-c\}). \\ \left\langle r_1[t], (\mathbf{Q}_2)^\top \mathbf{K}_2 r_1[i] \right\rangle &= c \left\langle \sum_{h=0}^{k-1} s_{x_{t-h}}^h, \sum_{h=1}^{k-1} s_{x_{i-h}}^{h-1} \right\rangle + \mathcal{O}(\exp\{-c\}), \\ &= c \left\langle \sum_{h=1}^{k} s_{x_{t+1-h}}^{h-1}, \sum_{h=1}^{k-1} s_{x_{i-h}}^{h-1} \right\rangle + \mathcal{O}(\exp\{-c\}), \end{aligned}$$

Now we use the fact that the embeddings $s_i^{h_1}, s_j^{h_2}$ are orthogonal for $i \neq j$ or $h_1 \neq h_2$. Using this, we can write the above expression as,

$$\begin{aligned}\left\langle r_1[t], (\mathbf{Q}_2)^\top \mathbf{K}_2 r_1[i] \right\rangle &= c \sum_{h=1}^{k-1} \left\langle s^{h-1}_{x_{t+1-h}}, s^{h-1}_{x_{i-h}} \right\rangle + \mathcal{O}(\exp\{-c\}), \\ &= c \sum_{h=1}^{k-1} \mathbb{1}\{x_{i-h} = x_{t+1-h}\} + \mathcal{O}(\exp\{-c\}).\end{aligned}$$

If the k-history of $t+1$ and i match, i.e., $i \in \mathsf{M}_t^k$, then the summation is maximum at $(k-1)c^2$ otherwise it will be $\leqslant (k-2)c^2$ as there is atleast one mismatch. Using this, we can write the attention scores as,

$$\sum_{j=1}^{t} \exp \left\langle \mathbf{K}_2 r_1[j], \mathbf{Q}_2 r_1[t] \right\rangle \leqslant |\mathsf{M}_t| \exp\{(k-1)c\} + (t - |\mathsf{M}_t|) \exp\{(k-2)c\}.$$

Hence, the attention scores are given by neglecting the terms with double exponentiation (i.e., $\exp\{\exp\{-c\}\}$),

$$\begin{aligned}\text{For } i \in \mathsf{M}_t^k, \quad a^{(2)}_{(i,t)} &= \frac{\exp{(k-1)c^2}}{\sum_{j=1}^{t} \exp \left\langle \mathbf{K}_2 r_1[j], \mathbf{Q}_2 r_1[t] \right\rangle} \geqslant \frac{\exp{(k-1)c^2}}{|\mathsf{M}_t| \exp\{(k-1)c\} + (t - |\mathsf{M}_t|) \exp\{(k-2)c\}}, \\ \frac{1}{|\mathsf{M}_t|} - a^{(2)}_{(i,t)} &\leqslant \frac{(t-|\mathsf{M}_t|)\exp\{(k-2)c\}}{|\mathsf{M}_t| \left[|\mathsf{M}_t| \exp\{(k-1)c\} + (t - |\mathsf{M}_t|) \exp\{(k-2)c\} \right]} \leqslant \frac{t - |\mathsf{M}_t|}{|\mathsf{M}_t|^2} \exp\{-c\} \\ \text{For } i \notin \mathsf{M}_t^k, \quad a^{(2)}_{(i,t)} &\leqslant \frac{\exp{(k-2)c}}{|\mathsf{M}_t| \exp{(k-1)c}} = \mathcal{O}(\exp\{-c\}) \quad \text{o.w..}\end{aligned}$$

Hence, the attention scores are given by,

$$\begin{aligned}\text{For } i \in \mathsf{M}_t^k, \quad a^{(2)}_{(i,t)} &= \frac{1}{|\mathsf{M}_t^k|} - \frac{t\mathcal{O}(\exp\{-c\})}{|\mathsf{M}_t^k|}, \\ \text{For } i \notin \mathsf{M}_t^k, \quad a^{(2)}_{(i,t)} &= \mathcal{O}(\exp\{-c\}) \quad \text{o.w..}\end{aligned}$$

This completes the proof of the lemma. ■

F. Possible Extensions of The Results

In this section, we discuss the possible extensions of the results presented in the main text. First we discuss how the results can be extended to a general transformer architecture. Then we discuss the possible constructions for stationary points and beyond suffixes.

Other stationary points. Before we begin, we note that there are other possible constructions of the stationary points. In the main text, we focus on the case where one head activates for the (-h)-tokens, while the other heads are turned off. However, there can be multiple heads computing the (-h)-tokens for $h \leqslant k-1$. Since these heads are symmetric, the gradients of these heads can be shown to vanish, similar to the proof of Theorem 4.1, by leveraging this symmetry.

For example, consider the bigram MLE estimator, the consturctions is done when one head is activated which computes the $(-1)-$token and the other heads are turned off. However, it can also be the case where all the heads are activated and the $(-1)-$token is computed by all the heads as seen in Figure 3. Our proof of Theorem 4.1 works in this case too, with the arguments of symmetry the gradients of these heads will vanish. This can be shown using the same arguments as in the proof of Theorem 4.1. We give an explicit construction for this simple bigram case.

$$(\mathbf{Q}_2)^\top = \sqrt{c}\sum_{j=1}^{\mathcal{S}}\sum_{h=1}^{n-1} s_j^{0}(s_j^h)^\top, \tag{26a}$$

$$\mathbf{K}_2 = \sqrt{c}\sum_{j=1}^{\mathcal{S}}\sum_{h=1}^{n-1} s_j^h(s_j^h)^\top, \tag{26b}$$

$$\mathbf{A}_1^{(h)} = c\sum_{l=1}^{T-1} e_{l-1}e_l^\top \text{ for } h \in [n-1], \tag{26c}$$

$$\mathbf{V}_1^{(h)} = \sqrt{c}\sum_{j=1}^{\mathcal{S}} s_j s_j^\top \text{ for } h \in [n-1]. \tag{26d}$$

Here all heads are switched on and the $(-1)-$token is computed by all the heads. The query is used to compare the x_t token with all the previous heads (notice 0 in blue in the superscript for all h in $\mathbf{Q}_2$). This can be generalized to any k-gram, where the $(-h)$-token for $h < k$ is computed by different heads with varying multiplicities.

F.1. General Transformer Architecture

The results presented in the main text are given for a simplified transformer architecture. In this section, we discuss how the results can be extended to a general transformer architecture. We particularly discuss adding the value matrix in the second layer and moving beyond concatenation of head embeddings of the first layer.

Token and position embeddings. First we define some block embeddings in a dimension $\mathbb{R}^d$ and later lift them into the dimension of the transformers, i.e., nd and sequences upto length T.

$$\text{Token embeddings} \rightarrow s_0, s_1, \dots, s_{\mathcal{S}-1} \in \mathbb{R}^d$$
$$\text{Position embeddings} \rightarrow p_0, p_1, \dots, p_{T-1} \in \mathbb{R}^d.$$

The embeddings are mutually orthogonal, i.e.,

$$\begin{aligned} s_i &\perp s_j, \text{ for all } i \neq j \in [N]. \\ p_i &\perp p_j, \text{ for all } i \neq j \in [T]. \\ p_i &\perp s_j, \text{ for all } i \in [T],\ j \in [N]. \end{aligned}$$

For $h \in [k+1]$, we denote the $s_i^h, p_i^h \in \mathbb{R}^{nd}$, the lift of the block embeddings which are defined as the following,

$$\begin{aligned} s_i^h &= x\begin{bmatrix} \overbrace{0_d\ 0_d\ \cdot\ \cdot}^{\text{h blocks}} & s_i & \cdot & \cdot \end{bmatrix} \\ p_i^h &= \begin{bmatrix} \underbrace{0_d\ 0_d\ \cdot\ \cdot}_{\text{h blocks}} & p_i & \cdot & \cdot \end{bmatrix} \end{aligned}$$

The embedding layer maps to $q_0 \in \mathbb{R}^{T\times nd}$ where each row i is the embedding of i^{th} element in the sequence along with its positional embedding, i.e., $q_0[i] = s_{x_i}^0 + p_i^0 \in \mathbb{R}^{nd}$.

The attention layers. The first layer has k attention heads and also has a skip connection. For a head $h \in [k]$, let $Q_1^{(h)}, K_1^{(h)}, V_1^{(h)}$ denote the query, key and value matrices. The forward pass on q_0 writes

$$r_1 = r_0 + \sum_{h=1}^{k} \boldsymbol{\sigma}\left(r_0\,(\mathbf{Q}_1^{(h)})^\top \mathbf{K}_1^{(h)}\, r_0^\top\right)\ r_0\ \left(\mathbf{V}_1^{((h))}\right)^\top,$$

Here $\boldsymbol{\sigma}(.)$ is a row-wise softmax operator with a causal masking. Note that here we are adding in comparison to the simplified architecture where we are concatenating.

The second layer has just one head and also no skip connection (the skip connection can be included and the value matrix should be scaled appropriately so that it is ignored after normalization). Let K_2, Q_2, V_2 be the key, query and value matrices of the second layer. The forward pass for the second layer writes

$$r_2 = \boldsymbol{\sigma}\left(r_1\ (\mathbf{Q}_2)^\top \mathbf{K}_2\ r_1^\top\right)\ r_1\ \left(\mathbf{V}_2\right)^\top .$$

Now that r_2 are not normalized, we have to normalize them using a softmax operator to get the final output. In this case, the counting algorithm (MLE) implemented by the transformer cannot compute logits, hence we use an MLP layer to compute the logits.

An MLP layer. Furthermore, there is an MLP layer which computes the logarithm of a input. let M be the set of parameter m be the function implemented, $m(.) : \mathbb{R}^{\mathcal{S}} \to \mathbb{R}^{\mathcal{S}}$ does component wise logarithm $m(x) = \log(|x| + \epsilon)$ for some small epsilon.

Writing it together, the forward pass of the transformer writes,

$$\begin{aligned} r_1 &= r_0 + \sum_{h=1}^{k} \boldsymbol{\sigma}\left(r_0\ (\mathbf{Q}_1^{(h)})^\top \mathbf{K}_1^{(h)}\ r_0^\top\right)\ r_0\ \left(\mathbf{V}_1^{((h))}\right)^\top , \\ r_2 &= \boldsymbol{\sigma}\left(r_1\ (\mathbf{Q}_2)^\top \mathbf{K}_2\ r_1^\top\right)\ r_1\ \left(\mathbf{V}_2\right)^\top , \\ p_{\boldsymbol{\theta}}(x^t) &= \log r_2[t] \end{aligned}$$

This model improves on the previous simplified model by adding a value matrix in the second layer and moving beyond concatenation of head embeddings of the first layer and also using positional encoding explicitly. Since this model is a generalization of the simplified model, the representation results of the simplified model can be extended to this model. However, to show that the gradient vanishes needs a bit of work particularly for the value matrices.

For the t^{th} token, the output after the second layer of transformer writes,

$$r_2[t] = \sum_{i=1}^{t} a_{(i,t)}^{(2)}\ \mathbf{V}_2 r_1[i],$$

where the attention scores in the second layer $a_{(i,t)}^{(2)}$ for key i and query t are given by,

$$a_{(i,t)}^{(2)} = \frac{\exp\langle \mathbf{K}_2 r_1[i], \mathbf{Q}_2 r_1[t]\rangle}{\sum_{j=1}^{t} \exp\langle \mathbf{K}_2 r_1[j], \mathbf{Q}_2 r_1[t]\rangle}.$$

The final output probabilities are given by

$$p_{\boldsymbol{\theta}}(x^t) = U r_2[t] = \sum_{i=1}^{t} a_{(i,t)}^{(2)} U r_0[i] = \sum_{i=1}^{t} a_{(i,t)}^{(2)} e_{x_i}.$$

Here, we show that the gradient with respect to $\mathbf{V}_2$ vanishes. The others follow the same pattern as the simplified model and will be not be precisely computed here. The partial derivative with respect to gradient gives us,

$$\frac{\partial r_2[t]}{\partial \mathbf{V}_2} = \sum_{i=1}^{t} a_{(i,t)}^{(2)}\ I_{\mathcal{S}} \otimes r_1[i], = I_{\mathcal{S}} \otimes \bar{r}_t^{(2)}$$

Define $\boldsymbol{\theta}_*^k$ by

$$(Q_1^{(h)})^\top K_1^{(h)} = \begin{cases} c_h \sum\limits_{l=h}^{T-1} p_{l-h} p_l^\top \text{ where} & \text{for } h \in [k-1] \\ 0 \quad \text{o.w.} \end{cases}, \tag{28}$$

$$V_1^{(h)} = \begin{cases} \sum\limits_{j=1}^{\mathcal{S}} s_j^h (s_j^0)^\top, & \text{for } h \in \mathcal{N} \\ 0 \quad \text{o.w.} \end{cases}, \tag{29}$$

$$(Q_2)^\top K_2 = c \sum_{j=1}^{\mathcal{S}} \sum_{h \in \mathcal{N}} s_j^{h-1} (s_j^h)^\top, \tag{30}$$

$$V_2 = \sum_{j=1}^{\mathcal{S}} e_j (s_j^0)^\top. \tag{31}$$

Intuitively, this is just an expansion on simplified transformer with lifting in the first layer and multiplication with value made explicit. Recalling $\bar{s}_t^0 = \frac{1}{|\mathsf{M}_t^k|} \sum_{i \in \mathsf{M}_t^k} s_{x_t}^0$, using this,

$$\bar{r}_t^{(1)} = \sqrt{c}\bar{s}_t^0 + \sum_{h=1}^{k-1} s_{x_{t+1-h}}^h + \mathcal{O}(\exp\{-c\}).$$

At the limit $T \to \infty$, $\bar{s}_t^0$ depends only on the context and k-history of the token $t+1$ and the other summation only depends on the k-history of the token $t+1$. Hence, the gradient vanishes as $c \to \infty$. This gives a glimpse to how the computation can be extended to a general transformer architecture.

F.2. Other Constructions for Stationary points and Beyond Contigous history

Note that we only consider estimators conditioned on the suffices. Consider again the simple example of bigram, which is conditioning on a single element in the n-history. There can be many such estimators, for example, conditioning on the first element in the n-history $p(x_n = .|x_{n-1})$, the second element in the n-history $p(x_n = .|x_{n-2})$, and so on. It is natural to wonder and extend our results to all these possible estimators.

To generalize this, we define $\mathcal{N}-$MLE estimators for some set $\mathcal{N} \subseteq [n-1]$. For $\mathcal{N}$, we define a subsequence $x_{\mathcal{N}}^t = (x_{t-h})_{h \in \mathcal{N}}$ as $\mathcal{N}$-history and the $\mathcal{N}$-MLE estimator is given by,

$$\frac{\sum\limits_{l=0}^{T} \mathbb{1}\{x_{\mathcal{N}}^l = x_{\mathcal{N}}^{t+1}\} \, \mathbb{1}\{x_l = i\}}{\sum\limits_{l=0}^{T} \mathbb{1}\{x_{\mathcal{N}}^l = x_{\mathcal{N}}^{t+1}\}}.$$

which compares the $\mathcal{N}$-history of the token $t+1$ with the $\mathcal{N}$-history of all the tokens in the sequence.

There is a small technical challenge for a transformer to implement this. To compare the $\mathcal{N}$-history, it must compare the $(-h)$-token of i with the $(-h)$-token $(t{+}1)$ (not t) for $h \in \mathcal{N}$. This essentially requires comparing the $(-h)$-token of i with the $(-(h-1))$-token t. As a result, an induction head must attend to the $(-(h-1))$-token in the first layer. When $\mathcal{N}$ forms a suffix, this step isn't explicitly necessary since $h-1$ will also be in $\mathcal{N}$. Otherwise, it must be computed explicitly using additional heads or changing the attention matrix specifically depending on the last token.

When the inputs are of fixed length the later approach could be used. We specify the construction first and then give the intuition on why it would work.

$$(\mathbf{Q}_2)^\top = \sqrt{c}\sum_{j=1}^{\mathcal{S}}\sum_{h\in\mathcal{N}} s_j^h (s_j^h)^\top, \tag{32a}$$

$$\mathbf{K}_2 = \sqrt{c}\sum_{j=1}^{\mathcal{S}}\sum_{h\in\mathcal{N}} s_j^h (s_j^h)^\top, \tag{32b}$$

$$\mathbf{A}_1^{(h)} = c\sum_{l=h}^{T-2} e_{l-h}e_l^\top + e_{T-h+1}e_{T-1}^\top \text{ for } h\in\mathcal{N}, \tag{32c}$$

$$\mathbf{V}_1^{(h)} = \begin{cases}\sqrt{c}\sum_{j=1}^{\mathcal{S}} s_j s_j^\top \text{ for } h\in\mathcal{N}, \\ 0 \quad \text{o.w.}\end{cases}. \tag{32d}$$

The main modification is in the attention matrix of the first layer, i.e., now head h computes $(-h)$ token for $i \neq T$ but $(-(h-1))$ token for $i = T$. The last token acts as a special token (Nichani et al., 2024) and the attention matrix is modified accordingly. The gradients of the heads will vanish similar to the proof of Theorem 4.1. However the major drawback is that it only works for fixed length inputs and does not even work for shorter sequence lengths. This is not the case for the previous constructions for suffix.

G. Derivatives of the self attention map.

We calculate the derivatives of the masked self-attention map with respect to key, value, query and the input embeddings. Let $q \in \mathbb{R}^{T\times d}$ be the input embeddings, $Q, K, V \in \mathbb{R}^{d\times d}$ be the query, key and value matrices. Let q_i denote the i^{th} row of q. For any t, the output embeddings of the t^{th}-token of the transformer layer can be written as

$$q_t^+ = \sum_{i=1}^{t} p_i V q_i,$$
$$p_i = \frac{\exp\langle Kq_i, Qq_t\rangle}{\sum_{j=1}^{t}\exp\langle Kq_j, Qq_t\rangle}.$$

Lemma G.1. *For a self attention map, defined by*

$$q_t^+ = \sum_{i=1}^{t} p_i V q_i, \quad \textit{where} \quad p_i = \frac{\exp\langle Kq_i, Qq_t\rangle}{\sum_{j=1}^{t}\exp\langle Kq_j, Qq_t\rangle}.$$

Define

$$\bar{q} = \sum_{i=1}^{t} p_i q_i.$$

The partial derivatives are given by,

$$
\begin{aligned}
\frac{\partial q_t^+}{\partial V} &= \sum_{i=1}^{t} p_i \, \left(q_i^\top \otimes I_d\right) = \bar{q}^\top \otimes I_d, \\
\frac{\partial q_t^+}{\partial K} &= \sum_{j=1}^{t} p_j \, (V q_j) \otimes \mathrm{vec}\left(Q q_t (q_j - \bar{q})^\top\right)^\top, \\
\frac{\partial q_t^+}{\partial Q} &= \sum_{j=1}^{t} p_j \, (V q_j) \otimes \mathrm{vec}\left(K (q_j - \bar{q}) q_t^\top\right)^\top, \\
\textit{For } i \neq t, \; \frac{\partial q_t^+}{\partial q_i} &= p_i \, V + p_i \, (V(q_i - \bar{q})) \otimes \left(K^\top Q q_t\right)^\top, \\
\frac{\partial q_t^+}{\partial q_t} &= p_t \, V + p_t \, (V(q_t - \bar{q})) \otimes \left(K^\top Q q_t\right)^\top + \sum_{j=1}^{t} p_j \, (V q_j) \otimes \left(Q^\top K (q_j - \bar{q})\right)^\top.
\end{aligned}
$$

Proof First taking the derivative with respect to the value V gives us,

$$
\frac{\partial q_t^+}{\partial V} = \sum_{i=1}^{t} p_i \, \left(q_i^\top \otimes I_d\right).
$$

The derivative wrt to q_i gives us

$$
\begin{aligned}
\frac{\partial q_t^+}{\partial q_i} &= p_i \, V + \sum_{j=1}^{t} (V q_j) \otimes \frac{\partial p_j}{\partial q_i}, \\
\frac{\partial q_t^+}{\partial K} &= \sum_{j=1}^{t} (V q_j) \otimes \frac{\partial p_j}{\partial K}, \\
\frac{\partial q_t^+}{\partial Q} &= \sum_{j=1}^{t} (V q_j) \otimes \frac{\partial p_j}{\partial Q}.
\end{aligned}
$$

To compute the derivative of p wrt q, Q, K, we begin with definition of intermediate functions,

$$
\begin{aligned}
g : \mathbb{R}^t \to \mathbb{R}^t \quad \text{where} \quad g(x) = \frac{\exp x_i}{\sum_{j=1}^{t} \exp x_j}, \\
h : \mathbb{R}^{d \times d} \times \mathbb{R}^{d \times d} \times \mathbb{R}^{t \times d} \to \mathbb{R}^t, \quad \text{where} \quad h(K, Q, q) = \left(\langle K q_i, Q q_t \rangle\right)_{i=1}^{t}.
\end{aligned}
$$

Using the above definitions, p can be written as

$$
p = g(h(K, Q, q)).
$$

The partial derivative of g writes

$$
\begin{aligned}
\frac{\partial g_j}{\partial x_k} &= g_j \mathbb{1}\,(k = j) - g_k g_j, \\
\frac{\partial h_k}{\partial q_i} &= \begin{cases} \mathbb{1}\{k = i\} \left(K^\top Q q_t\right)^\top, & \text{for } i \neq t, \\ (Q^\top K q_k)^\top + \mathbb{1}\{k = t\} \left(K^\top Q q_t\right)^\top, & \text{for } i = t. \end{cases} \qquad , \\
&= \mathbb{1}\{k = i\} \left(K^\top Q q_t\right)^\top + \mathbb{1}\{i = t\} (Q^\top K q_k)^\top. \\
\frac{\partial h_k}{\partial K} &= \mathrm{vec}\left(Q \, q_t q_k^\top\right)^\top, \\
\frac{\partial h_k}{\partial Q} &= \mathrm{vec}\left(K \, q_k q_t^\top\right)^\top.
\end{aligned}
$$

Using the chain rule to take the derivative gives us,

$$\begin{aligned}
\frac{\partial p_j}{\partial q_i} &= \sum_{k=1}^{t} \frac{\partial g_j}{\partial h_k}\,\frac{\partial h_k}{\partial q_i}, \\
&= \sum_{k=1}^{t} \left[p_j\,\mathbb{1}\left(k=j\right) - p_k p_j\right] \left[\mathbb{1}\{k=i\}\left(K^\top Q q_t\right)^\top + \mathbb{1}\{i=t\}(Q^\top K q_k)^\top\right], \\
&= p_j\left[\mathbb{1}\{j=i\}\left(K^\top Q q_t\right)^\top + \mathbb{1}\{i=t\}(Q^\top K q_j)^\top\right] \\
&\qquad\qquad - p_i p_j\left(K^\top Q q_t\right)^\top - \mathbb{1}\{i=t\} p_j \sum_{k=1}^{t} p_k (Q^\top K q_k)^\top. \\
\frac{\partial p_j}{\partial K} &= \sum_{k=1}^{t} \frac{\partial p_j}{\partial h_k}\,\frac{\partial h_k}{\partial K} = \sum_{k=1}^{t} \left[p_j \mathbb{1}\left(k=j\right) - p_k p_j\right] \mathrm{vec}\left(Q\,q_t q_k^\top\right)^\top, \\
&= p_j \mathrm{vec}\left(Q\,q_t q_j^\top\right)^\top - p_j \sum_{k=1}^{t} p_k \mathrm{vec}\left(Q\,q_t q_k^\top\right)^\top,
\end{aligned}$$

Define $\widehat{q} = \sum_{k=1}^{t} p_k q_k$,

$$\frac{\partial p_j}{\partial K} = p_j \mathrm{vec}\left(Q q_t (q_j - \widehat{q})^\top\right)^\top.$$

$$\begin{aligned}
\frac{\partial p_j}{\partial Q} &= \sum_{k=1}^{t} \frac{\partial p_j}{\partial h_k}\,\frac{\partial h_k}{\partial Q} = \sum_{k=1}^{t} \left[p_j \mathbb{1}\left(k=j\right) - p_k p_j\right] \mathrm{vec}\left(K\,q_k q_t^\top\right)^\top, \\
&= p_j \mathrm{vec}\left(K\,q_j q_t^\top\right)^\top - p_j \sum_{k=1}^{t} p_k \mathrm{vec}\left(Q\,q_k q_t^\top\right)^\top, \\
&= p_j \mathrm{vec}\left(K(q_j - \widehat{q}) q_t^\top\right)^\top.
\end{aligned}$$

Substituting the following derivates,

$$\begin{aligned}
\frac{\partial p_j}{\partial K} &= p_j \mathrm{vec}\left(Q q_t (q_j - \widehat{q})^\top\right)^\top, \\
\frac{\partial p_j}{\partial Q} &= p_j \mathrm{vec}\left(K(q_j - \widehat{q}) q_t^\top\right)^\top.
\end{aligned}$$

$$\begin{aligned}
\frac{\partial q_t^+}{\partial K} &= \sum_{j=1}^{t} (V q_j) \otimes \frac{\partial K}{\partial p_j} = \sum_{j=1}^{t} p_j\,(V q_j) \otimes \mathrm{vec}\left(Q q_t (q_j - \widehat{q})^\top\right)^\top, \\
\frac{\partial q_t^+}{\partial Q} &= \sum_{j=1}^{t} (V q_j) \otimes \frac{\partial p_j}{\partial Q} = \sum_{j=1}^{t} p_j\,(V q_j) \otimes \mathrm{vec}\left(K(q_j - \widehat{q}) q_t^\top\right)^\top,
\end{aligned}$$

Coming to the derivatives wrt q_i, we have,

$$\begin{aligned}
\frac{\partial p_j}{\partial q_i} &= p_j\left[\mathbb{1}\{j=i\}\left(K^\top Q q_t\right)^\top + \mathbb{1}\{i=t\}(Q^\top K q_j)^\top\right] \\
&\qquad\qquad - p_i p_j\left(K^\top Q q_t\right)^\top - \mathbb{1}\{i=t\} p_j \sum_{k=1}^{t} p_k (Q^\top K q_k)^\top.
\end{aligned}$$

Writing the above expression on case by case basis, we get,

$$\frac{\partial p_j}{\partial q_i} = \begin{cases} p_t(1-p_t)\left(K^\top Q q_t\right)^\top + p_j\left(Q^\top K(q_j - \widehat{q})\right)^\top, \text{ when } i = j = t, \\ p_j(1-p_j)\left(K^\top Q q_t\right)^\top, \text{ when } i = j \neq t, \\ -p_i p_j\left(K^\top Q q_t\right)^\top, \text{ when } i \neq j \quad \text{and} \quad i \neq t, \\ -p_t p_j\left(K^\top Q q_t\right)^\top + p_j\left(Q^\top K(q_j - \widehat{q})\right)^\top, \text{ when } i = t \quad \text{and} \quad i \neq j, \end{cases}$$

First computing the derivative wrt to q_i for $i \neq t$, we get,

$$\begin{aligned} \frac{\partial q_t^+}{\partial q_i} &= p_i\, V + \sum_{j=1}^{t} (V q_j) \otimes \frac{\partial p_j}{\partial q_i}, \\ &= p_i\, V + p_i\, (V q_i) \otimes \left(K^\top Q q_t\right)^\top - p_i \sum_{j=1}^{t} p_j\, (V q_j) \otimes \left(K^\top Q q_t\right)^\top, \\ &= p_i\, V + p_i\, (V q_i) \otimes \left(K^\top Q q_t\right)^\top - p_i\, (V \widehat{q}) \otimes \left(K^\top Q q_t\right)^\top, \\ &= p_i\, V + p_i\, (V(q_i - \widehat{q})) \otimes \left(K^\top Q q_t\right)^\top. \end{aligned}$$

$$\begin{aligned} \frac{\partial q_t^+}{\partial q_i} &= p_t\, V + p_t\, (V q_t) \otimes \left(K^\top Q q_t\right)^\top - p_t \sum_{j=1}^{t} p_j\, (V q_j)\left(K^\top Q q_t\right)^\top + \sum_{j=1}^{t} p_j\, (V q_j) \otimes \left(Q^\top K(q_j - \widehat{q})\right)^\top, \\ &= p_t\, V + p_t\, (V(q_t - \widehat{q})) \otimes \left(K^\top Q q_t\right)^\top + \sum_{j=1}^{t} p_j\, (V q_j) \otimes \left(Q^\top K(q_j - \widehat{q})\right)^\top. \end{aligned}$$

This proves the lemma. ■

H. Higher Order Markov Chain

Definition H.1 (Higher Order Markov Process). A Markov process p_τ of order k generates a sequence of random variables $x_1, x_2, \ldots \in [\mathcal{S}]$ such that the conditional distribution of x_{t+1} given x^t depends only on x_{t-k+1}^t, i.e., for any $t \geqslant k$,

$$p_\tau\left(x_{t+1} \,\middle|\, x^t\right) = p_\tau\left(x_{t+1} \,\middle|\, x_{t-k+1}^t\right).$$

Now, we define the transition tensor with is given by p_τ.

Definition H.2 (Transition Tensor). The transition tensor of a Markov process of order k is a k-dimensional tensor $\mathcal{P}$ such that

$$\mathcal{P}_{i^{k+1}} := p_\tau\left(x_{k+1} = i_{k+1} \,\middle|\, x^k = i^k\right).$$

Now, we lift the process into the higher dimension and write it as a simple Markov process of order 1. We create a Markov chain Ξ on the space $[S]^k$ of order 1 Markov chain with transition matrix $\mathcal{T}$. Index the states by $i^k \in [S]^k$. Now the transition probabilities for this are defined as

$$\mathcal{T}\left(Y^k \middle| i^k\right) = \begin{cases} \neq 0 & \text{if } Y^k = (i_2^k, i_{k+1}), \text{ for some } i_{k+1} \in [S], \\ = 0 & \text{otherwise} \end{cases}.$$

The transition probabilities are explicitly given by

$$\mathcal{T}\left(i_2^{k+1} \middle| i^k\right) = \mathcal{P}_{i^{k+1}}.$$

Stationary distribution. Let $\pi \in \mathbb{R}^{|\mathcal{S}|^k}$ denote the stationary distribution of the chain Ξ, i.e.,

$$\pi^\top \mathcal{T} = \pi^\top.$$

In the summation form , for the coordinate indexed by i_2^{k+1}, we get that,

$$\sum_{j^k} \pi_{j^k} \cdot \mathcal{T}\left(i_2^{k+1} \,\middle|\, j^k\right) = \pi_{i_2^{k+1}}.$$

Note that $\mathcal{T}\left(i_2^{k+1} \,\middle|\, j^k\right) = 0$ whenever $j_2^k \neq i_2^k$. Using this, we get,

$$\begin{aligned}
\sum_{j^k} \pi_{j^k} \cdot \mathcal{T}\left(i_2^{k+1} \,\middle|\, j^k\right) &= \sum_{j^k} \pi_{j^k} \cdot \mathcal{T}\left(i_2^{k+1} \,\middle|\, j^k\right)\left(\mathbb{1}\left\{j_2^k \neq i_2^k\right\} + \mathbb{1}\left\{j_2^k = i_2^k\right\}\right), \\
&= \sum_{j^k} \pi_{j^k} \cdot \mathcal{T}\left(i_2^{k+1} \,\middle|\, j^k\right) \mathbb{1}\left\{j_2^k = i_2^k\right\}, \\
&= \sum_{j_1} \pi_{(j_1, i_2^k)} \mathcal{T}\left(i_2^{k+1} \,\middle|\, (j_1, i_2^k)\right), \\
&= \sum_{i_1} \pi_{i^k} \mathcal{T}\left(i_2^{k+1} \,\middle|\, i^k\right).
\end{aligned}$$

Using this expansion, we get the following stationarity condition,

$$\pi_{i_2^{k+1}} = \sum_{i_1} \pi_{i^k} \mathcal{T}\left(i_2^{k+1} \,\middle|\, i^k\right). \tag{33}$$

H.1. Generating Sequences with The Markov Chain Ξ

In this subsection, we show the subsequences of length k follow the stationary distribution when generated from the Markov Chain Ξ. To generate sequences from the Markov Chain Ξ, we do the following generation,

a) Generate a sequence of length k sampling from the stationary distribution π (say i^k)

b) Sample the next token from the distribution $\mathcal{P}_{i^k,(.)}$ say i_{k+1}

c) After generating a sequence i^t of length $t > k$, the next token is generated by the sampling from the distribution $\mathcal{P}_{i^t_{t-k+1},(.)}$

Lemma H.3 (Stationarity of k-tuples)**.** *Let x^t be the random variable representing the sequences generated from Ξ, the distribution of subsequence x^l_{l-k+1} of length k is given by π.*

Proof We will prove this using induction. For the base case, observe that for $l = k$, the value is sampled from π by construction. Use the induction hypothesis, any k length subsequence of x^l obeys the law given by π. It remains to show that x^{l+1}_{l-k+2} are distributed according to π.

$$\begin{aligned}
p_\mathcal{T}\left(x^{l+1}_{l-k+2} = i^{l+1}_{l-k+2}\right) &= \sum_{i^{l-k+1}} p_\mathcal{T}\left(x^{l+1} = i^{l+1}\right), \\
&= \sum_{i^{l-k+1}} p_\mathcal{T}\left(x^l = i^l\right) p_\mathcal{T}\left(x_{l+1} = i_{l+1} \,\middle|\, x^l = i^l\right)
\end{aligned}$$

Using the Markov property of the sequence generation, we have,

$$p_\mathcal{T}\left(x_{l+1} = i_{l+1} \,\middle|\, x^l = i^l\right) = p_\mathcal{T}\left(x_{l+1} = i_{l+1} \,\middle|\, x^l_{l-k+1} = i^l_{l-k+1}\right).$$

Substuting this expression, we have,

$$\begin{aligned} p_{\mathcal{T}}\left(x_{l-k+2}^{l+1}=i_{l-k+2}^{l+1}\right) &= \sum_{i^{l-k+1}} p_{\mathcal{T}}\left(x^{l}=i^{l}\right) \mathcal{P}\left(x_{l+1}=i_{l+1} \,\middle|\, x_{l-k+1}^{l}=i_{l-k+1}^{l}\right), \\ &= \sum_{i_{l-k+1}} \sum_{i^{l-k}} p_{\mathcal{T}}\left(x^{l}=i^{l}\right) p_{\mathcal{T}}\left(x_{l+1}=i_{l+1} \,\middle|\, x_{l-k+1}^{l}=i_{l-k+1}^{l}\right), \\ &= \sum_{i_{l-k+1}} \left[\sum_{i^{l-k}} p_{\mathcal{T}}\left(x^{l}=i^{l}\right)\right] p_{\mathcal{T}}\left(x_{l+1}=i_{l+1} \,\middle|\, x_{l-k+1}^{l}=i_{l-k+1}^{l}\right). \end{aligned}$$

Using the fact that $\sum_{i^{l-k}} p_{\mathcal{T}}\left(x^{l}=i^{l}\right) = p_{\mathcal{T}}\left(x_{l-k+1}^{l}=i_{l-k+1}^{l}\right)$. Using the induction hypothesis, we have that $p_{\mathcal{T}}\left(x_{l-k+1}^{l}=i_{l-k+1}^{l}\right)=\pi_{i_{l-k+1}^{l}}$.

$$\begin{aligned} p_{\mathcal{T}}\left(x_{l-k+2}^{l+1}=i_{l-k+2}^{l+1}\right) &= \sum_{i_{l-k+1}} \pi_{i_{l-k+1}^{l}} p_{\mathcal{T}}\left(x_{l+1}=i_{l+1} \,\middle|\, x_{l-k+1}^{l}=i_{l-k+1}^{l}\right), \\ &= \sum_{i_{l-k+1}} \pi_{i_{l-k+1}^{l}} \mathcal{T}\left(i_{l-k+2}^{l+1} \,\middle|\, i_{l-k+1}^{l}\right). \end{aligned}$$

Using the Eq. (33),

$$\begin{aligned} p_{\mathcal{T}}\left(x_{l-k+2}^{l+1}=i_{l-k+2}^{l+1}\right) &= \sum_{i_{l-k+1}} \pi_{i_{l-k+1}^{l}} \mathcal{T}\left(i_{l-k+2}^{l+1} \,\middle|\, i_{l-k+1}^{l}\right), \\ &= \pi_{i_{l-k+2}^{l+1}}. \end{aligned}$$

This proves the hypothesis for length $l+1$. Hence, by induction, the hypothesis holds for any length. ■

Lemma H.4 (Stationarity of sub-k-tuples)**.** *For any $t \geqslant 0$ and $l \leqslant k$, we have the following shift invariant property for the marginals,*

$$p_{\mathcal{T}}\left(x^{l}=i^{l}\right)=p_{\mathcal{T}}\left(x_{t}^{t+l-1}=i^{l}\right)$$

Proof Using Lemma H.3, we have that the distribution of the subsequence x_t^{t+k-1} is given by π for any $t \geqslant 0$, i.e.,

$$p_{\mathcal{T}}\left(x^{k}=i^{k}\right)=p_{\mathcal{T}}\left(x_{t}^{t+k-1}=i^{k}\right)=\pi_{i^{k}}.$$

Summing the above expression over all possible values of i_{l+1}^{k}, we get,

$$\sum_{i_{l+1}^{k}} p_{\mathcal{T}}\left(x^{k}=i^{k}\right)=\sum_{i_{l+1}^{k}} p_{\mathcal{T}}\left(x_{t}^{t+k-1}=i^{k}\right)=\sum_{i_{l+1}^{k}} \pi_{i^{k}}.$$

We know that

$$\begin{gathered} \sum_{i_{l+1}^{k}} p_{\mathcal{T}}\left(x^{k}=i^{k}\right)=p_{\mathcal{T}}\left(x^{l}=i^{l}\right), \\ \sum_{i_{l+1}^{k}} p_{\mathcal{T}}\left(x_{t}^{t+k-1}=i^{k}\right)=p_{\mathcal{T}}\left(x_{t}^{t+l-1}=i^{l}\right) \end{gathered}$$

This proves the lemma. ■

H.2. Lower-order Conditional Probabilities

The following lemma show that the lower order conditional probabilities are time homogenous and are given by the stationary distribution π.

Lemma H.5. *For the sequences generated by the markov chain Ξ, for $l \leqslant k$, we have,*

$$p_\tau\left(x_{T+1} = . \,\middle|\, x_{T-l+1}^T = i^l\right) = p_\tau\left(x_{l+1} = . \,\middle|\, x^l = i^l\right) = \frac{\sum_{i_2^{k-l}} \pi_{i_2^{k+1}}}{\sum_{i^{k-l}} \pi_{i^k}}$$

Proof For $l = k$, this holds due the Markov property of order k and

$$p_\tau\left(x_{T+1} = . \,\middle|\, x_{T-k+1}^T = i^k\right) = p_\tau\left(x_{k+1} = . \,\middle|\, x^k = i^k\right).$$

Now, for $l < k$, we have,

$$p_\tau\left(x_{T+1} = i_{l+1} \,\middle|\, x_{T-l+1}^T = i^l\right) = \frac{p_\tau\left(x_{T-l+1}^{T+1} = i^{l+1}\right)}{p_\tau\left(x_{T-l+1}^T = i^l\right)},$$

Using the shift-invariant property of the marginals, we have that,

$$p_\tau\left(x_{T-l+1}^{T+1} = i^{l+1}\right) = p_\tau\left(x^{l+1} = i^{l+1}\right) = \sum_{i_{l+2}^k} \pi_{i^k},$$
$$p_\tau\left(x_{T-l+1}^T = i^l\right) = p_\tau\left(x^l = i^l\right) = \sum_{i_{l+1}^k} \pi_{i^k}.$$

This proves the lemma. ■

Conditional Probability for Any Subset of Sequence. Previously, we defined it for contiguous history; now, we can extend the definition to any subset of sequence as well. Consider any set $\mathbf{K} \subseteq [0, \mathcal{S} - 1]$, $[\mathcal{S}] - \mathbf{K}$ is an ordered set defined as $\{k - i : i \in \mathbf{K}\}$. We define the sequence $x^{[\mathcal{S}]-\mathbf{K}}$ as the ordered set $\{x_i : i \in [\mathcal{S}] - \mathbf{K}\}$. We define the conditional probability of the next token given the sequence $x^{[\mathcal{S}]-\mathbf{K}}$ as

$$\mathcal{P}_{[\mathcal{S}]-\mathbf{K}}\left(i_{[\mathcal{S}]-\mathbf{K}}, i_{k+1}\right) := p_\tau\left(x_{k+1} = i_{k+1} \,|\, x^{[\mathcal{S}]-\mathbf{K}} = i^{[\mathcal{S}]-\mathbf{K}}\right),$$
$$= \frac{p_\tau\left(x_{k+1} = i_{k+1}, x^{[\mathcal{S}]-\mathbf{K}} = i^{[\mathcal{S}]-\mathbf{K}}\right)}{p_\tau\left(x^{[\mathcal{S}]-\mathbf{K}} = i^{[\mathcal{S}]-\mathbf{K}}\right)},$$
$$= \frac{\sum_{i^{([\mathcal{S}]-\mathbf{K})^c}} p_\tau\left(x^{k+1} = i^{k+1}\right)}{\sum_{i^{([\mathcal{S}]-\mathbf{K})^c}} p_\tau\left(x^k = i^k\right)}.$$

Consistency of the sub-k counting estimators. Here, we use the ergodic properties of the chain Ξ to show that the lower order estimators convergence as $T \to \infty$.

Lemma H.6. *Consider the $l{+}1$-gram estimator $\widehat{p}_l$ for $0 < l \leqslant k-1$, then as $T \to \infty$, $\widehat{p}_l(x^T) \to p_\tau(.|x^l)$.*

Proof Using the ergodicity of the chain Ξ (Penev, 1991), the $k{+}1-$gram estimator converges to the stationary distribution as it equivalent to counting the empirical frequency on the chain of higher order. Now we write the $l-$gram estimator in the following way:

$$\widehat{p}_l(i_{k+1}|i_{k-l+1}^k) = \frac{\sum_{j=1}^T \mathbb{1}\{x_{j-l+1}^{j+1} = i_{k-l+1}^{k+1}\}}{\sum_{j=1}^T \mathbb{1}\{x_{j-l+1}^j = i_{k-l+1}^k\}}$$

Now we can write the following expression as

$$\sum_{j=1}^{T} \mathbb{1}\{x_{j-l+1}^{j+1} = i_{k-l+1}^{k+1}\} = \sum_{i_2^l} \sum_{j=1}^{T} \mathbb{1}\{x_{j-k+1}^{j+1} = i_2^{k+1}\},$$

$$\sum_{j=1}^{T} \mathbb{1}\{x_{j-k+1}^{j} = i_{k-l+1}^{k+1}\} = \sum_{i^l} \sum_{j=1}^{T} \mathbb{1}\{x_{j-k+1}^{j} = i^{k}\}.$$

Now, using the ergodicity of the lifted markov chain, we can say that

$$\frac{\sum_{j=1}^{T} \mathbb{1}\{x_{j-k+1}^{j+1} = i_2^{k+1}\}}{T} \overset{t\to\infty}{\to} \pi_{i_2^{k+1}},$$

$$\sum_{i_2^l} \frac{\sum_{j=1}^{T} \mathbb{1}\{x_{j-k+1}^{j+1} = i_2^{k+1}\}}{T} \overset{t\to\infty}{\to} \sum_{i_2^l} \pi_{i_2^{k+1}}$$

Using the similar argument, the ratio can now be written as

$$\frac{\sum_{j=1}^{T} \mathbb{1}\{x_{j-l+1}^{j+1} = i_{k-l+1}^{k+1}\}}{\sum_{j=1}^{T} \mathbb{1}\{x_{j-l+1}^{j} = i_{k-l+1}^{k}\}} = \frac{\sum_{i_2^l} \pi_{i_2^{k+1}}}{\sum_{i^l} \pi_{i^k}} = p_\tau(x_{l+1} = i_{k+1} | x^l = i_{k-l+1}^{k})$$

The last equality flows from the lemma H.5. ■

I. More Experiments

I.1. Experiments with Attention Only Transformer

In this section, we repeat the main experiment in the paper with the common non-disentangled transformer architecture, demonstrating the generality of our results. Figure 5, shows the evolution of attention matrices during training with both one-hot and learned embeddings for the general architecture.

For the transformer experiments, we use vocabulary size $\mathcal{S} = 3$. The length of the input sequences is $T = 32$, and the sequences are sampled from an in-context tri-gram language model, i.e., $n = 3$. The transition matrices for a fixed context are sampled from a Dirichlet prior with $\alpha = 1$. The embedding dimension is set to $d = \mathcal{S} + T$, to be consistent in both setups, and we use one-hot or learned embeddings for both positional and semantic embeddings in different experiments. The transformer is trained with Adam with a weight decay of 0.0001 for 4096 iterations, with a constant learning rate of 0.005 and a batch size of 128. The test loss is evaluated over 2^{16} test sequences.

I.2. The Contiguous Solutions are Preferred during Training

In Figure 6, we repeat the procedure in Figure 3 for different seeds and plot the attention heads in the second plateau. Through all the experiments transformer more often attends to the (-1)-token first, rather than the (-2)-token. Recall that the underlying mechanism behind the sub-n-gram estimators is checking if the history of the token x_{T+1} and x_j matches and adding x_j if they do. We conjecture that, since the token x_T is always provided through the skip connection, it is easier to learn to match x_T with x_{j-1} than, for example, x_{T-1} and x_{j-2}.

I.3. Plot of norms of the gradients along the trajectory

In Figures 7, 8, we plot the norm of the gradient during training along the similar lines as (Odonnat et al., 2025). The plots demonstrate how the norm of gradients stays low during the plateau stages and spikes during the jump between the plateaus.

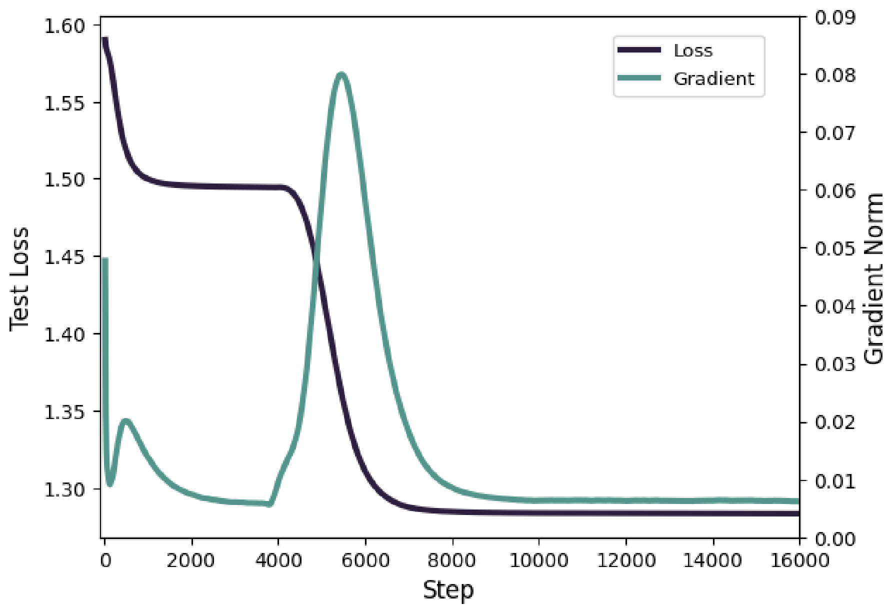


Figure 7: **Norms of gradients for simplified transformer.** A two layer simplified transformer for $S = 5$ and a sequence length of $L = 128$ from a 3-gram language model. The plots show that the norm of gradients stays low during the plateau stages and spikes during the jump between the plateaus.

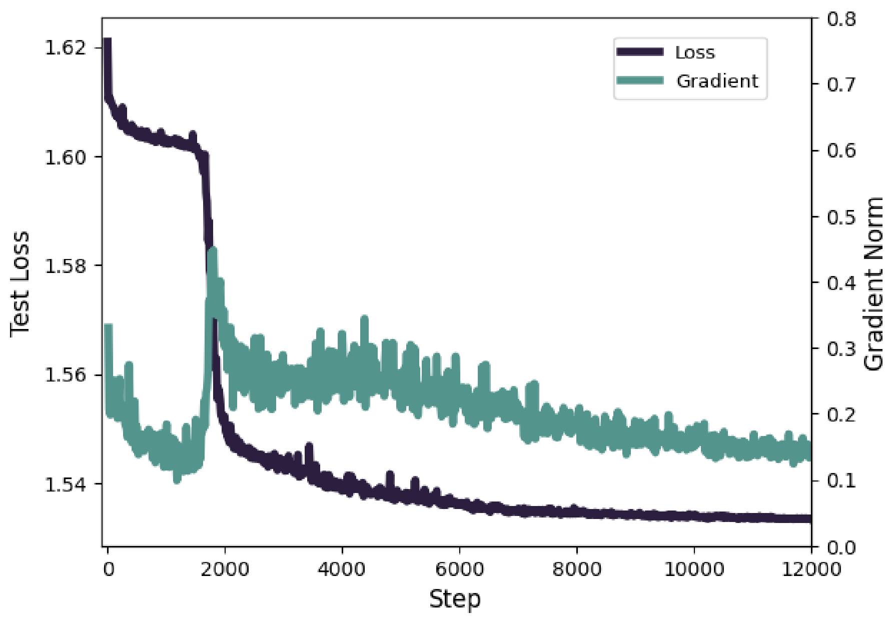


Figure 8: **Norms of gradients for transformers with MLP layers.** A two layer simplified transformer for $S = 5$ and a sequence length of $L = 64$ from a 3-gram language model. The plots show that the norm of gradients stays low during the plateau stages and spikes during the jump between the plateaus. However, there is only a single intermediate plateau, unlike attention-only transformers, as a single head attends to both the $(-1, -2)$-tokens, and it emerges after the plateau. The plateau corresponds to the unigram.

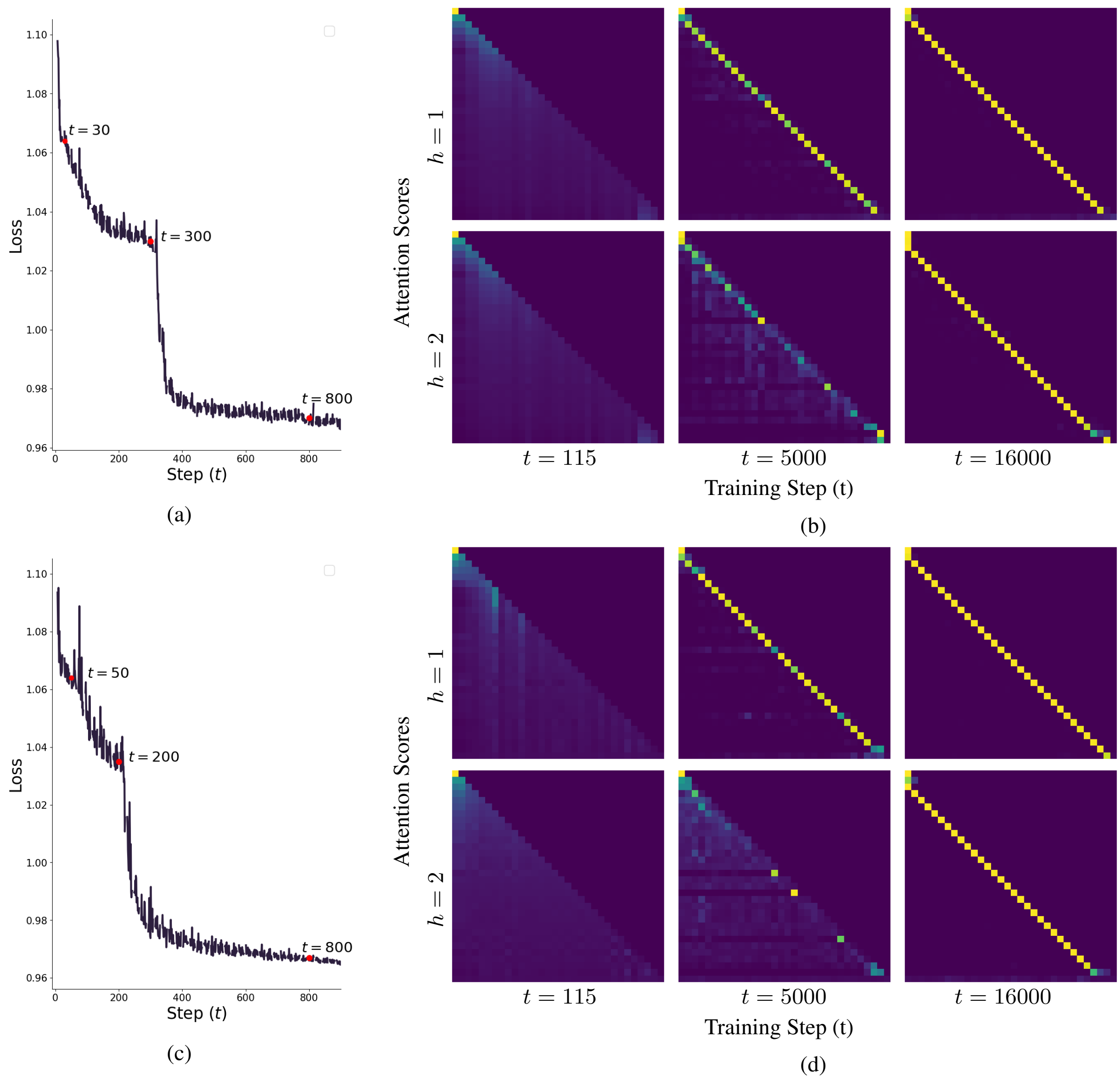
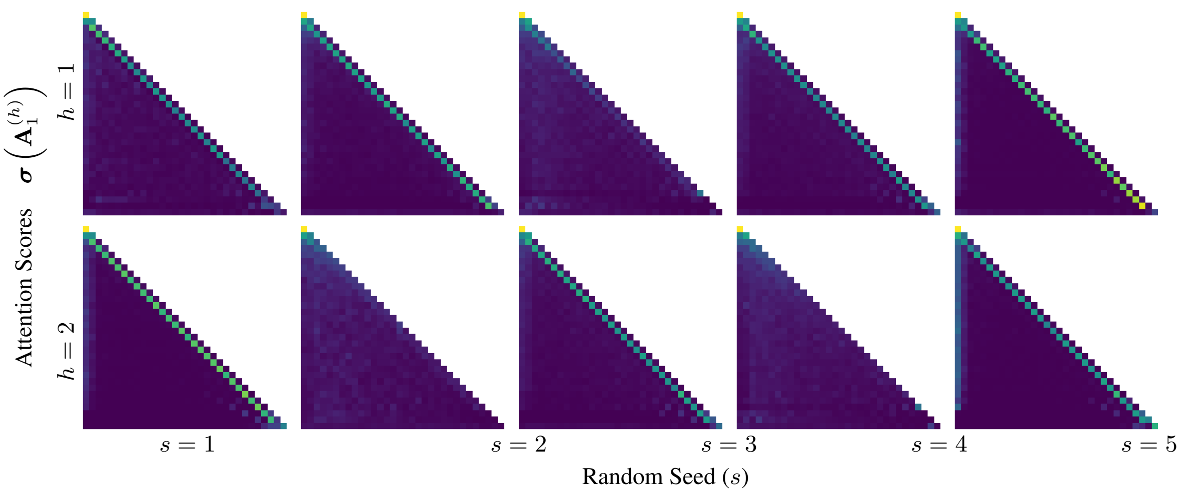


Figure 5: The evolution of the attention heads in the first layer during training of an attention-only transformer with one-hot (a-b) and learned embeddings(c-d). (a-c) Progression of the test loss during training. The highlighted points are the iterations on the plateaus for which we demonstrate the attention matrices. (b-d) The evolution of attention scores of the heads of the transformer during training representing the tokens it is attending. First, both of the attention heads attend to all the previous tokens uniformly, i.e., the induction heads are not formed. At the second plateau, they both attend to the previous token, or one head is not formed yet while the other attends to the previous token. Finally, as the model escapes this plateau, the second attention head learns to attend to (-2)-token at the end of training.



(a)

Figure 6: Attention maps of the two heads in the second plateau for different random seeds denoted by s. It shows how the transformers attends to (-1)-token first and never attends the (-2)-token before attending (-1)-token across 5 random seeds.