DYNAMIC ANALYSIS OF LARGE LANGUAGE MODELS FROM A BASIS REPRESENTATION PERSPECTIVE

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce a basis representation framework for analyzing how the representation geometry of large language models evolves during training by characterizing the representation geometry of internal architectures through Gram matrices. By integrating gradient flow analysis, we identify three stages of pretraining: a focus stage in which representation geometry condense onto a low-dimensional structure, a dilution stage in which previously insignificant directions are activated, and a stabilization stage in which the representation geometry converges while the parameters continue to evolve. We empirically validate these predicted dynamics across multiple LLMs and datasets. In particular, we consistently observe the theoretically predicted dynamics when training open-source LLMs on real-world datasets.

1 INTRODUCTION

Understanding how large language models (LLMs) learn during training is important for characterizing their internal learning dynamics and understanding the emergence of their capabilities. However, despite the success of modern LLMs Brown et al. (2020); Kaplan et al. (2020), their training dynamics remain difficult to characterize due to the high complexity of Transformer architectures, the coupling between different computational components, and the dependence on the training data.

Existing theoretical studies have analyzed the training dynamics of MLPs Rumelhart et al. (1986) and Transformers Vaswani et al. (2017), but typically focus on simplified architectures Yang et al. (2024a); Huang et al. (2025), idealized architectures Jacot et al. (2018); Malladi et al. (2023), or specific synthetic datasets Ildiz et al. (2024). Moreover, multi-stage training dynamics have also been theoretically studied: Guan et al. (2025) proposes a three-stage framework for LLM fine-tuning in a continual-learning setting; Varre et al. (2025) explains the stage-wise plateau-jump dynamics of Transformers in in-context n-gram learning; and Chen et al. (2026) identifies a token-frequency-driven, four-stage focus-dilution cycle in Transformer attention learning. However, a theoretical understanding of what representations are learned within these stages remains lacking.

In this work[1], we analyze the learning dynamics of LLMs through a basis representation and theoretically characterize and empirically validate a three-stage training dynamic in which the network first condenses to a low-dimensional structure, then dilutes to a higher-dimensional structure, and finally stabilizes. In Section 2, we introduce the basis representation framework for MLPs and LLMs, then develop a theoretical framework for these dynamics using basis representation and gradient flow. In Section 3, we empirically validate the three-stage dynamics on different LLMs (single-layer LLM, LLaMA2, and Qwen3) and different datasets (Markov and WikiText-2).

Our specific contributions can be summarized as follows:

- We introduce basis representation as a new perspective for measuring the representation geometry and analyzing the learning dynamics of LLMs.
- We provide a rigorous theoretical analysis of LLM gradient flow dynamics, revealing a three-stage training dynamic for pretraining.
- We empirically validate our theoretical findings, especially on real-world datasets and open-source LLMs.

[1]The related works are discussed in Appendix A

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2 METHODOLOGY

Given a function $\boldsymbol{f} : \mathbb{R}^{d_{\text{in}}} \to \mathbb{R}^{d_{\text{out}}}$, we represent the function by

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{j=1}^{N} \boldsymbol{\phi}_j(\boldsymbol{x}), \tag{1}$$

where $\boldsymbol{\phi}_j(\boldsymbol{x}) : \mathbb{R}^{d_{\text{in}}} \to \mathbb{R}^{d_{\text{out}}}$ are basis functions, which are not required to be either orthogonal or orthonormal. The Gram matrix $M = \{M_{ij}\} \in \mathbb{R}^{N \times N}$ of $\boldsymbol{f}$ is

$$M_{ij} = \int_{\Omega} \langle \boldsymbol{\phi}_i(\boldsymbol{x}), \boldsymbol{\phi}_j(\boldsymbol{x}) \rangle \mathrm{d}\boldsymbol{x}, \tag{2}$$

where $\Omega \subset \mathbb{R}^{d_{\text{in}}}$ is the input domain, and $\langle \cdot, \cdot \rangle$ denotes the inner product in $\mathbb{R}^{d_{\text{out}}}$. Notably, since $\boldsymbol{\phi}$ is not a normalized basis function, the diagonal entries of the Gram matrix may not equal 1. Thus, in the following analysis, we will use the normalized Gram matrix $\tilde{M}$, defined as

$$\tilde{M}_{ij} = \frac{M_{ij}}{\sqrt{M_{ii} M_{jj}}}. \tag{3}$$

Proposition 2.1. *Suppose $\tilde{\boldsymbol{f}}(\boldsymbol{x}) = \sum_{j=1}^{N} \alpha_j \tilde{\boldsymbol{\phi}}_j(\boldsymbol{x})$ is a function with the corresponding Gram matrix $M_{\tilde{f}}$ defined by Equation* (2)*, where $\tilde{\boldsymbol{\phi}}_j$ is the normalized basis function of $\boldsymbol{\phi}_j$ and α_j is the corresponding scaling factor. Then, if $M \succ 0$, the normalized Gram matrix of $\boldsymbol{f}$ is equal to the Gram matrix of $\tilde{\boldsymbol{f}}$, i.e., $\tilde{M} = M_{\tilde{f}}$.*

Proposition 2.1 reveals that the normalized Gram matrix is invariant to the scaling of the basis functions. Therefore, we can use the normalized Gram matrix to measure the representation space of neural networks without representing $\boldsymbol{f}$ by normalized basis functions. In what follows, unless otherwise specified, M denotes the normalized Gram matrix.

2.1 BASIS REPRESENTATION OF NEURAL NETWORKS

For a single-layer MLP with input dimension d_{in} and output dimension d_{out}, let the input be $\boldsymbol{x} \in \mathbb{R}^{d_{\text{in}}}$. The output is $\boldsymbol{u}(\boldsymbol{x}) = W\boldsymbol{x} + \boldsymbol{b}$. Suppose $W = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{d_{\text{in}}}] \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, where each column vector $\boldsymbol{w}_i \in \mathbb{R}^{d_{\text{out}}}$. Then we can define $\boldsymbol{\phi}_{d_{\text{in}}+1}(\boldsymbol{x}) = \boldsymbol{b}$ and $\boldsymbol{\phi}_i(\boldsymbol{x}) = x_i \boldsymbol{w}_i$ for $i \in [d_{\text{in}}]$, where $[d_{\text{in}}]$ denotes the set $\{1, 2, \cdots, d_{\text{in}}\}$. Then $\boldsymbol{u}(\boldsymbol{x}) = \sum_{j=1}^{d_{\text{in}}+1} \boldsymbol{\phi}_j(\boldsymbol{x})$.

If $X = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{d_{\text{in}}}] \in \mathbb{R}^{L_s \times d_{\text{in}}}$ is an input matrix rather than a vector, with $\boldsymbol{x}_i \in \mathbb{R}^{L_s}$ denoting its i-th column vector, then we can define $\Phi_{d_{\text{in}}+1}(X; \theta) = \mathbf{1}_{L_s} \boldsymbol{b}^\top$, where $\mathbf{1}_{L_s}$ is an all-one column vector, and $\Phi_i(X; \theta) = \boldsymbol{x}_i \boldsymbol{w}_i^\top$ for $i \in [d_{\text{in}}]$, yielding $\mathcal{U}(X; \theta) = \sum_{j=1}^{d_{\text{in}}+1} \Phi_j(X; \theta) \in \mathbb{R}^{L_s \times d_{\text{out}}}$.

For an L-layer MLP with input $\boldsymbol{x} \in \mathbb{R}^{d_{\text{in}}}$, the input to the last linear layer is the output of the preceding $L - 1$ layers. Therefore, the same representation applies by replacing $\boldsymbol{x}$ with an intermediate feature map $\boldsymbol{h}^{(L-1)}(\boldsymbol{x})$ that represents the computation of the first $L - 1$ layers. That is $\boldsymbol{u}(\boldsymbol{x}) = \sum_{j=1}^{d_{\text{in}}^L+1} \boldsymbol{\phi}_j(\boldsymbol{h}^{(L-1)}(\boldsymbol{x}))$, where d_{in}^L is the input dimension of the last linear layer.

For a standard transformer layer, the output $\mathcal{U}(X; \theta) = X + \text{MHSA}(X) + \text{MLP}(X + \text{MHSA}(X))$. Here MHSA is the multi-head self-attention module, and we ignore the layer norm layer since it can be regarded as a function on X. Suppose $A(X) \in \mathbb{R}^{L_s \times d_{\text{attn}}}$ denotes the concatenated multi-head attention matrix before the output projection. Then the MHSA output $X_1 := \text{MHSA}(X)$ can be written as

$$X_1 = A(X) W_O^\top + \mathbf{1}_{L_s} \boldsymbol{b}_O^\top, \tag{4}$$

where $W_O = [\boldsymbol{w}_{O,1}, \ldots, \boldsymbol{w}_{O,d_{\text{attn}}}] \in \mathbb{R}^{d_{\text{out}} \times d_{\text{attn}}}$ and $\boldsymbol{b}_O \in \mathbb{R}^{d_{\text{out}}}$. Let $\boldsymbol{a}_i(X)$ be the i-th column of $A(X)$. Similar to the MLP case, we represent

$$X_1 = \mathbf{1}_{L_s} \boldsymbol{b}_O^\top + \sum_{i=1}^{d_{\text{attn}}} \boldsymbol{a}_i(X) \boldsymbol{w}_{O,i}^\top. \tag{5}$$

The output of the attention residual branch is therefore $\gamma(X) = X + X_1$. This representation is then fed into the MLP block. Let $H(\gamma(X))$ denote the input to the last linear layer of the MLP, where H represents the computation of the preceding MLP layers and activation functions. If the last MLP projection is parameterized by $W_L = [\boldsymbol{w}_{L,1}, \ldots, \boldsymbol{w}_{L,d_h}] \in \mathbb{R}^{d_{\text{out}} \times d_h}$ and $\boldsymbol{b}_L \in \mathbb{R}^{d_{\text{out}}}$, then

$$\text{MLP}(\gamma(X)) = H(\gamma(X))W_L^\top + \mathbf{1}_{L_s}\boldsymbol{b}_L^\top. \tag{6}$$

Let $\boldsymbol{h}_i(\gamma(X))$ be the i-th column of $H(\gamma(X))$. The full attention-layer output can then be expressed as

$$\mathcal{U}(X;\theta) = \Phi_0^{\text{res}}(X) + \sum_{j=1}^{d_{\text{attn}}+1} \Phi_j^{\text{attn}}(X;\theta) + \sum_{j=1}^{d_h+1} \Phi_j^{\text{mlp}}(X;\theta), \tag{7}$$

where $\Phi_0^{\text{res}}(X) = X$, $\Phi_{d_{\text{attn}}+1}^{\text{attn}}(X;\theta) = \mathbf{1}_{L_s}\boldsymbol{b}_O^\top$, $\Phi_i^{\text{attn}}(X;\theta) = \boldsymbol{a}_i(X)\boldsymbol{w}_{O,i}^\top$, $\Phi_{d_h+1}^{\text{mlp}}(X;\theta) = \mathbf{1}_{L_s}\boldsymbol{b}_L^\top$, and $\Phi_i^{\text{mlp}}(X;\theta) = \boldsymbol{h}_i(\gamma(X))\boldsymbol{w}_{L,i}^\top$. Thus, the Gram matrix M is a $(3 + d_{\text{attn}} + d_h) \times (3 + d_{\text{attn}} + d_h)$ matrix.

Using the same approach as for L-layer MLPs, one can calculate the Gram matrix for LLMs. Notably, here the language model head (LM head) and the embedding layer are ignored in the Gram matrix computation, as they are not part of the representation space of the model: the LM head is a linear layer that maps the hidden states to the vocabulary space, and the embedding layer is a dictionary-like layer that maps discrete tokens to continuous vectors.

Notably, the basis functions are induced and uniquely determined by the network architecture and the parameters since our goal is to track the geometry evolution of the representation space during training rather than finding optimal or intrinsic basis functions.

2.1.1 Metrics for Gram Matrix

Here we introduce several metrics to measure the (normalized) Gram matrix as well as the representation space of neural networks. Suppose the Gram matrix M has eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$ and $p_i = \lambda_i / \left(\sum_{j=1}^N \lambda_j\right)$.

Effective rank. The effective rank Roy & Vetterli (2007) is defined as

$$r_{\text{eff}} = \exp\left(-\sum_{i=1}^{N} p_i \log p_i\right). \tag{8}$$

The effective rank $r_{\text{eff}} \in [1, N]$ measures the number of effective dimensions in the representation space. A larger effective rank indicates that the representation space is more complex, while a smaller effective rank indicates that the representation space is simpler. Furthermore, $\log r_{\text{eff}}$ is the von Neumann entropy Boes et al. (2019), which measures the uniformity of the eigenvalue distribution.

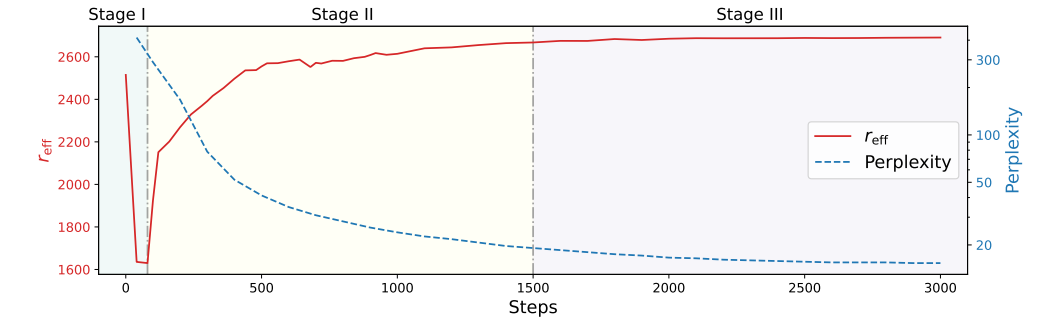
Maximum eigenvalue and minimum eigenvalue. The maximum eigenvalue λ_1 and the minimum eigenvalue λ_N of the Gram matrix M can be used to measure the representation space of neural networks. The eigenvalue λ_1 reflects the dominant scale of the representation along the principal direction, while λ_N measures the weakest direction in the space.

Orthogonality. The degree of orthogonality is defined as

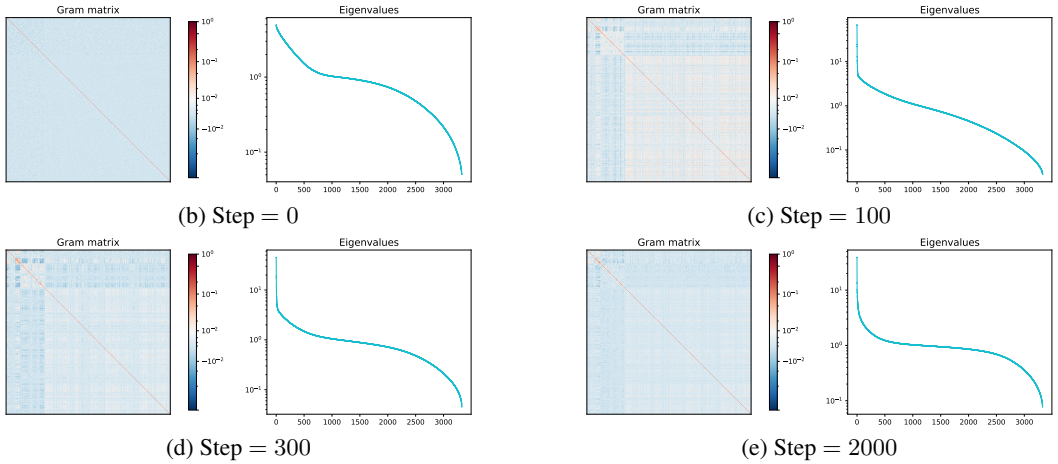
$$\rho = \frac{\sum_{i \neq j} |M_{ij}|}{N(N-1)}. \tag{9}$$

$\rho \in [0, 1]$ measures the degree of orthogonality of the basis functions. A smaller ρ indicates that the basis functions are more orthogonal, while a larger ρ indicates that the basis functions are more correlated.

Figure 1a shows the evolution of the effective rank r_{eff} during training. The evolution of the other metrics is discussed in Section 3. In our experiments, all aforementioned metrics exhibit three-stage behavior, which is consistent with the theoretical analysis in the following sections. Additionally, Figure 1 demonstrates the evolution of the Gram matrix and eigenvalue distribution over different training steps to provide a detailed view of the representation space dynamics.



(a) Perplexity and effective rank vs. training step



(b) Step $= 0$ (c) Step $= 100$

(d) Step $= 300$ (e) Step $= 2000$

Figure 1: Training dynamics of LLaMA2 on WikiText-2 from the perspective of basis representation. a shows test perplexity and effective rank r_{eff} as functions of training step. r_{eff} has three stages: it decreases rapidly in the focus stage, increases rapidly at the beginning of the dilution stage and then more slowly, and stabilizes in the stabilization stage. In contrast, perplexity decreases throughout training, including after r_{eff} has stabilized. b, c, d, and e show the Gram matrices and their eigenvalue distributions at steps 0, 100, 300, and 2000, respectively. At initialization, the off-diagonal Gram-matrix entries are close to zero and the eigenvalue distribution is more uniform. From step 100 to step 2000, the large eigenvalues decrease rapidly, the middle eigenvalues decrease progressively more slowly, and the small eigenvalues decrease progressively more rapidly.

2.2 Theoretical Analysis

In this section, we provide a theoretical analysis from the perspective of basis representation for LLMs. All proofs of the theoretical results are provided in Appendix E.

Setup. Let $\{\Phi_i\}_{i=1}^N \subset \mathbb{R}^{L_s \times d_{\text{out}}}$ denote normalized basis functions, and $\{X_k\}_{k=1}^{\mathcal{D}}$ denote the dataset. Then the output of LLMs can be expressed as $\mathcal{U}(X) = \sum_{i=1}^N \alpha_i \Phi_i(X) : \mathbb{R}^{L_s \times d_{\text{in}}} \to \mathbb{R}^{L_s \times d_{\text{out}}}$. L_s denotes the sequence length, d_{in} denotes the input dimension, d_{out} denotes the output dimension, and $\mathcal{D}$ denotes the number of samples in the dataset. Since the input is a discrete set, instead of Equation (2), the Gram matrix M is computed by

$$M_{ij} = \frac{1}{\mathcal{D}} \sum_{k=1}^{\mathcal{D}} \langle \Phi_i(X_k), \Phi_j(X_k) \rangle_F . \tag{10}$$

Proposition 2.2. *For every single input X_k, the basis functions $\{\Phi_i\}_{i=1}^N$ are linearly dependent if $N > L_s d_{\text{out}}$. For the entire dataset $\{X_k\}_{k=1}^{\mathcal{D}}$, the basis functions $\{\Phi_i\}_{i=1}^N$ are linearly dependent if $N > \mathcal{D} L_s d_{\text{out}}$.*

Proposition 2.2 indicates that the necessary condition for the linear independence of the basis functions is $N \leq \mathcal{D} L_s d_{\text{out}}$. Otherwise, the Gram matrix M is singular. One can also remove some

basis functions by some rank-revealing decompositions Kolda & Bader (2009). In practice, since the number of basis functions N is much smaller than $\mathcal{D}L_s d_{\text{out}}$, the necessary condition is satisfied.

Proposition 2.3. *For any LLMs with parameters randomly initialized from a non-atomic distribution (e.g. normal distribution), if $N \leq \mathcal{D}L_s d_{\text{out}}$, then for some constant $T > 0$, $\det M(t) > 0$ for all $t \in [0, T]$ almost surely, i.e., $\{\Phi_i(t)\}_{i=1}^N$ are linearly independent for all $t \in [0, T]$ almost surely.*

Proposition 2.3 reveals that the basis functions are linearly independent from the beginning of training up to some time $T > 0$. Therefore, the Gram matrix M is positive definite and invertible during $[0, T]$. In our experiments, Figures 2c, 2g and 2k show that the Gram matrix is positive definite for the entire training process. However, it is theoretically difficult to prove that the Gram matrix remains positive definite for $T \to \infty$; we discuss the value of T in Appendix E.3.

The following analysis is based on a single-layer LLM, excluding the input embedding layer and the LM head. That is, the input $X \in \mathbb{R}^{L_s \times d_{\text{in}}}$ is the output of the embedding layer, and the output of the LLM before the LM head is

$$\mathcal{U}(X) = X + \text{Softmax}(QK^\top)VW_O + \text{SiLU}(ZW_{\text{up}})W_{\text{down}} \tag{11}$$

where $Q = XW_Q$, $K = XW_K$, and $V = XW_V$ are the query, key, and value matrices, respectively. Without loss of generality, set the number of attention heads $d_{\text{head}} = 1$. The single-layer LLM is initialized with the truncated Gaussian initialization with standard deviation $\sigma = 0.02$, which is widely used in practical LLMs Radford et al. (2019); Wang et al. (2024a). For notational convenience, let W_{lm} denote the weight matrix of the LM head, $Z = X + \text{Softmax}(QK^\top)VW_O$, and $Z_{\text{up}} = ZW_{\text{up}}$.

Thus, for the basis representation of the single-layer LLM, we have

$$\mathcal{U}(X) = \alpha_0^{\text{res}}\Phi_0^{\text{res}}(X) + \sum_{j=1}^{d_{\text{attn}}} \alpha_j^{\text{attn}}\Phi_j^{\text{attn}}(X) + \sum_{j=1}^{d_h} \alpha_j^{\text{mlp}}\Phi_j^{\text{mlp}}(X), \tag{12}$$

Here, the basis functions are normalized, and α is the normalization scaling factor. Let $A = \text{Softmax}(QK^\top)V \in \mathbb{R}^{L_s \times d_{\text{attn}}}$. Denote the i-th column of $A(X)$ by $\boldsymbol{a}_i(X)$ and the transpose of the i-th row of W_O by $\boldsymbol{w}_{O,i}^\top$. Let $G = \text{SiLU}(Z_{\text{up}}) \in \mathbb{R}^{L_s \times d_h}$. We denote the i-th column of $G(X)$ by $\boldsymbol{g}_i(X)$ and the transpose of the i-th row of W_{down} by $\boldsymbol{w}_{\text{down},i}^\top$. Then,

$$\begin{aligned} \Phi_0^{\text{res}}(X) &= X/\alpha_0^{\text{res}}, \\ \Phi_i^{\text{attn}}(X) &= \boldsymbol{a}_i \boldsymbol{w}_{O,i}^\top/\alpha_i^{\text{attn}}, \quad i \in [d_{\text{attn}}], \\ \Phi_i^{\text{mlp}}(X) &= \boldsymbol{g}_i \boldsymbol{w}_{\text{down},i}^\top/\alpha_i^{\text{mlp}}, \quad i \in [d_h]. \end{aligned} \tag{13}$$

Then, the number of basis functions is $N = 1 + d_{\text{attn}} + d_h$. Recall $\mathcal{U}(X) = \sum_{i=1}^N \alpha_i \Phi_i(X)$ from setup, thus $\alpha_1 = \alpha_0^{\text{res}}$ and $\Phi_1 = \Phi_0^{\text{res}}$, $\alpha_{i+1} = \alpha_i^{\text{attn}}$ and $\Phi_{i+1} = \Phi_i^{\text{attn}}$ for $i \in [d_{\text{attn}}]$, and $\alpha_{d_{\text{attn}}+i+1} = \alpha_i^{\text{mlp}}$ and $\Phi_{d_{\text{attn}}+i+1} = \Phi_i^{\text{mlp}}$ for $i \in [d_h]$.

Assumption 2.1. *We assume $1 + d_{\text{attn}} + d_h \leq \mathcal{D}L_s d_{\text{out}}$. Based on Proposition 2.3, this guarantees the linear independence of the basis functions, as well as $M \succ 0$ and $\alpha_i > 0$ for all $i \in [N]$.*

Suppose $[0, T]$ is the training interval under consideration, and assume that Assumption 2.1 holds so $\lambda_{\min}(M(t)) > 0$ for all $t \in [0, T]$ in our experiments. Let $\theta(t)$ follow gradient flow for a differentiable loss $\mathcal{L}(\theta)$ [2], i.e., $\dot{\theta}(t) = -\nabla_\theta \mathcal{L}(\theta(t))$. Then, the Gram matrix at time t is defined as

$$M_{ij}(t) = \frac{1}{\mathcal{D}} \sum_{k=1}^{\mathcal{D}} \langle \Phi_i(X_k; \theta(t)), \Phi_j(X_k; \theta(t)) \rangle_F . \tag{14}$$

Since $\theta(t)$ is continuously differentiable and each $\Phi_i(X_k; \theta)$ is continuously differentiable with respect to θ, it follows that $\Phi_i(X_k; \theta(t))$ and hence $M(t)$ are continuously differentiable in t. Let $t_0 = 0$ and θ_0 denote the initial time and initial parameter. Since θ_0 is initialized from a truncated Gaussian distribution, we have $\theta_0 = \mathcal{O}(\sigma)$. Let t_1^*, θ_1^* and t_2^*, θ_2^* denote the critical times and

[2] $\mathcal{L}(\theta)$ is the empirical cross-entropy loss

parameters of the first and second stages, respectively. The training dynamics then proceeds through three stages: the focus stage on $[t_0, t_1^*]$, the dilution stage on $[t_1^*, t_2^*]$, and the stabilization stage on $[t_2^*, T]$.

Then we can theoretically analyze the three-stage training dynamics (namely, focus, dilution, and stabilization) from the perspective of the basis representation framework and gradient flow. Starting from a truncated Gaussian initialization, the gradient flow trajectory first rapidly condenses on a low-dimensional structure spanned by a small subset of basis functions, corresponding to the focus stage. It then gradually escapes this low-dimensional structure and activates previously insignificant directions, corresponding to the dilution stage. Finally, once the relevant directions are sufficiently captured, the trajectory approaches a stable geometry with slight oscillations and eventually converges, corresponding to the stabilization stage.

Accordingly, in the following analysis, the effective rank r_{eff} serves as a suitable and representative metric for distinguishing these three stages of training; that is, r_{eff} first decreases during the focus stage, then increases during the dilution stage, and finally stabilizes during the stabilization stage.

2.2.1 STAGE I: FOCUS STAGE $[0, t_1^*]$

We first characterize $r_{\text{eff}}(M(0))$, which serves as the reference point for the subsequent analysis.

Proposition 2.4. *Suppose the model is initialized using a truncated Gaussian distribution with standard deviation σ, and write the Gram matrix as $M(0) = I_N + \mathcal{E}_0$, where $I_N \in \mathbb{R}^{N\times N}$ is the identity matrix. If $\|\mathcal{E}_0\|_2 \le \varepsilon_0 < 1$, then the effective rank at $t = 0$ satisfies*

$$r_{\text{eff}}(M(0)) = N \exp\left(-\mathcal{O}(\varepsilon_0^2)\right) = N\left(1 - \mathcal{O}(\varepsilon_0^2)\right). \tag{15}$$

The magnitude of ε_0 is primarily governed by σ, N, $\mathcal{D}$, L_s, and d_{out}. Under $\sigma = 0.02$, $N \ll \mathcal{D}L_s d_{\text{out}}$ provides a favorable high-dimensional regime and the sufficient condition for $\varepsilon_0 \ll 1$. Based on Proposition 2.4, we make the Assumption 2.2 to exclude degenerate cases.

Assumption 2.2. *We assume $\varepsilon_0 \ll 1$, so that the effective rank $r_{\text{eff}}(M(0))$ is close to N.*

In Figures 2d, 2h and 2l, $\rho(M(0))$ is close to 0, which also empirically supports the assumption of $\varepsilon_0 \ll 1$. With the characterization of $r_{\text{eff}}(M(0))$, we next analyze the dynamics of $r_{\text{eff}}(M(t))$ during the focus stage. First, we define the critical time t_1^* by Definition 2.1.

Definition 2.1. Let $t_1^* > 0$ be such that, for all $t \in [0, t_1^*]$, every weight matrix in Equation (11) and W_{lm} remains $\mathcal{O}(\sigma)$, where σ is the standard deviation of the truncated Gaussian initialization.

Theorem 2.1. *If $\sigma \ll 1$ and the fixed embedding outputs satisfy $\max_{i\in[\mathcal{D}]} \|X_i\|_F = \mathcal{O}(1)$, then the following estimates hold uniformly for $t \in [0, t_1^*]$:*

$$\begin{aligned} &\dot{W}_Q(t) = \mathcal{O}(\sigma^4), \quad \dot{W}_K(t) = \mathcal{O}(\sigma^4), \qquad \dot{W}_V(t) = \mathcal{O}(\sigma^2), \\ &\dot{W}_O(t) = \mathcal{O}(\sigma^2), \quad \dot{W}_{\text{up}}(t) = \mathcal{O}(\sigma^2), \quad \dot{W}_{\text{down}}(t) = \mathcal{O}(\sigma^2). \end{aligned} \tag{16}$$

Theorem 2.1 characterizes the weight scales during the focus stage and reveals that W_Q and W_K are quasi-static to leading order during $[0, t_1^*]$. The quasi-static behavior of W_Q and W_K during the initial $[0, t_1^*]$ is empirically validated in Li et al. (2025); Chen et al. (2026) for realistic LLMs. Since no causal mask is applied and W_Q and W_K are sufficiently small, the row-wise softmax is approximately uniform; that is, $\text{Softmax}(QK^\top) \approx \frac{1}{L_s}\mathbf{1}\mathbf{1}^\top$. Then,

$$\text{Softmax}(QK^\top)VW_O \approx \frac{1}{L_s}\mathbf{1}\mathbf{1}^\top XW_V W_O. \tag{17}$$

Thus, during this stage, the attention behaves approximately as a linear MLP layer without activation functions. Consequently, the single-layer LLM can be viewed as two MLP layers operating in parallel, together with the residual connection.

Assumption 2.3. *For $r \in \{V, O, \text{up}, \text{down}\}$, suppose $W_r(t) \in \mathbb{R}^{d_r^{\text{in}} \times d_r^{\text{out}}}$, and let $N_r(t)$ be the smallest integer such that*

$$W_r(t) = \sum_{j=1}^{N_r(t)} \lambda_{r,j}(t)\boldsymbol{u}_{r,j}(t)\boldsymbol{v}_{r,j}(t)^\top + R_r(t), \tag{18}$$

where $R_r(t)$ is the relatively small residual matrix satisfying $\|R_r(t)\|_F \leq \eta\|W_r(t)\|_F$, for some $0 < \eta \ll 1$. Each $N_r(t)$ satisfies

$$N_r(t_1^*) \ll \min\{d_r^{\text{in}}, d_r^{\text{out}}\}. \tag{19}$$

Given the small parameter values of truncated Gaussian initialization, the implicit low-rank bias of gradient-based optimization observed in deep matrix factorization Arora et al. (2019); Razin & Cohen (2020) provides theoretical motivation for Assumption 2.3.

Theorem 2.2. *Suppose the model satisfies the conditions of Theorem 2.1. Under Assumption 2.3, during the focus stage $t \in [0, t_1^*]$, the effective rank satisfies*

$$r_{\text{eff}}(M(t)) \lesssim 1 + N_V(t)N_O(t) + N_G(t)N_{\text{down}}(t), \qquad t \in [0, t_1^*]. \tag{20}$$

Here $N_G(t) = N_{\text{up}}(t)(N_{\text{up}}(t) + 3)/2$.

Together with Assumption 2.2 and Assumption 2.3, Theorem 2.2 establishes a transition from an initially near full effective rank representation to a low effective rank representation during the focus stage.

2.2.2 Stage II: Dilution stage $[t_1^*, t_2^*]$

We first define the critical point between the dilution stage and the stabilization stage as t_2^* by

Definition 2.2. Let $t_2^* \in (t_1^*, T)$ be such that, for a given $\varepsilon_{t_2^*} > 0$, for all $t \in [t_2^*, T]$, we have $\|\nabla_\theta \mathcal{L}(\theta(t))\|_2 \leq \varepsilon_{t_2^*}$.

Thus, t_2^* is an artificial critical point depending on the chosen threshold $\varepsilon_{t_2^*}$, distinguishing the dilution stage from the stabilization stage. However, since a small $\varepsilon_{t_2^*}$ may lead to nonexistence of t_2^* in $[t_1^*, T]$, we make Assumption 2.4 to guarantee the existence of the stabilization stage, assuming that the training process converges before the terminal time T.

Assumption 2.4. *For $\varepsilon_{t_2^*} \ll 1$, assume there exists $t_2^* \in (t_1^*, T)$ satisfying Definition 2.2, i.e., $\|\nabla_\theta \mathcal{L}(\theta(t))\|_2 \leq \varepsilon_{t_2^*} \ll 1$ for all $t \in [t_2^*, T]$.*

Then, we analyze the evolution of $r_{\text{eff}}(M(t))$ during $[t_1^*, t_2^*]$. Let $\boldsymbol{p}(t) = (p_1(t), \ldots, p_N(t))$ be the eigenvalue distribution of $M(t)$. Recall that $p_i(t) = \lambda_i(t)/\sum_{i=1}^N \lambda_i(t) = \lambda_i(t)/N$. For simplicity, $r_{\text{eff}}(\boldsymbol{p}(t)) = r_{\text{eff}}(M(t))$. Let $\boldsymbol{p}^\star$ denote the normalized eigenvalue distribution of the target representation associated with the optimization trajectory on the dataset. Suppose $\mathcal{B} = \{B \in \mathbb{R}^{N\times N}, B_{ij} \geq 0, \sum_{j=1}^N B_{ij} = 1, \sum_{i=1}^N B_{ij} = 1, \forall i, j \in [N]\}$ is the set of doubly stochastic matrices.

Theorem 2.3. *If $r_{\text{eff}}(\boldsymbol{p}^\star) > r_{\text{eff}}(\boldsymbol{p}(t_1^*))$ and there exist $\alpha \in (0, 1]$ and a matrix $B \in \mathcal{B}$ such that*

$$\boldsymbol{p}(t_2^*) = (1 - \alpha)\boldsymbol{p}(t_1^*) + \alpha B\boldsymbol{p}^\star, \tag{21}$$

then we have

$$r_{\text{eff}}(M(t_2^*)) > r_{\text{eff}}(M(t_1^*)). \tag{22}$$

For real-world datasets, the complex structure is generally not adequately represented by the low-dimensional focus-stage subspace, making $r_{\text{eff}}(\boldsymbol{p}^\star) > r_{\text{eff}}(\boldsymbol{p}(t_1^*))$ natural. In practical gradient-based training, the stable optimization trajectory satisfies the condition of Equation (21)[3].

Monotonicity of $r_{\text{eff}}(M(t))$. The theoretical analysis above characterizes the effective rank at the critical points t_0, t_1^*, and t_2^*, but does not establish its monotonicity throughout intervals. In practical optimization, finite learning rates and curved parameter trajectories may not move straightly toward the target representation, so $r_{\text{eff}}(M(t))$ may exhibit transient deviations from the stage-wise trend. Moreover, since effective rank is determined by the entropy of the normalized eigenvalue distribution, competing spectral changes can offset each other. Herein, a monotonicity guarantee would therefore require stronger assumptions and may conflict with the realistic dynamics.

[3]The further discussion of the two conditions refers to Appendix E.7

2.2.3 STAGE III: STABILIZATION STAGE $[t_2^*, T]$

For the stabilization stage, Proposition 2.5 reveals the relationship between $\varepsilon_{t_2^*}$ and r_{eff}, indicating the stabilization of $r_{\text{eff}}(M(t))$ in $[t_2^*, T]$ under Assumption 2.4.

Proposition 2.5. *If $M(t)$ satisfies Assumption 2.1 and Assumption 2.4 holds, then there exists a constant $C_{[t_2^*,T]} > 0$ such that, for every $t \in [t_2^*, T]$,*

$$\left| \frac{\mathrm{d}}{\mathrm{d}t} r_{\text{eff}}(M(t)) \right| \leq C_{[t_2^*,T]} \varepsilon_{t_2^*}, \tag{23}$$

where $C_{[t_2^*,T]}$ depends only on N and uniform upper bounds on $\|\log M(t)\|_2$ and $\max_{i,j\in[N]} \|\nabla_\theta M_{ij}(t)\|_2$ for $t \in [t_2^*, T]$.

More interestingly, as shown in Figure 1, before $\mathcal{L}(\theta)$ fully converges, r_{eff} may have already stabilized with slight oscillations around a stable value, indicating that the model has reached a point where the effective rank stabilizes, while optimization may continue to tune the parameters. We next characterize how the underlying components can continue to evolve during the stabilization stage.

Proposition 2.6. *For $t \in [0, T]$, if $M(t)$ satisfies Assumption 2.1, then*

$$\frac{\mathrm{d}}{\mathrm{d}t} \log r_{\text{eff}}(M(t)) = -\frac{2}{N} \sum_{1\leq i<j\leq N} (\log M(t))_{ij} \dot{M}_{ij}(t), \tag{24}$$

thus,

$$\sum_{1\leq i<j\leq N} (\log M(t))_{ij} \dot{M}_{ij}(t) = 0 \quad \Longleftrightarrow \quad \frac{\mathrm{d}}{\mathrm{d}t} r_{\text{eff}}(M(t)) = 0. \tag{25}$$

Proposition 2.6 gives a necessary and sufficient condition for the stability of r_{eff}. However, unchanged r_{eff} does not fully determine the behavior of the underlying components. In the following, we change to focusing on the entries of the Gram matrix.

Proposition 2.7. *For every $i, j \in [N]$ and $t \in [0, T]$,*

$$\dot{M}_{ij}(t) = -\left\langle \nabla_\theta M_{ij}(t), \nabla_\theta \mathcal{L}(\theta(t)) \right\rangle, \tag{26}$$

thus,

$$\left\langle \nabla_\theta M_{ij}(t), \nabla_\theta \mathcal{L}(\theta(t)) \right\rangle = 0 \quad \Longleftrightarrow \quad \dot{M}_{ij}(t) = 0. \tag{27}$$

Proposition 2.7 reveals that Gram stabilization only requires the projection of the loss-gradient flow onto the Gram matrix to vanish. To further analyze the behavior of the underlying components during the stabilization stage with M_{ij}, we introduce Assumption 2.5 for the stability of $\{X_k\}_{k=1}^{\mathcal{D}}$ during the stabilization stage.

Assumption 2.5. *Assume that the parameters of the embedding layer change negligibly during the stabilization stage $[t_2^*, T]$. That is, $\{X_k\}_{k=1}^{\mathcal{D}}$, Φ_1, and α_1 remain nearly unchanged.*

Theorem 2.4. *For all $i \in [N] \setminus \{1\}$, suppose $\Phi_i(X; t) = \boldsymbol{b}_i(X; t)\boldsymbol{v}_i(t)^\top / \alpha_i(t)$, $v_{ij}(t) = \boldsymbol{v}_i(t)^\top \boldsymbol{v}_j(t)$, and $b_{ij}(t) = \frac{1}{\mathcal{D}} \sum_{k=1}^{\mathcal{D}} \boldsymbol{b}_i(X_k; t)^\top \boldsymbol{b}_j(X_k; t)$. If Assumption 2.1 holds, then for the (normalized) Gram matrix $M(t)$, we have*

$$M_{ij}(t) = \frac{b_{ij}(t) v_{ij}(t)}{\sqrt{b_{ii}(t) b_{jj}(t)} \sqrt{v_{ii}(t) v_{jj}(t)}}. \tag{28}$$

Furthermore, for all $i, j \in [N] \setminus \{1\}$ such that $M_{ij}(t) \neq 0$, we have

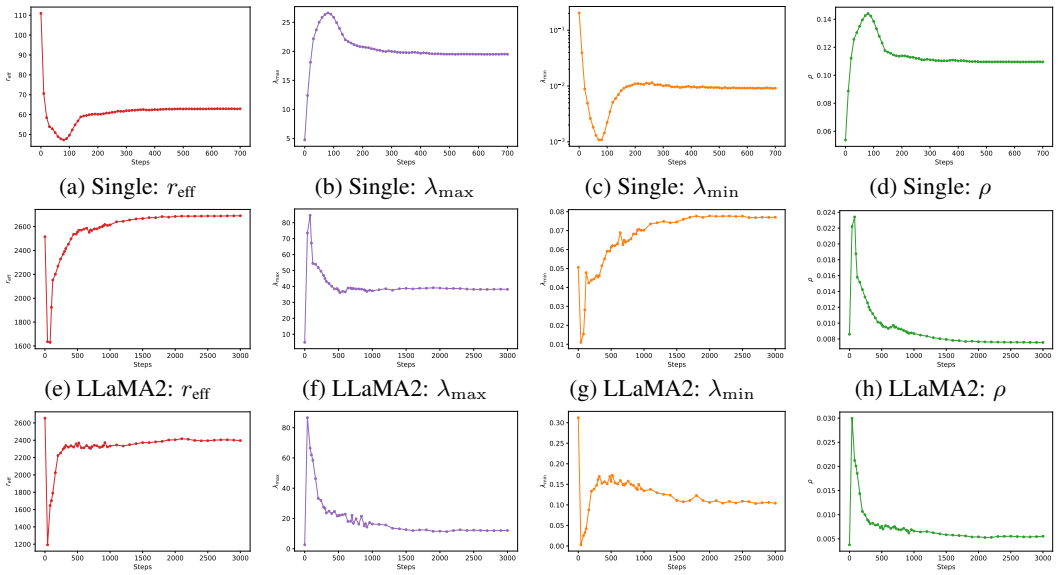
$$\frac{\dot{b}_{ij}(t)}{b_{ij}(t)} + \frac{\dot{v}_{ij}(t)}{v_{ij}(t)} = \frac{1}{2} \left(\frac{\dot{b}_{ii}(t)}{b_{ii}(t)} + \frac{\dot{b}_{jj}(t)}{b_{jj}(t)} + \frac{\dot{v}_{ii}(t)}{v_{ii}(t)} + \frac{\dot{v}_{jj}(t)}{v_{jj}(t)} \right) \iff \dot{M}_{ij}(t) = 0. \tag{29}$$

Together with Assumption 2.5, Theorem 2.4 shows that, when the input representations remain unchanged, Gram stabilization constrains only the normalized pairwise correlations of the activation and output-weight factors. We also provide extra empirical experiments for the stabilization stage in Appendix B to demonstrate that, for $\mathcal{U} = \sum_{j=1}^{N} \alpha_j \Phi_j$, the basis geometry $\{\Phi_j\}_{j=1}^{N}$ changes slightly during the stabilization stage, while coefficients α_j continue to evolve.

3 EXPERIMENTS

Here we present the experimental results for the single-layer LLM on artificial Markov datasets[4] and for the LLaMA2 Touvron et al. (2023) and Qwen3 Yang et al. (2025) models on WikiText-2 Merity et al. (2016). The model size of LLaMA2 is about 400 MB, and the model size of Qwen3 is about 220 MB. Other details of the experimental setup, including hyperparameters, model configurations, and training dynamics, are provided in Appendix D.

Figure 2 shows the dynamics of effective rank r_{eff}, degree of orthogonality ρ, and the maximum and minimum eigenvalues $\lambda_{\max}, \lambda_{\min}$. The effective rank first decreases, then increases, and eventually converges to a stable value. Combined with the dynamics of the other metrics, these experiments empirically validate the theoretically predicted three-stage training dynamics in Section 2.2. To validate the generalization, we provide additional experiments in Appendix C, including training on different token-frequency distributions and measurements of r_{eff} in intermediate layers.



(a) Single: r_{eff} (b) Single: $\lambda_{\max}$ (c) Single: $\lambda_{\min}$ (d) Single: ρ

(e) LLaMA2: r_{eff} (f) LLaMA2: $\lambda_{\max}$ (g) LLaMA2: $\lambda_{\min}$ (h) LLaMA2: ρ

(i) Qwen3: r_{eff} (j) Qwen3: $\lambda_{\max}$ (k) Qwen3: $\lambda_{\min}$ (l) Qwen3: ρ

Figure 2: Dynamics of effective rank r_{eff}, degree of orthogonality ρ, and maximum and minimum eigenvalues $\lambda_{\max}, \lambda_{\min}$ for the single-layer LLM on Markov datasets (a, d, b, c), LLaMA2 (e, h, f, g), and Qwen3 (i, l, j, k) on WikiText-2. The range of ρ is $[0, 1]$ for all models. The range of r_{eff} is $[1, 177]$ for the single-layer LLM, $[1, 3329]$ for LLaMA2, and $[1, 2817]$ for Qwen3.

4 CONCLUSION

This paper introduced basis representation as a framework for analyzing the representation geometry of LLMs during pretraining. Both our analysis and experiments support a three-stage dynamic: during the focus stage, the representation becomes dominated by a low-dimensional subset of basis directions; during the dilution stage, additional data-dependent directions become active; and during the stabilization stage, the representation varies slightly. More interestingly, the stabilization stage does not imply that optimization has stopped: the Gram geometry can stabilize while the underlying parameters and basis coefficients continue to evolve. Experiments on open-source LLMs trained on real-world data show that this geometric representation is consistent with our theoretical findings.

Limitations and Future Work. The theoretical framework does not consider architectural components in state-of-the-art LLMs, such as gated mechanisms Qiu et al. (2026). Extending the theoretical framework to more advanced architectures is an important direction for future work. Second, computational constraints limit our experiments to relatively small-scale models. Testing whether these dynamics persist at scale remains to be investigated.

[4]See Appendix C.2 for details of the Markov datasets.

AI USE STATEMENT

In this work, we used generative AI tools for [generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, assist in the writing of proofs, design or provide feedback on research methodology or experiments, assist with translation]. We have not used generative AI tools for [propose or refine hypotheses, provide critical ingredients for proving mathematical claims, interpret results, implement methods], and [clean and reformat dataset, support qualitative and thematic data analysis] are not applicable to this work. Additionally, we used generative AI tools for [edit a research paper to improve readability, identify relevant literature, format references]. We have reviewed all AI-assisted work. [All mathematical proofs assisted by generative AI tools were carefully checked and independently verified by the authors]. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The code and configuration files used to reproduce the experiments will be made publicly available upon publication. Our implementation is based on the public repository released with Zhao et al. (2024), which has also been widely used and adapted by others, including Li et al. (2024); Sun et al. (2026). We use the publicly available WikiText-2 dataset. Details of the experimental setup including hyperparameters, model configurations, and training dynamics are provided in Appendix D. The complete proofs of the theoretical results are provided in Appendix E.

REFERENCES

David Ifeoluwa Adelani, Jade Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, et al. Masakhaner: Named entity recognition for african languages. *Transactions of the Association for Computational Linguistics*, 9:1116–1131, 2021.

Kumar K. Agrawal, Arnab Kumar Mondal, Arna Ghosh, and Blake Richards. α-req: Assessing representation quality in self-supervised learning by measuring eigenspectrum decay. *Advances in Neural Information Processing Systems*, 35:17626–17638, 2022.

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. *Advances in neural information processing systems*, 32, 2019.

Paul Boes, Jens Eisert, Rodrigo Gallego, Markus P Müller, and Henrik Wilming. Von neumann entropy from unitarity. *Physical review letters*, 122(21):210402, 2019.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Zheng-An Chen, Pengxiao Lin, Zhi-Qin John Xu, and Tao Luo. Focus and dilution: The multi-stage learning process of attention. *arXiv preprint arXiv:2605.01199*, 2026.

Fanny Dufossé, Kamer Kaya, Ioannis Panagiotas, and Bora Uçar. Further notes on birkhoff–von neumann decomposition of doubly stochastic matrices. *Linear Algebra and its Applications*, 554: 68–78, 2018.

Changhao Guan, Chao Huang, Hongliang Li, You Li, Ning Cheng, Zihe Liu, Yufeng Chen, Jinan Xu, and Jian Liu. Multi-stage llm fine-tuning with a continual learning setting. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 5499–5513, 2025.

Godfrey H. Hardy, John E. Littlewood, and George Pólya. *Inequalities*. Cambridge University Press, 1934.

Ruiquan Huang, Yingbin Liang, and Jing Yang. How transformers learn regular language recognition: A theoretical study on training dynamics and implicit bias. *arXiv preprint arXiv:2505.00926*, 2025.

Muhammed Emrullah Ildiz, Yixiao Huang, Yingcong Li, Ankit Singh Rawat, and Samet Oymak. From self-attention to Markov models: Unveiling the dynamics of generative transformers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20955–20982. PMLR, 21–27 Jul 2024.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3): 455–500, 2009.

Bingrui Li, Wei Huang, Andi Han, Zhanpeng Zhou, Taiji Suzuki, Jun Zhu, and Jianfei Chen. On the optimization and generalization of two-layer transformers with sign gradient descent. In *International Conference on Learning Representations*, volume 2025, pp. 64406–64484, 2025.

Melody Li, Kumar Krishna Agrawal, Arna Ghosh, Komal Teru, Adam Santoro, Guillaume Lajoie, and Blake Richards. Tracing the representation geometry of language models from pretraining to post-training. *Advances in Neural Information Processing Systems*, 38:54691–54724, 2026.

Pengxiang Li, Lu Yin, and Shiwei Liu. Mix-ln: Unleashing the power of deeper layers by combining pre-ln and post-ln. *arXiv preprint arXiv:2412.13795*, 2024.

Andy Zeyi Liu, Elliot Paquette, and John Sous. Spectral lens: Activation and gradient spectra as diagnostics of llm optimization. *arXiv preprint arXiv:2605.05683*, 2026.

Xinliang Liu, Tong Mao, and Jinchao Xu. Condition numbers and eigenvalue spectra of shallow networks on spheres. *arXiv preprint arXiv:2511.02625*, 2025.

Ashok Makkuva, Marco Bondaschi, Adway Girish, Alliot Nagle, Martin Jaggi, Hyeji Kim, and Michael Gastpar. Attention with Markov: A curious case of single-layer transformers. In *International Conference on Learning Representations*, volume 2025, pp. 853–879, 2025.

Sadhika Malladi, Alexander Wettig, Dingli Yu, Danqi Chen, and Sanjeev Arora. A kernel-based view of language model fine-tuning. In *International Conference on Machine Learning*, pp. 23610–23641. PMLR, 2023.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, et al. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. *Advances in Neural Information Processing Systems*, 38:100092–100118, 2026.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Noam Razin and Nadav Cohen. Implicit regularization in deep learning may not be explainable by norms. *Advances in neural information processing systems*, 33:21174–21187, 2020.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*, pp. 606–610. IEEE, 2007.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

Jie Shen, Tao Tang, and Li-Lian Wang. *Spectral methods: algorithms, analysis and applications*. Springer Science & Business Media, 2011.

Khemraj Shukla, Juan Diego Toscano, Zhicheng Wang, Zongren Zou, and George Em Karniadakis. A comprehensive and fair comparison between mlp and kan representations for differential equations and operator networks. *Computer Methods in Applied Mechanics and Engineering*, 431: 117290, 2024.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.

Wenfang Sun, Xinyuan Song, Pengxiang Li, Lu Yin, Yefeng Zheng, and Shiwei Liu. The curse of depth in large language models. *Advances in Neural Information Processing Systems*, 38: 163104–163136, 2026.

Vimal Thilak, Chen Huang, Omid Saremi, Laurent Dinh, Hanlin Goh, Preetum Nakkiran, Joshua Susskind, and Etai Littwin. Lidar: Sensing linear probing performance in joint embedding ssl architectures. In *International Conference on Learning Representations*, volume 2024, pp. 56726–56765, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Aditya Varre, Gizem Yüce, and Nicolas Flammarion. Learning in-context n-grams with transformers: Sub-n-grams are near-stationary points. *arXiv preprint arXiv:2508.12837*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. Deepnet: Scaling transformers to 1,000 layers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(10):6761–6774, 2024a.

Yifan Wang, Hehu Xie, and Pengzhan Jin. Tensor neural network and its numerical integration. *Journal of Computational Mathematics*, 42(6):1714–1742, 2024b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Hongru Yang, Bhavya Kailkhura, Zhangyang Wang, and Yingbin Liang. Training dynamics of transformers to recognize word co-occurrence via gradient flow analysis. *Advances in Neural Information Processing Systems*, 37:46047–46117, 2024a.

Jiang Yang, Yuxiang Zhao, and Quanhui Zhu. Effective rank and the staircase phenomenon: New insights into neural network training dynamics. *arXiv preprint arXiv:2412.05144*, 2024b.

Tianchi Yu, Yiming Qi, Ivan Oseledets, and Shiyi Chen. Spectral informed neural networks. *Journal of Computational and Applied Mathematics*, pp. 117178, 2025.

Tianchi Yu, Jingwei Qiu, Jiang Yang, and Ivan Oseledets. Sinc kolmogorov-arnold network and its application for solving pdes with singularities. *Neural Networks*, pp. 109033, 2026.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. Galore: Memory-efficient llm training by gradient low-rank projection. *arXiv preprint arXiv:2403.03507*, 2024.

Yufa Zhou, Yixiao Wang, Xunjian Yin, Shuyan Zhou, and Anru Zhang. The geometry of reasoning: Flowing logics in representation space. In *International Conference on Learning Representations*, volume 2026, pp. 72711–72739, 2026.

A Related Works

Basis representation has been widely used in tensor decomposition Oseledets (2011) (tensor bases) and spectral analysis Shen et al. (2011) (spectral bases), providing a powerful tool for understanding the underlying structure of given data and functions. For the study of deep neural networks, Wang et al. (2024b) proposes to use tensor bases to construct tensor neural networks, Yu et al. (2025) proposes to use spectral bases to construct spectrally informed neural networks, and Sitzmann et al. (2020); Shukla et al. (2024); Yu et al. (2026) employ spectral bases to compose the activation function. Additionally, Yang et al. (2024b) proposes a novel metric ϵ-rank and reveals the staircase phenomenon in MLP training dynamics, while Liu et al. (2025) studies the spectral bias of shallow ReLU^k networks using the condition number and eigenvalue spectra.

For the study of the representations and geometric properties of LLMs, Thilak et al. (2024) proposes a novel metric to measure the effective intrinsic dimensionality of representations, Li et al. (2026) empirically analyzes the evolution of last-token hidden representations via effective rank and α-ReQ Agrawal et al. (2022) across pretraining and fine-tuning stages, Zhou et al. (2026) employs differential geometry to analyze manifold structures underlying LLM reasoning trajectories, and Liu et al. (2026) proposes a dual-SVD-based diagnostic framework to investigate hidden training dynamics of decoder-only LLMs beyond loss.

B Empirical evidence for stabilization stage

In this section, we use $M(t)$ to denote the unnormalized Gram matrix and $\tilde{M}(t)$ to denote the corresponding normalized Gram matrix.

For the stabilization stage, we empirically measure three metrics between $M(t)$ and $\tilde{M}(t)$:

$$E_1 = \left(\sum_{i=1}^{N} \left(M_{ii}(t) - \tilde{M}_{ii}(t) \right)^2 \right)^{1/2}, \tag{30}$$

$$E_2 = \left(\sum_{i \neq j} \left(M_{ij}(t) - \tilde{M}_{ij}(t) \right)^2 \right)^{1/2}, \tag{31}$$

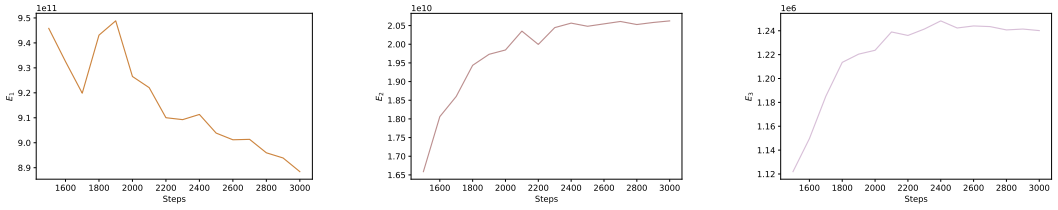
and

$$E_3 = \frac{1}{N(N-1)} \sum_{i \neq j} \sqrt{M_{ii}(t) M_{jj}(t)}. \tag{32}$$

Here, E_1 measures the difference in diagonal entries induced by normalization and reflects the deviation of the basis-function scales from unit norm; E_2 quantifies the corresponding difference in off-diagonal entries, capturing how normalization modifies pairwise correlations between distinct basis functions; E_3 computes the average geometric-mean scale over pairs of basis functions, representing the average scaling factor applied to the off-diagonal entries of $M(t)$. Since ρ represents the mean absolute cosine similarity between distinct basis functions, its stability in the stabilization stage is empirically supported by Figure 2.

During the stabilization stage, Figure 3 shows that E_1 decreases rapidly, whereas E_2 and E_3 approach plateaus with mild oscillations. Together with the experimental results showing that ρ remains small and stable, these results indicate that the basis functions preserve an approximately orthogonal and stable angular geometry. Therefore, for $\mathcal{U} = \sum_{j=1}^{N} \alpha_j \Phi_j$, the normalized basis geometry changes slightly during the stabilization stage, while the basis scaling coefficients α_j continue to evolve.

Taken together with the results in Figure 2, these observations show that the proposed three-stage dynamical process is consistently manifested across the examined layers: the focus, dilution, and stabilization behaviors remain qualitatively unchanged as depth increases. This cross-layer consistency provides empirical evidence that the mechanism is governed by the shared geometric structure of the representations rather than by an artifact of a particular layer or model size, and therefore suggests that the same analysis can remain informative for deeper and larger models.



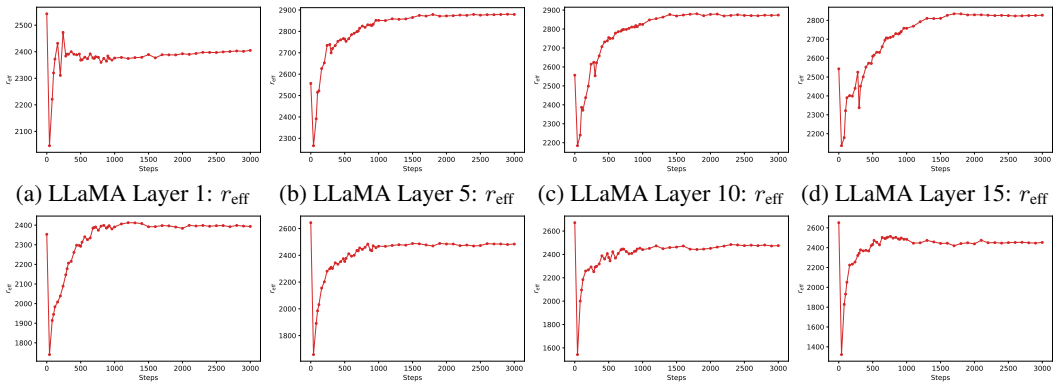
(a) E_1 (b) E_2 (c) E_3

Figure 3: Stabilization stage metrics for LLaMA on WikiText-2. Here, E_1, E_2, and E_3 are defined in Equations (30) to (32).

C Additional Experimental Results

C.1 Effective Rank of Intermediate Layers

We compute the effective rank of the chosen intermediate layers for LLaMA and Qwen models. Figure 4 shows layers 1, 5, 10, and 15 for LLaMA, and layers 1, 4, 8, and 12 for Qwen.



(a) LLaMA Layer 1: r_{eff} (b) LLaMA Layer 5: r_{eff} (c) LLaMA Layer 10: r_{eff} (d) LLaMA Layer 15: r_{eff}

(e) Qwen Layer 1: r_{eff} (f) Qwen Layer 4: r_{eff} (g) Qwen Layer 8: r_{eff} (h) Qwen Layer 12: r_{eff}

Figure 4: r_{eff} for intermediate layers of LLaMA and Qwen.

C.2 Experiments on Markov Dataset

The Markov dataset is widely used in theoretical analysis of LLMs Adelani et al. (2021); Makkuva et al. (2025).

Markov dataset. Let $\mathcal{V}$ be a finite token vocabulary with $|\mathcal{V}| \geq 2$, let $L_s \geq 1$, and let $\boldsymbol{\pi} = (\pi_v)_{v \in \mathcal{V}}$ be a probability distribution. For $u, v \in \mathcal{V}$ and $\beta \in (0, 1)$, define

$$P_{uv}(\boldsymbol{\pi}) = (1 - \beta)\mathbf{1}_{u=v} + \beta\pi_v. \tag{33}$$

For the input sequence $\boldsymbol{s} = (s_1, \ldots, s_{L_s}) \in \mathcal{V}^{L_s}$, let

$$p_{\boldsymbol{\pi}}(\boldsymbol{s}) = \pi_{s_1} \prod_{t=1}^{L_s - 1} P_{s_t, s_{t+1}}(\boldsymbol{\pi}). \tag{34}$$

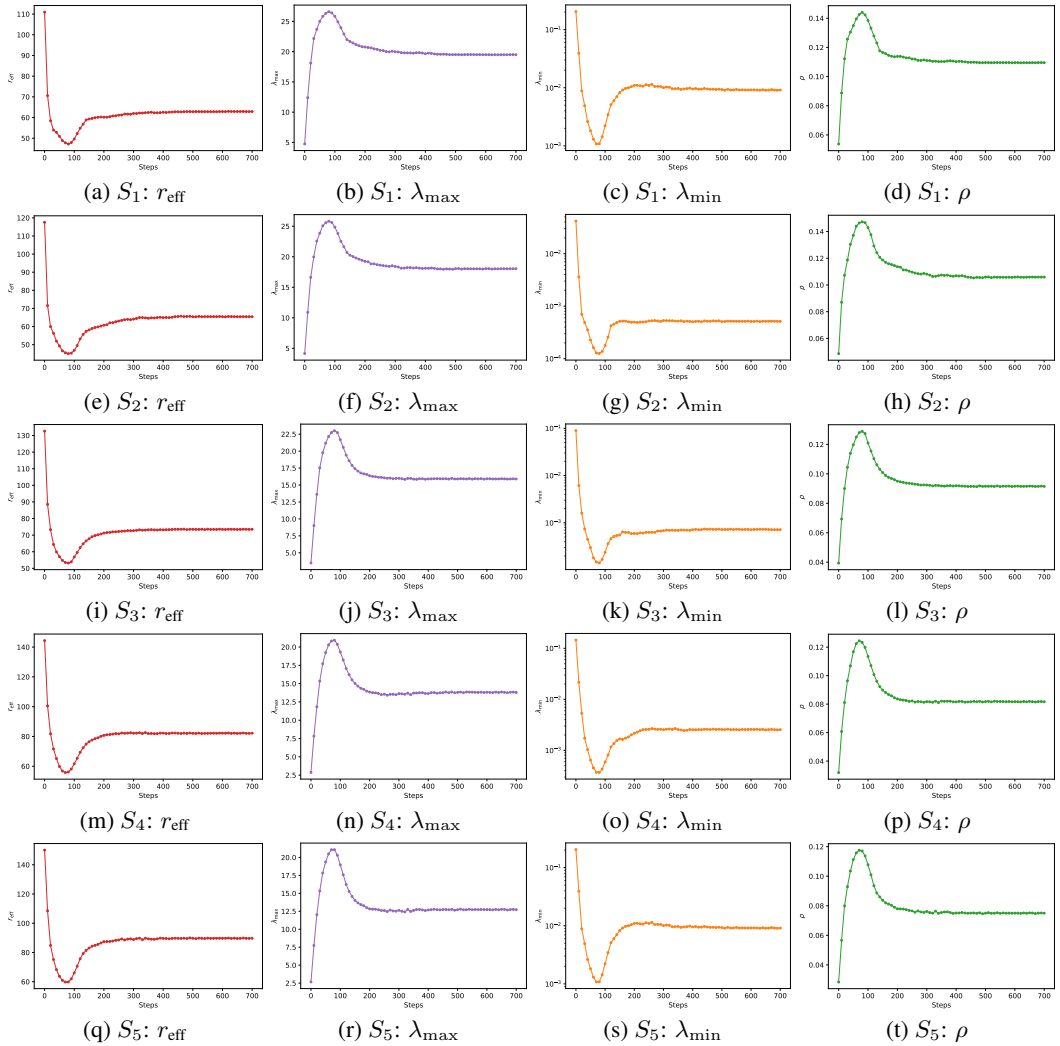
The entries of $P(\boldsymbol{\pi})$ are nonnegative and each row sums to one, so it is a Markov transition matrix. Moreover, $\boldsymbol{\pi}P(\boldsymbol{\pi}) = \boldsymbol{\pi}$, and hence Equation (34) is the length-L_s marginal of a stationary chain. Thus, the Markov dataset is $\{\boldsymbol{s}_i, y_i\}_{i=1}^{\mathcal{D}}$, where $\boldsymbol{s}_i = (s_{i,1}, \ldots, s_{i,L_s})$, $y_i = s_{i,L_s+1}$ is the corresponding target output and X_i is the output of the embedding layer for $\boldsymbol{s}_i$.

For Markov chains, different datasets are characterized by different choices of $\boldsymbol{\pi}$: although the sequence space $\mathcal{V}^{L_s}$ remains fixed, the corresponding probability measure $p_{\boldsymbol{\pi}} \in \mathcal{P}(\mathcal{V}^{L_s})$ on that space changes.

Here we use five artificial Markov datasets, each of which has the same vocabulary $\mathcal{V} = [5]$ and sequence length $L_s = 21$. We set $\beta = 0.2$ in Equation (33), but the probability distribution $\boldsymbol{\pi}$ differs for each dataset, as shown in Table 1. The results in Figure 2 are from training on S_1.

Table 1: Frequency of tokens in the Markov datasets

Dataset	**Token 1**	**Token 2**	**Token 3**	**Token 4**	**Token 5**
S_1	75.02%	18.80%	4.68%	1.18%	0.32%
S_2	46.88%	46.93%	4.68%	1.18%	0.32%
S_3	32.85%	32.78%	32.87%	1.18%	0.32%
S_4	24.97%	24.80%	24.97%	24.94%	0.32%
S_5	20.03%	19.94%	19.89%	20.11%	20.03%



(a) S_1: r_{eff} (b) S_1: $\lambda_{\max}$ (c) S_1: $\lambda_{\min}$ (d) S_1: ρ

(e) S_2: r_{eff} (f) S_2: $\lambda_{\max}$ (g) S_2: $\lambda_{\min}$ (h) S_2: ρ

(i) S_3: r_{eff} (j) S_3: $\lambda_{\max}$ (k) S_3: $\lambda_{\min}$ (l) S_3: ρ

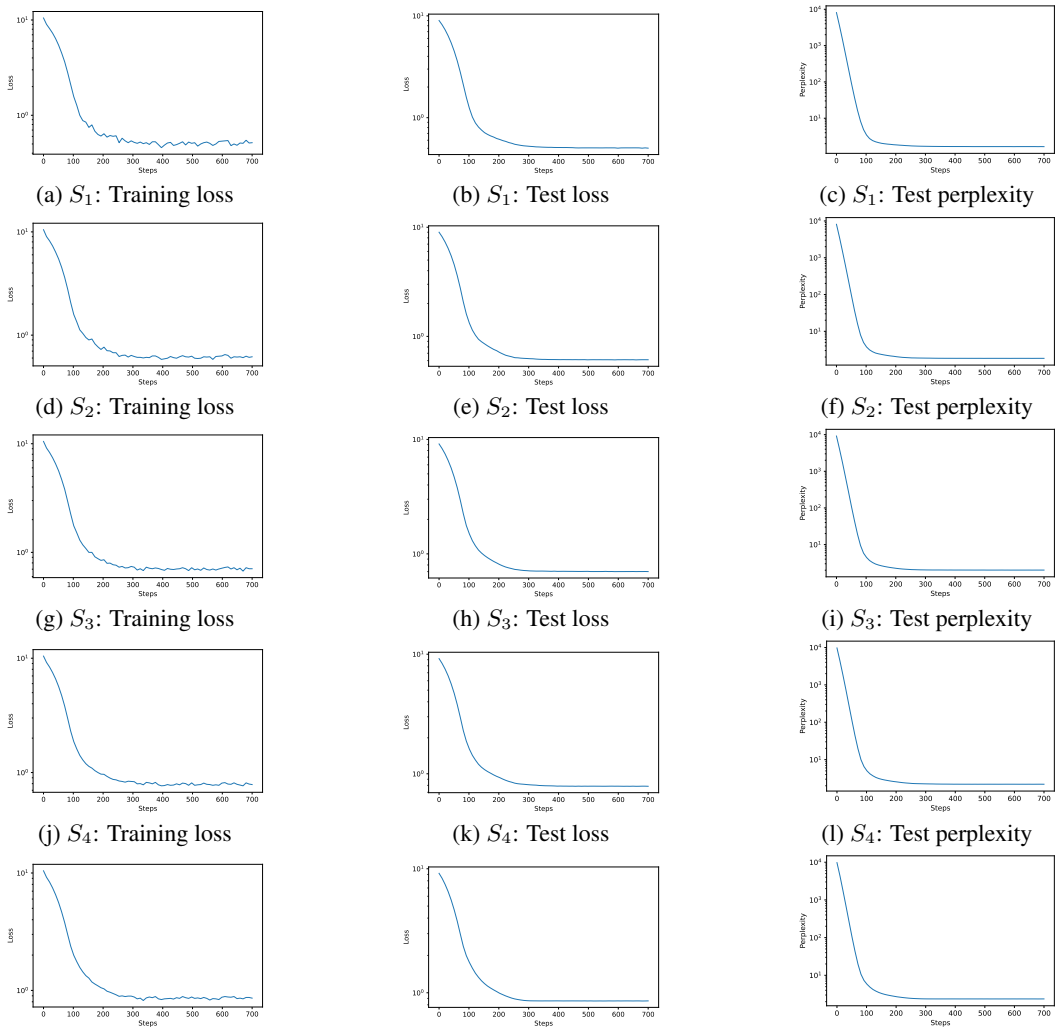
(m) S_4: r_{eff} (n) S_4: $\lambda_{\max}$ (o) S_4: $\lambda_{\min}$ (p) S_4: ρ

(q) S_5: r_{eff} (r) S_5: $\lambda_{\max}$ (s) S_5: $\lambda_{\min}$ (t) S_5: ρ

Figure 5: Dynamics of effective rank r_{eff}, degree of orthogonality ρ, maximum and minimum eigenvalues $\lambda_{\max}, \lambda_{\min}$ for Single-layer LLM on Markov with varying frequency.

D Details for Experiments

We train the LLMs from scratch on an NVIDIA A40 GPU with 40 GB of memory using CUDA 12.4. The GitHub repository will be made available upon publication.



(a) S_1: Training loss (b) S_1: Test loss (c) S_1: Test perplexity

(d) S_2: Training loss (e) S_2: Test loss (f) S_2: Test perplexity

(g) S_3: Training loss (h) S_3: Test loss (i) S_3: Test perplexity

(j) S_4: Training loss (k) S_4: Test loss (l) S_4: Test perplexity

(m) S_5: Training loss (n) S_5: Test loss (o) S_5: Test perplexity

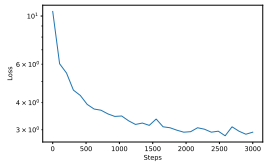
Figure 6: Training and test loss and perplexity for single-layer models S_1 to S_5.

Table 2 summarizes the model and training hyperparameters used for the single-layer LLM, LLaMA2, and Qwen3 experiments. The single-layer LLM is trained on the Markov datasets, whereas LLaMA2 and Qwen3 are trained on WikiText-2. The model size of LLaMA2 is about 400 MB, and the model size of Qwen3 is about 220 MB. The architecture of the single-layer model is similar to that of Qwen3, including, for example, the gated attention mechanism. Since the test dataset has a similar distribution to the training dataset, the metrics are calculated on the test dataset for efficiency. The tokenizer used is llama-7b-hf for both the LLaMA2 and Qwen3 models. The single-layer LLM does not need a tokenizer.

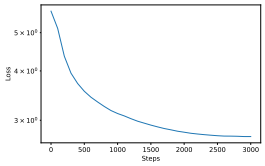
Figure 7 shows the training and test loss and perplexity for the LLaMA2 and Qwen3 models on WikiText-2. The dynamics of the single-layer models are shown in Figure 6.

Table 2: Hyperparameters for the Single-layer LLM, LLaMA2, and Qwen3 experiments.

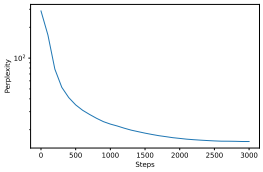
Hyperparameter	Single-layer	LLaMA2	Qwen3
Dataset	Markov	WikiText-2	WikiText-2
Number of layers	1	24	14
Hidden size	48	768	768
Intermediate size	128	2,560	2,048
Attention heads	4	16	16
Activation	SiLU	SiLU	SiLU
RMSNorm configuration	Pre-Norm + Q/K Norm + final Norm	Pre-Norm + final Norm	Pre-Norm + Q/K Norm + final Norm
RMSNorm epsilon	10^{-6}	10^{-6}	10^{-6}
Weight tying	No	No	No
Linear bias	None	None	None
RoPE	Yes	Yes	Yes
Causal mask	Yes	Yes	Yes
σ	0.02	0.02	0.02
Learning rate	10^{-3}	10^{-3}	10^{-3}
Learning rate schedule	Cosine ($\eta_{\min} = 10^{-4}$)	Cosine ($\eta_{\min} = 10^{-4}$)	Cosine ($\eta_{\min} = 10^{-4}$)
Per-device batch size	128	128	128
Total batch size	512	512	512
Training steps	700	3,000	3,000
Optimizer	Adam	Adam	AdamW
Precision	bfloat16	bfloat16	bfloat16
Gradient clipping	0	0	0



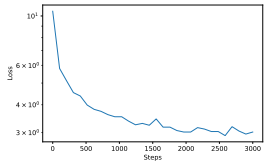
(a) LLaMA2: Training loss



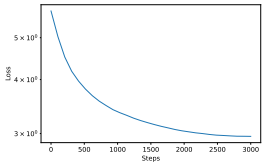
(b) LLaMA2: Test loss



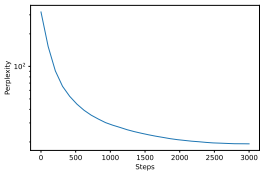
(c) LLaMA2: Test perplexity



(d) Qwen3: Training loss



(e) Qwen3: Test loss



(f) Qwen3: Test perplexity

Figure 7: Training and test loss and perplexity for LLaMA2 and Qwen3.

E PROOFS

This section contains the proofs presented in Section 2.2. The following list indexes the results and their corresponding proofs.

- Proposition 2.1: Appendix E.1.
- Proposition 2.2: Appendix E.2.
- Proposition 2.3: Appendix E.3.
- Proposition 2.4: Appendix E.4.
- Theorem 2.1: Appendix E.5.
- Theorem 2.2: Appendix E.6.
- Theorem 2.3: Appendix E.7.
- Proposition 2.5: Appendix E.8.
- Proposition 2.6: Appendix E.9.
- Proposition 2.7: Appendix E.10.
- Theorem 2.4: Appendix E.11.

E.1 Proof of Proposition 2.1

Proof. By the definition of normalization,

$$\tilde{\boldsymbol{\phi}}_j(\boldsymbol{x}) = \frac{\boldsymbol{\phi}_j(\boldsymbol{x})}{\sqrt{(M)_{jj}}}. \tag{35}$$

Since $M \succ 0$, we have $(M)_{jj} > 0$ for any $j \in \{1, \dots, N\}$, and hence $\tilde{\boldsymbol{\phi}}_j$ is well-defined. Let $M_{\tilde{f}}$ be the Gram matrix of $\{\tilde{\boldsymbol{\phi}}_j\}_{j=1}^N$. For any $i, j \in \{1, \dots, N\}$, using the bilinearity of the inner product gives

$$(M_{\tilde{f}})_{ij} = \int_\Omega \left\langle \tilde{\boldsymbol{\phi}}_i(\boldsymbol{x}), \tilde{\boldsymbol{\phi}}_j(\boldsymbol{x}) \right\rangle \mathrm{d}\boldsymbol{x} \tag{36}$$

$$= \frac{1}{\sqrt{(M)_{ii}(M)_{jj}}} \int_\Omega \langle \boldsymbol{\phi}_i(\boldsymbol{x}), \boldsymbol{\phi}_j(\boldsymbol{x}) \rangle \mathrm{d}\boldsymbol{x} \tag{37}$$

$$= \frac{(M)_{ij}}{\sqrt{(M)_{ii}(M)_{jj}}} = (\tilde{M})_{ij}. \tag{38}$$

Therefore, every entry of $M_{\tilde{f}}$ equals the corresponding entry of $\tilde{M}$, and hence

$$M_{\tilde{f}} = \tilde{M}. \tag{39}$$

□

E.2 Proof of Proposition 2.2

Proof. For a fixed input X_k, each basis output $\Phi_i(X_k) \in \mathbb{R}^{L_s \times d_{\text{out}}}$ can be viewed as a vector in $\mathbb{R}^{L_s d_{\text{out}}}$ by vectorization. Therefore, we have N vectors in a space of dimension $L_s d_{\text{out}}$. Since $N > L_s d_{\text{out}}$, these vectors must be linearly dependent.

For the whole dataset $\{X_k\}_{k=1}^{\mathcal{D}}$, define the stacked representation

$$\mathbf{z}_i := \left[\mathrm{vec}(\Phi_i(X_1)); \dots; \mathrm{vec}(\Phi_i(X_{\mathcal{D}}))\right] \in \mathbb{R}^{\mathcal{D} L_s d_{\text{out}}}. \tag{40}$$

Then linear independence of $\{\Phi_i(X_k)\}_{i=1}^N$ over the dataset is equivalent to linear independence of $\{\mathbf{z}_i\}_{i=1}^N$ in $\mathbb{R}^{\mathcal{D} L_s d_{\text{out}}}$. Hence, since $N > \mathcal{D} L_s d_{\text{out}}$, $\{\Phi_i(X_k)\}_{i=1}^N$ must be linearly dependent over the dataset. □

E.3 Proof of Proposition 2.3

Proof. Let $M(t)$ denote the Gram matrix of the basis $\{\Phi_i(t)\}_{i=1}^N$ at training time $t \in [0, T]$, where T denotes the terminal training time. We have

$$M(t) = \begin{bmatrix} \langle \Phi_1(t), \Phi_1(t) \rangle & \cdots & \langle \Phi_1(t), \Phi_N(t) \rangle \\ \vdots & \ddots & \vdots \\ \langle \Phi_N(t), \Phi_1(t) \rangle & \cdots & \langle \Phi_N(t), \Phi_N(t) \rangle \end{bmatrix}. \tag{41}$$

Since the parameters of LLMs are randomly initialized from a non-atomic distribution, then the possibility $P\left[\Phi_i(0) \in \mathrm{span}\{\Phi_j(0)\}_{j \neq i}\right] = 0$ for any i, hence $P\left[\{\Phi_i(0)\}_{i=1}^N \text{ are linearly dependent}\right] = 0$, so the initial Gram matrix $M(0)$ is positive definite almost surely, i.e., $\det M(0) > 0$ almost surely.

Suppose $\mathcal{L}(\theta_{\mathcal{U}})$ is the loss function and $\theta_{\mathcal{U}}$ collects the parameters underlying $\mathcal{U}$, excluding the LM head. Let $r = \frac{\partial \mathcal{L}}{\partial \mathcal{U}}$ be the output residual backpropagated through the LM head and let $J = \frac{\partial \mathcal{U}}{\partial \theta_{\mathcal{U}}}$ be the Jacobian of $\mathcal{U}$. Under gradient flow, $\dot{\theta}_{\mathcal{U}} = -J^\top r$. Since the output $\mathcal{U}(X) = \sum_{i=1}^N \alpha_i \Phi_i(X)$ is a linear combination of the basis functions, we have $J = \sum_{i=1}^N J_i$, where $J_i = \frac{\partial(\alpha_i \Phi_i(X))}{\partial \theta_{\mathcal{U}}}$. Suppose $\tilde{J}_i = (J_i - \frac{\partial \alpha_i}{\partial \theta_{\mathcal{U}}} \Phi_i(X))/\alpha_i$, then

$$\dot{\Phi}_i = \frac{\partial \Phi_i(X)}{\partial \theta_{\mathcal{U}}} \dot{\theta}_{\mathcal{U}} = -\tilde{J}_i J^\top r = -\tilde{J}_i J_i^\top r - \sum_{j \neq i} \tilde{J}_i J_j^\top r. \tag{42}$$

Thus, since $\alpha_i(t) > 0$, then Φ_i is differentiable with respect to t. Then,

$$\dot{M}_{ij}(t) = \langle \dot{\Phi}_i(t), \Phi_j(t) \rangle + \langle \Phi_i(t), \dot{\Phi}_j(t) \rangle. \tag{43}$$

makes $M(t)$ continuous with respect to t. Therefore, $\lambda_{\min}(M(t))$ is also continuous with respect to t. Thus, there exists a time $T > 0$ such that $\lambda_{\min}(M(t)) > 0$ for any $t \in [0, T]$, which implies $\det M(t) > 0$ for any $t \in [0, T]$.

Now we discuss the upper bound of T. Let $\gamma_0 := \lambda_{\min}(M(0)) > 0$. Assume there exists a constant $\eta > 0$ such that

$$\lambda_{\max}(M(t)) \leq -\eta, \qquad \forall t \in [0, T_\star], \tag{44}$$

where

$$T_\star := \frac{\gamma_0}{\eta}. \tag{45}$$

For differentiable symmetric matrix trajectories, the minimum eigenvalue satisfies

$$\lambda_{\min}(M(t)) \leq \lambda_{\min}(M(0)) + \int_0^t \lambda_{\max}(\dot{M}(s))\, \mathrm{d}s \leq \gamma_0 - \eta t. \tag{46}$$

At $t = T_\star$, we obtain

$$\lambda_{\min}(M(T_\star)) \leq 0. \tag{47}$$

On the other hand, $M(t) \succeq 0$ and hence

$$\lambda_{\min}(M(T_\star)) = 0, \tag{48}$$

which implies

$$\det M(T_\star) = 0. \tag{49}$$

So $T_\star = \gamma_0/\eta$ is the time when the Gram matrix degenerates, as well as the basis functions become linearly dependent. Therefore, for any $t \in [0, T_\star]$, we have $\det M(t) > 0$. □

E.4 Proof of Proposition 2.4

Proof. Write $\lambda_i(0) = 1 + \delta_i$ for the eigenvalues of $M(0)$. Since $\|\mathcal{E}_0\|_2 \leq \varepsilon_0$, Weyl's inequality gives

$$|\delta_i| \leq \varepsilon_0, \qquad i \in [N]. \tag{50}$$

Since $\varepsilon_0 < 1$, all eigenvalues are positive. Moreover, the normalization of the Gram matrix gives $M_{ii}(0) = 1$ for every i, and hence

$$\sum_{i=1}^N \delta_i = \mathrm{Tr}(M(0)) - N = 0. \tag{51}$$

Let $f(x) = x \log x$. Taylor's theorem around $x = 1$ yields, uniformly for $|\delta| \leq \varepsilon_0 < 1$,

$$f(1 + \delta) = \delta + \mathcal{O}(\delta^2). \tag{52}$$

Using the definition of effective rank and the preceding trace identity, we obtain

$$\begin{aligned} \log r_{\text{eff}}(M(0)) &= \log N - \frac{1}{N} \sum_{i=1}^N \lambda_i(0) \log \lambda_i(0) \\ &= \log N - \frac{1}{N} \sum_{i=1}^N \left(\delta_i + \mathcal{O}(\delta_i^2) \right) \\ &= \log N - \mathcal{O}\left(\frac{1}{N} \sum_{i=1}^N \delta_i^2 \right) \\ &= \log N - \mathcal{O}\left(\varepsilon_0^2 \right). \end{aligned} \tag{53}$$

Exponentiating this identity proves

$$r_{\text{eff}}(M(0)) = N \exp\left(-\mathcal{O}(\varepsilon_0^2) \right) = N \left(1 - \mathcal{O}(\varepsilon_0^2) \right). \tag{54}$$

□

E.5 Proof of Theorem 2.1

Before we prove Theorem 2.1, we first establish two lemmas.

Lemma E.1. *Given $X \in X_{i\,i=1}^{\mathcal{D}}$, suppose $\mathcal{U} = X + \mathcal{U}_{\text{attn}} + \mathcal{U}_{\text{mlp}}$, where $\mathcal{U}_{\text{attn}} = W_{SQK}VW_O$, $\mathcal{U}_{\text{mlp}} = GW_{\text{down}}$, and $W_{SQK} = \text{Softmax}(QK^\top)$. Define $E_{\text{attn}} := \frac{\partial \mathcal{L}}{\partial \mathcal{U}_{\text{attn}}}$ and $E_{\text{mlp}} := \frac{\partial \mathcal{L}}{\partial \mathcal{U}_{\text{mlp}}}$ as the local partial derivatives at the final additive node, treating the two summands as independent arguments. In particular, E_{attn} contains only the direct contribution through the attention residual branch and excludes the contribution backpropagated through the MLP. The weight matrices of the single-layer LLM follow the gradient flow*

$$\begin{aligned}
\dot{W}_V &= -X^\top W_{SQK}^\top E_A W_O^\top, \\
\dot{W}_Q &= -X^\top E_{QK} K, \\
\dot{W}_K &= -X^\top E_{QK}^\top Q, \\
\dot{W}_O &= -(W_{SQK}V)^\top E_A, \\
\dot{W}_{\text{down}} &= -G^\top E_{\text{mlp}}, \\
\dot{W}_{\text{up}} &= -Z^\top E_{\text{up}},
\end{aligned} \tag{55}$$

where $E_{\text{up}} = \left(E_{\text{mlp}} W_{\text{down}}^\top\right) \odot \text{SiLU}'(Z_{\text{up}})$, $E_Z = E_{\text{up}} W_{\text{up}}^\top$, $E_A = E_{\text{attn}} + E_Z$, $E_S = \left(E_A W_O^\top\right) V^\top$, and $E_{QK} = W_{SQK} \odot \left(E_S - \left((W_{SQK} \odot E_S)\,\mathbf{1}_{L_s}\right)\mathbf{1}_{L_s}^\top\right)$. $\odot$ denotes the element-wise (Hadamard) product.

Proof. We use the Frobenius inner product $\langle B, C\rangle_F = \text{Tr}(B^\top C)$. The dependence of the scalar loss $\mathcal{L}$ on the dataset, targets, and the LM-head logits is implicit. Taking its total differential with respect to the LM head and the two trainable additive summands gives

$$\mathrm{d}\mathcal{L} = \left\langle \frac{\partial \mathcal{L}}{\partial W_{\text{lm}}}, \mathrm{d}W_{\text{lm}} \right\rangle_F + \langle E_{\text{attn}}, \mathrm{d}\mathcal{U}_{\text{attn}}\rangle_F + \langle E_{\text{mlp}}, \mathrm{d}\mathcal{U}_{\text{mlp}}\rangle_F . \tag{56}$$

Let $A = W_{SQK}VW_O$, so that $Z = X + A$ and $\mathcal{U} = Z + GW_{\text{down}}$. Since X is fixed,

$$\mathrm{d}\mathcal{L} - \left\langle \frac{\partial \mathcal{L}}{\partial W_{\text{lm}}}, \mathrm{d}W_{\text{lm}} \right\rangle_F = \langle E_{\text{attn}}, \mathrm{d}A\rangle_F + \langle E_{\text{mlp}}, \mathrm{d}GW_{\text{down}} + G\mathrm{d}W_{\text{down}}\rangle_F . \tag{57}$$

Since $G = \text{SiLU}(Z_{\text{up}})$ and $Z_{\text{up}} = ZW_{\text{up}}$, we have

$$\begin{aligned}
\langle E_{\text{mlp}}, \mathrm{d}GW_{\text{down}}\rangle_F &= \left\langle E_{\text{mlp}} W_{\text{down}}^\top, \mathrm{d}G\right\rangle_F \\
&= \left\langle \left(E_{\text{mlp}} W_{\text{down}}^\top\right) \odot \text{SiLU}'(Z_{\text{up}}), \mathrm{d}Z_{\text{up}}\right\rangle_F \\
&= \langle E_{\text{up}}, \mathrm{d}ZW_{\text{up}} + Z\mathrm{d}W_{\text{up}}\rangle_F \\
&= \left\langle E_{\text{up}} W_{\text{up}}^\top, \mathrm{d}Z\right\rangle_F + \left\langle Z^\top E_{\text{up}}, \mathrm{d}W_{\text{up}}\right\rangle_F \\
&= \langle E_Z, \mathrm{d}Z\rangle_F + \left\langle Z^\top E_{\text{up}}, \mathrm{d}W_{\text{up}}\right\rangle_F .
\end{aligned} \tag{58}$$

Moreover,

$$\langle E_{\text{mlp}}, G\mathrm{d}W_{\text{down}}\rangle_F = \left\langle G^\top E_{\text{mlp}}, \mathrm{d}W_{\text{down}}\right\rangle_F . \tag{59}$$

Combining the preceding identities and using $\mathrm{d}Z = \mathrm{d}A$ gives

$$\mathrm{d}\mathcal{L} - \left\langle \frac{\partial \mathcal{L}}{\partial W_{\text{lm}}}, \mathrm{d}W_{\text{lm}} \right\rangle_F = \langle E_A, \mathrm{d}A\rangle_F + \left\langle Z^\top E_{\text{up}}, \mathrm{d}W_{\text{up}}\right\rangle_F + \left\langle G^\top E_{\text{mlp}}, \mathrm{d}W_{\text{down}}\right\rangle_F . \tag{60}$$

Therefore,

$$\frac{\partial \mathcal{L}}{\partial W_{\text{down}}} = G^\top E_{\text{mlp}}, \qquad \frac{\partial \mathcal{L}}{\partial W_{\text{up}}} = Z^\top E_{\text{up}}. \tag{61}$$

It remains to compute the differential of the attention term. By $A = W_{SQK} V W_O$,

$$\begin{aligned}
\langle E_A, \mathrm{d}A \rangle_F &= \langle E_A, \mathrm{d}W_{SQK} V W_O + W_{SQK} \mathrm{d}V W_O + W_{SQK} V \mathrm{d}W_O \rangle_F \\
&= \left\langle \left(E_A W_O^\top\right) V^\top, \mathrm{d}W_{SQK} \right\rangle_F + \left\langle W_{SQK}^\top E_A W_O^\top, \mathrm{d}V \right\rangle_F \\
&\quad + \left\langle (W_{SQK} V)^\top E_A, \mathrm{d}W_O \right\rangle_F \\
&= \langle E_S, \mathrm{d}W_{SQK} \rangle_F + \left\langle W_{SQK}^\top E_A W_O^\top, \mathrm{d}V \right\rangle_F \\
&\quad + \left\langle (W_{SQK} V)^\top E_A, \mathrm{d}W_O \right\rangle_F .
\end{aligned} \tag{62}$$

Since $V = X W_V$, the second term satisfies

$$\left\langle W_{SQK}^\top E_A W_O^\top, \mathrm{d}V \right\rangle_F = \left\langle X^\top W_{SQK}^\top E_A W_O^\top, \mathrm{d}W_V \right\rangle_F . \tag{63}$$

Hence,

$$\frac{\partial \mathcal{L}}{\partial W_O} = (W_{SQK} V)^\top E_A, \qquad \frac{\partial \mathcal{L}}{\partial W_V} = X^\top W_{SQK}^\top E_A W_O^\top . \tag{64}$$

Finally, write $R = Q K^\top$. The differential of the row-wise softmax is

$$\mathrm{d}W_{SQK} = W_{SQK} \odot \left(\mathrm{d}R - \left((W_{SQK} \odot \mathrm{d}R)\, \mathbf{1}_{L_s} \right) \mathbf{1}_{L_s}^\top \right) . \tag{65}$$

Consequently,

$$\begin{aligned}
\langle E_S, \mathrm{d}W_{SQK} \rangle_F &= \left\langle W_{SQK} \odot \left(E_S - \left((W_{SQK} \odot E_S)\, \mathbf{1}_{L_s} \right) \mathbf{1}_{L_s}^\top \right), \mathrm{d}R \right\rangle_F \\
&= \left\langle E_{QK}, \mathrm{d}Q K^\top + Q \mathrm{d}K^\top \right\rangle_F \\
&= \langle E_{QK} K, \mathrm{d}Q \rangle_F + \left\langle E_{QK}^\top Q, \mathrm{d}K \right\rangle_F \\
&= \left\langle X^\top E_{QK} K, \mathrm{d}W_Q \right\rangle_F + \left\langle X^\top E_{QK}^\top Q, \mathrm{d}W_K \right\rangle_F ,
\end{aligned} \tag{66}$$

where the last equality uses $Q = X W_Q$ and $K = X W_K$. Thus,

$$\frac{\partial \mathcal{L}}{\partial W_Q} = X^\top E_{QK} K, \qquad \frac{\partial \mathcal{L}}{\partial W_K} = X^\top E_{QK}^\top Q. \tag{67}$$

Collecting the coefficients of the seven differentials and applying $\dot{W} = -\partial \mathcal{L} / \partial W$ proves Equation (55). □

Remark E.1. The identities in Equation (55) are written for a generic sample X. For the empirical loss $\mathcal{L} = \mathcal{D}^{-1} \sum_{k=1}^{\mathcal{D}} \ell_k$, let $E_{A,k}$ denote the analogue of E_A associated with the k-th sample loss ℓ_k. The full gradient flow averages the per-sample contributions; for example,

$$\dot{W}_V = -\frac{1}{\mathcal{D}} \sum_{k=1}^{\mathcal{D}} X_k^\top W_{SQK,k}^\top E_{A,k} W_O^\top , \tag{68}$$

and the other weight flows are averaged analogously. Since $\mathcal{D}$ is finite and the estimates used below hold uniformly over $k \in [\mathcal{D}]$, this averaging does not change any of the subsequent $\mathcal{O}(\sigma^q)$ orders.

We next make the output derivatives explicit for the empirical cross-entropy loss $\mathcal{L}$ defined above. For $i \in [\mathcal{D}]$, let

$$\boldsymbol{p}_{\theta,i} = \mathrm{Softmax}\left(W_{\mathrm{lm}} \mathcal{U}(X_i)_{L_s,:}^\top \right), \tag{69}$$

and let $\boldsymbol{e}_{y_i}$ be the one-hot vector associated with y_i. Define the cross-entropy residual by

$$\mathbb{A}_i := \boldsymbol{p}_{\theta,i} - \boldsymbol{e}_{y_i} = \frac{\partial \ell(\mathcal{U}(X_i)_{L_s}, y_i)}{\partial \left(W_{\mathrm{lm}} \mathcal{U}(X_i)_{L_s,:}^\top \right)}. \tag{70}$$

Lemma E.2. *For $r \in \{\mathrm{attn}, \mathrm{mlp}\}$, the contribution of the i-th cross-entropy term to the gradient with respect to the r-th summand is*

$$\frac{\partial \ell(\mathcal{U}(X_i)_{L_s}, y_i)}{\partial \mathcal{U}_r(X_i)} = \boldsymbol{e}_{L_s} \left[W_{\mathrm{lm}}^\top \mathbb{A}_i \right]^\top . \tag{71}$$

Consequently,

$$\frac{\partial \mathcal{L}}{\partial \mathcal{U}_r(X_i)} = \frac{1}{\mathcal{D}} \boldsymbol{e}_{L_s} \left[W_{\mathrm{lm}}^\top \mathbb{A}_i \right]^\top, \qquad r \in \{\mathrm{attn}, \mathrm{mlp}\}, \tag{72}$$

and

$$\frac{\partial \mathcal{L}}{\partial W_{\mathrm{lm}}} = \frac{1}{\mathcal{D}} \sum_{i=1}^{\mathcal{D}} \mathbb{A}_i \mathcal{U}(X_i)_{L_s,:}. \tag{73}$$

Proof. Let $\boldsymbol{h}_i = \mathcal{U}(X_i)_{L_s,:}^\top$ and $\boldsymbol{z}_i = W_{\mathrm{lm}} \boldsymbol{h}_i$. Then

$$\begin{aligned} \ell(\mathcal{U}(X_i)_{L_s}, y_i) &= -\boldsymbol{e}_{y_i}^\top \boldsymbol{z}_i + \log \left(\sum_{v \in \mathcal{V}} \exp\left((\boldsymbol{z}_i)_v \right) \right), \\ \mathrm{d}\ell(\mathcal{U}(X_i)_{L_s}, y_i) &= -\boldsymbol{e}_{y_i}^\top \mathrm{d}\boldsymbol{z}_i + \frac{\sum_{v \in \mathcal{V}} \exp\left((\boldsymbol{z}_i)_v \right) \mathrm{d}(\boldsymbol{z}_i)_v}{\sum_{v \in \mathcal{V}} \exp\left((\boldsymbol{z}_i)_v \right)} \\ &= \mathbb{A}_i^\top \mathrm{d}\boldsymbol{z}_i. \end{aligned} \tag{74}$$

Since

$$\mathrm{d}\boldsymbol{z}_i = \mathrm{d}W_{\mathrm{lm}} \boldsymbol{h}_i + W_{\mathrm{lm}} \mathrm{d}\boldsymbol{h}_i, \qquad \mathrm{d}\boldsymbol{h}_i = \mathrm{d}\mathcal{U}(X_i)^\top \boldsymbol{e}_{L_s}, \tag{75}$$

we obtain

$$\begin{aligned} \mathrm{d}\ell(\mathcal{U}(X_i)_{L_s}, y_i) &= \mathbb{A}_i^\top \left(\mathrm{d}W_{\mathrm{lm}} \boldsymbol{h}_i + W_{\mathrm{lm}} \mathrm{d}\boldsymbol{h}_i \right) \\ &= \mathrm{Tr}\left(\boldsymbol{h}_i \mathbb{A}_i^\top \mathrm{d}W_{\mathrm{lm}} \right) \\ &\quad + \left[W_{\mathrm{lm}}^\top \mathbb{A}_i \right]^\top \mathrm{d}\mathcal{U}(X_i)^\top \boldsymbol{e}_{L_s} \\ &= \left\langle \mathbb{A}_i \boldsymbol{h}_i^\top, \mathrm{d}W_{\mathrm{lm}} \right\rangle_F \\ &\quad + \left\langle \boldsymbol{e}_{L_s} \left[W_{\mathrm{lm}}^\top \mathbb{A}_i \right]^\top, \mathrm{d}\mathcal{U}(X_i) \right\rangle_F. \end{aligned} \tag{76}$$

Using $\mathcal{U} = X + \mathcal{U}_{\mathrm{attn}} + \mathcal{U}_{\mathrm{mlp}}$ and $\mathrm{d}X = 0$, it follows that

$$\begin{aligned} \mathrm{d}\mathcal{U}(X_i) &= \mathrm{d}\mathcal{U}_{\mathrm{attn}}(X_i) + \mathrm{d}\mathcal{U}_{\mathrm{mlp}}(X_i), \\ \mathrm{d}\ell(\mathcal{U}(X_i)_{L_s}, y_i) &= \left\langle \mathbb{A}_i \boldsymbol{h}_i^\top, \mathrm{d}W_{\mathrm{lm}} \right\rangle_F \\ &\quad + \sum_{r \in \{\mathrm{attn}, \mathrm{mlp}\}} \left\langle \boldsymbol{e}_{L_s} \left[W_{\mathrm{lm}}^\top \mathbb{A}_i \right]^\top, \mathrm{d}\mathcal{U}_r(X_i) \right\rangle_F. \end{aligned} \tag{77}$$

Thus, by the definition of the matrix derivative,

$$\frac{\partial \ell(\mathcal{U}(X_i)_{L_s}, y_i)}{\partial W_{\mathrm{lm}}} = \mathbb{A}_i \mathcal{U}(X_i)_{L_s,:}, \tag{78}$$

and Equation (71) follows. Summing over $i \in [\mathcal{D}]$ and using $\mathcal{L} = \mathcal{D}^{-1} \sum_{i=1}^{\mathcal{D}} \ell(\mathcal{U}(X_i)_{L_s}, y_i)$ yields

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathcal{U}_r(X_i)} &= \frac{1}{\mathcal{D}} \frac{\partial \ell(\mathcal{U}(X_i)_{L_s}, y_i)}{\partial \mathcal{U}_r(X_i)}, \\ \frac{\partial \mathcal{L}}{\partial W_{\mathrm{lm}}} &= \frac{1}{\mathcal{D}} \sum_{i=1}^{\mathcal{D}} \frac{\partial \ell(\mathcal{U}(X_i)_{L_s}, y_i)}{\partial W_{\mathrm{lm}}}. \end{aligned} \tag{79}$$

□

Then we can prove Theorem 2.1 based on the above two lemmas.

Proof. Since $\mathcal{D}$ is finite and $\max_{i \in [\mathcal{D}]} \|X_i\|_F = \mathcal{O}(1)$ by hypothesis, every $\mathcal{O}(\sigma^k)$ estimate below carries a constant that can be chosen uniformly over the sample index $i \in [\mathcal{D}]$; below we therefore write the estimates for a generic embedding output X, and they hold uniformly over the dataset. At initialization, $Q = XW_Q$, $K = XW_K$, and $V = XW_V$ are $\mathcal{O}(\sigma)$. Hence $QK^\top = \mathcal{O}(\sigma^2)$ and

$$W_{SQK} = \mathrm{Softmax}(\mathcal{O}(\sigma^2)) = \mathcal{O}(1). \tag{80}$$

Moreover, $Z = X + \mathcal{O}(\sigma^2) = \mathcal{O}(1)$ and $Z_{\text{up}} = ZW_{\text{up}} = \mathcal{O}(\sigma)$. Since $\text{SiLU}(x) = \mathcal{O}(x)$ and $\text{SiLU}'(x) = \mathcal{O}(1)$ in a neighborhood of zero, $G = \mathcal{O}(\sigma)$. Thus, $\mathcal{U} = X + \mathcal{O}(\sigma^2) = \mathcal{O}(1)$.

The logits are $W_{\text{lm}}\mathcal{U}_{L_s,:}^\top = \mathcal{O}(\sigma)$. The cross-entropy residual in Equation (70) is therefore $\mathbb{A}_i = \mathcal{O}(1)$, and Lemma E.2 yields

$$E_{\text{attn}} = E_{\text{mlp}} = \mathcal{O}(\sigma). \tag{81}$$

Using the definitions in Lemma E.1, we obtain

$$E_{\text{up}} = \mathcal{O}(\sigma^2), \qquad E_Z = \mathcal{O}(\sigma^3), \qquad E_A = \mathcal{O}(\sigma), \qquad E_S = \mathcal{O}(\sigma^3), \qquad E_{QK} = \mathcal{O}(\sigma^3). \tag{82}$$

Substitution into Equation (55) gives

$$\begin{aligned} \dot{W}_Q &= -X^\top E_{QK} K = \mathcal{O}(\sigma^4), & \dot{W}_K &= -X^\top E_{QK}^\top Q = \mathcal{O}(\sigma^4), \\ \dot{W}_V &= -X^\top W_{SQK}^\top E_A W_O^\top = \mathcal{O}(\sigma^2), & \dot{W}_O &= -(W_{SQK} V)^\top E_A = \mathcal{O}(\sigma^2), \\ \dot{W}_{\text{up}} &= -Z^\top E_{\text{up}} = \mathcal{O}(\sigma^2), & \dot{W}_{\text{down}} &= -G^\top E_{\text{mlp}} = \mathcal{O}(\sigma^2). \end{aligned} \tag{83}$$

□

E.6 Proof of Theorem 2.2

Before we prove Theorem 2.2, we establish a lemma.

Lemma E.3. *Suppose every entry of M is $\mathcal{O}(\sigma)$, then, entrywise as $\sigma \to 0$,*

$$\text{SiLU}(M) = \frac{1}{2}M + \frac{1}{4}(M \odot M) + \mathcal{O}(\sigma^4). \tag{84}$$

Hence, if M is effectively dominated by N rank-one components, then, up to an entrywise $\mathcal{O}(\sigma^4)$ residual, $\text{SiLU}(M)$ is effectively dominated by at most $\frac{N(N+3)}{2}$ rank-one components.

Proof. The scalar Taylor expansion at the origin is

$$\text{SiLU}(x) = \frac{x}{2} + \frac{x^2}{4} - \frac{x^4}{48} + \mathcal{O}(x^6). \tag{85}$$

Applying this expansion entrywise gives the stated matrix expansion. Moreover, if $M = \sum_{j=1}^N \lambda_j \boldsymbol{u}_j \boldsymbol{v}_j^\top$, then $M \odot M$ is spanned by the $N(N+1)/2$ rank-one matrices $(\boldsymbol{u}_j \odot \boldsymbol{u}_k)(\boldsymbol{v}_j \odot \boldsymbol{v}_k)^\top$ with $1 \le j \le k \le N$. Together with the N components of M, this gives at most $N + N(N+1)/2 = N(N+3)/2$ rank-one components. □

Then we can prove Theorem 2.2.

Proof. Suppose $u_{r,j,k}$ and $v_{r,j,k}$ denote the k-th elements of the vectors $\boldsymbol{u}_{r,j}$ and $\boldsymbol{v}_{r,j}$, respectively. Then, $\boldsymbol{w}_{r,i}^\top \approx \sum_{j=1}^{N_r} \lambda_{r,j} u_{r,j,i} \boldsymbol{v}_{r,j}^\top$ for $r \in \{V, O, \text{up}, \text{down}\}$. Thus, for $i \in [d_{\text{attn}}]$,

$$\begin{aligned} \boldsymbol{a}_i(X) &\approx \frac{1}{L_s} \mathbf{1}_{L_s} \mathbf{1}_{L_s}^\top X W_V \boldsymbol{e}_i \\ &\approx \sum_{j=1}^{N_V} \lambda_{V,j} v_{V,j,i} \left(\frac{1}{L_s} \mathbf{1}_{L_s} \mathbf{1}_{L_s}^\top X \boldsymbol{u}_{V,j} \right), \\ \Phi_i^{\text{attn}}(X) &= \frac{\boldsymbol{a}_i(X) \boldsymbol{w}_{O,i}^\top}{\alpha_i^{\text{attn}}} \\ &\approx \sum_{j=1}^{N_V} \sum_{k=1}^{N_O} \frac{\lambda_{V,j} \lambda_{O,k} v_{V,j,i} u_{O,k,i}}{\alpha_i^{\text{attn}}} \left(\frac{1}{L_s} \mathbf{1}_{L_s} \mathbf{1}_{L_s}^\top X \boldsymbol{u}_{V,j} \right) \boldsymbol{v}_{O,k}^\top \\ &= \frac{1}{L_s \alpha_i^{\text{attn}}} \mathbf{1}_{L_s} \mathbf{1}_{L_s}^\top X \left(\sum_{j=1}^{N_V} \sum_{k=1}^{N_O} \lambda_{j,k,i}^{V,O} \boldsymbol{u}_{V,j} \boldsymbol{v}_{O,k}^\top \right), \end{aligned} \tag{86}$$

where $\lambda^{V,O}_{j,k,i} = \lambda_{V,j}\lambda_{O,k}v_{V,j,i}u_{O,k,i}$. Therefore, during the focus stage, every attention basis function is effectively contained in a subspace spanned by at most $N_V N_O$ rank-one matrices with a common sequence-direction factor $\mathbf{1}_{L_s}$. That is, the effective dimensionality of $\mathrm{span}\{\Phi^{\mathrm{attn}}_i(X)\}_{i=1}^{d_h}$ is less than or equal to $N_V N_O$.

For $\Phi^{\mathrm{mlp}}_i(X)$, substituting the low-rank decomposition of W_{up} into $Z_{\mathrm{up}} = ZW_{\mathrm{up}}$ and using Equation (17) gives

$$\begin{aligned} Z_{\mathrm{up}}(X) &\approx \left(X + \frac{1}{L_s}\mathbf{1}_{L_s}\mathbf{1}_{L_s}^\top XW_VW_O\right)\left(\sum_{j=1}^{N_{\mathrm{up}}}\lambda_{\mathrm{up},j}\boldsymbol{u}_{\mathrm{up},j}\boldsymbol{v}_{\mathrm{up},j}^\top\right) \\ &= \sum_{j=1}^{N_{\mathrm{up}}}\lambda_{\mathrm{up},j}\left(X + \frac{1}{L_s}\mathbf{1}_{L_s}\mathbf{1}_{L_s}^\top XW_VW_O\right)\boldsymbol{u}_{\mathrm{up},j}\boldsymbol{v}_{\mathrm{up},j}^\top. \end{aligned} \tag{87}$$

Thus, $Z_{\mathrm{up}}(X)$ is effectively dominated by N_{up} rank-one matrices.

Since $Z_{\mathrm{up}} = \mathcal{O}(\sigma)$ during the focus stage, by Lemma E.3, we have

$$G(X) = \mathrm{SiLU}(Z_{\mathrm{up}}(X)) \approx \sum_{j=1}^{N_G}\lambda_{G,j}\boldsymbol{u}_{G,j}(X)\boldsymbol{v}_{G,j}^\top(X), \tag{88}$$

where $N_G = \frac{N_{\mathrm{up}}(N_{\mathrm{up}}+3)}{2}$, $\boldsymbol{u}_{G,j}(X), \boldsymbol{v}_{G,j}^\top(X)$ denote the singular vectors of $G(X)$ depending on X. Then $\boldsymbol{g}_i(X) \approx \sum_{j=1}^{N_G}\lambda_{G,j}v_{G,j,i}(X)\boldsymbol{u}_{G,j}(X)$. Thus, for $i \in [d_h]$,

$$\begin{aligned} \Phi^{\mathrm{mlp}}_i(X) &= \frac{\boldsymbol{g}_i(X)\boldsymbol{w}_{\mathrm{down},i}^\top}{\alpha^{\mathrm{mlp}}_i} \\ &\approx \sum_{j=1}^{N_G}\sum_{k=1}^{N_{\mathrm{down}}}\frac{\lambda_{G,j}\lambda_{\mathrm{down},k}v_{G,j,i}(X)u_{\mathrm{down},k,i}}{\alpha^{\mathrm{mlp}}_i}\boldsymbol{u}_{G,j}(X)\boldsymbol{v}_{\mathrm{down},k}^\top, \\ &= \frac{1}{\alpha^{\mathrm{mlp}}_i}\sum_{j=1}^{N_G}\sum_{k=1}^{N_{\mathrm{down}}}\lambda^{G,\mathrm{down}}_{j,k,i}(X)\boldsymbol{u}_{G,j}(X)\boldsymbol{v}_{\mathrm{down},k}^\top, \end{aligned} \tag{89}$$

where $\lambda^{G,\mathrm{down}}_{j,k,i}(X) = \lambda_{G,j}\lambda_{\mathrm{down},k}v_{G,j,i}(X)u_{\mathrm{down},k,i}$. That is, the effective dimensionality of $\mathrm{span}\{\Phi^{\mathrm{mlp}}_i(X)\}_{i=1}^{d_h}$ is at most $N_G N_{\mathrm{down}}$. Hence, up to the approximation error in Assumption 2.3,

$$r_{\mathrm{eff}}(M(t)) \lesssim 1 + N_V(t)N_O(t) + N_G(t)N_{\mathrm{down}}(t), \qquad t \in [0, t_1^*]. \tag{90}$$

□

E.7 Proof of Theorem 2.3

Proof. By the Hardy–Littlewood–Pólya theorem Hardy et al. (1934), $\boldsymbol{p}^\star$ majorizes $B\boldsymbol{p}^\star$. Since the Shannon entropy is Schur-concave,

$$\log r_{\mathrm{eff}}(B\boldsymbol{p}^\star) \geq \log r_{\mathrm{eff}}(\boldsymbol{p}^\star) > \log r_{\mathrm{eff}}(\boldsymbol{p}(t_1^*)). \tag{91}$$

Moreover, the concavity of Shannon entropy and Equation (21) give

$$\begin{aligned} \log r_{\mathrm{eff}}(\boldsymbol{p}(t_2^*)) &\geq (1-\alpha)\log r_{\mathrm{eff}}(\boldsymbol{p}(t_1^*)) + \alpha\log r_{\mathrm{eff}}(B\boldsymbol{p}^\star) \\ &> \log r_{\mathrm{eff}}(\boldsymbol{p}(t_1^*)). \end{aligned} \tag{92}$$

Therefore,

$$r_{\mathrm{eff}}(M(t_2^*)) > r_{\mathrm{eff}}(M(t_1^*)). \tag{93}$$

□

Remark E.2. The condition $r_{\mathrm{eff}}(\boldsymbol{p}^\star) > r_{\mathrm{eff}}(\boldsymbol{p}(t_1^*))$ is necessary for interpreting the increase in effective rank as being driven by the additional representational complexity required by the dataset. It suggests that the dominant focus-stage subspace does not adequately capture all relevant structure in the dataset, so subsequent output gradients can activate previously negligible basis directions.

Remark E.3. The condition in Equation (21) is necessary to ensure the stability of the spectral update and to relax straight convergence toward the target representation. Let $\mathcal{P}(\boldsymbol{p}^\star) = \{B\boldsymbol{p}^\star : B \in \mathcal{B}\}$. By the Birkhoff–von Neumann theorem Dufossé et al. (2018), $\mathcal{P}(\boldsymbol{p}^\star)$ is the permutahedron generated by $\boldsymbol{p}^\star$. Since $\boldsymbol{p}(t_1^*) \notin \mathcal{P}(\boldsymbol{p}^\star)$, so Equation (21) confines $\boldsymbol{p}(t_2^*)$ to the truncated pointed cone with apex $\boldsymbol{p}(t_1^*)$ and base $\mathcal{P}(\boldsymbol{p}^\star)$. It therefore restricts both the direction and the magnitude of the spectral update. Furthermore, it relaxes direct convergence to $\boldsymbol{p}^\star$ by allowing gradient-based movement toward a $\boldsymbol{p}$ which has higher effective rank than $\boldsymbol{p}^*$.

E.8 Proof of Proposition 2.5

Proof. Define

$$\begin{aligned} m_M &:= \min_{t\in[t_2^*,T]} \lambda_{\min}(M(t)), \\ B_{\nabla M} &:= \max_{i,j\in[N]} \sup_{t\in[t_2^*,T]} \|\nabla_\theta M_{ij}(t)\|_2. \end{aligned} \tag{94}$$

Because $M(t)$ is a normalized Gram matrix, $M_{ii}(t) = 1$ and $|M_{ij}(t)| \le 1$. Hence $\mathrm{Tr}(M(t)) = N$ and $\|M(t)\|_2 \le \mathrm{Tr}(M(t)) = N$. The positive definiteness in Assumption 2.1 and continuity on the compact interval $[t_2^*, T]$ give $m_M > 0$. Moreover, the continuous differentiability of $M(\theta)$ and compactness of the parameter trajectory give $B_{\nabla M} < \infty$. Let

$$B_{\log M} := \max\bigl\{|\log m_M|, \log N\bigr\}. \tag{95}$$

The spectrum of $M(t)$ is contained in $[m_M, N]$, and hence

$$\|\log M(t)\|_2 \le B_{\log M}, \qquad |(\log M(t))_{ij}| \le B_{\log M}. \tag{96}$$

Proposition 2.7 gives

$$\dot{M}_{ij}(t) = -\left\langle \nabla_\theta M_{ij}(t), \nabla_\theta \mathcal{L}(\theta(t)) \right\rangle. \tag{97}$$

Thus, Definition 2.2 and the Cauchy–Schwarz inequality imply

$$|\dot{M}_{ij}(t)| \le B_{\nabla M} \varepsilon_{t_2^*}. \tag{98}$$

Proposition 2.6 gives

$$\frac{\mathrm{d}}{\mathrm{d}t} \log r_{\text{eff}}(M(t)) = -\frac{2}{N} \sum_{1\le i<j\le N} (\log M(t))_{ij} \dot{M}_{ij}(t). \tag{99}$$

Therefore,

$$\begin{aligned} \left|\frac{\mathrm{d}}{\mathrm{d}t} \log r_{\text{eff}}(M(t))\right| &\le \frac{2}{N} \left(\sum_{1\le i<j\le N} (\log M(t))_{ij}^2\right)^{1/2} \left(\sum_{1\le i<j\le N} \dot{M}_{ij}(t)^2\right)^{1/2} \\ &\le \sqrt{N-1}\, B_{\log M} B_{\nabla M} \varepsilon_{t_2^*}, \end{aligned} \tag{100}$$

where we used

$$\sum_{1\le i<j\le N} (\log M(t))_{ij}^2 \le \frac{N}{2} B_{\log M}^2, \qquad \sum_{1\le i<j\le N} \dot{M}_{ij}(t)^2 \le \frac{N(N-1)}{2} B_{\nabla M}^2 \varepsilon_{t_2^*}^2. \tag{101}$$

Multiplying both sides by $r_{\text{eff}}(M(t))$ and using $r_{\text{eff}}(M(t)) \le N$ yields

$$\left|\frac{\mathrm{d}}{\mathrm{d}t} r_{\text{eff}}(M(t))\right| \le N\sqrt{N-1}\, B_{\log M} B_{\nabla M} \varepsilon_{t_2^*}. \tag{102}$$

Substituting the definition of $B_{\log M}$ proves Equation (23) by taking

$$C_{[t_2^*,T]} = N\sqrt{N-1}\, B_{\nabla M} \max\bigl\{|\log m_M|, \log N\bigr\}. \tag{103}$$

□

E.9 Proof of Proposition 2.6

Proof. Since $M(t)$ is symmetric positive definite, its decomposition is $M(t) = Q(t)\Lambda(t)Q(t)^\top$, where $\Lambda(t) = \operatorname{diag}(\lambda_1(t), \dots, \lambda_N(t))$, its matrix logarithm is $\log M(t) = Q(t)(\log \Lambda(t))Q(t)^\top$, where $\log \Lambda(t) = \operatorname{diag}(\log \lambda_1(t), \dots, \log \lambda_N(t))$. Since $\sum_{i=1}^N \lambda_i(t) = \operatorname{Tr} M(t) = N$, we have

$$\begin{aligned} \log r_{\text{eff}}(M(t)) &= -\sum_{i=1}^N \frac{\lambda_i(t)}{N} \log \frac{\lambda_i(t)}{N} \\ &= \log N - \frac{1}{N} \sum_{i=1}^N \lambda_i(t) \log \lambda_i(t) \\ &= \log N - \frac{1}{N} \operatorname{Tr}\left(M(t) \log M(t)\right). \end{aligned} \tag{104}$$

Define the scalar function $f : (0, \infty) \to \mathbb{R}$ by $f(x) = x \log x$. Its derivative is

$$f'(x) = 1 + \log x. \tag{105}$$

For a differentiable symmetric positive definite matrix path $M(t)$, the differential formula for spectral trace functions gives

$$\frac{\mathrm{d}}{\mathrm{d}t} \operatorname{Tr}(f(M(t))) = \operatorname{Tr}\left(f'(M(t))\dot{M}(t)\right). \tag{106}$$

This identity follows by diagonalizing $M(t) = Q(t)\Lambda(t)Q(t)^\top$: the trace is invariant under the orthogonal change of basis, and the contributions from the derivatives of $Q(t)$ cancel in the trace. Substituting $f'(M(t)) = I + \log M(t)$ and $f(M(t)) = M(t) \log M(t)$ yields

$$\frac{\mathrm{d}}{\mathrm{d}t} \operatorname{Tr}\left(M(t) \log M(t)\right) = \operatorname{Tr}\left((I + \log M(t))\dot{M}(t)\right). \tag{107}$$

Therefore,

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} \log r_{\text{eff}}(M(t)) &= -\frac{1}{N} \operatorname{Tr}\left((I + \log M(t))\dot{M}(t)\right) \\ &= -\frac{1}{N} \operatorname{Tr} \dot{M}(t) - \frac{1}{N} \operatorname{Tr}\left((\log M(t))\dot{M}(t)\right). \end{aligned} \tag{108}$$

Since $\operatorname{Tr} M(t) \equiv N \Rightarrow \operatorname{Tr} \dot{M}(t) = 0$, we have

$$\frac{\mathrm{d}}{\mathrm{d}t} \log r_{\text{eff}}(M(t)) = -\frac{1}{N} \operatorname{Tr}\left((\log M(t))\dot{M}(t)\right). \tag{109}$$

Since $M(t)$ is symmetric, both $\log M(t)$ and $\dot{M}(t)$ are symmetric. Moreover, $M_{ii}(t) = 1$ implies $\dot{M}_{ii}(t) = 0$. Therefore,

$$\begin{aligned} \operatorname{Tr}\left((\log M(t))\dot{M}(t)\right) &= \sum_{i=1}^N \sum_{j=1}^N (\log M(t))_{ij} \dot{M}_{ji}(t) \\ &= 2 \sum_{1 \le i < j \le N} (\log M(t))_{ij} \dot{M}_{ij}(t). \end{aligned} \tag{110}$$

Consequently,

$$\frac{\mathrm{d}}{\mathrm{d}t} \log r_{\text{eff}}(M(t)) = -\frac{2}{N} \sum_{1 \le i < j \le N} (\log M(t))_{ij} \dot{M}_{ij}(t). \tag{111}$$

Finally, since $r_{\text{eff}}(M(t)) > 0$,

$$\frac{\mathrm{d}}{\mathrm{d}t} r_{\text{eff}}(M(t)) = r_{\text{eff}}(M(t)) \frac{\mathrm{d}}{\mathrm{d}t} \log r_{\text{eff}}(M(t)). \tag{112}$$

Hence,

$$\sum_{1 \le i < j \le N} (\log M(t))_{ij} \dot{M}_{ij}(t) = 0 \quad \Longleftrightarrow \quad \frac{\mathrm{d}}{\mathrm{d}t} r_{\text{eff}}(M(t)) = 0. \tag{113}$$

□

E.10 Proof of Proposition 2.7

Proof. Write $\theta = (\theta_1, \ldots, \theta_p)$ for all trainable parameters. Since $M_{ij}(t) = M_{ij}(\theta(t))$, the chain rule gives

$$\begin{aligned} \dot{M}_{ij}(t) &= \sum_{r=1}^{p} \frac{\partial M_{ij}(\theta(t))}{\partial \theta_r} \dot{\theta}_r(t) \\ &= \left\langle \nabla_\theta M_{ij}(t), \dot{\theta}(t) \right\rangle. \end{aligned} \tag{114}$$

Substituting the gradient flow $\dot{\theta}(t) = -\nabla_\theta \mathcal{L}(\theta(t))$ yields

$$\dot{M}_{ij}(t) = -\left\langle \nabla_\theta M_{ij}(t), \nabla_\theta \mathcal{L}(\theta(t)) \right\rangle. \tag{115}$$

To make the empirical-data dependence explicit, define $\boldsymbol{z}_k(t) := W_{\mathrm{lm}}(t)\mathcal{U}(X_k;\theta(t))^\top_{L_s,:}$ and $J_k(t) := \partial \boldsymbol{z}_k(t)/\partial\theta$. By Equation (70),

$$\nabla_\theta \mathcal{L}(\theta(t)) = \frac{1}{\mathcal{D}} \sum_{k=1}^{\mathcal{D}} J_k(t)^\top \mathbb{A}_k(t). \tag{116}$$

Therefore, equivalently,

$$\dot{M}_{ij}(t) = -\frac{1}{\mathcal{D}} \sum_{k=1}^{\mathcal{D}} \mathbb{A}_k(t)^\top J_k(t) \nabla_\theta M_{ij}(t). \tag{117}$$

Finally, the gradient-flow identity above directly gives

$$\langle \nabla_\theta M_{ij}(t), \nabla_\theta \mathcal{L}(\theta(t)) \rangle = 0 \quad \Longleftrightarrow \quad \dot{M}_{ij}(t) = 0, \tag{118}$$

which proves the stated equivalence. □

E.11 Proof of Theorem 2.4

Proof. By Equation (13) and the unified indexing convention, the non-residual basis functions satisfy

$$(\boldsymbol{b}_i(X;t), \boldsymbol{v}_i(t)) = \begin{cases} (\boldsymbol{a}_{i-1}(X;t), \boldsymbol{w}_{O,i-1}(t)), & i \in \{2, \ldots, d_{\mathrm{attn}}+1\}, \\ (\boldsymbol{g}_{i-d_{\mathrm{attn}}-1}(X;t), \boldsymbol{w}_{\mathrm{down},i-d_{\mathrm{attn}}-1}(t)), & i \in \{d_{\mathrm{attn}}+2, \ldots, N\}. \end{cases} \tag{119}$$

Thus, although both factors may evolve with t, every non-residual basis function remains an outer product of two vectors and hence has rank at most one.

For any $i, j \in [N] \setminus \{1\}$, the Frobenius identity $\langle \boldsymbol{u}\boldsymbol{v}^\top, \boldsymbol{x}\boldsymbol{y}^\top \rangle_F = (\boldsymbol{u}^\top \boldsymbol{x})(\boldsymbol{v}^\top \boldsymbol{y})$ gives

$$\begin{aligned} M_{ij}(t) &= \frac{1}{\mathcal{D}} \sum_{k=1}^{\mathcal{D}} \langle \Phi_i(X_k;t), \Phi_j(X_k;t) \rangle_F \\ &= \frac{1}{\alpha_i(t)\alpha_j(t)} \left(\frac{1}{\mathcal{D}} \sum_{k=1}^{\mathcal{D}} \boldsymbol{b}_i(X_k;t)^\top \boldsymbol{b}_j(X_k;t) \right) \left(\boldsymbol{v}_i(t)^\top \boldsymbol{v}_j(t) \right) \\ &= \frac{b_{ij}(t) v_{ij}(t)}{\alpha_i(t)\alpha_j(t)}, \end{aligned} \tag{120}$$

where the output-side inner product can be taken outside the sum because $\boldsymbol{v}_i(t)$ and $\boldsymbol{v}_j(t)$ do not depend on the sample index.

Setting $j = i$ and using the normalization $M_{ii}(t) = 1$ yields

$$\alpha_i^2(t) = b_{ii}(t) v_{ii}(t). \tag{121}$$

Moreover,

$$b_{ii}(t) = \frac{1}{\mathcal{D}} \sum_{k=1}^{\mathcal{D}} \|\boldsymbol{b}_i(X_k;t)\|_2^2, \qquad v_{ii}(t) = \|\boldsymbol{v}_i(t)\|_2^2. \tag{122}$$

Since Assumption 2.1 ensures $\alpha_i(t) > 0$, the preceding identity implies $b_{ii}(t) > 0$, $v_{ii}(t) > 0$, and

$$\alpha_i(t) = \sqrt{b_{ii}(t)v_{ii}(t)}. \tag{123}$$

Therefore,

$$\alpha_i(t)\alpha_j(t) = \sqrt{b_{ii}(t)b_{jj}(t)}\sqrt{v_{ii}(t)v_{jj}(t)}. \tag{124}$$

Substituting this expression into the formula for $M_{ij}(t)$ gives

$$M_{ij}(t) = \frac{b_{ij}(t)v_{ij}(t)}{\sqrt{b_{ii}(t)b_{jj}(t)}\sqrt{v_{ii}(t)v_{jj}(t)}}, \tag{125}$$

which proves Equation (28).

Finally, suppose $M_{ij}(t) \neq 0$. Since the denominator in the preceding identity is positive, $b_{ij}(t) \neq 0$ and $v_{ij}(t) \neq 0$. Taking the logarithm of the absolute value of that identity and differentiating yields

$$\frac{\dot{M}_{ij}(t)}{M_{ij}(t)} = \frac{\dot{b}_{ij}(t)}{b_{ij}(t)} + \frac{\dot{v}_{ij}(t)}{v_{ij}(t)} - \frac{1}{2}\left(\frac{\dot{b}_{ii}(t)}{b_{ii}(t)} + \frac{\dot{b}_{jj}(t)}{b_{jj}(t)} + \frac{\dot{v}_{ii}(t)}{v_{ii}(t)} + \frac{\dot{v}_{jj}(t)}{v_{jj}(t)}\right). \tag{126}$$

Because $M_{ij}(t) \neq 0$, the left-hand side vanishes if and only if $\dot{M}_{ij}(t) = 0$. Hence,

$$\frac{\dot{b}_{ij}(t)}{b_{ij}(t)} + \frac{\dot{v}_{ij}(t)}{v_{ij}(t)} = \frac{1}{2}\left(\frac{\dot{b}_{ii}(t)}{b_{ii}(t)} + \frac{\dot{b}_{jj}(t)}{b_{jj}(t)} + \frac{\dot{v}_{ii}(t)}{v_{ii}(t)} + \frac{\dot{v}_{jj}(t)}{v_{jj}(t)}\right) \quad \Longleftrightarrow \quad \dot{M}_{ij}(t) = 0, \tag{127}$$

which proves Equation (29). □