Training Dynamics of In-Context Learning in Linear Attention

Yedi Zhang [1] **Aaditya K. Singh** [1] **Peter E. Latham** [1 *] **Andrew Saxe** [1 2 *]

Abstract

While attention-based models have demonstrated the remarkable ability of in-context learning (ICL), the theoretical understanding of how these models acquired this ability through gradient descent training is still preliminary. Towards answering this question, we study the gradient descent dynamics of multi-head linear self-attention trained for in-context linear regression. We examine two parametrizations of linear self-attention: one with the key and query weights merged as a single matrix (common in theoretical studies), and one with separate key and query matrices (closer to practical settings). For the merged parametrization, we show that the training dynamics has two fixed points and the loss trajectory exhibits a single, abrupt drop. We derive an analytical time-course solution for a certain class of datasets and initialization. For the separate parametrization, we show that the training dynamics has exponentially many fixed points and the loss exhibits saddle-to-saddle dynamics, which we reduce to scalar ordinary differential equations. During training, the model implements principal component regression in context with the number of principal components increasing over training time. Overall, we provide a theoretical description of how ICL abilities evolve during gradient descent training of linear attention, revealing abrupt acquisition or progressive improvements depending on how the key and query are parametrized.

1. Introduction

Self-attention-based models, such as transformers (Vaswani et al., 2017), exhibit a remarkable ability known as in-context learning (Brown et al., 2020). That is, these models can solve unseen tasks based on exemplars in the context of an input prompt. In-context learning (ICL) is critical to the flexibility of large language models, allowing them to solve tasks not explicitly included in their training data. However, it remains unclear how architectures like self-attention acquire this ability through gradient descent training.

Seminal work by Olsson et al. (2022) identified an intriguing trait in the training dynamics of ICL: the ICL ability often emerges abruptly, coinciding with an abrupt drop in loss during training. This abrupt learning phase can reflect the formation of an induction head in the ICL setting (Olsson et al., 2022; Reddy, 2024; Singh et al., 2024; Edelman et al., 2024), and can also occur more broadly in transformer training dynamics (Nanda et al., 2023; Chen et al., 2024a; Hoffmann et al., 2024; Gopalani et al., 2024). Furthermore, Singh et al. (2023) found that ICL may often be a transient ability that the transformers acquire and then lose over the course of long training time, a phenomenon that has since been reproduced in many settings (He et al., 2024; Anand et al., 2025; Chan et al., 2025; Nguyen & Reddy, 2025; Park et al., 2025; Singh et al., 2025). These findings underscore the importance of understanding not only the ICL ability in trained models, but its full training dynamics.

This work aims to provide a theoretical description of how the ICL ability evolves in gradient descent training. To do so, we consider the increasingly common setup of linear attention[1] (Von Oswald et al., 2023) trained on an in-context linear regression task (Garg et al., 2022). The in-context linear regression task, in which the model needs to perform linear regression on the data in context, is a canonical instantiation of ICL (Garg et al., 2022; Akyürek et al., 2023; Von Oswald et al., 2023; Ahn et al., 2023; Bai et al., 2023). The linear attention model, which has been used in many prior studies (Schlag et al., 2021; Von Oswald et al., 2023; Ahn et al., 2023; Zhang et al., 2024a; Wu et al., 2024; Fu et al., 2024; Mahankali et al., 2024; Duraisamy, 2024; Li et al., 2024; Yau et al., 2024; Lu et al., 2025; Frei & Vardi, 2025), reproduces key optimization properties of practical transformers (Ahn et al., 2024) and is more amenable to theoretical analysis. Importantly, despite its name, linear attention is a nonlinear model, as it removes the softmax operation but is still a nonlinear function of the input.

[*]Equal contribution [1]Gatsby Computational Neuroscience Unit, University College London [2]Sainsbury Wellcome Centre, University College London. Correspondence to: Yedi Zhang <yedi@gatsby.ucl.ac.uk>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

[1]We refer to linear self-attention as linear attention in this paper.

We study two common parametrizations of multi-head linear attention: (i) ATTN_M, linear attention where the key and query matrices in each head are merged into a single matrix, a reparametrization procedure widely used in theoretical studies on transformers (Ahn et al., 2023; Tian et al., 2023; Ataee Tarzanagh et al., 2023; Zhang et al., 2024a;b; Chen et al., 2024b; Wu et al., 2024; Kim & Suzuki, 2024; Huang et al., 2024b; Wang et al., 2024b; Ildiz et al., 2024; Ren et al., 2024; Tarzanagh et al., 2024; Yau et al., 2024; Julistiono et al., 2024; Anwar et al., 2024; Vasudeva et al., 2025; Lu et al., 2025; Chen & Li, 2025; Huang et al., 2025a); (ii) ATTN_S, linear attention with separate key and query matrices, which is closer to the implementation of attention in real-world transformers (Vaswani et al., 2017). We specify the fixed points in the loss landscapes, as well as how gradient descent training dynamics traverses the landscape. Our findings are summarized as follows.

- We find two fixed points in the training dynamics of ATTN_M, and exponentially many fixed points in that of ATTN_S.
- We show a single, abrupt loss drop in training ATTN_M from small initialization and derive an analytical time-course solution when the input token covariance is white. We show saddle-to-saddle training dynamics in training ATTN_S from small initialization and reduce the high-dimensional training dynamics to scalar ordinary differential equations through an ansatz. We demonstrate the rank of the separate key and query weights affects the dynamics by shortening the duration of certain plateaus.
- We identify the in-context algorithm of the converged and early stopped models. When ATTN_M and ATTN_S are trained to convergence, they approximately implement least squares linear regression in context. When the training of ATTN_S early stops during the $(m+1)$-th loss plateau, it approximately implements principal component regression in context with the first m principal components.
- As a tool for our analysis, we show that when trained on in-context linear regression tasks, ATTN_M is equivalent to a two-layer fully-connected linear network with a cubic feature map as input, and ATTN_S is equivalent to a sum of three-layer convolutional linear networks with the same cubic feature map as input.
- We empirically demonstrate that the single and multiple loss drops also occur in softmax ATTN_M and ATTN_S, respectively.

Comparing the two models, we find that the ICL ability evolves differently in them: ATTN_M acquires the in-context linear regression ability through one abrupt loss drop, while ATTN_S acquires this ability by *progressively improving* on in-context principal component regression. This makes a theoretical case for the progressive improvements of ICL in gradient descent training. Our results also reveal how parametrization, such as merged versus separate key and query and the rank of the separate key and query weights, influences the loss landscape and training dynamics. This motivates future research to take the parametrization factor into account when studying the landscape and dynamics of attention models.

2. Preliminaries

Notation. Non-bold small and capital symbols are scalars. Bold small symbols are column vectors. Bold capital symbols are matrices. $\|\cdot\|$ denotes the ℓ^2 norm of a vector or the Frobenius norm of a matrix. $\mathtt{vec}(\cdot)$ represents flattening a matrix to a column vector by stacking its columns. For example, $\mathtt{vec}\begin{bmatrix}1 & 3\\ 2 & 4\end{bmatrix} = \begin{bmatrix}1 & 2 & 3 & 4\end{bmatrix}^\top$. We use $i = 1, \cdots, H$ to denote the index of an attention head, $\mu = 1, \cdots, P$ to denote the index of a training sample, and $n = 1, \cdots, N$ to denote the index of a token in a sample.

2.1. In-Context Linear Regression Task

We study a standard ICL task of predicting the next token. The input is a sequence $\{\boldsymbol{x}_1, y_1, \boldsymbol{x}_2, y_2, \cdots, \boldsymbol{x}_N, y_N, \boldsymbol{x}_q\}$ and the desired output is y_q. We refer to $\boldsymbol{x}_q$ as the query token, $\{\boldsymbol{x}_1, y_1, \boldsymbol{x}_2, y_2, \cdots, \boldsymbol{x}_N, y_N\}$ as the context, and N as the context length. By convention (Ahn et al., 2023; Zhang et al., 2024a;b; Chen et al., 2024b; Huang et al., 2024b), the input sequence is presented to the model as a matrix $\boldsymbol{X}$, defined as

$$\boldsymbol{X} = \begin{bmatrix}\boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_N & \boldsymbol{x}_q\\ y_1 & y_2 & \cdots & y_N & 0\end{bmatrix} \in \mathbb{R}^{(D+1)\times(N+1)}, \quad (1)$$

where $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N, \boldsymbol{x}_q \in \mathbb{R}^D$ and $y_1, \cdots, y_N \in \mathbb{R}$.

We are given a training dataset $\{\boldsymbol{X}_\mu, y_{\mu,q}\}_{\mu=1}^P$ consisting of P samples. All $\boldsymbol{x}$ tokens are independently sampled from a D-dimensional zero-mean normal distribution with covariance $\boldsymbol{\Lambda}$,

$$\boldsymbol{x}_{\mu,n}, \boldsymbol{x}_{\mu,q} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Lambda}), n = 1, \cdots, N, \mu = 1, \cdots, P. \quad (2)$$

We consider the in-context linear regression task, where the y_n in context and the target output y_q are generated as a linear map of the corresponding $\boldsymbol{x}_n$ and $\boldsymbol{x}_q$ (Garg et al., 2022). For each sequence $\boldsymbol{X}_\mu$, we independently sample a task vector $\boldsymbol{w}_\mu$ from a D-dimensional standard normal distribution, $\boldsymbol{w}_\mu \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, and generate $y_{\mu,n} = \boldsymbol{w}_\mu^\top \boldsymbol{x}_{\mu,n}, y_{\mu,q} = \boldsymbol{w}_\mu^\top \boldsymbol{x}_{\mu,q}, n = 1, \cdots, N$, $\mu = 1, \cdots, P$. Note that the task vector $\boldsymbol{w}_\mu$ is fixed for all tokens in one sample sequence but varies across different samples, and is independent of the tokens $\boldsymbol{x}_{\mu,1}, \cdots, \boldsymbol{x}_{\mu,N}, \boldsymbol{x}_{\mu,q}$.

2.2. Multi-Head Self-Attention

A standard multi-head softmax self-attention layer (Vaswani et al., 2017) takes the matrix $\boldsymbol{X}$ as input and returns a matrix

of the same size,

$$\mathsf{ATTN}(\boldsymbol{X}) = \boldsymbol{X} + \sum_{i=1}^{H} \boldsymbol{W}_i^V \boldsymbol{X} \mathsf{smax}\left(\frac{\boldsymbol{X}^\top \boldsymbol{W}_i^{K\top} \boldsymbol{W}_i^Q \boldsymbol{X}}{\rho}\right)$$

where H is the number of heads, ρ is a scaling factor, and $\boldsymbol{W}_i^V, \boldsymbol{W}_i^K, \boldsymbol{W}_i^Q$ are the trainable value, key, and query matrices in the i-th head. The prediction for y_q is the bottom right entry of the output matrix:

$$\hat{y}_q = \mathsf{ATTN}(\boldsymbol{X})_{D+1,N+1}. \tag{3}$$

In this work, we consider multi-head linear self-attention, where we remove the softmax operation and take $\rho = N$. Specifically, we study two common parametrizations of linear attention: (i) linear attention with merged key and query introduced in Section 2.3 and analyzed in Section 3; (ii) linear attention with separate key and query introduced in Section 2.4 and analyzed in Sections 4 and 5.

2.3. Linear Attention with Merged Key and Query

The multi-head linear attention ATTN_M with the key and query matrices in each head merged as a single matrix $\boldsymbol{W}_i^{K\top} \boldsymbol{W}_i^Q = \boldsymbol{W}_i^{KQ}$ computes

$$\mathsf{ATTN}_\mathrm{M}(\boldsymbol{X}) = \boldsymbol{X} + \sum_{i=1}^{H} \frac{1}{N} \boldsymbol{W}_i^V \boldsymbol{X} \boldsymbol{X}^\top \boldsymbol{W}_i^{KQ} \boldsymbol{X},$$

where the terms can be written in block form,

$$\boldsymbol{X}\boldsymbol{X}^\top = \begin{bmatrix} \left(\boldsymbol{x}_q \boldsymbol{x}_q^\top + \sum_{n=1}^N \boldsymbol{x}_n \boldsymbol{x}_n^\top\right) & \sum_{n=1}^N \boldsymbol{x}_n y_n \\ \sum_{n=1}^N y_n \boldsymbol{x}_n^\top & \sum_{n=1}^N y_n^2 \end{bmatrix},$$

and

$$\boldsymbol{W}_i^V = \begin{bmatrix} * & * \\ \boldsymbol{v}_i^\top & v_i \end{bmatrix}, \boldsymbol{W}_i^{KQ} = \begin{bmatrix} \boldsymbol{U}_i & * \\ \boldsymbol{u}_i^\top & * \end{bmatrix}.$$

The blocks have dimensionalities $\boldsymbol{v}_i, \boldsymbol{u}_i \in \mathbb{R}^D, v_i \in \mathbb{R}, \boldsymbol{U}_i \in \mathbb{R}^{D\times D}$. The $*$ blocks denote entries that do not contribute to the computation of $\mathsf{ATTN}(\boldsymbol{X})_{D+1,N+1}$. Following Ahn et al. (2023); Zhang et al. (2024a); Kim & Suzuki (2024); Huang et al. (2024b), we initialize $\boldsymbol{v}_i, \boldsymbol{u}_i = \boldsymbol{0}$ as they are not required for this model to achieve global minimum loss on the in-context linear regression task. When $\boldsymbol{v}_i$ and $\boldsymbol{u}_i$ are initialized to zero, they will remain zero throughout training (see Appendix D.1). With the reduction $\boldsymbol{v}_i, \boldsymbol{u}_i = \boldsymbol{0}$, the prediction for y_q, which is the bottom right entry of $\mathsf{ATTN}_\mathrm{M}(\boldsymbol{X})$, is

$$\mathsf{ATTN}_\mathrm{M}(\boldsymbol{X})_{D+1,N+1} = \sum_{i=1}^{H} v_i \boldsymbol{\beta}^\top \boldsymbol{U}_i \boldsymbol{x}_q, \tag{M}$$

where $\boldsymbol{\beta}$ is the correlation between $\boldsymbol{x}_n$ and y_n in context,

$$\boldsymbol{\beta} \equiv \frac{1}{N} \sum_{n=1}^{N} y_n \boldsymbol{x}_n. \tag{4}$$

2.4. Linear Attention with Separate Key and Query

In multi-head attention with separate key and query, we follow the standard practice (Vaswani et al., 2017) of using low-rank key and query matrices where the rank $R \le D$ and $RH \ge D$. In practice, usually $RH = D$. The multi-head linear attention ATTN_S with separate rank-R key and query matrices computes

$$\mathsf{ATTN}_\mathrm{S}(\boldsymbol{X}) = \boldsymbol{X} + \sum_{i=1}^{H} \frac{1}{N} \boldsymbol{W}_i^V \boldsymbol{X} \boldsymbol{X}^\top \boldsymbol{W}_i^{K\top} \boldsymbol{W}_i^Q \boldsymbol{X}.$$

We can write the value, key, and query weights in block form,

$$\boldsymbol{W}_i^V = \begin{bmatrix} * & * \\ \boldsymbol{v}_i^\top & v_i \end{bmatrix}, \boldsymbol{W}_i^K = \begin{bmatrix} \boldsymbol{k}_{i,1}^\top & k_{i,1} \\ \vdots & \vdots \\ \boldsymbol{k}_{i,R}^\top & k_{i,R} \end{bmatrix}, \boldsymbol{W}_i^Q = \begin{bmatrix} \boldsymbol{q}_{i,1}^\top & * \\ \vdots & \vdots \\ \boldsymbol{q}_{i,R}^\top & * \end{bmatrix}.$$

The blocks have dimensionalities $v_i, k_{i,r} \in \mathbb{R}$ and $\boldsymbol{v}_i, \boldsymbol{k}_{i,r}, \boldsymbol{q}_{i,r} \in \mathbb{R}^D$ $(r = 1, \cdots, R)$. Similarly to the case with merged key and query, we initialize $\boldsymbol{v}_i = \boldsymbol{0}, k_{i,r} = 0$; they will remain zero throughout training (see Appendix F.1). With $\boldsymbol{v}_i = \boldsymbol{0}$ and $k_{i,r} = 0$, the multi-head linear attention with separate rank-one key and query matrices computes

$$\mathsf{ATTN}_\mathrm{S}(\boldsymbol{X})_{D+1,N+1} = \sum_{i=1}^{H} \sum_{r=1}^{R} v_i \boldsymbol{\beta}^\top \boldsymbol{k}_{i,r} \boldsymbol{q}_{i,r}^\top \boldsymbol{x}_q, \tag{S}$$

where $\boldsymbol{\beta}$ is the input-output correlation in context defined in Equation (4). The expression of Equation (S) already reveals interesting insight. It implies that linear attention with H heads and rank-R key and query differs from linear attention with RH heads and rank-one key and query only in the sharing of certain value weights.

2.5. Gradient Flow Training Dynamics

We train the linear attention model using gradient descent on squared loss of the query token[2], that is $\mathcal{L} = \mathbb{E}(y_q - \hat{y}_q)^2$. We analyze the gradient flow dynamics on the loss, given by

$$\tau \frac{\mathrm{d}\boldsymbol{W}}{\mathrm{d}t} = -\frac{1}{2} \frac{\partial \mathcal{L}}{\partial \boldsymbol{W}} = \mathbb{E}\left[(y_q - \hat{y}_q) \frac{\partial \hat{y}_q}{\partial \boldsymbol{W}}\right], \tag{5}$$

where τ is the time constant. The gradient flow dynamics captures the behavior of gradient descent in the limit of a small learning rate.

3. Linear Attention with Merged Key and Query

We first study multi-head linear attention with the key and query matrices merged as a single matrix, as described by Equation (M).

[2]We can also handle next token prediction loss (Appendix A.3).

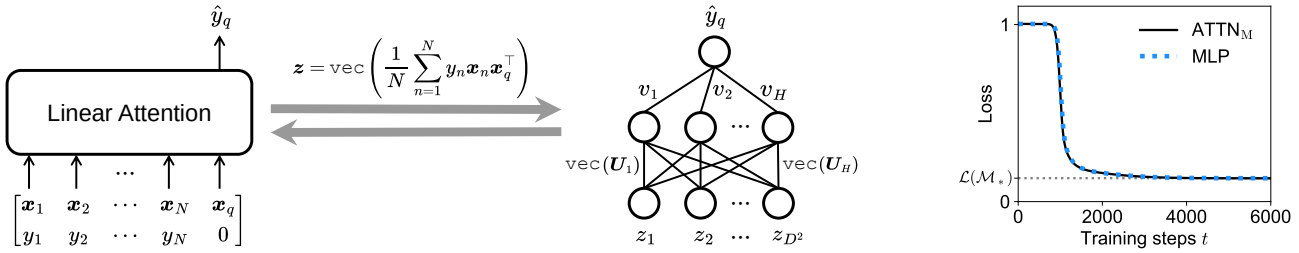


Figure 1. Multi-head linear attention with merged key and query, $\mathsf{ATTN}_\mathrm{M}(\boldsymbol{X})_{D+1,N+1}$, is equivalent to a two-layer fully-connected linear network with cubic feature input, $\mathsf{MLP}(\boldsymbol{z})$. *Left*: Schematic of the equivalence. *Right*: Loss trajectories of linear attention and the fully-connected linear network match well. The two models are trained with the same data and initialization. Both exhibit the characteristic abrupt loss drop documented by prior work on the ICL dynamics in linear (Von Oswald et al., 2023) and softmax attention (Singh et al., 2024). Here $D = 4, N = 31, H = 8$.

3.1. Connection to A Fully-Connected Linear Network

The H-head linear attention with input sequence $\boldsymbol{X}$ defined in Equation (M) can be viewed as a two-layer width-H fully-connected linear network with a cubic feature $\boldsymbol{z}(\boldsymbol{X})$ as input,

$$\begin{aligned}\mathsf{ATTN}_\mathrm{M}(\boldsymbol{X})_{D+1,N+1} &= \sum_{i=1}^{H} v_i \boldsymbol{\beta}^\top \boldsymbol{U}_i \boldsymbol{x}_q \\ &= \sum_{i=1}^{H} v_i \mathrm{vec}(\boldsymbol{U}_i)^\top \mathrm{vec}\left(\boldsymbol{\beta}\boldsymbol{x}_q^\top\right) \\ &= \boldsymbol{w}_2^\top \boldsymbol{W}_1 \boldsymbol{z} = \mathsf{MLP}(\boldsymbol{z}), \end{aligned} \tag{6}$$

where

$$\boldsymbol{w}_2 = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_H \end{bmatrix}, \boldsymbol{W}_1 = \begin{bmatrix} \mathrm{vec}(\boldsymbol{U}_1)^\top \\ \mathrm{vec}(\boldsymbol{U}_2)^\top \\ \vdots \\ \mathrm{vec}(\boldsymbol{U}_H)^\top \end{bmatrix}, \boldsymbol{z}(\boldsymbol{X}) = \mathrm{vec}\left(\boldsymbol{\beta}\boldsymbol{x}_q^\top\right). \tag{7}$$

The feature $\boldsymbol{z} \in \mathbb{R}^{D^2}$, whose entries are cubic functions of the entries in the original sequence $\boldsymbol{X}$, is the input to the equivalent two-layer fully-connected linear network. The stacked value weights correspond to the second-layer weights $\boldsymbol{w}_2 \in \mathbb{R}^H$ of the fully-connected linear network. The stacked merged key-query weights correspond to the first-layer weights $\boldsymbol{W}_1 \in \mathbb{R}^{H \times D^2}$ of the fully-connected linear network. A schematic of this equivalence is given in Figure 1.

3.2. Loss Landscape: Two Fixed Points

The gradient flow training dynamics of the linear attention or the equivalent two-layer fully-connected linear network given in Equation (6) is

$$\tau \dot{\boldsymbol{W}}_1 = \boldsymbol{w}_2 \left(\mathbb{E}\left(y_q \boldsymbol{z}^\top\right) - \boldsymbol{w}_2^\top \boldsymbol{W}_1 \mathbb{E}\left(\boldsymbol{z}\boldsymbol{z}^\top\right)\right), \tag{8a}$$

$$\tau \dot{\boldsymbol{w}}_2 = \boldsymbol{W}_1 \left(\mathbb{E}\left(y_q \boldsymbol{z}^\top\right) - \boldsymbol{w}_2^\top \boldsymbol{W}_1 \mathbb{E}\left(\boldsymbol{z}\boldsymbol{z}^\top\right)\right)^\top. \tag{8b}$$

There are two manifolds of fixed points in this dynamical system: one is the unstable fixed point at zero, denoted $\mathcal{M}_0$, and the other is a manifold of stable fixed points at the global minimum, denoted $\mathcal{M}_*$,

$$\mathcal{M}_0 = \{\boldsymbol{w}_2 = \boldsymbol{0}, \boldsymbol{W}_1 = \boldsymbol{0}\} \tag{9a}$$

$$\mathcal{M}_* = \left\{\boldsymbol{w}_2, \boldsymbol{W}_1 \middle| \boldsymbol{w}_2^\top \boldsymbol{W}_1 = \mathbb{E}\left(y_q \boldsymbol{z}^\top\right) \mathbb{E}\left(\boldsymbol{z}\boldsymbol{z}^\top\right)^{-1}\right\} \tag{9b}$$

3.3. Training Dynamics: An Abrupt Drop in the Loss

We have shown the linear attention defined in Equation (M) is equivalent to a fully-connected linear network with cubic feature input. Since this equivalence holds at the level of the computation of the model, the equivalence applies to the training dynamics with any initialization and optimizer. Here we discuss the training dynamics from small initialization, commonly referred to as the rich learning regime (Woodworth et al., 2020).

With small initialization, the network is initially near the unstable fixed point, $\mathcal{M}_0$, at zero. As training progresses, the network escapes from the unstable fixed point, and subsequently converges to a stable fixed point on the global minimum manifold, $\mathcal{M}_*$. The time it takes to escape from the unstable fixed point is approximately $\frac{\tau}{\|\boldsymbol{\Lambda}^2\|} \ln \frac{1}{w_\mathrm{init}}$, where the initialization scale w_init is the initial ℓ^2 norm of a layer (see Appendix D.6.1). Because the time to escape from the unstable fixed point starting from small initialization is long, the loss exhibits an initial plateau followed by an abrupt drop, as validated by simulations in Figure 1. In particular, when the input token covariance is white $\boldsymbol{\Lambda} = \boldsymbol{I}$ and the initialization is infinitesimally small, we exploit the equivalence between linear attention and linear networks to derive an analytical time-course solution (see Appendix D.5) and obtain

$$\mathsf{ATTN}_\mathrm{M}(\boldsymbol{X}; t)_{D+1,N+1} = \sigma(t) \boldsymbol{\beta}^\top \boldsymbol{x}_q,$$

$$\text{where } \sigma(t) = \frac{e^{2\sqrt{D}\frac{t}{\tau}}}{\left(1 + \frac{1+D}{N}\right)\left(e^{2\sqrt{D}\frac{t}{\tau}} - 1\right) + \frac{\sqrt{D}}{w_\mathrm{init}^2}}. \tag{10}$$

Since $\sigma(t)$ is a rescaled and shifted sigmoid function, the weights and the loss trajectories have sigmoidal shapes, characterized by a plateau followed by a rapid drop.

3.4. ICL Algorithm: Least Squares Regression

When the linear attention model converges to the global minimum manifold $\mathcal{M}_*$ at the end of training, the model implements

$$\begin{aligned}\mathsf{ATTN}_{\mathrm{M}}(\boldsymbol{X})_{D+1,N+1} &= \mathbb{E}\left(y_q \boldsymbol{z}^\top\right) \mathbb{E}\left(\boldsymbol{z}\boldsymbol{z}^\top\right)^{-1} \boldsymbol{z} \\ &= \boldsymbol{\beta}^\top \left(\boldsymbol{\Lambda} + \frac{\boldsymbol{\Lambda} + \mathtt{tr}(\boldsymbol{\Lambda})\boldsymbol{I}}{N}\right)^{-1} \boldsymbol{x}_q, \end{aligned} \tag{11}$$

where the first equality follows directly from Equations (6) and (9b) and the second equality is proved in Appendix D.4. Equation (11) reveals an intriguing duality: the linear regression solution in the cubic feature space of $\boldsymbol{z}$ is the in-context linear regression solution in the original space of the $\boldsymbol{x}_n, y_n$ token pairs in a sequence $\boldsymbol{X}$. The first line of Equation (11) is the linear regression solution of fitting $y_{\mu,q}$ with $\boldsymbol{z}_\mu$ for all training sequences $\mu = 1, \cdots, P$. The second line of Equation (11) is approximately the in-context linear regression solution, which fits $y_{\mu,n}$ with $\boldsymbol{x}_{\mu,n} (n = 1, \cdots, N)$ for each sequence $\boldsymbol{X}_\mu$. When the sequence length N is large, the model recovers the inverse of the true covariance matrix,

$$\lim_{N\to\infty} \boldsymbol{\beta}^\top \left(\boldsymbol{\Lambda} + \frac{\boldsymbol{\Lambda} + \mathtt{tr}(\boldsymbol{\Lambda})\boldsymbol{I}}{N}\right)^{-1} \boldsymbol{x}_q = \boldsymbol{\beta}^\top \boldsymbol{\Lambda}^{-1} \boldsymbol{x}_q.$$

Here $\boldsymbol{\beta}$ is the $\boldsymbol{x}_n, y_n$ correlation in a sequence $\boldsymbol{X}$, and $\boldsymbol{\Lambda}$ is the covariance of all $\boldsymbol{x}_n$ tokens in all training sequences, which approximates the covariance of $\boldsymbol{x}_n$ in each individual sequence.

4. Linear Attention with Separate Rank-One Key and Query

We now study multi-head linear attention with separate low-rank key and query matrices. Because the rank-one case captures most of the behaviors of the general rank-R case, we focus on the rank-one case in this section and defer the rank-R case to Section 5. When $R = 1$, the model definition in Equation (S) simplifies to

$$\mathsf{ATTN}_{\mathrm{S}}(\boldsymbol{X})_{D+1,N+1} = \sum_{i=1}^{H} v_i \boldsymbol{\beta}^\top \boldsymbol{k}_i \boldsymbol{q}_i^\top \boldsymbol{x}_q. \tag{12}$$

4.1. Connection to Convolutional Linear Networks

The H-head linear attention with separate rank-one key and query can be viewed as a sum of H three-layer convolutional linear network with the cubic feature $\boldsymbol{z}$ defined in Equation (7) as input. Specifically, Equation (12) can be rewritten as

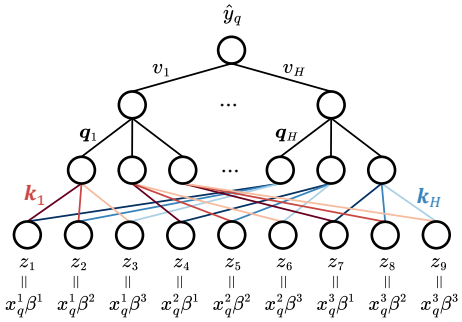


Figure 2. Multi-head linear attention with separate rank-one key and query $\mathsf{ATTN}_{\mathrm{S}}(\boldsymbol{X})_{D+1,N+1}$ is a sum of H (number of heads) three-layer convolutional linear networks with the cubic feature $\boldsymbol{z}$ as input. Here we take $D = 3$ to avoid clutter. Entries in the vectors are denoted as $\boldsymbol{x}_q = \left[x_q^1, x_q^2, x_q^3\right]^\top, \boldsymbol{\beta} = \left[\beta^1, \beta^2, \beta^3\right]^\top$.

$$\mathsf{ATTN}_{\mathrm{S}}(\boldsymbol{X})_{D+1,N+1} = \sum_{i=1}^{H} v_i \boldsymbol{q}_i^\top \boldsymbol{K}_i \boldsymbol{z},$$

$$\text{where } \boldsymbol{K}_i = \begin{bmatrix} \boldsymbol{k}_i^\top & \boldsymbol{0}_D^\top & \dots & \boldsymbol{0}_D^\top \\ \boldsymbol{0}_D^\top & \boldsymbol{k}_i^\top & \dots & \boldsymbol{0}_D^\top \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0}_D^\top & \boldsymbol{0}_D^\top & \dots & \boldsymbol{k}_i^\top \end{bmatrix} \in \mathbb{R}^{D\times D^2}. \tag{13}$$

The matrix $\boldsymbol{K}_i$ is a convolutional matrix with kernel size D and stride D. A schematic of the three-layer convolutional linear network is given in Figure 2.

When the number of heads satisfies $H \geq D$, the linear attention with separate rank-one key and query, $\mathsf{ATTN}_{\mathrm{S}}(\boldsymbol{X})$, can express any linear map of $\boldsymbol{z}(\boldsymbol{X})$ and has the same expressivity as linear attention with merged key and query, $\mathsf{ATTN}_{\mathrm{M}}(\boldsymbol{X})$. However, the two models correspond to multi-layer linear networks with different connectivity and depths, resulting in different loss landscape (Kohn et al., 2022; 2024) and training dynamics (Saxe et al., 2014; 2019).

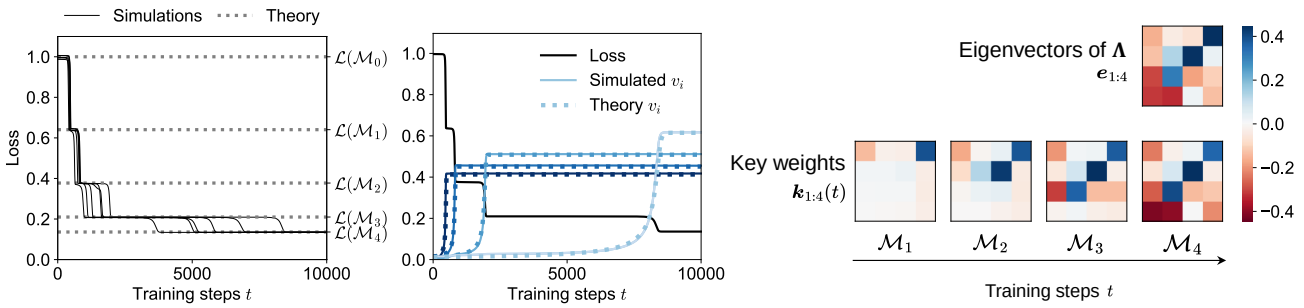
4.2. Loss Landscape: Exponentially Many Fixed Points

The gradient flow training dynamics of linear attention with separate rank-one key and query, derived in Appendix E.2, is given by

$$\tau \dot{v}_i = \boldsymbol{k}_i^\top \left(\boldsymbol{\Lambda}^2 - \mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right) \sum_{i'=1}^{H} v_{i'} \boldsymbol{k}_{i'} \boldsymbol{q}_{i'}^\top \boldsymbol{\Lambda}\right) \boldsymbol{q}_i, \tag{14a}$$

$$\tau \dot{\boldsymbol{k}}_i = v_i \left(\boldsymbol{\Lambda}^2 - \mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right) \sum_{i'=1}^{H} v_{i'} \boldsymbol{k}_{i'} \boldsymbol{q}_{i'}^\top \boldsymbol{\Lambda}\right) \boldsymbol{q}_i, \tag{14b}$$

$$\tau \dot{\boldsymbol{q}}_i = v_i \left(\boldsymbol{\Lambda}^2 - \boldsymbol{\Lambda} \sum_{i'=1}^{H} v_{i'} \boldsymbol{k}_{i'} \boldsymbol{q}_{i'}^\top \mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right)\right) \boldsymbol{k}_i, \tag{14c}$$



(a) Loss (b) Loss and value weights (c) Key weights and eigenvectors of $\mathbf{\Lambda}$

Figure 3. Multi-head linear attention with separate rank-one key and query exhibits saddle-to-saddle dynamics. (a) The loss curve has D abrupt drops, separated by plateaus (six runs from different random initialization are plotted). The loss at each plateau matches our theoretical prediction in Equation (19) (dashed gray lines). (b) The value weight v_i in each head for one of the runs in (a) is plotted in solid blue curves. The numerical solutions of v_i from Equation (21) are plotted in dashed blue curves and match the simulations well. The shades of blue distinguish different heads. (c) The key weights during the loss plateau are plotted in color. When the model moves from one fixed point to the next, the key weight in a head, $\boldsymbol{k}_i$, aligns with a new eigenvector of the input token covariance $\mathbf{\Lambda}$. The key weights $\boldsymbol{k}_{1:4}$ and the eigenvectors $\boldsymbol{e}_{1:4}$ are rows in the heatmaps. A video of the dynamics is provided at URL. Here $D = 4, N = 31, H = 4$, and $\mathbf{\Lambda}$ has eigenvalues $0.4, 0.3, 0.2, 0.1$ and eigenvectors as plotted in (c).

where we denote the in-context covariance of $\boldsymbol{x}_n$ tokens as $\hat{\mathbf{\Lambda}} = \sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^\top / N$ and the expectation of $\hat{\mathbf{\Lambda}}^2$ is

$$\mathbb{E}\left(\hat{\mathbf{\Lambda}}^2\right) = \mathbf{\Lambda}^2 + \frac{\mathbf{\Lambda} + \mathtt{tr}(\mathbf{\Lambda})\boldsymbol{I}}{N}\mathbf{\Lambda} \tag{15}$$

This dynamical system contains 2^D fixed points in the function space of $\mathsf{ATTN}_\mathrm{S}(\boldsymbol{X})_{D+1,N+1}$. We specify the fixed points below and prove their validity in Appendix E.3.

Let $\lambda_1, \cdots, \lambda_D$ be the eigenvalues of the covariance matrix $\mathbf{\Lambda}$ arranged in descending order, and $\boldsymbol{e}_1, \cdots, \boldsymbol{e}_D$ be the corresponding normalized eigenvectors. We use $\mathcal{M}(\mathcal{S}_m)$ to denote a set of fixed points that correspond to learning m ($m = 0, 1, \cdots, D$) out of the D eigenvectors,

$$\mathcal{M}(\mathcal{S}_m) = \left\{(v, \boldsymbol{k}, \boldsymbol{q})_{1:H} \middle| \text{conditions (C1)-(C3)}\right\}, \tag{16}$$

where the set $\mathcal{S}_m$ specifies the indices of the learned eigenvectors,

$$\mathcal{S}_m \subseteq \{1, 2, \cdots, D\},\ |\mathcal{S}_m| = m. \tag{17}$$

The three conditions for Equation (16) are:

(C1) The heads sum to fit the eigenvectors with indices in the set $\mathcal{S}_m$

$$\sum_{i=1}^{H} v_i \boldsymbol{k}_i \boldsymbol{q}_i^\top = \sum_{d \in \mathcal{S}_m} \lambda_d^{-1} \left(1 + \frac{1 + \mathtt{tr}(\mathbf{\Lambda})/\lambda_d}{N}\right)^{-1} \boldsymbol{e}_d \boldsymbol{e}_d^\top. \tag{18}$$

(C2) For heads with a nonzero value weight, $v_i \neq 0$, both $\boldsymbol{k}_i$ and $\boldsymbol{q}_i$ lie in the span of $\{\boldsymbol{e}_d\}_{d \in \mathcal{S}_m}$.

(C3) For heads with a zero value weight, $v_i = 0$, at least one of $\boldsymbol{k}_i$ or $\boldsymbol{q}_i$ lies in the span of $\{\boldsymbol{e}_d\}_{d \in \mathcal{S}_m}$.

Since there are $\binom{D}{m}$ possible ways of choosing m out of D indices to define $\mathcal{S}_m$ in Equation (17), the total number of possible choices summed over $m = 0, \cdots, D$ is $\sum_{m=0}^{D} \binom{D}{m} = 2^D$. Each choice corresponds to a different condition (C1) in Equation (18) and thus a different function, $\mathsf{ATTN}_\mathrm{S}(\boldsymbol{X})_{D+1,N+1}$. Hence, the gradient flow dynamics in Equation (14) has 2^D fixed points in the function space. [3]

The two fixed points of ATTN_M (Section 3.2) are contained in the 2^D fixed points of ATTN_S: the zero fixed point in Equation (9a) corresponds to $\mathcal{M}(\mathcal{S}_0)$, i.e., learning no eigenvector; the global minimum fixed point in Equation (9b) corresponds to $\mathcal{M}(\mathcal{S}_D)$, i.e., learning all D eigenvectors.

4.3. Training Dynamics: Saddle-to-Saddle Dynamics

Building on the exponentially many fixed points we have identified, we now analyze which fixed points are actually visited in gradient flow training and in what order. We find that starting from small initialization, the model visits $(D + 1)$ out of the 2^D fixed points.

With small initialization, the model is initially near the unstable zero fixed point, $\mathcal{M}_0 = \mathcal{M}(\varnothing)$. As training progresses, the model sequentially visits the fixed points in $\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_D$, where $\mathcal{M}_m = \mathcal{M}(\{1, 2, \cdots, m\})$. That is, the model trained from small initialization sequentially learns to fit the first eigenvector (the eigenvector of $\mathbf{\Lambda}$ with the largest eigenvalue), the second eigenvector, and so on. As shown in Figure 3a, the loss goes through D abrupt drops in training, each corresponding to the transition from one fixed point to the next. The abrupt drops of loss are separated by plateaus, during which the model lingers near

[3] A fixed point in function space corresponds to a set of fixed points in weight space that implement the same input-output map.

an unstable fixed point. Because the time required for a head to learn the eigenvector $\boldsymbol{e}_m$ from small initialization scales with λ_m^{-2} (see Appendix E.6), eigenvectors associated with larger eigenvalues are learned faster. This explains why the model learns to fit the eigenvectors sequentially in descending order of the eigenvalues, as well as why we empirically see the later plateaus last longer in Figure 3a.

When the model is at a fixed point in $\mathcal{M}_m$, we compute the loss in Appendix E.4 and obtain

$$\mathcal{L}(\mathcal{M}_m) = \mathtt{tr}(\boldsymbol{\Lambda}) - \sum_{d=1}^{m} \lambda_d \left(1 + \frac{1 + \mathtt{tr}(\boldsymbol{\Lambda})/\lambda_d}{N}\right)^{-1}. \tag{19}$$

Equation (19) is highly interpretable in the limit of a large sequence length N. The loss, $\mathcal{L}(\mathcal{M}_m)$, is the sum of the eigenvalues associated with the remaining unlearned eigenvectors

$$\lim_{N\to\infty} \mathcal{L}(\mathcal{M}_m) = \mathtt{tr}(\boldsymbol{\Lambda}) - \sum_{d=1}^{m} \lambda_d = \sum_{d=m+1}^{D} \lambda_d.$$

Thus, the loss decreases by approximately λ_m during the m-th abrupt loss drop. We plot Equation (19) as dashed gray lines in Figure 3a and find they match the plateaus of simulated loss trajectories well.

When the model reaches $\mathcal{M}_m$ from small initialization, its weights take on a highly structured form, which is a specific instance of the general definition in Equation (16). As shown in Figure 3c, the key and query weights in a head grow in scale and align with a new eigenvector of the input token covariance $\boldsymbol{\Lambda}$ during each abrupt loss drop. Based on simulations in Figure 3 and derivations in Appendices E.5 and E.6, we propose an ansatz that during the $(m+1)$-th plateau ($0 \le m < D$) and the subsequent abrupt drop of loss, the weights are approximately given by[4]

$$\boldsymbol{k}_i = \boldsymbol{q}_i = v_i \boldsymbol{e}_i,\ v_i = \lambda_i^{-\frac{1}{3}} \left(1 + \frac{1 + \mathtt{tr}(\boldsymbol{\Lambda})/\lambda_i}{N}\right)^{-\frac{1}{3}}, \quad 1 \le i \le m, \tag{20a}$$

$$\boldsymbol{k}_i = \boldsymbol{q}_i = v_i(t) \boldsymbol{e}_{m+1}, \quad i = m+1, \tag{20b}$$

$$\boldsymbol{k}_i = \boldsymbol{q}_i = \boldsymbol{0}, v_i = 0, \quad m+2 \le i \le H, \tag{20c}$$

where $v_{m+1}(t)$ is small during the $(m+1)$-th loss plateau and grows during the $(m+1)$-th abrupt loss drop. Equation (20) implies that the ℓ^2 norms of $v_i, \boldsymbol{k}_i, \boldsymbol{q}_i$ in a head are equal, which is a consequence of small initialization and the conservation law in Appendix E.8. With this ansatz, the high-dimensional training dynamics during the $(m+1)$-th plateau and the subsequent abrupt drop of loss reduces to an ordinary differential equation about $v_i(t), i = m+1$:

$$\tau \dot{v}_i = \lambda_{m+1}^2 v_i^2 - \lambda_{m+1}^3 \left(1 + \frac{1 + \mathtt{tr}(\boldsymbol{\Lambda})/\lambda_{m+1}}{N}\right) v_i^5. \tag{21}$$

Equation (21) is a separable differential equation but does not admit a general analytical solution of $v_{m+1}(t)$ in terms of t (see Equation (71)). Nonetheless, it greatly simplifies the high-dimensional dynamics in Equation (14) and provides a good approximation of the true dynamics: during each plateau and the subsequent abrupt loss drop, weights in one of the heads grow in scale with the key and query weights aligning with the next eigenvector, while the rest of the heads remain approximately unchanged. In Figure 3b, we compare the numerical solution of Equation (21) with the value weights trajectories in the simulation and find excellent agreement.

In summary, the loss trajectory of linear attention with separate rank-one key and query trained from small initialization exhibits D abrupt drops, each followed by a plateau. The amount of the m-th abrupt loss drop ($1 \le m \le D$) is approximately the eigenvalue λ_m, during which the key and query weights in an attention head grow in scale and align with the eigenvector $\boldsymbol{e}_m$.

4.4. ICL Algorithm: Principal Component Regression

When the linear attention model is at a fixed point in $\mathcal{M}_m$, based on Equation (18), the model implements

$$\mathsf{ATTN}_\mathsf{S}(\boldsymbol{X})_{D+1,N+1} = \boldsymbol{\beta}^\top \sum_{d=1}^{m} \lambda_d^{-1} \left(1 + \frac{1 + \mathtt{tr}(\boldsymbol{\Lambda})/\lambda_d}{N}\right)^{-1} \boldsymbol{e}_d \boldsymbol{e}_d^\top \boldsymbol{x}_q. \tag{22}$$

In the limit of a large sequence length N, Equation (22) simplifies and can be interpreted as principal component regression in context with m principal components

$$\lim_{N\to\infty} \mathsf{ATTN}_\mathsf{S}(\boldsymbol{X})_{D+1,N+1} = \boldsymbol{w}^\top \sum_{d=1}^{m} \boldsymbol{e}_d \boldsymbol{e}_d^\top \boldsymbol{x}_q.$$

Here $\boldsymbol{w}$ is the task vector for the sequence $\boldsymbol{X}$, and $\sum_{d=1}^{m} \boldsymbol{e}_d \boldsymbol{e}_d^\top \boldsymbol{x}_q$ is query input $\boldsymbol{x}_q$ projected onto the first m principal components. Hence, if training stops during the $(m+1)$-th plateau, the linear attention approximately implements the principal component regression algorithm in context with m principal components.

After the model has undergone D plateaus, it converges to the global minimum fixed point, $\mathcal{M}_D$, and approximately implements principal component regression in context with all D components, which is least square regression. Thus, the linear attention model with either merged or separate key and query undergoes different training dynamics but converges to the same global minimum solution.

[4] We permute the heads so that the head aligned with the d-th eigenvector have index d. The signs of any two among $v_i, \boldsymbol{k}_i, \boldsymbol{q}_i$ can be flipped with trivial effect on the analysis.

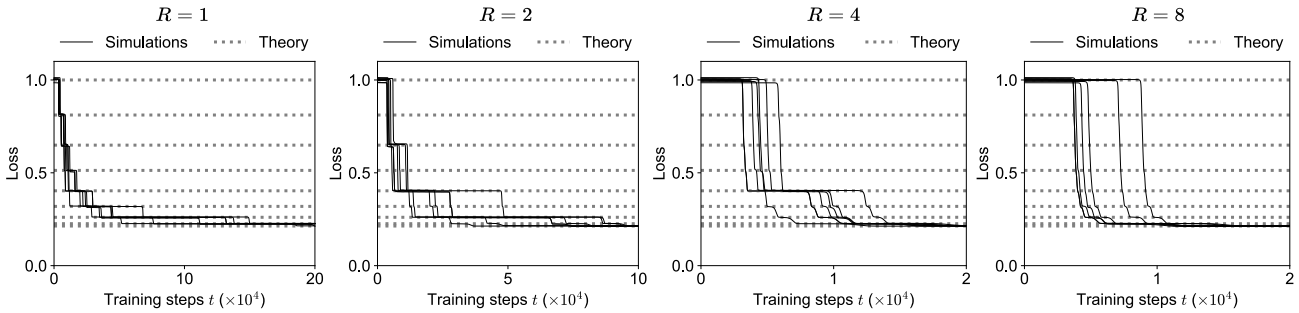


Figure 4. Multi-head linear attention with separate low-rank key and query exhibits saddle-to-saddle dynamics, with the duration of plateaus depending on the rank R. Solid black curves are loss trajectories from six random initializations. Dashed gray lines mark the loss values predicted by Equation (19) at nine fixed points, which are $\mathcal{L}(\mathcal{M}_0), \mathcal{L}(\mathcal{M}_1), \cdots, \mathcal{L}(\mathcal{M}_8)$ from top to bottom. The four panels differ only in the rank of the key and query weights. Here $D = 8, N = 31, H = 9$, $\mathbf{\Lambda}$ has trace 1 and eigenvalues $\lambda_d \propto d^{-1}$.

5. Linear Attention with Separate Low-Rank Key and Query

The linear attention model with separate rank-R key and query shares many behaviors with its rank-one counterpart. For loss landscape, linear attention with rank-R key and query has the same 2^D fixed points in the function space as its rank-one counterpart, corresponding to the model implementing in-context principal component regression with a subset of all D principal components (see Appendix F.3).

For training dynamics, the loss trajectories differ slightly, depending on the rank R. We plot the loss trajectories with input token dimension $D = 8$ and different ranks $R = 1, 2, 4, 8$ in Figure 4 (see Figure 12 for $R = 3, 5, 6, 7$). For $R = 1$, the loss exhibits plateaus at eight values $\mathcal{L}(\mathcal{M}_m)\,(m = 0, 1, \cdots, 7)$. For $R = 2$, the loss exhibits plateaus at four values $\mathcal{L}(\mathcal{M}_m)\,(m = 0, 2, 4, 6)$, and either brief plateaus or no plateau at the other four values. For $R = 4$, the loss exhibits conspicuous plateaus at only two values $\mathcal{L}(\mathcal{M}_m)\,(m = 0, 4)$. To summarize, with rank-R key and query, the loss trajectory exhibits conspicuous plateaus at value $\mathcal{L}(\mathcal{M}_m)$ for m that divides R.

The difference in the loss trajectories arises from the structure of the model defined in Equation (S). Each attention head has a single value weight v_i that is associated with all R pairs of key and query weights in that head, $\boldsymbol{k}_{i,r}, \boldsymbol{q}_{i,r}\,(r = 1, \cdots, R)$. During a conspicuous plateau, a new value weight escapes from the unstable zero fixed point and grows in scale. Once the value weight has grown, it leads to larger gradient updates for all the key and query weights in that head, speeding up their escape from the zero fixed point. Hence, in the rank-R case, a conspicuous plateau occurs when m divides R, corresponding to learning a new head from small initialization. Brief or no plateau occurs when m does not divide R, corresponding to learning a new pair of key and query weights in a head whose value weight has already grown, as shown in Figure 11. See Appendix F.4 for further details.

6. Related Work

Recent theoretical research on linear attention has investigated its expressivity (Vladymyrov et al., 2024; Gatmiry et al., 2024), learnability (Yau et al., 2024), loss landscape (Mahankali et al., 2024; Li et al., 2024), convergence (Zhang et al., 2024a;b; Ren et al., 2024; Fu et al., 2024), and generalization (Wu et al., 2024; Mahankali et al., 2024; Duraisamy, 2024; Abedsoltan et al., 2024; Lu et al., 2025; Frei & Vardi, 2025). The seminal work by Zhang et al. (2024a) analyzed the gradient flow training dynamics of linear attention to prove convergence guarantees, showing what the model converges to at the end of training. Our work also analyzes the gradient flow training dynamics but goes beyond existing convergence results to describe the entire training dynamics. Moreover, we study multi-head attention with merged or separate key and query weights, while Zhang et al. (2024a) focused on single-head attention with merged key and query.

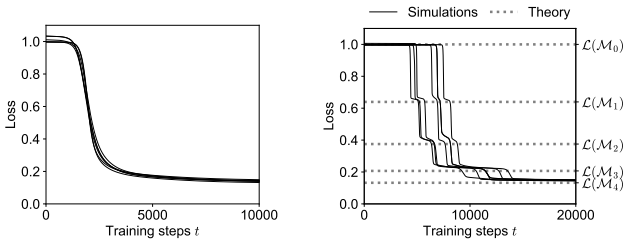
Another line of recent research on the training dynamics of softmax attention models has shown stage-wise dynamics. Due to the intractability of softmax attention training dynamics in general, many of these studies made strong assumptions to enable theoretical analyses, including a simplified layer-wise training algorithm in place of standard gradient descent (Tian et al., 2023; Nichani et al., 2024; Chen et al., 2024c; Wang et al., 2024a), restricted weights (Boix-Adsera et al., 2023; Chen et al., 2024b; Rende et al., 2024; Edelman et al., 2024), and specifically chosen datasets (Huang et al., 2024b). In comparison, our work leverages the linear attention model without the softmax operation, enabling us to study in fine detail the dynamics of standard gradient descent training without restrictions on weights. Namely, we derive an analytical time-course solution and reduce the high-dimensional dynamics to one-dimensional ordinary differential equations for the two models we study, respectively. Furthermore, we characterize how parametrization (i.e., merged or separate key and query, and rank of the separate key and query weights) affects the loss landscape and training dynamics, an aspect not previously examined.

7. Discussion

We studied the gradient flow training dynamics of multi-head linear attention and demonstrated how it acquires ICL abilities in training. We begin with a simple setting of linear attention with merged key and query trained for in-context linear regression, following the setting in seminal works (Von Oswald et al., 2023; Ahn et al., 2023; Zhang et al., 2024a). We show an abrupt loss drop in training and give an analytical time-course solution in the case of a white input token covariance and small initialization. However, a single abrupt loss drop does not fully capture the evolution of ICL in training practical transformers, where the abilities continue to develop throughout training (Xia et al., 2023; Park et al., 2025). We thus extend our analysis to a parametrization closer to the attention in practical transformers: attention with separate key and query. In the separate case, we find that the loss exhibits saddle-to-saddle dynamics with multiple abrupt drops. The ICL ability evolves progressively, manifesting as implementing principal component regression in context, with the number of principal components increasing over training time. We thus characterize how the linear attention model develops increasingly sophisticated ICL abilities in gradient descent training.

Softmax Attention. We empirically find that the different training dynamics of linear ATTN_M and linear ATTN_S also occur in their softmax counterparts. Figure 5 follows the same setup as Figures 1 and 3 for linear attention, with the only difference being adding the softmax activation function for the attention calculation. We observe that softmax attention with merged key and query exhibits a single abrupt loss drop, whereas softmax attention with separate rank-one key and query undergoes multiple loss drops, separated by phases of conspicuously slower training. This suggests that our findings and theoretical intuition are not unique to linear attention but may also extend to softmax attention.

Effect of Initialization. Having analyzed the small initialization case, we now examine how the initialization scale affects training dynamics. For linear ATTN_M, Figure 6a shows increasing initialization shortens the plateau before the single abrupt loss drop. For linear ATTN_S, Figure 6b shows increasing initialization shortens all plateaus between successive abrupt loss drops. At the largest initialization, both models exhibit an exponential-shaped loss decay – a hallmark of lazy learning (Chizat et al., 2019). In contrast, rich learning typically exhibits abrupt sigmoid-shaped loss curves as seen in our main result. Theory typically focuses on either the lazy or rich regime, while practical initializations often fall in between. In Figure 6, dynamics from the intermediate initialization seems like a mix of the exponential-shaped and sigmoid-shaped curves, which are often seen in practice, e.g. in induction head emergence in natural language settings (Olsson et al., 2022, Argument 1).



(a) Softmax ATTN_M (b) Softmax ATTN_S with $R = 1$

Figure 5. Loss trajectories of softmax attention with merged or separate key and query. Six runs from different random initialization are plotted. Similar to the linear attention case, softmax ATTN_M exhibits one abrupt loss drop, while softmax ATTN_S exhibits multiple loss drops. The dataset and model setup are the same as Figures 1 and 3 except adding the softmax activation function.

Dynamics of In-Context and In-Weight Learning. Our work studies the training dynamics of in-context learning. Other than in-context learning, attention models can also learn in weight; that is, solving the task by memorizing the map between the query input and the target output without using the information in context. The arbitration between in-context and in-weight learning may depend on properties of training data (Chan et al., 2022). To focus on the dynamics of ICL, we used a purely ICL task, which is in-context linear regression with the task vector sampled from a zero-mean standard normal distribution, $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. Since memorizing any particular task vector does not effectively decrease the loss, linear attention develops only in-context learning ability during training, as shown in Figure 14a. If the task vector $\boldsymbol{w}$ follows a different distribution, the training dynamics involves the development of both in-context and in-weight learning abilities, as shown in Figure 14. We provide more details in Appendix G.

Implications for Future Theory. In our analysis, we draw connections between linear attention and multi-layer linear networks, enabling us to employ the rich theoretical machinery built for linear networks to understand linear attention training dynamics. Beyond training dynamics, many other theoretical results for linear networks can apply to linear attention through the equivalence we draw. For example, the convergence guarantee for multi-head linear attention trained on in-context linear regression tasks can be obtained from the convergence proofs for deep linear networks (Arora et al., 2019; Shamir, 2019). In contrast, without the equivalence, Zhang et al. (2024a) previously obtained a convergence guarantee for single-head linear attention, which required highly non-trivial derivations. Hence, we believe the connections we draw are useful in enabling the applications of theory from one architecture to the other.

Additionally, we have shown that parametrization significantly affects the loss landscape and training dynamics, motivating future research to examine how their results may or may not be influenced by the parametrization choice.

Acknowledgement

We thank Jin Hwa Lee, Sara Dragutinović, Andrew Lampinen, Basile Confavreux and William Tong for helpful conversations.

We thank the following funding sources: Gatsby Charitable Foundation (GAT3850) to YZ, AKS, PEL, and AS; Wellcome Trust (110114/Z/15/Z) to PEL; Sainsbury Wellcome Centre Core Grant from Wellcome (219627/Z/19/Z) to AS; Schmidt Science Polymath Award to AS. AS is a CIFAR Azrieli Global Scholar in the Learning in Machines & Brains program.

Impact Statement

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

References

Abedsoltan, A., Radhakrishnan, A., Wu, J., and Belkin, M. Context-scaling versus task-scaling in in-context learning, 2024. URL https://arxiv.org/abs/2410.12783.

Ahn, K., Cheng, X., Daneshmand, H., and Sra, S. Transformers learn to implement preconditioned gradient descent for in-context learning. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 45614–45650. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/8ed3d610ea4b68e7afb30ea7d01422c6-Paper-Conference.pdf.

Ahn, K., Cheng, X., Song, M., Yun, C., Jadbabaie, A., and Sra, S. Linear attention is (maybe) all you need (to understand transformer optimization). In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=0uI5415ry7.

Akyürek, E., Schuurmans, D., Andreas, J., Ma, T., and Zhou, D. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=0g0X4H8yN4I.

Anand, S., Lepori, M. A., Merullo, J., and Pavlick, E. Dual process learning: Controlling use of in-context vs. in-weights strategies with weight forgetting. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=jDsmB4o5S0.

Anwar, U., Oswald, J. V., Kirsch, L., Krueger, D., and Frei, S. Adversarial robustness of in-context learning in transformers for linear regression, 2024. URL https://arxiv.org/abs/2411.05189.

Arora, S., Cohen, N., Golowich, N., and Hu, W. A convergence analysis of gradient descent for deep linear neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkMQg3C5K7.

Ataee Tarzanagh, D., Li, Y., Zhang, X., and Oymak, S. Max-margin token selection in attention mechanism. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 48314–48362. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/970f59b22f4c72aec75174aae63c7459-Paper-Conference.pdf.

Atanasov, A., Bordelon, B., and Pehlevan, C. Neural networks as kernel learners: The silent alignment effect. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=1NvflqAdoom.

Bai, Y., Chen, F., Wang, H., Xiong, C., and Mei, S. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 57125–57211. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/b2e63e36c57e153b9015fece2352a9f9-Paper-Conference.pdf.

Boix-Adsera, E., Littwin, E., Abbe, E., Bengio, S., and Susskind, J. Transformers learn through gradual rank increase. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 24519–24551. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/4d69c1c057a8bd570ba4a7b71aae8331-Paper-Conference.pdf.

Boix-Adserà, E., Saremi, O., Abbe, E., Bengio, S., Littwin, E., and Susskind, J. M. When can transformers reason with abstract symbols? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=STUGfUz8ob.

Bordelon, B., Chaudhry, H., and Pehlevan, C. Infinite limits of multi-head transformer dynamics. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 35824–35878. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/3eff068e195daace49955348de9f8398-Paper-Conference.pdf.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Chan, B., Chen, X., György, A., and Schuurmans, D. Toward understanding in-context vs. in-weight learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=aKJr5NnN8U.

Chan, S., Santoro, A., Lampinen, A., Wang, J., Singh, A., Richemond, P., McClelland, J., and Hill, F. Data distributional properties drive emergent in-context learning in transformers. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 18878–18891. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/77c6ccacfd9962e2307fc64680fc5ace-Paper-Conference.pdf.

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

Chen, S. and Li, Y. Provably learning a multi-head attention layer. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25, pp. 1744–1754, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400715105. doi: 10.1145/3717823.3718174. URL https://doi.org/10.1145/3717823.3718174.

Chen, S., Sheen, H., Wang, T., and Yang, Z. Training dynamics of multi-head softmax attention for in-context learning: Emergence, convergence, and optimality. In Agrawal, S. and Roth, A. (eds.), *Proceedings of Thirty Seventh Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pp. 4573–4573. PMLR, 30 Jun–03 Jul 2024b. URL https://proceedings.mlr.press/v247/siyu24a.html.

Chen, S., Sheen, H., Wang, T., and Yang, Z. Unveiling induction heads: Provable training dynamics and feature learning in transformers. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 66479–66567. Curran Associates, Inc., 2024c. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/7aae9e3ec211249e05bd07271a6b1441-Paper-Conference.pdf.

Chizat, L., Oyallon, E., and Bach, F. On lazy training in differentiable programming. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/ae614c557843b1df326cb29c57225459-Paper.pdf.

Du, S. S., Hu, W., and Lee, J. D. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/fe131d7f5a6b38b23cc967316c13dae2-Paper.pdf.

Duraisamy, K. Finite sample analysis and bounds of generalization error of gradient descent in in-context linear regression, 2024. URL https://arxiv.org/abs/2405.02462.

Edelman, E., Tsilivis, N., Edelman, B., Malach, E., and Goel, S. The evolution of statistical induction heads: In-context learning markov chains. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 64273–64311. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/75b0edb869e2cd509d64d0e8ff446bc1-Paper-Conference.pdf.

Frei, S. and Vardi, G. Trained transformer classifiers generalize and exhibit benign overfitting in-context. In *The*

Thirteenth International Conference on Learning Representations, 2025. URL https://openreview.net/forum?id=jwsPS8yRe4.

Fu, D., Chen, T.-q., Jia, R., and Sharan, V. Transformers learn to achieve second-order convergence rates for in-context linear regression. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 98675–98716. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/b2d4051f03a7038a2771dfbbe5c7b54e-Paper-Conference.pdf.

Fukumizu, K. Effect of batch learning in multilayer neural networks. *Gen*, 1(04):1E–03, 1998.

Garg, S., Tsipras, D., Liang, P. S., and Valiant, G. What can transformers learn in-context? a case study of simple function classes. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 30583–30598. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/c529dba08a146ea8d6cf715ae8930cbe-Paper-Conference.pdf.

Gatmiry, K., Saunshi, N., Reddi, S. J., Jegelka, S., and Kumar, S. On the role of depth and looping for in-context learning with task diversity, 2024. URL https://arxiv.org/abs/2410.21698.

Geshkovski, B., Koubbi, H., Polyanskiy, Y., and Rigollet, P. Dynamic metastability in the self-attention model, 2024. URL https://arxiv.org/abs/2410.06833.

Gopalani, P., Lubana, E. S., and Hu, W. Abrupt learning in transformers: A case study on matrix completion. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 55053–55085. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/630d293833e09e1ecd892a898a20b074-Paper-Conference.pdf.

He, J., Pan, X., Chen, S., and Yang, Z. In-context linear regression demystified: Training dynamics and mechanistic interpretability of multi-head softmax attention, 2025. URL https://arxiv.org/abs/2503.12734.

He, T., Doshi, D., Das, A., and Gromov, A. Learning to grok: Emergence of in-context learning and skill composition in modular arithmetic tasks. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 13244–13273. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/17d60fef592086d1a5cb136f1946df59-Paper-Conference.pdf.

Hoffmann, D. T., Schrodi, S., Bratulić, J., Behrmann, N., Fischer, V., and Brox, T. Eureka-moments in transformers: Multi-step tasks reveal softmax induced optimization problems. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 18409–18438. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/hoffmann24a.html.

Huang, J., Wang, Z., and Lee, J. D. Transformers learn to implement multi-step gradient descent with chain of thought. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=r3DF5sOo5B.

Huang, R., Liang, Y., and Yang, J. Non-asymptotic convergence of training transformers for next-token prediction. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 80634–80673. Curran Associates, Inc., 2024a. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/9370fa016d6a14af78f5048bfcb0582b-Paper-Conference.pdf.

Huang, Y., Cheng, Y., and Liang, Y. In-context convergence of transformers. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 19660–19722. PMLR, 21–27 Jul 2024b. URL https://proceedings.mlr.press/v235/huang24d.html.

Huang, Y., Wen, Z., Chi, Y., and Liang, Y. A theoretical analysis of self-supervised learning for vision transformers. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=Antib6Uovh.

Ildiz, M. E., Huang, Y., Li, Y., Rawat, A. S., and Oymak, S. From self-attention to Markov models: Unveiling the dynamics of generative transformers. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *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. URL https://proceedings.mlr.press/v235/ildiz24a.html.

Jang, U., Lee, J. D., and Ryu, E. K. LoRA training in the NTK regime has no spurious local minima. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 21306–21328. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/jang24d.html.

Jelassi, S., Sander, M., and Li, Y. Vision transformers provably learn spatial structure. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 37822–37836. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/f69707de866eb0805683d3521756b73f-Paper-Conference.pdf.

Ji, Z. and Telgarsky, M. Gradient descent aligns the layers of deep linear networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJflg30qKX.

Jiang, J., Huang, W., Zhang, M., Suzuki, T., and Nie, L. Unveil benign overfitting for transformer in vision: Training dynamics, convergence, and generalization. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 135464–135625. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/f49287371916715b9209fa41a275851e-Paper-Conference.pdf.

Julistiono, A. A. K., Tarzanagh, D. A., and Azizan, N. Optimizing attention with mirror descent: Generalized max-margin token selection, 2024. URL https://arxiv.org/abs/2410.14581.

Kim, J. and Suzuki, T. Transformers learn nonlinear features in context: Nonconvex mean-field dynamics on the attention landscape. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 24527–24561. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/kim24af.html.

Kohn, K., Merkh, T., Montúfar, G., and Trager, M. Geometry of linear convolutional networks. *SIAM Journal on Applied Algebra and Geometry*, 6(3):368–406, 2022. doi: 10.1137/21M1441183. URL https://doi.org/10.1137/21M1441183.

Kohn, K., Montúfar, G., Shahverdi, V., and Trager, M. Function space and critical points of linear convolutional networks. *SIAM Journal on Applied Algebra and Geometry*, 8(2):333–362, 2024. doi: 10.1137/23M1565504. URL https://doi.org/10.1137/23M1565504.

Lee, I., Jiang, N., and Berg-Kirkpatrick, T. Is attention required for ICL? exploring the relationship between model architecture and in-context learning ability. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Qwq4cpLtoX.

Li, Y., Rawat, A., and Oymak, S. Fine-grained analysis of in-context linear estimation: Data, architecture, and beyond. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 138324–138364. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/f9dc462382fef56d58279e75de2438f3-Paper-Conference.pdf.

Lu, Y. M., Letey, M., Zavatone-Veth, J. A., Maiti, A., and Pehlevan, C. Asymptotic theory of in-context learning by linear attention. *Proceedings of the National Academy of Sciences*, 122(28):e2502599122, 2025. doi: 10.1073/pnas.2502599122. URL https://www.pnas.org/doi/abs/10.1073/pnas.2502599122.

Mahankali, A. V., Hashimoto, T., and Ma, T. One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=8p3fu56lKc.

Makkuva, A. V., Bondaschi, M., Girish, A., Nagle, A., Kim, H., Gastpar, M., and Ekbote, C. Local to global: Learning dynamics and effect of initialization for transformers. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 86243–86308. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/9cdb4f8c4dfa13284d2d5a6e7853e5a2-Paper-Conference.pdf.

Nanda, N., Chan, L., Lieberum, T., Smith, J., and Steinhardt, J. Progress measures for grokking via mechanistic

interpretability. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=9XFSbDPmdW.

Nguyen, A. and Reddy, G. Differential learning kinetics govern the transition from memorization to generalization during in-context learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=INyi7qUdjZ.

Nichani, E., Damian, A., and Lee, J. D. How transformers learn causal structure with gradient descent. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 38018–38070. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/nichani24a.html.

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Johnston, S., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., and Olah, C. In-context learning and induction heads, 2022. URL https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

Otto, F. and Reznikoff, M. G. Slow motion of gradient flows. *Journal of Differential Equations*, 237(2):372–420, 2007. ISSN 0022-0396. doi: https://doi.org/10.1016/j.jde.2007.03.007. URL https://www.sciencedirect.com/science/article/pii/S0022039607000824.

Park, C. F., Lubana, E. S., and Tanaka, H. Algorithmic phases of in-context learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=XgH1wfHSX8.

Reddy, G. The mechanistic basis of data dependence and abrupt learning in an in-context classification task. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=aN4Jf6Cx69.

Ren, Y., Wang, Z., and Lee, J. D. Learning and transferring sparse contextual bigrams with linear transformers. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 20304–20357. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/2428ff361a08bc6864fb240bc83fba42-Paper-Conference.pdf.

Rende, R., Gerace, F., Laio, A., and Goldt, S. A distributional simplicity bias in the learning dynamics of transformers. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 96207–96228. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/ae6c81a39079ddeb88b034b6ef18c7fe-Paper-Conference.pdf.

Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *The Second International Conference on Learning Representations*, 2014. URL https://openreview.net/forum?id=_wzZwKpTDF_9C.

Saxe, A. M., McClelland, J. L., and Ganguli, S. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23):11537–11546, 2019. doi: 10.1073/pnas.1820226116. URL https://www.pnas.org/doi/abs/10.1073/pnas.1820226116.

Schlag, I., Irie, K., and Schmidhuber, J. Linear transformers are secretly fast weight programmers. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9355–9366. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/schlag21a.html.

Shamir, O. Exponential convergence time of gradient descent for one-dimensional deep linear neural networks. In Beygelzimer, A. and Hsu, D. (eds.), *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pp. 2691–2713. PMLR, 25–28 Jun 2019. URL https://proceedings.mlr.press/v99/shamir19a.html.

Sheen, H., Chen, S., Wang, T., and Zhou, H. H. Implicit regularization of gradient flow on one-layer softmax attention, 2024. URL https://arxiv.org/abs/2403.08699.

Singh, A. K., Chan, S., Moskovitz, T., Grant, E., Saxe, A., and Hill, F. The transient nature of emergent in-context learning in transformers. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 27801–27819. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/p

aper_files/paper/2023/file/58692a1701314e09cbd7a5f5f3871cc9-Paper-Conference.pdf.

Singh, A. K., Moskovitz, T., Hill, F., Chan, S. C., and Saxe, A. M. What needs to go right for an induction head? A mechanistic study of in-context learning circuits and their formation. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 45637–45662. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/singh24c.html.

Singh, A. K., Moskovitz, T., Dragutinovic, S., Hill, F., Chan, S. C. Y., and Saxe, A. M. Strategy coopetition explains the emergence and transience of in-context learning, 2025. URL https://arxiv.org/abs/2503.05631.

Song, B., Han, B., Zhang, S., Ding, J., and Hong, M. Unraveling the gradient descent dynamics of transformers. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 92317–92351. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/a7d36e5cb41a1f21c46db25cb1aafab9-Paper-Conference.pdf.

Tarzanagh, D. A., Li, Y., Thrampoulidis, C., and Oymak, S. Transformers as support vector machines, 2024. URL https://arxiv.org/abs/2308.16898.

Tian, Y., Wang, Y., Chen, B., and Du, S. S. Scan and snap: Understanding training dynamics and token composition in 1-layer transformer. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 71911–71947. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/e359ebe56ba306b674e8952349c6049e-Paper-Conference.pdf.

Tong, W. L. and Pehlevan, C. MLPs learn in-context on regression and classification tasks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=MbX0t1rUlp.

Vasudeva, B., Deora, P., and Thrampoulidis, C. Implicit bias and fast convergence rates for self-attention. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=pKilnjQsb0.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Vladymyrov, M., von Oswald, J., Sandler, M., and Ge, R. Linear transformers are versatile in-context learners. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 48784–48809. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/57a3c602f0a1c8980cc5ed07e49d9490-Paper-Conference.pdf.

Von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A., and Vladymyrov, M. Transformers learn in-context by gradient descent. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 35151–35174. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/von-oswald23a.html.

Wang, M., Yu, R., E, W., and Wu, L. How transformers implement induction heads: Approximation and optimization analysis, 2024a. URL https://arxiv.org/abs/2410.11474.

Wang, Z., Wei, S., Hsu, D., and Lee, J. D. Transformers provably learn sparse token selection while fully-connected nets cannot. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 51854–51912. PMLR, 21–27 Jul 2024b. URL https://proceedings.mlr.press/v235/wang24ca.html.

Woodworth, B., Gunasekar, S., Lee, J. D., Moroshko, E., Savarese, P., Golan, I., Soudry, D., and Srebro, N. Kernel and rich regimes in overparametrized models. In Abernethy, J. and Agarwal, S. (eds.), *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pp. 3635–3673. PMLR, 09–12 Jul 2020. URL https://proceedings.mlr.press/v125/woodworth20a.html.

Wu, J., Zou, D., Chen, Z., Braverman, V., Gu, Q., and Bartlett, P. How many pretraining tasks are needed for

in-context learning of linear regression? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=vSh5ePa0ph.

Xia, M., Artetxe, M., Zhou, C., Lin, X. V., Pasunuru, R., Chen, D., Zettlemoyer, L., and Stoyanov, V. Training trajectories of language models across scales. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13711–13738, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.767. URL https://aclanthology.org/2023.acl-long.767.

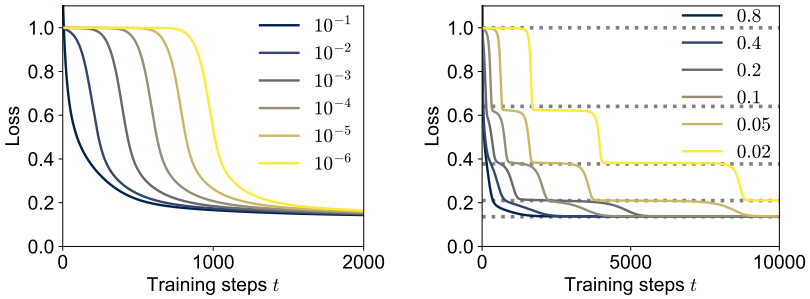
Yau, M., Akyürek, E., Mao, J., Tenenbaum, J. B., Jegelka, S., and Andreas, J. Learning linear attention in polynomial time, 2024. URL https://arxiv.org/abs/2410.10101.

Zhang, R., Frei, S., and Bartlett, P. L. Trained transformers learn linear models in-context. *Journal of Machine Learning Research*, 25(49):1–55, 2024a. URL http://jmlr.org/papers/v25/23-1042.html.

Zhang, R., Wu, J., and Bartlett, P. In-context learning of a linear transformer block: Benefits of the mlp component and one-step gd initialization. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 18310–18361. Curran Associates, Inc., 2024b. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/20b6b87ca1792337f414d948af7b0e8-Paper-Conference.pdf.

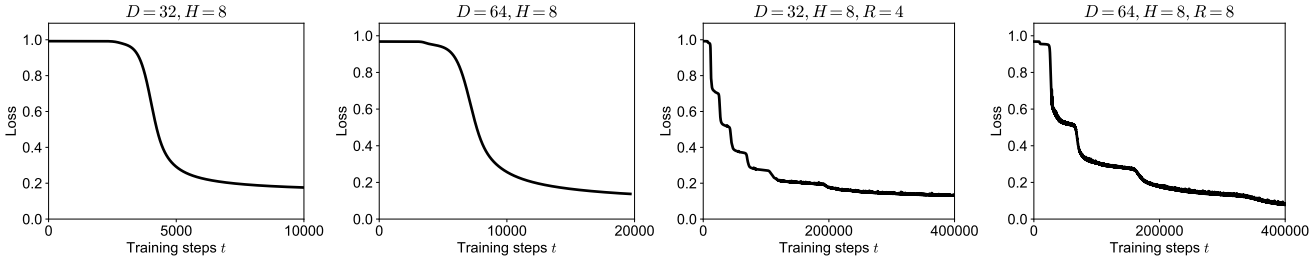
Table of Contents

Appendix



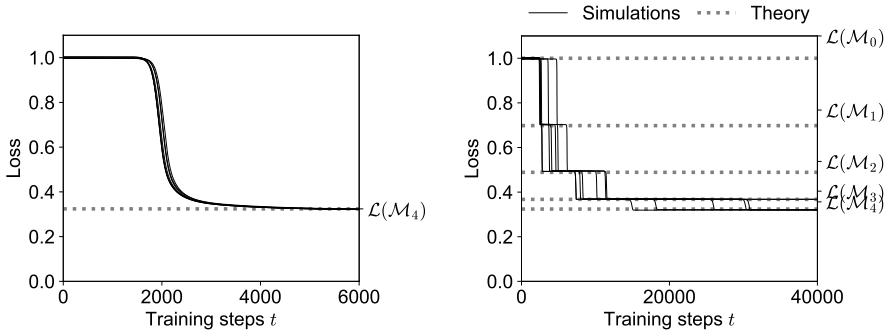
(a) Linear ATTN_M (b) Linear ATTN_S with $R = 1$

Figure 6. Loss trajectories of linear ATTN_M and ATTN_S with varying initialization scales. The colors indicate the initialization scale. Increasing the initialization scale shortens the plateaus. With small initialization, the models are in the rich feature learning regime, exhibiting abrupt sigmoid-shaped dynamics. With large initialization, they are in the lazy learning regime, exhibiting exponential-shaped loss decay. The loss curve from intermediate initialization seems like a mix of the exponential-shaped and sigmoid-shaped curves. Such mixed curves are often seen in practice, such as in induction head emergence in natural language settings (Olsson et al., 2022, Argument 1). The dataset and model setup are the same as Figures 1 and 3, except that we vary the initialization scale.



(a) Linear ATTN_M (b) Linear ATTN_S

Figure 7. Loss trajectories of linear ATTN_M and ATTN_S with high-dimensional data. Here the sequence length is $N = 127$, $\mathbf{\Lambda}$ has trace 1 and eigenvalues $\lambda_d \propto d^{-1}$. Other hyperparameters are labeled at the top of each panel.



(a) Linear ATTN_M (b) Linear ATTN_S with $R = 1$

Figure 8. Loss trajectories of linear ATTN_M and ATTN_S trained with the next token prediction loss defined in Equation (23) with $N_{\max} = 31$. In this case, the models are trained on sequences of varying lengths, which they can handle due to the $1/N$ scaling factor in Equations (4), (M) and (S). The dataset and model setup are the same as Figures 1 and 3, except that we switch to the next token prediction loss.

A. Additional Figures

A.1. Effect of Initialization Scale

We vary the initialization scale of the linear attention models and plot their loss trajectories in Figure 6.

A.2. Higher Dimensions

We train ATTN_M and ATTN_S on a dataset with larger N, D. The loss trajectories are qualitatively similar to those in lower-dimensional cases, despite being noisier. This suggests that our findings do not break in high-dimensional settings.

A.3. Varying Context Lengths

In our main results, we consider a fixed context length N, because our training sequences have the same length and the loss is computed only for the last query token, $\mathcal{L} = \mathbb{E}(y_q - \hat{y}_q)^2$. In practice, however, the training sequences may have varying lengths, and the loss can be computed for every token in the sequence, that is

$$\mathcal{L}_{\text{ntp}} = \mathbb{E}\left[\frac{1}{N_{\max}} \sum_{n=2}^{N_{\max}+1} (y_n - \hat{y}_n)^2\right], \tag{23}$$

where $y_{N+1} = y_q$, and $\hat{y}_n$ is the attention model's prediction for y_n when given only the first n columns of $\boldsymbol{X}$ as input. We demonstrate how our results apply to the case of varying context lengths. Specifically, the distribution of the context lengths only influences our results through a statistic, $\mathbb{E}(1/N)$.

For ATTN_M, derivations in Appendix D.4 show that the converged model implements

$$\mathsf{ATTN}_\mathrm{M}(\boldsymbol{X})_{D+1,N+1} = \sum_{i=1}^{H} v_i \boldsymbol{\beta}^\top \boldsymbol{U}_i \boldsymbol{x}_q = \boldsymbol{\beta}^\top \left[\mathbb{E}\left(\frac{1}{N} \boldsymbol{x}_n \boldsymbol{x}_n^\top\right)^2\right]^{-1} \boldsymbol{\Lambda} \boldsymbol{x}_q. \tag{24}$$

Substituting Equation (30) into Equation (24), we obtain

$$\mathsf{ATTN}_\mathrm{M}(\boldsymbol{X})_{D+1,N+1} = \boldsymbol{\beta}^\top \left[\boldsymbol{\Lambda} + \mathbb{E}\left(\frac{1}{N}\right)(\boldsymbol{\Lambda} + \mathtt{tr}(\boldsymbol{\Lambda})\boldsymbol{I})\right]^{-1} \boldsymbol{x}_q. \tag{25}$$

The distribution of context lengths only influences Equation (25) through the expectation $\mathbb{E}(1/N)$. For a fixed context length, $\mathbb{E}(1/N) = 1/N$, which recovers Equation (11) in the main text. For the next token prediction loss, the distribution of context lengths, $p(N)$, follows a uniform distribution over $\{1, 2, \cdots, N_{\max}\}$. The expectation $\mathbb{E}(1/N)$ is the harmonic number divided by $N_{\max}$, which doesn't have a closed-form expression but can be easily computed for a specific finite $N_{\max}$.

Similarly, for ATTN_S trained with varying context lengths, the fixed point condition (C1) takes the form

$$\sum_{i=1}^{H} v_i \boldsymbol{k}_i \boldsymbol{q}_i^\top = \sum_{d \in \mathcal{S}_m} \lambda_d^{-1} \left[1 + \mathbb{E}\left(\frac{1}{N}\right)(1 + \mathtt{tr}(\boldsymbol{\Lambda})/\lambda_d)\right]^{-1} \boldsymbol{e}_d \boldsymbol{e}_d^\top,$$

where the expectation $\mathbb{E}(1/N)$ reduces to $1/N$ in the fixed context length case as in Equation (18). Consequently, when the model is at a fixed point in $\mathcal{M}_m$, the loss value is

$$\mathcal{L}(\mathcal{M}_m) = \mathtt{tr}(\boldsymbol{\Lambda}) - \sum_{d=1}^{m} \lambda_d \left[1 + \mathbb{E}\left(\frac{1}{N}\right)(1 + \mathtt{tr}(\boldsymbol{\Lambda})/\lambda_d)\right]^{-1}, \tag{26}$$

which reduces to Equation (19) when $\mathbb{E}(1/N) = 1/N$.

We train ATTN_M and ATTN_S with the next token prediction loss as in Equation (23) and plot the loss trajectories in Figure 8. The loss trajectories are qualitatively similar to those in Figures 1 and 3a, modulo the different loss values during the plateaus. We plot the loss values computed from Equation (26) as dashed gray lines and find they match the plateaus of the simulated loss trajectories well.

B. Additional Related Work

A concurrent work by Geshkovski et al. (2024) studies saddle-to-saddle-like dynamics in softmax attention models following a mathematical framework for slow motion of gradient flows (Otto & Reznikoff, 2007). A subsequent work by He et al. (2025) examines the training dynamics of softmax attention trained on the in-context linear regression task; that is, the case we briefly touch on in Figure 5.

A broader body of theoretical literature have explored the transformers training dynamics but addressed different problem from ours, such as the effect of initialization (Makkuva et al., 2024), convergence results (Song et al., 2024; Huang et al., 2024a), sample complexity guarantees (Ildiz et al., 2024), scaling limits (Bordelon et al., 2024), and implicit regularization (Ataee Tarzanagh et al., 2023; Tarzanagh et al., 2024; Julistiono et al., 2024; Vasudeva et al., 2025; Sheen et al., 2024). Other studies considered special training regimes, such as the neural tangent kernel regime (Jang et al., 2024) and the mean-field regime (Kim & Suzuki, 2024). A few works focused on vision transformers (Jelassi et al., 2022; Jiang et al., 2024; Huang et al., 2025b). In contrast, our work focuses on the training dynamics and the development of ICL abilities over time.

It is recognized that transformers can perform ICL, whereas it is an open question whether fully-connected networks can perform ICL (Lee et al., 2024; Boix-Adserà et al., 2024; Tong & Pehlevan, 2025). In Section 3.1, we revealed an equivalence between linear attention and a fully-connected linear network with cubic feature input, which is an instance of a fully-connected network performing ICL. Furthermore, we demonstrate that fully-connected networks may perform ICL more comparably to attention models when provided with polynomial features instead of the original sequence. This may explain why Boix-Adserà et al. (2024, Figure 25) observed that fully-connected networks fail to learn ICL with the original sequence as input, but succeed when the input is augmented with $\boldsymbol{X}\boldsymbol{X}^\top$.

C. Additional Preliminaries

C.1. Data Statistics

Recall that we use $\boldsymbol{\beta}$ to denote the in-context correlation between $\boldsymbol{x}_n$ and y_n in a sequence $\boldsymbol{X}$, as defined in Equation (4). We additionally denote the in-context covariance of $\boldsymbol{x}_n$ in a sequence as $\hat{\boldsymbol{\Lambda}}$

$$\hat{\boldsymbol{\Lambda}} \equiv \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^\top. \tag{27}$$

We can thus write $\boldsymbol{X}\boldsymbol{X}^\top/N$ as a block matrix

$$\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^\top = \begin{bmatrix} \frac{1}{N}\left(\boldsymbol{x}_q\boldsymbol{x}_q^\top + \sum_n \boldsymbol{x}_n\boldsymbol{x}_n^\top\right) & \frac{1}{N}\sum_n \boldsymbol{x}_n y_n \\ \frac{1}{N}\sum_n y_n \boldsymbol{x}^\top & \frac{1}{N}\sum_n y_n^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{N}\boldsymbol{x}_q\boldsymbol{x}_q^\top + \hat{\boldsymbol{\Lambda}} & \boldsymbol{\beta} \\ \boldsymbol{\beta}^\top & \boldsymbol{w}^\top\hat{\boldsymbol{\Lambda}}\boldsymbol{w} \end{bmatrix}. \tag{28}$$

Due to the definition of the in-context linear regression task, we have that

$$\boldsymbol{\beta} = \hat{\boldsymbol{\Lambda}}\boldsymbol{w}. \tag{29}$$

We will need a statistic, $\mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right)$. Let $p(N)$ denote the distribution of context lengths, and recall that $\boldsymbol{x}_n \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Lambda})$. We obtain:

$$\begin{aligned}
\mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right) &\equiv \mathbb{E}\left(\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n\boldsymbol{x}_n^\top\right)^2 \\
&= \mathbb{E}\left(\frac{N^2-N}{N^2}\sum_{n\neq n'}\boldsymbol{x}_n\boldsymbol{x}_n^\top\boldsymbol{x}_{n'}\boldsymbol{x}_{n'}^\top + \frac{N}{N^2}\sum_{n=1}^{N}\boldsymbol{x}_n\boldsymbol{x}_n^\top\boldsymbol{x}_n\boldsymbol{x}_n^\top\right) \\
&= \mathbb{E}_N\left(\frac{N-1}{N}\right)\mathbb{E}_{\boldsymbol{x}}\left(\boldsymbol{x}_n\boldsymbol{x}_n^\top\right)\mathbb{E}_{\boldsymbol{x}}\left(\boldsymbol{x}_{n'}\boldsymbol{x}_{n'}^\top\right) + \mathbb{E}_N\left(\frac{1}{N}\right)\mathbb{E}_{\boldsymbol{x}}\left(\boldsymbol{x}_n\boldsymbol{x}_n^\top\boldsymbol{x}_n\boldsymbol{x}_n^\top\right) \\
&= \left(1-\mathbb{E}\left(\frac{1}{N}\right)\right)\boldsymbol{\Lambda}^2 + \mathbb{E}\left(\frac{1}{N}\right)\left(2\boldsymbol{\Lambda}^2 + \mathtt{tr}(\boldsymbol{\Lambda})\boldsymbol{\Lambda}\right) \\
&= \boldsymbol{\Lambda}^2 + \mathbb{E}\left(\frac{1}{N}\right)\left(\boldsymbol{\Lambda} + \mathtt{tr}(\boldsymbol{\Lambda})\boldsymbol{I}\right)\boldsymbol{\Lambda}.
\end{aligned} \tag{30}$$

For our main results, we use a fixed context length, that is $p(N)$ is a point mass distribution and $\mathbb{E}(1/N) = 1/N$. In this case, Equation (30) simplifies to

$$\mathbb{E}\left(\hat{\mathbf{\Lambda}}^2\right) = \mathbf{\Lambda}^2 + \frac{\mathbf{\Lambda} + \mathtt{tr}(\mathbf{\Lambda})\boldsymbol{I}}{N}\mathbf{\Lambda}. \tag{31}$$

We note that the eigenvectors of $\mathbb{E}\left(\hat{\mathbf{\Lambda}}^2\right)$ are the same as those of $\mathbf{\Lambda}$, which are $\boldsymbol{e}_1, \cdots, \boldsymbol{e}_D$,

$$\mathbb{E}\left(\hat{\mathbf{\Lambda}}^2\right)\boldsymbol{e}_d = \left(1 + \frac{1}{N}\right)\mathbf{\Lambda}^2\boldsymbol{e}_d + \frac{\mathtt{tr}(\mathbf{\Lambda})}{N}\mathbf{\Lambda}\boldsymbol{e}_d = \left[\left(1 + \frac{1}{N}\right)\lambda_d^2 + \frac{\mathtt{tr}(\mathbf{\Lambda})}{N}\lambda_d\right]\boldsymbol{e}_d.$$

We denote the eigenvalues of $\mathbb{E}(\hat{\mathbf{\Lambda}}^2)$ corresponding to eigenvectors $\boldsymbol{e}_1, \cdots, \boldsymbol{e}_D$ as $a_1, \cdots, a_D$. These eigenvalues are given by

$$a_d = \left[\left(1 + \frac{1}{N}\right)\lambda_d^2 + \frac{\mathtt{tr}(\mathbf{\Lambda})}{N}\lambda_d\right] = \lambda_d^2\left(1 + \frac{1 + \mathtt{tr}(\mathbf{\Lambda})/\lambda_d}{N}\right). \tag{32}$$

The matrix $\mathbb{E}(\hat{\mathbf{\Lambda}}^2)$ can be expressed through its eigen-decomposition, which will be useful in later derivations:

$$\mathbb{E}\left(\hat{\mathbf{\Lambda}}^2\right) = \sum_{d=1}^{D} a_d \boldsymbol{e}_d \boldsymbol{e}_d^\top. \tag{33}$$

C.2. Initialization

For linear attention with merged key and query, we initialize the entries of the value and the merged key-query weights as

$$v_i \sim \mathcal{N}(0, w_{\text{init}}^2/H), \quad U_i^{d,d'} \sim \mathcal{N}(0, w_{\text{init}}^2/HD^2). \tag{34}$$

At initialization, the following ℓ^2 norms are

$$\sqrt{\sum_{i=1}^{H} v_i^2}, \sqrt{\sum_{i=1}^{H} \|\boldsymbol{U}_i\|^2} \sim O(w_{\text{init}}). \tag{35}$$

For linear attention with separate rank-R key and query, we initialize the entries of the value, key, and query weights as

$$v_i \sim \mathcal{N}(0, w_{\text{init}}^2/H), \quad k_{i,r}^d \sim \mathcal{N}(0, w_{\text{init}}^2/HRD), \quad q_{i,r}^d \sim \mathcal{N}(0, w_{\text{init}}^2/HRD). \tag{36}$$

At initialization, the following ℓ^2 norms are

$$\sqrt{\sum_{i=1}^{H} v_i^2}, \sqrt{\sum_{i=1}^{H}\sum_{r=1}^{R} \|\boldsymbol{k}_{i,r}\|^2}, \sqrt{\sum_{i=1}^{H}\sum_{r=1}^{R} \|\boldsymbol{q}_{i,r}\|^2} \sim O(w_{\text{init}}). \tag{37}$$

C.3. Kronecker Product

The Kronecker product, denoted as $\otimes$, is defined for two matrices of arbitrary sizes. The Kronecker product of the matrix $\boldsymbol{A} \in \mathbb{R}^{p\times q}$ and the matrix $\boldsymbol{B} \in \mathbb{R}^{r\times s}$ is a block matrix of shape $pr \times qs$

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11} & \cdots & a_{1q} \\ \vdots & \ddots & \vdots \\ a_{p1} & \cdots & a_{pq} \end{bmatrix} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11}\boldsymbol{B} & \cdots & a_{1q}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{p1}\boldsymbol{B} & \cdots & a_{pq}\boldsymbol{B} \end{bmatrix}.$$

Based on the definition, it holds for any pair of column vectors $\boldsymbol{a}$ and $\boldsymbol{b}$

$$\boldsymbol{a} \otimes \boldsymbol{b} = \mathtt{vec}(\boldsymbol{b}\boldsymbol{a}^\top). \tag{38}$$

We quote some properties of the Kronecker product to be used in our derivations:

$$(c\boldsymbol{A}) \otimes \boldsymbol{B} = \boldsymbol{A} \otimes (c\boldsymbol{B}) = c(\boldsymbol{A} \otimes \boldsymbol{B}) \quad \text{for any scalar } c, \tag{39a}$$

$$(\boldsymbol{A} \otimes \boldsymbol{B})^\top = \boldsymbol{A}^\top \otimes \boldsymbol{B}^\top \quad \text{for any matrices } \boldsymbol{A}, \boldsymbol{B}, \tag{39b}$$

$$(\boldsymbol{A} \otimes \boldsymbol{B})^{-1} = \boldsymbol{A}^{-1} \otimes \boldsymbol{B}^{-1} \quad \text{for invertible matrices } \boldsymbol{A}, \boldsymbol{B}, \tag{39c}$$

$$(\boldsymbol{A} \otimes \boldsymbol{B})(\boldsymbol{C} \otimes \boldsymbol{D}) = (\boldsymbol{A}\boldsymbol{C}) \otimes (\boldsymbol{B}\boldsymbol{D}) \quad \text{for compatible matrices } \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}, \tag{39d}$$

$$(\boldsymbol{B}^\top \otimes \boldsymbol{A})\mathtt{vec}(\boldsymbol{M}) = \mathtt{vec}(\boldsymbol{A}\boldsymbol{M}\boldsymbol{B}) \quad \text{for compatible matrices } \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{M}. \tag{39e}$$

D. Linear Attention with Merged Key and Query

D.1. Justification for Zero Blocks Assumption

We prove our claim in Section 2.3 that $\boldsymbol{v}_i$ and $\boldsymbol{u}_i$ remain zero throughout training if their initialization is zero.

Proof. The bottom right entry of $\mathsf{ATTN}_{\mathrm{M}}(\boldsymbol{X})$ is given by

$$\begin{aligned}
\hat{y}_q \equiv \mathsf{ATTN}_{\mathrm{M}}(\boldsymbol{X})_{D+1,N+1} &= \sum_{i=1}^{H} \begin{bmatrix} \boldsymbol{v}_i^\top & v_i \end{bmatrix} \begin{bmatrix} \frac{1}{N}\left(\boldsymbol{x}_q\boldsymbol{x}_q^\top + \sum_n \boldsymbol{x}_n\boldsymbol{x}_n^\top\right) & \frac{1}{N}\sum_n \boldsymbol{x}_n y_n \\ \frac{1}{N}\sum_n y_n \boldsymbol{x}_n^\top & \frac{1}{N}\sum_n y_n^2 \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_i \\ \boldsymbol{u}_i^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_q \\ 0 \end{bmatrix} \\
&= \sum_{i=1}^{H} \left(\boldsymbol{v}_i^\top \left(\hat{\boldsymbol{\Lambda}} + \frac{1}{N}\boldsymbol{x}_q\boldsymbol{x}_q^\top \right) \boldsymbol{U}_i + v_i \boldsymbol{\beta}^\top \boldsymbol{U}_i + \boldsymbol{v}_i^\top \boldsymbol{\beta} \boldsymbol{u}_i^\top + v_i \boldsymbol{w}^\top \hat{\boldsymbol{\Lambda}} \boldsymbol{w} \boldsymbol{u}_i^\top \right) \boldsymbol{x}_q.
\end{aligned}$$

If we initialize $\boldsymbol{v}_i, \boldsymbol{u}_i = \boldsymbol{0}$, $\hat{y}_q$ is

$$\hat{y}_q = \sum_{i=1}^{H} v_i \boldsymbol{\beta}^\top \boldsymbol{U}_i \boldsymbol{x}_q = \boldsymbol{w}^\top \hat{\boldsymbol{\Lambda}} \sum_{i=1}^{H} v_i \boldsymbol{U}_i \boldsymbol{x}_q.$$

We now calculate the gradient updates of $\boldsymbol{v}_i, \boldsymbol{u}_i$ and prove their gradients are zero if their initialization is zero. The gradient update of $\boldsymbol{v}_i$ contains $\mathbb{E}(\boldsymbol{w})$, which is zero. Specifically, we have, from Equation (5),

$$\begin{aligned}
\tau \dot{\boldsymbol{v}}_i &= \mathbb{E}\left[(y_q - \hat{y}_q) \left(\left(\hat{\boldsymbol{\Lambda}} + \frac{1}{N}\boldsymbol{x}_q\boldsymbol{x}_q^\top \right) \boldsymbol{U}_i + \boldsymbol{\beta}\boldsymbol{u}_i^\top \right) \boldsymbol{x}_q \right] \\
&= \mathbb{E}\left[\left(\boldsymbol{w}^\top \boldsymbol{x}_q - \boldsymbol{w}^\top \hat{\boldsymbol{\Lambda}} \sum_{i=1}^{H} v_i \boldsymbol{U}_i \boldsymbol{x}_q \right) \left(\hat{\boldsymbol{\Lambda}} + \frac{1}{N}\boldsymbol{x}_q\boldsymbol{x}_q^\top \right) \boldsymbol{U}_i \boldsymbol{x}_q \right] \\
&= \mathbb{E}_{\boldsymbol{w}}(\boldsymbol{w})^\top \mathbb{E}\left[\left(\boldsymbol{x}_q - \hat{\boldsymbol{\Lambda}} \sum_{i=1}^{H} v_i \boldsymbol{U}_i \boldsymbol{x}_q \right) \left(\hat{\boldsymbol{\Lambda}} + \frac{1}{N}\boldsymbol{x}_q\boldsymbol{x}_q^\top \right) \boldsymbol{U}_i \boldsymbol{x}_q \right] \\
&= \boldsymbol{0}.
\end{aligned} \tag{40}$$

Note that we separated the expectation of $\boldsymbol{w}$ because of the independence between $\boldsymbol{w}$ and all $\boldsymbol{x}$ tokens.

The gradient update of $\boldsymbol{v}_i$ contains $\mathbb{E}_{\boldsymbol{w}}\left(\boldsymbol{w}^\top \hat{\boldsymbol{\Lambda}} \boldsymbol{w} \boldsymbol{w}^\top\right)$, whose entries are linear combinations of third moments of the zero-mean normal random variable $\boldsymbol{w}$, and are thus zero. Specifically, we have

$$\begin{aligned}
\tau \dot{\boldsymbol{u}}_i &= \mathbb{E}\left[\left(\boldsymbol{v}_i^\top \boldsymbol{\beta} + v_i \boldsymbol{w}^\top \hat{\boldsymbol{\Lambda}} \boldsymbol{w} \right) (y_q - \hat{y}_q) \boldsymbol{x}_q \right] \\
&= \mathbb{E}\left[v_i \boldsymbol{w}^\top \hat{\boldsymbol{\Lambda}} \boldsymbol{w} \left(\boldsymbol{w}^\top \boldsymbol{x}_q - \boldsymbol{w}^\top \hat{\boldsymbol{\Lambda}} \sum_{i=1}^{H} v_i \boldsymbol{U}_i \boldsymbol{x}_q \right) \boldsymbol{x}_q \right] \\
&= \mathbb{E}_{\boldsymbol{w}}\left(\boldsymbol{w}^\top \hat{\boldsymbol{\Lambda}} \boldsymbol{w} \boldsymbol{w}^\top \right) \mathbb{E}\left[v_i \left(\boldsymbol{x}_q - \hat{\boldsymbol{\Lambda}} \sum_{i=1}^{H} v_i \boldsymbol{U}_i \boldsymbol{x}_q \right) \boldsymbol{x}_q \right] \\
&= \boldsymbol{0}.
\end{aligned} \tag{41}$$

■

D.2. Gradient Flow Equations

We here derive the gradient flow dynamics for linear attention with merged key and query given in Equation (8).

For linear attention with merged key and query, the prediction for the query output can be written as $\hat{y}_q = \boldsymbol{w}_2^\top \boldsymbol{W}_1 \boldsymbol{z}$ due to Equation (6). Based on the gradient flow training rule in Equation (5), the gradient flow dynamics is

$$\tau \dot{\boldsymbol{W}}_1 = \mathbb{E}\left[\boldsymbol{w}_2 \left(y_q - \boldsymbol{w}_2^\top \boldsymbol{W}_1 \boldsymbol{z}\right) \boldsymbol{z}^\top\right] = \boldsymbol{w}_2 \left(\mathbb{E}\left(y_q \boldsymbol{z}^\top\right) - \boldsymbol{w}_2^\top \boldsymbol{W}_1 \mathbb{E}\left(\boldsymbol{z}\boldsymbol{z}^\top\right)\right),$$
$$\tau \dot{\boldsymbol{w}}_2 = \mathbb{E}\left[\boldsymbol{W}_1 \left(y_q - \boldsymbol{w}_2^\top \boldsymbol{W}_1 \boldsymbol{z}\right) \boldsymbol{z}\right] = \boldsymbol{W}_1 \left(\mathbb{E}\left(y_q \boldsymbol{z}^\top\right) - \boldsymbol{w}_2^\top \boldsymbol{W}_1 \mathbb{E}\left(\boldsymbol{z}\boldsymbol{z}^\top\right)\right)^\top,$$

which was introduced in Equation (8) in the main text.

D.3. Fixed Points

To find the fixed points, we set the gradients in Equation (8) to zero

$$\tau \dot{\boldsymbol{W}}_1 = \boldsymbol{w}_2 \left(\mathbb{E}\left(y_q \boldsymbol{z}^\top\right) - \boldsymbol{w}_2^\top \boldsymbol{W}_1 \mathbb{E}\left(\boldsymbol{z}\boldsymbol{z}^\top\right)\right) \overset{\text{set}}{=} \boldsymbol{0},$$
$$\tau \dot{\boldsymbol{w}}_2 = \boldsymbol{W}_1 \left(\mathbb{E}\left(y_q \boldsymbol{z}^\top\right) - \boldsymbol{w}_2^\top \boldsymbol{W}_1 \mathbb{E}\left(\boldsymbol{z}\boldsymbol{z}^\top\right)\right)^\top \overset{\text{set}}{=} \boldsymbol{0},$$

which yield the two manifolds of fixed points introduced in Equation (9) in the main text:

$$\boldsymbol{w}_2 = \boldsymbol{0},\ \boldsymbol{W}_1 = \boldsymbol{0} \quad \Rightarrow \quad \mathcal{M}_0 = \{\boldsymbol{w}_2 = \boldsymbol{0}, \boldsymbol{W}_1 = \boldsymbol{0}\},$$
$$\mathbb{E}\left(y_q \boldsymbol{z}^\top\right) - \boldsymbol{w}_2^\top \boldsymbol{W}_1 \mathbb{E}\left(\boldsymbol{z}\boldsymbol{z}^\top\right) = \boldsymbol{0} \quad \Rightarrow \quad \mathcal{M}_* = \left\{\boldsymbol{w}_2, \boldsymbol{W}_1 \middle| \boldsymbol{w}_2^\top \boldsymbol{W}_1 = \mathbb{E}\left(y_q \boldsymbol{z}^\top\right) \mathbb{E}\left(\boldsymbol{z}\boldsymbol{z}^\top\right)^{-1}\right\}.$$

D.4. Duality of the Global Minimum Solution

We here prove the second equality in Equation (11), that is

$$\mathbb{E}\left(y_q \boldsymbol{z}^\top\right) \mathbb{E}\left(\boldsymbol{z}\boldsymbol{z}^\top\right)^{-1} \boldsymbol{z} = \boldsymbol{\beta}^\top \left(\boldsymbol{\Lambda} + \frac{\boldsymbol{\Lambda} + \mathtt{tr}(\boldsymbol{\Lambda})\boldsymbol{I}}{N}\right)^{-1} \boldsymbol{x}_q.$$

This equality implies the intriguing duality that the linear regression solution in the cubic feature space of $\boldsymbol{z}$ is the in-context linear regression solution in the original space of the $\boldsymbol{x}_n, y_n$ token pairs for each sequence $\boldsymbol{X}$.

Proof. We first calculate the input and input-output correlations in the cubic feature space. We denote $\boldsymbol{\Lambda}_q \equiv \mathbb{E}\left(\boldsymbol{x}_q \boldsymbol{x}_q^\top\right)$. While $\boldsymbol{\Lambda}_q = \boldsymbol{\Lambda}$, this equality is not needed in this proof.

Due to the property in Equation (38), the cubic feature $\boldsymbol{z}$ can be written as

$$\boldsymbol{z} = \mathtt{vec}\left(\boldsymbol{\beta}\boldsymbol{x}_q^\top\right) = \boldsymbol{x}_q \otimes \boldsymbol{\beta}. \tag{42}$$

We substitute in $y_q = \boldsymbol{x}_q^\top \boldsymbol{w}$, $\boldsymbol{z} = \boldsymbol{x}_q \otimes \boldsymbol{\beta}$ and use the properties of the Kronecker product in Equation (39) to obtain

$$\begin{aligned}
\mathbb{E}\left(y_q \boldsymbol{z}^\top\right) &= \mathbb{E}\left[\boldsymbol{x}_q^\top \boldsymbol{w} (\boldsymbol{x}_q^\top \otimes \boldsymbol{\beta}^\top)\right] \\
&= \mathbb{E}\left(\boldsymbol{x}_q^\top \otimes \boldsymbol{x}_q^\top \boldsymbol{w}\boldsymbol{w}^\top \hat{\boldsymbol{\Lambda}}\right) \\
&= \mathbb{E}\left(\boldsymbol{x}_q^\top \otimes \boldsymbol{x}_q^\top \hat{\boldsymbol{\Lambda}}\right) \\
&= \mathbb{E}\mathtt{vec}\left(\hat{\boldsymbol{\Lambda}}\boldsymbol{x}_q \boldsymbol{x}_q^\top\right)^\top \\
&= \mathtt{vec}(\boldsymbol{\Lambda}\boldsymbol{\Lambda}_q)^\top.
\end{aligned} \tag{43}$$

Similarly, we have

$$\begin{aligned}
\mathbb{E}\left(\boldsymbol{z}\boldsymbol{z}^\top\right) &= \mathbb{E}\left[(\boldsymbol{x}_q \otimes \boldsymbol{\beta})(\boldsymbol{x}_q^\top \otimes \boldsymbol{\beta}^\top)\right] \\
&= \mathbb{E}\left[(\boldsymbol{x}_q \boldsymbol{x}_q^\top) \otimes (\boldsymbol{\beta}\boldsymbol{\beta}^\top)\right] \\
&= \mathbb{E}(\boldsymbol{x}_q \boldsymbol{x}_q^\top) \otimes \mathbb{E}(\hat{\boldsymbol{\Lambda}}\boldsymbol{w}\boldsymbol{w}^\top \hat{\boldsymbol{\Lambda}}) \\
&= \boldsymbol{\Lambda}_q \otimes \mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right).
\end{aligned} \tag{44}$$

Using Equation (39c), the inverse of $\mathbb{E}\left(\boldsymbol{z}\boldsymbol{z}^\top\right)$ is

$$\mathbb{E}\left(\boldsymbol{z}\boldsymbol{z}^\top\right)^{-1} = \boldsymbol{\Lambda}_q^{-1} \otimes \mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right)^{-1}. \tag{45}$$

Multiplying Equations (43) and (45) with $\boldsymbol{z} = \boldsymbol{x}_q \otimes \boldsymbol{\beta}$, and using Equation (39e) twice, we obtain

$$\begin{aligned}
\mathbb{E}\left(y_q\boldsymbol{z}^\top\right)\mathbb{E}\left(\boldsymbol{z}\boldsymbol{z}^\top\right)^{-1}\boldsymbol{z} &= \mathtt{vec}(\boldsymbol{\Lambda}\boldsymbol{\Lambda}_q)^\top\boldsymbol{\Lambda}_q^{-1} \otimes \mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right)^{-1}(\boldsymbol{x}_q \otimes \boldsymbol{\beta}) \\
&= \mathtt{vec}\left[\mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right)^{-1}\boldsymbol{\Lambda}\boldsymbol{\Lambda}_q\boldsymbol{\Lambda}_q^{-1}\right]^\top(\boldsymbol{x}_q \otimes \boldsymbol{\beta}) \\
&= \boldsymbol{\beta}^\top\mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right)^{-1}\boldsymbol{\Lambda}\boldsymbol{x}_q.
\end{aligned} \tag{46}$$

Substituting in $\mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right)$ obtained from Equation (31) finishes the proof

$$\mathbb{E}\left(y_q\boldsymbol{z}^\top\right)\mathbb{E}\left(\boldsymbol{z}\boldsymbol{z}^\top\right)^{-1}\boldsymbol{z} = \boldsymbol{\beta}^\top\left(\boldsymbol{\Lambda}^2 + \frac{\boldsymbol{\Lambda} + \mathtt{tr}(\boldsymbol{\Lambda})\boldsymbol{I}}{N}\boldsymbol{\Lambda}\right)^{-1}\boldsymbol{\Lambda}\boldsymbol{x}_q = \boldsymbol{\beta}^\top\left(\boldsymbol{\Lambda} + \frac{\boldsymbol{\Lambda} + \mathtt{tr}(\boldsymbol{\Lambda})\boldsymbol{I}}{N}\right)^{-1}\boldsymbol{x}_q.$$

■

D.5. Analytical Time-Course Solution for White Covariance

We include a derivation of the time-course solution of two-layer fully-connected linear network with white input covariance and vanishing initialization following (Saxe et al., 2014), and then apply it to linear attention. With vanishing initialization, the conserved quantity given in Equation (55) is exactly zero throughout learning,

$$\boldsymbol{w}_2\boldsymbol{w}_2^\top - \boldsymbol{W}_1\boldsymbol{W}_1^\top = \boldsymbol{0}.$$

Hence, there exists a unit norm vector $\boldsymbol{m}$ such that $\boldsymbol{W}_1 = \boldsymbol{w}_2\boldsymbol{m}^\top$. With the assumption of white covariance, $\mathbb{E}\left(\boldsymbol{z}\boldsymbol{z}^\top\right) = \alpha\boldsymbol{I}_{D^2}$, (Saxe et al., 2014; Atanasov et al., 2022) have shown that the unit norm vector $\boldsymbol{m}$ is parallel with the correlation between y_q and $\boldsymbol{z}$ throughout training, that is

$$\boldsymbol{W}_1 = \boldsymbol{w}_2\boldsymbol{m}^\top, \quad \text{where } \boldsymbol{m} = \frac{\mathbb{E}(y_q\boldsymbol{z})}{\|\mathbb{E}(y_q\boldsymbol{z})\|}. \tag{47}$$

We substitute Equation (47) and the white covariance assumption, $\mathbb{E}\left(\boldsymbol{z}\boldsymbol{z}^\top\right) = \alpha\boldsymbol{I}_{D^2}$, into the gradient flow dynamics given in Equation (8) and obtain

$$\tau\dot{\boldsymbol{w}}_2 = \boldsymbol{w}_2\boldsymbol{m}^\top\left(\mathbb{E}(y_q\boldsymbol{z}) - \alpha\boldsymbol{w}_2^\top\boldsymbol{w}_2\boldsymbol{m}\right) = \boldsymbol{w}_2\left(\gamma - \alpha\boldsymbol{w}_2^\top\boldsymbol{w}_2\right), \quad \text{where } \gamma \equiv \|\mathbb{E}(y_q\boldsymbol{z})\|.$$

Notice that the square of the ℓ^2 norm of $\boldsymbol{w}_2$ follows a solvable ordinary differential equation. Let $s = \boldsymbol{w}_2^\top\boldsymbol{w}_2$. The dynamics of $s(t)$ is

$$\tau\dot{s} = 2\boldsymbol{w}_2^\top\tau\dot{\boldsymbol{w}}_2 = 2\boldsymbol{w}_2^\top\boldsymbol{w}_2\left(\gamma - \alpha\boldsymbol{w}_2^\top\boldsymbol{w}_2\right) = 2s(\gamma - \alpha s). \tag{48}$$

We can solve this differential equation by separating variables and integrating both sides,

$$\int_{s(0)}^{s(t)}\frac{1}{s(\gamma - \alpha s)}\mathrm{d}s = \int_0^t\frac{2}{\tau}\mathrm{d}t \quad \Rightarrow \quad \frac{1}{\gamma}\ln\frac{s(t)(\gamma - \alpha s(0))}{s(0)(\gamma - \alpha s(t))} = \frac{2}{\tau}t.$$

The solution of $s(t)$ is given by

$$s(t) = \frac{\gamma e^{2\gamma\frac{t}{\tau}}}{\alpha\left(e^{2\gamma\frac{t}{\tau}} - 1\right) + \frac{\gamma}{s(0)}}.$$

The time-course of the total weights is given by

$$\boldsymbol{w}_2^\top \boldsymbol{W}_1 = s(t)\boldsymbol{m}^\top = s(t)\frac{\mathbb{E}(y_q\boldsymbol{z})^\top}{\|\mathbb{E}(y_q\boldsymbol{z})^\top\|}. \tag{49}$$

We now apply this solution to linear attention. If the input token covariance is identity, $\boldsymbol{\Lambda} = \boldsymbol{I}_D$, we calculate the input and input-output correlations in the cubic feature space according to Equations (43) and (44) and get

$$\mathbb{E}\left(y_q\boldsymbol{z}^\top\right) = \mathtt{vec}(\boldsymbol{I}_D)^\top,$$
$$\mathbb{E}\left(\boldsymbol{z}\boldsymbol{z}^\top\right) = \boldsymbol{I}_D \otimes \left(1+\frac{1+D}{N}\right)\boldsymbol{I}_D = \left(1+\frac{1+D}{N}\right)\boldsymbol{I}_{D^2}.$$

The parameters in the dynamics of the equivalent two-layer fully-connected linear network are

$$\alpha = 1+\frac{1+D}{N},\ \gamma = \|\mathtt{vec}(\boldsymbol{I}_D)\| = \sqrt{D}. \tag{50}$$

Substituting Equation (50) into Equation (49), we obtain

$$\boldsymbol{w}^\top \boldsymbol{W}_1(t) = s(t)\frac{\mathtt{vec}(\boldsymbol{I}_D)^\top}{\sqrt{D}}, \quad \text{where } s(t) = \frac{\sqrt{D}e^{2\sqrt{D}\frac{t}{\tau}}}{\left(1+\frac{1+D}{N}\right)\left(e^{2\sqrt{D}\frac{t}{\tau}}-1\right)+\frac{\sqrt{D}}{s(0)}}.$$

Due to the equivalence between linear attention and the two-layer fully-connected linear network given in Equation (6), we obtain

$$\mathsf{ATTN}_{\mathrm{M}}(\boldsymbol{X};t)_{D+1,N+1} = \boldsymbol{w}_2^\top \boldsymbol{W}_1 \boldsymbol{z} = s(t)\frac{\mathtt{vec}(\boldsymbol{I}_D)^\top}{\sqrt{D}}\boldsymbol{x}_q \otimes \boldsymbol{\beta} = \frac{1}{\sqrt{D}}s(t)\boldsymbol{\beta}^\top \boldsymbol{I}_D \boldsymbol{x}_q = \frac{1}{\sqrt{D}}s(t)\boldsymbol{\beta}^\top \boldsymbol{x}_q.$$

which is Equation (10) in the main text where we have rewritten $\sigma(t) = s(t)/\sqrt{D}$.

The time-course of loss can also be expressed in terms of $\sigma(t)$ as

$$\mathcal{L}(t) = \left(1-2\sigma(t)+\left(1+\frac{1+D}{N}\right)\sigma(t)^2\right)D. \tag{51}$$

D.6. Training Dynamics for General Covariance

D.6.1. Early Dynamics Predicts Duration of Plateau

For a general input covariance matrix, the full time-course solution to two-layer fully-connected linear networks is currently unavailable. Nonetheless, the training dynamics is well understood and we can analyze the early phase dynamics to estimate the duration of the loss plateau.

In the early phase of training when the loss plateaus, the weights have not moved much away from their small initialization. The training dynamics of $\boldsymbol{W}_1$ is mainly driven by the first term in Equation (8a), and similarly for $\boldsymbol{w}_2$ in Equation (8b)

$$\tau\dot{\boldsymbol{W}}_1 = \boldsymbol{w}_2\left(\mathbb{E}\left(y_q\boldsymbol{z}^\top\right) - \boldsymbol{w}_2^\top \boldsymbol{W}_1 \mathbb{E}\left(\boldsymbol{z}\boldsymbol{z}^\top\right)\right) = \boldsymbol{w}_2\mathbb{E}\left(y_q\boldsymbol{z}^\top\right) + O(w_{\text{init}}^3),$$
$$\tau\dot{\boldsymbol{w}}_2 = \boldsymbol{W}_1\left(\mathbb{E}\left(y_q\boldsymbol{z}^\top\right) - \boldsymbol{w}_2^\top \boldsymbol{W}_1 \mathbb{E}\left(\boldsymbol{z}\boldsymbol{z}^\top\right)\right)^\top = \boldsymbol{W}_1\mathbb{E}\left(y_q\boldsymbol{z}\right) + O(w_{\text{init}}^3).$$

Thus, the early training dynamics is well approximated by the linear dynamical system

$$\tau\dot{\boldsymbol{W}}_1 = \boldsymbol{w}_2\mathbb{E}\left(y_q\boldsymbol{z}^\top\right), \quad \tau\dot{\boldsymbol{w}}_2 = \boldsymbol{W}_1\mathbb{E}\left(y_q\boldsymbol{z}\right).$$

In the case of nonwhite covariance, the change of variable in Equation (47) is valid in the early phase of training but no longer valid when the loss starts to decrease appreciably (Atanasov et al., 2022). For the early training dynamics, we apply the change of variable in Equation (47) and obtain

$$\tau\dot{\boldsymbol{w}}_2 = \boldsymbol{w}_2\boldsymbol{m}^\top\mathbb{E}\left(y_q\boldsymbol{z}\right) = \gamma\boldsymbol{w}_2.$$

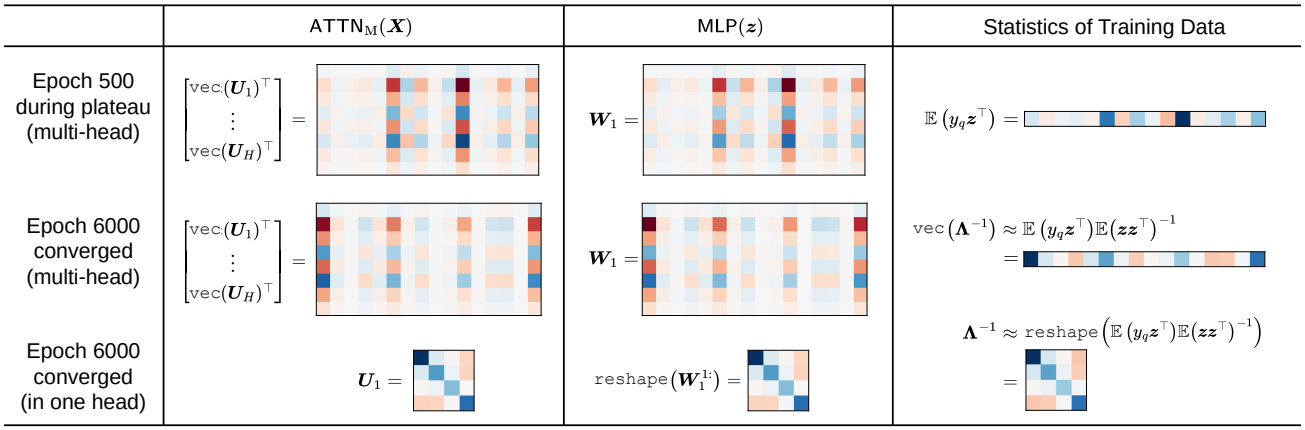


Figure 9. The dynamics of weights in multi-head linear attention with merged key and query can be predicted with statistics of the training dataset. We plot the weights at different times in training, corresponding to the loss trajectories in Figure 1 (right). The weights in linear attention (first column) stay close to the weights in the fully-connected linear network (second column) throughout training. During the initial plateau, the vectorized key-query weights in attention $\texttt{vec}(\boldsymbol{U}_1), \cdots, \texttt{vec}(\boldsymbol{U}_i)$ and the first-layer weight in the fully-connected network $\boldsymbol{W}_1$ are rank-one and align with the correlation between the output and the cubic feature input $\mathbb{E}\left(y_q \boldsymbol{z}^\top\right)$ (top row). At convergence, $\texttt{vec}(\boldsymbol{U}_1), \cdots, \texttt{vec}(\boldsymbol{U}_i)$ in attention and $\boldsymbol{W}_1$ in the fully-connected linear network are rank-one and align with the linear regression solution in the cubic feature space $\mathbb{E}\left(y_q \boldsymbol{z}^\top\right) \mathbb{E}\left(\boldsymbol{z}\boldsymbol{z}^\top\right)^{-1}$ (middle row), which is also the in-context linear regression solution in the original token space $\boldsymbol{\Lambda}^{-1}$ (bottom row) as described by Equation (11). The approximate equality in the third column is exact when the sequence length $N \to \infty$.

Recall that $s = \boldsymbol{w}_2^\top \boldsymbol{w}_2$. The early phase dynamics of $s(t)$ is approximately

$$\tau \dot{s} = 2\gamma s.$$

We solve the differential equation and obtain

$$t = \frac{\tau}{2\gamma}(\ln s(t) - \ln s(0)).$$

Due to small initialization, $\ln s(t)$ at the end of the plateau is much smaller compared to $-\ln s(0)$. Hence, the duration of the initial plateau of loss, t_{plateau}, is

$$t_{\text{plateau}} \approx \frac{\tau}{2\gamma} \ln \frac{1}{s(0)}. \tag{52}$$

Here, the scalar γ is

$$\begin{aligned}
\gamma \equiv \|\mathbb{E}(y_q \boldsymbol{z})\| = \left\|\mathbb{E}\left(\boldsymbol{w}^\top \boldsymbol{x}_q \texttt{vec}(\boldsymbol{\beta}\boldsymbol{x}_q^\top)\right)\right\| &= \left\|\mathbb{E}\left(\boldsymbol{w}^\top \boldsymbol{x}_q \boldsymbol{\beta} \boldsymbol{x}_q^\top\right)\right\|_{\text{F}} \\
&= \left\|\mathbb{E}\left(\hat{\boldsymbol{\Lambda}} \boldsymbol{w}\boldsymbol{w}^\top \boldsymbol{x}_q \boldsymbol{x}_q^\top\right)\right\|_{\text{F}} \\
&= \left\|\mathbb{E}\left(\hat{\boldsymbol{\Lambda}}\right) \mathbb{E}_{\boldsymbol{w}}\left(\boldsymbol{w}\boldsymbol{w}^\top\right) \mathbb{E}_{\boldsymbol{x}_q}\left(\boldsymbol{x}_q \boldsymbol{x}_q^\top\right)\right\|_{\text{F}} \\
&= \left\|\boldsymbol{\Lambda}^2\right\|_{\text{F}}
\end{aligned} \tag{53}$$

Substituting Equation (53) into Equation (52), we obtain the approximate duration of the loss plateau

$$t_{\text{plateau}} \approx \frac{\tau}{2\left\|\boldsymbol{\Lambda}^2\right\|_{\text{F}}} \ln \frac{1}{s(0)} \approx \frac{\tau}{\left\|\boldsymbol{\Lambda}^2\right\|_{\text{F}}} \ln \frac{1}{w_{\text{init}}}, \tag{54}$$

where we used the definition $s(0) = \|\boldsymbol{w}_2(0)\|^2 = w_{\text{init}}^2$.

D.6.2. WEIGHTS DYNAMICS

For a white input covariance, the training dynamics reduces to a scalar ordinary differential equation about $s(t)$ given in Equation (48). For a general input covariance, the vector $\boldsymbol{m}$ in the change of variable defined in Equation (47) rotates during training. As shown in the top row of Figure 9, during the initial loss plateau, the rows of the first-layer weight align with the input-output correlation $\mathbb{E}\left(y_q \boldsymbol{z}^\top\right)$ but do not change appreciably in scale (Atanasov et al., 2022). Later, when the loss decreases rapidly, the first-layer weight grows in scale and rotates to align with the global minimum solution, $\mathbb{E}\left(y_q \boldsymbol{z}^\top\right) \mathbb{E}\left(\boldsymbol{z}\boldsymbol{z}^\top\right)^{-1}$. The alignment and rotation behaviors apply to the rows of the first-layer weight in the fully-connected network, corresponding to the merged key-query weights in the different heads in linear attention, as shown in Figure 9.

D.7. Conservation Law: All Heads Are Parallel

The weights in a fully-connected linear network are known to obey a conservation law during training (Fukumizu, 1998; Saxe et al., 2014; Du et al., 2018; Ji & Telgarsky, 2019)

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{w}_2 \boldsymbol{w}_2^\top - \boldsymbol{W}_1 \boldsymbol{W}_1^\top\right) = \boldsymbol{0}, \tag{55}$$

which follows directly from the gradient flow dynamics in Equation (8). Under small initialization, the quantity $\boldsymbol{w}_2 \boldsymbol{w}_2^\top - \boldsymbol{W}_1 \boldsymbol{W}_1^\top \approx \boldsymbol{0}$ is small at initialization and remains small throughout training. Since the vector $\boldsymbol{w}_2$ is rank-one, the conservation law forces $\boldsymbol{W}_1$ to also be approximately rank-one, which means that the rows of $\boldsymbol{W}_1$ are approximately parallel. Since each row of $\boldsymbol{W}_1$ is the vectorized merged key-query matrix of a head, $\mathtt{vec}(\boldsymbol{U}_i)$, a rank-one $\boldsymbol{W}_1$ implies that the key-query weight matrices of all heads are parallel, differing only in scale. As shown in Figure 9, simulations indeed show that the key-query weights in different heads are parallel.

E. Linear Attention with Separate Rank-One Key and Query

E.1. Justification for Zero Blocks Assumption

This is a special case of linear attention with separate rank-R key and query. The proof for the more general rank-R case can be found in Appendix F.1.

E.2. Gradient Flow Equations

We here derive the gradient flow dynamics for linear attention with separate rank-one key and query introduced in Equation (14).

Based on the gradient flow training rule in Equation (5), the gradient flow dynamics for the value, key, and query weights in the i-th head are

$$\tau \dot{v}_i = \boldsymbol{k}_i^\top \mathbb{E}\left(\boldsymbol{\beta}(y_q - \hat{y}_q)\boldsymbol{x}_q^\top\right) \boldsymbol{q}_i, \tag{56a}$$

$$\tau \dot{\boldsymbol{k}}_i = v_i \mathbb{E}\left(\boldsymbol{\beta}(y_q - \hat{y}_q)\boldsymbol{x}_q^\top\right) \boldsymbol{q}_i, \tag{56b}$$

$$\tau \dot{\boldsymbol{q}}_i = v_i \mathbb{E}\left(\boldsymbol{x}_q(y_q - \hat{y}_q)\boldsymbol{\beta}^\top\right) \boldsymbol{k}_i. \tag{56c}$$

We calculate the common term in Equation (56), that is

$$\begin{aligned}
\mathbb{E}\left(\boldsymbol{\beta}(y_q - \hat{y}_q)\boldsymbol{x}_q^\top\right) &= \mathbb{E}\left[\boldsymbol{\beta}\left(\boldsymbol{w}^\top \boldsymbol{x}_q - \sum_{i=1}^H v_i \boldsymbol{\beta}^\top \boldsymbol{k}_i \boldsymbol{q}_i^\top \boldsymbol{x}_q\right) \boldsymbol{x}_q^\top\right] \\
&= \mathbb{E}\left[\hat{\boldsymbol{\Lambda}} \boldsymbol{w} \boldsymbol{w}^\top \left(\boldsymbol{I} - \sum_{i=1}^H v_i \hat{\boldsymbol{\Lambda}} \boldsymbol{k}_i \boldsymbol{q}_i^\top\right) \boldsymbol{x}_q \boldsymbol{x}_q^\top\right] \\
&= \mathbb{E}\left(\hat{\boldsymbol{\Lambda}}\right) \mathbb{E}_{\boldsymbol{w}}\left(\boldsymbol{w}\boldsymbol{w}^\top\right) \mathbb{E}_{\boldsymbol{x}_q}\left(\boldsymbol{x}_q \boldsymbol{x}_q^\top\right) - \mathbb{E}\left(\hat{\boldsymbol{\Lambda}} \boldsymbol{w} \boldsymbol{w}^\top \hat{\boldsymbol{\Lambda}}\right) \sum_{i=1}^H v_i \boldsymbol{k}_i \boldsymbol{q}_i^\top \mathbb{E}_{\boldsymbol{x}_q}\left(\boldsymbol{x}_q \boldsymbol{x}_q^\top\right) \\
&= \boldsymbol{\Lambda}^2 - \mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right) \sum_{i=1}^H v_i \boldsymbol{k}_i \boldsymbol{q}_i^\top \boldsymbol{\Lambda}
\end{aligned} \tag{57}$$

Substituting Equation (57) into Equation (56), we arrive at the same equations as Equation (14) in the main text

$$\tau \dot{v}_i = \boldsymbol{k}_i^\top \left(\boldsymbol{\Lambda}^2 - \mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right) \sum_{i'=1}^{H} v_{i'} \boldsymbol{k}_{i'} \boldsymbol{q}_{i'}^\top \boldsymbol{\Lambda} \right) \boldsymbol{q}_i,$$
$$\tau \dot{\boldsymbol{k}}_i = v_i \left(\boldsymbol{\Lambda}^2 - \mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right) \sum_{i'=1}^{H} v_{i'} \boldsymbol{k}_{i'} \boldsymbol{q}_{i'}^\top \boldsymbol{\Lambda} \right) \boldsymbol{q}_i,$$
$$\tau \dot{\boldsymbol{q}}_i = v_i \left(\boldsymbol{\Lambda}^2 - \boldsymbol{\Lambda} \sum_{i'=1}^{H} v_{i'} \boldsymbol{k}_{i'} \boldsymbol{q}_{i'}^\top \mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right) \right) \boldsymbol{k}_i.$$

where the data statistics $\mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right)$ is calculated in Equation (31).

E.3. Fixed Points

We prove that the fixed points given in Equation (16) are valid.

Proof. When the model is at a fixed point in set $\mathcal{M}(\mathcal{S}_m)$, it satisfies Equation (18). Equation (18) can be rewritten using a_d (defined in Equation (32)) as

$$\sum_{i=1}^{H} v_i \boldsymbol{k}_i \boldsymbol{q}_i^\top = \sum_{d \in \mathcal{S}_m} \frac{\lambda_d}{a_d} \boldsymbol{e}_d \boldsymbol{e}_d^\top. \tag{59}$$

Using Equations (33) and (59), we can simplify a common term in the gradient descent dynamics in Equation (14) to

$$\begin{aligned}
\boldsymbol{\Lambda}^2 - \mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right) \sum_{i=1}^{H} v_i \boldsymbol{k}_i \boldsymbol{q}_i^\top \boldsymbol{\Lambda} &= \sum_{d=1}^{D} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top - \sum_{d'=1}^{D} a_{d'} \boldsymbol{e}_{d'} \boldsymbol{e}_{d'}^\top \sum_{d \in \mathcal{S}_m} \frac{\lambda_d}{a_d} \boldsymbol{e}_d \boldsymbol{e}_d^\top \boldsymbol{\Lambda} \\
&= \sum_{d=1}^{D} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top - \sum_{d \in \mathcal{S}_m} \lambda_d \boldsymbol{e}_d \boldsymbol{e}_d^\top \boldsymbol{\Lambda} \\
&= \sum_{d \notin \mathcal{S}_m} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top.
\end{aligned} \tag{60}$$

Substituting Equation (60) into Equation (14), we obtain the dynamics when the model is at a fixed point in $\mathcal{M}(\mathcal{S}_m)$

$$\tau \dot{v}_i = \boldsymbol{k}_i^\top \left(\sum_{d \notin \mathcal{S}_m} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top \right) \boldsymbol{q}_i, \tag{61a}$$
$$\tau \dot{\boldsymbol{k}}_i = v_i \left(\sum_{d \notin \mathcal{S}_m} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top \right) \boldsymbol{q}_i, \tag{61b}$$
$$\tau \dot{\boldsymbol{q}}_i = v_i \left(\sum_{d \notin \mathcal{S}_m} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top \right) \boldsymbol{k}_i. \tag{61c}$$

(i) For the heads with a nonzero value weight, $v_i \neq 0$, the key and query weights at a fixed point satisfy condition (C2) for Equation (16), that is the key and query weights lie in the span of $\{\boldsymbol{e}_d\}_{d \in \mathcal{S}_m}$ and thus can be written as

$$\boldsymbol{k}_i = \sum_{d \in \mathcal{S}_m} b_d \boldsymbol{e}_d, \quad b_d \in \mathbb{R}, \tag{62a}$$
$$\boldsymbol{q}_i = \sum_{d \in \mathcal{S}_m} c_d \boldsymbol{e}_d, \quad c_d \in \mathbb{R}. \tag{62b}$$

Substituting Equation (62) into the gradient flow dynamics given in Equation (61), we obtain

$$\tau \dot{v}_i = \boldsymbol{k}_i^\top \left(\sum_{d \notin \mathcal{S}_m} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top \right) \sum_{d' \in \mathcal{S}_m} c_{d'} \boldsymbol{e}_{d'} = 0,$$
$$\tau \dot{\boldsymbol{k}}_i = v_i \left(\sum_{d \notin \mathcal{S}_m} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top \right) \sum_{d' \in \mathcal{S}_m} c_{d'} \boldsymbol{e}_{d'} = \boldsymbol{0},$$
$$\tau \dot{\boldsymbol{q}}_i = v_i \left(\sum_{d \notin \mathcal{S}_m} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top \right) \sum_{d' \in \mathcal{S}_m} b_{d'} \boldsymbol{e}_{d'} = \boldsymbol{0},$$

where we have used the fact that $\boldsymbol{e}_d^\top \boldsymbol{e}_{d'} = 0$ if $d \neq d'$, because eigenvectors of the covariance matrix $\boldsymbol{\Lambda}$ are orthogonal.

(ii) For the heads with a zero value weight, $v_i = 0$, the gradients of the key and query weights in Equations (61b) and (61c) contain v_i and are thus zero, $\dot{\boldsymbol{k}}_i = \boldsymbol{0}, \dot{\boldsymbol{q}}_i = \boldsymbol{0}$. Further, the key and query weights of a head with a zero value weight satisfy condition (C3) for Equation (16). Without loss of generality, suppose that $\boldsymbol{q}_i$ lies in the span of $\{\boldsymbol{e}_d\}_{d \in \mathcal{S}_m}$, that is $\boldsymbol{q}_i$ satisfies Equation (62b). Substituting Equation (62b) into the gradient of v_i given in Equation (61a), we obtain

$$\dot{v}_i = \boldsymbol{k}_i^\top \left(\sum_{d \notin \mathcal{S}_m} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top \right) \sum_{d' \in \mathcal{S}_m} c_{d'} \boldsymbol{e}_{d'} = 0,$$

where we have again used the fact that eigenvectors of $\boldsymbol{\Lambda}$ are orthogonal.

Hence, when the model has weights specified in Equation (16), the gradients of the weights are zero, meaning that the fixed points are valid. ■

E.4. Loss Value at A Fixed Point

We derive the loss when the model is at a fixed point in set $\mathcal{M}(\mathcal{S}_m)$, where the loss is given by

$$\mathcal{L}(\mathcal{M}(\mathcal{S}_m)) = \mathtt{tr}(\boldsymbol{\Lambda}) - \sum_{d \in \mathcal{S}_m} \lambda_d \left(1 + \frac{1 + \mathtt{tr}(\boldsymbol{\Lambda})/\lambda_d}{N} \right)^{-1}. \tag{63}$$

Equation (19) in the main text follows directly from Equation (63) when taking $S_m = \{1, 2, \cdots, m\}$.

Proof. We substitute Equations (33) and (59) into the mean square loss and obtain

$$\begin{aligned}
\mathcal{L}(\mathcal{M}(\mathcal{S}_m)) &= \mathbb{E}(y_q - \hat{y}_q)^2 \\
&= \mathbb{E}\left(\boldsymbol{w}^\top \boldsymbol{x}_q - \sum_{d \in \mathcal{S}_m} \frac{\lambda_d}{a_d} \boldsymbol{w}^\top \hat{\boldsymbol{\Lambda}} \boldsymbol{e}_d \boldsymbol{e}_d^\top \boldsymbol{x}_q \right)^2 \\
&= \mathbb{E}\left[\boldsymbol{x}_q^\top \left(\boldsymbol{I} - \sum_{d \in \mathcal{S}_m} \frac{\lambda_d}{a_d} \hat{\boldsymbol{\Lambda}} \boldsymbol{e}_d \boldsymbol{e}_d^\top \right) \mathbb{E}_{\boldsymbol{w}}(\boldsymbol{w}\boldsymbol{w}^\top) \left(\boldsymbol{I} - \sum_{d \in \mathcal{S}_m} \frac{\lambda_d}{a_d} \hat{\boldsymbol{\Lambda}} \boldsymbol{e}_d \boldsymbol{e}_d^\top \right) \boldsymbol{x}_q \right] \\
&= \mathbb{E}\left[\boldsymbol{x}_q^\top \left(\boldsymbol{I} - \sum_{d \in \mathcal{S}_m} \frac{\lambda_d}{a_d} \hat{\boldsymbol{\Lambda}} \boldsymbol{e}_d \boldsymbol{e}_d^\top \right) \left(\boldsymbol{I} - \sum_{d \in \mathcal{S}_m} \frac{\lambda_d}{a_d} \hat{\boldsymbol{\Lambda}} \boldsymbol{e}_d \boldsymbol{e}_d^\top \right) \boldsymbol{x}_q \right] \\
&= \mathbb{E}\left[\boldsymbol{x}_q^\top \left(\boldsymbol{I} - 2 \sum_{d \in \mathcal{S}_m} \frac{\lambda_d}{a_d} \hat{\boldsymbol{\Lambda}} \boldsymbol{e}_d \boldsymbol{e}_d^\top + \left(\sum_{d \in \mathcal{S}_m} \frac{\lambda_d}{a_d} \hat{\boldsymbol{\Lambda}} \boldsymbol{e}_d \boldsymbol{e}_d^\top \right)^2 \right) \boldsymbol{x}_q \right].
\end{aligned} \tag{64}$$

Since $\hat{\boldsymbol{\Lambda}}$ is independent of $\boldsymbol{x}_q$, we can calculate the expectation of the purple and teal terms first,

$$\begin{aligned}
\mathbb{E}\left(\sum_{d\in\mathcal{S}_m}\frac{\lambda_d}{a_d}\hat{\boldsymbol{\Lambda}}\boldsymbol{e}_d\boldsymbol{e}_d^\top\right) &= \sum_{d\in\mathcal{S}_m}\frac{\lambda_d}{a_d}\boldsymbol{\Lambda}\boldsymbol{e}_d\boldsymbol{e}_d^\top = \sum_{d\in\mathcal{S}_m}\frac{\lambda_d^2}{a_d}\boldsymbol{e}_d\boldsymbol{e}_d^\top, \\
\mathbb{E}\left[\left(\sum_{d\in\mathcal{S}_m}\frac{\lambda_d}{a_d}\hat{\boldsymbol{\Lambda}}\boldsymbol{e}_d\boldsymbol{e}_d^\top\right)^2\right] &= \mathbb{E}\left[\sum_{d\in\mathcal{S}_m}\frac{\lambda_d^2}{a_d^2}\boldsymbol{e}_d\boldsymbol{e}_d^\top\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{\Lambda}}\boldsymbol{e}_d\boldsymbol{e}_d^\top + \sum_{d,d'\in\mathcal{S}_m,d\neq d'}\frac{\lambda_d\lambda_{d'}}{a_d a_{d'}}\hat{\boldsymbol{\Lambda}}\boldsymbol{e}_d\boldsymbol{e}_d^\top\boldsymbol{e}_{d'}\boldsymbol{e}_{d'}^\top\hat{\boldsymbol{\Lambda}}\right] \\
&= \sum_{d\in\mathcal{S}_m}\frac{\lambda_d^2}{a_d^2}\boldsymbol{e}_d\boldsymbol{e}_d^\top\mathbb{E}\left(\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{\Lambda}}\right)\boldsymbol{e}_d\boldsymbol{e}_d^\top + \boldsymbol{0} \\
&= \sum_{d\in\mathcal{S}_m}\frac{\lambda_d^2}{a_d^2}\boldsymbol{e}_d\boldsymbol{e}_d^\top\sum_{d'=1}^{D}a_{d'}\boldsymbol{e}_{d'}\boldsymbol{e}_{d'}^\top\boldsymbol{e}_d\boldsymbol{e}_d^\top \\
&= \sum_{d\in\mathcal{S}_m}\frac{\lambda_d^2}{a_d}\boldsymbol{e}_d\boldsymbol{e}_d^\top.
\end{aligned}$$

Substituting them back into Equation (64), we get

$$\begin{aligned}
\mathcal{L}(\mathcal{M}(\mathcal{S}_m)) &= \mathbb{E}\left[\boldsymbol{x}_q^\top\left(\boldsymbol{I} - 2\sum_{d\in\mathcal{S}_m}\frac{\lambda_d^2}{a_d}\boldsymbol{e}_d\boldsymbol{e}_d^\top + \sum_{d\in\mathcal{S}_m}\frac{\lambda_d^2}{a_d}\boldsymbol{e}_d\boldsymbol{e}_d^\top\right)\boldsymbol{x}_q\right] \\
&= \mathbb{E}\left[\boldsymbol{x}_q^\top\left(\boldsymbol{I} - \sum_{d\in\mathcal{S}_m}\frac{\lambda_d^2}{a_d}\boldsymbol{e}_d\boldsymbol{e}_d^\top\right)\boldsymbol{x}_q\right] \\
&= \mathbb{E}\left(\boldsymbol{x}_q^\top\boldsymbol{x}_q\right) - \sum_{d\in\mathcal{S}_m}\frac{\lambda_d^2}{a_d}\mathbb{E}\left(\boldsymbol{x}_q^\top\boldsymbol{e}_d\boldsymbol{e}_d^\top\boldsymbol{x}_q\right) \\
&= \mathtt{tr}(\boldsymbol{\Lambda}) - \sum_{d\in\mathcal{S}_m}\frac{\lambda_d^2}{a_d}\boldsymbol{e}_d^\top\boldsymbol{\Lambda}\boldsymbol{e}_d \\
&= \mathtt{tr}(\boldsymbol{\Lambda}) - \sum_{d\in\mathcal{S}_m}\frac{\lambda_d^3}{a_d}
\end{aligned}$$

We plug in the definition of a_d in Equation (32) and arrive at the desired result:

$$\begin{aligned}
\mathcal{L}(\mathcal{M}(\mathcal{S}_m)) &= \mathtt{tr}(\boldsymbol{\Lambda}) - \sum_{d\in\mathcal{S}_m}\lambda_d^3\frac{1}{\lambda_d^2}\left(1 + \frac{1+\mathtt{tr}(\boldsymbol{\Lambda})/\lambda_d}{N}\right)^{-1} \\
&= \mathtt{tr}(\boldsymbol{\Lambda}) - \sum_{d\in\mathcal{S}_m}\lambda_d\left(1 + \frac{1+\mathtt{tr}(\boldsymbol{\Lambda})/\lambda_d}{N}\right)^{-1}.
\end{aligned}$$

■

E.5. Saddle-to-Saddle Dynamics: From $\mathcal{M}_0$ to $\mathcal{M}_1$

We denote the time at which the loss has just undergone the d-th abrupt drop as t_d ($d = 1, \ldots, D$), as illustrated in Figure 10.

E.5.1. Alignment During the Plateau.

In the initial loss plateau, the weights have not moved much away from their small initialization and thus the training dynamics are mainly driven by the first terms in Equation (14), which are

$$\tau\dot{v}_i = \boldsymbol{k}_i^\top\boldsymbol{\Lambda}^2\boldsymbol{q}_i + O(w_{\text{init}}^5), \tag{65a}$$

$$\tau\dot{\boldsymbol{k}}_i = v_i\boldsymbol{\Lambda}^2\boldsymbol{q}_i + O(w_{\text{init}}^5), \tag{65b}$$

$$\tau\dot{\boldsymbol{q}}_i = v_i\boldsymbol{\Lambda}^2\boldsymbol{k}_i + O(w_{\text{init}}^5). \tag{65c}$$

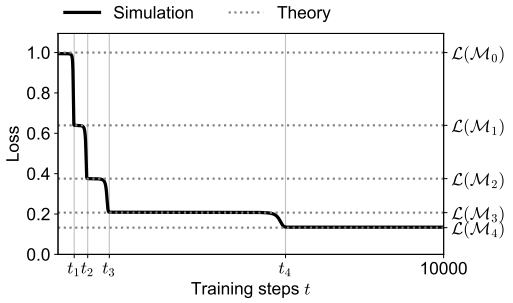


Figure 10. Illustration of $t_1, \cdots, t_D$. The loss trajectory plotted is one of the trajectories of linear attention with separate rank-one key and query in Figure 3a. The time t_d $(d = 1, \ldots, D)$ denotes the time when the loss has just undergone the d-th abrupt drop.

With a small initialization scale w_{init}, the key and query weights in a head evolve approximately as

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \boldsymbol{k}_i \\ \boldsymbol{q}_i \end{bmatrix} = v_i \begin{bmatrix} \mathbf{0} & \boldsymbol{\Lambda}^2 \\ \boldsymbol{\Lambda}^2 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{k}_i \\ \boldsymbol{q}_i \end{bmatrix}. \tag{66}$$

The matrix $\begin{bmatrix} \mathbf{0} & \boldsymbol{\Lambda}^2 \\ \boldsymbol{\Lambda}^2 & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{2D \times 2D}$ has eigenvalues $\left\{\lambda_d^2, -\lambda_d^2\right\}_{d=1}^D$, corresponding to eigenvectors

$$\begin{bmatrix} \mathbf{0} & \boldsymbol{\Lambda}^2 \\ \boldsymbol{\Lambda}^2 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_d \\ \boldsymbol{e}_d \end{bmatrix} = \lambda_d^2 \begin{bmatrix} \boldsymbol{e}_d \\ \boldsymbol{e}_d \end{bmatrix}, \quad \begin{bmatrix} \mathbf{0} & \boldsymbol{\Lambda}^2 \\ \boldsymbol{\Lambda}^2 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_d \\ -\boldsymbol{e}_d \end{bmatrix} = -\lambda_d^2 \begin{bmatrix} \boldsymbol{e}_d \\ -\boldsymbol{e}_d \end{bmatrix}, \quad d = 1, \cdots, D.$$

where recall that $\lambda_d, \boldsymbol{e}_d (d = 1, \cdots, D)$ are eigenvalues and eigenvectors of $\boldsymbol{\Lambda}$. Hence, the solution to Equation (66) takes the following form

$$\begin{aligned} \begin{bmatrix} \boldsymbol{k}_i(t) \\ \boldsymbol{q}_i(t) \end{bmatrix} = & \frac{1}{2} \sum_{d=1}^D \boldsymbol{e}_d^\top \left(\boldsymbol{k}_i(0) + \boldsymbol{q}_i(0)\right) \exp\left(\frac{\lambda_d^2}{\tau} \int_0^t v_i(t') \mathrm{d}t'\right) \begin{bmatrix} \boldsymbol{e}_d \\ \boldsymbol{e}_d \end{bmatrix} \\ & + \frac{1}{2} \sum_{d=1}^D \boldsymbol{e}_d^\top \left(\boldsymbol{k}_i(0) - \boldsymbol{q}_i(0)\right) \exp\left(-\frac{\lambda_d^2}{\tau} \int_0^t v_i(t') \mathrm{d}t'\right) \begin{bmatrix} \boldsymbol{e}_d \\ -\boldsymbol{e}_d \end{bmatrix}. \end{aligned} \tag{67}$$

If $v_i > 0$, the first summation term in Equation (67) grows and the second summation term decays. The key and query weights $\boldsymbol{k}_i, \boldsymbol{q}_i$ both grow in size along the directions of the eigenvectors $\boldsymbol{e}_d$. If $v_i < 0$, the first summation term in Equation (67) decays and the second summation term grows. The key and query weights $\boldsymbol{k}_i, \boldsymbol{q}_i$ grow in opposite directions, $\boldsymbol{e}_d$ and $-\boldsymbol{e}_d$ respectively. In either case, the multiplication $v_i \boldsymbol{k}_i \boldsymbol{q}_i^\top$ grows along $\boldsymbol{e}_d \boldsymbol{e}_d^\top$.

E.5.2. Reduction to Scalar Dynamics with An Alignment Ansatz.

The dominating term in Equation (67) is the term with the largest positive eigenvalue. In other words, the key and query weights grow the fastest along the first eigenvector $\boldsymbol{e}_1$ and thus are approximately aligned with $\boldsymbol{e}_1$. Motivated by this insight, we make an ansatz that the key and query weights in a head are exactly aligned with $\boldsymbol{e}_1$ and the rest of the heads are zero[5]:

$$\boldsymbol{k}_1 = \boldsymbol{q}_1 = v_1 \boldsymbol{e}_1, \tag{68a}$$

$$\boldsymbol{k}_i = \boldsymbol{q}_i = \mathbf{0}, v_i = 0, \; i = 2, \cdots, H. \tag{68b}$$

Note that Equation (68) also assumes that the ℓ^2 norms of $\boldsymbol{k}_1, \boldsymbol{q}_1, v_1$ are equal, which is true under vanishing initialization due to the conservation law in Equation (83). This ansatz can greatly simplify the training dynamics and provide a good approximation of the true dynamics, where weights in one of the heads grow in scale with the key and query weights aligning with $\boldsymbol{e}_1$, while the rest of the heads remain near zero from time 0 to t_1.

We substitute the ansatz into the training dynamics in Equation (14) to reduce the high-dimensional dynamics to a one-dimensional ordinary differential equation. To do that, we first calculate the common expectation term in the training

[5]We let the head aligned with $\boldsymbol{e}_1$ to have index 1.

dynamics with the ansatz,

$$\mathbf{\Lambda}^2 - \mathbb{E}\left(\hat{\mathbf{\Lambda}}^2\right)\sum_{i=1}^{H} v_i \boldsymbol{k}_i \boldsymbol{q}_i^\top \mathbf{\Lambda} = \mathbf{\Lambda}^2 - \sum_{d=1}^{D} a_d \boldsymbol{e}_d \boldsymbol{e}_d^\top v_1^3 \boldsymbol{e}_1 \boldsymbol{e}_1^\top \mathbf{\Lambda} = \mathbf{\Lambda}^2 - \lambda_1 a_1 \boldsymbol{e}_1 \boldsymbol{e}_1^\top v_1^3 \tag{69}$$

where a_1 is the first eigenvalue of $\mathbb{E}(\hat{\mathbf{\Lambda}}^2)$ defined in Equation (32). Substituting Equations (68) and (69) into Equation (14), we find that the training dynamics of the first head simplify and the dynamics of the rest of the heads are zero

$$\begin{aligned}
\tau \dot{v}_1 &= v_1^2 \boldsymbol{e}_1^\top \left(\mathbf{\Lambda}^2 - \lambda_1 a_1 \boldsymbol{e}_1 \boldsymbol{e}_1^\top v_1^3\right) \boldsymbol{e}_1 = \lambda_1^2 v_1^2 - \lambda_1 a_1 v_1^5, \\
\tau \dot{\boldsymbol{k}}_1 &= v_1^2 \left(\mathbf{\Lambda}^2 - \lambda_1 a_1 \boldsymbol{e}_1 \boldsymbol{e}_1^\top v_1^3\right) \boldsymbol{e}_1 = \lambda_1^2 v_1^2 \boldsymbol{e}_1 - \lambda_1 a_1 v_1^5 \boldsymbol{e}_1, \\
\tau \dot{\boldsymbol{q}}_1 &= v_1^2 \left(\mathbf{\Lambda}^2 - \lambda_1 a_1 \boldsymbol{e}_1 \boldsymbol{e}_1^\top v_1^3\right) \boldsymbol{e}_1 = \lambda_1^2 v_1^2 \boldsymbol{e}_1 - \lambda_1 a_1 v_1^5 \boldsymbol{e}_1, \\
\dot{v}_i &= 0, \dot{\boldsymbol{k}}_i = \boldsymbol{0}, \dot{\boldsymbol{q}}_i = \boldsymbol{0},\ i = 2, \cdots, H.
\end{aligned}$$

We further substitute in $\dot{\boldsymbol{k}}_1 = \dot{v}_1 \boldsymbol{e}_1, \dot{\boldsymbol{q}}_1 = \dot{v}_1 \boldsymbol{e}_1$ and find that the high-dimensional training dynamics reduce to one-dimensional dynamics about $v_1(t)$

$$\begin{cases}
\tau \dot{v}_1 = \lambda_1^2 v_1^2 - \lambda_1 a_1 v_1^5 \\
\tau \dot{v}_1 \boldsymbol{e}_1 = \lambda_1^2 v_1^2 \boldsymbol{e}_1 - \lambda_1 a_1 v_1^5 \boldsymbol{e}_1 \\
\tau \dot{v}_1 \boldsymbol{e}_1 = \lambda_1^2 v_1^2 \boldsymbol{e}_1 - \lambda_1 a_1 v_1^5 \boldsymbol{e}_1
\end{cases} \quad \Rightarrow \quad \tau \dot{v}_1 = \lambda_1^2 v_1^2 - \lambda_1 a_1 v_1^5 \tag{70}$$

Equation (70) is a separable ordinary differential equation. By separating variables and integrating both sides, we can solve t in terms of v_1

$$\begin{aligned}
\frac{\lambda_1^2}{\tau} t &= \int_{v_1(0)}^{v_1(t)} \frac{1}{v_1^2 - \frac{a_1}{\lambda_1} v_1^5} \mathrm{d}v_1 \\
&= \frac{\sqrt[3]{\frac{a_1}{\lambda_1}}}{6} \left[\ln \left(\frac{\sqrt[3]{\frac{a_1^2}{\lambda_1^2}} v_1^2 + \sqrt[3]{\frac{a_1}{\lambda_1}} v_1 + 1}{\sqrt[3]{\frac{a_1^2}{\lambda_1^2}} v_1^2 - 2\sqrt[3]{\frac{a_1}{\lambda_1}} v_1 + 1} \right) - 2\sqrt{3} \tan^{-1} \left(\frac{2\sqrt[3]{\frac{a_1}{\lambda_1}} v_1 + 1}{\sqrt{3}} \right) \right] - \frac{1}{v_1} + \text{constant}.
\end{aligned} \tag{71}$$

Since Equation (71) does not have a straight-forward inverse, we cannot obtain a general analytical solution of $v_1(t)$ in terms of t. Nonetheless, we can readily generate numerical solutions and obtain approximate analytical solutions when v_1 is near its small initialization to estimate the duration of the first loss plateau.

When v_1 is small, the dominating term in Equation (70) is $\lambda_1^2 v_1^2$ and thus the dynamics can be approximated by

$$\tau \dot{v}_1 = \lambda_1^2 v_1^2 \quad \Rightarrow \quad t = \frac{\tau}{\lambda_1^2} \left(\frac{1}{v_1(0)} - \frac{1}{v_1(t)} \right).$$

At the end of the plateau, $v_1(t)$ has grown to be much larger than $v_1(0)$. Hence, the duration of the first loss plateau, t_1, is

$$t_1 \approx \frac{\tau}{\lambda_1^2 v_1(0)}. \tag{72}$$

E.6. Saddle-to-Saddle Dynamics: From $\mathcal{M}_m$ to $\mathcal{M}_{m+1}$

In Appendix E.5, we have analyzed the training dynamics from time 0 to t_1, during which the model moves from saddle $\mathcal{M}_0$ to saddle $\mathcal{M}_1$. We now analyze the general saddle-to-saddle dynamics from time t_m to t_{m+1} $(m = 0, \cdots, D-1)$, during which the model moves from $\mathcal{M}_m$ to $\mathcal{M}_{m+1}$.

E.6.1. Alignment During the Plateau.

Based on our dynamics analysis from time 0 to t_1 and by induction, the weights during the m-th plateau are approximately described by Equation (20). Namely, there are m heads whose key and query weights have grown and become aligned with

the first m eigenvectors while weights in the rest of the heads have not moved much from their small initialization. Thus, similarly to Equation (61), the heads that are near small initialization have the following training dynamics

$$\tau \dot{v}_i = \boldsymbol{k}_i^\top \left(\sum_{d=m+1}^{D} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top \right) \boldsymbol{q}_i + O(w_{\text{init}}^5),$$
$$\tau \dot{\boldsymbol{k}}_i = v_i \left(\sum_{d=m+1}^{D} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top \right) \boldsymbol{q}_i + O(w_{\text{init}}^5),$$
$$\tau \dot{\boldsymbol{q}}_i = v_i \left(\sum_{d=m+1}^{D} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top \right) \boldsymbol{k}_i + O(w_{\text{init}}^5).$$

With a small initialization scale w_{init}, the key and query weights in this head evolve approximately as

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \boldsymbol{k}_i \\ \boldsymbol{q}_i \end{bmatrix} = v_i \begin{bmatrix} \boldsymbol{0} & \boldsymbol{\Omega} \\ \boldsymbol{\Omega} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{k}_i \\ \boldsymbol{q}_i \end{bmatrix}, \quad \text{where } \boldsymbol{\Omega} = \sum_{d=m+1}^{D} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top. \tag{73}$$

The matrix $\begin{bmatrix} \boldsymbol{0} & \boldsymbol{\Omega} \\ \boldsymbol{\Omega} & \boldsymbol{0} \end{bmatrix} \in \mathbb{R}^{2D \times 2D}$ has $2m$ zero eigenvalues and $(2D-2m)$ nonzero eigenvalues, which are $\left\{\lambda_d^2, -\lambda_d^2\right\}_{d=m+1}^{D}$. The nonzero eigenvalues correspond to eigenvectors

$$\begin{bmatrix} \boldsymbol{0} & \boldsymbol{\Omega} \\ \boldsymbol{\Omega} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_d \\ \boldsymbol{e}_d \end{bmatrix} = \lambda_d^2 \begin{bmatrix} \boldsymbol{e}_d \\ \boldsymbol{e}_d \end{bmatrix}, \quad \begin{bmatrix} \boldsymbol{0} & \boldsymbol{\Omega} \\ \boldsymbol{\Omega} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_d \\ -\boldsymbol{e}_d \end{bmatrix} = -\lambda_d^2 \begin{bmatrix} \boldsymbol{e}_d \\ -\boldsymbol{e}_d \end{bmatrix}, \quad d = m+1, \cdots, D.$$

Hence, the solution to Equation (73) takes the following the form

$$\begin{aligned} \begin{bmatrix} \boldsymbol{k}_i(t) \\ \boldsymbol{q}_i(t) \end{bmatrix} = & \frac{1}{2} \sum_{d=m+1}^{D} \boldsymbol{e}_d^\top \left(\boldsymbol{k}_i(t_m) + \boldsymbol{q}_i(t_m) \right) \exp\left(\frac{\lambda_d^2}{\tau} \int_{t_m}^{t} v_i(t') \mathrm{d}t' \right) \begin{bmatrix} \boldsymbol{e}_d \\ \boldsymbol{e}_d \end{bmatrix} \\ & + \frac{1}{2} \sum_{d=m+1}^{D} \boldsymbol{e}_d^\top \left(\boldsymbol{k}_i(t_m) - \boldsymbol{q}_i(t_m) \right) \exp\left(-\frac{\lambda_d^2}{\tau} \int_{t_m}^{t} v_i(t') \mathrm{d}t' \right) \begin{bmatrix} \boldsymbol{e}_d \\ -\boldsymbol{e}_d \end{bmatrix} \\ & + \sum_{d=1}^{m} \boldsymbol{e}_d^\top \left(\boldsymbol{k}_i(t_m) + \boldsymbol{q}_i(t_m) \right) \begin{bmatrix} \boldsymbol{e}_d \\ \boldsymbol{e}_d \end{bmatrix}. \end{aligned} \tag{74}$$

For $v_i > 0$, the first term grows and the second term decays with time. The third term does not change with respect to time.

E.6.2. Reduction to Scalar Dynamics with An Alignment Ansatz.

The dominating term in Equation (74) is the term with the largest positive eigenvalue. In other words, during the $(m+1)$-th plateau, the key and query weights that are still near small initialization grow the fastest along the $(m+1)$-th eigenvector $\boldsymbol{e}_{m+1}$. Based on this insight, we make the ansatz in Equation (20). This ansatz can reduce the high-dimensional training dynamics to a one-dimensional ordinary differential equation and provides a good approximation of the true dynamics, where weights in one of the heads grow in scale with the key and query weights aligning with $\boldsymbol{e}_{m+1}$, while the rest of the heads do not change much from time t_m to t_{m+1}.

To calculate the training dynamics in Equation (14) with the ansatz, we first calculate a common term with the ansatz

$$\begin{aligned} \boldsymbol{\Lambda}^2 - \mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right) \sum_{i=1}^{H} v_i \boldsymbol{k}_i \boldsymbol{q}_i^\top \boldsymbol{\Lambda} &= \boldsymbol{\Lambda}^2 - \sum_{d=1}^{D} a_d \boldsymbol{e}_d \boldsymbol{e}_d^\top \left(\sum_{i=1}^{m} \frac{\lambda_d}{a_d} \boldsymbol{e}_i \boldsymbol{e}_i^\top + v_{m+1}^3 \boldsymbol{e}_{m+1} \boldsymbol{e}_{m+1}^\top \right) \boldsymbol{\Lambda} \\ &= \boldsymbol{\Lambda}^2 - \sum_{d=1}^{m} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top - \lambda_{m+1} a_{m+1} \boldsymbol{e}_{m+1} \boldsymbol{e}_{m+1}^\top v_{m+1}^3 \end{aligned} \tag{75}$$

By substituting Equations (20) and (75) into Equation (14), we find that the dynamics for the heads with index $i \neq m+1$ are zero

$$\dot{v}_i = 0, \dot{\boldsymbol{k}}_i = \boldsymbol{0}, \dot{\boldsymbol{q}}_i = \boldsymbol{0}, \; i \neq m+1.$$

For the head with index $i = m + 1$, the dynamics reduce to one-dimensional dynamics about $v_i(t)$

$$\begin{aligned}
\tau \dot{v}_i &= v_i^2 \boldsymbol{e}_{m+1}^\top \left(\boldsymbol{\Lambda}^2 - \sum_{d=1}^{m} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top - \lambda_{m+1} a_{m+1} \boldsymbol{e}_{m+1} \boldsymbol{e}_{m+1}^\top v_i^3 \right) \boldsymbol{e}_{m+1} \\
&= \lambda_{m+1}^2 v_i^2 - \lambda_{m+1} a_{m+1} v_i^5 \\
\tau \dot{\boldsymbol{k}}_i = \tau \dot{v}_i \boldsymbol{e}_{m+1} &= v_i^2 \left(\boldsymbol{\Lambda}^2 - \sum_{d=1}^{m} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top - \lambda_{m+1} a_{m+1} \boldsymbol{e}_{m+1} \boldsymbol{e}_{m+1}^\top v_i^3 \right) \boldsymbol{e}_{m+1} \\
&= \lambda_{m+1}^2 v_i^2 \boldsymbol{e}_{m+1} - \lambda_{m+1} a_{m+1} v_i^5 \boldsymbol{e}_{m+1} \\
\tau \dot{\boldsymbol{q}}_i = \tau \dot{v}_i \boldsymbol{e}_{m+1} &= v_i^2 \left(\boldsymbol{\Lambda}^2 - \sum_{d=1}^{m} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top - \lambda_{m+1} a_{m+1} \boldsymbol{e}_{m+1} \boldsymbol{e}_{m+1}^\top v_i^3 \right) \boldsymbol{e}_{m+1} \\
&= \lambda_{m+1}^2 v_i^2 \boldsymbol{e}_{m+1} - \lambda_{m+1} a_{m+1} v_i^5 \boldsymbol{e}_{m+1} \\
\Rightarrow \quad \tau \dot{v}_i &= \lambda_{m+1}^2 v_i^2 - \lambda_{m+1} a_{m+1} v_i^5
\end{aligned} \tag{76}$$

Equation (76) is the same ordinary differential equation as Equation (70) modulo the constant coefficients. Therefore, with the same analysis, we can estimate the duration of the $(m+1)$-th loss plateau.

When v_{m+1} is small, the dominating term in Equation (76) is $\lambda_{m+1}^2 v_i^2$ and thus the dynamics is well approximated by

$$\tau \dot{v}_{m+1} = \lambda_{m+1}^2 v_{m+1}^2 \quad \Rightarrow \quad t - t_m = \frac{\tau}{\lambda_{m+1}^2} \left(\frac{1}{v_{m+1}(t_m)} - \frac{1}{v_{m+1}(t)} \right).$$

At the end of the plateau, $v_{m+1}(t_{m+1})$ has grown to be much larger than $v_{m+1}(t_m)$. Hence, the duration of the $(m+1)$-th loss plateau is

$$t_{m+1} - t_m \approx \frac{\tau}{\lambda_{m+1}^2 v_{m+1}(t_m)}. \tag{77}$$

We note that the Equation (77) involves $v_{m+1}(t_m)$, which depends on the random initialization and the dynamics from time 0 to t_m. This explains why we observe the variance of t_m increases with a larger m, that is the timing of a later abrupt loss drop varies more across random seeds as shown in Figure 3a.

E.7. Weight Configuration with Minimal L2 Norm

We prove that Equation (20) with $v_{m+1} = 0$ is the weight configuration with minimal ℓ^2 norm that satisfies Equation (18). To do this, we find the weight configuration with minimal ℓ^2 norm satisfying a general equality constrain and apply the solution to Equation (18).

Consider the equality constrained optimization problem

$$\begin{aligned}
&\text{minimize} \sum_{i=1}^{H} v_i^2 + \|\boldsymbol{k}_i\|^2 + \|\boldsymbol{q}_i\|^2 \\
&\text{subject to} \sum_{i=1}^{H} v_i \boldsymbol{k}_i \boldsymbol{q}_i^\top = \boldsymbol{A}
\end{aligned}$$

where $\boldsymbol{A}$ is a positive semi-definite matrix.

Proof. We use Lagrange multiplier to solve this equality constrained optimization problem. First, we construct the Lagrangian function $L(\boldsymbol{M})$ where the Lagrange multiplier $\boldsymbol{M} \in \mathbb{R}^{D \times D}$ is a symmetric matrix

$$\begin{aligned}
L(\boldsymbol{M}) &= \frac{1}{2} \sum_{i=1}^{H} \left(v_i^2 + \|\boldsymbol{k}_i\|^2 + \|\boldsymbol{q}_i\|^2 \right) + \texttt{vec}(\boldsymbol{M})^\top \texttt{vec} \left(\boldsymbol{A} - \sum_{i=1}^{H} v_i \boldsymbol{k}_i \boldsymbol{q}_i^\top \right) \\
&= \frac{1}{2} \sum_{i=1}^{H} \left(v_i^2 + \|\boldsymbol{k}_i\|^2 + \|\boldsymbol{q}_i\|^2 \right) + \texttt{tr} \left[\boldsymbol{M} \left(\boldsymbol{A} - \sum_{i=1}^{H} v_i \boldsymbol{k}_i \boldsymbol{q}_i^\top \right) \right]
\end{aligned}$$

Differentiating the Lagrangian with respect to all the variables and setting them to zero, we get

$$\frac{\partial L}{\partial v_i} = v_i - \boldsymbol{k}_i^\top \boldsymbol{M} \boldsymbol{q}_i = 0 \tag{78a}$$

$$\frac{\partial L}{\partial \boldsymbol{k}_i} = \boldsymbol{k}_i - v_i \boldsymbol{M} \boldsymbol{q}_i = \boldsymbol{0} \tag{78b}$$

$$\frac{\partial L}{\partial \boldsymbol{q}_i} = \boldsymbol{q}_i - v_i \boldsymbol{M} \boldsymbol{k}_i = \boldsymbol{0} \tag{78c}$$

$$\frac{\partial L}{\partial \boldsymbol{M}} = \boldsymbol{A} - \sum_{i=1}^{H} v_i \boldsymbol{k}_i \boldsymbol{q}_i^\top = 0 \tag{78d}$$

Equation (78) suggests that, for each head, the value, key, and query weights are either all zero or satisfy a constraint; i.e., for each i, either $v_i = \boldsymbol{k}_i = \boldsymbol{q}_i = 0$ or

$$\boldsymbol{k}_i = v_i \boldsymbol{M} \boldsymbol{q}_i = v_i^2 \boldsymbol{M}^2 \boldsymbol{k}_i. \tag{79}$$

We got Equation (79) by substituting Equation (78c) into Equation (78b). Equation (79) implies that $\boldsymbol{k}_i$ is an eigenvector of $\boldsymbol{M}^2$. Let us denote the normalized eigenvector of $\boldsymbol{M}^2$ as $\boldsymbol{\xi}_i$. Substituting Equation (79) into Equation (78a) and rearranging, we get

$$\boldsymbol{k}_i = \boldsymbol{q}_i = v_i \boldsymbol{\xi}_i. \tag{80}$$

With Equations (78d) and (80), we obtain

$$\boldsymbol{A} = \sum_i v_i^3 \boldsymbol{\xi}_i \boldsymbol{\xi}_i^\top \quad \Rightarrow \quad v_i = \lambda_i^{1/3},\ \boldsymbol{\xi}_i = \boldsymbol{e}_i, \tag{81}$$

where $\lambda_i, \boldsymbol{e}_i$ are the eigenvalue and eigenvector of $\boldsymbol{A}$.

For the optimization problem, the solution is that there are $\texttt{rank}(\boldsymbol{A})$ heads with nonzero weights and $(H - \texttt{rank}(\boldsymbol{A}))$ heads with zero weights. The nonzero heads have weights

$$\boldsymbol{k}_i = \boldsymbol{q}_i = v_i \boldsymbol{e}_i,\ v_i = \lambda_i^{1/3}, \quad i = 1, \cdots, \texttt{rank}(\boldsymbol{A}). \tag{82}$$

The indices of heads can be trivially permuted. The signs of any two among $v_i, \boldsymbol{k}_i, \boldsymbol{q}_i$ can be flipped without affecting the optimization problem. ■

We apply the solution in Equation (82) to find a weight configuration with the minimal ℓ^2 norm that satisfies Equation (18). Equation (18) can be rewritten as Equation (59), namely

$$\sum_{i=1}^{H} v_i \boldsymbol{k}_i \boldsymbol{q}_i^\top = \sum_{d \in \mathcal{S}_m} \frac{\lambda_d}{a_d} \boldsymbol{e}_d \boldsymbol{e}_d^\top.$$

The matrix on the right hand side has rank m and eigenvectors $\boldsymbol{e}_d$ with eigenvalues λ_d / a_d $(d \in \mathcal{S}_m)$. Hence, the weight configuration with minimal ℓ^2 norm has $(H - m)$ heads with zero weights and m heads with nonzero weights. The nonzero heads have weights

$$\boldsymbol{k}_i = \boldsymbol{q}_i = v_i \boldsymbol{e}_i,\ v_i = \left(\frac{\lambda_d}{a_d}\right)^{\frac{1}{3}} = \lambda_i^{-\frac{1}{3}} \left(1 + \frac{1 + \texttt{tr}(\boldsymbol{\Lambda})/\lambda_i}{N}\right)^{-\frac{1}{3}}, \quad i = 1, \cdots, m.$$

This is the same weight configuration as Equation (20) with $v_{m+1} = 0$.

E.8. Conservation Law

The gradient flow dynamics of linear attention with separate rank-one key and query in Equation (14) implies a conservation law. The value, key, and query weights in a head obey

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{k}_i^\top \boldsymbol{k}_i - \boldsymbol{q}_i^\top \boldsymbol{q}_i\right) = \boldsymbol{0}, \quad \frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{k}_i^\top \boldsymbol{k}_i - v_i^2\right) = 0, \tag{83}$$

Under small initialization, the quantities $\boldsymbol{k}_i^\top \boldsymbol{k}_i - \boldsymbol{q}_i^\top \boldsymbol{q}_i \approx \boldsymbol{0}$ and $\boldsymbol{k}_i^\top \boldsymbol{k}_i - v_i^2 \approx 0$ are small at initialization and remain small throughout training. Thus, the conservation law enforces the ℓ^2 norms of the value, key, and query to be approximately the same throughout training, $\|\boldsymbol{k}_i\|^2 \approx \|\boldsymbol{q}_i\|^2 \approx v_i^2$.

We here prove that Equation (83) holds regardless of the choice of the loss function.

Proof. We can use the generic gradient flow equation in Equation (5) to calculate the gradients of $\boldsymbol{k}_i^\top \boldsymbol{k}_i$, $\boldsymbol{q}_i^\top \boldsymbol{q}_i$, and v_i^2,

$$\begin{aligned}
\frac{\mathrm{d}\boldsymbol{k}_i^\top \boldsymbol{k}_i}{\mathrm{d}t} &= 2\boldsymbol{k}_i^\top \frac{\mathrm{d}\boldsymbol{k}_i}{\mathrm{d}t} = 2\mathbb{E}\left(-\boldsymbol{k}_i^\top \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\hat{y}_q}\frac{\mathrm{d}\hat{y}_q}{\mathrm{d}\boldsymbol{k}_i}\right) = 2\mathbb{E}\left(-\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\hat{y}_q} v_i \boldsymbol{k}_i^\top \boldsymbol{\beta} \boldsymbol{q}_i^\top \boldsymbol{x}_q\right) \\
\frac{\mathrm{d}\boldsymbol{q}_i^\top \boldsymbol{q}_i}{\mathrm{d}t} &= 2\boldsymbol{q}_i^\top \frac{\mathrm{d}\boldsymbol{q}_i}{\mathrm{d}t} = 2\mathbb{E}\left(-\boldsymbol{q}_i^\top \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\hat{y}_q}\frac{\mathrm{d}\hat{y}_q}{\mathrm{d}\boldsymbol{q}_i}\right) = 2\mathbb{E}\left(-\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\hat{y}_q} v_i \boldsymbol{q}_i^\top \boldsymbol{x}_q \boldsymbol{k}_i^\top \boldsymbol{\beta}\right) \\
\frac{\mathrm{d}v_i^2}{\mathrm{d}t} &= 2v_i \frac{\mathrm{d}v_i}{\mathrm{d}t} = 2\mathbb{E}\left(-v_i \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\hat{y}_q}\frac{\mathrm{d}\hat{y}_q}{\mathrm{d}v_i}\right) = 2\mathbb{E}\left(-\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\hat{y}_q} v_i \boldsymbol{\beta}^\top \boldsymbol{k}_i \boldsymbol{q}_i^\top \boldsymbol{x}_q\right)
\end{aligned}$$

We see that the gradients of $\boldsymbol{k}_i^\top \boldsymbol{k}_i$, $\boldsymbol{q}_i^\top \boldsymbol{q}_i$, and v_i^2 are equal, regardless of the specific choice of the loss function $\mathcal{L}$. Hence, the following conservation law holds for any loss function:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{k}_i^\top \boldsymbol{k}_i - \boldsymbol{q}_i^\top \boldsymbol{q}_i\right) = \boldsymbol{0}, \quad \frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{k}_i^\top \boldsymbol{k}_i - v_i^2\right) = 0.$$

■

F. Linear Attention with Separate Low-Rank Key and Query

F.1. Justification for Zero Blocks Assumption

We initialize $\boldsymbol{v}_i = \boldsymbol{0}, k_{i,r} = 0\,(i = 1, \cdots, H, r = 1, \cdots, R)$, and prove that they will stay zero throughout training.

Proof. The bottom right entry of the output of linear attention with separate rank-R key and query is

$$\begin{aligned}
\hat{y}_q &\equiv \mathsf{ATTN}_\mathsf{S}(\boldsymbol{X})_{D+1,N+1} \\
&= \sum_{i=1}^{H} \begin{bmatrix} \boldsymbol{v}_i^\top & v_i \end{bmatrix} \begin{bmatrix} \frac{1}{N}\left(\boldsymbol{x}_q \boldsymbol{x}_q^\top + \sum_n \boldsymbol{x}_n \boldsymbol{x}_n^\top\right) & \frac{1}{N}\sum_n \boldsymbol{x}_n y_n \\ \frac{1}{N}\sum_n y_n \boldsymbol{x}^\top & \frac{1}{N}\sum_n y_n^2 \end{bmatrix} \begin{bmatrix} \boldsymbol{k}_{i,1} & \cdots & \boldsymbol{k}_{i,R} \\ k_{i,1} & \cdots & k_{i,R} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_{i,1}^\top \\ \vdots \\ \boldsymbol{q}_{i,R}^\top \end{bmatrix} \boldsymbol{x}_q \\
&= \sum_{i=1}^{H} \left(\boldsymbol{v}_i^\top \left(\hat{\boldsymbol{\Lambda}} + \frac{1}{N}\boldsymbol{x}_q \boldsymbol{x}_q^\top\right) \sum_{r=1}^{R} \boldsymbol{k}_{i,r}\boldsymbol{q}_{i,r}^\top + v_i \boldsymbol{\beta}^\top \sum_{r=1}^{R} \boldsymbol{k}_{i,r}\boldsymbol{q}_{i,r}^\top + \boldsymbol{v}_i^\top \boldsymbol{\beta} \sum_{r=1}^{R} k_{i,r}\boldsymbol{q}_{i,r}^\top + v_i \boldsymbol{w}^\top \hat{\boldsymbol{\Lambda}} \boldsymbol{w} \sum_{r=1}^{R} k_{i,r}\boldsymbol{q}_{i,r}^\top \right) \boldsymbol{x}_q
\end{aligned}$$

If we initialize $\boldsymbol{v}_i = \boldsymbol{0}, k_{i,r} = 0$, $\hat{y}_q$ is

$$\hat{y}_q = \sum_{i=1}^{H}\sum_{r=1}^{R} v_i \boldsymbol{\beta}^\top \boldsymbol{k}_{i,r}\boldsymbol{q}_{i,r}^\top \boldsymbol{x}_q = \boldsymbol{w}^\top \hat{\boldsymbol{\Lambda}} \sum_{i=1}^{H}\sum_{r=1}^{R} v_i \boldsymbol{k}_{i,r}\boldsymbol{q}_{i,r}^\top \boldsymbol{x}_q.$$

We now calculate the gradient updates of $\boldsymbol{v}_i = \boldsymbol{0}, k_{i,r} = 0$ and prove their gradients are zero if their initialization is zero.

The gradient update of $\boldsymbol{v}_i$ contains $\mathbb{E}(\boldsymbol{w})$, which is zero. Similarly to Equation (40), we have

$$\begin{aligned}
\tau\dot{\boldsymbol{v}}_i &= \mathbb{E}\left[(y_q-\hat{y}_q)\left(\left(\hat{\boldsymbol{\Lambda}}+\frac{1}{N}\boldsymbol{x}_q\boldsymbol{x}_q^\top\right)\sum_{r=1}^{R}\boldsymbol{k}_{i,r}\boldsymbol{q}_{i,r}^\top+\boldsymbol{\beta}\sum_{r=1}^{R}k_{i,r}\boldsymbol{q}_{i,r}^\top\right)\boldsymbol{x}_q\right] \\
&= \mathbb{E}\left[\left(\boldsymbol{w}^\top\boldsymbol{x}_q-\boldsymbol{w}^\top\hat{\boldsymbol{\Lambda}}\sum_{i=1}^{H}\sum_{r=1}^{R}v_i\boldsymbol{k}_{i,r}\boldsymbol{q}_{i,r}^\top\boldsymbol{x}_q\right)\left(\hat{\boldsymbol{\Lambda}}+\frac{1}{N}\boldsymbol{x}_q\boldsymbol{x}_q^\top\right)\sum_{r=1}^{R}\boldsymbol{k}_{i,r}\boldsymbol{q}_{i,r}^\top\boldsymbol{x}_q\right] \\
&= \mathbb{E}_{\boldsymbol{w}}(\boldsymbol{w})^\top\mathbb{E}\left[\left(\boldsymbol{x}_q-\hat{\boldsymbol{\Lambda}}\sum_{i=1}^{H}\sum_{r=1}^{R}v_i\boldsymbol{k}_{i,r}\boldsymbol{q}_{i,r}^\top\boldsymbol{x}_q\right)\left(\hat{\boldsymbol{\Lambda}}+\frac{1}{N}\boldsymbol{x}_q\boldsymbol{x}_q^\top\right)\sum_{r=1}^{R}\boldsymbol{k}_{i,r}\boldsymbol{q}_{i,r}^\top\boldsymbol{x}_q\right] \\
&= \mathbf{0}.
\end{aligned}$$

The gradient update of $k_{i,r}$ contains $\mathbb{E}_{\boldsymbol{w}}\left(\boldsymbol{w}^\top\hat{\boldsymbol{\Lambda}}\boldsymbol{w}\boldsymbol{w}^\top\right)$, whose entries are linear combinations of third moments the zero-mean normal random variable $\boldsymbol{w}$, and are thus zero. Similarly to Equation (41), we have

$$\begin{aligned}
\tau\dot{k}_{i,r} &= \mathbb{E}\left[\left(\boldsymbol{v}_i^\top\boldsymbol{\beta}+v_i\boldsymbol{w}^\top\hat{\boldsymbol{\Lambda}}\boldsymbol{w}\right)(y_q-\hat{y}_q)\boldsymbol{q}_{i,r}^\top\boldsymbol{x}_q\right] \\
&= \mathbb{E}\left[v_i\boldsymbol{w}^\top\hat{\boldsymbol{\Lambda}}\boldsymbol{w}\left(\boldsymbol{w}^\top\boldsymbol{x}_q-\boldsymbol{w}^\top\hat{\boldsymbol{\Lambda}}\sum_{i=1}^{H}\sum_{r'=1}^{R}v_i\boldsymbol{k}_{i,r'}\boldsymbol{q}_{i,r'}^\top\boldsymbol{x}_q\right)\boldsymbol{q}_{i,r}^\top\boldsymbol{x}_q\right] \\
&= \mathbb{E}_{\boldsymbol{w}}\left(\boldsymbol{w}^\top\hat{\boldsymbol{\Lambda}}\boldsymbol{w}\boldsymbol{w}^\top\right)\mathbb{E}\left[v_i\left(\boldsymbol{x}_q-\hat{\boldsymbol{\Lambda}}\sum_{i=1}^{H}\sum_{r'=1}^{R}v_i\boldsymbol{k}_{i,r'}\boldsymbol{q}_{i,r'}^\top\boldsymbol{x}_q\right)\boldsymbol{q}_{i,r}^\top\boldsymbol{x}_q\right] \\
&= \mathbf{0}.
\end{aligned}$$

■

F.2. Gradient Flow Equations

Based on the gradient flow training rule in Equation (5), the gradient flow dynamics of linear attention with separate rank-R key and query is

$$\tau\dot{v}_i = \sum_{r=1}^{R}\boldsymbol{k}_{i,r}^\top\mathbb{E}\left(\boldsymbol{\beta}(y_q-\hat{y}_q)\boldsymbol{x}_q^\top\right)\boldsymbol{q}_{i,r} = \sum_{r=1}^{R}\boldsymbol{k}_{i,r}^\top\left(\boldsymbol{\Lambda}^2-\mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right)\sum_{i=1}^{H}\sum_{r'=1}^{R}v_i\boldsymbol{k}_{i,r'}\boldsymbol{q}_{i,r'}^\top\boldsymbol{\Lambda}\right)\boldsymbol{q}_{i,r}, \tag{84a}$$

$$\tau\dot{\boldsymbol{k}}_{i,r} = v_i\mathbb{E}\left(\boldsymbol{\beta}(y_q-\hat{y}_q)\boldsymbol{x}_q^\top\right)\boldsymbol{q}_{i,r} = v_i\left(\boldsymbol{\Lambda}^2-\mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right)\sum_{i=1}^{H}\sum_{r'=1}^{R}v_i\boldsymbol{k}_{i,r'}\boldsymbol{q}_{i,r'}^\top\boldsymbol{\Lambda}\right)\boldsymbol{q}_{i,r}, \tag{84b}$$

$$\tau\dot{\boldsymbol{q}}_{i,r} = v_i\boldsymbol{k}_{i,r}^\top\mathbb{E}\left(\boldsymbol{\beta}(y_q-\hat{y}_q)\boldsymbol{x}_q\right) = v_i\left(\boldsymbol{\Lambda}^2-\boldsymbol{\Lambda}\sum_{i=1}^{H}\sum_{r'=1}^{R}v_i\boldsymbol{q}_{i,r'}\boldsymbol{k}_{i,r'}^\top\mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right)\right)\boldsymbol{k}_{i,r}. \tag{84c}$$

where $i=1,\cdots,H$, $r=1,\cdots,R$, and the data statistics $\mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right)$ is calculated in Equation (31).

F.3. Fixed Points

We use $\mathcal{M}(\mathcal{S}_m)$ to denote a set of fixed points that correspond to learning m ($m=0,1,\cdots,D$) out of the D eigenvectors,

$$\mathcal{M}(\mathcal{S}_m) = \left\{v_{1:H},\boldsymbol{W}_{1:H}^K,\boldsymbol{W}_{1:H}^Q\middle|\text{conditions (C1)-(C3) are met}\right\}, \tag{85}$$

where the set $\mathcal{S}_m$ specifies the indices of the learned eigenvectors,

$$\mathcal{S}_m\subseteq\{1,2,\cdots,D\},\ |\mathcal{S}_m|=m. \tag{86}$$

The three conditions for Equation (85) are:

(C1) The heads sum up to fit the eigenvectors with indices $\mathcal{S}_m$

$$\sum_{i=1}^{H}\sum_{r=1}^{R} v_i \boldsymbol{k}_{i,r}\boldsymbol{q}_{i,r}^\top = \sum_{d\in\mathcal{S}_m} \lambda_d^{-1}\left(1+\frac{1+\mathtt{tr}(\boldsymbol{\Lambda})/\lambda_d}{N}\right)^{-1} \boldsymbol{e}_d\boldsymbol{e}_d^\top. \tag{87}$$

(C2) For heads with a nonzero value weight, $v_i \neq 0$, $\boldsymbol{k}_{i,r}, \boldsymbol{q}_{i,r}$ $(r = 1, \cdots, R)$ all lie in the span of $\{\boldsymbol{e}_d\}_{d\in\mathcal{S}_m}$.

(C3) For heads with a zero value weight, $v_i = 0$,

$$\sum_{r=1}^{R}\sum_{d\notin\mathcal{S}_m} \lambda_d^2 \boldsymbol{k}_{i,r}^\top \boldsymbol{e}_d\boldsymbol{e}_d^\top \boldsymbol{q}_{i,r} = 0. \tag{88}$$

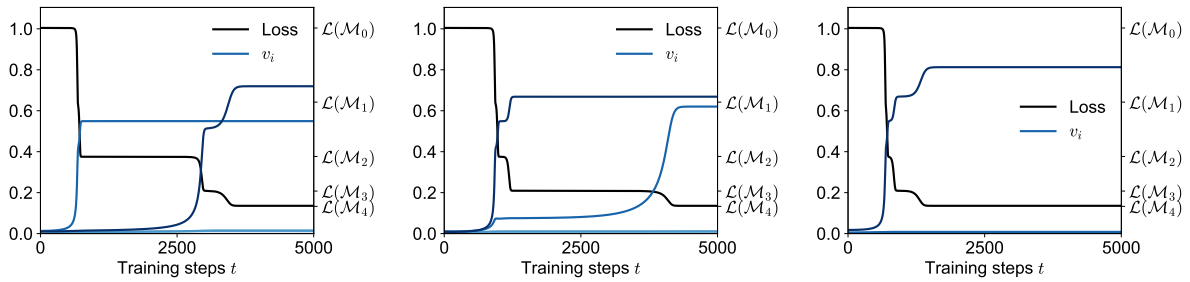
With the same reasoning as Appendix E.3, one can show the weights satisfying these three conditions have zero gradients and thus are fixed points. Though conditions (C1,C3) do not explicitly specify the weights, they are feasible conditions. One possible weight configuration that satisfies all three conditions is to let $\boldsymbol{k}_{i,r}, \boldsymbol{q}_{i,r}$ $(r \neq 1)$ be zero and let $v_i, \boldsymbol{k}_{i,1}, \boldsymbol{q}_{i,1}$ be the same as the fixed point for linear attention with rank-one key query, where the low-rank case falls back into the rank-one case. Therefore, the fixed points described in Equation (85) are valid and feasible. Linear attention with separate rank-R key and query has the same 2^D fixed points in the function space as its rank-one counterpart.

F.4. Saddle-to-Saddle Dynamics

For linear attention with rank-R key and query, the gradient updates of the key and query weights in Equation (84), $\dot{\boldsymbol{k}}_{i,r}, \dot{\boldsymbol{q}}_{i,r}$, include the factor v_i, which is the shared across ranks $r = 1, \cdots, R$ but unique to each head. In linear attention with rank-one key and query initialized with small weights, the weights in a head, $v_i, \boldsymbol{k}_i, \boldsymbol{q}_i$, escape from the unstable zero fixed point to drive the first abrupt drop of loss. Similarly, in the rank-R model, the value weight v_i and a pair of key and query weights $\boldsymbol{k}_{i,r}, \boldsymbol{q}_{i,r}$ in a head escape from the zero fixed point to drive the first abrupt drop of loss.

However, the subsequent dynamics differ between the the rank-one and rank-R models. In the rank-one model, the loss will undergo a conspicuous plateau until weights in a new head, $v_{i'}, \boldsymbol{k}_{i'}, \boldsymbol{q}_{i'}$ $(i' \neq i)$, escape from the zero fixed point to grow. By contrast, in the rank-R model $(R > 1)$, the loss will plateau briefly or not plateau because a new pair of key and query weights in the same i-th head, $\boldsymbol{k}_{i,r'}, \boldsymbol{q}_{i,r'}$ $(r' \neq r)$, can quickly grow to drive the loss drop. A new pair of key and query weights in the i-th head grows faster than the key and query weights in a new head, because the value weight in the i-th head, v_i, has already grown during the first abrupt loss drop. Since the gradient updates of all key and query weights in the i-th head include the factor v_i, a larger value weight leads to larger gradient updates for the associated key and query weights. We plot the value weights with $D = 4$ and ranks $R = 1, 2, 3, 4$ in Figures 3b and 11 to show: the loss drop after a conspicuous plateau corresponds to a new value weight escaping from zero, while the loss drop after a brief plateau does not.

We plot the loss trajectories with $D = 8$ and different ranks in Figure 12 to complement Figure 4 in the main text.



(a) $R = 2$ (b) $R = 3$ (c) $R = 4$

Figure 11. Loss and value weights trajectories. The setting is the same as Figure 3b except different ranks $R = 2, 3, 4$. In the rank-one case in Figure 3b, value weights in four heads grow, each corresponding to an abrupt loss drop from $\mathcal{L}(\mathcal{M}_m)$ to $\mathcal{L}(\mathcal{M}_{m+1})$ $(m = 0, 1, 2, 3)$. In the rank-R case, a new value weight grows big from small initialization when the loss decreases from $\mathcal{L}(\mathcal{M}_m)$ to $\mathcal{L}(\mathcal{M}_{m+1})$ for m that divides R. Here $D = 4, N = 31, H = 5$, and $\boldsymbol{\Lambda}$ has eigenvalues $0.4, 0.3, 0.2, 0.1$.

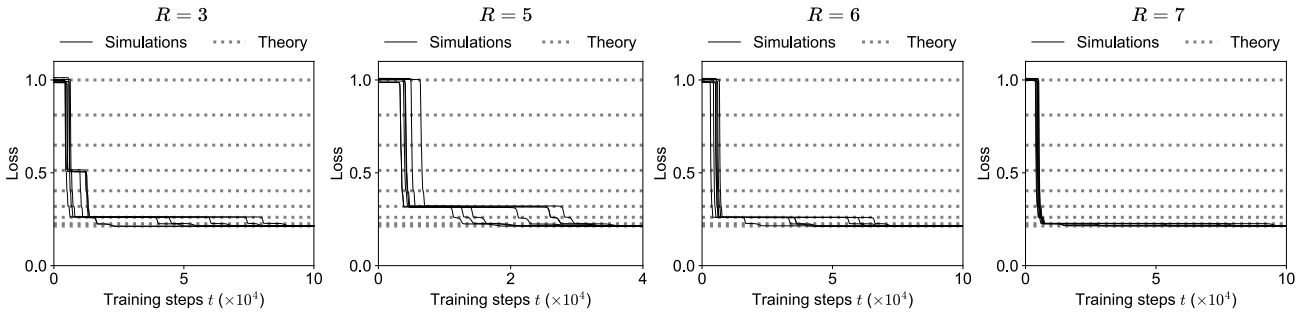


Figure 12. Same as Figure 4 but with ranks $R = 3, 5, 6, 7$. Here $D = 8, N = 31, H = 9$, $\mathbf{\Lambda}$ has trace 1 and eigenvalues $\lambda_d \propto d^{-1}$.

F.5. Dynamics with Repeated Eigenvalues

We have demonstrated that linear attention with separate key and query exhibits loss plateaus during training when the eigenvalues of the input token covariance matrix, $\mathbf{\Lambda}$, are distinct. When $\mathbf{\Lambda}$ has repeated eigenvalues, linear attention with separate key and query can also exhibit loss plateaus due to the different random initial weights in each head. In the case with distinct eigenvalues (Figure 3a), the plateau duration is determined by both the size of the eigenvalues and the random initialization. In the case with repeated eigenvalues (Figure 13, leftmost panel), the plateau duration is determined solely by the random initialization.

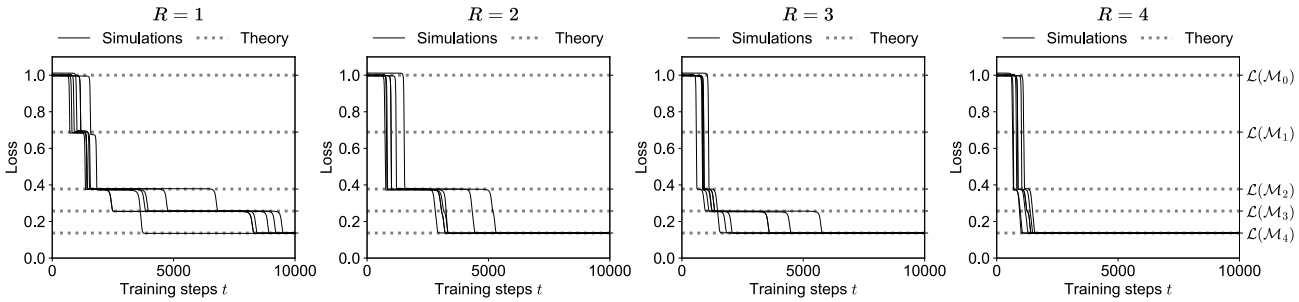


Figure 13. Loss trajectories of multi-head linear attention with separate key and query. The setup is the same as in Figure 3a except that $\mathbf{\Lambda}$ has eigenvalues $0.35, 0.35, 0.15, 0.15$. The four panels differ only in the rank of the key and query weights. Although some eigenvalues are equal, the loss trajectory of ATTN_S with $R = 1$ can still exhibit plateaus when learning them, due to the different random initial weights in each head. The plateaus may also be skipped for certain random seeds.

F.6. Conservation Law

The gradient flow dynamics of linear attention with separate key and query in Equation (84) implies a conservation law. The value, key, and query weights in a head obey

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{k}_{i,r}^\top \boldsymbol{k}_{i,r} - \boldsymbol{q}_{i,r}^\top \boldsymbol{q}_{i,r}\right) = \mathbf{0}, \quad \frac{\mathrm{d}}{\mathrm{d}t}\left(\sum_{r=1}^{R} \boldsymbol{k}_{i,r}^\top \boldsymbol{k}_{i,r} - v_i^2\right) = 0. \tag{89}$$

We here prove that Equation (89) holds regardless of the choice of the loss function.

Proof. We can use the generic gradient flow equation in Equation (5) to calculate the relevant gradients

$$\frac{\mathrm{d}\boldsymbol{k}_{i,r}^\top \boldsymbol{k}_{i,r}}{\mathrm{d}t} = 2\boldsymbol{k}_{i,r}^\top \frac{\mathrm{d}\boldsymbol{k}_{i,r}}{\mathrm{d}t} = 2\mathbb{E}\left(-\boldsymbol{k}_i^\top \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\hat{y}_q}\frac{\mathrm{d}\hat{y}_q}{\mathrm{d}\boldsymbol{k}_{i,r}}\right) = 2\mathbb{E}\left(-\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\hat{y}_q} v_i \boldsymbol{k}_{i,r}^\top \boldsymbol{\beta} \boldsymbol{q}_{i,r}^\top \boldsymbol{x}_q\right) \tag{90a}$$

$$\frac{\mathrm{d}\boldsymbol{q}_{i,r}^\top \boldsymbol{q}_{i,r}}{\mathrm{d}t} = 2\boldsymbol{q}_{i,r}^\top \frac{\mathrm{d}\boldsymbol{q}_{i,r}}{\mathrm{d}t} = 2\mathbb{E}\left(-\boldsymbol{q}_i^\top \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\hat{y}_q}\frac{\mathrm{d}\hat{y}_q}{\mathrm{d}\boldsymbol{q}_{i,r}}\right) = 2\mathbb{E}\left(-\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\hat{y}_q} v_i \boldsymbol{q}_{i,r}^\top \boldsymbol{x}_q \boldsymbol{k}_{i,r}^\top \boldsymbol{\beta}\right) \tag{90b}$$

$$\frac{\mathrm{d}v_i^2}{\mathrm{d}t} = 2v_i\frac{\mathrm{d}v_i}{\mathrm{d}t} = 2\mathbb{E}\left(-v_i\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\hat{y}_q}\frac{\mathrm{d}\hat{y}_q}{\mathrm{d}v_i}\right) = 2\sum_{r=1}^{R}\mathbb{E}\left(-\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\hat{y}_q}v_i\boldsymbol{\beta}^\top\boldsymbol{k}_{i,r}\boldsymbol{q}_{i,r}^\top\boldsymbol{x}_q\right) \tag{90c}$$

Comparing Equations (90a) and (90b), we see that the following holds regardless of the specific choice of the loss function $\mathcal{L}$

$$\frac{\mathrm{d}\boldsymbol{k}_{i,r}^\top\boldsymbol{k}_{i,r}}{\mathrm{d}t} = \frac{\mathrm{d}\boldsymbol{q}_{i,r}^\top\boldsymbol{q}_{i,r}}{\mathrm{d}t}.$$

Similarly, comparing Equations (90a) and (90b) with Equation (90c), we obtain

$$\sum_{r=1}^{R}\frac{\mathrm{d}\boldsymbol{k}_{i,r}^\top\boldsymbol{k}_{i,r}}{\mathrm{d}t} = \frac{\mathrm{d}v_i^2}{\mathrm{d}t}.$$

■

G. Training Dynamics of In-Context and In-Weight Learning

In this work, we focused on the training dynamics of ICL abilities. Other than ICL, attention models can also learn in weight, that is solving the task by memorizing the map between the query input and the target output without using the information in context. The arbitration between in-context and in-weight learning may depend on the properties of the training data (Chan et al., 2022). To focus on the dynamics of ICL, we considered a purely ICL task, which is in-context linear regression with the task vector sampled from a zero-mean standard normal distribution, $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. Since memorizing any particular task vector does not effectively decrease the loss, linear attention develops only ICL ability during training, as shown in Figure 14a.

If the task vector $\boldsymbol{w}$ follows a different distribution, the training dynamics involves the development of both in-context and in-weight learning abilities. In Figure 14, we let the task vector for some of the training sequences be fixed and sample the rest from a standard normal distribution to elicit in-weight learning ability. We plot the training loss, in-context learning test loss, and in-weight learning test loss for varying portions of fixed task vectors in Figure 14. The larger the portion of fixed task vectors, the lower the loss the model can achieve by memorizing the fixed task vector in weight. We indeed observe the training loss and in-weight learning test loss are lower right after the first abrupt loss drop when the portion is larger.

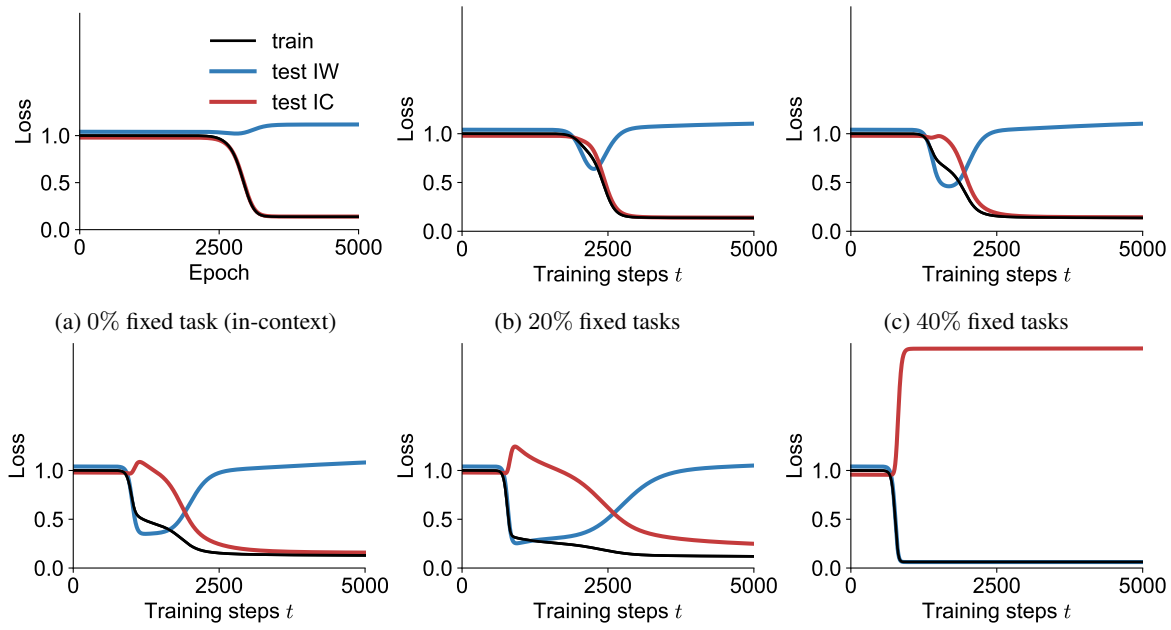
The technical consequence of fixing some of the task vectors is that Equations (40) and (41) break. In other words, we cannot assume the certain blocks of the value and the merged key-query matrices are zero as in Appendix D.1. Without the zero block assumption, the linear attention model implements

$$\hat{y}_q = \sum_{i=1}^{H}\left(\boldsymbol{v}_i^\top\left(\hat{\boldsymbol{\Lambda}} + \frac{1}{N}\boldsymbol{x}_q\boldsymbol{x}_q^\top\right)\boldsymbol{U}_i + v_i\boldsymbol{\beta}^\top\boldsymbol{U}_i + \boldsymbol{v}_i^\top\boldsymbol{\beta}\boldsymbol{u}_i^\top + v_i\frac{1}{N}\sum_{n=1}^{N}y_n^2\boldsymbol{u}_i^\top\right)\boldsymbol{x}_q. \tag{91}$$

Equation (91) include not only a linear map of the cubic feature $\boldsymbol{z} = \mathtt{vec}(\boldsymbol{\beta}\boldsymbol{x}_q^\top)$ but also linear maps of additional features, $\left(\hat{\boldsymbol{\Lambda}} + \frac{1}{N}\boldsymbol{x}_q\boldsymbol{x}_q^\top\right) \otimes \boldsymbol{x}_q$, $\frac{1}{N}\sum_{n=1}^{N}y_n^2\boldsymbol{x}_q$. Future work could analyze the gradient descent dynamics of the model described by Equation (91), building on our results on the dynamics of in-context learning to explore its interactions with in-weight learning.

Reproducibility

Code reproducing our main results is available at GitHub: `https://github.com/yedizhang/linattn-icl`



(a) 0% fixed task (in-context) (b) 20% fixed tasks (c) 40% fixed tasks

(d) 60% fixed tasks (e) 80% fixed tasks (f) 100% fixed tasks (in-weight)

Figure 14. Dynamics of in-context and in-weight learning in linear attention with merged key and query. The training set is the same as the in-context linear regression task described in Section 2.1 except that a portion of the task vectors $\boldsymbol{w}$ are fixed. The portion of fixed task vectors indicates how much training samples can be fitted with the in-weight learning solution, that is memorizing the fixed task vector. The in-context learning test loss is evaluated on test sequences whose task vectors are all sampled from $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$. The in-weight learning test loss is evaluated on test sequences whose task vector is the same fixed task vector from the training set. Here $D = 4, N = 31, H = 8, \boldsymbol{\Lambda} = \boldsymbol{I}/D$.