Training Dynamics of In-Context Learning in Linear Attention

Yedi Zhang YEDI@GATSBY.UCL.AC.UK
Gatsby Unit, University College London

Aaditya K. Singh AADITYA.SINGH.21@UCL.AC.UK
Gatsby Unit, University College London

Peter E. Latham* PEL@GATSBY.UCL.AC.UK
Gatsby Unit, University College London

Andrew Saxe* A.SAXE@UCL.AC.UK
Gatsby Unit & SWC, University College London

*Co-senior authors

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 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 progressively improve during the gradient descent training of multi-head linear self-attention.

1. Introduction

Self-attention-based models, such as transformers [58], exhibit a remarkable ability known as in-context learning [10]. 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. [43] 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 [17, 43, 46, 52], and can also occur more broadly in transformer training dynamics [12, 25, 40]. Furthermore, Singh et al. [51] 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 [5, 11, 24, 41, 45, 53]. 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 self-attention [60] trained on an in-context linear regression task [20]. 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 [2, 4, 20, 60]. The linear attention model, which has been used in many prior studies [2, 16,

18, 19, 36–38, 49, 60, 62–64], reproduces key optimization properties of practical transformers [3] 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.

We study the common parametrizations of multi-head linear attention with low-rank key and query matrices. We specify the fixed points in the loss landscape, as well as how gradient descent training dynamics traverses the landscape. Our findings are summarized as follows: (i) We specify exponentially many fixed points in the function space. (ii) We show saddle-to-saddle training dynamics in training from small initialization and reduce the high-dimensional training dynamics to scalar ordinary differential equations through an ansatz. We demonstrate the rank of the key and query weights affects the dynamics by shortening the duration of certain plateaus. (iii) We identify the in-context algorithm of the converged and early stopped models. When training early stops during the $(m+1)$-th loss plateau, it approximately implements principal component regression in context with the first m principal components.

2. Preliminaries

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 [2, 13, 27, 64, 65], the input sequence is presented to the model as a matrix $\boldsymbol{X}$, defined as

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

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 $\boldsymbol{\Lambda}$, that is $\boldsymbol{x}_{\mu,n}, \boldsymbol{x}_{\mu,q} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Lambda}), n = 1, \cdots, N, \mu = 1, \cdots, P$. 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 $\boldsymbol{x}_q$ [20]. 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 Linear Attention

The multi-head linear attention takes the matrix $\boldsymbol{X}$ as input and returns a matrix of the same size,

$$\begin{aligned} \mathsf{ATTN}(\boldsymbol{X}) &= \boldsymbol{X} + \sum_{i=1}^H \frac{1}{N} \boldsymbol{W}_i^V \boldsymbol{X}\boldsymbol{X}^\top {\boldsymbol{W}_i^K}^\top \boldsymbol{W}_i^Q \boldsymbol{X} \\ &= \boldsymbol{X} + \sum_{i=1}^H \begin{bmatrix} * & * \\ \boldsymbol{v}_i^\top & v_i \end{bmatrix} \frac{\boldsymbol{X}\boldsymbol{X}^\top}{N} \begin{bmatrix} \boldsymbol{k}_{i,1} & \cdots & \boldsymbol{k}_{i,R} \\ k_{i,1} & \cdots & k_{i,R} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_{i,1}^\top & * \\ \vdots & \vdots \\ \boldsymbol{q}_{i,R}^\top & * \end{bmatrix} \boldsymbol{X} \end{aligned}$$

where H is the number of heads, 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. We write the value, key, and query weights in block form, whose entries have dimensionalities $v_i, k_{i,r} \in \mathbb{R}$ and $\boldsymbol{v}_i, \boldsymbol{k}_{i,r}, \boldsymbol{q}_{i,r} \in \mathbb{R}^D$ $(r = 1, \cdots, R)$. Following [2, 27, 34, 64], we initialize $\boldsymbol{v}_i = \boldsymbol{0}, k_{i,r} = 0$ as they are not required for achieving global minimum loss in our setting. These weights remain zero throughout training (see Appendix E.1). With this initialization, the multi-head linear attention computes

$$\hat{y}_q \equiv \mathsf{ATTN}(\boldsymbol{X})_{D+1,N+1} = \sum_{i=1}^{H} \sum_{r=1}^{R} v_i \boldsymbol{\beta}^\top \boldsymbol{k}_{i,r} \boldsymbol{q}_{i,r}^\top \boldsymbol{x}_q, \quad \text{where } \boldsymbol{\beta} \equiv \frac{1}{N} \sum_{n=1}^{N} y_n \boldsymbol{x}_n. \tag{2}$$

We only take the bottom right entry of the output matrix because it is the prediction for y_q.

We train the multi-head linear attention model with gradient descent on mean square error loss, that is $\mathcal{L} = \mathbb{E}(y_q - \hat{y}_q)^2$. We analyze the gradient flow dynamics on the loss, given by $\tau \frac{\mathrm{d}\boldsymbol{W}}{\mathrm{d}t} = -\frac{1}{2} \frac{\partial \mathcal{L}}{\partial \boldsymbol{W}}$, where τ is the time constant.

3. Multi-Head Linear Attention with Rank-One Key and Query

Because the multi-head linear attention with rank-one key and query 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 4. When $R = 1$, the model definition in Eq. (2) simplifies to $\hat{y}_q = \sum_{i=1}^{H} v_i \boldsymbol{\beta}^\top \boldsymbol{k}_i \boldsymbol{q}_i^\top \boldsymbol{x}_q$.

3.1. Loss Landscape: Exponentially Many Fixed Points

The gradient flow training dynamics of linear attention with rank-one key and query is given by Eq. (24) in Appendix D.2. The dynamics contains 2^D fixed points in the function space of $\mathsf{ATTN}(\boldsymbol{X})_{D+1,N+1}$. We specify them below.

Let $\lambda_1, \cdots, \lambda_D$ be the eigenvalues of the covariance matrix $\boldsymbol{\Lambda}$ 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,

$$\mathcal{M}(\mathcal{S}_m) = \left\{ (v, \boldsymbol{k}, \boldsymbol{q})_{1:H} \middle| \text{conditions (C1)-(C3)} \right\}, \quad \text{where } \mathcal{S}_m \subseteq \{1, 2, \cdots, D\}, |\mathcal{S}_m| = m. \tag{3}$$

Here the set $\mathcal{S}_m$ specifies the indices of learned eigenvectors. The three conditions for Eq. (3) are:

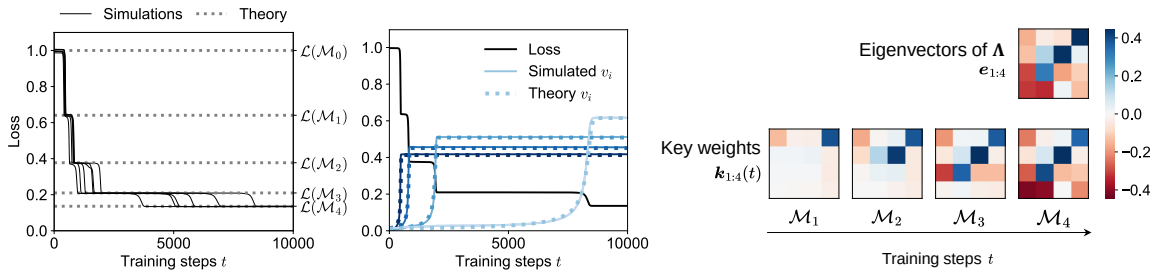
(C1) The heads sum to fit the eigenvectors with indices in the set $\mathcal{S}_m$

$$\sum_{i=1}^{H} v_i \boldsymbol{k}_i \boldsymbol{q}_i^\top = \sum_{d \in \mathcal{S}_m} \lambda_d^{-1} \left(1 + \frac{1 + \mathtt{tr}(\boldsymbol{\Lambda})/\lambda_d}{N} \right)^{-1} \boldsymbol{e}_d \boldsymbol{e}_d^\top. \tag{4}$$

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

(C3) For a head with $v_i = 0$, 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 Eq. (3), 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 Eq. (4) and thus a different function, $\mathsf{ATTN}(\boldsymbol{X})_{D+1,N+1}$. Hence, the gradient flow dynamics in Eq. (24) has 2^D fixed points in the function space.



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

Figure 1: Multi-head linear attention with 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 Eq. (5) (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 Eq. (7) 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. 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).

3.2. 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 Fig. 1(*a*), 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 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 D.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 Fig. 1(*a*).

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

$$\mathcal{L}(\mathcal{M}_m) = \mathtt{tr}(\mathbf{\Lambda}) - \sum_{d=1}^{m} \lambda_d \left(1 + \frac{1 + \mathtt{tr}(\mathbf{\Lambda})/\lambda_d}{N}\right)^{-1}. \tag{5}$$

In the limit of a large sequence length N, Eq. (5) is highly interpretable: it is the sum of the eigenvalues associated with the remaining unlearned eigenvectors, $\lim_{N\to\infty} \mathcal{L}(\mathcal{M}_m) = \sum_{d=m+1}^{D} \lambda_d$. Thus, the loss decreases by approximately λ_m during the m-th abrupt loss drop. We plot Eq. (5) as dashed gray lines in Fig. 1(*a*) 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 Eq. (3). As shown in Fig. 1(*c*), the key and query weights in a head grow in scale and align with a new eigenvector of the input token covariance $\mathbf{\Lambda}$ during each abrupt loss drop. Based on simulations in Fig. 1 and derivations in Appendices D.5 and D.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

$$\boldsymbol{k}_i = \boldsymbol{q}_i = v_i \boldsymbol{e}_i,\ v_i = \lambda_i^{-\frac{1}{3}} \left(1 + \frac{1 + \mathtt{tr}(\mathbf{\Lambda})/\lambda_i}{N}\right)^{-\frac{1}{3}}, \quad 1 \le i \le m, \tag{6a}$$

$$\boldsymbol{k}_i = \boldsymbol{q}_i = v_i(t) \boldsymbol{e}_{m+1}, \quad i = m+1, \tag{6b}$$

$$\boldsymbol{k}_i = \boldsymbol{q}_i = \boldsymbol{0}, v_i = 0, \quad m+2 \le i \le H, \tag{6c}$$

where $v_{m+1}(t)$ is small during the $(m+1)$-th loss plateau and grows during the $(m+1)$-th abrupt loss drop. Eq. (6) 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 D.7. 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 $v_i(t), i = m+1$:

$$\tau \dot{v}_i = \lambda_{m+1}^2 v_i^2 - \lambda_{m+1}^3 \left(1 + \frac{1 + \mathtt{tr}(\mathbf{\Lambda})/\lambda_{m+1}}{N}\right) v_i^5. \tag{7}$$

Eq. (7) is a separable differential equation but does not admit a general analytical solution of $v_{m+1}(t)$ in terms of t (see Eq. (37)). Nonetheless, it greatly simplifies the high-dimensional dynamics in Eq. (24) 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 Fig. 1(*b*), we compare the numerical solution of Eq. (7) with the value weights trajectories in the simulation and find excellent agreement.

In summary, the loss trajectory of linear attention with 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 $\boldsymbol{e}_m$.

3.3. ICL Algorithm: Principal Component Regression

When the linear attention model is at a fixed point in $\mathcal{M}_m$, based on Eq. (4), the model implements

$$\mathsf{ATTN}(\boldsymbol{X})_{D+1,N+1} = \boldsymbol{\beta}^\top \sum_{d=1}^{m} \lambda_d^{-1} \left(1 + \frac{1 + \mathtt{tr}(\mathbf{\Lambda})/\lambda_d}{N}\right)^{-1} \boldsymbol{e}_d \boldsymbol{e}_d^\top \boldsymbol{x}_q. \tag{8}$$

In the limit of a large sequence length N, Eq. (8) simplifies and can be interpreted as principal component regression in context with m principal components

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

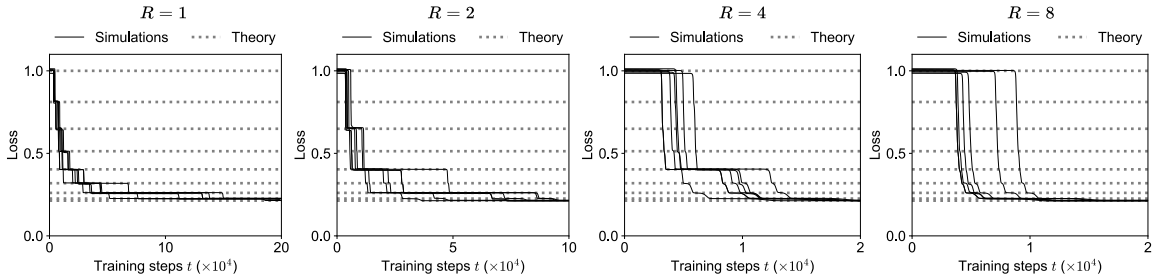


Figure 2: Multi-head linear attention with low-rank key and query exhibits saddle-to-saddle dynamics, with the duration of plateaus depending on the rank R. Solid black curves are loss trajectories from six random initializations. Dashed gray lines mark the loss values predicted by Eq. (5) at nine fixed 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 $\lambda_d \propto d^{-1}$.

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 squares linear regression in context.

4. Linear Attention with Low-Rank Key and Query

The linear attention model with 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 E.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 Fig. 2 (see Fig. 8 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 Eq. (2). 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 Fig. 7. See Appendix E.4 for further details.

Reproducibility

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

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.

References

[1] Amirhesam Abedsoltan, Adityanarayanan Radhakrishnan, Jingfeng Wu, and Mikhail Belkin. Context-scaling versus task-scaling in in-context learning, 2024. URL https://arxiv.org/abs/2410.12783.

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

[3] Kwangjun Ahn, Xiang Cheng, Minhak Song, Chulhee Yun, Ali Jadbabaie, and Suvrit Sra. 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.

[4] Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 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.

[5] Suraj Anand, Michael A. Lepori, Jack Merullo, and Ellie Pavlick. 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.

[6] Davoud Ataee Tarzanagh, Yingcong Li, Xuechen Zhang, and Samet Oymak. Max-margin token selection in attention mechanism. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 48314–48362. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/970f59b22f4c72aec75174aae63c7459-Paper-Conference.pdf.

[7] Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 57125–57211. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/b2e63e36c57e153b9015fece2352a9f9-Paper-Conference.pdf.

[8] Enric Boix-Adsera, Etai Littwin, Emmanuel Abbe, Samy Bengio, and Joshua Susskind. Transformers learn through gradual rank increase. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 24519–24551. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/4d69c1c057a8bd570ba4a7b71aae8331-Paper-Conference.pdf.

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

[10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[11] Bryan Chan, Xinyi Chen, András György, and Dale Schuurmans. Toward understanding in-context vs. in-weight learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=aKJr5NnN8U.

[12] Angelica Chen, Ravid Shwartz-Ziv, Kyunghyun Cho, Matthew L Leavitt, and Naomi Saphra. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in MLMs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=MO5PiKHELW.

[13] Siyu Chen, Heejune Sheen, Tianhao Wang, and Zhuoran Yang. Training dynamics of multi-head softmax attention for in-context learning: Emergence, convergence, and optimality. In Shipra Agrawal and Aaron Roth, editors, *Proceedings of Thirty Seventh Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pages 4573–4573. PMLR, 30 Jun–03 Jul 2024. URL https://proceedings.mlr.press/v247/siyu24a.html.

[14] Siyu Chen, Heejune Sheen, Tianhao Wang, and Zhuoran Yang. Unveiling induction heads: Provable training dynamics and feature learning in transformers. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 66479–66567. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/7aae9e3ec211249e05bd07271a6b1441-Paper-Conference.pdf.

[15] Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *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.

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

[17] Ezra Edelman, Nikolaos Tsilivis, Benjamin Edelman, Eran Malach, and Surbhi Goel. The evolution of statistical induction heads: In-context learning markov chains. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 64273–64311. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/75b0edb869e2cd509d64d0e8ff446bc1-Paper-Conference.pdf.

[18] Spencer Frei and Gal Vardi. 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.

[19] Deqing Fu, Tian-qi Chen, Robin Jia, and Vatsal Sharan. Transformers learn to achieve second-order convergence rates for in-context linear regression. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 98675–98716. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/b2d4051f03a7038a2771dfbbe5c7b54e-Paper-Conference.pdf.

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

[21] Khashayar Gatmiry, Nikunj Saunshi, Sashank J. Reddi, Stefanie Jegelka, and Sanjiv Kumar. On the role of depth and looping for in-context learning with task diversity, 2024. URL https://arxiv.org/abs/2410.21698.

[22] Borjan Geshkovski, Hugo Koubbi, Yury Polyanskiy, and Philippe Rigollet. Dynamic metastability in the self-attention model, 2024. URL https://arxiv.org/abs/2410.06833.

[23] Jianliang He, Xintian Pan, Siyu Chen, and Zhuoran Yang. In-context linear regression demystified: Training dynamics and mechanistic interpretability of multi-head softmax attention. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=3TM3fxwTps.

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

[25] David T Hoffmann, Simon Schrodi, Jelena Bratulić, Nadine Behrmann, Volker Fischer, and Thomas Brox. Eureka-moments in transformers: Multi-step tasks reveal softmax induced optimization problems. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 18409–18438. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/hoffmann24a.html.

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

[27] Yu Huang, Yuan Cheng, and Yingbin Liang. In-context convergence of transformers. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 19660–19722. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/huang24d.html.

[28] Yu Huang, Zixin Wen, Yuejie Chi, and Yingbin Liang. A theoretical analysis of self-supervised learning for vision transformers. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Antib6Uovh.

[29] Muhammed Emrullah Ildiz, Yixiao Huang, Yingcong Li, Ankit Singh Rawat, and Samet Oymak. From self-attention to Markov models: Unveiling the dynamics of generative transformers. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 20955–20982. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/ildiz24a.html.

[30] Uijeong Jang, Jason D. Lee, and Ernest K. Ryu. LoRA training in the NTK regime has no spurious local minima. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 21306–21328. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/jang24d.html.

[31] Samy Jelassi, Michael Sander, and Yuanzhi Li. Vision transformers provably learn spatial structure. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 37822–37836. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/f69707de866eb0805683d3521756b73f-Paper-Conference.pdf.

[32] Jiarui Jiang, Wei Huang, Miao Zhang, Taiji Suzuki, and Liqiang Nie. Unveil benign overfitting for transformer in vision: Training dynamics, convergence, and generalization. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 135464–135625. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/f49287371916715b9209fa41a275851e-Paper-Conference.pdf.

[33] Aaron Alvarado Kristanto Julistiono, Davoud Ataee Tarzanagh, and Navid Azizan. Optimizing attention with mirror descent: Generalized max-margin token selection, 2024. URL https://arxiv.org/abs/2410.14581.

[34] Juno Kim and Taiji Suzuki. Transformers learn nonlinear features in context: Nonconvex mean-field dynamics on the attention landscape. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 24527–24561. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/kim24af.html.

[35] Andrew Kyle Lampinen, Stephanie C. Y. Chan, Aaditya K. Singh, and Murray Shanahan. The broader spectrum of in-context learning, 2024. URL https://arxiv.org/abs/2412.03782.

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

[37] Yue M. Lu, Mary I. Letey, Jacob A. Zavatone-Veth, Anindita Maiti, and Cengiz Pehlevan. Asymptotic theory of in-context learning by linear attention, 2024. URL https://arxiv.org/abs/2405.11751.

[38] Arvind V. Mahankali, Tatsunori Hashimoto, and Tengyu Ma. 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.

[39] Ashok Vardhan Makkuva, Marco Bondaschi, Adway Girish, Alliot Nagle, Hyeji Kim, Michael Gastpar, and Chanakya Ekbote. Local to global: Learning dynamics and effect of initialization for transformers. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 86243–86308. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/9cdb4f8c4dfa13284d2d5a6e7853e5a2-Paper-Conference.pdf.

[40] Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=9XFSbDPmdW.

[41] Alex Nguyen and Gautam Reddy. 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.

[42] Eshaan Nichani, Alex Damian, and Jason D. Lee. How transformers learn causal structure with gradient descent. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 38018–38070. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/nichani24a.html.

[43] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads, 2022. URL https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

[44] Felix Otto and Maria G. Reznikoff. 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.

[45] Core Francisco Park, Ekdeep Singh Lubana, and Hidenori Tanaka. Algorithmic phases of in-context learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=XgH1wfHSX8.

[46] Gautam Reddy. 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.

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

[48] Riccardo Rende, Federica Gerace, Alessandro Laio, and Sebastian Goldt. A distributional simplicity bias in the learning dynamics of transformers. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 96207–96228. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/ae6c81a39079ddeb88b034b6ef18c7fe-Paper-Conference.pdf.

[49] Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9355–9366. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/schlag21a.html.

[50] Heejune Sheen, Siyu Chen, Tianhao Wang, and Harrison H. Zhou. Implicit regularization of gradient flow on one-layer softmax attention, 2024. URL https://arxiv.org/abs/2403.08699.

[51] Aaditya K Singh, Stephanie Chan, Ted Moskovitz, Erin Grant, Andrew Saxe, and Felix Hill. The transient nature of emergent in-context learning in transformers. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 27801–27819. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/58692a1701314e09cbd7a5f5f3871cc9-Paper-Conference.pdf.

[52] Aaditya K Singh, Ted Moskovitz, Felix Hill, Stephanie C.Y. Chan, and Andrew M Saxe. What needs to go right for an induction head? A mechanistic study of in-context learning circuits and their formation. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 45637–45662. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/singh24c.html.

[53] Aaditya K Singh, Ted Moskovitz, Sara Dragutinović, Felix Hill, Stephanie C.Y. Chan, and Andrew M Saxe. Strategy coopetition explains the emergence and transience of in-context learning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=esBoQFmD7v.

[54] Bingqing Song, Boran Han, Shuai Zhang, Jie Ding, and Mingyi Hong. Unraveling the gradient descent dynamics of transformers. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 92317–92351. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/a7d36e5cb41a1f21c46db25cb1aafab9-Paper-Conference.pdf.

[55] Davoud Ataee Tarzanagh, Yingcong Li, Christos Thrampoulidis, and Samet Oymak. Transformers as support vector machines, 2024. URL https://arxiv.org/abs/2308.16898.

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

[57] Bhavya Vasudeva, Puneesh Deora, and Christos Thrampoulidis. 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.

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *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.

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

[60] Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, Joao Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 35151–35174. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/von-oswald23a.html.

[61] Mingze Wang, Ruoxi Yu, Weinan E, and Lei Wu. How transformers implement induction heads: Approximation and optimization analysis, 2024. URL https://arxiv.org/abs/2410.11474.

[62] Jingfeng Wu, Difan Zou, Zixiang Chen, Vladimir Braverman, Quanquan Gu, and Peter Bartlett. 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.

[63] Morris Yau, Ekin Akyürek, Jiayuan Mao, Joshua B. Tenenbaum, Stefanie Jegelka, and Jacob Andreas. Learning linear attention in polynomial time, 2024. URL https://arxiv.org/abs/2410.10101.

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

[65] Ruiqi Zhang, Jingfeng Wu, and Peter Bartlett. In-context learning of a linear transformer block: Benefits of the mlp component and one-step gd initialization. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 18310–18361. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/20b6b87ca17792337f414d948af7b0e8-Paper-Conference.pdf.

Appendix A. Related Work

A.1. Theory of Linear Attention

Recent theoretical research on linear attention has investigated its expressivity [21, 59], learnability [63], loss landscape [36, 38], convergence [19, 47, 64, 65], and generalization [1, 16, 18, 37, 38, 62]. The seminal work by Zhang et al. [64] 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.

A.2. Theory of Training Dynamics In Attention Models

A 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 [14, 42, 56, 61], restricted weights [8, 13, 17, 48], and specifically chosen datasets [27]. 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.

A concurrent work by Geshkovski et al. [22] studies saddle-to-saddle-like dynamics in softmax attention models following a mathematical framework for slow motion of gradient flows [44]. A subsequent work by He et al. [23] examines the training dynamics of softmax attention trained on the in-context linear regression task; that is, the case we briefly touch on in Fig. 3.

A broader body of theoretical literature have explored the transformers training dynamics but addressed different problem from ours, such as the effect of initialization [39], convergence results [26, 54], sample complexity guarantees [29], scaling limits [9], and implicit regularization [6, 33, 50, 55, 57]. Other studies considered special training regimes, such as the neural tangent kernel regime [30] and the mean-field regime [34]. A few works focused on vision transformers [28, 31, 32]. In contrast, our works focuses on characterizing the process of training and the development of in-context learning abilities over time.

Appendix B. Discussion

We study the gradient flow training dynamics of multi-head linear attention and demonstrated how it acquires ICL abilities in training. 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. Building on prior findings showing that transformers can implement different forms of ICL [7, 35], we show that different forms of ICL can indeed emerge in gradient descent training. We thus characterize how the linear attention model develops increasingly sophisticated ICL abilities in gradient descent training.

B.1. Softmax Attention

We empirically find that the training dynamics of linear attention also occur in their softmax counterparts. Fig. 3 follows the same setup as Fig. 1 for linear attention, with the only difference being

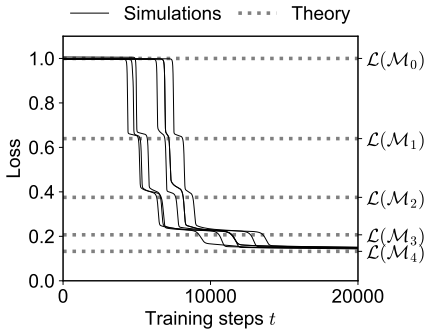


Figure 3: Loss trajectories of softmax attention with rank-one key and query. Six runs from different random initialization are plotted. Similar to the linear attention case, softmax attention exhibits multiple loss drops. The dataset and model setup are the same as Fig. 1 except adding the softmax activation function.

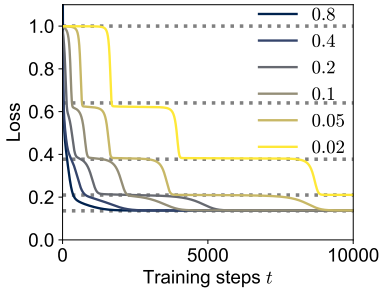


Figure 4: Loss trajectories of linear attention with varying initialization scales. The colors indicate the initialization scale. Increasing the initialization scale shortens the plateaus. With small initialization, the models are in the rich feature learning regime, exhibiting abrupt sigmoid-shaped 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 [43, Argument 1].

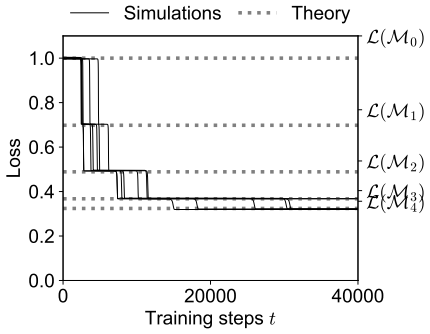


Figure 5: Loss trajectories of linear attention trained with next token prediction loss in Eq. (9). In this case, the models are trained on sequences of varying lengths, which they can handle due to the $1/N$ scaling factor in Eq. (2).

adding the softmax activation function for the attention calculation. We observe that softmax attention with 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.

B.2. Effect of Initialization

Having analyzed the small initialization case, we now examine how the initialization scale affects training dynamics. We vary the initialization scale of the linear attention models and plot their loss trajectories in Fig. 4. We see that increasing initialization shortens all plateaus between successive abrupt loss drops. At the largest initialization, the model exhibits an exponential-shaped loss decay – a hallmark of lazy learning [15]. 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 Fig. 4, 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 [43, Argument 1]. Our analysis focuses on the rich learning regime and provides analytical insight into such phenomena, which we believe is a first step toward understanding dynamics in naturalistic settings.

B.3. Varying Context Lengths

In our main results, we consider a fixed context length N, since 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

$$\mathcal{L}_{\text{ntp}} = \mathbb{E}\left[\frac{1}{N}\sum_{n=2}^{N+1}(y_n - \hat{y}_n)^2\right], \tag{9}$$

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 influence our results through a statistic, $\mathbb{E}(1/N)$.

Derivations in Appendix D.3 show that the converged model implements

$$\mathsf{ATTN}(\boldsymbol{X})_{D+1,N+1} = \boldsymbol{\beta}^\top \boldsymbol{\Lambda}\left[\mathbb{E}\left(\frac{1}{N}\boldsymbol{x}_n\boldsymbol{x}_n^\top\right)^2\right]^{-1}\boldsymbol{x}_q. \tag{10}$$

Substituting Eq. (16) into Eq. (10), we obtain

$$\mathsf{ATTN}(\boldsymbol{X})_{D+1,N+1} = \boldsymbol{\beta}^\top\left[\boldsymbol{\Lambda} + \mathbb{E}\left(\frac{1}{N}\right)(\boldsymbol{\Lambda} + \mathtt{tr}(\boldsymbol{\Lambda})\boldsymbol{I})\right]^{-1}\boldsymbol{x}_q. \tag{11}$$

The distribution of context lengths only influences Eq. (11) through the expectation $\mathbb{E}(1/N)$. For a fixed context length, $\mathbb{E}(1/N) = 1/N$. 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, the fixed point condition (C1) takes the form

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

where the expectation $\mathbb{E}(1/N)$ reduces to $1/N$ in the fixed context length case as in Eq. (4). Consequently, when the model is at a fixed point in $\mathcal{M}_m$, the loss value is

$$\mathcal{L}(\mathcal{M}_m) = \mathtt{tr}(\boldsymbol{\Lambda}) - \sum_{d=1}^{m} \lambda_d \left[1 + E\left(\frac{1}{N}\right)(1 + \mathtt{tr}(\boldsymbol{\Lambda})/\lambda_d) \right]^{-1}, \tag{12}$$

which reduces to Eq. (5) when $\mathbb{E}(1/N) = 1/N$.

We train the linear attention model with the next token prediction loss as in Eq. (9) and plot the loss trajectories in Fig. 5. The loss trajectories are qualitatively similar to those in Fig. 1(*a*), except for the different loss values during the plateaus. We plot the loss values computed from Eq. (12) as dashed gray lines and find they match the plateaus of the simulated loss trajectories well.

Appendix 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 Eq. (2). We additionally denote the in-context covariance of $\boldsymbol{x}_n$ in a sequence as $\hat{\boldsymbol{\Lambda}}$

$$\hat{\boldsymbol{\Lambda}} \equiv \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^\top. \tag{13}$$

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

$$\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^\top = \begin{bmatrix} \frac{1}{N}\left(\boldsymbol{x}_q \boldsymbol{x}_q^\top + \sum_n \boldsymbol{x}_n \boldsymbol{x}_n^\top\right) & \frac{1}{N}\sum_n \boldsymbol{x}_n y_n \\ \frac{1}{N}\sum_n y_n \boldsymbol{x}^\top & \frac{1}{N}\sum_n y_n^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{N}\boldsymbol{x}_q \boldsymbol{x}_q^\top + \hat{\boldsymbol{\Lambda}} & \boldsymbol{\beta} \\ \boldsymbol{\beta}^\top & \boldsymbol{w}^\top \hat{\boldsymbol{\Lambda}} \boldsymbol{w} \end{bmatrix}. \tag{14}$$

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

$$\boldsymbol{\beta} = \hat{\boldsymbol{\Lambda}} \boldsymbol{w}. \tag{15}$$

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

$$\begin{aligned}
\mathbb{E}\left(\hat{\mathbf{\Lambda}}^2\right) &\equiv \mathbb{E}\left(\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n\boldsymbol{x}_n^\top\right)^2 \\
&= \mathbb{E}\left(\frac{N^2-N}{N^2}\sum_{n\neq n'}\boldsymbol{x}_n\boldsymbol{x}_n^\top\boldsymbol{x}_{n'}\boldsymbol{x}_{n'}^\top + \frac{N}{N^2}\sum_{n=1}^{N}\boldsymbol{x}_n\boldsymbol{x}_n^\top\boldsymbol{x}_n\boldsymbol{x}_n^\top\right) \\
&= \mathbb{E}_N\left(\frac{N-1}{N}\right)\mathbb{E}_{\boldsymbol{x}}\left(\boldsymbol{x}_n\boldsymbol{x}_n^\top\right)\mathbb{E}_{\boldsymbol{x}}\left(\boldsymbol{x}_{n'}\boldsymbol{x}_{n'}^\top\right) + \mathbb{E}_N\left(\frac{1}{N}\right)\mathbb{E}_{\boldsymbol{x}}\left(\boldsymbol{x}_n\boldsymbol{x}_n^\top\boldsymbol{x}_n\boldsymbol{x}_n^\top\right) \\
&= \left(1-\mathbb{E}\left(\frac{1}{N}\right)\right)\mathbf{\Lambda}^2 + \mathbb{E}\left(\frac{1}{N}\right)\left(2\mathbf{\Lambda}^2 + \mathtt{tr}(\mathbf{\Lambda})\mathbf{\Lambda}\right) \\
&= \mathbf{\Lambda}^2 + \mathbb{E}\left(\frac{1}{N}\right)\left(\mathbf{\Lambda} + \mathtt{tr}(\mathbf{\Lambda})\boldsymbol{I}\right)\mathbf{\Lambda}. \qquad (16)
\end{aligned}$$

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, Eq. (16) simplifies to

$$\mathbb{E}\left(\hat{\mathbf{\Lambda}}^2\right) = \mathbf{\Lambda}^2 + \frac{\mathbf{\Lambda} + \mathtt{tr}(\mathbf{\Lambda})\boldsymbol{I}}{N}\mathbf{\Lambda}. \qquad (17)$$

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

$$a_d = \left[\left(1+\frac{1}{N}\right)\lambda_d^2 + \frac{\mathtt{tr}(\mathbf{\Lambda})}{N}\lambda_d\right] = \lambda_d^2\left(1 + \frac{1+\mathtt{tr}(\mathbf{\Lambda})/\lambda_d}{N}\right). \qquad (18)$$

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

$$\mathbb{E}\left(\hat{\mathbf{\Lambda}}^2\right) = \sum_{d=1}^{D} a_d\boldsymbol{e}_d\boldsymbol{e}_d^\top. \qquad (19)$$

C.2. Initialization

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

$$v_i \sim \mathcal{N}(0, w_{\text{init}}^2/H), \quad k_{i,r}^d \sim \mathcal{N}(0, w_{\text{init}}^2/HRD), \quad q_{i,r}^d \sim \mathcal{N}(0, w_{\text{init}}^2/HRD). \qquad (20)$$

At initialization, the following ℓ^2 norms are

$$\sqrt{\sum_{i=1}^{H} v_i^2}, \sqrt{\sum_{i=1}^{H}\sum_{r=1}^{R}\|\boldsymbol{k}_{i,r}\|^2}, \sqrt{\sum_{i=1}^{H}\sum_{r=1}^{R}\|\boldsymbol{q}_{i,r}\|^2} \sim O(w_{\text{init}}). \qquad (21)$$

Appendix D. Linear Attention with Rank-One Key and Query

D.1. Justification for Zero Blocks Assumption

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

D.2. Gradient Flow Equations

We here derive the gradient flow dynamics for linear attention with rank-one key and query introduced in Eq. (24).

Based on the gradient flow training rule, $\tau \frac{\mathrm{d}\boldsymbol{W}}{\mathrm{d}t} = -\frac{1}{2}\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}}$, the gradient flow dynamics for the value, key, and query weights in the i-th head are

$$\tau \dot{v}_i = \boldsymbol{k}_i^\top \mathbb{E}\left(\boldsymbol{\beta}(y_q - \hat{y}_q)\boldsymbol{x}_q^\top\right)\boldsymbol{q}_i, \tag{22a}$$

$$\tau \dot{\boldsymbol{k}}_i = v_i \mathbb{E}\left(\boldsymbol{\beta}(y_q - \hat{y}_q)\boldsymbol{x}_q^\top\right)\boldsymbol{q}_i, \tag{22b}$$

$$\tau \dot{\boldsymbol{q}}_i = v_i \mathbb{E}\left(\boldsymbol{x}_q(y_q - \hat{y}_q)\boldsymbol{\beta}^\top\right)\boldsymbol{k}_i. \tag{22c}$$

We calculate the common term in Eq. (22), that is

$$\begin{aligned}
\mathbb{E}\left(\boldsymbol{\beta}(y_q - \hat{y}_q)\boldsymbol{x}_q^\top\right) &= \mathbb{E}\left[\boldsymbol{\beta}\left(\boldsymbol{w}^\top \boldsymbol{x}_q - \sum_{i=1}^H v_i \boldsymbol{\beta}^\top \boldsymbol{k}_i \boldsymbol{q}_i^\top \boldsymbol{x}_q\right)\boldsymbol{x}_q^\top\right] \\
&= \mathbb{E}\left[\hat{\boldsymbol{\Lambda}}\boldsymbol{w}\boldsymbol{w}^\top\left(\boldsymbol{I} - \sum_{i=1}^H v_i \hat{\boldsymbol{\Lambda}}\boldsymbol{k}_i\boldsymbol{q}_i^\top\right)\boldsymbol{x}_q\boldsymbol{x}_q^\top\right] \\
&= \mathbb{E}\left(\hat{\boldsymbol{\Lambda}}\right)\mathbb{E}_{\boldsymbol{w}}\left(\boldsymbol{w}\boldsymbol{w}^\top\right)\mathbb{E}_{\boldsymbol{x}_q}\left(\boldsymbol{x}_q\boldsymbol{x}_q^\top\right) - \mathbb{E}\left(\hat{\boldsymbol{\Lambda}}\boldsymbol{w}\boldsymbol{w}^\top\hat{\boldsymbol{\Lambda}}\right)\sum_{i=1}^H v_i\boldsymbol{k}_i\boldsymbol{q}_i^\top\mathbb{E}_{\boldsymbol{x}_q}\left(\boldsymbol{x}_q\boldsymbol{x}_q^\top\right) \\
&= \boldsymbol{\Lambda}^2 - \mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right)\sum_{i=1}^H v_i\boldsymbol{k}_i\boldsymbol{q}_i^\top\boldsymbol{\Lambda}
\end{aligned} \tag{23}$$

Substituting Eq. (23) into Eq. (22), we arrive at the same equations as Eq. (24) in the main text

$$\tau \dot{v}_i = \boldsymbol{k}_i^\top\left(\boldsymbol{\Lambda}^2 - \mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right)\sum_{i'=1}^H v_{i'}\boldsymbol{k}_{i'}\boldsymbol{q}_{i'}^\top\boldsymbol{\Lambda}\right)\boldsymbol{q}_i, \tag{24a}$$

$$\tau \dot{\boldsymbol{k}}_i = v_i\left(\boldsymbol{\Lambda}^2 - \mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right)\sum_{i'=1}^H v_{i'}\boldsymbol{k}_{i'}\boldsymbol{q}_{i'}^\top\boldsymbol{\Lambda}\right)\boldsymbol{q}_i, \tag{24b}$$

$$\tau \dot{\boldsymbol{q}}_i = v_i\left(\boldsymbol{\Lambda}^2 - \boldsymbol{\Lambda}\sum_{i'=1}^H v_{i'}\boldsymbol{k}_{i'}\boldsymbol{q}_{i'}^\top\mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right)\right)\boldsymbol{k}_i. \tag{24c}$$

where the data statistics $\mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right)$ is calculated in Eq. (17).

D.3. Fixed Points

We prove that the fixed points given in Eq. (3) are valid.

Proof When the model is at a fixed point in set $\mathcal{M}(\mathcal{S}_m)$, it satisfies Eq. (4). Eq. (4) can be rewritten using a_d (defined in Eq. (18)) as

$$\sum_{i=1}^{H} v_i \boldsymbol{k}_i \boldsymbol{q}_i^\top = \sum_{d \in \mathcal{S}_m} \frac{\lambda_d}{a_d} \boldsymbol{e}_d \boldsymbol{e}_d^\top. \tag{25}$$

Using Eqs. (19) and (25), we can simplify a common term in the gradient descent dynamics in Eq. (24) to

$$\begin{aligned} \boldsymbol{\Lambda}^2 - \mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right) \sum_{i=1}^{H} v_i \boldsymbol{k}_i \boldsymbol{q}_i^\top \boldsymbol{\Lambda} &= \sum_{d=1}^{D} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top - \sum_{d'=1}^{D} a_{d'} \boldsymbol{e}_{d'} \boldsymbol{e}_{d'}^\top \sum_{d \in \mathcal{S}_m} \frac{\lambda_d}{a_d} \boldsymbol{e}_d \boldsymbol{e}_d^\top \boldsymbol{\Lambda} \\ &= \sum_{d=1}^{D} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top - \sum_{d \in \mathcal{S}_m} \lambda_d \boldsymbol{e}_d \boldsymbol{e}_d^\top \boldsymbol{\Lambda} \\ &= \sum_{d \notin \mathcal{S}_m} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top. \end{aligned} \tag{26}$$

Substituting Eq. (26) into Eq. (24), we obtain the gradient flow dynamics when the model is at a fixed point in $\mathcal{M}(\mathcal{S}_m)$

$$\tau \dot{v}_i = \boldsymbol{k}_i^\top \left(\sum_{d \notin \mathcal{S}_m} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top \right) \boldsymbol{q}_i, \tag{27a}$$

$$\tau \dot{\boldsymbol{k}}_i = v_i \left(\sum_{d \notin \mathcal{S}_m} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top \right) \boldsymbol{q}_i, \tag{27b}$$

$$\tau \dot{\boldsymbol{q}}_i = v_i \left(\sum_{d \notin \mathcal{S}_m} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top \right) \boldsymbol{k}_i. \tag{27c}$$

(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 Eq. (3), 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

$$\boldsymbol{k}_i = \sum_{d \in \mathcal{S}_m} b_d \boldsymbol{e}_d, \quad b_d \in \mathbb{R}, \tag{28a}$$

$$\boldsymbol{q}_i = \sum_{d \in \mathcal{S}_m} c_d \boldsymbol{e}_d, \quad c_d \in \mathbb{R}. \tag{28b}$$

Substituting Eq. (28) into the gradient flow dynamics given in Eq. (27), 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 Eqs. (27b) and (27c) 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 Eq. (3). 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 Eq. (28b). Substituting Eq. (28b) into the gradient of v_i given in Eq. (27a), 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 Eq. (3), the gradients of the weights are zero, meaning that the fixed points are valid. ■

D.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

$$\mathcal{L}(\mathcal{M}(\mathcal{S}_m)) = \mathtt{tr}(\boldsymbol{\Lambda}) - \sum_{d \in \mathcal{S}_m} \lambda_d \left(1 + \frac{1 + \mathtt{tr}(\boldsymbol{\Lambda})/\lambda_d}{N} \right)^{-1}. \tag{29}$$

Eq. (5) in the main text follows directly from Eq. (29) when taking $S_m = \{1, 2, \cdots, m\}$.

Proof We substitute Eqs. (19) and (25) into the mean square loss and obtain

$$\begin{aligned}
\mathcal{L}(\mathcal{M}(\mathcal{S}_m)) &= \mathbb{E}(y_q - \hat{y}_q)^2 \\
&= \mathbb{E}\left(\boldsymbol{w}^\top \boldsymbol{x}_q - \sum_{d\in\mathcal{S}_m} \frac{\lambda_d}{a_d} \boldsymbol{w}^\top \hat{\boldsymbol{\Lambda}} \boldsymbol{e}_d \boldsymbol{e}_d^\top \boldsymbol{x}_q\right)^2 \\
&= \mathbb{E}\left[\boldsymbol{x}_q^\top \left(\boldsymbol{I} - \sum_{d\in\mathcal{S}_m} \frac{\lambda_d}{a_d} \hat{\boldsymbol{\Lambda}} \boldsymbol{e}_d \boldsymbol{e}_d^\top\right) \mathbb{E}_{\boldsymbol{w}}(\boldsymbol{w}\boldsymbol{w}^\top) \left(\boldsymbol{I} - \sum_{d\in\mathcal{S}_m} \frac{\lambda_d}{a_d} \hat{\boldsymbol{\Lambda}} \boldsymbol{e}_d \boldsymbol{e}_d^\top\right) \boldsymbol{x}_q\right] \\
&= \mathbb{E}\left[\boldsymbol{x}_q^\top \left(\boldsymbol{I} - \sum_{d\in\mathcal{S}_m} \frac{\lambda_d}{a_d} \hat{\boldsymbol{\Lambda}} \boldsymbol{e}_d \boldsymbol{e}_d^\top\right) \left(\boldsymbol{I} - \sum_{d\in\mathcal{S}_m} \frac{\lambda_d}{a_d} \hat{\boldsymbol{\Lambda}} \boldsymbol{e}_d \boldsymbol{e}_d^\top\right) \boldsymbol{x}_q\right] \\
&= \mathbb{E}\left[\boldsymbol{x}_q^\top \left(\boldsymbol{I} - 2\sum_{d\in\mathcal{S}_m} \frac{\lambda_d}{a_d} \hat{\boldsymbol{\Lambda}} \boldsymbol{e}_d \boldsymbol{e}_d^\top + \left(\sum_{d\in\mathcal{S}_m} \frac{\lambda_d}{a_d} \hat{\boldsymbol{\Lambda}} \boldsymbol{e}_d \boldsymbol{e}_d^\top\right)^2\right) \boldsymbol{x}_q\right]. \qquad (30)
\end{aligned}$$

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 Eq. (30), 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 Eq. (18) and arrive at the desired result:

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

■

D.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 Fig. 6.

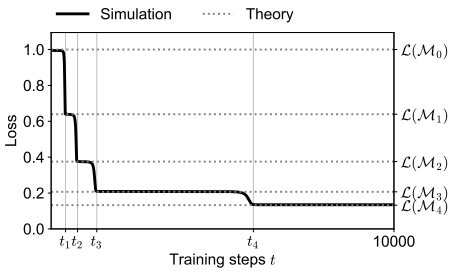


Figure 6: Illustration of $t_1, \cdots, t_D$. The loss trajectory plotted is one of the trajectories of linear attention with rank-one key and query in Fig. 1(*a*). The time $t_d\,(d = 1, \ldots, D)$ denotes the time when the loss has just undergone the d-th abrupt drop.

D.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 Eq. (24), which are

$$\tau \dot{v}_i = \boldsymbol{k}_i^\top \boldsymbol{\Lambda}^2 \boldsymbol{q}_i + O(w_{\text{init}}^5), \tag{31a}$$

$$\tau \dot{\boldsymbol{k}}_i = v_i \boldsymbol{\Lambda}^2 \boldsymbol{q}_i + O(w_{\text{init}}^5), \tag{31b}$$

$$\tau \dot{\boldsymbol{q}}_i = v_i \boldsymbol{\Lambda}^2 \boldsymbol{k}_i + O(w_{\text{init}}^5). \tag{31c}$$

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

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \boldsymbol{k}_i \\ \boldsymbol{q}_i \end{bmatrix} = v_i \begin{bmatrix} \boldsymbol{0} & \boldsymbol{\Lambda}^2 \\ \boldsymbol{\Lambda}^2 & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{k}_i \\ \boldsymbol{q}_i \end{bmatrix}. \tag{32}$$

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

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

where recall that $\lambda_d, \boldsymbol{e}_d (d = 1, \cdots, D)$ are eigenvalues and eigenvectors of $\boldsymbol{\Lambda}$. Hence, the solution to Eq. (32) takes the following form

$$\begin{aligned} \begin{bmatrix} \boldsymbol{k}_i(t) \\ \boldsymbol{q}_i(t) \end{bmatrix} = & \frac{1}{2} \sum_{d=1}^D \boldsymbol{e}_d^\top \left(\boldsymbol{k}_i(0) + \boldsymbol{q}_i(0)\right) \exp\left(\frac{\lambda_d^2}{\tau} \int_0^t v_i(t') \mathrm{d}t'\right) \begin{bmatrix} \boldsymbol{e}_d \\ \boldsymbol{e}_d \end{bmatrix} \\ & + \frac{1}{2} \sum_{d=1}^D \boldsymbol{e}_d^\top \left(\boldsymbol{k}_i(0) - \boldsymbol{q}_i(0)\right) \exp\left(-\frac{\lambda_d^2}{\tau} \int_0^t v_i(t') \mathrm{d}t'\right) \begin{bmatrix} \boldsymbol{e}_d \\ -\boldsymbol{e}_d \end{bmatrix}. \end{aligned} \tag{33}$$

If $v_i > 0$, the first summation term in Eq. (33) 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 Eq. (33) 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 $\boldsymbol{e}_d \boldsymbol{e}_d^\top$.

D.5.2. REDUCTION TO SCALAR DYNAMICS WITH AN ALIGNMENT ANSATZ.

The dominating term in Eq. (33) 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[1]:

$$\boldsymbol{k}_1 = \boldsymbol{q}_1 = v_1 \boldsymbol{e}_1, \tag{34a}$$

$$\boldsymbol{k}_i = \boldsymbol{q}_i = \boldsymbol{0}, v_i = 0, \; i = 2, \cdots, H. \tag{34b}$$

Note that Eq. (34) 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 Eq. (44). This ansatz can greatly simplify the

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

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 Eq. (24) 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 dynamics with the ansatz,

$$\boldsymbol{\Lambda}^2 - \mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right)\sum_{i=1}^{H} v_i \boldsymbol{k}_i \boldsymbol{q}_i^\top \boldsymbol{\Lambda} = \boldsymbol{\Lambda}^2 - \sum_{d=1}^{D} a_d \boldsymbol{e}_d \boldsymbol{e}_d^\top v_1^3 \boldsymbol{e}_1 \boldsymbol{e}_1^\top \boldsymbol{\Lambda} = \boldsymbol{\Lambda}^2 - \lambda_1 a_1 \boldsymbol{e}_1 \boldsymbol{e}_1^\top v_1^3 \tag{35}$$

where a_1 is the first eigenvalue of $\mathbb{E}(\hat{\boldsymbol{\Lambda}}^2)$ defined in Eq. (18). Substituting Eqs. (34) and (35) into Eq. (24), we find that the training dynamics of the first head simplify and the dynamics of the rest of the heads are zero

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

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 $v_1(t)$

$$\begin{cases}
\tau \dot{v}_1 = \lambda_1^2 v_1^2 - \lambda_1 a_1 v_1^5 \\
\tau \dot{v}_1 \boldsymbol{e}_1 = \lambda_1^2 v_1^2 \boldsymbol{e}_1 - \lambda_1 a_1 v_1^5 \boldsymbol{e}_1 \\
\tau \dot{v}_1 \boldsymbol{e}_1 = \lambda_1^2 v_1^2 \boldsymbol{e}_1 - \lambda_1 a_1 v_1^5 \boldsymbol{e}_1
\end{cases} \quad \Rightarrow \quad \tau \dot{v}_1 = \lambda_1^2 v_1^2 - \lambda_1 a_1 v_1^5 \tag{36}$$

Eq. (36) is a separable ordinary differential equation. By separating variables and integrating both sides, we can solve t in terms of v_1

$$\begin{aligned}
\frac{\lambda_1^2}{\tau} t &= \int \frac{1}{v_1^2 - \frac{a_1}{\lambda_1} v_1^2} \mathrm{d}v_1 \\
&= \frac{\sqrt[3]{\frac{a_1}{\lambda_1}}}{6} \left[\ln \left(\frac{\sqrt[3]{\frac{a_1^2}{\lambda_1^2}} v_1^2 + \sqrt[3]{\frac{a_1}{\lambda_1}} v_1 + 1}{\sqrt[3]{\frac{a_1^2}{\lambda_1^2}} v_1^2 - 2\sqrt[3]{\frac{a_1}{\lambda_1}} v_1 + 1} \right) - 2\sqrt{3} \tan^{-1} \left(\frac{2\sqrt[3]{\frac{a_1}{\lambda_1}} v_1 + 1}{\sqrt{3}} \right) \right] - \frac{1}{v_1}.
\end{aligned} \tag{37}$$

Since Eq. (37) 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 Eq. (36) is $\lambda_1^2 v_1^2$ and thus the dynamics can be approximated by

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

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

$$t_1 \approx \frac{\tau}{\lambda_1^2 v_1(0)}. \tag{38}$$

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

In Appendix D.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 $\mathcal{M}_{m+1}$.

D.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 Eq. (6). 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 Eq. (27), the heads that are near small initialization have the following training dynamics

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

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

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \boldsymbol{k}_i \\ \boldsymbol{q}_i \end{bmatrix} = v_i \begin{bmatrix} \boldsymbol{0} & \boldsymbol{\Omega} \\ \boldsymbol{\Omega} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{k}_i \\ \boldsymbol{q}_i \end{bmatrix}, \quad \text{where } \boldsymbol{\Omega} = \sum_{d=m+1}^{D} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top. \tag{39}$$

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 $\{\lambda_d^2, -\lambda_d^2\}_{d=m+1}^{D}$. 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 Eq. (39) takes the following the form

$$\begin{aligned}\begin{bmatrix}\boldsymbol{k}_i(t)\\ \boldsymbol{q}_i(t)\end{bmatrix} &= \frac{1}{2}\sum_{d=m+1}^{D}\boldsymbol{e}_d^\top\left(\boldsymbol{k}_i(t_m)+\boldsymbol{q}_i(t_m)\right)\exp\left(\frac{\lambda_d^2}{\tau}\int_{t_m}^{t}v_i(t')\mathrm{d}t'\right)\begin{bmatrix}\boldsymbol{e}_d\\ \boldsymbol{e}_d\end{bmatrix}\\ &+\frac{1}{2}\sum_{d=m+1}^{D}\boldsymbol{e}_d^\top\left(\boldsymbol{k}_i(t_m)-\boldsymbol{q}_i(t_m)\right)\exp\left(-\frac{\lambda_d^2}{\tau}\int_{t_m}^{t}v_i(t')\mathrm{d}t'\right)\begin{bmatrix}\boldsymbol{e}_d\\ -\boldsymbol{e}_d\end{bmatrix}\\ &+\sum_{d=1}^{m}\boldsymbol{e}_d^\top\left(\boldsymbol{k}_i(t_m)+\boldsymbol{q}_i(t_m)\right)\begin{bmatrix}\boldsymbol{e}_d\\ \boldsymbol{e}_d\end{bmatrix}.\end{aligned}\tag{40}$$

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.

D.6.2. REDUCTION TO SCALAR DYNAMICS WITH AN ALIGNMENT ANSATZ.

The dominating term in Eq. (40) 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 Eq. (6). 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 Eq. (24) with the ansatz, we first calculate a common term with the ansatz

$$\begin{aligned}\boldsymbol{\Lambda}^2-\mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right)\sum_{i=1}^{H}v_i\boldsymbol{k}_i\boldsymbol{q}_i^\top\boldsymbol{\Lambda} &= \boldsymbol{\Lambda}^2-\sum_{d=1}^{D}a_d\boldsymbol{e}_d\boldsymbol{e}_d^\top\left(\sum_{i=1}^{m}\frac{\lambda_d}{a_d}\boldsymbol{e}_i\boldsymbol{e}_i^\top+v_{m+1}^3\boldsymbol{e}_{m+1}\boldsymbol{e}_{m+1}^\top\right)\boldsymbol{\Lambda}\\ &= \boldsymbol{\Lambda}^2-\sum_{d=1}^{m}\lambda_d^2\boldsymbol{e}_d\boldsymbol{e}_d^\top-\lambda_{m+1}a_{m+1}\boldsymbol{e}_{m+1}\boldsymbol{e}_{m+1}^\top v_{m+1}^3\end{aligned}\tag{41}$$

By substituting Eqs. (6) and (41) into Eq. (24), 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 $v_i(t)$

$$\begin{aligned}
\tau \dot{v}_i &= v_i^2 \boldsymbol{e}_{m+1}^\top \left(\boldsymbol{\Lambda}^2 - \sum_{d=1}^{m} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top - \lambda_{m+1} a_{m+1} \boldsymbol{e}_{m+1} \boldsymbol{e}_{m+1}^\top v_i^3 \right) \boldsymbol{e}_{m+1} \\
&= \lambda_{m+1}^2 v_i^2 - \lambda_{m+1} a_{m+1} v_i^5 \\
\tau \dot{\boldsymbol{k}}_i = \tau \dot{v}_i \boldsymbol{e}_{m+1} &= v_i^2 \left(\boldsymbol{\Lambda}^2 - \sum_{d=1}^{m} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top - \lambda_{m+1} a_{m+1} \boldsymbol{e}_{m+1} \boldsymbol{e}_{m+1}^\top v_i^3 \right) \boldsymbol{e}_{m+1} \\
&= \lambda_{m+1}^2 v_i^2 \boldsymbol{e}_{m+1} - \lambda_{m+1} a_{m+1} v_i^5 \boldsymbol{e}_{m+1} \\
\tau \dot{\boldsymbol{q}}_i = \tau \dot{v}_i \boldsymbol{e}_{m+1} &= v_i^2 \left(\boldsymbol{\Lambda}^2 - \sum_{d=1}^{m} \lambda_d^2 \boldsymbol{e}_d \boldsymbol{e}_d^\top - \lambda_{m+1} a_{m+1} \boldsymbol{e}_{m+1} \boldsymbol{e}_{m+1}^\top v_i^3 \right) \boldsymbol{e}_{m+1} \\
&= \lambda_{m+1}^2 v_i^2 \boldsymbol{e}_{m+1} - \lambda_{m+1} a_{m+1} v_i^5 \boldsymbol{e}_{m+1} \\
\Rightarrow \quad \tau \dot{v}_i &= \lambda_{m+1}^2 v_i^2 - \lambda_{m+1} a_{m+1} v_i^5
\end{aligned} \tag{42}$$

Eq. (42) is the same ordinary differential equation as Eq. (36) 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 Eq. (42) 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

$$t_{m+1} - t_m \approx \frac{\tau}{\lambda_{m+1}^2 v_{m+1}(t_m)}. \tag{43}$$

We note that the Eq. (43) 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 Fig. 1(*a*).

D.7. Conservation Law

The gradient flow dynamics of linear attention with rank-one key and query in Eq. (24) implies a conservation law. The value, key, and query weights in a head obey

$$\frac{\mathrm{d}}{\mathrm{d}t} \left(\boldsymbol{k}_i^\top \boldsymbol{k}_i - \boldsymbol{q}_i^\top \boldsymbol{q}_i \right) = \boldsymbol{0}, \quad \frac{\mathrm{d}}{\mathrm{d}t} \left(\boldsymbol{k}_i^\top \boldsymbol{k}_i - v_i^2 \right) = 0, \tag{44}$$

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 Eq. (44) holds regardless of the choice of the loss function.

Proof We can use the generic gradient flow equation, $\tau \frac{\mathrm{d}\boldsymbol{W}}{\mathrm{d}t} = -\frac{1}{2}\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}}$, 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:

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

■

Appendix E. Linear Attention with Low-Rank Key and Query

E.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 rank-R key and query is

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

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. Specifically, we have

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

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. Specifically, we have

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

■

E.2. Gradient Flow Equations

Based on the gradient flow training rule, $\tau\frac{\mathrm{d}\boldsymbol{W}}{\mathrm{d}t} = -\frac{1}{2}\frac{\partial\mathcal{L}}{\partial\boldsymbol{W}}$, the gradient flow dynamics of linear attention with rank-R key and query is

$$\tau\dot{\boldsymbol{v}}_i = \sum_{r=1}^{R}\boldsymbol{k}_{i,r}^\top\mathbb{E}\left(\boldsymbol{\beta}(y_q - \hat{y}_q)\boldsymbol{x}_q^\top\right)\boldsymbol{q}_{i,r} = \sum_{r=1}^{R}\boldsymbol{k}_{i,r}^\top\left(\boldsymbol{\Lambda}^2 - \mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right)\sum_{i=1}^{H}\sum_{r'=1}^{R}v_i\boldsymbol{k}_{i,r'}\boldsymbol{q}_{i,r'}^\top\boldsymbol{\Lambda}\right)\boldsymbol{q}_{i,r}, \tag{45a}$$

$$\tau\dot{\boldsymbol{k}}_{i,r} = v_i\mathbb{E}\left(\boldsymbol{\beta}(y_q - \hat{y}_q)\boldsymbol{x}_q^\top\right)\boldsymbol{q}_{i,r} = v_i\left(\boldsymbol{\Lambda}^2 - \mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right)\sum_{i=1}^{H}\sum_{r'=1}^{R}v_i\boldsymbol{k}_{i,r'}\boldsymbol{q}_{i,r'}^\top\boldsymbol{\Lambda}\right)\boldsymbol{q}_{i,r}, \tag{45b}$$

$$\tau\dot{\boldsymbol{q}}_{i,r} = v_i\boldsymbol{k}_{i,r}^\top\mathbb{E}\left(\boldsymbol{\beta}(y_q - \hat{y}_q)\boldsymbol{x}_q\right) = v_i\left(\boldsymbol{\Lambda}^2 - \boldsymbol{\Lambda}\sum_{i=1}^{H}\sum_{r'=1}^{R}v_i\boldsymbol{q}_{i,r'}\boldsymbol{k}_{i,r'}^\top\mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right)\right)\boldsymbol{k}_{i,r}. \tag{45c}$$

where $i = 1, \cdots, H$, $r = 1, \cdots, R$, and the data statistics $\mathbb{E}\left(\hat{\boldsymbol{\Lambda}}^2\right)$ is calculated in Eq. (17).

E.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,

$$\mathcal{M}(\mathcal{S}_m) = \left\{ v_{1:H}, \boldsymbol{W}_{1:H}^K, \boldsymbol{W}_{1:H}^Q \middle| \text{conditions (C1)-(C3) are met} \right\}, \tag{46}$$

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

$$\mathcal{S}_m \subseteq \{1, 2, \cdots, D\},\ |\mathcal{S}_m| = m. \tag{47}$$

The three conditions for Eq. (46) are:

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

$$\sum_{i=1}^{H} \sum_{r=1}^{R} v_i \boldsymbol{k}_{i,r} \boldsymbol{q}_{i,r}^\top = \sum_{d \in \mathcal{S}_m} \lambda_d^{-1} \left(1 + \frac{1 + \mathtt{tr}(\boldsymbol{\Lambda})/\lambda_d}{N} \right)^{-1} \boldsymbol{e}_d \boldsymbol{e}_d^\top. \tag{48}$$

(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, $v_i = 0$,

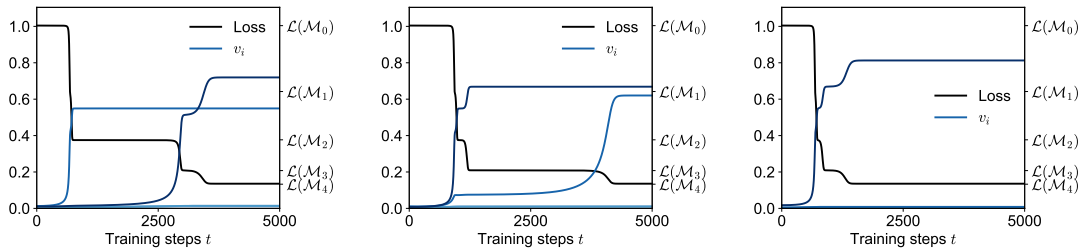
$$\sum_{r=1}^{R} \sum_{d \notin \mathcal{S}_m} \lambda_d^2 \boldsymbol{k}_{i,r}^\top \boldsymbol{e}_d \boldsymbol{e}_d^\top \boldsymbol{q}_{i,r} = 0. \tag{49}$$

With the same reasoning as Appendix D.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 Eq. (46) are valid and feasible. Linear attention with rank-R key and query has the same 2^D fixed points in the function space as its rank-one counterpart.

E.4. Saddle-to-Saddle Dynamics

For linear attention with rank-R key and query, the gradient updates of the key and query weights in Eq. (45), $\dot{\boldsymbol{k}}_{i,r}, \dot{\boldsymbol{q}}_{i,r}$, 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



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

Figure 7: Loss and value weights trajectories. The setting is the same as Fig. 1(*b*) except different ranks $R = 2, 3, 4$. In the rank-one case in Fig. 1(*b*), 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 $D = 4, N = 32, H = 5$, and $\mathbf{\Lambda}$ has eigenvalues $0.4, 0.3, 0.2, 0.1$.

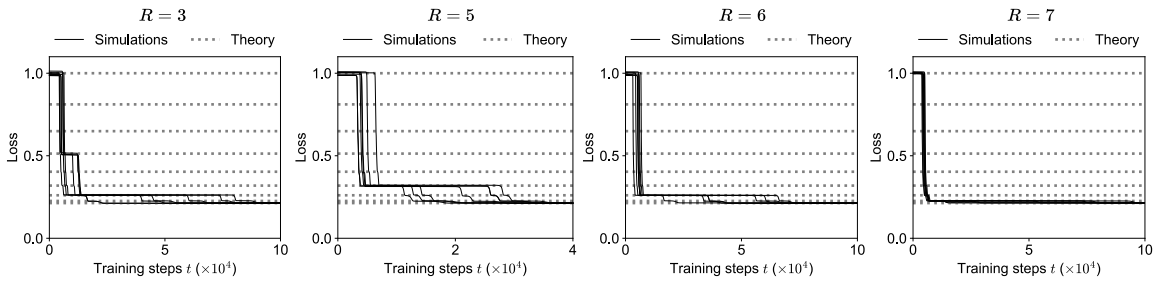


Figure 8: Same as Fig. 2 but with ranks $R = 3, 5, 6, 7$. Here $D = 8, N = 31, H = 9$, $\mathbf{\Lambda}$ has trace 1 and eigenvalues $\lambda_d \propto d^{-1}$.

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 Figs. 1(*b*) and 7 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 Fig. 8 to complement Fig. 2 in the main text.

E.5. Dynamics with Repeated Eigenvalues

We have demonstrated that linear attention 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 can also exhibit loss plateaus due to the different random initial weights in each head. In the case with distinct eigenvalues (Fig. 1(*a*)), the plateau duration is determined by both the size of

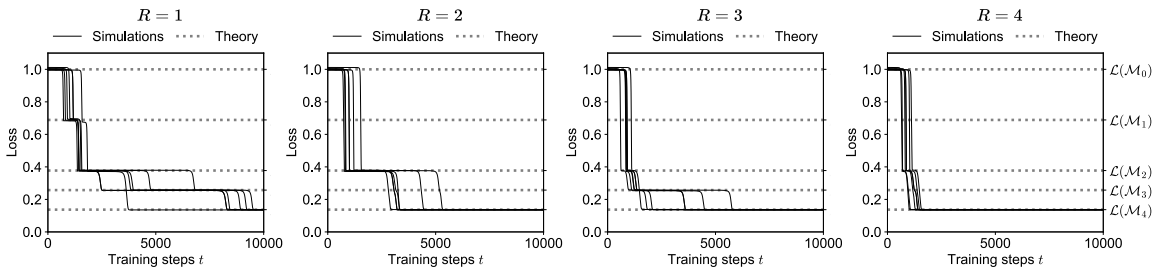


Figure 9: Loss trajectories of multi-head linear attention when the input token covariance has repeated eigenvalues. The setup is the same as in Fig. 1(*a*) except that $\mathbf{\Lambda}$ has eigenvalues $0.35, 0.35, 0.15, 0.15$. The four panels differ only in the rank of the key and query weights. Although some eigenvalues are equal, the loss trajectory of linear attention 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.

the eigenvalues and the random initialization. In the case with repeated eigenvalues (Fig. 9, leftmost panel), the plateau duration is determined solely by the random initialization.

E.6. Conservation Law

The gradient flow dynamics of linear attention with rank-R key and query in Eq. (45) implies a conservation law. The value, key, and query weights in a head obey

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{k}_{i,r}^\top \boldsymbol{k}_{i,r} - \boldsymbol{q}_{i,r}^\top \boldsymbol{q}_{i,r}\right) = \boldsymbol{0}, \quad \frac{\mathrm{d}}{\mathrm{d}t}\left(\sum_{r=1}^{R} \boldsymbol{k}_{i,r}^\top \boldsymbol{k}_{i,r} - v_i^2\right) = 0. \tag{50}$$

We here prove that Eq. (50) holds regardless of the choice of the loss function.

Proof We can use the generic gradient flow equation, $\tau \frac{\mathrm{d}\boldsymbol{W}}{\mathrm{d}t} = -\frac{1}{2}\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}}$, to calculate the gradients

$$\frac{\mathrm{d}\boldsymbol{k}_{i,r}^\top \boldsymbol{k}_{i,r}}{\mathrm{d}t} = 2\boldsymbol{k}_{i,r}^\top \frac{\mathrm{d}\boldsymbol{k}_{i,r}}{\mathrm{d}t} = 2\mathbb{E}\left(-\boldsymbol{k}_i^\top \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\hat{y}_q}\frac{\mathrm{d}\hat{y}_q}{\mathrm{d}\boldsymbol{k}_{i,r}}\right) = 2\mathbb{E}\left(-\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\hat{y}_q} v_i \boldsymbol{k}_{i,r}^\top \boldsymbol{\beta} \boldsymbol{q}_{i,r}^\top \boldsymbol{x}_q\right) \tag{51a}$$

$$\frac{\mathrm{d}\boldsymbol{q}_{i,r}^\top \boldsymbol{q}_{i,r}}{\mathrm{d}t} = 2\boldsymbol{q}_{i,r}^\top \frac{\mathrm{d}\boldsymbol{q}_{i,r}}{\mathrm{d}t} = 2\mathbb{E}\left(-\boldsymbol{q}_i^\top \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\hat{y}_q}\frac{\mathrm{d}\hat{y}_q}{\mathrm{d}\boldsymbol{q}_{i,r}}\right) = 2\mathbb{E}\left(-\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\hat{y}_q} v_i \boldsymbol{q}_{i,r}^\top \boldsymbol{x}_q \boldsymbol{k}_{i,r}^\top \boldsymbol{\beta}\right) \tag{51b}$$

$$\frac{\mathrm{d}v_i^2}{\mathrm{d}t} = 2v_i \frac{\mathrm{d}v_i}{\mathrm{d}t} = 2\mathbb{E}\left(-v_i \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\hat{y}_q}\frac{\mathrm{d}\hat{y}_q}{\mathrm{d}v_i}\right) = 2\sum_{r=1}^{R}\mathbb{E}\left(-\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\hat{y}_q} v_i \boldsymbol{\beta}^\top \boldsymbol{k}_{i,r} \boldsymbol{q}_{i,r}^\top \boldsymbol{x}_q\right) \tag{51c}$$

Comparing Eqs. (51a) and (51b), 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 Eqs. (51a) and (51b) with Eq. (51c), we obtain

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

■