Theory on Attention Dynamics for Out-of-Distribution In-Context Learning

Junze Deng
Ohio State University
Ohio, USA
deng.942@osu.edu

Daouda Sow
Ohio State University
Ohio, USA
sow.53@osu.edu

Sen Lin
University of Houston
Houston, USA
slin50@central.uh.edu

Yingbin Liang
Ohio State University
Ohio, USA
liang.889@osu.edu

Abstract

Transformers have demonstrated remarkable in-context learning (ICL) capabilities, enabling them to perform new tasks without additional fine-tuning. However, their performance often deteriorates when encountering out-of-distribution (OOD) inputs that deviate from the training distribution, and the underlying theory remains poorly understood. To fill this gap, we characterize the OOD error under the input distribution shift through the interplay between the dynamics of the so-called α-type and β-type attention weights, which represent the transformer's confidence in identifying the correct and incorrect features, respectively. Our results indicate that the OOD error for each feature depends on all pairwise interactions between the training features and OOD features, and under certain cases the transformer performs no better than random guessing. To improve the OOD generalization performance, we next investigate the impact of model fine-tuning with the OOD data, and particularly, characterize the model forgetting performance on the source domain. Interestingly, the performance on the source domain may not always degrade after fine-tuning, which highly depends on the nature of the feature shift: fine-tuning on OOD domain keeps enhancing the confidence of identifying correct features from the original distribution, while the interference from other incorrect features may either increase or decrease. Extensive experiments on both synthetic and real data are conducted to corroborate the theoretical insights.

1 Introduction

Large language models (LLMs) built on transformers have demonstrated a remarkable capability known as *in-context learning* (ICL). In this paradigm, a transformer is pre-trained on a wide range of tasks and can solve a new task without any fine-tuning, relying solely on inference from task-specific prompts [9]. This remarkable success has spurred extensive efforts to empirically investigate the underlying mechanisms of ICL from various perspectives [20, 43, 59, 53, 62, 23]. In particular, [20] introduced a useful framework to study ICL through both simple linear regression and more complex nonlinear tasks. A notable result from their work shows that models pretrained on linear regression tasks can, without any fine-tuning, solve new linear regression problems with performance matching that of conventional least-squares regression trained specifically for the task.

This framework of ICL has further inspired a growing body of theoretical research on characterizing the training dynamics of transformers, i.e., understanding how transformers acquire ICL capabilities

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

through training [68, 11, 35, 44] (see Section 2 for additional references). These studies aim to explain how training transformers with gradient descent can lead to emergent ICL behavior. A common approach is to analyze the structure of attention mechanisms at or near convergence, showing how such structures encode learning algorithms capable of solving new tasks from context alone.

Though transformers have demonstrated remarkable ICL capabilities across a wide range of tasks, their performance can drop dramatically in OOD scenarios [67, 64, 26]. Existing theoretical studies are predominantly limited to linear self-attention [66, 34, 5] or softmax self-attention in some special problems [58, 38]. Hence, a general theoretical understanding of ICL under distribution shift remains elusive. In this work, we aim to fill this gap by studying the OOD generalization of softmax attention within a linear regression framework, following recent theoretical analyses [20, 30]. Also, studies in domain adaptation [63, 50] have shown that fine-tuning the model on OOD data can improve the OOD generalization performance. Thus inspired, we further analyze the dynamics of fine-tuning and theoretically characterize the fine-tuned model's performance on the source domain, with particular attention to the emergence of catastrophic forgetting.

Our main contributions can be summarized as follows:

1) We provide an explicit expression of the OOD generalization error for ICL with a single-layer *softmax* transformer model, revealing its dependence on the geometric separation between the training and OOD subspaces and on the embedding locations of tokens within these subspaces. A key insight here is that the performance is governed by the interplay between two types of bilinear attention weights: (i) the α-type weights, which reflect the model's confidence in identifying the correct feature, and (ii) the β-type weights, which quantify the interference from other incorrect features.

2) To improve performance on the OOD domain, we fine-tune the trained transformer on the shifted features. By characterizing the training dynamics, we establish explicit upper and lower bounds on the model performance drop on the original training domain (i.e., forgetting[54, 16]). The changing rates of α and β type attention, which depend on the nature of the distribution shift, determine whether the model exhibits positive or negative forgetting. To our best knowledge, this provides the first theoretical performance characterization of transformer trained under nonstationary data.

3) We conduct comprehensive experiments to support our theoretical findings. Simulations on synthetic data that closely adhere to the theoretical assumptions validate the correctness of our analysis. We further extend our evaluation to more realistic settings, including *TinyTransformer* model with varying depths, BERT model on real-world datasets and natural distribution shift on real-world data. These demonstrate that the theoretical insights persist beyond idealized conditions.

2 Related Work

In-context learning: *ICL* has been identified as a superior capability in LLMs [9], which has inspired extensive research to *empirically* understand its underlying mechanisms from various aspects [20, 43, 59, 53, 62, 23, 60, 7, 45, 10, 72]. Recent studies have also started to explore the theoretical properties of ICL from multiple perspectives, including Bayesian learning [62, 2, 25, 32, 57, 60, 71, 31, 24, 41], expressive power [4, 21, 6, 29], generalization and stability [37, 33, 39, 13, 48], and internal mechanisms [15, 53, 6]. In particular, an important line of work that bridges theory and practice has focused on characterizing the training dynamics of transformers [20, 68, 42, 1, 30, 65, 35, 44, 58, 55, 47, 12, 19, 46], aiming to explain how transformers trained via gradient descent acquire ICL capabilities. However, the above studies on training dynamics have mainly focused on static settings. In stark contrast, we advance the theoretical analysis to a dynamic setting in which input features undergo opposing rotations, and investigate the forgetting that may arise in such scenarios.

OOD generalization of transformer: In recent years, a bunch of empirical works study the OOD generalization of transformer-based models[49, 70, 69, 40, 28, 27, 22, 3]. A growing body of theoretical work seeks to understand OOD in-context learning in *linear* transformers from multiple perspectives, including low-dimensional subspace perspective [34], different types of distribution shift [68], local adaptation dynamics [56], and sensitivity to adversarial perturbations [5]. However, the understanding of *softmax* self-attention remains underexplored. [38] demonstrated that transformers can behave like 1-nearest-neighbor algorithms and discussed their OOD behavior. [58] investigated the length generalization on sparse token selection problem. [14] shows that transformers generalize across Lipschitz functions only when their Lipschitz constants are similar.

In this paper, we analyze a single-layer *softmax* attention model under a covariate shift setting, in which the OOD features deviate from those in the training feature set. More importantly, we characterize the behavior of the model during fine-tuning on the OOD domain and adopt the concept of forgetting to evaluate whether fine-tuning the transformer improves or degrades its performance on the original training domain.

3 Preliminaries

In-Context Learning: Consider the following standard framework of ICL introduced in [20], which has been adopted widely in the recent line of theoretical studies [30, 1, 44, 68]. More specifically, consider a task distribution over linear regression tasks, denoted by $\mathcal{F} = \{f : \mathcal{X} \to \mathbb{R} \mid f(x) = \langle w, x \rangle\}$, where the weight vector w is drawn from a standard Gaussian distribution. Here x is the input uniformly sampled from a feature set $V = \{v_k \in \mathbb{R}^d, k = 1, 2, .., K\}$ with K orthonormal feature vectors. During ICL, each training prompt P is generated based on the following procedure: 1) randomly sampling a function f from $\mathcal{F}$, 2) uniformly sampling a set of N inputs $\{x_1, x_2, ..., x_N\}$ and a query x_{query} from the feature set V, 3) generating the model response y_i by computing the value of f given any input x_i for $i \in [1, N]$, i.e., $y_i = f(x_i)$ [1], and 4) constructing the prompt $P = (x_1, y_1, x_2, y_2, ..., x_N, y_N, x_{\text{query}})$. Based on this prompt, a well-trained model should be able to generate a prediction $\hat{y}_{\text{query}} \approx f(x_{\text{query}})$ for the query x_{query}.

Self-Attention Layer with Reparameterization: We consider a one-layer transformer as an in-context learner, with the standard single-head self-attention layer [52] parameterized by a key matrix $W_K \in \mathbb{R}^{(d+1)\times(d+1)}$, a query matrix $W_Q \in \mathbb{R}^{(d+1)\times(d+1)}$, and a value matrix $W_V \in \mathbb{R}^{(d+1)\times(d+1)}$. Given a prompt $P = (x_1, y_1, x_2, y_2, ..., x_N, y_N, x_{\text{query}})$, we adopt the following natural embedding, which arranges the corresponding input and label as columns:

$$E = \begin{pmatrix} x_1 & x_2 & ... & x_N & x_{\text{query}} \\ y_1 & y_2 & ... & y_N & 0 \end{pmatrix} = \begin{pmatrix} E_x \\ E_y \end{pmatrix}, \tag{1}$$

where $E \in \mathbb{R}^{(d+1)\times(N+1)}$. Here we denote the first d rows and the last row of E as E_x and E_y, respectively. Then the output of the self-attention layer can be characterized by

$$F_{\text{SA}}(W_K, W_Q, W_V; E) = W_V E \cdot \text{softmax}\left((W_K E)^\top W_Q E\right)$$

with a column-wise softmax$(\cdot)$ function, where the i^{th} entry of softmax(z) is $\exp(z_i)/\sum_s \exp(z_s)$.

We further modify Equation (1) based on two widely used techniques in previous theoretical studies: 1) We remove the last column of the embedding matrix E when it is multiplied to the key matrix W_K and the value matrix W_V, in order to prevent the query token from attending to itself [30, 44]. 2) We consolidate the query and key matrices into one matrix W_{KQ}, and further consider the following forms of W_V and W_{KQ}, which are commonly used in recent theoretical studies and have been shown to achieve the optimum or nearly optimum loss value [68, 1, 30]:

$$W_V = \begin{pmatrix} 0_{d\times d} & 0_{d\times 1} \\ 0_{1\times d} & 1 \end{pmatrix}, \quad W_{KQ} = \begin{pmatrix} Q_{d\times d} & 0_{d\times 1} \\ 0_{1\times d} & 0 \end{pmatrix}.$$

The above reparameterization leads to a simplified attention layer with the output given by

$$F_{\text{SA}}(Q; E) = (E_y)_{1:N} \cdot \text{softmax}\left((E_x)_{1:N}^\top Q E_x\right).$$

The model prediction for the query x_{query} is given by the last entry of F_{SA}, i.e.,

$$\hat{y}_{\text{query}}(Q; E) = [F_{\text{SA}}(Q; E)]_{N+1}. \tag{2}$$

4 OOD Generalization and Fine-tuning

[1] We focus on the noiseless setting in the main text. In Appendix K, we extend the analysis to noisy observations of the form $y_i = f(x_i) + \epsilon_i$.

We denote the training and OOD feature sets as $V = \{v_k\}_{k=1}^K$ and $U = \{u_k\}_{k=1}^K$ with orthonormal features, respectively. In the training phase, the inputs are sampled from the feature set V. Once the transformer is trained, it will be tested on the OOD data, where the inputs are sampled from the feature set U. We assume the task distribution is same for training and OOD evaluation. For simplicity, we further assume that the training prompts consist of the same number of features from the corresponding feature sets.[2]

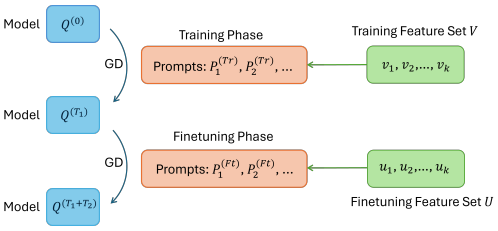


Figure 1: Illustration: Training and Fine-tuning

Training Loss and Algorithm: We define the training loss with respect to (w.r.t.) any query feature v as the expected value of the squared prediction error as follows:

$$L_v^{(\mathrm{tr})}(Q) = \frac{1}{2}\mathbb{E}\Big[\mathbf{1}_{\{x_{\mathrm{query}}=v\}}\Big(\widehat{y}_{\mathrm{query}}\,(Q;E) - w^\top x_{\mathrm{query}}\Big)^2\Big],$$

where the expectation is taken over the task distribution of w, the distribution of training prompt E and the distribution of query. The overall training loss can be given by

$$L^{(\mathrm{tr})}(Q) = \textstyle\sum_{v\in V} L_v^{(\mathrm{tr})}(Q). \tag{3}$$

The gradient descent (GD) is then applied to minimize the above overall training loss with the learning rate η_t by the following iterative updates:

$$Q^{(t+1)} = Q^{(t)} - \eta_t \nabla_Q L^{(\mathrm{tr})}(Q^{(t)}). \tag{4}$$

We note that the gradient $\nabla_Q L$ is calculated entry-wisely. To avoid situations where only a subset of features is well learned, we apply the condition $L_k(Q) \le \frac{\epsilon}{K}$ for all $k \in [K]$ as the stopping criterion for training. This is stronger than requiring the total loss to be below ϵ, and is introduced solely to ensure a uniform level of feature learning in our analysis.

OOD Testing Data: Recall that the testing feature set $U = \{u_k\}_{k=1}^K$ are a set of orthonormal vectors. Once the transformer is trained on the feature set V, we will evaluate the model performance on OOD data using

$$L^{(\mathrm{ft})}(Q) = \sum_{u\in U} L_u^{(\mathrm{ft})}(Q). \tag{5}$$

Here $L_u^{(\mathrm{ft})}(Q)$ is the OOD error w.r.t. the query feature u:

$$L_u^{(\mathrm{ft})}(Q) = \frac{1}{2}\mathbb{E}\Big[\mathbf{1}_{\{x_{\mathrm{query}}=u\}}\Big(\widehat{y}_{\mathrm{query}}\,(Q;E) - w^\top x_{\mathrm{query}}\Big)^2\Big],$$

where the expectation is taken over the task distribution of w, the distribution of testing prompt E and the query.

Fine-tune and Forgetting: To enhance the model performance on OOD data, we fine-tune the attention model on the OOD data to minimize the error $L^{(\mathrm{ft})}$ by using GD:

$$Q^{(t+1)} = Q^{(t)} - \eta_t \nabla_Q L^{(\mathrm{ft})}(Q^{(t)}). \tag{6}$$

The fine-tuning process shares the same stopping criteria for the model training in Equation (4), which implies that the model can achieve good performance on each OOD feature after fine-tuning. Suppose the training process converges at T_1 and the fine-tune process stops after another period of time T_2, as illustrated in fig. 1. Inspired by the studies in continual learning, an interesting question to explore here is whether the model will forget the knowledge of the original training domain after fine-tuning on OOD data. To answer this question, we define the following measure to evaluate the forgetting w.r.t. the training feature v

$$F_v = L_v^{(\mathrm{tr})}\left(Q^{(T_1+T_2)}\right) - L_v^{(\mathrm{tr})}\left(Q^{(T_1)}\right)$$

Note that the training loss converges simultaneously for all features due to the symmetry property of training features in the training phase; specifically, $L_v^{(\mathrm{tr})}\big(Q^{(T_1)}\big) = \frac{\epsilon}{K}, \forall\, v \in V$. A positive forgetting [36] indicates a performance drop after fine-tuning, whereas a negative forgetting[51] corresponds to positive backward transfer (BWT), where fine-tuning improves performance on the retained capability. Moreover, we define the overall forgetting as the sum of all the individual forgetting: $F = \sum_{v\in V} F_v$.

[2]This assumption characterizes the asymptotic behavior of the prompts when the inputs are uniformly sampled and the prompt length N is large, as discussed in Appendix I.

5 Main Results

In this section, we present our main theoretical results. Before that, we first introduce several important notations that serve as the foundations of our theoretical analysis.

Attention scores. For a given prompt, the attention score for the i^{th} input x_i under model parameter Q is given by

$$\textbf{attn}_i(Q; E) = \frac{\exp({x_i}^\top Q x_{\text{query}})}{\sum_n \exp({x_n}^\top Q x_{\text{query}})}.$$

Then, the attention score of feature v is computed by aggregating the attention scores of tokens that are v:

$$\textbf{Attn}_v(Q; E) = \textstyle\sum_{i:x_i=v} \textbf{attn}_i(Q; E). \tag{7}$$

The prediction in eq. (2) can be rewritten as

$$\hat{y}_{\text{query}} = \sum\nolimits_{v\in W} \textbf{Attn}_v(Q; E)\langle v, w\rangle, \tag{8}$$

where $W = V$ during the training phase and $W = U$ during the fine-tuning phase. Since $\sum_{v\in W} \textbf{Attn}_v(Q; E) = 1$, if $x_{\text{query}} = v$, the attention weight $\textbf{Attn}_v(Q; E)$ reflects the probability that the model identifies the correct feature.

Two types of attention weights. We further define two types of attention weights, i.e., α-type weights $\alpha_v(Q)$ and β-type weights $\beta_{v',v}(Q)$, as follows:

$$\alpha_v(Q) = v^\top Q v, \;\; \beta_{v',v}(Q) = {v'}^\top Q v, \quad v \neq v'. \tag{9}$$

The α-type weight captures the projection of the attention matrix onto the key feature that is the same as the query feature, while the β-type weight captures the projection onto the key feature that is different from the query feature. During the training phase, we observe that if $x_{\text{query}} = v$, eq. (7) can be rewritten as

$$\textbf{Attn}_v(Q; E) = \frac{\exp(\alpha_v(Q))}{\exp(\alpha_v(Q)) + \sum\limits_{v'\in V, v'\neq v} \exp(\beta_{v',v}(Q))}. \tag{10}$$

Clearly, the attention score for the query feature v depends on both the α-type weights, reflecting the model's confidence to identify the correct feature, and the β-type weights, characterizing the interference from incorrect features. The attention score during the fine-tuning phase can be captured similarly. As we will show later, the evolution of these two types of weights under feature shifts plays a key role in determining the model's performance. To simplify notation, we denote $\alpha_v^{(t)} = \alpha_v(Q^{(t)})$ and $\beta_{v,v'}^{(t)} = \beta_{v,v'}(Q^{(t)})$. In what follows, we will present our results on OOD generalization, and then characterize the training dynamics during fine-tuning, followed by the forgetting analysis.

5.1 OOD generalization

Prior work on ICL [30] has shown that, after a training duration of $T_1 = \text{poly}(\frac{1}{\epsilon}, K, \frac{1}{\eta})$, GD will yield a model with prediction error $L_v^{(\text{tr})}(Q^{(T_1)}) \leq \frac{\epsilon}{K}$ for all $v \in V$. Therefore, we evaluate the obtained transformer at time T_1 on the OOD features U. We use boldface letters to denote matrices formed by concatenating the training (or OOD) features: $\mathbf{V} = [v_1, v_2, \ldots, v_K]$ and $\mathbf{U} = [u_1, u_2, \ldots, u_K]$. Then, we can have the following theorem.

Theorem 5.1. *Suppose the training phase ends at time T_1. The attention scores associated with OOD features are:*

$$\alpha_{u_k}^{(T_1)} = \tfrac{K\alpha_{K,\epsilon}^*}{K-1}\left([\mathbf{M}^\top\mathbf{M}]_{k,k} - \langle u_k, \bar{v}\rangle^2\right),\; \beta_{u_{k'},u_k}^{(T_1)} = \tfrac{K\alpha_{K,\epsilon}^*}{K-1}\left([\mathbf{M}^\top\mathbf{M}]_{k',k} - \langle u_k, \bar{v}\rangle\langle u_{k'}, \bar{v}\rangle\right),$$

where $\alpha_{K,\epsilon}^* = \left(1 - \frac{1}{K}\right)\log\left(\frac{K-1}{\sqrt{\epsilon}}\left(\sqrt{\frac{K}{2(K-1)}} - \sqrt{\epsilon}\right)\right)$, $\mathbf{M} = \mathbf{V}^\top\mathbf{U}$, $[\cdot]_{k',k}$ denotes the element at $(k')^{th}$ row and k^{th} column, and $\bar{v} = \frac{1}{\sqrt{K}}\sum_{i=1}^K v_i$. Then, the OOD error can be characterized by:

$$L_{u_k}^{(ft)}(Q^{(T_1)}) = \frac{(\sum\limits_{n\neq k}\exp(\beta_{u_n,u_k}^{(T_1)}))^2 + \sum\limits_{n\neq k}\exp(\beta_{u_n,u_k}^{(T_1)})^2}{2K(\exp(\alpha_{u_k}^{(T_1)}) + \sum\limits_{n\neq k}\exp(\beta_{u_n,u_k}^{(T_1)}))^2}. \tag{11}$$

We can have the following remarks about this theorem.

(1) Element-wise, the (i, j)-element of $\mathbf{M}^\top\mathbf{M}$ is $u_i^\top\mathbf{V}\mathbf{V}^\top u_j$, which captures the inner product between the components of u_i and u_j projected onto span(V). Geometrically, the matrix $\mathbf{M}^\top\mathbf{M}$ characterizes the alignment between the training subspace span(V) and the OOD subspace span(U): its eigenvalues are the squared cosines of the principal angles between these two subspaces. When span(V) = span(U), the principal angles are 0 and we have $\mathbf{M}^\top\mathbf{M} = I_K$. Since $u_k^\top\mathbf{V}\mathbf{V}^\top u_k \leq 1$, the diagonal elements of $\mathbf{M}^\top\mathbf{M}$ attain their maximal values, indicating large α-type attention weights. Meanwhile, the off-diagonal elements are zero, implying low β-type attention. Together, these lead to a small OOD error. When span$(V) \perp$ span(U), the principal angles are $\frac{\pi}{2}$ and we have $\mathbf{M}^\top\mathbf{M} = \mathbf{0}_K$. In this case, the training subspace and the OOD subspace are extremely distinct, making the OOD generalization error as poor as a random guessing, as presented in Corollary 5.2.

(2) The OOD generalization is not determined solely by the distance between the training subspace and the OOD subspace; it also depends on the embedding positions of tokens within these subspaces. When the query is u_k, the α-type attention weight decreases as u_k becomes closer to $\bar{v}$, indicating that the model's confidence in identifying the correct feature decreases when the test feature u_k resembles the average representation of the training features $\bar{v}$. The β-type attention weight depends on the inner products $\langle u_k, \bar{v}\rangle\langle u_{k'}, \bar{v}\rangle$. When the directions of projections from the query feature u_k and another feature u_n onto the training average representation are inconsistent, i.e., $\langle u_k, \bar{v}\rangle\langle u_{k'}, \bar{v}\rangle < 0$, the interference from this incorrect feature u_n becomes significant, leading to a larger OOD error. Note that even if the training space and the OOD space coincide, i.e., span(V) = span(U), there still exists an OOD scenario in which the model performs no better than random guessing when the query token is a certain OOD feature.

(3) Theorem 5.1 indicates that the OOD error for each feature depends on all pairwise interactions between the training features and the OOD features. Based on this, we further discuss the OOD error under different distribution shift levels as follows. Suppose span(U) = span(V) and $x_{\text{query}} = u_k$. When the OOD features are slightly shifted from the training features, i.e., $\langle u_n, \bar{v}\rangle = \frac{1}{\sqrt{K}} + \delta_n$ with $\delta_n \ll \frac{1}{\sqrt{K}}$, the OOD error is tolerated by $L_{u_k}^{(\text{ft})}(Q^{(T_1)}) \leq \mathcal{O}(\epsilon)$. When the shift gets larger, e.g., $\delta_k \geq \frac{2}{\sqrt{K}}$ and $\delta_n \ll \frac{1}{\sqrt{K}}$ for $n \neq k$, we have $L_{u_k}^{(\text{ft})}(Q^{(T_1)}) \geq \Omega(\epsilon)$. We further observe that the trained transformer performs no better than random guessing in two distinct regimes: (1) an in-space distribution shift, where $\text{span}(V) = \text{span}(U)$, and (2) a severe out-space distribution shift, where $\text{span}(V) \perp \text{span}(U)$. This observation is formalized in the following corollary.

Corollary 5.2. *The following two statements hold. (1) Suppose span(V) = span(U). If the query token is u_k and $u_k = \bar{v}$, we have $\boldsymbol{Attn}_u = \frac{1}{K}, \forall\, u \in U$. (2) Suppose span$(V) \perp$ span(U). For any query token, we have $\boldsymbol{Attn}_u = \frac{1}{K}, \forall\, u \in U$.*

The first statement of this corollary implies that if the OOD subspace coincides with the training subspace and the query token equals the average representation of the training features, the trained transformer assigns uniform attention across all features, effectively behaving like random guessing. The second statement is even stronger: when the OOD subspace is orthogonal to the training subspace, the transformer behaves like random guessing for **any** query from the OOD feature set. This corollary highlights that trained transformers can perform poorly on OOD data.

5.2 Fine-tuning Analysis

To build a comprehensive understanding of model fine-tuning on OOD data by following from eq. (6), we first have the following lemma to capture the relationship between two types of attention weights.

Lemma 5.3. *Let $\Delta_{k,\alpha}^{(t)}$ and $\Lambda_{k,n,\beta}^{(t)}$ denote the updates of α-type and β-type attention weights during the fine-tuning time t, respectively, such that $\alpha_{u_k}^{(T_1+t)} = \alpha_{u_k}^{(T_1)} + \Delta_{k,\alpha}^{(t)}$ and $\beta_{u_n,u_k}^{(T_1+t)} = \beta_{u_n,u_k}^{(T_1)} - \Lambda_{n,k,\beta}^{(t)}$ for $n \neq k$. Then, we have $\Delta_{k,\alpha}^{(t)} = \sum_{n\neq k} \Lambda_{n,k,\beta}^{(t)}$ at time t.*

This lemma implies that the update rate of the α-type attention weight equals the total update rate of the corresponding β-type attention weights. In fact, this property can be also observed from the training phase. As shown in Lemma F.2 in the appendix, we have $\Delta_{k,\alpha}^{(t)}, \Lambda_{k,n,\beta}^{(t)} > 0$, indicating that the α-type attention weights consistently increase, while the β-type attention weights steadily

decrease. This simultaneously enhances the model's ability to identify the correct features and reduces interference from incorrect features. To simplify our analysis of fine-tuning phase, we adopt the following assumption on the feature shifts. We assume the training and testing features have the following parallel relationship.

Assumption 5.4. For all $k \in [K]$, we assume $\langle v_k, u_k \rangle = \gamma_k$, where $\gamma_k \in [\gamma_{\min}, \gamma_{\max}]$. For all $n \neq k$, we assume $\langle v_k, u_n \rangle = 0$.

We note that this assumption is set to get a clean result to analyze, while our proof idea can be extended beyond this assumption, as discussed in Appendix J. Under Assumption 5.4, we establish the following theorem to characterize the training dynamics in the fine-tuning phase.

Theorem 5.5. *Suppose Assumption 5.4 holds and the stepsize is constant $\eta = \mathcal{O}(1)$. During the fine-tuning phase, there exists $t_1 \leq t_2 \leq ... \leq t_K$, such that at each time $t_m = \mathcal{O}(poly(K, \frac{1}{\epsilon}, \frac{1}{\eta}))$, we have $L^{(ft)}_{u_{k(m)}}(Q^{(T_1+t_m)}) \leq \frac{\epsilon}{K}$. Moreover, for $\Delta^{(t)}_{k,\alpha}$ and $\Lambda^{(t)}_{k,n,\beta}$ defined in Lemma 5.3, we can have:*

$$\Delta^{(t_m)}_{k(m),\alpha} = \Theta\left((1 - \gamma^2_{k(m)} - \frac{\gamma_{k(m)}\gamma_\alpha}{K}) \log \frac{K}{\sqrt{\epsilon}}\right), \ \Lambda^{(t_m)}_{n,k(m),\beta} = \Theta\left(\frac{\Delta^{(t_m)}_{k(m),\alpha}}{K} + \frac{\gamma_{k(m)}\gamma_\beta}{K} \log \frac{K}{\sqrt{\epsilon}}\right)$$

with some constant $\gamma_\alpha, \gamma_\beta \in [\gamma_{\min}, \gamma_{\max}]$.

Theorem 5.5 indicates that there exists a sequence of time t_m such that the fine-tuning loss for the feature $u_{k(m)}$ converges to $\frac{\epsilon}{K}$, so that the model performance on OOD data can be guaranteed. At each t_m, we characterize two types of attention weights associated with the feature $k(m)$ by providing upper and lower bounds on the attention weights. At time $T_2 := t_K$, all the individual fine-tuning loss converges to $\frac{\epsilon}{K}$ and the fine-tuning phase stops.

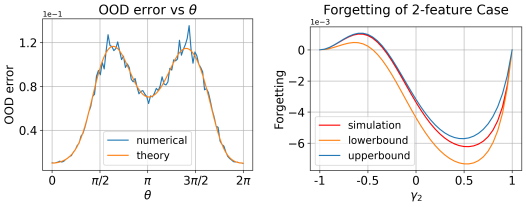
More importantly, as the model has been modified during fine-tuning, understanding whether it still preserves the knowledge of the original training domain is critical. To this end, we characterize the loss of the fine-tuned model on the training domain in the following theorem:

Theorem 5.6. *Suppose Assumption 5.4 holds and the stepsize is of constant order $\eta = \mathcal{O}(1)$. After fine-tuning, the model performance on the original feature set can be captured by*

$$L^{(tr)}_{v_k}(Q^{(T_1+T_2)}) = \frac{(\sum_{n\neq k} \exp(\beta^{(T_1+T_2)}_{v_n,v_k}))^2 + \sum_{n\neq k} \exp(\beta^{(T_1+T_2)}_{v_n,v_k})^2}{2K(\exp(\alpha^{(T_1+T_2)}_{v_k}) + \sum_{n\neq k} \exp(\beta^{(T_1+T_2)}_{v_n,v_k}))^2},$$

where $\alpha^{(T_1+T_2)}_{v_k} = \alpha^{(T_1)}_{v_k} + \gamma^2_k \Delta^{(T_2)}_{k,\alpha}$ and $\beta^{(T_1+T_2)}_{v_n,v_k} = \beta^{(T_1)}_{v_n,v_k} - \gamma_k\gamma_n\Lambda^{(T_2)}_{n,k,\beta}$, for $n \neq k$.

To explain, we first note that Theorem 5.5 characterizes the cumulative updates of the α-type and β-type attention weights with respect to fine-tuning features up to time t during the fine-tuning phase, denoted by $\Delta^{(t)}_{k,\alpha}$ and $\Lambda^{(t)}_{k,n,\beta}$, respectively. These quantities can be interpreted as the cumulative projection of the gradient updates onto the fine-tuning feature directions under gradient descent. Since the same gradient updates simultaneously affect the attention weights associated with the training features through the same update rule, the resulting changes in training-feature attention weights can be obtained by re-expressing these accumulated updates in the training feature basis. In other words, the cumulative updates with respect to training features are equivalently given by projecting $\Delta^{(t)}_{k,\alpha}$ and $\Lambda^{(t)}_{k,n,\beta}$ onto the directions of the training features.



(a) OOD evaluation (b) 2-Feature Forgetting

Figure 2: Numerical Simulation Results

One interesting observation here is that the model's performance on the training domain may either improve or deteriorate, depending on the nature of the distribution shift. By analyzing the two-feature case and the general K-feature case, we explicitly characterize the regions in which forgetting is positive or negative, as discussed in Appendix H.

6 Experimental Results

6.1 Simulations on Synthetic Data

In this section, we conduct numerical simulations on synthetic data to validate our theoretical results. Follow our theoretical setup, set $K = 4$ and $d = 8$, $\epsilon = 0.01$, and $\eta = 0.1$. We randomly generate orthonormal training features $\mathbf{V}$ and construct OOD features as $\mathbf{U} = R_{1,2}\mathbf{V}$. Here,

$$R_{i,j}(\theta) = I + (\cos\theta - 1)(e_i e_i^\top + e_j e_j^\top) + \sin\theta(e_i e_j^\top - e_j e_i^\top), \tag{12}$$

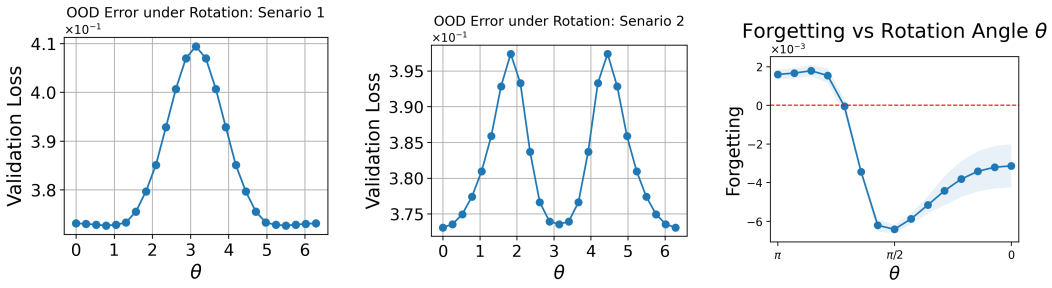
where e_i is the i^{th} orthonormal basis. As shown in fig. 2a, the simulation result is consistent with Theorem 5.1. We observe that the OOD error $L^{(\text{ft})}(Q^{(T_1)})$ is consistently higher than the training error $L^{(\text{tr})}(Q^{(T_1)})$, which implies that the model performance drops under distribution shifts. We then investigate the forgetting behavior in a special two-feature case, where $\gamma_1 = 1$ and $\gamma_2 \in [-1, 1]$. Following our theoretical setup, we fine-tune the transformer on OOD features and evaluate the model performance on original training domain. As shown in fig. 2b, the performance on the training domain degrades after fine-tuning when γ_2 is close to -1. Conversely, the performance improves when γ_2 is slightly negative or positive. These results validate our discussion of the forgetting behavior in the two-feature case, as presented in Appendix H.1.

As an extension of our synthetic-data experiments, we further investigate the OOD generalization and forgetting behavior of a standard multi-layer *TinyTransformer*, as discussed in appendix A. We observe that the OOD error exhibits two distinct patterns depending on the direction of rotation, both of which are consistent with our theoretical predictions. We also analyze the forgetting behavior and track several representative cases to examine how performance evolves during fine-tuning. A more detailed discussion is provided in Appendix A.

6.2 Experiments with BERT on Real-World Data

In this section, we present experimental results using a BERT-base model on the Microsoft Research Paraphrase Corpus (MRPC) to validate the implications of our theory. We construct distribution shifts by applying a rotation to the hidden states before the last encoder layer. Specifically, we select a subset of dimensions $D_0 = \{d_{s_1}, d_{s_2}, \ldots, d_{s_{2n}}\}$. The rotation matrix is then defined as $R = \prod_{m=1}^{n} R_{s_{2m-1}, s_{2m}}$, where $R_{i,j}$ is defined in eq. (12). More details are provided in Appendix C.

To construct the rotation, we select 256 dimensions (one third of the total dimension) based on the magnitude of $|W_0 - W_1|$, where W_0 and W_1 denote the weight vectors of the binary classifier. In Scenario 1 and Scenario 2, we respectively select the 256 dimensions with the largest and the smallest values of $|W_0 - W_1|$. As shown in figs. 3a and 3b, these two rotation constructions induce distinct OOD generalization behaviors as the rotation angle θ varies. Notably, these behaviors match the two patterns predicted by our theoretical analysis in figs. 5a and 5b of Appendix A, further validating our theoretical insights beyond the theoretical assumptions.



(a) OOD generalization of BERT (b) OOD generalization of BERT (c) BERT Model Forgetting

Figure 3: OOD generalization and Forgetting of BERT model on MRPC

To evaluate forgetting, we randomly select 384 dimensions to rotate before the last layer and fine-tune only the last layer. After fine-tuning, we evaluate the cross-entropy loss on the source validation data. As shown in fig. 3c, the reported forgetting is averaged over 10 runs with error bars indicating standard

deviation. For comparison with fig. 2b, we vary the rotation angle from π to 0. As predicted by our theory, the model exhibits positive forgetting when the rotation angle approaches π. We observe that forgetting is minimal around $\frac{\pi}{2}$, suggesting that fine-tuning in an approximately orthogonal subspace has little negative impact on, and may even improve, generalization on the source domain.

6.3 Experiments of Natural Distribution Shift

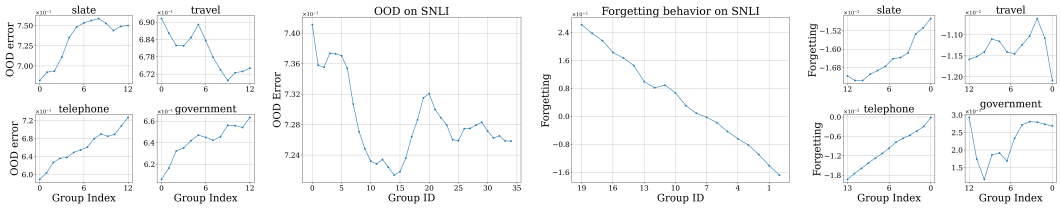
To provide evidence on a more natural distribution shift and support our conclusion and theoretical observation, we use BERT-base model and conduct experiments on two datasets: Multi-Genre Natural Language Inference (MNLI) and Stanford Natural Language Inference (SNLI). For dataset MNLI, there are five genres in total: *'fiction', 'government', 'telephone', 'travel', 'slate'*, with around 80k samples in each genre. During the training phase, the model is trained over *'fiction'* genre. Unlike our synthetic experiments, where we can arbitrarily rotate the latent space, real-world distribution shifts are inherently limited. Therefore, we construct different levels of distribution shift as follows (with full experimental details provided in Appendix C).

Small OOD Shift: Since the model is trained on the *fiction* genre, we treat the remaining genres in the MNLI dataset (i.e., *government, telephone, travel, slate*) as exhibiting relatively small OOD shifts. Within each OOD genre, we further partition the data into groups (with size 10K) using a sliding-window procedure based on the cosine similarity between each sample's embedding and the average embedding (center) of the fiction genre.

Large OOD Shift: We treat SNLI as the domain with large OOD shift. Similar to MNLI, we partition the data into groups (with size 20K) based on the cosine similarity between each sample's embedding and the average embedding (center) of the fiction genre.

To align with fig. 2a, we sort the groups in figs. 4a and 4b in decreasing order of cosine similarity, so that the x-axis reflects distribution shift from low to high. As shown in fig. 4a, the OOD error for *government*, *telephone*, and *slate* exhibits a clear increasing trend as the distribution shift becomes more pronounced, whereas the OOD error for *travel* fluctuates within a relatively narrow range. This behavior is consistent with our synthetic results (fig. 2a) under small rotation angles. For larger distribution shifts, we observe a "decreasing–increasing–decreasing" pattern, as illustrated in fig. 4b, which also aligns with fig. 2a when the rotation angle is large.

To analyze forgetting behavior, we instead sort the groups in increasing order of cosine similarity in figs. 4c and 4d, such that the x-axis represents distribution shift from high to low, consistent with fig. 2b. Under large distribution shifts, we observe a clear decreasing trend, as shown in fig. 4c, which matches the pattern in fig. 2b when the correlation coefficient γ is negative. Furthermore, as shown in fig. 4d, the forgetting for *government*, *telephone*, and *slate* increases steadily as the distribution shift becomes more pronounced, while the forgetting for *travel* again remains relatively stable. This observation is consistent with fig. 2b when the correlation coefficient γ approaches 1.



(a) OOD error on MNLI (b) OOD error on SNLI (c) Forgetting on SNLI (d) Forgetting on MNLI

Figure 4: OOD and forgetting behavior on MNLI and SNLI. Curves use moving average (MNLI: window=2, SNLI: window=5)

A potential implication of this work is that fine-tuning data in domain adaptation should be calibrated to the magnitude of distribution shift. When the shift is small, fine-tuning on the target domain can potentially improve generalization on the source domain; when the shift is large, catastrophic forgetting becomes more severe and degrades source-domain performance. A practical strategy is therefore to account for shift severity when constructing fine-tuning datasets, for example by selecting or reweighting samples according to their similarity to the source distribution, so as to better balance target adaptation and knowledge retention.

7 Conclusion

In this work, we derive an explicit expression for the OOD error of a single-layer softmax transformer and characterize how the geometric structure of distribution shifts influences OOD generalization. We further analyze the fine-tuning dynamics of the transformer on the OOD domain and quantify performance degradation caused by forgetting. We design both synthetic and real-world experiments to demonstrate that our insights can be extended beyond the theoretical regime. **Limitations:** Our theoretical analysis is restricted to one-layer softmax transformers, and the characterization of forgetting relies on a structural assumption on feature correlations. Extending the theory toward more practical settings is an interesting but open future direction for the community.

Acknowledgments and Disclosure of Funding

The work of J. Deng and Y. Liang is supported in part by the U.S. National Science Foundation via the grants ECCS-2515482, MFAI 2601957, and AI-EDGE Institute CNS-2112471.

References

[1] Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. *arXiv preprint arXiv:2306.00297*, 2023.

[2] Kabir Ahuja, Madhur Panwar, and Navin Goyal. In-context learning through the bayesian prism. *arXiv preprint arXiv:2306.04891*, 2023.

[3] Kartik Ahuja and David Lopez-Paz. A closer look at in-context learning under distribution shifts. *arXiv preprint arXiv:2305.16704*, 2023.

[4] Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.

[5] Usman Anwar, Johannes von Oswald, Louis Kirsch, David Krueger, and Spencer Frei. Understanding in-context learning of linear models in transformers through an adversarial lens. *Transactions on Machine Learning Research*, 2025.

[6] Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *arXiv preprint arXiv:2306.04637*, 2023.

[7] Amanda Bertsch, Maor Ivgi, Emily Xiao, Uri Alon, Jonathan Berant, Matthew R Gormley, and Graham Neubig. In-context learning with long-context models: An in-depth exploration. *arXiv preprint arXiv:2405.00200*, 2024.

[8] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of EMNLP*, pages 632–642, 2015.

[9] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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 *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[10] Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, James McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. *Advances in neural information processing systems*, 35:18878–18891, 2022.

[11] Siyu Chen, Heejune Sheen, Tianhao Wang, and Zhuoran Yang. Training dynamics of multi-head softmax attention for in-context learning: Emergence, convergence, and optimality. *arXiv preprint arXiv:2402.19442*, 2024.

[12] Siyu Chen, Heejune Sheen, Tianhao Wang, and Zhuoran Yang. Unveiling induction heads: Provable training dynamics and feature learning in transformers. *Advances in Neural Information Processing Systems(Neurips)*, 37:66479–66567, 2024.

[13] Frank Cole, Yulong Lu, Wuzhe Xu, and Tianhao Zhang. In-context learning of linear systems: Generalization theory and applications to operator learning. *arXiv preprint arXiv:2409.12293*, 2024.

[14] Liam Collins, Advait Parulekar, Aryan Mokhtari, Sujay Sanghavi, and Sanjay Shakkottai. In-context learning with transformers: Softmax attention adapts to function lipschitzness. *Advances in Neural Information Processing Systems*, 37:92638–92696, 2024.

[15] Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can gpt learn in-context? language models implicitly perform gradient descent as meta-optimizers. *arXiv preprint arXiv:2212.10559*, 2022.

[16] Junze Deng, Qinhang Wu, Peizhong Ju, Sen Lin, Yingbin Liang, and Ness Shroff. Unlocking the power of rehearsal in continual learning: A theoretical perspective. *arXiv preprint arXiv:2506.00205*, 2025.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[18] William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing*, 2005.

[19] Spencer Frei and Gal Vardi. Trained transformer classifiers generalize and exhibit benign overfitting in-context. In *International Conference on Learning Representations*, volume 2025, pages 32438–32467, 2025.

[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 *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[21] Angeliki Giannou, Shashank Rajput, Jy-yong Sohn, Kangwook Lee, Jason D Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. *arXiv preprint arXiv:2301.13196*, 2023.

[22] Chase Goddard, Lindsay M Smith, Vudtiwat Ngampruetikorn, and David J Schwab. When can in-context learning generalize out of task distribution? *arXiv preprint arXiv:2506.05574*, 2025.

[23] Tianyu Guo, Wei Hu, Song Mei, Huan Wang, Caiming Xiong, Silvio Savarese, and Yu Bai. How do transformers learn in-context beyond simple functions? a case study on learning with representations. *arXiv preprint arXiv:2310.10616*, 2023.

[24] Michael Hahn and Navin Goyal. A theory of emergent in-context learning as implicit structure induction. *arXiv preprint arXiv:2303.07971*, 2023.

[25] Chi Han, Ziqi Wang, Han Zhao, and Heng Ji. In-context learning of large language models explained as kernel regression. *arXiv preprint arXiv:2305.12766*, 2023.

[26] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8340–8349, 2021.

[27] Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. Pretrained transformers improve out-of-distribution robustness. *arXiv preprint arXiv:2004.06100*, 2020.

[28] Arian Hosseini, Ankit Vani, Dzmitry Bahdanau, Alessandro Sordoni, and Aaron Courville. On the compositional generalization gap of in-context learning. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 272–280, 2022.

[29] Alexander Hsu, Zhaiming Shen, Wenjing Liao, and Rongjie Lai. Understanding in-context learning for nonlinear regression with transformers: Attention as featurizer. *arXiv preprint arXiv:2605.05176*, 2026.

[30] Yu Huang, Yuan Cheng, and Yingbin Liang. In-context convergence of transformers. In *Proc. International Conference on Machine Learning (ICML)*, 2024.

[31] Hong Jun Jeon, Jason D Lee, Qi Lei, and Benjamin Van Roy. An information-theoretic analysis of in-context learning. *arXiv preprint arXiv:2401.15530*, 2024.

[32] Hui Jiang. A latent space theory for emergent abilities in large language models. *arXiv preprint arXiv:2304.09960*, 2023.

[33] Yanhao Jin, Krishnakumar Balasubramanian, and Lifeng Lai. In-context learning for mixture of linear regressions: Existence, generalization and training dynamics. *arXiv preprint arXiv:2410.14183*, 2024.

[34] Soo Min Kwon, Alec S Xu, Can Yaras, Laura Balzano, and Qing Qu. Out-of-distribution generalization of in-context learning: A low-dimensional subspace perspective. *arXiv preprint arXiv:2505.14808*, 2025.

[35] Hongkang Li, Meng Wang, Songtao Lu, Xiaodong Cui, and Pin-Yu Chen. Training nonlinear transformers for efficient in-context learning: A theoretical learning and generalization analysis. *arXiv preprint arXiv:2402.15607*, 2024.

[36] Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. Revisiting catastrophic forgetting in large language model tuning. In *Findings of the association for computational linguistics: EMNLP 2024*, pages 4297–4308, 2024.

[37] Yingcong Li, M Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. Transformers as algorithms: Generalization and implicit model selection in in-context learning. *arXiv preprint arXiv:2301.07067*, 2023.

[38] Zihao Li, Yuan Cao, Cheng Gao, Yihan He, Han Liu, Jason Klusowski, Jianqing Fan, and Mengdi Wang. One-layer transformer provably learns one-nearest neighbor in context. *Advances in Neural Information Processing Systems*, 37:82166–82204, 2024.

[39] Haodong Liang, Krishnakumar Balasubramanian, and Lifeng Lai. Transformers handle endogeneity in in-context linear regression. *arXiv preprint arXiv:2410.01265*, 2024.

[40] Tianyin Liao, Ziwei Zhang, Yufei Sun, Chunyu Hu, and Jianxin Li. Invariant graph transformer for out-of-distribution generalization. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 807–818, 2026.

[41] Fei Lu and Yue Yu. Transformer learns the cross-task prior and regularization for in-context learning. *arXiv preprint arXiv:2505.12138*, 2025.

[42] Arvind Mahankali, Tatsunori B Hashimoto, and Tengyu Ma. One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention. *arXiv preprint arXiv:2307.03576*, 2023.

[43] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

[44] Eshaan Nichani, Alex Damian, and Jason D Lee. How transformers learn causal structure with gradient descent. In *International Conference on Machine Learning (ICML)*, 2024.

[45] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

[46] Ruifeng Ren and Yong Liu. Towards understanding how transformers learn in-context through a representation learning lens. *Advances in Neural Information Processing Systems*, 37:892–933, 2024.

[47] Wei Shen, Ruida Zhou, Jing Yang, and Cong Shen. On the training convergence of transformers for in-context classification of gaussian mixtures. *arXiv preprint arXiv:2410.11778*, 2024.

[48] Zhaiming Shen, Alexander Hsu, Rongjie Lai, and Wenjing Liao. Understanding in-context learning on structured manifolds: Bridging attention to kernel methods. In *International Conference on Learning Representations*, volume 2026, pages 42067–42103, 2026.

[49] Jiajun Song, Zhuoyan Xu, and Yiqiao Zhong. Out-of-distribution generalization via composition: a lens through induction heads in transformers. *Proceedings of the National Academy of Sciences*, 122(6):e2417182122, 2025.

[50] Tao Sun, Cheng Lu, Tianshuo Zhang, and Haibin Ling. Safe self-refinement for transformer-based domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7191–7200, 2022.

[51] Anushka Tiwari and Kaiyi Ji. Turning back without forgetting: Selective backward refinement for parameter-efficient continual learning. *arXiv preprint arXiv:2606.01379*, 2026.

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[53] Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning (ICML)*, 2023.

[54] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[55] Mingze Wang, Ruoxi Yu, Weinan E, and Lei Wu. How transformers implement induction heads: Approximation and optimization analysis. *arXiv preprint arXiv:2410.11474*, 2024.

[56] Qixun Wang, Yifei Wang, Xianghua Ying, and Yisen Wang. Can in-context learning really generalize to out-of-distribution tasks? In *The Thirteenth International Conference on Learning Representations*, 2024.

[57] Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. Large language models are implicitly topic models: Explaining and finding good demonstrations for in-context learning. In *Workshop on Efficient Systems for Foundation Models at ICML*, 2023.

[58] Zixuan Wang, Stanley Wei, Daniel Hsu, and Jason D Lee. Transformers provably learn sparse token selection while fully-connected nets cannot. In *International Conference on Machine Learning (ICML)*, 2024.

[59] Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*, 2023.

[60] Noam Wies, Yoav Levine, and Amnon Shashua. The learnability of in-context learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[61] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL-HLT*, pages 1112–1122, 2018.

[62] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

[63] Tongkun Xu, Weihua Chen, Pichao Wang, Fan Wang, Hao Li, and Rong Jin. Cdtrans: Cross-domain transformer for unsupervised domain adaptation. *arXiv preprint arXiv:2109.06165*, 2021.

[64] Xingcheng Xu, Zihao Pan, Haipeng Zhang, and Yanqing Yang. It ain't that bad: understanding the mysterious performance drop in ood generalization for generative transformer models. *arXiv preprint arXiv:2308.08268*, 2023.

[65] Tong Yang, Yu Huang, Yingbin Liang, and Yuejie Chi. In-context learning with representations: Contextual generalization of trained transformers. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[66] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

[67] Chongzhi Zhang, Mingyuan Zhang, Shanghang Zhang, Daisheng Jin, Qiang Zhou, Zhongang Cai, Haiyu Zhao, Xianglong Liu, and Ziwei Liu. Delving deep into the generalization of vision transformers under distribution shifts. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 7277–7286, 2022.

[68] 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.

[69] Xingxuan Zhang, Jiansheng Li, Wenjing Chu, Junjia Hai, Renzhe Xu, Yuqing Yang, Shikai Guan, Jiazheng Xu, and Peng Cui. On the out-of-distribution generalization of multimodal large language models. *arXiv preprint arXiv:2402.06599*, 2024.

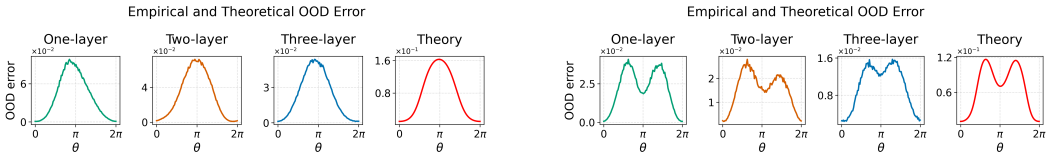
[70] Xingxuan Zhang, Jiansheng Li, Wenjing Chu, Renzhe Xu, Yuqing Yang, Shikai Guan, Jiazheng Xu, Liping Jing, Peng Cui, et al. On the out-of-distribution generalization of large multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10315–10326, 2025.

[71] Yufeng Zhang, Fengzhuo Zhang, Zhuoran Yang, and Zhaoran Wang. What and how does in-context learning learn? bayesian model averaging, parameterization, and generalization. *arXiv preprint arXiv:2305.19420*, 2023.

[72] Siyan Zhao, Tung Nguyen, and Aditya Grover. Probing the decision boundaries of in-context learning in large language models. *Advances in Neural Information Processing Systems*, 37:130408–130432, 2024.

A Experiments with TinyTransformer on Synthetic Data

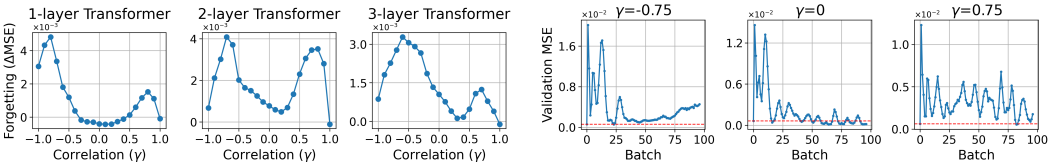
To evaluate whether our theory generalizes to more practical Transformer architectures, we study a standard 1-head *TinyTransformer* with 16-dimensional embeddings and varying depths. Using the same rotation matrix $R_{1,2}$, we generate random embeddings $\mathbf{V}$ to simulate rotations along different directions. Among various results, we observe two distinct patterns of OOD behaviors, as illustrated in fig. 5.



(a) Scenario 1 (b) Scenario 2

Figure 5: OOD Generalization on Standard Transformer

In Scenario 1, the OOD error increases monotonically as the rotation angle ranges from 0 to π, which aligns with the intuition that larger distribution shifts lead to higher OOD error. In contrast, Scenario 2 exhibits a "double-peak" phenomenon when the rotation is applied along specific directions. To explain this, we note that the OOD error is closely related to α-type attention, whose form is of $u^\top Qu$. Our analysis reveals that the eigenvectors of the parameter matrix Q are trained to align with training features, which indicates that the α-type attention can be approximated by $\sum_k \lambda_k (v_k^\top u)^2$ for some eigenvalues λ_k. An example of Scenario 2 arises when u is rotated within $\mathrm{span}(v_1, v_o)$, where $v_o \notin V$. Under this rotation, the projection of u onto v_1 is $v_1 \cos(\theta)$, and its projections onto all other training features are zero. Then, the α-type attention has the same trend as $\cos^2(\theta)$ and the trend of OOD error is opposite.



(a) Forgetting vs. Correlation for 1-layer, 2-layer, and 3-Layer Transformers

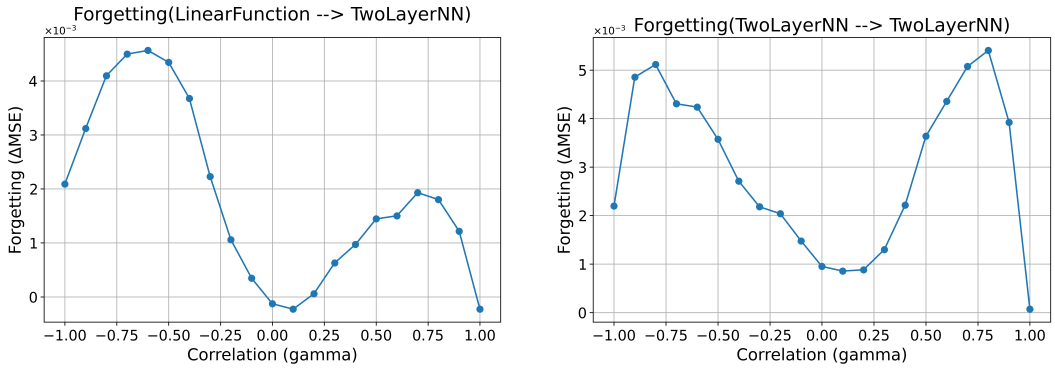
(b) Performance on Training Domain During Fine-tuning under $\gamma = -0.75, 0, 0.75$

Figure 6: Forgetting Behavior on Standard Transformer

Moreover, we investigate the behavior of forgetting of standard transformers with two features to validate our observation in fig. 2b. We investigate one, two and three-layers transformers and fine-tune on their last layer. As shown in fig. 6a, we observe that the changing behaviors of the forgetting are similar to those captured by our theory, i.e., fig. 2b. When the correlation coefficient approaches 1, the forgetting is not increasing as expected because the model is fine-tuned on almost the same data distribution, which leads to a forgetting near 0. To have a closer look at the 1-layer transformer in fig. 6a, we track the performance on the training domain under three distinct correlation condition: -0.75, 0 and 0.75, as shown in fig. 6b. The forgetting is positive when $\gamma = \pm 0.75$, whereas it becomes negative when $\gamma = 0$.

B Extension of Multi-layer Transformer

As an extension, we further investigate the behavior of forgetting under two distinct function classes, as shown in appendix B. We conduct experiments on one, two and three-layer standard 16-dim *TinyTransformer*. The data is generated the same as fig. 2b. Clearly, the performance of forgetting reveals a similar behavior as two linear function classes, which further validates our theory.



(a) Linear function class to two-layer NN class

(b) Two-layer NN class to two-layer NN class

Figure 7: Forgetting Behavior Across Different Function Classes

C Setup of Bert Experiments

Dataset. We conduct experiments on the Microsoft Research Paraphrase Corpus (MRPC), which contains 3,668 training pairs, 408 validation pairs, and 1,725 test pairs. The training data is further split into two subsets: 90% for the training phase and the remaining 10% for the fine-tuning phase. Unless otherwise stated, all evaluations are performed on the validation set.

Model. We use the `BERT-base-uncased` model, which consists of 12 Transformer encoder layers with a hidden dimension of 768. A linear classification head is applied to the `[CLS]` token representation for binary classification.

Preprocessing. We tokenize inputs using the BERT tokenizer with a maximum sequence length of 128. All sequences are padded or truncated to this length.

GPU information. All experiments were conducted on a high-performance computing cluster equipped with NVIDIA Volta V100 GPUs (16GB/32GB memory per GPU) and Intel Xeon CPU processors. We used 1 GPU per job for training and evaluation.

Configuration of training phase in Section 6.2 The model is trained using the AdamW optimizer with a learning rate of 5×10^{-5} and a batch size of 8. We initialize our model from the pretrained BERT-base-uncased checkpoint and train it on the last six encoder layers. Training is performed for 2 epochs using the cross-entropy loss.

Configuration of fine-tuning phase in Section 6.2 The model is trained using the AdamW optimizer with a learning rate of 1×10^{-5} and a batch size of 32. We initialize the model with the weights obtained from the training phase and fine-tune only the last encoder layer. Fine-tuning is performed for 1 epoch using the cross-entropy loss.

Configuration of training phase in Section 6.3 The model is trained using the AdamW optimizer with a learning rate of 5×10^{-5} and a batch size of 32. We initialize our model from the pretrained BERT-base-uncased checkpoint and train it on all the 12 encoder layers. Training is performed for 3 epochs using the cross-entropy loss.

Configuration of fine-tuning phase in Section 6.3 The model is trained using the AdamW optimizer with a learning rate of 2×10^{-6} and a batch size of 32. We initialize the model with the weights obtained from the training phase and fine-tune only the last encoder layer. Fine-tuning is performed for only 1 epoch using the cross-entropy loss.

Existing assets and licenses. We use BERT-base-uncased [17], MRPC [18], MultiNLI [61], and SNLI [8]. BERT-base-uncased is released under the Apache License 2.0. MRPC is released under the Microsoft Research Shared Source License Agreement (MSR-SSLA), which permits non-commercial

use, including academic research. SNLI is released under CC BY-SA 4.0. MultiNLI primarily uses the OANC license; its fiction sources include works under CC BY 3.0 and CC BY-SA 3.0, as well as works in the public domain in the United States.

D Supporting Lemmas

Notations. For simplification, we adopt the following notations. For any feature v, v' ($v, v' \in V$ or $v, v' \in U$, $v \neq v'$), we have:

$$\begin{aligned}\alpha_v^{(t)} &= \alpha_v(Q^{(t)}), \qquad \beta_{v',v}^{(t)} = \beta_{v',v}(Q^{(t)}) \\ \delta_{v,\alpha}^{(t)} &= -v^\top \nabla L^{(\text{ft})}(Q^{(t)})v, \\ \delta_{v',v,\beta}^{(t)} &= -{v'}^\top \nabla L^{(\text{ft})}(Q^{(t)})v.\end{aligned}$$

For all $k \in [K]$, we denote $\mathbf{Attn}_v^{(t)} = \mathbf{Attn}_v(Q^{(t)}; E)$. Additionally, we denote

$$\alpha^{(*)} = \left(1 - \frac{1}{K}\right) \log \left(\frac{(K-1)\left(\sqrt{\frac{K}{2(K-1)}} - \sqrt{\epsilon}\right)}{\sqrt{\epsilon}} \right). \tag{13}$$

In this section, we present our supporting lemmas. First of all, we derive the expression of the prediction formally, which is introduced in eq. (8).

Lemma D.1. *Given a model parameter Q, for any embedding matrix E with task parameter w, the prediction output is given by:*

$$\hat{y}(Q; E) = \sum_{v \in W}^{K} \boldsymbol{Attn}_v(Q; E)\langle w, v\rangle,$$

where $W = V$ if the prompt is constructed based on V and $W = U$ if the prompt is constructed based on U.

Proof. The proof follows by combining eqs. (2) and (7). □

We next provide the following lemma on the expression of the entry-wise gradient in eq. (4).

Lemma D.2. *Given a model parameter Q and an embedding E defined in eq.* (1)*, let $\hat{y}_{query} = \hat{y}_{query}(Q; E)$, $E_{x,i}$ be the i^{th} column of E_x, and $\boldsymbol{Attn}_v = \boldsymbol{Attn}_v(Q; E)$. The gradient of the fine-tuning loss defined in eq.* (5) *w.r.t. Q is given by*

$$\nabla L^{(ft)}(Q) = \mathbb{E}\left[(\hat{y}_{query} - \langle w, x_{query}\rangle) \sum_{i,l \in [N]} \boldsymbol{Attn}_{u_i} \boldsymbol{Attn}_{u_l} (E_{x,i} - E_{x,l}) {E_{x,N+1}}^\top y_i \right],$$

where the expectation is taken over the distribution of the embedding matrix E during fine-tuning phase and the corresponding functions w. Similarly, the gradient of the training loss defined in eq. (3) *w.r.t. Q is:*

$$\nabla L^{(tr)}(Q) = \mathbb{E}\left[(\hat{y}_{query} - \langle w, x_{query}\rangle) \sum_{i,l \in [N]} \boldsymbol{Attn}_{v_i} \boldsymbol{Attn}_{v_l} (E_{x,i} - E_{x,l}) {E_{x,N+1}}^\top y_i \right],$$

where the expectation is taken over the distribution of the training embedding matrix E and the corresponding functions w.

Proof. The proof follows from Lemma A.1 in [30]. □

By combining eqs. (4) and (9), the updates of the two types of attention weights in the fine-tuning phase are given as follows. For any $u, u' \in U$ ($u \neq u'$), we have

$$\alpha_u^{(t+1)} = \alpha_u^{(t)} - \eta_t u^\top \nabla L^{(\text{ft})}(Q^{(t)})u$$

$$
\begin{aligned}
&= \alpha_u^{(t)} + \eta_t \delta_{u,\alpha}^{(t)}, \\
\beta_{u',u}^{(t+1)} &= \beta_{u',u}^{(t)} - \eta_t u'^{\top} \nabla L^{(\mathrm{ft})}(Q^{(t)}) u \\
&= \beta_{u',u}^{(t)} + \eta_t \delta_{u',u,\beta}^{(t)}.
\end{aligned} \tag{14}
$$

We present the following lemma to calculate the projection of gradients onto feature directions.

Lemma D.3. *Suppose $u, u' \in U$ where $u \neq u'$. For any $t \geq T_1$, we have:*

$$
\begin{aligned}
\delta_{u,\alpha}^{(t)} &= \mathbb{E}\left[\mathbf{1}_{\{x_{query}=u\}} \boldsymbol{Attn}_u^{(t)} \left(\sum_{w\in U, w\neq u} \boldsymbol{Attn}_w^{(t)\,2} + \left(1 - \boldsymbol{Attn}_u^{(t)}\right)^2\right)\right], \\
\delta_{u',u,\beta}^{(t)} &= \mathbb{E}\left[\mathbf{1}_{\{x_{query}=u\}} \boldsymbol{Attn}_{u'}^{(t)} \left(\sum_{w\in U, w\neq u} \boldsymbol{Attn}_w^{(t)\,2} - \boldsymbol{Attn}_{u'}^{(t)} - \boldsymbol{Attn}_u^{(t)}(1 - \boldsymbol{Attn}_u^{(t)})\right)\right],
\end{aligned}
$$

where the expectation is taken on the distribution of the embedding matrix E.

Proof. The proof follows from Lemma A.2 in [30]. □

In the following lemma, we establish the connection between $\delta_{u,\alpha}^{(t)}$ and $\delta_{u',u,\beta}^{(t)}$, which is introduced in Lemma 5.3.

Lemma D.4. *For any time t and any $u \in U$, we have:*

$$
\delta_{u,\alpha}^{(t)} = -\sum_{u'\in U, u'\neq u} \delta_{u',u,\beta}^{(t)}.
$$

Proof. The proof is developed by directly deriving $\sum_{u'\in U, u'\neq u} \delta_{u',u,\beta}^{(t)}$ as follows.

$$
\begin{aligned}
&\sum_{u'\in U, u'\neq u} \delta_{u',u,\beta}^{(t)} \\
=& \sum_{u'\in U, u'\neq u} \mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u\}} \mathbf{Attn}_{u'}^{(t)} \left(\sum_{w\in U, w\neq u} \mathbf{Attn}_w^{(t)\,2} - \mathbf{Attn}_{u'}^{(t)} - \mathbf{Attn}_u^{(t)}(1 - \mathbf{Attn}_u^{(t)})\right)\right] \\
=& \mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u\}} \left(\sum_{u'\in U, u'\neq u} \mathbf{Attn}_{u'}^{(t)}\right) \cdot \left(\sum_{w\in U, w\neq u} \left(\mathbf{Attn}_w^{(t)}\right)^2 - \mathbf{Attn}_u^{(t)} \left(1 - \mathbf{Attn}_u^{(t)}\right)\right)\right] \\
&- \mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u\}} \sum_{u'\in U, u'\neq u} \left(\mathbf{Attn}_{u'}^{(t)}\right)^2\right] \\
=& \mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u\}} (1 - \mathbf{Attn}_u^{(t)}) \cdot \left(\sum_{w\in U, w\neq u} \left(\mathbf{Attn}_w^{(t)}\right)^2 - \mathbf{Attn}_u^{(t)} \left(1 - \mathbf{Attn}_u^{(t)}\right)\right)\right] \\
&- \mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u\}} \sum_{u'\in U, u'\neq u} \left(\mathbf{Attn}_{u'}^{(t)}\right)^2\right] \\
=& \mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u\}} \left(-\mathbf{Attn}_u^{(t)} \sum_{w\in U, w\neq u} \left(\mathbf{Attn}_w^{(t)}\right)^2 - \mathbf{Attn}_u^{(t)} \left(1 - \mathbf{Attn}_u^{(t)}\right)^2\right)\right] \\
=& -\mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u\}} \left(\mathbf{Attn}_u^{(t)}\right) \cdot \left(\sum_{w\in U, w\neq u} \left(\mathbf{Attn}_w^{(t)}\right)^2 + \left(1 - \mathbf{Attn}_u^{(t)}\right)^2\right)\right] \\
=& -\delta_{u,\alpha}^{(t)}.
\end{aligned}
$$

□

This property can be also observed during the training phase by following from the same argument. We present it formally as follows.

Lemma D.5. *For any time t and any $v \in V$, we have:*

$$v^\top \nabla L^{(tr)}(Q^{(t)})v = -\sum_{v' \in V, v' \neq v} {v'}^\top \nabla L^{(tr)}(Q^{(t)})v.$$

According to Lemma D.3, we observe that the update of β-type attention during the training phase has a symmetric property because their initialization condition remains the same. We formally present our observation in the following two lemmas.

Lemma D.6. *Consider the problem formulated in this work. Suppose the model is in the training phase. Given any $v \in V$, we have:*

$$\beta_{v',v}^{(t)} = \beta_{v'',v}^{(t)}.$$

Proof. We prove this lemma by mathematical induction. When $t = 0$, we have $\beta_{v',v}^{(0)} = \beta_{v'',v}^{(0)}$ according to the problem condition. We assume that $\beta_{v',v}^{(t)} = \beta_{v'',v}^{(t)}$ holds and prove that it also holds for time $t+1$. Equivalently, we only need to prove $\delta_{v',v,\beta}^{(t)} = \delta_{v'',v,\beta}^{(t)}$. Under our assumption that the prompts consist of the same number of features from their corresponding feature sets, we have:

$$\mathbf{Attn}_{v'}^{(t)} = \frac{\exp(\beta_{v',v}^{(t)})}{\exp(\alpha_v^{(t)}) + \sum_{v' \in v', v \neq v} \exp(\beta_{v',v}^{(t)})},$$

$$\mathbf{Attn}_{v''}^{(t)} = \frac{\exp(\beta_{v'',v}^{(t)})}{\exp(\alpha_v^{(t)}) + \sum_{v'' \in V, v'' \neq v} \exp(\beta_{v'',v}^{(t)})}.$$

Then, we have $\mathbf{Attn}_{v'}^{(t)} = \mathbf{Attn}_{v''}^{(t)}$. By combining this observation and Lemma D.3, we complete the proof.

□

Lemma D.7. *Consider the problem formulated in this work. Suppose the model is in the training phase. Given $v \in V$, for all $v' \in V (v \neq v')$, we have:*

$$\beta_{v',v}^{(t)} = -\frac{\alpha_v^{(t)}}{K-1}.$$

Proof. First of all, we have:

$$\alpha_v^{(t)} = \alpha_v^{(0)} - \sum_{t'=0}^{t-1} \eta\, v^\top \nabla L^{(\text{tr})}(Q^{(t')})v,$$

$$\beta_{v',v}^{(t)} = \beta_{v',v}^{(0)} - \sum_{t'=0}^{t-1} \eta\, {v'}^\top \nabla L^{(\text{tr})}(Q^{(t')})v.$$

By combining Lemma D.5 and the fact that $\alpha_v^{(0)} = \beta_{v,v'}^{(0)} = 0$ for all $v' \neq v$, we have $\alpha_v^{(t)} = -\sum_{v' \in V, v' \neq v} \beta_{v,v'}^{(t)}$. Then, by following from Lemma D.6, we complete the proof. □

In the following two lemmas, we present the expression of the fine-tuning loss function $L_u^{(ft)}(Q)$ associated with feature $u \in U$ under any model parameter Q.

Lemma D.8. *Given any model parameter Q and any time t, the expression of $L_u^{(ft)}(Q^{(t)})$ can be derived as follows:*

$$L_u^{(ft)}(Q^{(t)}) = \frac{1}{2}\mathbb{E}\left[\mathbf{1}_{\{x_{query} = u\}}\left(\sum_{u' \in U, u' \neq u} \boldsymbol{Attn}_{u'}^{(t)^2} + \left(1 - \boldsymbol{Attn}_u^{(t)}\right)^2\right)\right],$$

where the expectation is taken over the distribution of the fine-tuning distribution of E.

Proof. The expression of $L_u^{(\text{ft})}(Q^{(t)})$ can be derived as follows.

$$\begin{aligned}
L_u^{(\text{ft})}(Q^{(t)}) &= \frac{1}{2}\mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u\}}\left(\hat{y}_{\text{query}}\left(Q^{(t)};E\right)-\langle w, x_{\text{query}}\rangle\right)^2\right] \\
&\overset{(i)}{=} \frac{1}{2}\mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u\}}\left(\sum_{u'\in U}\mathbf{Attn}_n^{(t)}\langle w, u'-u\rangle\right)^2\right] \\
&= \frac{1}{2}\mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u\}}\left(\sum_{\substack{u'\in U\\ u''\in U}}\mathbf{Attn}_{u'}^{(t)}\mathbf{Attn}_{u''}^{(t)}(u-u')^\top ww^\top(u-u'')\right)\right] \\
&\overset{(ii)}{=} \frac{1}{2}\mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u\}}\left(\sum_{\substack{u'\in U\\ u''\in U}}\mathbf{Attn}_{u'}^{(t)}\mathbf{Attn}_{u''}^{(t)}(u-u')^\top(u-u'')\right)\right] \\
&= \frac{1}{2}\mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u\}}\left\|u-\sum_{u'\in U}\mathbf{Attn}_{u'}^{(t)}u'\right\|^2\right] \\
&= \frac{1}{2}\mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u\}}\left(\sum_{u'\in U,u'\neq u}{\mathbf{Attn}_{u'}^{(t)}}^2+\left(1-\mathbf{Attn}_u^{(t)}\right)^2\right)\right].
\end{aligned}$$

where (i) follows from Lemma D.1, and (ii) follows the fact that the task w follows the standard Gaussian distribution. □

Lemma D.9. *Suppose $t \geq T_1$. Given any model parameter Q, the expression of $L_{u_k}^{(ft)}(Q^{(t)})$ can be derived as follows:*

$$L_{u_k}^{(ft)}(Q^{(t)}) = \frac{(\sum_{n\neq k}\exp(\beta_{u_n,u_k}^{(t)}))^2+\sum_{n\neq k}\exp(\beta_{u_n,u_k}^{(t)})^2}{2K(\exp(\alpha_{u_k}^{(t)})+\sum_{n\neq k}\exp(\beta_{u_n,u_k}^{(t)}))^2} \tag{15}$$

Proof. This lemma can be proved by directly combining Lemma D.8 and eq. (10). □

During fine-tuning phase, we project the gradient of fine-tuning loss onto training features. Under the Assumption 5.4, the detail expression are presented as follows.

Lemma D.10. *Consider the training phase and the fine-tuning phase formulated in this work. Suppose Assumption 5.4 holds. For $k, k' \in [K]$ and any $t \geq 0$, we have*

$$v_k^\top \nabla L^{(ft)}(Q^{(t)})v_{k'} = \gamma_k\gamma_{k'}u_k^\top\nabla L^{(ft)}(Q^{(t)})u_{k'}.$$

Proof. According to the expression of gradients as shown in Lemma D.2, we first observe that:

$$w^\top\nabla L^{(\text{ft})}(Q^{(t)})w' = 0 \tag{16}$$

if $w \notin \text{span}(U)$ or $w' \notin \text{span}(U)$. Let $\{v_i\}_{i=1}^d$ denote an orthonormal basis of $\mathbb{R}^d$ obtained by augmenting the training features V. Suppose $u_k = \sum_{i=1}^d a_{k,i}v_i$ and $u_{k'} = \sum_{i=1}^d a_{k',i}v_i$. Then we have:

$$\begin{aligned}
{v_k}^\top\nabla L^{(\text{ft})}(Q^{(t)})v_{k'} &= \left(\sum_{i=1}^d a_{k,i}v_i\right)^\top\nabla L^{(\text{ft})}(Q^{(t)})\left(\sum_{i=1}^d a_{k',i}v_i\right) \\
&\overset{(i)}{=} \left(\sum_{i=1}^K a_{k,i}v_i\right)^\top\nabla L^{(\text{ft})}(Q^{(t)})\left(\sum_{i=1}^K a_{k',i}v_i\right) \\
&\overset{(ii)}{=} (\gamma_k v_k)^\top\nabla L^{(\text{ft})}(Q^{(t)})\left(\gamma_{k'}v_{k'}\right)
\end{aligned}$$

where (i) follows from eq. (16) and (ii) follows from the problem setups. □

To simplify our proof, we provide the following supporting algebraic lemmas.

Lemma D.11. *Suppose $a_i > 0$ for $i = 0, 1, \ldots, n$ with $a_0 \geq \max_{i\geq 1} a_i$. If $b_i < a_i$ for $i \in [n]$, we have:*

$$\frac{\sum_{i=1}^n a_i^2 + (\sum_{i=1}^n a_i)^2}{(\sum_{i=0}^n a_i)^2} \geq \frac{\sum_{i=1}^n b_i^2 + (\sum_{i=1}^n b_i)^2}{(a_0 + \sum_{i=1}^n b_i)^2}.$$

Proof. We calculate the partial gradient as follows. For any a_i with $i \geq 1$, we have

$$\begin{aligned}
\frac{\partial}{\partial a_j} \cdot \frac{\sum_{i=1}^n a_i^2 + (\sum_{i=1}^n a_i)^2}{(\sum_{i=0}^n a_i)^2} &= \frac{2\left[(a_j + \sum_{i=1}^n a_i)(\sum_{i=0}^n a_i) - (\sum_{i=1}^n a_i)^2 - (\sum_{i=1}^n a_i^2)\right]}{(\sum_{i=0}^n a_i)^3} \\
&\geq \frac{2\left[(\sum_{i=1}^n a_i)^2 + a_0(\sum_{i=1}^n a_i) - (\sum_{i=1}^n a_i)^2 - (\sum_{i=1}^n a_i^2)\right]}{(\sum_{i=0}^n a_i)^3} \\
&\overset{(i)}{\geq} 0,
\end{aligned}$$

where (i) follows from $a_0 \geq \max_{i\geq 1} a_i$. Therefore, we have

$$\frac{\sum_{i=1}^n a_i^2 + (\sum_{i=1}^n a_i)^2}{(\sum_{i=0}^n a_i)^2} \geq \frac{\sum_{i=1}^n b_i^2 + (\sum_{i=1}^n b_i)^2}{(a_0 + \sum_{i=1}^n b_i)^2}.$$

□

E Proof of OOD generalization in Theorem 5.1

In this section, we first characterize the training dynamics of the transformer during training. We then analyze its OOD generalization by evaluating the trained model on the OOD domain.

E.1 Training Phase

In this subsection, we focus on the the training phase. Although the convergence of the training phase has been established by previous work[30], they did not provide an explicit expression for the attention weights at convergence. In our setting, this expression is essential because the OOD generalization can be captured by using the attention weights obtained at the end of the training phase. Moreover, the difference between the attention weights at the end of the training phase and those of the fine-tuning phase directly determines the forgetting behavior, which is the primary focus of our analysis.

We first characterize the attention weights at the convergence of the training phase as follows. Recall that

$$\alpha^*_{K,\epsilon} = \left(1 - \frac{1}{K}\right) \log\left(\frac{(K-1)\left(\sqrt{\frac{K}{2(K-1)}} - \sqrt{\epsilon}\right)}{\sqrt{\epsilon}}\right).$$

Lemma E.1. *Consider the training phase formulated in this work. For any $\epsilon \in (0, 1)$ and a constant step size $\eta_t = \eta$, there exists $T_1 = poly(\frac{1}{\epsilon}, K, \eta)$ such that the training phase converges, i.e., $L_v^{(tr)}(Q^{(T_1)}) \leq \frac{\epsilon}{K}$ for all $v \in V$. At time T_1, we have:*

$$\begin{aligned}
\alpha_v^{(1,T_1)} &= \alpha^*_{K,\epsilon}, \\
\beta_{v',v}^{(1,T_1)} &= -\frac{\alpha^*_{K,\epsilon}}{K-1}, \quad \textit{for all } v' \neq v.
\end{aligned}$$

Proof. Suppose $x_{\text{query}} = v$. We fist claim the existence of time T_1. The convergence of the training phase can be proved by following from Lemma D.9 directly.

Then, we prove the existence of time T_1 as follows. Since $Q^{(0)}$ is a zero matrix, we have $\alpha_v^{(0)} = \beta_{v',v}^{(0)} = 0$ for all $v' \neq v$. Following from Lemma D.7, we have:

$$\beta_{v',v}^{(t)} = -\frac{\alpha_v^{(t)}}{K-1},$$

for all $t \geq 0$ and $v' \neq v$. Also, according to Lemma D.3, we have $\delta_{v,\alpha}^{(t)} \geq 0$ for all $t \geq 0$, which implies that $\alpha_v^{(t)}$ is monotonically increasing and non-negative. By combining these and eq. (10), we conclude that $\mathbf{Attn}_v^{(t)}$ is monotonically increasing and $\mathbf{Attn}_v^{(t)} \geq \mathbf{Attn}_v^{(0)} = \frac{1}{K}$. Based on the above arguments, we first prove the following statement. If $\alpha_v^{(t)} \leq \log\left(\frac{K}{\sqrt{\epsilon}}\right)$, we have:

$$\begin{aligned}
\mathbf{Attn}_v^{(t)} &\leq \frac{\exp\left(\log\left(\frac{K}{\sqrt{\epsilon}}\right)\right)}{\exp\left(\log\left(\frac{K}{\sqrt{\epsilon}}\right)\right) + (K-1)\exp\left(-\frac{1}{K-1}\cdot\log\left(\frac{K}{\sqrt{\epsilon}}\right)\right)} \\
&\leq \frac{\frac{K}{\sqrt{\epsilon}}}{\frac{K}{\sqrt{\epsilon}} + \left(\frac{K}{\sqrt{\epsilon}}\right)^{-1}} \\
&\leq \frac{K^2}{K^2+\epsilon}.
\end{aligned}$$

Therefore, if $\alpha_v^{(t)} \leq \log\left(\frac{K}{\sqrt{\epsilon}}\right)$, we have $\mathbf{Attn}_v^{(t)} \in \left[\frac{1}{K}, \frac{K^2}{K^2+\epsilon}\right]$. We further observe that under this condition, we have

$$\begin{aligned}
-v^\top \nabla L^{(\mathrm{tr})}(Q^{(t)})v &\geq \mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=v\}}\mathbf{Attn}_v^{(t)}(1-\mathbf{Attn}_v^{(t)})^2\right] \\
&\geq \max\left\{\frac{1}{K^2}\left(1-\frac{1}{K}\right)^2, \frac{K\epsilon^2}{(K^2+\epsilon)^3}\right\}. \qquad (17)
\end{aligned}$$

Given any time T, we have

$$\begin{aligned}
\alpha_v^{(T)} &= \alpha_v^{(0)} - \sum_{t=0}^{T} \eta \cdot v^\top \nabla L^{(\mathrm{tr})}(Q^{(t)})v \\
&\geq \eta T \cdot \max\left\{\frac{1}{K^2}\left(1-\frac{1}{K}\right)^2, \frac{K\epsilon^2}{(K^2+\epsilon)^3}\right\}
\end{aligned}$$

Thus, there exists $T_1 = \mathrm{poly}(\frac{1}{\epsilon}, K, \eta)$ s.t. if $\alpha_v^{(t)} \leq \log\left(\frac{K}{\sqrt{\epsilon}}\right)$ for $t \leq T_1$, we have

$$\alpha_v^{(T_1)} \geq \alpha_{K,\epsilon}^*.$$

Finally, we note that:

$$\alpha_{K,\epsilon}^* \leq \log\left(\frac{K}{\sqrt{\epsilon}}\right),$$

which implies that our claim holds within time T_1 and completes the proof. □

E.2 Proof of Theorem 5.1

In this subsection, we characterize the OOD generalization as stated in Theorem 5.1.

Theorem E.2. ***(Restatement of Theorem 5.1)*** *Suppose the training phase ends at time T_1, as defined in Lemma E.1. Define $\mathbf{M} = \mathbf{V}^\top \mathbf{U}$. The attention score of the testing feature are:*

$$\begin{aligned}
\alpha_{u_k}^{(T_1)} &= \frac{K\alpha_{K,\epsilon}^*}{K-1}\left([\mathbf{M}^\top\mathbf{M}]_{k,k} - \langle u_k, \bar{v}\rangle^2\right), \\
\beta_{u_k,u_{k'}}^{(T_1)} &= \frac{K\alpha_{K,\epsilon}^*}{K-1}\left([\mathbf{M}^\top\mathbf{M}]_{k',k} - \langle u_k, \bar{v}\rangle\langle u_{k'}, \bar{v}\rangle\right),
\end{aligned}$$

where $[\cdot]_{k',k}$ denote the element at $(k')^{th}$ row and k^{th} column, $\bar{v} = \frac{1}{\sqrt{K}}\sum_{i=1}^K v_i$. Then, the OOD error can be characterized by:

$$L_{u_k}^{(ft)}(Q^{(T_1)}) = \frac{(\sum_{n\neq k}\exp(\beta_{u_n,u_k}^{(T_1)}))^2 + \sum_{n\neq k}\exp(\beta_{u_n,u_k}^{(T_1)})^2}{2K(\exp(\alpha_{u_k}^{(T_1)}) + \sum_{n\neq k}\exp(\beta_{u_n,u_k}^{(T_1)}))^2}$$

Proof. Let $\{v_i\}_{i=1}^d$ denote an orthonormal basis of $\mathbb{R}^d$ obtained by augmenting the training features V. We write $u_k = \sum_{j=1}^d \langle u_k, v_j\rangle v_j$ and $u_{k'} = \sum_{j=1}^d \langle u_{k'}, v_j\rangle v_j$. Then, we have:

$$
\begin{aligned}
u_k^\top Q^{(T_1)} u_{k'} &= \left(\sum_{j=1}^d \langle u_k, v_j\rangle v_j\right)^\top Q^{(T_1)} \left(\sum_{j=1}^d \langle u_{k'}, v_j\rangle v_j\right) \\
&\overset{(i)}{=} \sum_{i=1}^K \sum_{j=1}^K \langle u_k, v_i\rangle \langle u_{k'}, v_j\rangle v_i^\top Q^{(T_1)} v_j \\
&= \sum_{i=1}^K \langle u_k, v_i\rangle \langle u_{k'}, v_i\rangle v_i^\top Q^{(T_1)} v_i + \sum_{i\neq j} \langle u_k, v_i\rangle \langle u_{k'}, v_j\rangle v_i^\top Q^{(T_1)} v_j \\
&\overset{(ii)}{=} \sum_{i=1}^K \langle u_k, v_i\rangle \langle u_{k'}, v_i\rangle \alpha^*_{K,\epsilon} - \sum_{i\neq j} \langle u_k, v_i\rangle \langle u_{k'}, v_j\rangle \frac{\alpha_{K,\epsilon^*}}{K-1} \\
&= \frac{K\alpha^*_{K,\epsilon}}{K-1} \sum_{i=1}^K \langle u_k, v_i\rangle \langle u_{k'}, v_i\rangle - \frac{\alpha_{K,\epsilon^*}}{K-1} \sum_{i,j=1}^K \langle u_k, v_i\rangle \langle u_{k'}, v_j\rangle \\
&\overset{(iii)}{=} \frac{K\alpha^*_{K,\epsilon}}{K-1} [\mathbf{M}^\top \mathbf{M}]_{k,k'} - \frac{K\alpha_{K,\epsilon^*}}{K-1} \langle u_k, \frac{1}{\sqrt{K}} \sum_{i=1}^K v_i\rangle \langle u_{k'}, \frac{1}{\sqrt{K}} \sum_{j=1}^K v_j\rangle \\
&= \frac{K\alpha^*_{K,\epsilon}}{K-1} [\mathbf{M}^\top \mathbf{M}]_{k,k'} - \frac{K\alpha^*_{K,\epsilon}}{K-1} \langle u_k, \bar{v}\rangle \langle u_{k'}, \bar{v}\rangle.
\end{aligned}
$$

where (i) follows from the same argument as eq. (16), (ii) follows from Lemma E.1 and (iii) follows from the definition of $\mathbf{M}$. By discussing $k = k'$ and $k \neq k'$, we complete the proof of α-type and β-type attention weights. Then, the expression of $L^{(\text{ft})}_{u_k}(Q^{(T_1)})$ can be derived by Lemma D.9.

□

E.3 Proof of Corollary 5.2

We restate and prove Corollary 5.2 as follows.

Corollary E.3. ***(Restatement of Corollary 5.2)*** *The following two statements hold.*

1. *Suppose span(V) = span(U). If the query token is u_k and $u_k = \bar{v}$, we have: **Attn**$_u = \frac{1}{K}, \quad \forall\, u \in U$.*
2. *Suppose span$(V) \perp$ span(U). For any query token, we have: **Attn**$_u = \frac{1}{K}, \quad \forall\, u \in U$.*

Proof. We first prove the statement item *1*. When span(V) = span(U), we have $\mathbf{M}^\top \mathbf{M} = \mathbf{I}_d$. Suppose the query token is u_k and $u_k = \bar{v}$. According to Theorem E.2, we have:

$$
\begin{aligned}
\alpha^{(T_1)}_{u_k} &= \frac{K\alpha^*_{K,\epsilon}}{K-1} \left([\mathbf{I}_d]_{k,k} - \|\bar{v}\|^2\right) \\
&= 0.
\end{aligned}
$$

For β-type attention weights, we have:

$$
\begin{aligned}
\beta^{(T_1)}_{u_{k'}, u_k} &= \frac{K\alpha^*_{K,\epsilon}}{K-1} \left([\mathbf{I}_d]_{k',k} - \|\bar{v}\| \cdot \langle u_{k'}, \bar{v}\rangle\right) \\
&\overset{(i)}{=} 0
\end{aligned}
$$

where (i) follows from the assumption that $\{u_k\}_{k\in[K]}$ are orthogonal to each other. Therefore, according to the definition of attention scores, we have $\mathbf{Attn}_u = \frac{1}{K}, \quad \forall\, u \in U$ if the query token is $\bar{v}$.

Then, we prove the statement item *2*. When $\text{span}(V) \perp \text{span}(U)$, we have $\mathbf{M}^\top \mathbf{M} = \mathbf{0}_d$. It also indicates that $\langle u_k, \bar{v} \rangle = 0$ for all $k \in [K]$. Suppose $x_{\text{query}} = u, \forall u \in U$. According to Theorem E.2, we have:

$$\alpha_u^{(T_1)} = 0$$
$$\beta_{u',u}^{(T_1)} = 0, \quad \forall u' \in U, u' \neq u.$$

Therefore, for any query token, we have $\mathbf{Attn}_u = \frac{1}{K}, \forall\, u \in U$ according to the definition of attention scores. □

F Proof of Theorem 5.5

In this section, we focus on the fine-tuning phase, where the model is initialized as $Q^{(T_1)}$ and further fine-tuned on the OOD domain U. We will characterize the fine-tuning dynamics by proving Theorem 5.5, showing that there exists a polynomial convergence time of the fine-tuning phase. To do this, we first provide our observations during the fine-tuning phase in the following subsection

F.1 Observations During the Fine-tuning Phase

We first characterize the connection between α-type and β-type attention associated with the same feature, as presented in Lemma 5.3.

Lemma F.1. ***(Lemma 5.3)*** *Let $\Delta_{k,\alpha}^{(t)}$ and $\Lambda_{n,k,\beta}^{(t)}$ denote the updates of α-type and β-type attention weights during the fine-tuning time t, respectively, such that:*

$$\alpha_{u_k}^{(T_1+t)} = \alpha_{u_k}^{(T_1)} + \Delta_{k,\alpha}^{(t)},$$
$$\beta_{u_n,u_k}^{(T_1+t)} = \beta_{u_n,u_k}^{(T_1)} - \Lambda_{n,k,\beta}^{(t)}, \quad \textit{for} \quad n \neq k.$$

Then, we have $\Delta_{k,\alpha}^{(t)} = \sum_{n \neq k} \Lambda_{n,k,\beta}^{(t)}$ at any time t

Proof. According to the definition, we have:

$$\Delta_{k,\alpha}^{(t)} = \sum_{t'=1}^{t} \eta_{t'} \cdot \delta_{u_k,\alpha}^{(t')}$$
$$\Lambda_{n,k,\beta}^{(t)} = -\sum_{t'=1}^{t} \eta_{t'} \cdot \delta_{u_n,u_k,\alpha}^{(t')}.$$

Combining this and Lemma D.4, we complete the proof. □

We observe that, during the fine-tuning phase, the α-type attention weights keep increasing while the β-type attention weights continue decreasing, i.e., $\Delta_{k,\alpha}^{(t)} > 0$ and $\Lambda_{n,k,\beta}^{(t)} > 0$, as mentioned in the main body of this paper. We present this observation formally in the following lemma.

Lemma F.2. *Consider the fine-tuning phase formulated in this work and let $\Delta_{k,\alpha}^{(t)}$ and $\Lambda_{n,k,\beta}^{(t)}$ be as defined in Lemma F.1. If $\epsilon > \left(\frac{K}{\exp((K-1)^2)}\right)^2$, we have: $\Delta_{k,\alpha}^{(t)} > 0$ and $\Lambda_{n,k,\beta}^{(t)} > 0$ for any time $t \geq T_1$.*

Proof. According to Lemma D.3, we first have:

$$\delta_{u_n,u_k,\beta}^{(t)} = \mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u_k\}} \mathbf{Attn}_{u_n}^{(t)} \left(\sum_{m \neq k} {\mathbf{Attn}_{u_m}^{(t)}}^2 - \mathbf{Attn}_{u_n}^{(t)} - \mathbf{Attn}_{u_k}^{(t)}(1 - \mathbf{Attn}_{u_k}^{(t)})\right)\right]$$

$$= \mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u_k\}}\mathbf{Attn}_{u_n}^{(t)}\left(\underbrace{\sum_{\substack{m\neq k\\ m\neq n}}{\mathbf{Attn}_{u_m}^{(t)}}^2 - \mathbf{Attn}_{u_n}^{(t)}(1-\mathbf{Attn}_{u_n}^{(t)}) - \mathbf{Attn}_{u_k}^{(t)}(1-\mathbf{Attn}_{u_k}^{(t)})}_{\text{term } g_t}\right)\right]. \tag{18}$$

Define $C_t = \exp(\alpha_{u_k}^{(t)}) + \sum_{m\neq k}\exp(\beta_{u_m,u_k}^{(t)})$. Then, we rewrite term c_1 as:

$$\begin{aligned}
\text{term } g_t &= \frac{1}{C_t^2}\left(\sum_{\substack{m\neq k\\ m\neq n}}\exp^2(\beta_{u_m,u_k}^{(t)}) - \exp(\beta_{u_n,u_k})(\exp(\alpha_{u_k}^{(t)}) + \sum_{\substack{m\neq k\\ m\neq n}}\exp(\beta_{u_m,u_k}^{(t)})) - \exp(\alpha_{u_k}^{(t)})(\sum_{m\neq k}\exp(\beta_{u_m,u_k}^{(t)}))\right)\\
&= \frac{1}{C_t^2}\left(\sum_{\substack{m\neq k\\ m\neq n}}\exp^2(\beta_{u_m,u_k}^{(t)}) - (\exp(\alpha_{u_k}^{(t)}) + \exp(\beta_{u_n,u_k}^{(t)}))\sum_{\substack{m\neq k\\ m\neq n}}\exp(\beta_{u_m,u_k}^{(t)}) - 2\exp(\alpha_{u_k}^{(t)})\exp(\beta_{u_n,u_k}^{(t)})\right)\\
&< \frac{1}{C_t^2}\left(\sum_{\substack{m\neq k\\ m\neq n}}\exp^2(\beta_{u_m,u_k}^{(t)}) - (\exp(\alpha_{u_k}^{(t)}) + \exp(\beta_{u_n,u_k}^{(t)}))\sum_{\substack{m\neq k\\ m\neq n}}\exp(\beta_{u_m,u_k}^{(t)})\right) \qquad (19)
\end{aligned}$$

We first consider at time $t = T_1$. Fix γ_k, we notice that $\beta_{u',u_k}^{(T_1)} \in \left[-\frac{\gamma_k\alpha_{K,\epsilon}^*}{K-1}, \frac{\gamma_k\alpha_{K,\epsilon}^*}{K-1}\right]$, for all $u' \in U, u' \neq u_k$. Then, we have:

$$\begin{aligned}
&\sum_{\substack{m\neq k\\ m\neq n}}\exp^2(\beta_{u_m,u_k}^{(T_1)}) - (\exp(\alpha_{u_k}^{(T_1)}) + \exp(\beta_{u_n,u_k}^{(T_1)}))\sum_{\substack{m\neq k\\ m\neq n}}\exp(\beta_{u_m,u_k}^{(T_1)})\\
&< (K-2)\exp^2(\frac{\gamma_k\alpha_{K,\epsilon}^*}{K-1}) - \left(\exp(\gamma_k^2\alpha_{K,\epsilon}^*) + \exp(-\frac{\gamma_k\alpha_{K,\epsilon}^*}{K-1})\right)(K-2)\exp(\frac{\gamma_k\alpha_{K,\epsilon}^*}{K-1})\\
&= (K-2)\exp(\frac{\gamma_k\alpha_{K,\epsilon}^*}{K-1})\left(\exp(\frac{\gamma_k\alpha_{K,\epsilon}^*}{K-1}) - \exp(\gamma_k^2\alpha_{K,\epsilon}^*) - \exp(-\frac{\gamma_k\alpha_{K,\epsilon}^*}{K-1})\right)\\
&< (K-2)\exp(\frac{\gamma_k\alpha_{K,\epsilon}^*}{K-1})\left(\exp(\frac{\gamma_k^2{\alpha_{K,\epsilon}^*}^2}{(K-1)^2}) - \exp(\gamma_k^2\alpha_{K,\epsilon}^*)\right) \qquad (20)
\end{aligned}$$

Recall that $\alpha_{K,\epsilon}^* < \log\frac{K}{\sqrt{\epsilon}}$. When $\epsilon > \left(\frac{K}{\exp((K-1)^2)}\right)^2$, we have:

$$\exp(\frac{\gamma_k^2{\alpha_{K,\epsilon}^*}^2}{(K-1)^2}) - \exp(\gamma_k^2\alpha_{K,\epsilon}^*) < 0.$$

By combining this and eq. (20), we have term $g_{T_1} < 0$, which indicates that $\delta_{u_n,u_k,\beta}^{(T_1)} < 0$. It is straightforward to show that $\delta_{u_k,\alpha}^{(T_1)} > 0$. Therefore, we have $\alpha_{u_k}^{(T_1+1)} > \alpha_{u_k}^{(T_1)}$ and $\beta_{u_m,u_k}^{T_1+1} < \beta_{u_m,u_k}^{T_1}$. By iteratively applying this argument, we can show that $\delta_{u_n,u_k,\beta}^{(T_1+t)} < 0$ for all $t > 0$, which completes the proof. □

We note that the above lemma requires $\epsilon > \left(\frac{K}{\exp((K-1)^2)}\right)^2$, which does hold for $K = 2$. However, for the two-feature case, it is straightforward to check $\delta_{u_n,u_k,\beta}^{(t)} < 0$ according to Lemma F.1.

Lemma F.3. *Consider the fine-tuning phase formulated in this work. Suppose* $\eta_t \leq \min_{n\neq k}\left\{\frac{1}{\exp(\beta_{u_n,u_k}^{(T_1)})}\right\}$ *and* $\beta_{u_n,u_k}^{(T_1)} < \beta_{u_{n'},u_k}^{(T_1)}$ *, then for all* $t \geq 0$*, we have:*

$$\beta_{u_n,u_k}^{(T_1+t)} < \beta_{u_{n'},u_k}^{(T_1+t)}$$

$$\delta_{u_n,k,\beta}^{(T_1+t)} \geq \delta_{u_{n'},k,\beta}^{(T_1+t)}.$$

Proof. We first prove $\delta_{n,k,\beta}^{(t)} \geq \delta_{n',k,\beta}^{(t)}$ when $t \geq T_1$ as follows. To simplify notations, we denote $D = \sum_{m \neq k} \mathbf{Attn}_{u_m}^{(t)\,2} - \mathbf{Attn}_{u_k}^{(t)}(1 - \mathbf{Attn}_{u_k}^{(t)})$. Consider:

$$\begin{aligned}
\delta_{u_n,u_k,\beta}^{(t)} - \delta_{u_{n'},u_k,\beta}^{(t)} &= \mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u_k\}} \mathbf{Attn}_{u_n}^{(t)} \left(D - \mathbf{Attn}_{u_n}^{(t)}\right)\right] - \mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u_k\}} \mathbf{Attn}_{u'_n}^{(t)} \left(D - \mathbf{Attn}_{u'_n}^{(t)}\right)\right] \\
&= \mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u_k\}} D \left(\mathbf{Attn}_{u_n}^{(t)} - \mathbf{Attn}_{u_{n'}}^{(t)}\right)\right] - \mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u_k\}} \left(\mathbf{Attn}_{u_n}^{(t)\,2} - \mathbf{Attn}_{u'_n}^{(t)\,2}\right)\right] \\
&= \mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u_k\}} \left(D - \mathbf{Attn}_{u_n}^{(t)} - \mathbf{Attn}_{u'_n}^{(t)}\right)\left(\mathbf{Attn}_{u_n}^{(t)} - \mathbf{Attn}_{u'_n}^{(t)}\right)\right] \\
&\overset{(i)}{<} 0,
\end{aligned}$$

where (i) follows from the same argument as Equation (18) in Lemma F.2. On the other hand, we have:

$$\begin{aligned}
\delta_{u_n,u_k,\beta}^{(t)} - \delta_{u_{n'},u_k,\beta}^{(t)} &= \mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u_k\}} \left(D - \mathbf{Attn}_{u_n}^{(t)} - \mathbf{Attn}_{u_{n'}}^{(t)}\right)\left(\mathbf{Attn}_{u_n}^{(t)} - \mathbf{Attn}_{u_{n'}}^{(t)}\right)\right] \\
&\overset{(i)}{\geq} \mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u_k\}} \left(\mathbf{Attn}_{u_n}^{(t)} - \mathbf{Attn}_{u'_n}^{(t)}\right)\right] \\
&\overset{(ii)}{\geq} \exp(\beta_{u_n,u_k}^{(t)}) - \exp(\beta_{u_{n'},u_k}^{(t)}) \qquad (21)
\end{aligned}$$

where (i) follows from $\mathbf{Attn}_{u_n}^{(t)} - \mathbf{Attn}_{u'_n}^{(t)} \leq 0$ and $D - \mathbf{Attn}_{u_n}^{(t)} - \mathbf{Attn}_{u'_n}^{(t)} \leq 1$, (ii) follows from the definition of the attention score.

Next, we prove $\beta_{u_n,u_k}^{(T_1+t)} < \beta_{u_{n'},u_k}^{(T_1+t)}$. We apply mathematical induction to complete the proof. We hypothesis that $\beta_{u_n,u_k}^{(T_1+t)} < \beta_{u_{n'},u_k}^{(T_1+t)}$ and prove it for time $t+1$. Consider:

$$\begin{aligned}
\beta_{u_{n'},u_k}^{(T_1+t+1)} - \beta_{u_n,u_k}^{(T_1+t+1)} &= \beta_{u_{n'},u_k}^{(T_1+t)} - \beta_{u_n,u_k}^{(T_1+t)} + \eta_t(\delta_{u_{n'},u_k,\beta}^{(T_1+t)} - \delta_{u_n,u_k,\beta}^{(T_1+t)}) \\
&\overset{(i)}{>} \beta_{u_{n'},u_k}^{(T_1+t)} - \beta_{u_n,u_k}^{(T_1+t)} + \eta_t(\exp(\beta_{u_n,u_k}^{(T_1+t)}) - \exp(\beta_{u_{n'},u_k}^{(T_1+t)})) \\
&\overset{(ii)}{=} \beta_{u_{n'},u_k}^{(T_1+t)} - \beta_{u_n,u_k}^{(T_1+t)} + \eta_t \exp(\xi)(\beta_{u_n,u_k}^{(T_1+t)} - \beta_{u_{n'},u_k}^{(T_1+t)}) \\
&\overset{(iii)}{>} (1 - \eta_t \exp(\beta_{u_{n'},u_k}^{(T_1)}))(\beta_{u_{n'},u_k}^{(T_1+t)} - \beta_{u_n,u_k}^{(T_1+t)}) \\
&\overset{(iv)}{\geq} 0
\end{aligned}$$

where (i) follows from eq. (21), (ii) follows from mean value theorem and $\xi \in [\beta_{u_n,u_k}^{(T_1+t)}, \beta_{u_{n'},u_k}^{(T_1+t)}]$, (iii) follows from Lemma F.2 and (iv) follows from $\eta_t \leq \frac{1}{\exp(\beta_{u_{n'},u_k}^{(T_1)})}$. □

We have investigated the dynamics of attention scores and attention weights during the finetuning phase. Now, we establish the convergence formally as follows. Different from Theorem 5.5, we give a more detailed condition on the stepsize and explain it later.

Theorem F.4. ***(Theorem 5.5)*** *Suppose Assumption 5.4 holds and the GD stepsize is constant η, where $\eta \leq \min_{n \neq k}\left\{\frac{1}{\exp(\beta_{u_n,u_k}^{(T_1)})}\right\}$. During fine-tuning, there exists $t_1 \leq t_2 \leq ... \leq t_K$, such that at each time $t_m = \mathcal{O}(poly(K, \frac{1}{\epsilon}, \frac{1}{\eta}))$, we have $L_{u_{k(m)}}^{(ft)}(Q^{(T_1+t_m)}) \leq \frac{\epsilon}{K}$. Specifically, let $\Delta_{k,\alpha}^{(t)}$ and $\Lambda_{k,n,\beta}^{(t)}$ denote the updates of α-type and β-type attention weights during the fine-tuning time t, respectively, such that:*

$$\begin{aligned}
\alpha_{u_k}^{(T_1+t)} &= \alpha_{u_k}^{(T_1)} + \Delta_{k,\alpha}^{(t)}, \\
\beta_{u_n,u_k}^{(T_1+t)} &= \beta_{u_n,u_k}^{(T_1)} - \Lambda_{n,k,\beta}^{(t)}, \quad \textit{for} \quad n \neq k.
\end{aligned}$$

Then, we have:

$$\Delta_{k(m),\alpha}^{(t_m)} = \Theta\left((1 - \gamma_{k(m)}^2 - \frac{\gamma_{k(m)}\gamma_\alpha}{K}) \log \frac{K}{\sqrt{\epsilon}}\right),$$

$$\Lambda^{(t_m)}_{n,k(m),\beta} = \Theta\left(\frac{\Delta^{(t_m)}_{k(m),\alpha}}{K} + \frac{\gamma_{k(m)}\gamma_\beta}{K}\log\frac{K}{\sqrt{\epsilon}}\right),$$

for some constant $\gamma_\alpha, \gamma_\beta \in [\gamma_{\min}, \gamma_{\max}]$. *More specifically, for all* $n \neq m$, *we have:*

$$\Delta^{(t_m)}_{m,\alpha} \leq \left(1-\frac{1}{K}\right)\left[\log\left(\frac{(K-1)(1-\sqrt{\epsilon})}{\sqrt{\epsilon}}\right) - \left(\gamma_m^2 + \frac{\gamma_m\gamma_{N(m)}}{K-1}\cdot\alpha^*_{K,\epsilon}\right)\right]$$

$$\Delta^{(t_m)}_{m,\alpha} \geq \left(1-\frac{1}{K}\right)\left[\log\left(\frac{(K-1)(1-\sqrt{\frac{2K\epsilon}{K+1}})}{\sqrt{\frac{2K\epsilon}{K+1}}}\right) - \left(\gamma_m^2 + \frac{\gamma_m\gamma_{M(m)}}{K-1}\cdot\alpha^*_{K,\epsilon}\right)\right]$$

$$\Lambda^{(t_m)}_{n,m,\beta} \leq \frac{(\gamma_{M(m)}\gamma_m - \gamma_{N(m)}\gamma_m)\alpha^*_{K,\epsilon}}{K-1} + \frac{\Delta^{(t_m)}_{m,\alpha}}{K-1},$$

$$\Lambda^{(t_m)}_{n,m,\beta} \geq \frac{(\gamma_{N(m)}\gamma_m - \gamma_{M(m)}\gamma_m)\alpha^*_{K,\epsilon}}{K-1} + \frac{\Delta^{(t_m)}_{m,\alpha}}{K-1},$$

where $N(m) = \arg\min_{n\neq m}\gamma_m\cdot\gamma_n$ *and* $M(m) = \arg\max_{n\neq m}\gamma_m\cdot\gamma_n$.

Proof. With loss of generalization, we assume $k(m) = m$. We first claim the existence of t_m s.t. $L^{(\text{ft})}_{u_m}(Q^{(T_1+t_m)}) \leq \frac{\epsilon}{K}$. According to Theorem E.2, we have:

$$\alpha^{(T_1)}_{u_m} = \gamma_m^2\alpha^*_{K,\epsilon},$$
$$\beta^{(T_1)}_{u_n,u_m} = \frac{-\gamma_n\gamma_m\alpha^*_{K,\epsilon}}{K-1}. \tag{22}$$

By following from Lemmas D.4 and F.3, we have:

$$\Lambda^{(t)}_{N(m),m,\beta} \geq \frac{\Delta^{(t)}_{m,\alpha}}{K-1},$$
$$\Lambda^{(t)}_{M(m),m,\beta} \leq \frac{\Delta^{(t)}_{m,\alpha}}{K-1}. \tag{23}$$

Then, we consider:

$$\begin{aligned}
L^{(\text{ft})}_{u_m}(Q^{(T_1+t_m)}) &= \frac{(\sum_{n\neq m}\exp(\beta^{(T_1+t_m)}_{u_n,u_m}))^2 + \sum_{n\neq m}\exp(\beta^{(T_1+t_m)}_{u_n,u_m})^2}{2K(\exp(\alpha^{(T_1+t_m)}_{u_m}) + \sum_{n\neq m}\exp(\beta^{(T_1+t_m)}_{u_n,u_m}))^2} \\
&\overset{(i)}{\leq} \frac{((K-1)\exp(\beta^{(T_1+t_m)}_{u_{N(m)},u_m}))^2 + (K-1)\exp(\beta^{(T_1+t_m)}_{u_{N(m)},u_m})^2}{2K(\exp(\alpha^{(T_1+t_m)}_{u_m}) + (K-1)\exp(\beta^{(T_1+t_m)}_{u_{N(m)},u_m}))^2} \\
&= \frac{((K-1)\exp(\beta^{(T_1)}_{u_{N(m)},u_m} - \Lambda^{(t_m)}_{N(m),m,\beta}))^2 + (K-1)\exp(\beta^{(T_1)}_{u_{N(m)},u_m} - \Lambda^{(t_m)}_{N(m),m,\beta})^2}{2K(\exp(\alpha^{(T_1)}_{u_m} + \Delta^{(t_m)}_{m,\alpha}) + (K-1)\exp(\beta^{(T_1)}_{u_{N(m)},u_m} - \Lambda^{(t_m)}_{N(m),m,\beta}))^2} \\
&\overset{(ii)}{\leq} \frac{((K-1)\exp(\beta^{(T_1)}_{u_{N(m)},u_m} - \frac{\Delta^{(t)}_{m,\alpha}}{K-1}))^2 + (K-1)\exp(\beta^{(T_1)}_{u_{N(m)},u_m} - \frac{\Delta^{(t)}_{m,\alpha}}{K-1})^2}{2K(\exp(\alpha^{(T_1)}_{u_m} + \Delta^{(t_m)}_{m,\alpha}) + (K-1)\exp(\beta^{(T_1)}_{u_{N(m)},u_m} - \frac{\Delta^{(t_m)}_{m,\alpha}}{K-1}))^2}
\end{aligned} \tag{24}$$

where (i) follows from Lemma D.11, (ii) follows from Lemma D.11 and eq. (23). Let the above equation less than $\frac{\epsilon}{K}$, we derive an upper bound as follows.

$$\Delta^{(t_m)}_{m,\alpha} \leq \left(1-\frac{1}{K}\right)\left[\log\left(\frac{(K-1)(1-\sqrt{\epsilon})}{\sqrt{\epsilon}}\right) - \left(\gamma_m^2 + \frac{\gamma_m\gamma_{N(m)}}{K-1}\cdot\alpha^*_{K,\epsilon}\right)\right]. \tag{25}$$

On the other hand, we derive the lower bound on the α-type attention weights as follows.

$$L^{(\text{ft})}_{u_m}(Q^{(T_1+t_m)}) = \frac{(\sum_{n\neq m}\exp(\beta^{(T_1+t_m)}_{u_n,u_m}))^2 + \sum_{n\neq m}\exp(\beta^{(T_1+t_m)}_{u_n,u_m})^2}{2K(\exp(\alpha^{(T_1+t_m)}_{u_m}) + \sum_{n\neq m}\exp(\beta^{(T_1+t_m)}_{u_n,u_m}))^2}$$

$$
\begin{aligned}
&\overset{(i)}{\geq} \frac{((K-1)\exp(\beta^{(T_1+t_m)}_{u_{M(m)},u_m}))^2 + (K-1)\exp(\beta^{(T_1+t_m)}_{u_{M(m)},u_m})^2}{2K(\exp(\alpha^{(T_1+t_m)}_{u_m}) + (K-1)\exp(\beta^{(T_1+t_m)}_{u_{M(m)},u_m}))^2} \\
&= \frac{((K-1)\exp(\beta^{(T_1)}_{u_{M(m)},u_m} - \Lambda^{(t_m)}_{M(m),m,\beta}))^2 + (K-1)\exp(\beta^{(T_1)}_{u_{M(m)},u_m} - \Lambda^{(t_m)}_{M(m),m,\beta})^2}{2K(\exp(\alpha^{(T_1)}_{u_m} + \Delta^{(t_m)}_{m,\alpha}) + (K-1)\exp(\beta^{(T_1)}_{u_{M(m)},u_m} - \Lambda^{(t_m)}_{M(m),m,\beta}))^2} \\
&\overset{(ii)}{\geq} \frac{((K-1)\exp(\beta^{(T_1)}_{u_{M(m)},u_m} - \frac{\Delta^{(t)}_{m,\alpha}}{K-1}))^2 + (K-1)\exp(\beta^{(T_1)}_{u_{M(m)},u_m} - \frac{\Delta^{(t)}_{m,\alpha}}{K-1})^2}{2K(\exp(\alpha^{(T_1)}_{u_m} + \Delta^{(t_m)}_{m,\alpha}) + (K-1)\exp(\beta^{(T_1)}_{u_{M(m)},u_m} - \frac{\Delta^{(t_m)}_{m,\alpha}}{K-1}))^2}
\end{aligned}
$$

where (i) follows from Lemma D.11, (ii) follows from Lemma D.11 and eq. (23). Let the above equation larger than $\frac{\epsilon}{K}$. We derive a lower bound as follows.

$$
\Delta^{(t_m)}_{m,\alpha} \geq \left(1 - \frac{1}{K}\right)\left[\log\left(\frac{(K-1)(1-\sqrt{\frac{2K\epsilon}{K+1}})}{\sqrt{\frac{2K\epsilon}{K+1}}}\right) - \left(\gamma_m^2 + \frac{\gamma_m\gamma_{M(m)}}{K-1}\cdot\alpha^*_{K,\epsilon}\right)\right]. \tag{26}
$$

Then, we focus on β-type attention weights. According to Lemma F.3, for any $n \neq m$, we have

$$
\begin{aligned}
\Lambda^{(t)}_{M(m),m,\beta} \leq &\Lambda^{(t)}_{n,m,\beta} \leq \Lambda^{(t)}_{N(m),m,\beta}, \\
\beta^{(T_1+t)}_{u_{M(m)},u_m,\beta} \geq &\beta^{(T_1+t)}_{u_n,u_m,\beta} \geq \beta^{(T_1+t)}_{u_{N(m)},u_m,\beta},
\end{aligned}
$$

where the second inequality is equivalent to

$$
\beta^{(T_1)}_{u_{M(m)},u_m,\beta} - \Lambda^{(t)}_{M(m),m,\beta} \geq \beta^{(T_1+t)}_{u_n,u_m,\beta} - \Lambda^{(t)}_{n,m,\beta} \geq \beta^{(T_1+t)}_{u_{N(m)},u_m,\beta} - \Lambda^{(t)}_{N(m),m,\beta}.
$$

By combining this, eqs. (22) and (23), we have:

$$
\begin{aligned}
\Lambda^{(t)}_{N(m),m,\beta} &\leq \frac{(\gamma_{M(m)}\gamma_m - \gamma_{N(m)}\gamma_m)\alpha^*_{K,\epsilon}}{K-1} + \frac{\Delta^{(t)}_{m,\alpha}}{K-1}, \\
\Lambda^{(t)}_{M(m),m,\beta} &\geq \frac{(\gamma_{N(m)}\gamma_m - \gamma_{M(m)}\gamma_m)\alpha^*_{K,\epsilon}}{K-1} + \frac{\Delta^{(t)}_{m,\alpha}}{K-1}.
\end{aligned}
$$

To prove the existence of t_m, we first recall that for any $u \in U$, we have $\Delta^{(t_m)}_{u,\alpha} = \sum_{t=0}^{t_m}\eta_t\delta^{(t)}_{u,\alpha}$. By combining Lemma D.3 and the fact that $\mathbf{Attn}^{(t)}_u$ is monotonically increasing, for $t \in [T_1, T_1+t_m]$, we have

$$
\begin{aligned}
\delta^{(t)}_{u,\alpha} &\geq \left[\mathbf{1}_{\{x_{\text{query}}=u\}}\mathbf{Attn}^{(T_1)}_u\right]\mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u\}}\left(\sum_{w\in U, w\neq u}\mathbf{Attn}^{(t)}_w{}^2 + (1-\mathbf{Attn}^{(t)}_u)^2\right)\right] \\
&\overset{(i)}{\geq} \left[\mathbf{1}_{\{x_{\text{query}}=u\}}\mathbf{Attn}^{(T_1)}_u\right]\frac{\epsilon}{K}
\end{aligned}
$$

where (i) follows from Lemma D.8 and $t \geq T_1$. Consider:

$$
\begin{aligned}
\Delta^{(t_m)}_{m,\alpha} &= \sum_{t=0}^{t_m}\eta_t\delta^{(t)}_{u_m,\alpha} \\
&\geq \left[\mathbf{1}_{\{x_{\text{query}}=u_m\}}\mathbf{Attn}^{(T_1)}_{u_m}\right]\frac{\epsilon}{K}\cdot t_m\eta.
\end{aligned}
$$

Therefore, there exist $t_m = \mathcal{O}(\text{poly}(K, \frac{1}{\epsilon}, \frac{1}{\eta}))$ such that

$$
\Delta^{(t_m)}_{m,\alpha} \geq \left(1 - \frac{1}{K}\right)\left[\log\left(\frac{(K-1)(1-\sqrt{\epsilon})}{\sqrt{\epsilon}}\right) - \left(\gamma_m^2 + \frac{\gamma_m\gamma_{N(m)}}{K-1}\cdot\alpha^*_{K,\epsilon}\right)\right].
$$

By combining the above inequality and eq. (24), we can conclude that $L^{(\text{ft})}_{u_m}(Q^{(T_1+t_m)}) \leq \frac{\epsilon}{K}$. □

We note that T_1 denotes the end of the training phase. Here, the step size is constrained by β-type attention weights at time T_1, which can be derived from Theorem E.2. Under Assumption 5.4, this constraint has the order of $\mathcal{O}(\exp(\frac{\log(K/\sqrt{\epsilon})}{K}))$, which converges to 1 as K grows. Therefore, the allowable stepsize is asymptotically of constant order, as presented in Theorem 5.5 in the main body.

G Proof of Theorem 5.6

After the fine-tuning phase, we evaluate the fine-tuned model on the original training set, as presented in Theorem 5.6. We restate the theorem and prove it as follows.

Theorem G.1. ***(Theorem 5.6)*** *Consider the problem formulated in this work. Suppose Assumption 5.4 holds and $\Delta_{k,\alpha}^{(t)}, \Lambda_{n,k,\beta}^{(t)}$ are defined in Theorem F.4. After the fine-tuning phase, the performance of the original feature set can be captured by*

$$L_{v_k}^{(tr)}(Q^{(T_1+T_2)}) = \frac{(\sum_{n\neq k}\exp(\beta_{v_n,v_k}^{(T_1+T_2)}))^2 + \sum_{n\neq k}\exp(\beta_{v_n,v_k}^{(T_1+T_2)})^2}{2K(\exp(\alpha_{v_k}^{(T_1+T_2)}) + \sum_{n\neq k}\exp(\beta_{v_n,v_k}^{(T_1+T_2)}))^2},$$

where

$$\alpha_{v_k}^{(T_1+T_2)} = \alpha_{v_k}^{(T_1)} + \gamma_k^2\Delta_{k,\alpha}^{(T_2)},$$
$$\beta_{v_n,v_k}^{(T_1+T_2)} = \beta_{v_n,v_k}^{(T_1)} - \gamma_k\gamma_n\Lambda_{n,k,\beta}^{(T_2)}, \quad \textit{for} \quad n \neq k.$$

Proof. By following from Lemma D.8 and eq. (10), it is straightforward to prove the expression of $L_{v_k}^{(\text{tr})}(Q^{(T_1+T_2)})$. Then, we have:

$$\begin{aligned}
\alpha_{v_k}^{(T_1+T_2)} &= \alpha_{v_k}^{(T_1)} - \sum_{t=T_1}^{T_2}\eta_t \cdot v_k^\top \nabla L^{(\text{ft})}(Q^{(t)})v_{k'} \\
&\overset{(i)}{=} \alpha_{v_k}^{(T_1)} - \sum_{t=T_1}^{T_2}\eta_t \cdot \gamma_k^2 u_k^\top \nabla L^{(\text{ft})}(Q^{(t)})u_{k'} \\
&= \alpha_{v_k}^{(T_1)} + \gamma_k^2\Delta_{k,\alpha}^{(T_2)}.
\end{aligned}$$

where (i) follows from Lemma D.10. Similarly, we have:

$$\begin{aligned}
\beta_{v_n,v_k}^{(T_1+T_2)} &= \beta_{v_n,v_k}^{(T_1)} - \sum_{t=T_1}^{T_2}\eta_t \cdot v_n^\top \nabla L^{(\text{ft})}(Q^{(t)})v_k \\
&= \beta_{v_n,v_k}^{(T_1)} - \sum_{t=T_1}^{T_2}\eta_t \cdot \gamma_k\gamma_n u_n^\top \nabla L^{(\text{ft})}(Q^{(t)})u_k \\
&= \beta_{v_n,v_k}^{(T_1)} - \gamma_k\gamma_n\Lambda_{n,k,\beta}^{(T_2)}.
\end{aligned}$$

□

H Case Study

Two-features Case: Consider a two-feature case (i.e., $K = 2$) with correlation coefficients γ_1 and γ_2. We find that features with larger absolute correlation magnitudes converge earlier. After fine-tuning, the performance on the training features V is characterized as follows.

Theorem H.1. *Suppose $K = 2$, $|\gamma_1| > |\gamma_2|$, and Assumption 5.4 holds. After fine-tuning, the model performance on the original feature set can be captured by*

$$L_{v_1}^{(tr)}(Q^{(T_1+T_2)}) \in \left[\frac{1}{2}\left(\frac{1}{1+g_\epsilon^{c_1}}\right)^2, \frac{1}{2}\left(\frac{1}{1+g_\epsilon^{d_1}}\right)^2\right],$$
$$L_{v_2}^{(tr)}(Q^{(T_1+T_2)}) = \frac{1}{2}\left(\frac{1}{1+g_\epsilon^{c_2}}\right)^2,$$

for some constant c_1, c_2, d_1 depending on γ_1, γ_2, as presented in Theorem H.5 in the appendix.

Theorem H.1 characterizes the model performance in the training set after fine-tuning, based on which we can determine whether the forgetting is positive or negative.

Corollary H.2. *Suppose $K = 2$, $|\gamma_1| > |\gamma_2|$ and Assumption 5.4 holds. After finetuning, the following statements are true: (1) If $c_1 < 1$, we have $F_{v_1} > 0$; if $d_1 > 1$, we have $F_{v_1} < 0$. (2) $F_{v_2} > 0$ if and only if $\gamma_2^2 + \gamma_1\gamma_2 < 0$. (3) If $c_2 < \log_{g_\epsilon}\left(\frac{1}{\sqrt{2\epsilon - 1/(1+g_\epsilon^{c_1})^2}}\right)$, we have $F < 0$. (4) If $c_2 > \log_{g_\epsilon}\left(\frac{1}{\sqrt{2\epsilon - 1/(1+g_\epsilon^{d_1})^2}}\right)$, we have $F > 0$.*

The first two items construct the condition on which the individual forgetting will be positive or negative, reflecting the performance improvement or degradation. The last two items focus on the overall forgetting and identify a region where the overall forgetting is positive or negative. We note that the region is not empty, which can be easily checked by setting $\gamma_1 = 1$. A numerical simulation on the overall performance is provided in fig. 2b later in the experiments.

In Lemma 5.3, it is shown that the fine-tuning process can be interpreted as simultaneously increasing the confidence in identifying the correct feature and reducing the interference caused by incorrect features, i.e., the α-type attention increases while the β-type attention decreases. However, during fine-tuning, we observe that reducing the interference from incorrect features in the OOD data may inadvertently increase the interference induced by incorrect features in the original training set. Suppose the query token is v_k. This phenomenon arises when $\gamma_k\gamma_n < 0$. Although the α-type attention for the training features continues to increase during fine-tuning, as shown in Theorem 5.6, its rate of increase can be slower than that of the β-type attention. In other words, the growth in confidence for identifying the correct feature can be slower than the growth in interference from incorrect features, ultimately leading to the deterioration of the fine-tuned transformer on the training domain.

K-features Case: Directly characterizing the total loss for the general case is very challenging due to its complex expression as shown in Theorem 5.6. Therefore, we focus on individual feature performance and characterize the region in which the fine-tuned transformer is improved or degraded.

Theorem H.3. *Given an index $k \in [K]$, denote $m_0 = \arg\max_{m\neq k} \gamma_k\gamma_m$ and $m_1 = \arg\min_{m\neq k} \gamma_k\gamma_m$. The following statements hold: (1) If $-\gamma_k\gamma_{m_0} > \gamma_k^2$, $L_{v_k}^{(tr)}(Q^{(T_1+t)})$ is monotonically increasing with time t; (2) If $-\gamma_k\gamma_{m_1} < \gamma_k^2$, $L_{v_k}^{(tr)}(Q^{(T_1+t)})$ is monotonically decreasing.*

Under certain conditions on the distribution shift, we find that the loss w.r.t. a particular training feature can either monotonically increase or decrease. The regions are constructed based on our theoretical observations in the two-feature case. For example, consider the case with the query token is u_k, where γ_k is slightly positive, and the remaining γ_n for all $n \neq k$ are comparatively more negative. In this scenario, the interference from the incorrect features v_n $(n \neq k)$ exerts a more substantial influence on the fine-tuned transformer than the confidence associated with recognizing the correct feature v_k. This results in a net negative impact of fine-tuning on the training domain, leading to positive forgetting. We note that investigating the performance of individual features is also important because multi-layer transformers propagate and accumulate per-feature errors across layers, resulting in an amplified overall error.

H.1 Two-feature Case

For the two-feature case, we first prove that $L_{u_1}^{(\text{ft})}(Q^{(t)})$ converges to $\frac{\epsilon}{K}$ first and $L_{u_2}^{(\text{ft})}(Q^{(t)})$ converges to $\frac{\epsilon}{K}$ later.

Lemma H.4. *Suppose $K = 2$, $|\gamma_1| > |\gamma_2|$ and Assumption 5.4 holds. Set $\eta_t \leq \frac{1}{\alpha_{\max}}$ where $\alpha_{\max}$ denotes the upper bound on α-type attention defined in Theorem F.4. Then, for all $t \geq T_1$, we have:*

$$L_{u_1}^{(ft)}(Q^{(t)}) < L_{u_2}^{(ft)}(Q^{(t)}).$$

Proof. When $K = 2$, we have $\sum_{u'\in U, u'\neq u} \mathbf{Attn}_{u'}^{(t)^2} = \left(1 - \mathbf{Attn}_u^{(t)}\right)^2$. Then, we rewrite Lemma D.8 as follows.

$$L_u^{(\text{ft})}(Q^{(t)}) = \frac{1}{2}\mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u\}}\left(\sum_{u'\in U, u'\neq u} \mathbf{Attn}_{u'}^{(t)^2} + \left(1 - \mathbf{Attn}_u^{(t)}\right)^2\right)\right]$$

$$= \mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u\}}\left(1-\mathbf{Attn}_u^{(t)}\right)^2\right]$$

Equivalently, we only need to prove $\alpha_{u_1}^{(t)} > \alpha_{u_2}^{(t)}$ for all $t \geq T_1$ according to the above derivation and the definition of attention score. Since $|\gamma_1| > |\gamma_2|$, we have $\alpha_{u_1}^{(T_1)} > \alpha_{u_2}^{(T_1)}$ by following from Theorem E.2. Then, we complete the proof by mathematical induction as follows. We hypothesis $\alpha_{u_1}^{(t)} > \alpha_{u_2}^{(t)}$ holds and prove it at $t+1$. We first consider:

$$\begin{aligned}
\delta_{u_1,\alpha}^{(t)} - \delta_{u_2,\alpha}^{(t)} &\overset{(i)}{>} \mathbf{Attn}_{u_2}^{(t)}\left(\mathbf{Attn}_{u_2}^{(t)^2} + (1-\mathbf{Attn}_{u_1}^{(t)})^2 - \mathbf{Attn}_{u_1}^{(t)^2} - (1-\mathbf{Attn}_{u_2}^{(t)})^2\right) \\
&= 2\mathbf{Attn}_{u_2}^{(t)}(\mathbf{Attn}_{u_2}^{(t)} - \mathbf{Attn}_{u_1}^{(t)}) \\
&> 2(\exp(\alpha_{u_2}^{(t)}) - \exp(\alpha_{u_1}^{(t)})) \qquad (27)
\end{aligned}$$

where (i) follows from Lemma D.3. Then, consider:

$$\begin{aligned}
\alpha_{u_1}^{(t+1)} - \alpha_{u_2}^{(t+1)} &= \alpha_{u_1}^{(t)} - \alpha_{u_2}^{(t)} + \eta_t(\delta_{u_1,\alpha}^{(t)} - \delta_{u_2,\alpha}^{(t)}) \\
&\overset{(i)}{>} \alpha_{u_1}^{(t)} - \alpha_{u_2}^{(t)} + \eta_t(\exp(\alpha_{u_2}^{(t)}) - \exp(\alpha_{u_1}^{(t)})) \\
&\overset{(ii)}{=} \alpha_{u_1}^{(t)} - \alpha_{u_2}^{(t)} + \eta_t \exp(\xi)(\alpha_{u_2}^{(t)} - \alpha_{u_1}^{(t)}) \\
&\overset{(iii)}{>} (1-\eta_t\exp(\alpha_{\max}))(\alpha_{u_1}^{(T_1)} - \alpha_{u_2}^{(T_1)}) \\
&\overset{(iv)}{\geq} 0
\end{aligned}$$

where (i) follows from eq. (27), (ii) follows from mean value theorem where $\xi \in [\alpha_{u_2}^{(t)}, \alpha_{u_1}^{(t)}]$, (iii) follows from Lemma F.2 and (iv) follows from $\eta_t \leq \frac{1}{\alpha_{\max}}$. □

Now, we prove Theorem H.1, in which we characterize the forgetting of the fine-tuned transformer model.

Theorem H.5. ***(Theorem H.1)*** *Suppose $K = 2$, $|\gamma_1| > |\gamma_2|$ and Assumption 5.4 holds. After fine-tuning, the performance of the original feature set can be captured by*

$$L_{v_1}^{(tr)}(Q^{(T_1+T_2)}) \in \left[\frac{1}{2}\left(\frac{1}{1+g_\epsilon^{c_1}}\right)^2, \frac{1}{2}\left(\frac{1}{1+g_\epsilon^{d_1}}\right)^2\right]$$

$$L_{v_2}^{(tr)}(Q^{(T_1+T_2)}) = \frac{1}{2}\left(\frac{1}{1+g_\epsilon^{c_2}}\right)^2$$

where $g_\epsilon = \frac{1-\sqrt{\epsilon}}{\sqrt{\epsilon}}$ and

$$\begin{aligned}
c_1 &= 1 + \frac{1}{2}(\gamma_1^2 + \gamma_1\gamma_2)(1 - \frac{\gamma_2^2 + \gamma_1\gamma_2}{2}) \\
d_1 &= 1 + \frac{1}{2}(\gamma_1^2 + \gamma_1\gamma_2)(1 - \frac{\gamma_1^2 - \gamma_1\gamma_2}{2}) \\
c_2 &= 1 + \frac{1}{2}(\gamma_2^2 + \gamma_1\gamma_2)(1 - \frac{\gamma_2^2 + \gamma_1\gamma_2}{2})
\end{aligned}$$

Proof. As we discussed at the beginning of this subsection, $L_{u_1}^{(\text{ft})}(Q^{(t)})$ converges to $\frac{\epsilon}{2}$ first. Ignoring the minor error caused by the last gradient descent step, we assume $L_{u_1}^{(\text{ft})}(Q^{(T_1+t_1)}) = \frac{\epsilon}{2}$, where

$$\begin{aligned}
L_{u_1}^{(\text{ft})}(Q^{(T_1+t_1)}) &= \frac{1}{2}\left(\frac{\exp(\beta_{u_2,u_1}^{(T_1+t_1)})}{\exp(\alpha_{u_1}^{(T_1+t_1)}) + \exp(\beta_{u_2,u_1}^{(T_1+t_1)})}\right)^2 \\
&\overset{(i)}{=} \frac{1}{2}\left(\frac{\exp(\beta_{u_2,u_1}^{(T_1)} - \Delta_{1,\alpha}^{(t_1)})}{\exp(\alpha_{u_1}^{(T_1)} + \Delta_{1,\alpha}^{(t_1)}) + \exp(\beta_{u_2,u_1}^{(T_1)} - \Delta_{1,\alpha}^{(t_1)})}\right)^2
\end{aligned}$$

where (i) follows from Lemma F.1. Then, we have:

$$\Delta_{1,\alpha}^{(t_1)} = \frac{1}{2}\left(1 - \frac{\gamma_1^2 - \gamma_1\gamma_2}{2}\right)\log\left(\frac{1-\sqrt{\epsilon}}{\sqrt{\epsilon}}\right).$$

After this, $L_{u_1}^{(\text{ft})}(Q^{(t)})$ converges to $\frac{\epsilon}{2}$ at time $T_1 + t_2$. Similarly, we have:

$$\begin{aligned} L_{u_2}^{(\text{ft})}(Q^{(T_1+t_2)}) &= \frac{1}{2}\left(\frac{\exp(\beta_{u_1,u_2}^{(T_1+t_2)})}{\exp(\alpha_{u_2}^{(T_1+t_2)}) + \exp(\beta_{u_1,u_2}^{(T_1+t_2)})}\right)^2 \\ &\overset{(i)}{=} \frac{1}{2}\left(\frac{\exp(\beta_{u_1,u_2}^{(T_1)} - \Delta_{2,\alpha}^{(t_2)})}{\exp(\alpha_{u_2}^{(T_1)} + \Delta_{2,\alpha}^{(t_2)}) + \exp(\beta_{u_1,u_2}^{(T_1)} - \Delta_{2,\alpha}^{(t_2)})}\right)^2. \end{aligned}$$

When $L_{u_2}^{(\text{ft})}(Q^{(T_1+t_2)})$ achieves $\frac{\epsilon}{2}$, we have:

$$\Delta_{2,\alpha}^{(t_2)} = \frac{1}{2}\left(1 - \frac{\gamma_1^2 + \gamma_1\gamma_2}{2}\right)\log\left(\frac{1-\sqrt{\epsilon}}{\sqrt{\epsilon}}\right).$$

According to Lemma F.3 and Lemma F.1, we have:

$$\Delta_{1,\alpha}^{(t_1)} < \Delta_{1,\alpha}^{(t_2)} < \Delta_{2,\alpha}^{(t_2)}.$$

At last, we follow Theorem G.1 to characterize the forgetting performance and complete the proof. □

Recall that $L_{u_k}^{(\text{tr})}(Q^{(T_1)}) = \frac{\epsilon}{2}$. The proof of corollary H.2 follows from comparing $L_{u_k}^{(\text{ft})}(Q^{(T_1+t_2)})$ and $\frac{\epsilon}{2}$ directly.

H.2 K-features Case

As shown in Theorems 5.6 and F.4, the forgetting is highly dependent on the relation coefficients between the training and fine-tuning feature sets. In this subsection, we focus on individual performance and characterize the region in which the fine-tuned transformer is improved or degraded.

Theorem H.6. ***(Theorem H.3)*** *Fix an index $k \in [K]$ and denote $m_0 = \arg\max_{m\neq k} \gamma_k\gamma_m$ and $m_1 = \arg\min_{m\neq k} \gamma_k\gamma_m$, then the following statements hold.*

- *If $-\gamma_k\gamma_{m_0} > \gamma_k^2$, we have $L_{v_k}^{(tr)}(Q^{(T_1+t)})$ is monotonically increasing with time t.*
- *If $-\gamma_k\gamma_{m_1} < \gamma_k^2$, we have $L_{v_k}^{(tr)}(Q^{(T_1+t)})$ is monotonically decreasing with time t.*

Proof. We prove the first item as follows. We derive the loss on the training feature v_k during the fine-tuning phase as follows.

$$\begin{aligned} L_{v_k}^{(\text{tr})}(Q^{(T_1+t+1)}) &= \frac{(\sum_{n\neq k}\exp(\beta_{v_n,v_k}^{(T_1+t+1)}))^2 + \sum_{n\neq k}\exp(\beta_{v_n,v_k}^{(T_1+t+1)})^2}{2K(\exp(\alpha_{u_k}^{(T_1+t+1)}) + \sum_{n\neq k}\exp(\beta_{v_n,v_k}^{(T_1+t+1)}))^2} \\ &= \frac{(\sum_{n\neq k}\exp(\beta_{v_n,v_k}^{(T_1+t)} + \gamma_k\gamma_n\delta_{u_n,u_k,\beta}^{(T_1+t)}))^2 + \sum_{n\neq k}\exp(\beta_{v_n,v_k}^{(T_1+t)} + \gamma_k\gamma_n\delta_{u_n,u_k,\beta}^{(T_1+t)})^2}{2K(\exp(\alpha_{u_k}^{(T_1+t)} + \gamma_k^2\delta_{u_k,\alpha}^{(T_1+t)}) + \sum_{n\neq k}\exp(\beta_{v_n,v_k}^{(T_1+t)} + \gamma_k\gamma_n\delta_{u_n,u_k,\beta}^{(T_1+t)}))^2} \\ &\overset{(i)}{\geq} \frac{(\sum_{n\neq k}\exp(\beta_{v_n,v_k}^{(T_1+t)}))^2 + \sum_{n\neq k}\exp(\beta_{v_n,v_k}^{(T_1+t)})^2}{2K(\exp(\alpha_{u_k}^{(T_1+t)}) + \sum_{n\neq k}\exp(\beta_{v_n,v_k}^{(T_1+t)}))^2} \end{aligned}$$

where (i) follows from the fact that $-\gamma_k\gamma_n\delta_{u_n,u_k,\beta}^{(T_1+t)} < \gamma_k^2\delta_{u_k,\alpha}$. The second item can be prove by the same argument. □

I Extension

In this section, we will relax our assumption about the prompt construction. For simplicity, our proofs have assumed that each prompt consists of the same number of features from its corresponding feature set. We next show that the forgetting behavior remains nearly unchanged if each training token is uniformly drawn from the feature set as long as the prompt length N is large enough.

Specifically, given a prompt P, we denote N_u as the cardinality of $S_u := \{x_i \in P : x_i = u\}$. Then, $(N_{u_1}, N_{u_2}, ..., N_{u_K})$ follows the multinomial distribution $(N, \mathbf{p})$ where $\mathbf{p} = (\frac{1}{K}, \frac{1}{K}, \ldots, \frac{1}{K})$. We also note Lemmas D.1 to D.8 and D.10 still hold. In the following lemma, We present a high-probability event.

Lemma I.1. *For any $\epsilon_0 \in (0,1)$ and $j \in \{1,2\}$, suppose $N \geq \frac{20K}{\epsilon_0^2}$. Denote*

$$\mathcal{E}_0 = \left\{ P^{(j)} : \frac{N_k^{(j)}}{N} \in \left[\frac{1}{K} - \epsilon_0, \frac{1}{K} + \epsilon_0 \right], \textit{for all } k \in [K] \right\}.$$

Then, we have

$$\mathbb{P}(P^{(j)} \in \mathcal{E}_0) \geq 1 - 3\exp\left(-\frac{N\epsilon_0^2}{25}\right).$$

Proof. Suppose $N \geq \frac{20K}{\epsilon_0^2}$. We first present the tail bound of the multinomial distribution:

$$\mathbb{P}\left(\sum_{k\in[K]} \left| \frac{N_k^{(j)}}{N} - \frac{1}{K} \right| > \epsilon_0 \right) \leq 3\exp\left(-\frac{N\epsilon_0^2}{25}\right).$$

Therefore, we have:

$$\begin{aligned}
\mathbb{P}(P^{(j)} \notin \mathcal{E}_0) &= \mathbb{P}\left(\bigcup_{k\in[K]} \left\{ \left| \frac{N_k^{(j)}}{N} - \frac{1}{K} \right| > \epsilon_0 \right\} \right) \\
&\leq \mathbb{P}\left(\sum_{k\in[K]} \left| \frac{N_k^{(j)}}{N} - \frac{1}{K} \right| > \epsilon_0 \right) \\
&\leq 3\exp\left(-\frac{N\epsilon_0^2}{25}\right).
\end{aligned}$$

Consequently,

$$\mathbb{P}(P^{(j)} \in \mathcal{E}_0) \geq 1 - 3\exp\left(-\frac{N\epsilon_0^2}{25}\right).$$

□

The above lemma shows that if the prompt length is large, the number of occurrences of each feature in the prompt is nearly the same with high probability. Under this high-probability event, the behavior of the training remains almost unchanged. The probability of deviations from this event is exponentially small and therefore has a negligible impact. Below, we briefly explain why the characterization of the training and the fine-tuning phases, as well as the proofs of the main theorems, remain valid.

Consider the the training phase with uniformly drawn tokens. We argue that the main steps of the proof either remain valid or hold with high probability.

1. Conditioned on the high-probability event, we have $\delta_{k,\alpha}^{(t)} \geq \text{poly}(\frac{1}{K}, \epsilon)$. Therefore, there exists $T_1 = \text{poly}(\frac{1}{\epsilon}, K, \frac{1}{\eta})$ such that $\alpha_{v_k}^{(T_1)} = \alpha_{K,\epsilon}^{(*)}$. Also, $\beta_{v_n,v_k}^{(T_1^*)} = -\frac{\alpha_{K,\epsilon}^{(*)}}{K-1}$ still holds according to Lemma D.6.
2. The value of the loss function conditioned on the high-probability event is $\mathcal{O}(\frac{\epsilon}{K})$, while its value conditioned on the remaining low-probability event is exponentially small. Therefore, we still have $L_{v_k}^{(\text{tr})}(Q^{(T_1)}) = \mathcal{O}(\frac{\epsilon}{K})$.

Consider the fine-tuning phase with uniformly drawn tokens. We argue that the main steps in Appendix F either remain valid or hold with high probability.

1. The initialization of the fine-tuning phase follows from the same idea as Equation (22).
2. Let $N \geq \mathcal{O}(K^4)$. Conditioned on the high-probability event, we have $\delta^{(t)}_{u_n,u_k,\beta} = -\mathcal{O}(\frac{1}{K^3})$, while its value condition on the remaining low-probability event satisfies $\mathcal{O}(\exp(-K))$. Therefore, $\delta^{(t)}_{u_n,u_k,\beta} \leq 0$ still holds.
3. Following from the same argument as the convergence of the training phase, we have that there exists $T_2 = \text{poly}(K, \frac{1}{\epsilon}, \gamma)$ s.t. $L^{(\text{ft})}_{u_k}(Q^{(T_2)}) \leq \mathcal{O}(\frac{\epsilon}{K})$ for all $k \in [K]$.
4. Lemma F.1 still hold since their proofs do not depend on the assumption on token distribution.
5. Theorem F.4 still holds because if $N \geq \mathcal{O}(K^4)$, the bound on the forgetting will have an additional negligible error of $\mathcal{O}(\frac{\epsilon}{K^2})$ in addition to the dominating term of the order $\mathcal{O}(\frac{\epsilon}{K})$ conditioned on the high-probability event.
   Taking summations and expectations on both sides obtains the desired result.

J Discussion about Assumption 5.4

In this section, we discuss the setting without Assumption 5.4. We begin by noting that the initialization of the fine-tuning phase can be formulated as an OOD generalization problem, which is characterized by Theorem E.2.

Then, we follow the same steps as in appendix F. Even without Assumption 5.4, Lemma F.1 continues to hold, as its proof does not depend on the specific structure of the distribution shift. Although Lemma F.2 relies on Assumption 5.4, its argument only requires an initialization bound on the β-type attention. This condition can be extended beyond Assumption 5.4 by imposing a stricter requirement on ϵ. Building on Lemma F.2, we can further establish Lemma F.3 and theorem F.4, even though both α-type and β-type attention weights depend strongly on the distribution shift.

For the characterization of forgetting in Theorem G.1, the key step is to project the changes in attention weights onto the directions of the training features. The projection can be extended beyond Assumption 5.4, while the results become more intricate and introduces additional dependencies on the distribution shift.

K Noisy Setting

We follow the setup of the main text, but now both the context responses and the query target contain independent Gaussian noise:

$$y_i = f(x_i) + \xi_i = \langle w, x_i \rangle + \xi_i, \qquad y_{\text{query}} = \langle w, x_{\text{query}} \rangle + \xi_{\text{query}},$$

where $\xi_i, \xi_{\text{query}} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$ and the noises are independent of w, the inputs, and one another. We define the variance of the averaged context noise for each feature as $\sigma^2_{\text{eff}} := \frac{K\sigma^2}{N}$.

Throughout, we use a hat to denote quantities in the noisy setting; for example, $\hat{L}$ denotes the loss in the noisy setting. Compared with the noiseless setting, the training loss contains two additional variance terms:

$$\hat{L}^{(\text{tr})}(Q) = L^{(\text{tr})}(Q) + \frac{\sigma^2_{\text{eff}}}{2}\mathbb{E}\left[\sum_{k=1}^{K} \mathbf{1}_{\{x_{\text{query}} = v_k\}} \sum_{j=1}^{K} \text{Attn}^2_{v_j}\right] + \frac{\sigma^2}{2}. \tag{28}$$

The noise introduces the following irreducible error floor for the training loss

$$\epsilon_0 := \frac{1}{2}\left[\sigma^2 + \frac{\sigma^2_{\text{eff}}(K + \sigma^2_{\text{eff}})}{K(1 + \sigma^2_{\text{eff}})}\right].$$

so that the convergence will occur if $\hat{L}^{(\text{tr})} - \epsilon_0$ becomes arbitrarily small, i.e., training stops at the first iterate $\hat{T}_1$ satisfying

$$\hat{L}^{(\text{tr})}_{v_k}(Q(\hat{T}_1)) - \frac{\epsilon_0}{K} < \frac{\epsilon}{K}, \qquad \forall k \in [K],$$

where $\epsilon > 0$. The same argument in [30] guarantees that there exists such a convergence time $\hat{T}_1 = \text{poly}\Big(\frac{1}{\epsilon}, K, \frac{1}{\eta}, \sigma\Big)$.

Theorem K.1. *(Noisy version of theorem 5.1) Fix $\epsilon > 0$ and suppose the training phase ends at time $\hat{T}_1$ under the stopping criterion above. The attention scores associated with OOD features are:*

$$\hat{\alpha}^{(\hat{T}_1)}_{u_k} = \frac{K\hat{\alpha}^*_{K,\epsilon}}{K-1}\left([\mathbf{M}^\top\mathbf{M}]_{k,k} - \langle u_k, \bar{v}\rangle^2\right), \;\; \hat{\beta}^{(\hat{T}_1)}_{u_{k'},u_k} = \frac{K\hat{\alpha}^*_{K,\epsilon}}{K-1}\left([\mathbf{M}^\top\mathbf{M}]_{k',k} - \langle u_k, \bar{v}\rangle\langle u_{k'}, \bar{v}\rangle\right),$$

where $\mathbf{M}$, $[\cdot]_{k',k}$, $\bar{v}$ are defined same as the main paper and

$$\hat{\alpha}^*_{K,\epsilon} = \left(1 - \frac{1}{K}\right)\log\left(\frac{(K-1)\,\text{Attn}^{(\hat{T}_1)}_v}{1 - \text{Attn}^{(\hat{T}_1)}_v}\right),$$

$$\text{Attn}^{(\hat{T}_1)}_v = \frac{K + \sigma^2_{\mathit{eff}}}{K(1+\sigma^2_{\mathit{eff}})} - \sqrt{\frac{2(K-1)}{K(1+\sigma^2_{\mathit{eff}})}\epsilon}. \tag{29}$$

The OOD error under the noisy setting is:

$$\begin{aligned}\hat{L}^{(\mathit{ft})}_{u_k}(Q(\hat{T}_1)) = &\frac{\left(\sum_{n\neq k}\exp(\hat{\beta}^{(\hat{T}_1)}_{u_n,u_k})\right)^2 + \sum_{n\neq k}\exp(\hat{\beta}^{(\hat{T}_1)}_{u_n,u_k})^2}{2K\left(\exp(\hat{\alpha}^{(\hat{T}_1)}_{u_k}) + \sum_{n\neq k}\exp(\hat{\beta}^{(\hat{T}_1)}_{u_n,u_k})\right)^2} \\ &+ \frac{\sigma^2_{\mathit{eff}}\left(\exp(\hat{\alpha}^{(\hat{T}_1)}_{u_k})^2 + \sum_{n\neq k}\exp(\hat{\beta}^{(\hat{T}_1)}_{u_n,u_k})^2\right)}{2K\left(\exp(\hat{\alpha}^{(\hat{T}_1)}_{u_k}) + \sum_{n\neq k}\exp(\hat{\beta}^{(\hat{T}_1)}_{u_n,u_k})\right)^2} + \frac{\sigma^2}{2K}.\end{aligned} \tag{30}$$

Comparison with noiseless setting

1. **Same geometric structure.** In the noisy case, $\hat{\alpha}^{(\hat{T}_1)}_{u_k}$ and $\hat{\beta}^{(\hat{T}_1)}_{u_{k'},u_k}$ have exactly the same dependence on $\mathbf{M}^\top\mathbf{M}$ and $\bar{v}$ as their noiseless counterparts, and the only change is the scaling parameter $\alpha^*_{K,\epsilon} \to \hat{\alpha}^*_{K,\epsilon}$.
2. **Noise modifies the feature attention at the convergence.** The attention at the convergence of the training takes the form of Eq. (29), where the noise introduces a noise-dependent shift and rescaling of the convergent attention, compared to the following attention $\text{Attn}^{(T_1)}_v = 1 - \sqrt{\frac{2(K-1)\epsilon}{K}}$ in the noiseless setting.
3. **The noisy OOD error has three components.** Equation (30) consists of: (i) the original noiseless OOD-error expression evaluated at the noisy terminal scores $\hat{\alpha}$ and $\hat{\beta}$; (ii) an attention-dependent context-noise variance term proportional to σ^2_{eff}; and (iii) the irreducible query-noise term $\sigma^2/(2K)$.

Proof. We present the proof in a way to emphasize the changes from the noiseless proof.

We first derive the attention model $\hat{\alpha}$ and $\hat{\beta}$ at the time of convergence. To this end, we first derive the training loss given in Eq. (28). In particular, by symmetry across the features, the new noise-dependent term (i.e., the second term) has the following form

$$\mathbb{E}\left[\sum_{j=1}^{K}\mathbf{1}_{\{x_{\text{query}}=v_j\}}\sum_{n=1}^{K}\text{Attn}_{v_n}^2\right]=\frac{\exp\left(\frac{2K\hat{\alpha}_v}{K-1}\right)+K-1}{\left(\exp\left(\frac{K\hat{\alpha}_v}{K-1}\right)+K-1\right)^2}.$$

Thus, the total loss is given by

$$\hat{L}^{(\text{tr})}(Q)=\epsilon_0+\frac{K(1+\sigma_{\text{eff}}^2)}{2(K-1)}\left(\frac{\exp\left(\frac{K\hat{\alpha}_v}{K-1}\right)}{\exp\left(\frac{K\hat{\alpha}_v}{K-1}\right)+K-1}-\frac{K+\sigma_{\text{eff}}^2}{K(1+\sigma_{\text{eff}}^2)}\right)^2.$$

Letting $\hat{L}^{(\text{tr})}(Q(\hat{T}_1))-\epsilon_0=\epsilon$ satisfies the convergent condition and thus solves the corresponding $\hat{\alpha}_v$ to be $\hat{\alpha}_{K,\epsilon}^*$.

We next derive the noisy OOD error, which can be expressed as follows for the query feature u_k:

$$\hat{L}_{u_k}^{(\text{ft})}(Q)=L_{u_k}^{(\text{ft})}(Q)+\frac{\sigma_{\text{eff}}^2}{2}\mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u_k\}}\sum_{n=1}^{K}\text{Attn}_{u_n}^2\right]+\frac{\sigma^2}{2K}.$$

The first term can be obtained as the noiseless error in original Theorem 5.1 but replacing $\alpha_{K,\epsilon}^*$ with $\hat{\alpha}_{K,\epsilon}^*$. The second term can be derived as follows:

$$\mathbb{E}\left[\mathbf{1}_{\{x_{\text{query}}=u_k\}}\sum_{n=1}^{K}\text{Attn}_{u_n}^2\right]=\frac{1}{K}\frac{\exp(\hat{\alpha}_{u_k}^{(\hat{T}_1)})^2+\sum_{n\neq k}\exp(\hat{\beta}_{u_n,u_k}^{(\hat{T}_1)})^2}{\left(\exp(\hat{\alpha}_{u_k}^{(\hat{T}_1)})+\sum_{n\neq k}\exp(\hat{\beta}_{u_n,u_k}^{(\hat{T}_1)})\right)^2}.$$

which completes the proof. □

We use the same noise model and notation as in our noisy version of OOD analysis. Suppose that Assumption 5.4 in the original paper holds.

We additionally assume that the stepsize lies strictly inside the range allowed in the original paper. More precisely, if $\eta_{\max}$ is the upper bound in Lemma F.3, fix $\delta>0$ and assume that the noisy stepsize satisfies

$$\hat{\eta}\leq\eta_{\max}-\delta.$$

Theorem K.2. *(Noisy version of theorem 5.5) Fix a problem instance $\mathcal{I}=(U,V,K,\epsilon,\hat{\eta})$ and a constant $\xi>0$. Under Assumption 5.4 and the stepsize condition above, there exists a positive, instance-dependent noise radius*

$$s^\star=s^\star(\mathcal{I},\xi)>0$$

such that, whenever

$$\sigma_{\text{eff}}^2<s^\star\qquad\text{and}\qquad\epsilon\geq\xi,$$

the noisy fine-tuning dynamics preserve all the sign, ordering, and stopping properties used in the original proof up to the noisy stopping time.

In particular, there exist an ordering $k(1),k(2),\ldots,k(K)$ and times $\hat{t}_1\leq\hat{t}_2\leq\cdots\leq\hat{t}_K$ with $\hat{t}_m=O\Big(\text{poly}\Big(K,\frac{1}{\epsilon},\frac{1}{\hat{\eta}},\sigma\Big)\Big)$ such that:

$$\hat{L}_{u_{k(m)}}^{(\text{ft})}\Big(Q\Big(\hat{T}_1+\hat{t}_m\Big)\Big)-\frac{\epsilon_0}{K}\leq\frac{\epsilon}{K},\qquad m\in[K],$$

As in the original paper, define the noisy cumulative score updates by

$$\hat{\alpha}_{u_k}^{(\hat{T}_1+t)} = \hat{\alpha}_{u_k}^{(\hat{T}_1)} + \Delta_{k,\alpha}^{(t)},$$
$$\hat{\beta}_{u_n,u_k}^{(\hat{T}_1+t)} = \hat{\beta}_{u_n,u_k}^{(\hat{T}_1)} - \Lambda_{n,k,\beta}^{(t)}, \qquad n \neq k.$$

To state the cumulative-update bounds, define the noisy logarithmic scale

$$\hat{\rho}_{K,\epsilon,\sigma} := \frac{K}{K-1}\hat{\alpha}_{K,\epsilon}^{*}.$$

At the corresponding stopping times,

$$\Delta_{k(m),\alpha}^{(\hat{t}_m)} = \Theta\left(\left[1 - \gamma_{k(m)}^2 - \frac{\gamma_{k(m)}\hat{\gamma}_\alpha}{K}\right]\hat{\rho}_{K,\epsilon,\sigma}\right), \tag{31}$$

and, for every $n \neq k(m)$,

$$\Lambda_{n,k(m),\beta}^{(\hat{t}_m)} = \Theta\left(\frac{\Delta_{k(m),\alpha}^{(\hat{t}_m)}}{K} + \frac{\gamma_{k(m)}\hat{\gamma}_\beta}{K}\hat{\rho}_{K,\epsilon,\sigma}\right), \tag{32}$$

where $\hat{\gamma}_\alpha, \hat{\gamma}_\beta \in [\gamma_{\min}, \gamma_{\max}]$.

Comparison with the noiseless setting

1. **Two additional margin constants.** The stepsize margin δ ensures that the chosen stepsize remains strictly inside the stable range when the optimization dynamics are perturbed by noise. The tolerance margin ξ ensures that the noise-dependent terms in the training loss in Eq. (28) can be absorbed while its upper bound still satisfies the convergence criterion. For each fixed pair of positive margins (δ, ξ), there exists a sufficiently small noise radius to preserves both conditions.
2. **The optimization dynamics remain stable under small noise.** Within the above noise radius, the noisy dynamics preserve the main properties of the noiseless proof: the stopping criterion is reached in polynomial time, the ordering argument remains valid, and the directions of the attention-score updates are unchanged.
3. **The cumulative-update bounds retain the same form as in the original paper.** The conclusions in Eqs. (31)–(32) have the same structure as their noiseless counterparts. The clean terminal scale $\alpha_{K,\epsilon}^{*}$ is replaced by its noisy counterpart $\hat{\alpha}_{K,\epsilon}^{*}$, and the original logarithmic factor is replaced by the noisy logarithmic scale $\hat{\rho}_{K,\epsilon,\sigma}$. As $\sigma \to 0$, we have $\hat{\rho}_{K,\epsilon,\sigma} = \Theta\big(\log \frac{K}{\epsilon}\big)$. This noisy logarithmic scale remains within positive constant factors of its noiseless counterpart.

Proof. We first note that the proof of Theorem 5.5 relies on strict one-step inequalities governing the signs of the α-type and β-type updates, the ordering of the β-type scores, and a quantitative lower bound on optimization progress. Importantly, at every pre-stopping iterate, there is a positive local tolerance within which these inequalities remain valid.

At each iterate of the noisy fine-tuning trajectory, we compare the actual noisy gradient step with a counterfactual noiseless gradient step initialized at the same model parameter. Over the pre-stopping region, the two model updates differ by an $O\big(\hat{\eta}\sigma_{\text{eff}}^2\big)$ perturbation.

We use the noisy stepsize $\hat{\eta}$ introduced in the theorem, which lies strictly inside the admissible range by the margin δ. At every pre-stopping gradient-descent step, the strict inequalities used in the original argument provide a positive tolerance on the model-update perturbation: whenever the $O\big(\hat{\eta}\sigma_{\text{eff}}^2\big)$ perturbation is smaller than this tolerance, that step continues to satisfy the key sign, ordering, and progress properties of the original proof. Applying the same one-step induction and progress argument therefore gives a polynomial upper bound on the number of steps required to reach the stopping criterion, provided that these local perturbation tolerances are respected. This polynomial horizon contains only finitely many steps. Hence, for the fixed problem instance, the minimum of

the corresponding local tolerances is positive. We choose $s^\star$ so that the one-step perturbation is smaller than this minimum tolerance whenever $\sigma_{\text{eff}}^2 < s^\star$. It follows that every noisy update obeys the required properties and that the noisy dynamics reach the prescribed stopping threshold in polynomial time.

The auxiliary constant ξ plays a separate role and is not needed for the stability of the gradient dynamics. The noisy objective contains the additional attention-dependent variance term in Eq. (28). The margin ξ provides sufficient terminal-loss slack to absorb this term, ensuring that the resulting noisy excess-loss upper bound remains below the prescribed threshold ϵ.

Noise changes the fine-tuning initialization from $\alpha^*_{K,\epsilon}$ to $\hat{\alpha}^*_{K,\epsilon}$ and replaces the original logarithmic factor by $\hat{\rho}_{K,\epsilon,\sigma}$. Within the radius $s^\star$, this noisy logarithmic scale remains of the same order as its noiseless counterpart. Applying the original upper- and lower-bound arguments with these substitutions yields Eqs. (31)–(32). □

Theorem K.3. *(Noisy version of theorem 5.6) Suppose Assumption 5.4 holds and the stepsize satisfies the noisy condition. At time $\hat{T}_2 = \hat{t}_K$, the fine-tuning phase ends and the model performance on the original feature set can be captured by*

$$\begin{aligned}\hat{L}_{v_k}^{(\text{tr})}\Big(Q(\hat{T}_1+\hat{T}_2)\Big) = &\frac{\left(\sum_{n\neq k}\exp\left(\hat{\beta}_{v_n,v_k}^{(\hat{T}_1+\hat{T}_2)}\right)\right)^2 + \sum_{n\neq k}\exp\left(2\hat{\beta}_{v_n,v_k}^{(\hat{T}_1+\hat{T}_2)}\right)}{2\left(\exp\left(\hat{\alpha}_{v_k}^{(\hat{T}_1+\hat{T}_2)}\right)+\sum_{n\neq k}\exp\left(\hat{\beta}_{v_n,v_k}^{(\hat{T}_1+\hat{T}_2)}\right)\right)^2} \\ &+ \frac{\sigma_{\text{eff}}^2\left(\exp\left(2\hat{\alpha}_{v_k}^{(\hat{T}_1+\hat{T}_2)}\right)+\sum_{n\neq k}\exp\left(2\hat{\beta}_{v_n,v_k}^{(\hat{T}_1+\hat{T}_2)}\right)\right)}{2\left(\exp\left(\hat{\alpha}_{v_k}^{(\hat{T}_1+\hat{T}_2)}\right)+\sum_{n\neq k}\exp\left(\hat{\beta}_{v_n,v_k}^{(\hat{T}_1+\hat{T}_2)}\right)\right)^2} + \frac{\sigma^2}{2},\end{aligned} \tag{33}$$

where

$$\hat{\alpha}_{v_k}^{(\hat{T}_1+\hat{T}_2)} = \hat{\alpha}_{v_k}^{(\hat{T}_1)} + \gamma_k^2\Delta_{k,\alpha}^{(\hat{T}_2)}, \tag{34}$$

and, for every $n \neq k$,

$$\hat{\beta}_{v_n,v_k}^{(\hat{T}_1+\hat{T}_2)} = \hat{\beta}_{v_n,v_k}^{(\hat{T}_1)} - \gamma_k\gamma_n\Lambda_{n,k,\beta}^{(\hat{T}_2)}. \tag{35}$$

Comparison

1. Compared with the noiseless result, the form of the α/β attentions is unchanged. This is because the source-target geometry is unrelated to the noise.
2. In the detailed expression, α and β are replaced by $\hat{\alpha}$ and $\hat{\beta}$, and the loss contains the two additional noise terms in Eq. (28).

The proof follows the same projection argument of theorem 5.6.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the this paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We state our limitation in the conclusion section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide the full set of assumptions and complete proof for each theoretical result in Section 3-5 and Appendix D-K.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the information needed to reproduce our experiments in Section 6 and Appendix A-C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide our code in the supplementary material.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: This paper provided training and test details in Section 6 and Appendix A-C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report our results with standard deviation in fig. 3c.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide GPU information in appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This research follows the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work studies foundational theory, with no direct societal impacts identified.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This work does not release high-risk data or models requiring safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original model and dataset papers, document their licenses in Appendix C, and use these assets in accordance with their terms.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This work does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or human-subject research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve human-subject research requiring IRB approval.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not used as important, original, or non-standard components of the core methodology.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.