When and How Unlabeled Data Provably Improve In-Context Learning

Yingcong Li [1] Xiangyu Chang [2] Muti Kara [3] Xiaofeng Liu [1] Amit Roy-Chowdhury [2] Samet Oymak [1]

Abstract

Recent research shows that in-context learning (ICL) can be effective even when demonstrations have missing or incorrect labels. To shed light on this capability, we examine a canonical setting where the demonstrations are drawn according to a binary Gaussian mixture model (GMM) and a certain fraction of the demonstrations have missing labels. We provide a comprehensive theoretical study to show that: (1) The loss landscape of one-layer linear attention models recover the optimal fully-supervised estimator but completely fail to exploit unlabeled data; (2) In contrast, multilayer or looped transformers can effectively leverage unlabeled data by implicitly constructing estimators of the form $\sum_{i\geq 0} a_i(\boldsymbol{X}^\top\boldsymbol{X})^i\boldsymbol{X}^\top\boldsymbol{y}$ with $\boldsymbol{X}$ and $\boldsymbol{y}$ denoting features and partially-observed labels (with missing entries set to zero). We characterize the class of polynomials that can be expressed as a function of depth and draw connections to Expectation Maximization, an iterative pseudo-labeling algorithm commonly used in semi-supervised learning. Importantly, the leading polynomial power is exponential in depth, so mild amount of depth/looping suffices.

1. Introduction

In-Context Learning (ICL) is an intriguing capability of modern language models and has enjoyed remarkable empirical success (Brown et al., 2020; Min et al., 2022). The push toward long-context models (Snell et al., 2024; Guo et al., 2025) has further boosted the benefits of ICL by allowing the model to ingest a large number of demonstrations. For instance, in "Many-shot in-context learning" paper, (Agarwal et al., 2024) demonstrate that pushing more examples into context window can substantially boost the accuracy. The many-shot ICL setting naturally raises the question of when and how ICL can succeed with weaker supervision. This motivates our central question:

Q: How can transformers learn from unlabeled data?

We primarily investigate this question under a semisupervised ICL (SS-ICL) setting with GMMs. Formally, given a prompt containing a dataset of feature-label pairs $(\boldsymbol{x}_i, y_i)_{i=1}^n \in \mathbb{R}^d \times \mathbb{R}$ as demonstrations and a query feature $\boldsymbol{x}$, a model learns to predict the corresponding output y given prompt. This prompt model is well studied under various fully-supervised settings (Garg et al., 2022; Von Oswald et al., 2023; Ahn et al., 2023; Akyürek et al., 2023; Mahankali et al., 2024; Collins et al., 2024; Shen et al., 2024) where each demonstration includes a clearly labeled output. In our SS-ICL setting, only m out of n total samples have correct labels ($m \leq n$) either -1 or 1, and remaining labels are unknown and fed to the model as $y_i = 0$.

In this work, we provide a comprehensive theoretical and empirical study of attention models with varying depths when trained with SS-ICL. Our analysis reveals **the importance of depth**: despite being able to implement the optimal supervised learner, single-layer linear attention completely fails to leverage unlabeled examples. In contrast, deeper or looped transformer architectures can emulate strong semisupervision algorithms. Our specific contributions are:

- ◇ **Landscape of one-layer linear attention (§3):** We study the optimization landscape of single-layer linear attention for the SS-ICL problem under an isotropic task prior. We prove that the global minimum of the loss function is the plug-in estimator (cf. (SPI)). This implies that 1-layer model learns Bayes-optimal classifier in the fully-supervised setting, but completely fails to make use of unlabeled data.
- ◇ **Depth is crucial but shallow can suffice (§4):** We show that multilayer linear attention can emulate semisupervised learners by implementing polynomial estimators of the form $\hat{\boldsymbol{\mu}} = \sum_{i\geq 0}^K a_i(\boldsymbol{X}^\top\boldsymbol{X})^i\boldsymbol{X}^\top\boldsymbol{y}$, which can be interpreted as the model implicitly conducting *iterative pseudo-labeling*. We show that L-layer (or looped) attention can express up to $K = O(3^L)$ powers, highlighting exponentiation requires only logarithmic depth. Corroborating these, experiments reveal that shallow transformers with $L \geq 2$ already achieve strong results and their performance can

[*] Equal contribution [1] University of Michigan [2] University of California, Riverside [3] Bilkent University. Correspondence to: Samet Oymak <oymak@umich.edu>.

Proceedings of the 1^{st} ICML Workshop on Foundation Models for Structured Data, Vancouver, Canada. 2025. Copyright 2025 by the author(s).

be approximately predicted through an eigen-estimator combining $i = 0$ and ∞ (see (SSPI-k)).

◇ **Applications to Tabular FMs (§B):** Tabular foundation models represent a suitable application of theory as they also model the ICL examples with a single token. To harness unlabeled examples, we propose a novel strategy that iteratively creates soft pseudo-labels by *explicitly looping the tabular FM* while controlling validation risk. Focusing on the few-shot learning setting where TabPFN-v2 excels, we demonstrate that our approach can significantly improve performance on various real-world datasets.

2. Problem Setup and Preliminaries

2.1. Semi-supervised Data Model

Consider a d-dimensional semi-supervised binary GMM with n examples $(\boldsymbol{x}_i, y_i)_{i=1}^n$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ denotes the feature vector and $y_i \in \{-1, 0, 1\}$ represents the corresponding observed label, with $y_i = 0$ indicating a missing label, and each label is revealed independently with probability $p \in [0, 1]$. Specifically, the data is generated as follows (for each $i \in [n]$):

$$\boldsymbol{x}_i = y_i^c \cdot \boldsymbol{\mu} + \boldsymbol{\xi}_i \quad \text{and} \quad y_i = \begin{cases} y_i^c, & \text{w.p.} \quad p \\ 0, & \text{w.p.} \quad 1-p \end{cases} \tag{1}$$

where $\boldsymbol{y}_i^c \sim$ Unif{1,-1} denotes the true label. Here $\boldsymbol{\mu} \sim$ Unif($\mathbb{S}^{d-1}$) denotes the task mean, which is sampled uniformly from the unit sphere, and $\boldsymbol{\xi}_i \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$ is the random noise with $\sigma \geq 0$ being the noise level that controls the variability of $\boldsymbol{x}_i$ around its mean. Observe that $p = 1$ corresponds to fully supervised learning and $p = 0$ corresponds to fully-unsupervised learning.

2.2. In-context Learning and Linear Attention

We build on the setting of (Garg et al., 2022; Mahankali et al., 2024; Zhang et al., 2023; Li et al., 2024) and construct the in-context prompts with examples drawn from (1).

Prompt Generation: Given a task vector $\boldsymbol{\mu} \sim$ Unif($\mathbb{S}^{d-1}$), we sample $(n + 1)$ in-context demonstrations $(\boldsymbol{x}_i, y_i)_{i=1}^{n+1}$ according to (1) and construct the prompt

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_n & \boldsymbol{x} \\ y_1 & y_2 & \cdots & y_n & 0 \end{bmatrix}^\top \in \mathbb{R}^{(n+1)\times(d+1)}. \tag{2}$$

We will investigate training a transformer such that given $\boldsymbol{Z}$ as prompt, it correctly predicts the label $\boldsymbol{y} := y_{n+1}^c$ of the query $\boldsymbol{x} := \boldsymbol{x}_{n+1}$ through ICL.

Model Architecture: Given any prompt $\boldsymbol{Z} \in \mathbb{R}^{(n+1)\times(d+1)}$, the linear attention mechanism outputs

$$\text{att}(\boldsymbol{Z}; \mathcal{W}) = (\boldsymbol{Z}\boldsymbol{W}_q \boldsymbol{W}_k^\top \boldsymbol{Z}^\top)\boldsymbol{M}\boldsymbol{Z}\boldsymbol{W}_v \tag{3}$$

where $\mathcal{W} := \{\boldsymbol{W}_k, \boldsymbol{W}_q, \boldsymbol{W}_v \in \mathbb{R}^{(d+1)\times(d+1)}\}$ denotes the key, query and value weight matrices, respectively. Note that the label for the query $\boldsymbol{x}$ is excluded from the prompt $\boldsymbol{Z}$. Similar to Ahn et al. (2023), we consider a training objective with a mask $\boldsymbol{M} = \begin{bmatrix} \boldsymbol{I}_n & 0 \\ 0 & 0 \end{bmatrix}$ to ensure inputs cannot attend to their own labels and training can be parallelized.

Building upon the single-layer linear attention mechanism of (3), we can extend our model to multiple layers to capture more complex patterns. Consider optimizing an L-layer linear attention model and let $\boldsymbol{Z}_\ell$ be the input of ℓth layer, $\ell \in [L]$. Additionally, let $\mathcal{W}_\ell := \{\boldsymbol{W}_{k\ell}, \boldsymbol{W}_{q\ell}, \boldsymbol{W}_{v\ell} \in \mathbb{R}^{(d+1)\times(d+1)}\}$ be the corresponding weight matrices of ℓth layer. Then, the input prompt of ℓth layer is defined by

$$\boldsymbol{Z}_\ell = \boldsymbol{Z}_{\ell-1} + \text{att}(\boldsymbol{Z}_{\ell-1}; \mathcal{W}_{\ell-1}) \quad \text{for} \quad \ell = 2, \ldots L,$$

and $\boldsymbol{Z}_1 = \boldsymbol{Z}$. We focus on the next-token prediction setting, where the model makes a prediction based on the final query token $[\boldsymbol{x}^\top \ 0]^\top$. Let $\boldsymbol{h} \in \mathbb{R}^{d+1}$ denote the linear prediction head. We define the output of the L-layer linear attention model at the last (query) token as

$$f_{\text{att-}L}(\boldsymbol{Z}) = \boldsymbol{h}^\top \text{att}(\boldsymbol{Z}_L; \mathcal{W}_L)_{[n+1]}. \tag{4}$$

The predicted label is given by $y_{\text{att-}L}(\boldsymbol{Z}) = \text{sgn}(f_{\text{att-}L}(\boldsymbol{Z}))$.

Model Training: Consider the ICL setting where each input prompt $\boldsymbol{Z}$ (cf. (2)) corresponds to a randomly sampled task vector $\boldsymbol{\mu} \sim$ Unif($\mathbb{S}^{d-1}$) and let $\ell(\cdot) : \mathbb{R} \to \mathbb{R}$ be the loss function. Additionally, define the set of attention weights $\mathcal{W}^{(L)} := \cup_{\ell=1}^L \mathcal{W}_\ell \in (\mathbb{R}^{(d+1)\times(d+1)})^{3L}$. The objective of L-layer linear atention takes the following form:

$$\begin{aligned} &\min_{\mathcal{W}^{(L)}, \boldsymbol{h}} \mathcal{L}_{\text{att-}L}(\mathcal{W}^{(L)}, \boldsymbol{h}) \\ &\text{where} \quad \mathcal{L}_{\text{att-}L}(\mathcal{W}^{(L)}, \boldsymbol{h}) = \mathbb{E}\left[\ell(y, f_{\text{att-}L}(\boldsymbol{Z}))\right]. \end{aligned} \tag{5}$$

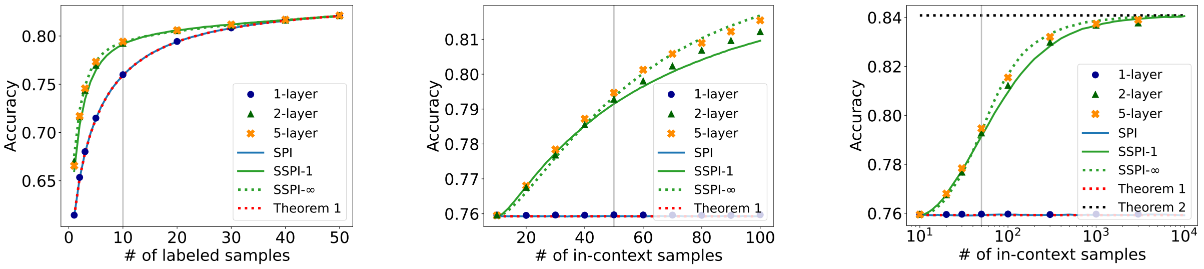
Here, $y := y_{n+1}^c$ and the expectation subsumes the randomness of $\boldsymbol{\mu}$ and $(\boldsymbol{\xi}_i, y_i)_{i=1}^{n+1}$.

3. Loss Landscape of One-layer Linear Attention under SS-ICL

In this section, we analyze the optimization behavior of single-layer linear attention under SS-ICL.

Supervised Plug-in (SPI) Estimator: Under our problem setting, SPI is the asymptotically Bayes-optimal estimator given only labeled data (Hastie et al., 2009; Devroye et al., 2013). Consider the binary semi-supervised GMM problem described in (1) with dataset $(\boldsymbol{x}_i, y_i)_{i=1}^n$, and let $\mathcal{I} \subset [n]$ represent the indices of labeled samples, e.g., $y_i \neq 0$ for $i \in \mathcal{I}$. The SPI estimator returns the task mean

$$\hat{\boldsymbol{\mu}}_s = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} y_i \boldsymbol{x}_i. \tag{SPI}$$



(a) $n = 50, p \in (0, 1]$. (b) $np = 10, n \in \{10, \cdots, 100\}$. (c) $np = 10, n \in \{10, \cdots, 10000\}$.

Figure 1. Experimental results support our theoretical findings presented in Sections 3 and 4.

Theorem 3.1. *Consider the objective (cf. (5)) with $L = 1$ and squared loss function $\ell(y, \hat{y}) = (y - \hat{y})^2$, and denote the optimal prediction as $y^\star_{\text{att-1}}(\boldsymbol{Z})$. Let $\hat{\boldsymbol{\mu}}_s$ represent the SPI estimator defined in (SPI). Then, for any $\boldsymbol{Z}$ from (2),*

$$y^\star_{\text{att-1}}(\boldsymbol{Z}) = sgn(\boldsymbol{x}^\top \hat{\boldsymbol{\mu}}_s). \tag{6}$$

Additionally, its classification error obeys $\mathbb{P}(y^\star_{\text{att-1}}(\boldsymbol{Z}) \neq y)$

$$= \mathbb{E}_{g \sim \mathcal{N}(0,1), h \sim \mathcal{X}^2_{d-1}} \left[Q\left(\frac{1 + \varepsilon_\sigma g}{\sigma \sqrt{(1 + \varepsilon_\sigma g)^2 + \varepsilon^2_\sigma h}} \right) \right] \tag{7}$$

$$\leq Q\left(\frac{1 - 10d\varepsilon^2_\sigma}{\sigma} \right) + e^{-d} + e^{-1/8\varepsilon^2_\sigma}$$

where we define $\varepsilon_\sigma = \sigma/\sqrt{np}$ and $\mathcal{X}^2_d$ defines chi-squared distribution with d degrees of freedom.

Eq. (6) shows that one-layer linear attention model indeed implements SPI predictor, assuming access to np labeled examples. Most existing work (Thrampoulidis et al., 2020; Wang & Thrampoulidis, 2022) focuses on a single classification task under asymptotic data regimes. In contrast, within the ICL framework considered in our setting, the task mean $\boldsymbol{\mu}$ is randomly sampled, and the classification error is computed by averaging over random draws of $\boldsymbol{Z}$, y, and $\boldsymbol{\mu}$. Accordingly, in (7), we express the error in a simplified form as an expectation.

The experimental results in Figure 1 support Theorem 3.1, where dark blue circular markers represent the performance of the single-layer linear attention model, blue curves show the classification accuracy of the SPI estimator, and the red dotted curves depict $1 - \mathbb{P}(y^\star_{\text{att-1}}(\boldsymbol{Z}) \neq y)$ as computed from (7). The alignments of these curves empirically validate Theorem 3.1. Based on these results, we can conclude: *1-layer linear attention learns optimal supervised estimator but doesn't benefit from unlabeled data.*

As shown in Figs 1(b) and 1(c), when the number of labeled samples ($np = 10$) is fixed, increasing the number of unlabeled examples (even up to $\sim$ 10000) has no effect on performance, as the dark blue markers remain at the same level. At first glance, this may seem counterintuitive—while the data is unlabeled, it still contains information about the classification feature. For instance, the mean of the data points carries relevant information, and one might expect the model to extract and leverage this for better predictions. This expectation is particularly reasonable when a large amount of unlabeled data is available, as the sample covariance matrix approximates the population covariance, i.e., $\mathbb{E}[\boldsymbol{X}^\top \boldsymbol{X}/n] = \boldsymbol{\mu}\boldsymbol{\mu}^\top + \sigma^2 \boldsymbol{I}$ where $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n]^\top \in \mathbb{R}^{n \times d}$. The key insight into why single-layer attention fails to leverage unlabeled data lies in the expectation structure. In our isotropic GMM setting where $\boldsymbol{\mu} \sim \text{Unif}(\mathbb{S}^{d-1})$, the sample covariance matrix converges to $\mathbb{E}[\boldsymbol{X}^\top \boldsymbol{X}/n] = \mathbb{E}[\boldsymbol{\mu}\boldsymbol{\mu}^\top] + \sigma^2 \boldsymbol{I} = (1/d + \sigma^2)\boldsymbol{I}$, which contains no task-specific information. The expectation across multiple tasks loses the signal from $\boldsymbol{\mu}$. This explains why single-layer attention, operating in a meta-learning framework across many tasks rather than optimizing for a single fixed task, cannot extract useful information from unlabeled data.

In the following section, we study multi-layer linear attention and demonstrate that it has the ability to propagate $\boldsymbol{X}^\top \boldsymbol{X}$ into deeper layers, thereby enabling the model to utilize the unlabeled data.

4. Multi-layer Attention and Benefits of Depth

In this section, we explore how deeper attention models can effectively utilize the unlabeled data. Let

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_n \end{bmatrix}^\top \text{ and } \boldsymbol{y} = \begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix}^\top. \tag{8}$$

We first present the following propositions to show that multi-layer as well as looped linear attention can be expressed as a polynomial function of $\boldsymbol{X}^\top \boldsymbol{X}$. This structure allows the models to leverage unlabeled data to improve the estimation of the task mean $\boldsymbol{\mu}$.

Proposition 4.1. *Given an L-layer linear attention model described in Section 2.2 with input prompt $\boldsymbol{Z}$ defined in (2), one can construct the key, query, value weight matrices and the linear prediction head such that the model outputs*

$$f_{\text{att-}L}(\boldsymbol{Z}) = \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{X}^\top \boldsymbol{y}. \tag{9}$$

Then, the following $\boldsymbol{A}$ matrices are achievable via label and feature updates:

- ***Label propagation:*** $\boldsymbol{A} = c\prod_{\ell=1}^{L-1}\left(\boldsymbol{I} + c_\ell \boldsymbol{X}^\top \boldsymbol{X}\right)$ *for arbitrary constants* $\{c, c_1, \cdots, c_{L-1}\}$*;*
- ***Feature propagation:*** $\boldsymbol{A} = c\left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{3^{L-1}-1}$ *for an arbitrary constant* c*.*

Proposition 4.2. *Consider the same setting as in Proposition 4.1. There exists a single-layer linear attention model whose parameters can be constructed such that, when looped* L *times, its output reproduces that of (9), with* $c_\ell \equiv c'$ *for some arbitrary constant* c'*.*

In the following, we provide further clarification on the label and feature propagation.

1. The final prediction of the label propagation process can be rewritten as (for $\ell \in [L-1]$)

   $$f_{\text{att-}L}(\boldsymbol{Z}) = c\boldsymbol{x}^\top \boldsymbol{X}^\top \boldsymbol{y}_L \;\text{ where }\; \boldsymbol{y}_{\ell+1} = (\boldsymbol{I} + c_\ell \boldsymbol{X}\boldsymbol{X}^\top)\boldsymbol{y}_\ell,$$

   with $\boldsymbol{y}_1 = \boldsymbol{y}$. Here, $\boldsymbol{y}_\ell$ can be interpreted as the pseudo-labels input to the ℓth layer, and each c_ℓ is parameterized by the attention mechanism in the corresponding layer. The L-layer linear attention process shares similarities with the Expectation-Maximization algorithm for semi-supervised learning, with L iterations of pseudo-labeling and a different label update strategy.
2. In contrast, the feature propagation process yields

   $$f_{\text{all-}L}(\boldsymbol{Z}) = c\boldsymbol{x}_L^\top \boldsymbol{X}_L^\top \boldsymbol{y} \;\text{ where }\; \begin{cases} \boldsymbol{X}_{\ell+1} = (\boldsymbol{X}_\ell \boldsymbol{X}_\ell^\top)\boldsymbol{X}_\ell \\ \boldsymbol{x}_{\ell+1} = (\boldsymbol{X}_\ell^\top \boldsymbol{X}_\ell)\boldsymbol{x}_\ell \end{cases}$$

   with $(\boldsymbol{X}_1, \boldsymbol{x}_1) = (\boldsymbol{X}, \boldsymbol{x})$. Here, $(\boldsymbol{X}_\ell, \boldsymbol{x}_\ell)$ can be viewed as the input features at the ℓth layer, encoding exponentially higher-order powers of $\boldsymbol{X}^\top \boldsymbol{X}$. This result highlights that a linear attention model requires only $O(\log K)$ layers to represent polynomial functions of degree K.

Our construction for *label propagation* is inherently related to the GD emulation capability of linear attention (Ahn et al., 2023). However, the *feature propagation* construction is fundamentally different. The lemma below shows that, even if the multilayer model can express polynomials of $\boldsymbol{X}^\top \boldsymbol{X}$ with exponential degrees in depth, the expressible manifold of polynomials has dimensionality linear in depth.

Lemma 4.3 (Label + Feature Propagation)**.** *For an* L*-layer linear attention model, the resulting eventual prediction corresponds to the matrix* $\boldsymbol{A}$ *in Proposition 4.1 of the form*

$$\boldsymbol{A} = \sum_{\ell=0}^{(3^L-3)/2} a_\ell (\boldsymbol{X}^\top \boldsymbol{X})^\ell. \tag{10}$$

The coefficients $\boldsymbol{a} := [a_0 \;\; \cdots \;\; a_{(3^L-3)/2}]^\top$ *lie on a manifold of dimension at most* $2L$ *as* $\boldsymbol{a}$ *can be expressed as* $\boldsymbol{a} = g(\boldsymbol{c})$ *for some smooth function* $g : \mathbb{R}^{2L} \to \mathbb{R}^{(3^L-3)/2}$ *with* $\boldsymbol{c}$ *representing the parameters of individual layers.*

Motivated by Proposition 4.1 that multi-layer linear attention can implement higher-degree polynomials of $\boldsymbol{X}^\top \boldsymbol{X}$, we introduce the following SSPI estimator.

Semisupervised Plug-in (SSPI) Estimator Observe that the feature covariance satisfies $\mathbb{E}[\boldsymbol{X}^\top \boldsymbol{X}]/n = \boldsymbol{\mu}\boldsymbol{\mu}^\top + \sigma^2 \boldsymbol{I}$. We propose the semisupervised plug-in estimator as follows:

$$\hat{\boldsymbol{\mu}}_{ss\text{-}k} = \alpha\hat{\boldsymbol{\mu}}_s + (1-\alpha)(\boldsymbol{X}^\top \boldsymbol{X}/n - \sigma^2 \boldsymbol{I})^k \hat{\boldsymbol{\mu}}_s \quad \text{(SSPI-}k\text{)}$$

where $\hat{\boldsymbol{\mu}}_s$ is the SPI estimator, and $\alpha \in [0, 1]$ controls the trade-off between the fully- and semi-supervised estimators. The optimal α depends on the problem parameters n, d, p. Note that as $k \to \infty$, $(\boldsymbol{X}^\top \boldsymbol{X}/n - \sigma^2 \boldsymbol{I})^k$ converges to a rank-one projection onto the top eigenvector of the debiased covariance matrix, serving as an estimator for $\boldsymbol{\mu}$ (up to sign).

In Figure 1, we present the prediction accuracies of 2-/5-layer linear attention models, and evaluate the SSPI algorithm with varying k values using their respective optimal choices of α. The results reveal a close alignment between multi-layer linear attention and SSPI estimators. Notably, the 2-layer model outperforms SSPI-1, due to its ability to implement higher-degree polynomials of $\boldsymbol{X}^\top \boldsymbol{X}$ (cf. Proposition 4.1 and (10)). Furthermore, since the 5-layer model is capable of representing higher-order functions than the 2-layer model, it can better estimate the top eigenvector, resulting in performance that closely matches that of SSPI-∞.

In the following, we analyze the optimal classifier of the form sgn$(\boldsymbol{x}^\top \boldsymbol{A}\hat{\boldsymbol{\mu}}_s)$ for a GMM, and provide insights into its behavior in the asymptotic regime as $n \to \infty$.

Theorem 4.4. *Consider a GMM defined in Section 2.1 and suppose that* $(\boldsymbol{x}_i, y_i)_{i=1}^{n+1}$ *is generated using a fixed* $\boldsymbol{\mu}$ *following (1). Given matrix* $\boldsymbol{A} \in \mathbb{R}^{d\times d}$*, define prediction*

$$\hat{y}_{\boldsymbol{A}} = sgn(\boldsymbol{x}^\top \boldsymbol{A}\hat{\boldsymbol{\mu}}_s).$$

where $\hat{\boldsymbol{\mu}}_s$ *is the SPI estimator defined in (SPI). Let* $\mathcal{A}^\star := \min_{\boldsymbol{A}\in\mathbb{R}^{d\times d}} \mathbb{P}(\hat{y}_{\boldsymbol{A}} \neq y)$ *be its optimal solution set. Then,* $\boldsymbol{\mu}\boldsymbol{\mu}^\top \in \mathcal{A}^\star$*. Additionally, it obeys* $\mathbb{P}(\hat{y}_{\boldsymbol{\mu}\boldsymbol{\mu}^\top} \neq y) =$

$$Q(1/\sigma) + Q(\sqrt{np}/\sigma) - 2Q(1/\sigma)Q(\sqrt{np}/\sigma). \tag{11}$$

Theorem 4.5. *Consider an* L*-layer linear attention model with* $L \geq 2$ *and* $n = \infty$*. Additionally, let* $\hat{\boldsymbol{\mu}}_s$ *be the SPI estimator defined in (SPI). There exist model constructions such that for any* $\boldsymbol{Z}$ *following (2), its prediction satisfies*

$$y_{att\text{-}L}(\boldsymbol{Z}) = sgn(\boldsymbol{x}^\top \boldsymbol{\mu}\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s).$$

The proof follows directly from Proposition 4.1 (label propagation). The results in Figure 1(c) validate Theorem 4.5, showing that as n becomes large enough, (i.e., $n = 10000$) the predictions from both 2-layer and 5-layer linear attention models, as well as the SSPI-1 and SSPI-∞ estimators, closely align with the classification error characterized in Theorem 4.4, depicted by the black dotted line. Non-asymptotic result is presented in Appendix E.5.

Acknowledgements

This work was supported in part by the National Science Foundation grants CCF-2046816, CCF-2403075, CCF-2008020, the Office of Naval Research grant N000142412289, and by gifts/awards from Open Philanthropy, Amazon Research, and Google Research.

References

Agarwal, R., Singh, A., Zhang, L. M., Bohnet, B., Chan, S., Anand, A., Abbas, Z., Nova, A., Co-Reyes, J. D., Chu, E., et al. Many-shot in-context learning. *arXiv preprint arXiv:2404.11018*, 2024.

Ahn, K., Cheng, X., Daneshmand, H., and Sra, S. Transformers learn to implement preconditioned gradient descent for in-context learning. *Advances in Neural Information Processing Systems*, 36, 2023.

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

Bai, Y., Chen, F., Wang, H., Xiong, C., and Mei, S. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *Advances in neural information processing systems*, 36:57125–57211, 2023.

Balakrishnan, S., Wainwright, M. J., and Yu, B. Statistical guarantees for the EM algorithm: From population to sample-based analysis. In *Annals of Statistics*, volume 45, pp. 77–120. Institute of Mathematical Statistics, 2017.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Collins, L., Parulekar, A., Mokhtari, A., Sanghavi, S., and Shakkottai, S. In-context learning with transformers: Softmax attention adapts to function lipschitzness. *arXiv preprint arXiv:2402.11639*, 2024.

Dai, D., Sun, Y., Dong, L., Hao, Y., Ma, S., Sui, Z., and Wei, F. Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 4005–4019, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.247. URL https://aclanthology.org/2023.findings-acl.247.

Devroye, L., Györfi, L., and Lugosi, G. *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media, 2013.

Fan, C., Liu, S., Motamed, S., Zhong, S., Savarese, S., Niebles, J. C., Anandkumar, A., Gaidon, A., and Scherer, S. Expectation maximization pseudo labels. *arXiv preprint arXiv:2305.01747*, 2023.

Garg, S., Tsipras, D., Liang, P. S., and Valiant, G. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

Hollmann, N., Müller, S., Purucker, L., Krishnakumar, A., Körfer, M., Hoo, S. B., Schirrmeister, R. T., and Hutter, F. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Krishnapuram, B., Williams, D., Xue, Y., Carin, L., Figueiredo, M., and Hartemink, A. On semi-supervised classification. *Advances in neural information processing systems*, 17, 2004.

Kumar, A., Engstrom, L., Ilyas, A., and Tsipras, D. Understanding self-training for gradient-boosted trees. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 1651–1662, 2020.

Kwon, J. and Caramanis, C. The em algorithm gives sample-optimality for learning mixtures of well-separated gaussians. In *Conference on Learning Theory*, pp. 2425–2487. PMLR, 2020.

Laurent, B. and Massart, P. Adaptive estimation of a quadratic functional by model selection. *Annals of statistics*, pp. 1302–1338, 2000.

Lelarge, M. and Miolane, L. Asymptotic bayes risk for gaussian mixture in a semi-supervised setting. In *2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pp. 639–643. IEEE, 2019.

Li, Y., Ildiz, M. E., Papailiopoulos, D., and Oymak, S. Transformers as algorithms: Generalization and stability in in-context learning. In *International Conference on Machine Learning*, pp. 19565–19594. PMLR, 2023.

Li, Y., Rawat, A. S., and Oymak, S. Fine-grained analysis of in-context linear estimation: Data, architecture, and beyond. *Advances in Neural Information Processing Systems*, 37:138324–138364, 2024.

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

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

Neopane, O. Lecture notes on high-dimensional statistics. https://www.stat.cmu.edu/~arinaldo/Teaching/36709/S19/Scribed_Lectures/Feb26_Ojash.pdf, 2018.

Nigam, K., McCallum, A. K., Thrun, S., and Mitchell, T. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39(2–3):103–134, 2000. doi: 10.1023/A:1007692713085.

Oymak, S. and Gulcu, T. C. A theoretical characterization of semi-supervised learning with self-training for gaussian mixture models. In *International Conference on Artificial Intelligence and Statistics*, pp. 3601–3609. PMLR, 2021.

Ratsaby, J. and Venkatesh, S. S. Learning from a mixture of labeled and unlabeled examples with parametric side information. In *Proceedings of the Eighth Annual Conference on Computational Learning Theory (COLT '95)*, pp. 412–417. ACM, 1995. doi: 10.1145/225298.225348.

Shen, W., Zhou, R., Yang, J., and Shen, C. On the training convergence of transformers for in-context classification. *arXiv preprint arXiv:2410.11778*, 2024.

Snell, C., Lee, J., Xu, K., and Kumar, A. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Thrampoulidis, C., Oymak, S., and Soltanolkotabi, M. Theoretical insights into multiclass classification: A high-dimensional asymptotic view. *Advances in Neural Information Processing Systems*, 33:8907–8920, 2020.

Von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A., and Vladymyrov, M. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pp. 35151–35174. PMLR, 2023.

Wang, K. and Thrampoulidis, C. Binary classification of gaussian mixtures: Abundance of support vectors, benign overfitting, and regularization. *SIAM Journal on Mathematics of Data Science*, 4(1):260–284, 2022.

Wang, Y., Wu, Y., Wei, Z., Jegelka, S., and Wang, Y. A theoretical understanding of self-correction through in-context alignment. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Wei, C., Shen, K., Chen, Y., and Ma, T. Theoretical analysis of self-training with deep networks on unlabeled data. In *International Conference on Learning Representations (ICLR)*, 2021.

Wu, J., Zou, D., Chen, Z., Braverman, V., Gu, Q., and Bartlett, P. L. How many pretraining tasks are needed for in-context learning of linear regression? *arXiv preprint arXiv:2310.08391*, 2023.

Xie, S. M., Raghunathan, A., Liang, P., and Ma, T. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=RdJVFCHjUMI.

Xu, J., Li, Z., Mukherjee, S., and Taylor, D. Towards understanding deep learning with persistent homology. *arXiv preprint arXiv:2106.06718*, 2021.

Yang, T., Huang, Y., Liang, Y., and Chi, Y. In-context learning with representations: Contextual generalization of trained transformers. *arXiv preprint arXiv:2408.10147*, 2024.

Ye, H.-J., Liu, S.-Y., and Chao, W.-L. A closer look at tabpfn v2: Strength, limitation, and extension. *arXiv preprint arXiv:2502.17361*, 2025.

Zhang, R., Frei, S., and Bartlett, P. L. Trained transformers learn linear models in-context. *arXiv preprint arXiv:2306.09927*, 2023.

Zhang, R., Frei, S., and Bartlett, P. L. Trained transformers learn linear models in-context. *Journal of Machine Learning Research*, 25(49):1–55, 2024.

A. Related Work

Theoretical Analysis of In-Context Learning Recent work has developed theoretical frameworks for understanding in-context learning in transformers. Akyürek et al. (2023), Von Oswald et al. (2023) and Dai et al. (2023) demonstrated that transformers emulate gradient descent during ICL. Xie et al. (2022) offered a Bayesian perspective, while Zhang et al. (2023; 2024) showed transformers learn linear models in-context. Ahn et al. (2023) established they implement preconditioned gradient descent, and Mahankali et al. (2024) proved one-step gradient descent is optimal for single-layer linear attention. Multiple works (Li et al., 2023; Yang et al., 2024; Li et al., 2024; Bai et al., 2023; Shen et al., 2024) studied the generalization capability of transformers. However, these exclusively focus on fully-supervised settings, leaving a critical gap in understanding how transformers handle partially labeled data—a common real-world scenario. Our work addresses this gap by providing the first theoretical characterization of semi-supervised in-context learning. (Wang et al., 2024) considers a setting where the model observes demonstrations of the form (query, response$_i$, reward$_i$) and aims to correct its response based on the reward sequence. Our work has a different focus as it highlights that the model can correct/impute the missing labels using implicit feedback from labeled demonstrations.

Semi-Supervised Learning Traditional semi-supervised learning (SSL) aims to leverage unlabeled data to improve classifier performance. For linear classifiers, Oymak & Gulcu (2021) characterized self-training iterations and demonstrated rejecting low-confidence samples; further theoretical analyses of self-training/pseudo-labeling cover deep networks (Wei et al., 2021) and models like gradient-boosted trees (Kumar et al., 2020). For Gaussian Mixture Models (GMMs), Lelarge & Miolane (2019) quantified maximal improvement from unlabeled data, while Krishnapuram et al. (2004) developed graph-based priors. Learning GMMs via Expectation-Maximization (EM) or pseudo-labeling, especially with few labels, is well-studied. Ratsaby & Venkatesh (1995) provided early PAC-style bounds for GMMs learned from few labeled and many unlabeled points. Balakrishnan et al. (2017) offered further statistical guarantees for EM. Nigam et al. (2000) demonstrated empirically that EM (viewable as iterative pseudo-labeling (Fan et al., 2023)) with pseudo-labels significantly reduces text classification error using unlabeled documents. These foundational works, with ongoing research in areas like agnostic learning (Kwon & Caramanis, 2020) and evolving theories (Xu et al., 2021), underpin many SSL concepts. While these works established fundamental principles, they did not consider how these concepts apply to in-context learning with transformers. Our contribution bridges this gap by showing how transformer depth enables effective utilization of unlabeled examples within the prompt, essentially implementing semi-supervised learning without parameter updates.

B. Experiments

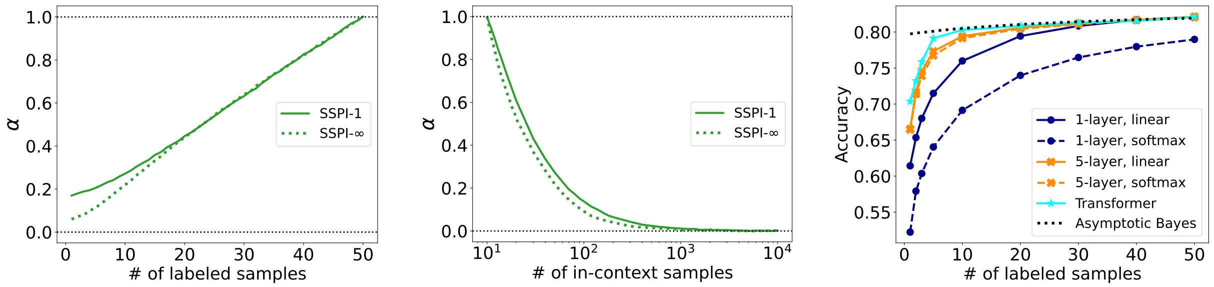
In Sections 3 and 4, we introduced Figure 1 and demonstrated its consistency with our theoretical results. In this section, we describe the experimental setup and implementation details. Additionally, we present further empirical findings to investigate additional questions of interest in Section B.1. Motivated by Proposition 4.2, which suggests that looping can help leverage unlabeled data, Section B.2 introduces an algorithm based on the TabPFN, showing how it can enhance prediction performance by incorporating a small amount of unlabeled data and iterative pseudo-labeling through model looping.

Experimental Setup Following Section 2, set $d = 10$ and noise level $\sigma = 1$. All models are trained using Adam optimizer with a learning rate of 10^{-3} for 40,000 epochs, with a batch size of 512. We use logistic loss in our experiments. Since our study focuses on the optimization landscape and model expressivity, and experiments are implemented via gradient descent, we repeat 10 trainings from random initialization and results are presented as the maximal test accuracy among those 10 trails.

B.1. Additional Observations

Exploration of Optimal α Values In Section 4, we introduced the SSPI-k estimator (cf. (SSPI-k)), but did not discuss the choice of the mixing parameter α, which plays a crucial role in balancing the contribution of the supervised estimator $\hat{\boldsymbol{\mu}}_s$. Specifically, α controls how much weight is given to the purely supervised signal. In the fully supervised case, the optimal choice is $\alpha = 1$, as $\hat{\boldsymbol{\mu}}_s$ corresponds to the optimal estimator.

In Figures 2(a) and 2(b), we empirically examine the optimal values of α. Given $\boldsymbol{\mu} \sim \mathrm{Unif}(\mathbb{S}^{d-1})$, we define the optimal α



(a) $n = 50, p \in (0, 1]$.

(b) $np = 10, n \in \{10, \cdots, 10000\}$.

(c) $n = 50, p \in (0, 1]$.

Figure 2. Additional experimental results. (a)&(b): Analysis of the optimal α values for the SSPI estimator (cf. (SSPI-k)) under varying (n, p, k). Green solid and dotted curves represent optimal α values for SSPI-1 and SSPI-∞, respectively. The SSPI results shown in Figure 1 use the corresponding α values from Figs. 2(a) and 2(b). (c): Comparison of different model architectures for the SS-ICL problem. Dark blue and orange curves show results for 1-layer and 5-layer attention models, with solid and dashed curves representing linear and softmax attention, respectively. Cyan curves correspond to 5-layer Transformers. The black dotted curve shows the asymptotic Bayes-optimal error (cf. (Lelarge & Miolane, 2019)). Results suggest the performance ordering: Transformer ¿ linear attention ¿ softmax attention. Further details are provided in Section B.

as the minimizer of the following cosine similarity-based objective:

$$\alpha^\star := \min_{\alpha \in [0,1]} \mathcal{L}(\alpha) \quad \text{where} \quad \mathcal{L}(\alpha) = 1 - \mathbb{E}[\texttt{cosine\_similarity}(\boldsymbol{\mu}_{ss\text{-}k}, \boldsymbol{\mu})].$$

For each setting, we optimize α using the Adam optimizer for 10,000 epochs with a batch size of 128 and a learning rate of 0.01. The results are shown in Figs 2(a) and 2(b).

In Figure 2(a), for both SSPI-1 and SSPI-∞, the optimal α starts near zero when the number of labeled examples is small, reflecting the limited utility of $\hat{\boldsymbol{\mu}}_s$ in low-supervision regimes. As the number of labeled samples increases, α grows approximately linearly and approaches 1 when the problem becomes fully supervised. In Figure 2(b), when $n = 10$ and $p = 1$ (i.e., all examples are labeled), the optimal α begins at 1. As n increases and the fraction of unlabeled data grows, α decreases significantly. This trend indicates that as the volume of unlabeled data increases, the SSPI estimator adaptively reduces reliance on the supervised component $\hat{\boldsymbol{\mu}}_s$ and increases reliance on the semi-supervised component, which leverages the structure of the unlabeled data through $\boldsymbol{X}^\top \boldsymbol{X}$.

Comparison Across Different Model Architectures Beyond linear attention, we investigate additional model architectures under our SS-ICL setting. The comparison results are presented in Fig. 2(c). The softmax attention model uses the same structure described in Section 2.2, with the only difference being the addition of a softmax operation in Eq. (3). The Transformer model introduces further nonlinearity and capacity by incorporating multi-layer perceptrons (MLPs) and layer normalization. The Transformer experiments are conducted with 5-layer models.

When comparing weaker models—such as 1-layer linear (dark blue solid) and softmax (dark blue dashed) attention—we observe that softmax attention consistently underperforms linear attention. Notably, softmax attention fails to match the performance of the optimal supervised estimator, even when all labels are observed (i.e., when the number of labeled samples equals $n = 50$). Furthermore, increasing the depth of softmax attention (orange dashed curve for 5-layer softmax) still does not surpass the performance of 5-layer linear attention (orange solid curve). Among all architectures, the Transformer achieves the best performance due to its increased model capacity and expressiveness. Compared with Fig. 1(a), where the orange and dark blue markers (linear attention) are identical, the Transformer significantly improves accuracy. This improvement highlights that SSPI, while effective, is not the optimal semi-supervised estimator. Although our semi-supervised setting assumes isotropic data, the characterization of its optimal algorithm remains an open and foundational problem for future exploration. In the figure, we also include the asymptotic Bayes-optimal curve (black dotted; derived from (Lelarge & Miolane, 2019)) . As the number of samples increases, the results from linear attention, softmax attention, and Transformer all converge toward this optimal curve. We attribute the initial performance gap, particularly at low values along x-axis (e.g., $np = 1$), to the scarcity of labeled data.

Algorithm 1 LoopTabFM: Looping Tabular FM with Soft Pseudo-labels and Risk-aware Updates

```
Require: Dataset $\mathcal{D}_{\text{lab}}, \mathcal{D}_{\text{unlab}}$, looping iterations $K$
function $\texttt{Looping}(\mathcal{D}_{\text{lab}}, \mathcal{D}_{\text{unlab}}, K)$
    $\texttt{FM}_0 \leftarrow$ TabPFN-v2$(\mathcal{D}_{\text{lab}})$ {$\texttt{FM}_k$ corresponds to model of Loop-$k$.}
    $\mathcal{D}_{\text{unlab}} \leftarrow \texttt{FM}_0(\mathcal{D}_{\text{unlab}})$ {Assign pseudo labels via $\hat{y}^{\text{soft}} \leftarrow \texttt{FM}_0(\boldsymbol{x} \in \mathcal{D}_{\text{unlab}})$.}
    $\texttt{FM}_{\text{best}} \leftarrow \texttt{FM}_0$
    $\mathcal{R}_{\text{val}} = \texttt{Val\_Risk}(\mathcal{D}_{\text{unlab}})$
    for Looping iteration $k = 1, \ldots, K$ do
        $\texttt{FM}_k \leftarrow$ TabPFN-v2$(\mathcal{D}_{\text{lab}} \cup \mathcal{D}_{\text{unlab}})$
        $\mathcal{D}_{\text{unlab}} \leftarrow \texttt{FM}_k(\mathcal{D}_{\text{unlab}})$ {Update pseudo labels via $\hat{y}^{\text{soft}} \leftarrow \texttt{FM}_k(\boldsymbol{x} \in \mathcal{D}_{\text{unlab}})$.}
        if $\texttt{Val\_Risk}(\mathcal{D}_{\text{unlab}}) < \mathcal{R}_{\text{val}}$ then
            $\texttt{FM}_{\text{best}} \leftarrow \texttt{FM}_k$
            $\mathcal{R}_{\text{val}} = \texttt{Val\_Risk}(\mathcal{D}_{\text{unlab}})$
        end if
    end for
    return $\texttt{FM}_{\text{best}}$
end function
function $\texttt{Val\_Risk}(\mathcal{D}_{\text{unlab}})$
    return $\frac{1}{|\mathcal{D}_{\text{unlab}}|}\sum_i \min\left(\left|\hat{y}_i^{\text{soft}} - 1\right|, \left|\hat{y}_i^{\text{soft}} + 1\right|\right)$ {$\hat{y}_{\text{soft}}$ corresponds to the assigned soft label for feature in $\mathcal{D}_{\text{unlab}}$.}
end function
```

B.2. Tabular Experiments

To further investigate how model looping (Proposition 4.2) can improve label prediction, we propose the LoopTabFM algorithm that addresses unlabeled data by iteratively assigning pseudo-labels, with its details outlined in Algorithm 1. Suppose that we are given labeled $\mathcal{D}_{\text{lab}}$ and unlabeled $\mathcal{D}_{\text{unlab}}$ datasets. The overall workflow of the algorithm proceeds as follows:

1. **Base Model:** Perform ICL using TabPFN on the labeled dataset $\mathcal{D}_{\text{lab}}$ and treat the resulting model as the base model (Loop-0). The corresponding test accuracies are reported in Table 1.

2. **Pseudo-Label Assignment:** Using the current model (e.g., Loop-k) to generate predictions for the unlabeled data $\mathcal{D}_{\text{unlab}}$. Assign soft pseudo-labels based on these predictions. Note that the model outputs are scalars (i.e., elements of $\mathbb{R}$) and can be interpreted as soft labels.

3. **Model Update:** Construct a new prompt by combining the labeled examples with their true labels and the unlabeled examples with their assigned soft pseudo-labels. Perform ICL using TabPFN on this combined prompt to obtain an updated model (Loop-$(k+1)$). Repeat this process from Step 2 until the maximum number of looping iterations is reached.

⋆ **Model Validation:** To improve the stability of the looping process, we introduce an additional validation step and retain the model with the lowest validation risk as the final (best) model. Specifically, after assigning soft pseudo-labels to the unlabeled data, i.e., $\mathcal{D}_{\text{unlab}} = \{(\boldsymbol{x}_i, \hat{y}_i^{\text{soft}})_{i=1}^n\}$, we compute the validation risk over these pseudo-labeled examples as follows:

$$\texttt{Val\_Risk}(\mathcal{D}_{\text{unlab}}) = \frac{1}{n}\sum_{i \in [n]} \min\left(\left|\hat{y}_i^{\text{soft}} - 1\right|, \left|\hat{y}_i^{\text{soft}} + 1\right|\right),$$

which penalizes predictions that deviate from confident binary labels ±1.

We evaluated the effectiveness of our proposed looping strategy by iteratively applying TabPFN-v2 on real-world binary classification benchmarks used in (Hollmann et al., 2025). The results are summarized in Table 1, where each entry represents an average over 100 random splits of the dataset, with 80% of the data used as the test set in each split.

For each experiment, we randomly sample 10 labeled and 10 unlabeled examples, ensuring that the labeled set includes at least one example from each class. As a baseline (Loop-0), we apply TabPFN-v2 using only the labeled data. The

Table 1. Comparison of test accuracy (%) between the baseline (Loop-0) and LoopTabFM (Algorithm 1) after 1 to 5 iterations using TabPFN-v2. Each result is averaged over 100 random trials. The highest test accuracy for each dataset is highlighted in bold. The final column reports the relative improvement (%) of Loop-5 over the baseline, computed as (Loop-5 − Loop-0)/Loop-0×100%. Positive signs indicate a performance improvement over the baseline, while negative signs indicate a performance drop.

OpenML ID	# of features	# of samples	Class imbalance	Loop-0	Loop-1	Loop-2	Loop-3	Loop-4	Loop-5	Rel. Imp. (%)
3	36	3196	1.09	58.62	58.63	58.45	58.69	**59.00**	58.97	0.60 (+)
31	20	1000	2.33	**66.18**	65.95	66.05	65.58	65.52	65.07	1.68 (−)
1049	37	1458	7.19	72.00	75.62	79.48	80.31	**81.49**	81.40	13.06 (+)
1067	21	2109	5.47	73.12	76.59	77.94	77.92	78.57	**78.60**	7.50 (+)
1464	4	748	3.20	60.46	63.96	70.20	71.29	**72.26**	72.18	19.38 (+)
1487	72	2534	14.84	82.54	87.67	88.57	88.27	**89.85**	89.56	8.51 (+)
1489	5	5404	2.41	66.40	67.62	**68.30**	68.14	68.21	68.18	2.69 (+)
1494	41	1055	1.96	62.24	63.05	64.62	65.94	**66.07**	66.05	6.12 (+)
40701	20	5000	6.07	66.45	70.65	75.99	**78.18**	78.00	77.70	16.93 (+)
40900	36	5100	67	**98.53**	98.41	98.39	98.39	98.27	98.26	0.28 (−)
40981	14	690	1.25	73.56	74.41	74.67	**74.99**	74.93	74.94	1.88 (+)
40983	5	4839	17.54	79.71	85.04	89.36	**92.94**	92.90	92.75	16.35 (+)
41143	144	2984	1	64.64	64.80	65.06	65.17	**65.29**	65.13	0.76 (+)
41144	259	3140	1.01	50.70	50.63	50.68	50.67	50.71	**50.77**	0.14 (+)
41145	308	5832	1	56.16	**56.28**	56.21	56.24	56.19	56.22	0.12 (+)
41146	20	5124	1	71.26	73.90	75.39	75.84	76.02	**77.07**	8.51 (+)
41156	48	4147	3.03	67.74	69.78	70.64	**71.82**	71.72	71.74	5.90 (+)
Average				68.84	70.76	72.35	72.96	**73.24**	73.21	6.35 (+)

corresponding test accuracies are reported in the "Loop-0" column of Table 1. We compare this to models updated through up to $k \leq 5$ iterations of pseudo-label update, with results shown in the "Loop-k" columns. The final column reports the relative improvement (Rel. Imp.) over the baseline. Our results demonstrate that the looping strategy can significantly improve test accuracy. For instance, on OpenML datasets 1049, 1464, 40701, and 40983, accuracy improves by more than 10% over the baseline using only 10 additional unlabeled samples. The last row of the table reports average performance across datasets, revealing that the majority of performance gains occur in the first two iterations. This observation aligns with our synthetic experiments using multi-layer models (Figure 1), where the improvement from 1-layer to 2-layer is substantially greater than the improvement from 2-layer to 5-layer. These findings highlight that explicitly looping the tabular foundation model to iteratively refine soft pseudo-labels of unlabeled data using only a few iterations can substantially enhance performance.

As shown and discussed, our LoopTabFM algorithm enhances model performance. However, this improvement is not consistent across all datasets. For example, performance drops on the OpenML datasets with IDs 31 and 40900. This may be attributed to factors such as noise levels in the raw data, class imbalance, or other dataset-specific characteristics. In contrast to our synthetic experimental setting, where the model is pretrained in a meta-learning fashion on the distribution of the given dataset, TabPFN is used as a general-purpose pretrained foundation model and applied directly to target datasets in a single-shot inference setting. Prior work (Ye et al., 2025) has also shown that TabPFN can be sensitive to input length, which may further affect performance consistency. Despite these limitations, our experiments with TabPFN offer an initial insight into how unlabeled data and iterative looping can be leveraged to improve predictive performance. These findings suggest promising future directions, such as designing data-aware looping algorithms that adapt to dataset-specific properties.

C. Discussion and Limitations

Our paper introduces a theoretical study of semisupervised in-context learning and characterizes how transformer, specifically linear attention, models can harness unlabeled data in their context window to make inference. We show that depth is crucial to go beyond supervised estimation and utilize unlabeled data, and the latter is achieved by constructing estimators of the form $\hat{\boldsymbol{\mu}} = \sum_{i=0}^{K} a_i (\boldsymbol{X}^\top \boldsymbol{X})^i \boldsymbol{X}^\top \boldsymbol{y}$. $\log K$ depth suffices to express a Kth order polynomial which is in line with our synthetic and real experiments that corroborate that mild amount of depth/looping already achieves most of the benefit. Our core theoretical results are limited to linear attention models and it is important to understand the capabilities of the full transformer architecture. Indeed, transformer (MLP+softmax) empirically outperforms a linear attention model with equal number of layers, well approximating the Bayes optimal semisupervised estimator. It would also be exciting to go beyond the classification setting and examine how self-generated CoT rationales, as in (Wu et al., 2023), can enhance ICL capabilities for tasks that require reasoning/autoregression. Additionally, our proposed LoopTabFM algorithm demonstrates that iteratively pseudo-labeling unlabeled data can indeed enhance predictive performance for tabular tasks. However, there remains significant potential for developing more intelligent, data-specific algorithms that more effectively leverage unlabeled data to further improve model performance.

D. Analysis of Single-layer Linear Attention

D.1. Supporting Lemmas

Recap the SPI estimator from (SPI). Given a semi-supervised dataset $(\boldsymbol{x}_i, y_i)_{i=1}^n$ as described in Section 2.1, let $\mathcal{I}$ denote the token indices set corresponding to the labeled demonstrations, that is, we have

$$y_i = \begin{cases} y_i^c, & i \in \mathcal{I} \\ 0, & otherwise. \end{cases} \tag{12}$$

Then, the SPI estimates the task mean via

$$\hat{\boldsymbol{\mu}}_s = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} y_i \boldsymbol{x}_i.$$

Let $\boldsymbol{W} \in \mathbb{R}^{d \times d}$ be the preconditioning matrix. We define the following objective:

$$\boldsymbol{W}^\star := \arg\min_{\boldsymbol{W} \in \mathbb{R}^{d\times d}} \tilde{\mathcal{L}}(\boldsymbol{W}) \quad \text{where} \quad \tilde{\mathcal{L}}(\boldsymbol{W}) = \mathbb{E}\left[\left(\boldsymbol{x}^\top \boldsymbol{W} \sum_{i \in \mathcal{I}} y_i \boldsymbol{x}_i - y\right)^2\right]. \tag{13}$$

Here, we set $(\boldsymbol{x}, y)$ to be the query feature and its corresponding true label. The expectation subsumes the randomness in $(\boldsymbol{x}_i, y_i), (\boldsymbol{x}, y)$ as described in Section 2.1.

In the following, we provide a lemma that establishes equivalence between optimizing $\mathcal{L}_{\text{att-1}}(\mathcal{W}^{(1)}, \boldsymbol{h})$ (cf. (5) and choosing $L = 1$) and $\tilde{\mathcal{L}}(\boldsymbol{W})$.

Lemma D.1. *Consider ICL problem described in Section 2.2 with prompt defined in (2). Consider training a single-layer linear attention with squared loss, that is, $L = 1$ and $\ell(y, \hat{y}) = (y - \hat{y})^2$. Recall the objectives from (5) and (13), and let $\mathcal{L}^\star_{\textit{att-1}}$ and $\tilde{\mathcal{L}}^\star := \tilde{\mathcal{L}}(\boldsymbol{W}^\star)$ be their corresponding optimal losses where $\boldsymbol{W}^\star$ is defined in (13). Then, we have*

$$\mathcal{L}^\star_{\textit{att-1}} = \tilde{\mathcal{L}}^\star. \tag{14}$$

Additionally, let $f^\star_{\textit{att-1}} : \mathbb{R}^{(n+1)\times(d+1)} \to \mathbb{R}$ denote the optimal prediction (associated with the optimal loss $\mathcal{L}^\star_{\textit{att-1}}$). We have that $f^\star_{\textit{att-1}}$ is unique and for any prompt $\boldsymbol{Z}$ (cf. (2))

$$f^\star_{\textit{att-1}}(\boldsymbol{Z}) = \boldsymbol{x}^\top \boldsymbol{W}^\star \sum_{i \in \mathcal{I}} y_i \boldsymbol{x}_i. \tag{15}$$

Proof. Recap the single-layer linear attention model and its prediction from (3) and (4). We have

$$f_{\text{att-1}}(\boldsymbol{Z}) = \boldsymbol{h}^\top \text{att}(\boldsymbol{Z}; \mathcal{W})_{[n+1]} \quad \text{where} \quad \text{att}(\boldsymbol{Z}; \mathcal{W}) = (\boldsymbol{Z}\boldsymbol{W}_q \boldsymbol{W}_k^\top \boldsymbol{Z}^\top)\boldsymbol{M}\boldsymbol{Z}\boldsymbol{W}_v \tag{16}$$

with $\mathcal{W} := \{\boldsymbol{W}_q, \boldsymbol{W}_k, \boldsymbol{W}_v\}$ being the set of the query, key and value matrices of the attention. Since $\mathcal{W}$ and $\boldsymbol{h}$ are tunable parameters, without loss of generality and for simplicity, let

$$\boldsymbol{W} := \boldsymbol{W}_q \boldsymbol{W}_k^\top \quad \text{and} \quad \bar{\boldsymbol{h}} := \boldsymbol{W}_v \boldsymbol{h}.$$

Following the proof of Li et al., 2024, Proposition 1, similarly, we denote

$$\boldsymbol{W} = \begin{bmatrix} \bar{\boldsymbol{W}} & \boldsymbol{w}_1 \\ \boldsymbol{w}_2^\top & w \end{bmatrix} \quad \text{and} \quad \bar{\boldsymbol{h}} = \begin{bmatrix} \boldsymbol{h}_1 \\ h \end{bmatrix},$$

where $\bar{\boldsymbol{W}} \in \mathbb{R}^{d\times d}$, $\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{h}_1 \in \mathbb{R}^d$, and $w, h \in \mathbb{R}$.

Additionally, let $\mathcal{I}$ denote the token indices set corresponding to the labeled demonstrations (cf. (12)). Recall the prompt $\boldsymbol{Z}$ from (2), and $\boldsymbol{X} = [\boldsymbol{x}_1 \ \cdots \ \boldsymbol{x}_n]^\top \in \mathbb{R}^{n \times d}$ and $\boldsymbol{y} = [y_1 \ \cdots \ y_n]^\top \in \mathbb{R}^n$ from (8). Then we get

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_n & \boldsymbol{x} \\ y_1 & y_2 & \cdots & y_n & 0 \end{bmatrix}^\top = \begin{bmatrix} \boldsymbol{X}^\top & \boldsymbol{x} \\ \boldsymbol{y}^\top & 0 \end{bmatrix}^\top \in \mathbb{R}^{(n+1)\times(d+1)}. \tag{17}$$

Combining (16) and (17) together, we can rewrite the one-layer linear prediction as

$$\begin{aligned} f_{\text{att-1}}(\boldsymbol{Z}) &= [\boldsymbol{x}^\top \ 0] \boldsymbol{W} \boldsymbol{Z}^\top \boldsymbol{M} \boldsymbol{Z} \bar{\boldsymbol{h}} \\ &= [\boldsymbol{x}^\top \ 0] \begin{bmatrix} \bar{\boldsymbol{W}} & \boldsymbol{w}_1 \\ \boldsymbol{w}_2^\top & w \end{bmatrix} \begin{bmatrix} \boldsymbol{X}^\top & \boldsymbol{x} \\ \boldsymbol{y}^\top & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_n & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{X}^\top & \boldsymbol{x} \\ \boldsymbol{y}^\top & 0 \end{bmatrix}^\top \begin{bmatrix} \boldsymbol{h}_1 \\ h \end{bmatrix} \\ &= [\boldsymbol{x}^\top \bar{\boldsymbol{W}} \ \ \boldsymbol{x}^\top \boldsymbol{w}_1] \begin{bmatrix} \boldsymbol{X}^\top \boldsymbol{X} & \boldsymbol{X}^\top \boldsymbol{y} \\ \boldsymbol{y}^\top \boldsymbol{X} & \boldsymbol{y}^\top \boldsymbol{y} \end{bmatrix} \begin{bmatrix} \boldsymbol{h}_1 \\ h \end{bmatrix} \\ &= [\boldsymbol{x}^\top \bar{\boldsymbol{W}} \ \ \boldsymbol{x}^\top \boldsymbol{w}_1] \begin{bmatrix} \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{h}_1 + h \boldsymbol{X}^\top \boldsymbol{y} \\ \boldsymbol{y}^\top \boldsymbol{X} \boldsymbol{h}_1 + h \boldsymbol{y}^\top \boldsymbol{y} \end{bmatrix} \\ &= \boldsymbol{x}^\top \bar{\boldsymbol{W}} (\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{h}_1 + h \boldsymbol{X}^\top \boldsymbol{y}) + \boldsymbol{x}^\top \boldsymbol{w}_1 (\boldsymbol{y}^\top \boldsymbol{X} \boldsymbol{h}_1 + h \boldsymbol{y}^\top \boldsymbol{y}) \\ &= \boldsymbol{x}^\top (h \bar{\boldsymbol{W}} + \boldsymbol{w}_1 \boldsymbol{h}_1^\top) \boldsymbol{X}^\top \boldsymbol{y} + \boldsymbol{x}^\top (\bar{\boldsymbol{W}} \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{h}_1 + h \boldsymbol{y}^\top \boldsymbol{y} \boldsymbol{w}_1) \\ &= \boldsymbol{x}^\top \tilde{\boldsymbol{W}} \boldsymbol{X}^\top \boldsymbol{y} + \boldsymbol{x}^\top (\bar{\boldsymbol{W}} \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{h}_1 + m h \boldsymbol{w}_1) \end{aligned}$$

where $\tilde{\boldsymbol{W}} := h \bar{\boldsymbol{W}} + \boldsymbol{w}_1 \boldsymbol{h}_1^\top$ and we define $m := |\mathcal{I}|$.

Next, recall the loss from (5) and consider the squared loss function, $\ell(y, \hat{y}) = (y - \hat{y})^2$. We have

$$\begin{aligned} \mathcal{L}_{\text{att-1}}(\mathcal{W}^{(1)}, \boldsymbol{h}) &= \mathbb{E}\left[(f_{\text{att-1}}(\boldsymbol{Z}) - y)^2\right] \\ &= \mathbb{E}\left[\left(\boldsymbol{x}^\top \tilde{\boldsymbol{W}} \boldsymbol{X} \boldsymbol{y} + \boldsymbol{x}^\top \left(\bar{\boldsymbol{W}} \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{h}_1 + m h \boldsymbol{w}_1\right) - y\right)^2\right] \\ &= \mathbb{E}\left[\left(y \boldsymbol{x}^\top \tilde{\boldsymbol{W}} \boldsymbol{X} \boldsymbol{y} + y \boldsymbol{x}^\top \left(\bar{\boldsymbol{W}} \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{h}_1 + m h \boldsymbol{w}_1\right) - 1\right)^2\right]. \end{aligned}$$

For simplicity and without loss of generality, we omit y and use $\boldsymbol{x}$ to represent $y\boldsymbol{x}$. Note that the distribution of (updated) $\boldsymbol{x}$ is not conditioned on its class and given mean vector $\boldsymbol{\mu}$, it follows $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 \boldsymbol{I})$. Similarly, let $\boldsymbol{x}_i$ represent $y_i^c \boldsymbol{x}_i$. We can then write

$$\begin{aligned} \mathcal{L}_{\text{att-1}}(\mathcal{W}^{(1)}, \boldsymbol{h}) &= \mathbb{E}\left[\left(\boldsymbol{x}^\top \tilde{\boldsymbol{W}} \sum_{i \in \mathcal{I}} \boldsymbol{x}_i + \boldsymbol{x}^\top \left(\bar{\boldsymbol{W}} \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{h}_1 + m h \boldsymbol{w}_1\right) - 1\right)^2\right] \\ &= \mathbb{E}\left[\left(\boldsymbol{x}^\top \tilde{\boldsymbol{W}} \sum_{i \in \mathcal{I}} \boldsymbol{x}_i - 1\right)^2\right] + \mathbb{E}\left[\left(\boldsymbol{x}^\top \left(\bar{\boldsymbol{W}} \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{h}_1 + m h \boldsymbol{w}_1\right)\right)^2\right] \\ &\quad + 2\,\mathbb{E}\left[\left(\boldsymbol{x}^\top \tilde{\boldsymbol{W}} \sum_{i \in \mathcal{I}} \boldsymbol{x}_i - 1\right) \left(\boldsymbol{x}^\top \left(\bar{\boldsymbol{W}} \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{h}_1 + m h \boldsymbol{w}_1\right)\right)\right]. \end{aligned} \tag{18}$$

We start with showing that for any given parameters $\boldsymbol{W} \in \mathbb{R}^{(d+1)\times(d+1)}, \boldsymbol{h} \in \mathbb{R}^{d+1}$, the component $\mathbb{E}[(\boldsymbol{x}^\top \tilde{\boldsymbol{W}} \sum_{i \in \mathcal{I}} \boldsymbol{x}_i - 1)(\boldsymbol{x}^\top (\bar{\boldsymbol{W}} \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{h}_1 + m h \boldsymbol{w}_1))] = 0$. To prove it, we first expand

$$\begin{aligned} &(\boldsymbol{x}^\top \tilde{\boldsymbol{W}} \sum_{i \in \mathcal{I}} \boldsymbol{x}_i - 1)(\boldsymbol{x}^\top (\bar{\boldsymbol{W}} \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{h}_1 + m h \boldsymbol{w}_1)) \\ &= \underbrace{(\boldsymbol{x}^\top \tilde{\boldsymbol{W}} \sum_{i \in \mathcal{I}} \boldsymbol{x}_i)(\boldsymbol{x}^\top \bar{\boldsymbol{W}} \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{h}_1)}_{(a)} - \underbrace{\boldsymbol{x}^\top \bar{\boldsymbol{W}} \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{h}_1}_{(b)} + \underbrace{(\boldsymbol{x}^\top \tilde{\boldsymbol{W}} \sum_{i \in \mathcal{I}} \boldsymbol{x}_i)(m h \boldsymbol{x}^\top \boldsymbol{w}_1)}_{(c)} - \underbrace{m h \boldsymbol{x}^\top \boldsymbol{w}_1}_{(d)}. \end{aligned}$$

In the following, we consider the expectations of $(a), (b), (c), (d)$ sequentially, all of which take the value zero. First note that since $\boldsymbol{\mu} \sim \text{Unif}(\mathbb{S}^{d-1})$ and $(\boldsymbol{\xi}_i)_{i=1}^n, \boldsymbol{\xi} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$, the odd moments of $\boldsymbol{\mu}$, $\boldsymbol{\xi}$ and $\boldsymbol{\xi}_i, i \in [n]$ are all zeros.

$$
\begin{aligned}
(a): \quad & \mathbb{E}\left[(\boldsymbol{x}^\top \tilde{\boldsymbol{W}} \sum_{i \in \mathcal{I}} \boldsymbol{x}_i)(\boldsymbol{x}^\top \bar{\boldsymbol{W}} \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{h}_1)\right] \\
& = \mathbb{E}\left[(\boldsymbol{\mu} + \boldsymbol{\xi})^\top \tilde{\boldsymbol{W}} \sum_{i \in \mathcal{I}} (\boldsymbol{\mu} + \boldsymbol{\xi}_i)(\boldsymbol{\mu} + \boldsymbol{\xi})^\top \bar{\boldsymbol{W}} \sum_{i \in [n]} (\boldsymbol{\mu} + \boldsymbol{\xi}_i)(\boldsymbol{\mu} + \boldsymbol{\xi}_i)^\top \boldsymbol{h}_1\right] \\
& = \sum_{i \in \mathcal{I}} \sum_{j \in [n]} \mathbb{E}\left[(\boldsymbol{\mu} + \boldsymbol{\xi})^\top \tilde{\boldsymbol{W}} (\boldsymbol{\mu} + \boldsymbol{\xi}_i)(\boldsymbol{\mu} + \boldsymbol{\xi})^\top \bar{\boldsymbol{W}} (\boldsymbol{\mu} + \boldsymbol{\xi}_j)(\boldsymbol{\mu} + \boldsymbol{\xi}_j)^\top \boldsymbol{h}_1\right] \\
& = \sum_{i \in \mathcal{I}} \sum_{j \in [n]} \mathbb{E}\left[\boldsymbol{\mu}^\top \tilde{\boldsymbol{W}} \boldsymbol{\mu} \boldsymbol{\mu}^\top \bar{\boldsymbol{W}} (\boldsymbol{\mu}\boldsymbol{\mu}^\top + \boldsymbol{\xi}_j \boldsymbol{\xi}_j^\top) \boldsymbol{h}_1 + \boldsymbol{\xi}^\top \tilde{\boldsymbol{W}} \boldsymbol{\mu} \boldsymbol{\xi}^\top \bar{\boldsymbol{W}} (\boldsymbol{\mu}\boldsymbol{\mu}^\top + \boldsymbol{\xi}_j \boldsymbol{\xi}_j^\top) \boldsymbol{h}_1\right] \\
& = 0,
\end{aligned}
$$

$$
\begin{aligned}
(b): \quad & \mathbb{E}\left[\boldsymbol{x}^\top \bar{\boldsymbol{W}} \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{h}_1\right] \\
& = \mathbb{E}\left[(\boldsymbol{\mu} + \boldsymbol{\xi})^\top \bar{\boldsymbol{W}} \sum_{i \in [n]} (\boldsymbol{\mu} + \boldsymbol{\xi}_i)(\boldsymbol{\mu} + \boldsymbol{\xi}_i)^\top \boldsymbol{h}_1\right] \\
& = \mathbb{E}\left[\boldsymbol{\mu}^\top \bar{\boldsymbol{W}} \sum_{i \in [n]} (\boldsymbol{\mu}\boldsymbol{\mu}^\top + \boldsymbol{\xi}_i \boldsymbol{\xi}_i^\top) \boldsymbol{h}_1\right] \\
& = 0,
\end{aligned}
$$

$$
\begin{aligned}
(c): \quad & \mathbb{E}\left[(\boldsymbol{x}^\top \tilde{\boldsymbol{W}} \sum_{i \in \mathcal{I}} \boldsymbol{x}_i)(mh \boldsymbol{x}^\top \boldsymbol{w}_1)\right] \\
& = mh \, \mathbb{E}\left[(\boldsymbol{\mu} + \boldsymbol{\xi})^\top \tilde{\boldsymbol{W}} \sum_{i \in \mathcal{I}} (\boldsymbol{\mu} + \boldsymbol{\xi}_i)(\boldsymbol{\mu} + \boldsymbol{\xi})^\top \boldsymbol{w}_1\right] \\
& = mh \sum_{i \in \mathcal{I}} \mathbb{E}\left[(\boldsymbol{\mu} + \boldsymbol{\xi})^\top \tilde{\boldsymbol{W}} \boldsymbol{\mu} (\boldsymbol{\mu} + \boldsymbol{\xi})^\top \boldsymbol{w}_1\right] \\
& = mh \sum_{i \in \mathcal{I}} \mathbb{E}\left[\boldsymbol{\mu}^\top \tilde{\boldsymbol{W}} \boldsymbol{\mu} \boldsymbol{\mu}^\top \boldsymbol{w}_1 + \boldsymbol{\xi}^\top \tilde{\boldsymbol{W}} \boldsymbol{\mu} \boldsymbol{\xi}^\top \boldsymbol{w}_1\right] \\
& = 0,
\end{aligned}
$$

$$
(d): \quad \mathbb{E}\left[mh \boldsymbol{x}^\top \boldsymbol{w}_1\right] = 0.
$$

Therefore, loss in (18) returns

$$
\mathcal{L}_{\text{att-1}}(\mathcal{W}^{(1)}, \boldsymbol{h}) = \underbrace{\mathbb{E}\left[\left(\boldsymbol{x}^\top \tilde{\boldsymbol{W}} \sum_{i \in \mathcal{I}} \boldsymbol{x}_i - 1\right)^2\right]}_{\tilde{\mathcal{L}}(\tilde{\boldsymbol{W}})} + \mathbb{E}\left[\left(\boldsymbol{x}^\top (\bar{\boldsymbol{W}} \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{h}_1 + mh \boldsymbol{w}_1)\right)^2\right].
$$

Here, the first term $\mathbb{E}[(\boldsymbol{x}^\top \tilde{\boldsymbol{W}} \sum_{i \in \mathcal{I}} \boldsymbol{x}_i - 1)^2] = \tilde{\mathcal{L}}(\tilde{\boldsymbol{W}})$ where $\tilde{\mathcal{L}}(\tilde{\boldsymbol{W}})$ is defined in (13).

Recall that $\tilde{\boldsymbol{W}} = h\bar{\boldsymbol{W}} + \boldsymbol{w}_1\boldsymbol{h}_1^\top$. Then for any $\tilde{\boldsymbol{W}} \in \mathbb{R}^{d\times d}$, setting $\boldsymbol{h}_1 = \boldsymbol{w}_1 = \boldsymbol{0}_d$ and $h = 1$ returns $\mathbb{E}\left[\left(\boldsymbol{x}^\top\left(\bar{\boldsymbol{W}}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{h}_1 + mh\boldsymbol{w}_1\right)\right)^2\right] = 0$, and then

$$\mathcal{L}_{\text{att-1}}(\mathcal{W}^{(1)}, \boldsymbol{h}) = \mathbb{E}\left[\left(\boldsymbol{x}^\top\bar{\boldsymbol{W}}\sum_{i\in\mathcal{I}}\boldsymbol{x}_i - 1\right)^2\right]$$

Therefore, optimizing $\mathcal{L}_{\text{att-1}}(\mathcal{W}^{(1)}, \boldsymbol{h})$ returns the same minima as optimizing $\tilde{\mathcal{L}}(\boldsymbol{W})$, which completes the proof of (14). Note that optimal loss $\mathcal{L}^\star_{\text{att-1}}$ depends on the labeled data $i \in \mathcal{I}$ only.

Furthermore, since $\tilde{\mathcal{L}}(\boldsymbol{W})$ is strongly convex (see (19)), $\boldsymbol{W}^\star$ exists and is unique. Therefore, (14) and uniqueness of $\boldsymbol{W}^\star$ leads to the conclusion (15). □

Lemma D.2. *Consider the objective defined in* (13) *with semi-supervised data following Section* 2. *Then the optimal solution* $\boldsymbol{W}^\star$ *satisfies*

$$\boldsymbol{W}^\star = c\boldsymbol{I}$$

for some $c > 0$.

Proof. Recap the Objective (13) and its optimal solution $\boldsymbol{W}^\star$. Let $\mathcal{I}$ be the index set corresponding the labeled in-context examples, and $|\mathcal{I}| = m$. Note that, m is also a random variable, independent of $\boldsymbol{x}_i, y_i^c, \boldsymbol{x}, y$.

As in the proof of Lemma D.1, we use $\boldsymbol{x}$ to represent $y\boldsymbol{x}$ and $\boldsymbol{x}_i$ to represent $y_i^c\boldsymbol{x}_i$ for simplicity, where (updated) $\boldsymbol{x}_i, \boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2\boldsymbol{I})$. Letting $\boldsymbol{\xi}', \boldsymbol{\xi}, \boldsymbol{\xi}_i \sim \mathcal{N}(0, \sigma^2\boldsymbol{I})$ be independent, we obtain

$$\begin{aligned}
\tilde{\mathcal{L}}(\boldsymbol{W}) &= \mathbb{E}\left[(\boldsymbol{x}^\top\boldsymbol{W}\sum_{i\in\mathcal{I}}\boldsymbol{x}_i - 1)^2\right] \qquad (19)\\
&= \mathbb{E}\left[((\boldsymbol{\mu}+\boldsymbol{\xi})^\top\boldsymbol{W}\sum_{i\in\mathcal{I}}(\boldsymbol{\mu}+\boldsymbol{\xi}_i) - 1)^2\right]\\
&= \mathbb{E}\left[((\boldsymbol{\mu}+\boldsymbol{\xi})^\top\boldsymbol{W}(m\boldsymbol{\mu}+\sqrt{m}\boldsymbol{\xi}') - 1)^2\right]\\
&= \mathbb{E}\left[m^2(\boldsymbol{\mu}^\top\boldsymbol{W}\boldsymbol{\mu})^2 + m(\boldsymbol{\mu}^\top\boldsymbol{W}\boldsymbol{\xi}')^2 + m^2(\boldsymbol{\xi}^\top\boldsymbol{W}\boldsymbol{\mu})^2 + m(\boldsymbol{\xi}^\top\boldsymbol{W}\boldsymbol{\xi}')^2 + 1\right] - 2\,\mathbb{E}\left[m\boldsymbol{\mu}^\top\boldsymbol{W}\boldsymbol{\mu}\right]\\
&= \frac{\mathbb{E}[m^2]}{d(d+2)}(\mathtt{tr}(\boldsymbol{W})^2 + \mathtt{tr}(\boldsymbol{W}\boldsymbol{W}^\top) + \mathtt{tr}(\boldsymbol{W}^2)) + \frac{\mathbb{E}[m+m^2]}{d}\sigma^2\mathtt{tr}(\boldsymbol{W}\boldsymbol{W}^\top)\\
&\quad + \mathbb{E}[m]\sigma^4\mathtt{tr}(\boldsymbol{W}\boldsymbol{W}^\top) + 1 - \frac{2\,\mathbb{E}[m]}{d}\mathtt{tr}(\boldsymbol{W}).
\end{aligned}$$

Differentiating it results in

$$\nabla_{\boldsymbol{W}}\tilde{\mathcal{L}}(\boldsymbol{W}) = \frac{2\,\mathbb{E}[m^2]}{d(d+2)}(\mathtt{tr}(\boldsymbol{W})\boldsymbol{I} + \boldsymbol{W} + \boldsymbol{W}^\top) + \frac{2\,\mathbb{E}[m+m^2]\sigma^2}{d}\boldsymbol{W} + 2\,\mathbb{E}[m]\sigma^4\boldsymbol{W} - \frac{2\,\mathbb{E}[m]}{d}\boldsymbol{I}.$$

Setting $\nabla_{\boldsymbol{W}}\tilde{\mathcal{L}}(\boldsymbol{W}) = 0$, we obtain the optimal $\boldsymbol{W}^\star$

$$\boldsymbol{W}^\star = \frac{1}{(1+\sigma^2)\,\mathbb{E}[m^2]/\,\mathbb{E}[m] + \sigma^2 + \sigma^4 d}\boldsymbol{I},$$

which leads to the conclusion that $\boldsymbol{W}^\star = c\boldsymbol{I}$, for $c = \frac{1}{(1+\sigma^2)\,\mathbb{E}[m^2]/\,\mathbb{E}[m]+\sigma^2+\sigma^4 d} > 0$. It completes the proof. □

D.2. Proof of Theorem 3.1

Proof. Note that (6) can be easily proven using Lemmas D.1 and D.2. Then, we focus on proving (7).

Given that (6) holds, we can rewrite its classification error as

$$\mathbb{P}(y^\star_{\text{att-1}}(\boldsymbol{Z}) \neq y) = \mathbb{P}(\text{sgn}(\boldsymbol{x}^\top\hat{\boldsymbol{\mu}}_s) \neq y) = \mathbb{P}(\text{sgn}(y\boldsymbol{x}^\top\hat{\boldsymbol{\mu}}_s) \neq 1) \qquad (20)$$

where $\hat{\boldsymbol{\mu}}_s = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} y_i \boldsymbol{x}_i$ defined in (SPI) and $\mathcal{I}$ is the index set of labeled samples. Let $m = |\mathcal{I}|$.

Recall from Section 2.1 where $\boldsymbol{x} \sim \mathcal{N}(y\cdot\boldsymbol{\mu}, \sigma^2\boldsymbol{I})$. We can rewrite

$$y\boldsymbol{x} = \boldsymbol{\mu} + \sigma\boldsymbol{g}_1 \quad \text{where} \quad \boldsymbol{g}_1 \sim \mathcal{N}(0, \boldsymbol{I}).$$

Then for any given $\boldsymbol{\mu}, \hat{\boldsymbol{\mu}}_s$, we get

$$\begin{aligned}\mathbb{P}\left(\operatorname{sgn}(y\boldsymbol{x}^\top\hat{\boldsymbol{\mu}}_s) \neq 1 \mid \boldsymbol{\mu}, \hat{\boldsymbol{\mu}}_s\right) &= \mathbb{P}\left(\left(\boldsymbol{\mu} + \sigma\boldsymbol{g}_1\right)^\top \hat{\boldsymbol{\mu}}_s < 0 \mid \boldsymbol{\mu}, \hat{\boldsymbol{\mu}}_s\right) \\ &= \mathbb{P}\left(\boldsymbol{\mu}^\top\hat{\boldsymbol{\mu}}_s < \sigma\boldsymbol{g}_1^\top\hat{\boldsymbol{\mu}}_s \mid \boldsymbol{\mu}, \hat{\boldsymbol{\mu}}_s\right) \\ &= Q\left(\frac{\boldsymbol{\mu}^\top\hat{\boldsymbol{\mu}}_s}{\sigma\left\|\hat{\boldsymbol{\mu}}_s\right\|_{\ell_2}}\right). \end{aligned} \tag{21}$$

Here Q-function is the tail distribution function of the standard normal distribution.

Next, similarly, given that $\boldsymbol{x}_i \sim \mathcal{N}(y_i\cdot\boldsymbol{\mu}, \sigma^2\boldsymbol{I})$ for $i \in \mathcal{I}$, we can rewrite

$$\hat{\boldsymbol{\mu}}_s = \frac{1}{m}\sum_{i\in\mathcal{I}} y_i\boldsymbol{x}_i = \boldsymbol{\mu} + \frac{\sigma}{\sqrt{m}}\boldsymbol{g}_2 \quad \text{where} \quad \boldsymbol{g}_2 \sim \mathcal{N}(0, \boldsymbol{I}).$$

Then combining (20) and (21), we have

$$\begin{aligned}\mathbb{P}(y^\star_{\text{att-1}}(\boldsymbol{Z}) \neq y) &= \mathbb{E}_{\boldsymbol{\mu},\boldsymbol{g}_2}\left[Q\left(\frac{\boldsymbol{\mu}^\top\hat{\boldsymbol{\mu}}_s}{\sigma\left\|\hat{\boldsymbol{\mu}}_s\right\|_{\ell_2}}\right)\right] \\ &= \mathbb{E}_{\boldsymbol{\mu},\boldsymbol{g}_2}\left[Q\left(\frac{\boldsymbol{\mu}^\top(\boldsymbol{\mu} + \frac{\sigma}{\sqrt{m}}\boldsymbol{g}_2)}{\sigma\left\|\boldsymbol{\mu} + \frac{\sigma}{\sqrt{m}}\boldsymbol{g}_2\right\|_{\ell_2}}\right)\right] \\ &= \mathbb{E}_{\boldsymbol{\mu},\boldsymbol{g}_2}\left[Q\left(\frac{1 + \frac{\sigma}{\sqrt{m}}\boldsymbol{\mu}^\top\boldsymbol{g}_2}{\sigma\sqrt{1 + 2\frac{\sigma}{\sqrt{m}}\boldsymbol{\mu}^\top\boldsymbol{g}_2 + \frac{\sigma^2}{m}\left\|\boldsymbol{g}_2\right\|_{\ell_2}^2}}\right)\right]. \end{aligned}$$

Note that for any $\boldsymbol{\mu}$ with $\|\boldsymbol{\mu}\|_{\ell_2} = 1$, we have $\boldsymbol{\mu}^\top\boldsymbol{g}_2 \sim \mathcal{N}(0,1)$. Therefore, we can write

$$\boldsymbol{\mu}^\top\boldsymbol{g}_2 = g \quad \text{where} \quad g \sim \mathcal{N}(0,1),$$

and let $\boldsymbol{U} \in \mathbb{R}^{d\times d}$ be a unitary matrix with first row being $\boldsymbol{\mu}$. We can write

$$\|\boldsymbol{g}_2\|_{\ell_2}^2 = \|\boldsymbol{U}\boldsymbol{g}_2\|_{\ell_2}^2 = g^2 + h \quad \text{where} \quad h \sim \mathcal{X}^2_{d-1}.$$

Here, $\mathcal{X}^2_{d-1}$ denotes chi-squared distribution with $(d-1)$ degrees of freedom. Then, we get

$$\begin{aligned}\mathbb{P}(y^\star_{\text{att-1}}(\boldsymbol{Z}) \neq y) &= \mathbb{E}_{g,h}\left[Q\left(\frac{1 + \frac{\sigma}{\sqrt{m}}g}{\sigma\sqrt{1 + 2\frac{\sigma}{\sqrt{m}}g + \frac{\sigma^2}{m}(g^2 + h)}}\right)\right] \\ &= \mathbb{E}_{g,h}\left[Q\left(\frac{1 + \frac{\sigma}{\sqrt{m}}g}{\sigma\sqrt{(1 + \frac{\sigma}{\sqrt{m}}g)^2 + \frac{\sigma^2}{m}h}}\right)\right], \\ &= \mathbb{E}_{g,h}\left[Q\left(\frac{1 + \varepsilon_\sigma g}{\sigma\sqrt{(1 + \varepsilon_\sigma g)^2 + \varepsilon_\sigma^2 h}}\right)\right], \end{aligned}$$

where $\varepsilon_\sigma := \sigma/\sqrt{m}$. It completes the proof of (7).

Next, we derive an upper bound for $\mathbb{P}(y^\star_{\text{att-1}}(\boldsymbol{Z}) \neq y)$. Let $c := \varepsilon_\sigma^{-1}$. Then we have

$$\begin{aligned}
\mathbb{P}(y^\star_{\text{att-1}}(\boldsymbol{Z}) \neq y) &= \mathbb{E}_{g,h}\left[Q\left(\frac{c+g}{\sigma\sqrt{(c+g)^2+h}}\right)\right] \\
&= \mathbb{E}_{g\geq -\frac{c}{2},h}\left[Q\left(\frac{c+g}{\sigma\sqrt{(c+g)^2+h}}\right)\right] + \mathbb{E}_{g< -\frac{c}{2},h}\left[Q\left(\frac{c+g}{\sigma\sqrt{(c+g)^2+h}}\right)\right] \\
&\leq \mathbb{E}_{g\geq -\frac{c}{2},h}\left[Q\left(\frac{c+g}{\sigma\sqrt{(c+g)^2+h}}\right)\right] + Q(c/2) \\
&= \mathbb{E}_{g\geq -\frac{c}{2},h}\left[Q\left(\frac{1}{\sigma\sqrt{1+h/(c+g)^2}}\right)\right] + Q(c/2), \qquad (22)
\end{aligned}$$

where the inequality comes from the fact that $\mathbb{P}(g \leq -c/2) = Q(c/2)$ and $Q(x) \leq 1$ for any $x \in \mathbb{R}$. Next, we have

$$\frac{1}{\sqrt{1+h/(c+g)^2}} \geq 1 - \frac{1}{2}\frac{h}{(c+g)^2} \geq 1 - \frac{2h}{c^2}.$$

Here the first inequality comes from that $\frac{1}{\sqrt{1+x}} \geq 1 - \frac{1}{2}x$ and the second utilizes that $g \geq -\frac{c}{2}$.

Since $h \sim \mathcal{X}^2_{d-1}$, from the Laurent-Massart inequality (Laurent & Massart, 2000), we have that

$$\mathbb{P}\left(h \geq d-1+2\sqrt{(d-1)t_1}+2t_1\right) \leq e^{-t_1}.$$

Therefore, we have that with probability at least $1-e^{-t_1}$

$$\frac{1}{\sqrt{1+h/(c+g)^2}} \geq 1 - \frac{2(d-1+2\sqrt{(d-1)t_1}+2t_1)}{c^2}.$$

Setting $t_1 = d$, we get with probability at least $1-e^{-d}$

$$\frac{1}{\sqrt{1+h/(c+g)^2}} \geq 1 - \frac{10d}{c^2}.$$

Combining the result with (22), since $Q(x) \leq 1$ for $x \in \mathbb{R}$ and $Q(x) \leq e^{-x^2/2}$ for $x > 1$, we get that

$$\begin{aligned}
\mathbb{P}(y^\star_{\text{att-1}}(\boldsymbol{Z}) \neq y) &\leq e^{-d} + Q(c/2) + Q\left(\frac{1}{\sigma}\left(1-\frac{10d}{c^2}\right)\right) \\
&\leq e^{-d} + e^{-1/8\varepsilon_\sigma^2} + Q\left(\frac{1}{\sigma}\left(1-10d\varepsilon_\sigma^2\right)\right).
\end{aligned}$$

It completes the proof.

□

E. Analysis of Multi-layer Linear Attention

E.1. Proof of Proposition 4.1

Proof. We consider the following model constructions for the attention matrices in the ℓth layer, $\ell \in [L]$ and the final linear prediction head:

$$\begin{aligned}
&\ell\text{th layer:} \quad \boldsymbol{W}_{q\ell}\boldsymbol{W}_{k\ell}^\top = \begin{bmatrix}\boldsymbol{I}_d & 0\\ 0 & 0\end{bmatrix} \quad \text{and} \quad \boldsymbol{W}_{v\ell} = \begin{bmatrix}a_\ell\boldsymbol{I}_d & 0\\ 0 & b_\ell\end{bmatrix}; \\
&\text{Prediction head:} \quad \boldsymbol{h} = \begin{bmatrix}\boldsymbol{0}_d\\ c\end{bmatrix}.
\end{aligned} \qquad (23)$$

Suppose the input to ℓth layer is

$$\boldsymbol{Z}_\ell = \begin{bmatrix} \boldsymbol{X}_\ell & \boldsymbol{y}_\ell \\ \boldsymbol{x}_\ell^\top & y_\ell \end{bmatrix} \in \mathbb{R}^{(n+1)\times(d+1)} \quad \text{where} \quad \boldsymbol{Z}_1 = \boldsymbol{Z} = \begin{bmatrix} \boldsymbol{X} & \boldsymbol{y} \\ \boldsymbol{x}^\top & 0 \end{bmatrix}.$$

Recapping the model construction from (23), the ℓth layer output returns

$$\begin{aligned}\left(\boldsymbol{Z}_\ell \boldsymbol{W}_{q\ell} \boldsymbol{W}_{k\ell}^\top \boldsymbol{Z}_\ell^\top \boldsymbol{M}\right) \boldsymbol{Z}_\ell \boldsymbol{W}_{v\ell} &= \begin{bmatrix} \boldsymbol{X}_\ell & \boldsymbol{y}_\ell \\ \boldsymbol{x}_\ell^\top & y_\ell \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_d & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{X}_\ell^\top & \boldsymbol{x}_\ell \\ \boldsymbol{y}_\ell^\top & y_\ell \end{bmatrix} \boldsymbol{M} \begin{bmatrix} \boldsymbol{X}_\ell & \boldsymbol{y}_\ell \\ \boldsymbol{x}_\ell^\top & y_\ell \end{bmatrix} \begin{bmatrix} a_\ell \boldsymbol{I}_d & 0 \\ 0 & b_\ell \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{X}_\ell \boldsymbol{X}_\ell^\top & \boldsymbol{X}_\ell \boldsymbol{x}_\ell \\ \boldsymbol{x}_\ell^\top \boldsymbol{X}_\ell^\top & \boldsymbol{x}_\ell^\top \boldsymbol{x}_\ell \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_n & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a_\ell \boldsymbol{X}_\ell & b_\ell \boldsymbol{y}_\ell \\ a_\ell \boldsymbol{x}_\ell^\top & b_\ell y_\ell \end{bmatrix} \\ &= \begin{bmatrix} a_\ell \boldsymbol{X}_\ell \boldsymbol{X}_\ell^\top \boldsymbol{X}_\ell & b_\ell \boldsymbol{X}_\ell \boldsymbol{X}_\ell^\top \boldsymbol{y}_\ell \\ a_\ell \boldsymbol{x}_\ell^\top \boldsymbol{X}_\ell^\top \boldsymbol{X}_\ell & b_\ell \boldsymbol{x}_\ell^\top \boldsymbol{X}_\ell^\top \boldsymbol{y}_\ell \end{bmatrix}. \end{aligned} \tag{24}$$

Therefore, after residual connection, the input of $(\ell+1)$th layer is

$$\begin{aligned}\boldsymbol{Z}_{\ell+1} &= \boldsymbol{Z}_\ell + \begin{bmatrix} a_\ell \boldsymbol{X}_\ell \boldsymbol{X}_\ell^\top \boldsymbol{X}_\ell & b_\ell \boldsymbol{X}_\ell \boldsymbol{X}_\ell^\top \boldsymbol{y}_\ell \\ a_\ell \boldsymbol{x}_\ell^\top \boldsymbol{X}_\ell^\top \boldsymbol{X}_\ell & b_\ell \boldsymbol{x}_\ell^\top \boldsymbol{X}_\ell^\top \boldsymbol{y}_\ell \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{X}_\ell + a_\ell \boldsymbol{X}_\ell \boldsymbol{X}_\ell^\top \boldsymbol{X}_\ell & \boldsymbol{y}_\ell + b_\ell \boldsymbol{X}_\ell \boldsymbol{X}_\ell^\top \boldsymbol{y}_\ell \\ \boldsymbol{x}_\ell^\top + a_\ell \boldsymbol{x}_\ell^\top \boldsymbol{X}_\ell^\top \boldsymbol{X}_\ell & y_\ell + b_\ell \boldsymbol{x}_\ell^\top \boldsymbol{X}_\ell^\top \boldsymbol{y}_\ell \end{bmatrix} \in \mathbb{R}^{(n+1)\times(d+1)}. \end{aligned} \tag{25}$$

• **Label propagation:** We first focus on deriving label propagation results. Suppose that we have

$$a_\ell = 0 \quad \text{for} \quad \ell \in [L].$$

Then following (24), the output of ℓ'th layer takes the following form:

$$\left(\boldsymbol{Z}_\ell \boldsymbol{W}_{q\ell} \boldsymbol{W}_{k\ell}^\top \boldsymbol{Z}_\ell^\top \boldsymbol{M}\right) \boldsymbol{Z}_\ell \boldsymbol{W}_{v\ell} = \begin{bmatrix} 0 & b_\ell \boldsymbol{X}_\ell \boldsymbol{X}_\ell^\top \boldsymbol{y}_\ell \\ 0 & b_\ell \boldsymbol{x}_\ell^\top \boldsymbol{X}_\ell^\top \boldsymbol{y}_\ell \end{bmatrix}.$$

Here, the first d coordinates of each token's output are zeros, and therefore, the corresponding input coordinates remain unchanged, and we have

$$\boldsymbol{X}_\ell \equiv \boldsymbol{X} \quad \text{and} \quad \boldsymbol{x}_\ell \equiv \boldsymbol{x} \quad \text{for} \quad \ell \in [L].$$

The prediction (based on the last token output and after applying prediction head) is given by

$$f_{\text{all-}L}(\boldsymbol{Z}) = c b_L \boldsymbol{x}^\top \boldsymbol{X}^\top \boldsymbol{y}_L. \tag{26}$$

We next focus on obtaining $\boldsymbol{y}_L$. From (25), we have

$$\boldsymbol{y}_{\ell+1} = \boldsymbol{y}_\ell + b_\ell \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{y}_\ell = (\boldsymbol{I} + b_\ell \boldsymbol{X}\boldsymbol{X}^\top)\boldsymbol{y}_\ell.$$

Therefore,

$$\boldsymbol{y}_L = \prod_{\ell=1}^{L-1} (\boldsymbol{I} + b_\ell \boldsymbol{X}\boldsymbol{X}^\top)\boldsymbol{y}.$$

Combining with (26) results in

$$f_{\text{all-}L}(\boldsymbol{Z}) = c b_L \boldsymbol{x}^\top \boldsymbol{X}^\top \prod_{\ell=1}^{L-1} (\boldsymbol{I} + b_\ell \boldsymbol{X}\boldsymbol{X}^\top)\boldsymbol{y} = c b_L \boldsymbol{x}^\top \prod_{\ell=1}^{L-1} (\boldsymbol{I} + b_\ell \boldsymbol{X}^\top \boldsymbol{X})\boldsymbol{X}^\top \boldsymbol{y}.$$

It completes the proof.

• **Feature propagation:** We now focus on the feature propagation setting. In contrast to the label propagation, let us assume that

$$a_\ell \to \infty \quad \text{and} \quad b_\ell \to 0^+ \quad \text{for} \quad \ell \in [L].$$

The prediction (following (24), based on the last token output and after applying prediction head) is given by

$$f_{\text{all-}L}(\boldsymbol{Z}) = cb_L \boldsymbol{x}_L^\top \boldsymbol{X}_L^\top \boldsymbol{y}_L. \tag{27}$$

We first obtain $\boldsymbol{y}_L$. From (25) (since $b_\ell \to 0$), we have

$$\boldsymbol{y}_{\ell+1} = \boldsymbol{y}_\ell + b_\ell \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{y}_\ell = \boldsymbol{y}_\ell.$$

Therefore,

$$\boldsymbol{y}_\ell \equiv \boldsymbol{y} \quad \text{for} \quad \ell \in [L].$$

Next, we focus on $\boldsymbol{X}_L, \boldsymbol{x}_L$. From (25), as $a_\ell \to \infty$, we have

$$\begin{aligned} \boldsymbol{X}_{\ell+1} &= \boldsymbol{X}_\ell + a_\ell \boldsymbol{X}_\ell \boldsymbol{X}_\ell^\top \boldsymbol{X}_\ell = \boldsymbol{X}_\ell(\boldsymbol{I} + a_\ell \boldsymbol{X}_\ell^\top \boldsymbol{X}_\ell) = a_\ell \boldsymbol{X}_\ell \boldsymbol{X}_\ell^\top \boldsymbol{X}_\ell; \\ \boldsymbol{x}_{\ell+1}^\top &= \boldsymbol{x}_\ell^\top + a_\ell \boldsymbol{x}_\ell^\top \boldsymbol{X}_\ell^\top \boldsymbol{X}_\ell = \boldsymbol{x}_\ell^\top(\boldsymbol{I} + a_\ell \boldsymbol{X}_\ell^\top \boldsymbol{X}_\ell) = a_\ell \boldsymbol{x}_\ell^\top \boldsymbol{X}_\ell^\top \boldsymbol{X}_\ell. \end{aligned}$$

Therefore,

$$\begin{aligned} \boldsymbol{X}_L &= a_{L-1}\boldsymbol{X}_{L-1}(\boldsymbol{X}_{L-1}^\top \boldsymbol{X}_{L-1}) \\ &= a_{L-1}a_{L-2}^3 \boldsymbol{X}_{L-2}(\boldsymbol{X}_{L-2}^\top \boldsymbol{X}_{L-2})^{\frac{3^2-1}{2}} \\ &= a_{L-1}a_{L-2}^3 a_{L-3}^{3^2} \boldsymbol{X}_{L-3}(\boldsymbol{X}_{L-3}^\top \boldsymbol{X}_{L-3})^{\frac{3^3-1}{2}} \\ &= \cdots \\ &= a_{L-1}a_{L-2}^3 a_{L-3}^{3^2} ... a_1^{3^{L-2}} \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{\frac{3^{L-1}-1}{2}}, \end{aligned}$$

and

$$\begin{aligned} \boldsymbol{x}_L^\top &= a_{L-1}\boldsymbol{x}_{L-1}^\top(\boldsymbol{X}_{L-1}^\top \boldsymbol{X}_{L-1}) \\ &= a_{L-1}a_{L-2}^3 \boldsymbol{x}_{L-2}^\top(\boldsymbol{X}_{L-2}^\top \boldsymbol{X}_{L-2})^{\frac{3^2-1}{2}} \\ &= a_{L-1}a_{L-2}^3 a_{L-3}^{3^2} \boldsymbol{x}_{L-3}^\top(\boldsymbol{X}_{L-3}^\top \boldsymbol{X}_{L-3})^{\frac{3^3-1}{2}} \\ &= \cdots \\ &= a_{L-1}a_{L-2}^3 a_{L-3}^{3^2} ... a_1^{3^{L-2}} \boldsymbol{x}^\top(\boldsymbol{X}^\top \boldsymbol{X})^{\frac{3^{L-1}-1}{2}}. \end{aligned}$$

Combining all together with (27), we have that

$$\begin{aligned} f_{\text{all-}L}(\boldsymbol{Z}) &= cb_L \boldsymbol{x}_L^\top \boldsymbol{X}_L^\top \boldsymbol{y}_L \\ &= cb_L \left(\prod_{\ell=1}^{L-1} a_\ell^{3^{L-1-\ell}} \right)^2 \boldsymbol{x}^\top (\boldsymbol{X}^\top \boldsymbol{X})^{3^{L-1}-1} \boldsymbol{X}^\top \boldsymbol{y}. \end{aligned}$$

It completes the proof. □

E.2. Proof of Proposition 4.2

Proof. The proof follows directly by adopting the same model construction and proof strategy as in Proposition 4.1, under the additional assumption that

$$a_\ell = a \quad \text{and} \quad b_\ell = b \quad \text{for} \quad \ell \in [L].$$

□

E.3. Proof of Lemma 4.3

Proof. In the proof of Proposition 4.1, we showed how to derive the label and feature propagation results by restricting the construction to either $a_\ell \equiv 0$ (for label propagation) or $(a_\ell \to \infty, b_\ell \to 0)$ (for feature propagation). Here, we consider a propagation process without imposing restrictions on the choices of (a_ℓ, b_ℓ), and study the form of the final prediction returned by the model.

To avoid the notation conflict, we express the matrix $\boldsymbol{A}$ in (10) as

$$\boldsymbol{A} = \sum_{k=0}^{K} e_k (\boldsymbol{X}^\top \boldsymbol{X})^k$$

and let $\boldsymbol{e} = [e_0 \; e_2 \; \cdots \; e_{(3^L-3)/2}]^\top \in \mathbb{R}^{K+1}$.

Recall the same model construction used in the proof of Proposition 4.1, defined in (23). From (24), we have that

$$f_{\text{att-}L}(\boldsymbol{Z}) = c b_L \boldsymbol{x}_L^\top \boldsymbol{X}_L^\top \boldsymbol{y}_L$$

where following (25), we have

$$\begin{aligned}
\boldsymbol{X}_{\ell+1} &= \boldsymbol{X}_\ell (\boldsymbol{I} + a_\ell \boldsymbol{X}_\ell^\top \boldsymbol{X}_\ell), \\
\boldsymbol{x}_{\ell+1}^\top &= \boldsymbol{x}_\ell^\top (\boldsymbol{I} + a_\ell \boldsymbol{X}_\ell^\top \boldsymbol{X}_\ell), \\
\boldsymbol{y}_{\ell+1} &= (\boldsymbol{I} + b_\ell \boldsymbol{X}_\ell \boldsymbol{X}_\ell^\top) \boldsymbol{y}_\ell.
\end{aligned}$$

At each layer, the operations performed are linear combinations and multiplications involving $\boldsymbol{X}_\ell^\top \boldsymbol{X}_\ell$ and identity matrices scaled by the parameters (a_ℓ, b_ℓ). Thus, each coefficient e_k of $(\boldsymbol{X}^\top \boldsymbol{X})^k$ depends smoothly on the scalar parameters (a_ℓ, b_ℓ).

From (24) and (25), we have that

$$\begin{aligned}
f_{\text{att-}L}(\boldsymbol{Z}) &= c b_L \boldsymbol{x}_L^\top \boldsymbol{X}_L^\top \boldsymbol{y}_L \\
&= c b_L \cdot \boldsymbol{x}_{L-1}^\top (\boldsymbol{I} + a_{L-1} \boldsymbol{X}_{L-1}^\top \boldsymbol{X}_{L-1})^2 (\boldsymbol{I} + b_{L-1} \boldsymbol{X}_{L-1}^\top \boldsymbol{X}_{L-1}) \boldsymbol{X}_{L-1}^\top \boldsymbol{y}_{L-1} \\
&= \cdots
\end{aligned} \tag{28}$$

That is, in the final $f_{\text{att-}L}(\boldsymbol{Z})$ expression, the coefficients corresponding to different degrees of $(\boldsymbol{X}^\top \boldsymbol{X})^k$ depend on the model parameters $c b_L$ and $(a_\ell, b_\ell)_{\ell=1}^{L-1}$, which together have at most $2L-1$ degrees of freedom. Let $\boldsymbol{c} = [c b_L \; a_1 \; \cdots \; a_{L-1} \; b_1 \; \cdots \; b_{L-1}]^\top$. This means there exists a smooth function $g : \mathbb{R}^{2L-1} \to \mathbb{R}^K$ such that: $\boldsymbol{e} = g(\boldsymbol{c})$.

It remains to show that an L-layer linear attention model can produce terms involving powers of $\boldsymbol{X}^\top \boldsymbol{X}$ up to degree $(3^L - 3)/2$.

Let $f(\boldsymbol{Z})$ be a function that contains terms of the form $\boldsymbol{x}^\top (\boldsymbol{X}^\top \boldsymbol{X})^k \boldsymbol{X}^\top \boldsymbol{y}$ for various powers k. Define $\mathcal{P}(f(\boldsymbol{Z}))$ as the projection that extracts the highest degree k present in $f(\boldsymbol{Z})$. For example, $\mathcal{P}\big(\boldsymbol{x}^\top (\boldsymbol{I} + (\boldsymbol{X}^\top \boldsymbol{X})^2) \boldsymbol{X}^\top \boldsymbol{y}\big) = 2$. Then from (28), we have

$$\begin{aligned}
\mathcal{P}(f_{\text{att-}L}(\boldsymbol{Z})) &= \mathcal{P}(\boldsymbol{x}_L^\top \boldsymbol{X}_L^\top \boldsymbol{y}_L) \\
&= \mathcal{P}(\boldsymbol{x}_{L-1}^\top (\boldsymbol{X}_{L-1}^\top \boldsymbol{X}_{L-1})^3 \boldsymbol{X}_{L-1}^\top \boldsymbol{y}_{L-1}) \\
&= \mathcal{P}(\boldsymbol{x}_{L-2}^\top (\boldsymbol{X}_{L-2}^\top \boldsymbol{X}_{L-2})(\boldsymbol{X}_{L-2}^\top \boldsymbol{X}_{L-2})^{3^2} (\boldsymbol{X}_{L-2}^\top \boldsymbol{X}_{L-2})^2 \boldsymbol{X}_{L-2}^\top \boldsymbol{y}_{L-2}) \\
&= \mathcal{P}(\boldsymbol{x}_{L-2}^\top (\boldsymbol{X}_{L-2}^\top \boldsymbol{X}_{L-2})^{3^2+3} \boldsymbol{X}_{L-2}^\top \boldsymbol{y}_{L-2}) \\
&= \mathcal{P}(\boldsymbol{x}_{L-3}^\top (\boldsymbol{X}_{L-3}^\top \boldsymbol{X}_{L-3})(\boldsymbol{X}_{L-3}^\top \boldsymbol{X}_{L-3})^{3^3+3^2} (\boldsymbol{X}_{L-3}^\top \boldsymbol{X}_{L-3})^2 \boldsymbol{X}_{L-3}^\top \boldsymbol{y}_{L-3}) \\
&= \mathcal{P}(\boldsymbol{x}_{L-3}^\top (\boldsymbol{X}_{L-3}^\top \boldsymbol{X}_{L-3})^{3^3+3^2+3} \boldsymbol{X}_{L-3}^\top \boldsymbol{y}_{L-3}) \\
&= \ldots \\
&= \mathcal{P}(\boldsymbol{x}^\top (\boldsymbol{X}^\top \boldsymbol{X})^{3^{L-1}+\cdots+3^2+3} \boldsymbol{X}^\top \boldsymbol{y}) \\
&= 3^{L-1} + \cdots + 3^2 + 3 = \frac{3^L - 3}{2}.
\end{aligned}$$

It completes the proof.

□

E.4. Proof of Theorem 4.4

Proof. Let $\boldsymbol{\xi} \sim \mathcal{N}(0, \boldsymbol{I})$ and rewrite $y\boldsymbol{x} = \boldsymbol{\mu} + \sigma\boldsymbol{\xi}$. For any matrix $\boldsymbol{A} \in \mathbb{R}^{d\times d}$, the prediction error of $\hat{y}_{\boldsymbol{A}} = \text{sgn}(\boldsymbol{x}^\top \boldsymbol{A}\hat{\boldsymbol{\mu}}_s)$ given $\hat{\boldsymbol{\mu}}_s$ returns

$$\begin{aligned}\mathbb{P}(\hat{\boldsymbol{y}}_{\boldsymbol{A}} \neq y \mid \hat{\boldsymbol{\mu}}_s) &= \mathbb{P}(y\boldsymbol{x}^\top \boldsymbol{A}\hat{\boldsymbol{\mu}}_s < 0 \mid \hat{\boldsymbol{\mu}}_s)\\ &= \mathbb{P}((\boldsymbol{\mu} + \sigma\boldsymbol{\xi})^\top \boldsymbol{A}\hat{\boldsymbol{\mu}}_s < 0 \mid \hat{\boldsymbol{\mu}}_s)\\ &= Q\left(\frac{\boldsymbol{\mu}^\top \boldsymbol{A}\hat{\boldsymbol{\mu}}_s}{\sigma \left\|\boldsymbol{A}\hat{\boldsymbol{\mu}}_s\right\|_{\ell_2}}\right). \end{aligned} \tag{29}$$

For any $\boldsymbol{A} \in \mathbb{R}^{d\times d}$, we can decompose it as

$$\boldsymbol{A} = \sum_{i=1}^{d} \lambda_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$$

where $\boldsymbol{u}_1 = \boldsymbol{\mu}$, $\|\boldsymbol{u}_i\|_{\ell_2} = 1$ and $\boldsymbol{u}_i^\top \boldsymbol{u}_j = 0$ for any $i \neq j$. Let $\lambda_1 > 0$. Then, we get

$$\begin{aligned}\boldsymbol{\mu}^\top \boldsymbol{A}\hat{\boldsymbol{\mu}}_s &= \boldsymbol{\mu}^\top (\sum_{i=1}^{d} \lambda_i \boldsymbol{u}_i \boldsymbol{v}_i^\top)\hat{\boldsymbol{\mu}}_s\\ &= \sum_{i=1}^{d} \lambda_i \boldsymbol{\mu}^\top \boldsymbol{u}_i \boldsymbol{v}_i^\top \hat{\boldsymbol{\mu}}_s\\ &= \lambda_1 \boldsymbol{\mu}^\top \boldsymbol{u}_1 \boldsymbol{v}_1^\top \hat{\boldsymbol{\mu}}_s\\ &= \lambda_1 \boldsymbol{v}_1^\top \hat{\boldsymbol{\mu}}_s. \end{aligned} \tag{30}$$

Now consider $\|\boldsymbol{A}\hat{\boldsymbol{\mu}}_s\|_{\ell_2}$ where we have

$$\begin{aligned}\boldsymbol{A}\hat{\boldsymbol{\mu}}_s &= \sum_{i=1}^{d} \lambda_i \boldsymbol{u}_i \boldsymbol{v}_i^\top \hat{\boldsymbol{\mu}}_s\\ &= \lambda_1 \boldsymbol{\mu} \boldsymbol{v}_1^\top \hat{\boldsymbol{\mu}}_s + \sum_{i=2}^{d} \lambda_i \boldsymbol{u}_i \boldsymbol{v}_i^\top \hat{\boldsymbol{\mu}}_s. \end{aligned}$$

Since $\boldsymbol{u}_i$, $i \neq 1$ is orthogonal to $\boldsymbol{\mu}$, $\lambda_1 \boldsymbol{\mu} \boldsymbol{v}_1^\top \hat{\boldsymbol{\mu}}_s$ is orthogonal to $\sum_{i=2}^{d} \lambda_i \boldsymbol{u}_i \boldsymbol{v}_i^\top \hat{\boldsymbol{\mu}}_s$. Therefore, given $\|\boldsymbol{u}_i\|_{\ell_2} = 1$ for all $i \in [d]$, it obeys

$$\|\boldsymbol{A}\hat{\boldsymbol{\mu}}_s\|_{\ell_2}^2 = \left\|\lambda_1 \boldsymbol{\mu} \boldsymbol{v}_1^\top \hat{\boldsymbol{\mu}}_s\right\|_{\ell_2}^2 + \sum_{i=2}^{d} \left\|\lambda_i \boldsymbol{u}_i \boldsymbol{v}_i^\top \hat{\boldsymbol{\mu}}_s\right\|_{\ell_2}^2 = (\lambda_1 \boldsymbol{v}_1^\top \hat{\boldsymbol{\mu}}_s)^2 + \lambda_1^2 \sum_{i=2}^{d} (\lambda_1^{-1} \lambda_i \boldsymbol{v}_i^\top \hat{\boldsymbol{\mu}}_s)^2. \tag{31}$$

For simplicity, define

$$\Delta(\hat{\boldsymbol{\mu}}_s) = \sum_{i=2}^{d} (\lambda_1^{-1} \lambda_i \boldsymbol{v}_i^\top \hat{\boldsymbol{\mu}}_s)^2$$

where $\Delta(\cdot)$ is a function of λ_1 and $(\lambda_i, \boldsymbol{v}_i)$'s for $i \geq 2$, and we have

$$\Delta(\hat{\boldsymbol{\mu}}_s) \geq 0 \quad \text{and} \quad \Delta(-\hat{\boldsymbol{\mu}}_s) = \Delta(\hat{\boldsymbol{\mu}}_s).$$

Recall that $\hat{\boldsymbol{\mu}}_s$ is the SPI estimator (cf. (SPI)). Let $|\mathcal{I}| = m$. We can write $\hat{\boldsymbol{\mu}}_s = \boldsymbol{\mu} + \boldsymbol{\xi}'/\sqrt{m}$ where $\boldsymbol{\xi}' \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$.

Using (29), (30) and (31), the classification error becomes

$$\begin{aligned}
\mathbb{P}(\hat{\boldsymbol{y}}_{\boldsymbol{A}} \neq y) &= \mathbb{E}_{\hat{\boldsymbol{\mu}}_s}\left[Q\left(\frac{\boldsymbol{\mu}^\top \boldsymbol{A}\hat{\boldsymbol{\mu}}_s}{\sigma \left\|\boldsymbol{A}\hat{\boldsymbol{\mu}}_s\right\|_{\ell_2}}\right)\right] \\
&= \mathbb{E}_{\hat{\boldsymbol{\mu}}_s}\left[Q\left(\frac{\boldsymbol{v}_1^\top \hat{\boldsymbol{\mu}}_s}{\sigma\sqrt{(\boldsymbol{v}_1^\top \hat{\boldsymbol{\mu}}_s)^2 + \Delta(\hat{\boldsymbol{\mu}}_s)}}\right)\right] \\
&= \mathbb{E}_{\boldsymbol{v}_1^\top \hat{\boldsymbol{\mu}}_s < 0}\left[Q\left(\frac{\boldsymbol{v}_1^\top \hat{\boldsymbol{\mu}}_s}{\sigma\sqrt{(\boldsymbol{v}_1^\top \hat{\boldsymbol{\mu}}_s)^2 + \Delta(\hat{\boldsymbol{\mu}}_s)}}\right)\right] + \mathbb{E}_{\boldsymbol{v}_1^\top \hat{\boldsymbol{\mu}}_s \geq 0}\left[Q\left(\frac{\boldsymbol{v}_1^\top \hat{\boldsymbol{\mu}}_s}{\sigma\sqrt{(\boldsymbol{v}_1^\top \hat{\boldsymbol{\mu}}_s)^2 + \Delta(\hat{\boldsymbol{\mu}}_s)}}\right)\right].
\end{aligned}$$

First, note that for any $x > 0$, $Q(x) < 0.5 < Q(-x)$. Therefore, the optimal choice of $\boldsymbol{v}_1 \in \mathbb{R}^d$ that minimizes $\mathbb{P}(\hat{y}_{\boldsymbol{A}} \neq y)$ is contained within the set of $\boldsymbol{v}_1$ values that maximize $\mathbb{P}(\boldsymbol{v}_1^\top \hat{\boldsymbol{\mu}}_s > 0)$. Let $\boldsymbol{v}_1^\star := \arg\max_{\boldsymbol{v}_1 \in \mathbb{R}^d} \mathbb{P}(\boldsymbol{v}_1^\top \hat{\boldsymbol{\mu}}_s > 0)$. Given that $\hat{\boldsymbol{\mu}}_s \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2/m\boldsymbol{I})$, we have that $\boldsymbol{v}_1^\star = c\boldsymbol{\mu}$ for $c > 0$. Let $c = 1$ and therefore, $\boldsymbol{v}_1^\star = \boldsymbol{\mu}$ without loss of generality (since λ_1 can be any positive scalar). Then we obtain

$$\min_{\boldsymbol{A} \in \mathbb{R}^{d \times d}} \mathbb{P}(\hat{\boldsymbol{y}}_{\boldsymbol{A}} \neq y) = \min_{\Delta} \mathbb{E}_{\hat{\boldsymbol{\mu}}_s}\left[Q\left(\frac{\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s}{\sigma\sqrt{(\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s)^2 + \Delta(\hat{\boldsymbol{\mu}}_s)}}\right)\right].$$

Let $f(\hat{\boldsymbol{\mu}}_s)$ be the probability density function of $\hat{\boldsymbol{\mu}}_s$. Since $\hat{\boldsymbol{\mu}}_s \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2/m\boldsymbol{I})$, then it satisfies

$$f(\hat{\boldsymbol{\mu}}_s) \geq f(-\hat{\boldsymbol{\mu}}_s) \quad \text{for any } \boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s > 0. \tag{32}$$

Therefore, the classification error becomes

$$\begin{aligned}
\mathbb{P}(\hat{\boldsymbol{y}}_{\boldsymbol{A}} \neq y \mid \boldsymbol{v}_1 = \boldsymbol{\mu}) &= \int_{\hat{\boldsymbol{\mu}}_s} f(\hat{\boldsymbol{\mu}}_s) Q\left(\frac{\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s}{\sigma\sqrt{(\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s)^2 + \Delta(\hat{\boldsymbol{\mu}}_s)}}\right) d\hat{\boldsymbol{\mu}}_s \\
&= \int_{\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s > 0} f(\hat{\boldsymbol{\mu}}_s) Q\left(\frac{\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s}{\sigma\sqrt{(\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s)^2 + \Delta(\hat{\boldsymbol{\mu}}_s)}}\right) + f(-\hat{\boldsymbol{\mu}}_s) Q\left(\frac{-\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s}{\sigma\sqrt{(\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s)^2 + \Delta(\hat{\boldsymbol{\mu}}_s)}}\right) d\hat{\boldsymbol{\mu}}_s \\
&= \int_{\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s > 0} \left(f(\hat{\boldsymbol{\mu}}_s) - f(-\hat{\boldsymbol{\mu}}_s)\right) Q\left(\frac{\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s}{\sigma\sqrt{(\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s)^2 + \Delta(\hat{\boldsymbol{\mu}}_s)}}\right) + f(-\hat{\boldsymbol{\mu}}_s) d\hat{\boldsymbol{\mu}}_s.
\end{aligned}$$

Following (32), to minimize the error, we need minimize $Q\left(\frac{\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s}{\sigma\sqrt{(\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s)^2 + \Delta(\hat{\boldsymbol{\mu}}_s)}}\right)$ for $\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s > 0$, which can be easily done by choosing $\lambda_i = 0$ for $i \geq 2$. Then we get $\Delta(\hat{\boldsymbol{\mu}}_s) \equiv 0$. Therefore, the optimal solution set $\mathcal{A}^\star$ defined in Theorem 4.4 satisfies:

$$\mathcal{A}^\star = \left\{\lambda_1 \boldsymbol{\mu}\boldsymbol{\mu}^\top \mid \lambda_1 > 0\right\}.$$

Combining all together, we obtain

$$\begin{aligned}
\min_{\boldsymbol{A} \in \mathbb{R}^{d \times d}} \mathbb{P}(\hat{\boldsymbol{y}}_{\boldsymbol{A}} \neq y) &= \int_{\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s > 0} \left(f(\hat{\boldsymbol{\mu}}_s) - f(-\hat{\boldsymbol{\mu}})\right) Q\left(\frac{1}{\sigma}\right) + f(-\hat{\boldsymbol{\mu}}_s) d\hat{\boldsymbol{\mu}}_s \\
&= \int_{\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s > 0} f(\hat{\boldsymbol{\mu}}_s) d\hat{\boldsymbol{\mu}}_s \cdot Q\left(\frac{1}{\sigma}\right) + \int_{\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s < 0} f(\hat{\boldsymbol{\mu}}_s) d\hat{\boldsymbol{\mu}}_s \cdot \left(1 - Q\left(\frac{1}{\sigma}\right)\right) \\
&= Q\left(-\frac{\sqrt{m}}{\sigma}\right) Q\left(\frac{1}{\sigma}\right) + Q\left(\frac{\sqrt{m}}{\sigma}\right)\left(1 - Q\left(\frac{1}{\sigma}\right)\right) \\
&= \left(1 - Q\left(\frac{\sqrt{m}}{\sigma}\right)\right) Q\left(\frac{1}{\sigma}\right) + Q\left(\frac{\sqrt{m}}{\sigma}\right)\left(1 - Q\left(\frac{1}{\sigma}\right)\right) \\
&= Q\left(\frac{1}{\sigma}\right) + Q\left(\frac{\sqrt{m}}{\sigma}\right) - 2Q\left(\frac{\sqrt{m}}{\sigma}\right) Q\left(\frac{1}{\sigma}\right).
\end{aligned}$$

It completes the proof. □

E.5. Non-asymptotic Analysis

In Section 4 and Theorem 4.5, we showed that with infinitely many unlabeled samples, an L-layer linear attention model (for $L \geq 2$) can implement the predictor described in Theorem 4.4 with optimal $\boldsymbol{A}$ choice, achieving the classification error given by (11). In this section, we turn to the non-asymptotic setting where n is finite, and analyze the model's performance under this regime.

Theorem E.1. *Let the prompt $\boldsymbol{Z}$ be generated as described in Section 2.2. Consider an L-layer linear attention model with $L \geq 2$ and denote its optimal prediction as $y^\star_{\text{att-}L}(\boldsymbol{Z})$. Additionally, let $\hat{\boldsymbol{\mu}}_s$ be the SPI estimator defined in (SPI). Suppose that the number of labeled samples satisfies $np \geq 8d\sigma^2$ and $n > O(d)$ is sufficiently large. Then, there exists a universal constant $C > 0$ such that the classification error satisfies*

$$\mathbb{P}(y^\star_{\text{att-}L}(\boldsymbol{Z}) \neq y) \leq Q\left(\frac{1 - C\sqrt{d/n}}{\sigma}\right) + e^{-d}.$$

Proof. Recap from Proposition 4.1. For any L-layer attention model with $L \geq 2$, it can output

$$f_{\text{att-}L}(\boldsymbol{Z}) = \boldsymbol{x}^\top(\boldsymbol{X}^\top\boldsymbol{X}/n - \sigma^2\boldsymbol{I})\hat{\boldsymbol{\mu}}_s. \tag{33}$$

Let

$$\hat{y} = \text{sgn}(f_{\text{att-}L}(\boldsymbol{Z}))$$

with $f_{\text{att-}L}(\boldsymbol{Z})$ defined in (33). Then we have

$$\mathbb{P}(y^\star_{\text{att-}L}(\boldsymbol{Z}) \neq y) \leq \mathbb{P}(\hat{y} \neq y).$$

Therefore, in the following, we focus on upper-bounding the classification error $\mathbb{P}(\hat{y} \neq y)$ corresponding to (33). Given that the optimal prediction under the form $\text{sgn}(\boldsymbol{x}^\top\boldsymbol{A}\hat{\boldsymbol{\mu}}_s)$ is given by $\hat{y}_{\boldsymbol{\mu}\boldsymbol{\mu}^\top} := \text{sgn}(\boldsymbol{x}^\top\boldsymbol{\mu}\boldsymbol{\mu}^\top\hat{\boldsymbol{\mu}}_s)$ (cf. Theorem 4.4), with its corresponding error presented in (11). To analyze the performance of $\hat{y}$, we study its difference from the prediction $\hat{y}_{\boldsymbol{\mu}\boldsymbol{\mu}^\top}$.

To begin with, let $\boldsymbol{g}_i = \boldsymbol{\xi}_i/\sigma \sim \mathcal{N}(0, \boldsymbol{I})$ and $\boldsymbol{g} = \sum_{i=1}^n \boldsymbol{\xi}_i/\sigma\sqrt{n} \sim \mathcal{N}(0, \boldsymbol{I})$. For simplicity, let $\boldsymbol{A} := \boldsymbol{X}^\top\boldsymbol{X}/n - \sigma^2\boldsymbol{I}$. We get

$$\begin{aligned}
\boldsymbol{A} &= \frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X} - \sigma^2\boldsymbol{I} \\
&= \frac{1}{n}\left(\sum_{i=1}^n \boldsymbol{\mu}\boldsymbol{\mu}^\top + \boldsymbol{\mu}\boldsymbol{\xi}_i^\top + \boldsymbol{\xi}_i\boldsymbol{\mu}^\top + \boldsymbol{\xi}_i\boldsymbol{\xi}_i^\top\right) - \sigma^2\boldsymbol{I} \\
&= \boldsymbol{\mu}\boldsymbol{\mu}^\top + \frac{\sigma}{\sqrt{n}}(\boldsymbol{\mu}\boldsymbol{g}^\top + \boldsymbol{g}\boldsymbol{\mu}^\top) + \sigma^2\left(\frac{\sum_{i=1}^n \boldsymbol{g}_i\boldsymbol{g}_i^\top}{n} - \boldsymbol{I}\right).
\end{aligned}$$

Recall (29) from the proof of Theorem 4.4. Our goal is to bound

$$\mathbb{P}(\hat{y} \neq y) = \mathbb{E}_{\hat{\boldsymbol{\mu}}}\left[Q\left(\frac{\boldsymbol{\mu}^\top\boldsymbol{A}\hat{\boldsymbol{\mu}}_s}{\sigma\left\|\boldsymbol{A}\hat{\boldsymbol{\mu}}_s\right\|_{\ell_2}}\right)\right].$$

Define

$$\boldsymbol{\Delta} := \boldsymbol{A} - \boldsymbol{\mu}\boldsymbol{\mu}^\top = \frac{\sigma}{\sqrt{n}}(\boldsymbol{\mu}\boldsymbol{g}^\top + \boldsymbol{g}\boldsymbol{\mu}^\top) + \sigma^2\left(\frac{\sum_{i=1}^n \boldsymbol{g}_i\boldsymbol{g}_i^\top}{n} - \boldsymbol{I}\right). \tag{34}$$

From the Laurent-Massart inequality (Laurent & Massart, 2000), we have that with probability at least $1 - e^{-t_1}$ (assuming $t_1 \geq d$), the first term of (34) can be bounded by

$$\frac{1}{\sqrt{n}}\left\|\boldsymbol{\mu}\boldsymbol{g}^\top + \boldsymbol{g}\boldsymbol{\mu}^\top\right\| \leq \frac{2\|\boldsymbol{g}\|}{\sqrt{n}} \leq 6\sqrt{\frac{t_1}{n}}. \tag{35}$$

Additionally, from (Neopane, 2018), we have that with probability at least $1-e^{-t_2}$ (assuming $t_2 \geq d$), the second term of $\boldsymbol{\Delta}$ (cf. (34)) is bounded by (with a universal constant $C > 0$)

$$\left\| \frac{\sum_{i=1}^{n} \boldsymbol{g}_i \boldsymbol{g}_i^\top}{n} - \boldsymbol{I} \right\| \leq C \cdot \sqrt{\frac{t_2}{n}}. \tag{36}$$

Combining (35) and (36), we get with probability at least $1 - 2e^{-t}$ (for $t \geq d$)

$$\|\boldsymbol{\Delta}\| \leq C_1 \sqrt{\frac{t}{n}} \quad \text{where} \quad C_1 := 6\sigma + C\sigma^2.$$

We also bound $\|\hat{\boldsymbol{\mu}}_s\|$ as follows. Let $\hat{\boldsymbol{\mu}}_s = \boldsymbol{\mu} + \sigma/\sqrt{m}\boldsymbol{g}' \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 m \boldsymbol{I})$, similar to (35), with probability at least $1 - e^{-t_3}$ (assuming $2d \leq t_3 \leq m/4\sigma^2$), we can bound

$$\|\hat{\boldsymbol{\mu}}_s\| \leq 1 + \frac{\sigma}{\sqrt{m}} \|\boldsymbol{g}'\| \leq 1 + 3\sigma \sqrt{\frac{t_3}{m}} \leq 3.$$

Then consider a significantly large n (to ensure that $\|\boldsymbol{\Delta}\| \leq 1/12$, e.g., $n \geq (12C_1)^2 t$). With probability at least $1 - 3e^{-\min(t,t_3)}$ and suppose that $\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s > 0.5$, we can bound

$$\begin{aligned}
\left| \frac{\boldsymbol{\mu}^\top \boldsymbol{A} \hat{\boldsymbol{\mu}}_s}{\|\boldsymbol{A}\hat{\boldsymbol{\mu}}_s\|_{\ell_2}} - \frac{\boldsymbol{\mu}^\top \boldsymbol{\mu}\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s}{\|\boldsymbol{\mu}\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s\|_{\ell_2}} \right| &= \left| \frac{\boldsymbol{\mu}^\top (\boldsymbol{\Delta} + \boldsymbol{\mu}\boldsymbol{\mu}^\top) \hat{\boldsymbol{\mu}}_s}{\|(\boldsymbol{\Delta} + \boldsymbol{\mu}\boldsymbol{\mu}^\top)\hat{\boldsymbol{\mu}}_s\|_{\ell_2}} - \frac{\boldsymbol{\mu}^\top \boldsymbol{\mu}\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s}{\|\boldsymbol{\mu}\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s\|_{\ell_2}} \right| \\
&\leq \left| \frac{\boldsymbol{\mu}^\top \boldsymbol{\Delta} \hat{\boldsymbol{\mu}}_s}{\min(\|(\boldsymbol{\Delta} + \boldsymbol{\mu}\boldsymbol{\mu}^\top)\hat{\boldsymbol{\mu}}_s\|_{\ell_2}, \|\boldsymbol{\mu}\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s\|_{\ell_2})} \right| \\
&\leq \frac{\|\boldsymbol{\Delta}\| \cdot \|\hat{\boldsymbol{\mu}}_s\|}{\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s - \|\boldsymbol{\Delta}\| \cdot \|\hat{\boldsymbol{\mu}}_s\|} \\
&\leq 4\|\boldsymbol{\Delta}\| \cdot \|\hat{\boldsymbol{\mu}}_s\| \\
&\leq C_2 \sqrt{\frac{t}{n}} \quad \text{where} \quad C_2 := 12C_1.
\end{aligned}$$

Now, we are ready to bound the classification error, where we get

$$\begin{aligned}
\mathbb{P}(\hat{y} \neq y) &= \mathbb{E}_{\hat{\boldsymbol{\mu}}} \left[Q\left(\frac{\boldsymbol{\mu}^\top \boldsymbol{A} \hat{\boldsymbol{\mu}}_s}{\sigma \|\boldsymbol{A}\hat{\boldsymbol{\mu}}_s\|_{\ell_2}} \right) \right] \\
&= \mathbb{E}_{\hat{\boldsymbol{\mu}}} \left[Q\left(\frac{1}{\sigma} + \frac{1}{\sigma} \left(\frac{\boldsymbol{\mu}^\top \boldsymbol{A} \hat{\boldsymbol{\mu}}_s}{\|\boldsymbol{A}\hat{\boldsymbol{\mu}}_s\|_{\ell_2}} - \frac{\boldsymbol{\mu}^\top \boldsymbol{\mu}\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s}{\|\boldsymbol{\mu}\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s\|_{\ell_2}} \right) \right) \right] \\
&\leq \mathbb{P}(\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s > 0.5) \left(Q\left(\frac{1 - C_2\sqrt{t/n}}{\sigma} \right) + 3e^{-\min(t,t_3)} \right) + \mathbb{P}(\boldsymbol{\mu}^\top \hat{\boldsymbol{\mu}}_s < 0.5) \\
&\leq Q\left(\frac{1 - C_2\sqrt{t/n}}{\sigma} \right) + 3e^{-\min(t,t_3)} + Q\left(\frac{\sqrt{m}}{2\sigma} \right).
\end{aligned}$$

Choosing $t = t_3 = 2d$, since $m/4\sigma^2 \geq 2d$, we obtain

$$\begin{aligned}
\mathbb{P}(\hat{y} \neq y) &\leq Q\left(\frac{1 - C_2\sqrt{2d/n}}{\sigma} \right) + 3e^{-2d} + 0.5e^{-d} \\
&\leq Q\left(\frac{1 - C_2\sqrt{2d/n}}{\sigma} \right) + e^{-d}.
\end{aligned}$$

It completes the proof. □