

# Provable Benefits of Task-Specific Prompts for In-context Learning

Xiangyu Chang<sup>1</sup> Yingcong Li<sup>2</sup> Muti Kara<sup>3</sup> Samet Oymak<sup>2</sup> Amit K. Roy-Chowdhury<sup>1</sup>

<sup>1</sup>University of California, Riverside <sup>2</sup>University of Michigan <sup>3</sup>Bilkent University

## Abstract

The in-context learning capabilities of modern language models have motivated a deeper mathematical understanding of sequence models. A line of recent work has shown that linear attention models can emulate projected gradient descent iterations to implicitly learn the task vector from the data provided in the context window. In this work, we consider a novel setting where the global task distribution can be partitioned into a union of conditional task distributions. We then examine the use of task-specific prompts and prediction heads for learning the prior information associated with the conditional task distribution using a one-layer attention model. Our results on loss landscape show that task-specific prompts facilitate a *covariance-mean decoupling* where prompt-tuning explains the conditional mean of the distribution whereas the variance is learned/explained through in-context learning. Incorporating task-specific head further aids this process by entirely decoupling estimation of mean and variance components. This covariance-mean perspective similarly explains how jointly training prompt and attention weights can provably help over fine-tuning after pretraining. The code for reproducing the numerical results is available at [GitHub](#).

## 1 Introduction

Modern language models possess a remarkable ability to learn new tasks or solve complex problems using examples provided within their context window [Brown et al., 2020, GeminiTeam et al., 2023, OpenAI, 2023, Touvron et al., 2023]. This capability, known as *in-context*

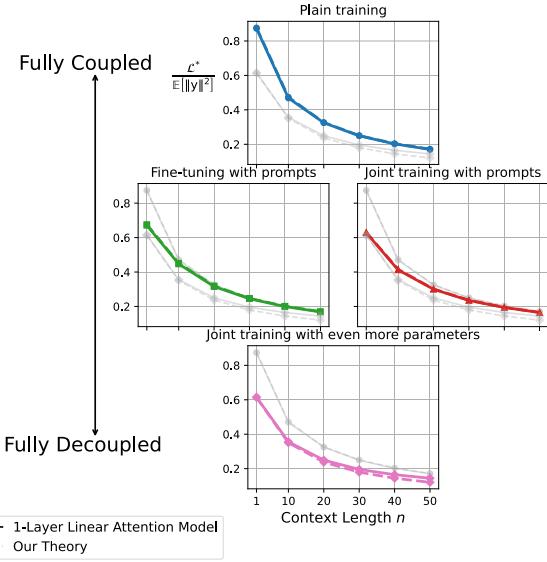


Figure 1: Overview of our work: We present a theoretical analysis of a 1-layer linear attention model for multi-task in-context learning (ICL), examining different configurations of task-specific parameters. By progressively introducing more task-specific parameters across various training settings, we achieve complete **covariance-mean decoupling**, leading to an optimal multi-task ICL loss. In our analysis, we derive upper and lower bounds for the multi-task ICL loss, corresponding to fully coupled and fully decoupled covariance-mean scenarios.

*learning* (ICL), offers a novel and efficient alternative to traditional fine-tuning methods. ICL enables models to adapt to a wide range of tasks through a single forward pass, eliminating the need for task-specific weight updates. This adaptability has made ICL a central feature in the use of large language models (LLMs), extending their utility across diverse applications.

In a basic ICL setting, we construct an input sequence  $Z$  that contains a query to label and related input-label demonstrations. We feed  $Z$  to a sequence model  $f$  to predict the label  $y$  of this query. Thus, the ICL

optimization can be written as

$$\min_{\mathbf{W}} \mathbb{E}_{y, \mathbf{Z}} [\text{loss}(y, f_{\mathbf{W}}(\mathbf{Z}))], \quad (1)$$

where  $\mathbf{W}$  denotes the weights of  $f$ . In practice, however, ICL examples are typically paired with task-specific prompts. These prompts can be manually crafted or automatically generated using methods like differentiable optimization or zero-order search. In fact, the model can often solve the task without any ICL example (zero-shot) by solely relying on the prompt. This motivates a deeper understanding of the synergies between prompt-tuning and in-context learning. Concretely, consider a multitask learning setting where we first sample a task  $t$  and then sample  $(\mathbf{Z}, y)$  from the associated task distribution. In this case, a standard approach is crafting a dedicated prompt  $\mathbf{p}_t$  to feed with the input sequence. The joint optimization of the weights  $\mathbf{W}$  and prompts  $\mathbf{P} = (\mathbf{p}_t)_{t \geq 1}$  takes the form

$$\min_{\mathbf{W}, \mathbf{P}} \mathbb{E} [\text{loss}(y, f_{\mathbf{W}}(\mathbf{Z}, \mathbf{p}_t))]. \quad (2)$$

Contrasting (1) with (2) motivate a few fundamental questions:

- Q1. How do ICL and prompt-tuning synergistically contribute to learning?
- Q2. In practice, we first pretrain a model via (1) and then tune a task-specific prompt. Does joint training have an advantage over this?
- Q3. Can utilizing additional task-specific parameters together with prompts, further boost performance?

To answer these questions, we conduct a comprehensive investigation of the optimization landscape for attention weights and task-specific prompts within a multitask dataset model, examining both joint optimization approaches and sequential strategies involving attention weight pretraining followed by prompt-tuning. We derive closed-form solutions for optimal parameters and loss landscapes, introducing the novel concept of "covariance-mean decoupling" to elucidate the impact of different training strategies on model performance. While previous theoretical research has primarily focused on attention weight optimization, we extend the analysis to include tunable prompts in a multi-task linear regression framework, addressing the practical scenario where models fine-tune task-specific modules on fixed backbones. Notably, our work advances beyond existing research by incorporating non-zero task mean and non-isotropic covariance considerations, revealing why fine-tuning may not consistently enhance performance. Our theoretical framework demonstrates how varying performance gains across training strategies can be attributed to covariance-mean decoupling,

providing both theoretical foundations and practical insights for optimizing attention-based models through careful design of tunable components.

Our key contributions are:

1. **Comprehensive analysis of training strategies:** We analyze multi-task linear regression with a linear attention layer and tunable parameters, covering joint optimization and pretraining-finetuning methods. A unified parameterization allows for closed-form solutions of optimal parameters and loss landscapes, generalizing prior work. (see Section 4)
2. **Mean-covariance decoupling concept:** We introduce covariance-mean decoupling through closed-form loss landscape analysis, demonstrating its correlation with model performance—the greater the decoupling, the better the model's predictions. (see Section 4.2)
3. **Path to optimal in-context learning:** Our analysis guides the design of attention models, emphasizing the importance of training sequence and parameter selection. We propose a model design achieving full covariance-mean decoupling, offering a strategy for improving in-context learning. (see Theorem 4)

These contributions provide a deeper understanding of prompt-tuning and weight optimization, offering insights for designing more effective models that minimize loss and maximize performance.

## 2 Related work

Understanding in-context learning (ICL) in large language models (LLMs) has become a key research focus [Brown et al., 2020, Liu et al., 2023, Rae et al., 2021], particularly due to LLMs' ability to generalize across diverse applications [GeminiTeam et al., 2023, OpenAI, 2023, Touvron et al., 2023]. ICL enables models to adapt to new tasks using examples provided during inference without parameter updates, effectively serving as meta-learners. This has led to research exploring how LLMs leverage in-context information.

Several studies have linked ICL with gradient-based learning mechanisms. Akyürek et al. [2023] and Von Oswald et al. [2023] show that Transformers can emulate gradient descent (GD) steps using in-context examples, suggesting that Transformers implicitly learn gradient-based updates within their attention mechanisms.

Recent work has provided theoretical perspectives on ICL in simpler models like single-layer linear attention. Zhang et al. [2024], Mahankali et al. [2024], Ahn

et al. [2023], Li et al. [2023b] examine how these models can emulate GD-like algorithms when trained on in-context prompts. Mahankali et al. [2024] and Ahn et al. [2023] demonstrate that models trained on isotropic Gaussian data perform GD steps at test time, while Li et al. [2023a] explores generalization bounds for multi-layer Transformers. Li et al. [2024] extends this to dependent data with single-task ICL, showing that centralized data enables optimal preconditioned GD steps. However, these works primarily focus on zero-mean distributions or single parameters ( $\mathbf{W}$ ), simplifying the optimization landscape.

Our work addresses multi-task ICL with non-zero means and varying covariances, expanding beyond zero-mean assumptions in prior studies [Li et al., 2023a, 2024]. We explore the joint optimization of  $\mathbf{W}$ ,  $\mathbf{p}$ , and  $\mathbf{h}$  parameters, introducing task-specific structures that reduce the influence of task-specific means. Unlike Li et al. [2024] who focus on single-task scenarios or Li et al. [2023a] who assume a single mean and covariance, we analyze diverse task distributions with distinct means and covariances, developing the novel concept of "mean-covariance decoupling" to reveal how task-specific parameters enhance ICL performance. Our approach provides theoretical guarantees and empirical evidence for prompt-tuning in complex, multi-task environments with real-world non-zero mean distributions.

### 3 Setup and Preliminaries

We begin with a brief note on notation. Let  $[n]$  denote set  $\{1, \dots, n\}$  for some integer  $n$ . Bold lowercase and uppercase letters (e.g.,  $\mathbf{x}$  and  $\mathbf{X}$ ) represent vectors and matrices, respectively.  $\mathbf{1}_d$  and  $\mathbf{0}_d$  refer to the  $d$ -dimensional all-ones and all-zeros vectors, respectively, while  $\mathbf{I}_d$  denotes the  $d \times d$  identity matrix. Additionally,  $\text{tr}(\mathbf{W})$  represents the trace of the square matrix  $\mathbf{W}$ .

Our results are presented in a finite-dimensional setting.

#### 3.1 In-context learning

We consider an in-context learning (ICL) problem with demonstrations  $(\mathbf{x}_i, y_i)_{i=1}^{n+1}$ , and the input sequence  $\mathbf{Z}$  is defined by removing  $y_{n+1}$  as follows:

$$\begin{aligned} \mathbf{Z} &= [\mathbf{z}_1 \dots \mathbf{z}_n \mathbf{z}]^\top = \begin{bmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_n & \mathbf{x} \end{bmatrix}^\top \\ &= \begin{bmatrix} \mathbf{X}^\top & \mathbf{x} \\ \mathbf{y}^\top & 0 \end{bmatrix}^\top \in \mathbb{R}^{(n+1) \times (d+1)}. \end{aligned} \quad (3)$$

Here,  $\mathbf{z} = [\mathbf{x}^\top \ 0]^\top$  is the query token where  $\mathbf{x} := \mathbf{x}_{n+1}$ , and  $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_n]^\top \in \mathbb{R}^{n \times d}$ ,  $\mathbf{y} = [y_1 \dots y_n]^\top \in \mathbb{R}^n$ . Then, we aim for a sequence model to predict the associated label  $y := y_{n+1}$  of the given input sequence  $\mathbf{Z}$ .

In this work, we consider the following data generation of  $(\mathbf{Z}, y)$ . We will refer to  $(\mathbf{X}, \mathbf{y})$ ,  $\mathbf{x}$ , and  $y$  as contexts, query feature and the label to predict, respectively.

**Definition 1 (Single-task ICL)** *Given a task mean  $\boldsymbol{\mu} \in \mathbb{R}^d$ , and covariances  $\boldsymbol{\Sigma}_\mathbf{x}, \boldsymbol{\Sigma}_\beta \succ 0 \in \mathbb{R}^{d \times d}$ . The input sequence and its associated label, i.e.,  $(\mathbf{Z}, y)$  with  $\mathbf{Z}$  denoted in (3), are generated as follows:*

- A task parameter  $\beta$  is generated from a Gaussian prior  $\beta \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}_\beta)$ .
- Conditioned on  $\beta$ , for  $i \in [n+1]$ ,  $(\mathbf{x}_i, y_i)$  is generated by  $\mathbf{x}_i \sim \mathcal{N}(0, \boldsymbol{\Sigma}_\mathbf{x})$  and  $y_i \sim \mathcal{N}(\mathbf{x}_i^\top \beta, \sigma^2)$ .

Here,  $\sigma \geq 0$  is the noise level.

In this work, we study the task-mixture ICL problem where the task parameter  $\beta$  of each input sequence is sampled from  $K$  different distributions,  $K \geq 1$ .

**Definition 2 (Multi-task ICL)** *Consider a multi-task ICL problem with  $K$  different tasks. Each task generates  $(\mathbf{Z}, y) \sim \mathcal{D}_k$  following Definition 1 using shared feature distribution  $\mathbf{x}_i \sim \mathcal{N}(0, \boldsymbol{\Sigma}_\mathbf{x})$ ,  $i \in [n+1]$  but distinct task distributions  $\beta_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_{\beta_k})$  with mean  $\boldsymbol{\mu}_k$  and covariance  $\boldsymbol{\Sigma}_{\beta_k}$  for  $k \in [K]$ .*

Additionally, let  $\{\pi_k\}_{k=1}^K$  be the probabilities of each task, satisfying  $\sum_{k=1}^K \pi_k = 1$  and  $\pi_k \geq 0$ .

We consider a task-aware multi-task ICL setting. Specifically, when a task is selected according to  $\pi_k$ , its task index  $k$  is known.

Let  $\bar{\mathcal{D}} := \sum_{k=1}^K \pi_k \mathcal{D}_k$  be the mixture of distributions and given sequence model  $f : \mathbb{R}^{(n+1) \times (d+1)} \rightarrow \mathbb{R}$ , we define the multi-task ICL objective as follows:

$$\mathcal{L}(f) = \mathbb{E}_{(\mathbf{Z}, y) \sim \bar{\mathcal{D}}}[(y - f(\mathbf{Z}))^2]. \quad (4)$$

Notably, the multi-task ICL defined in Definition 2 differs from conventional multi-task learning [Caruana, 1997, Zhang and Yang, 2021, Li and Oymak, 2023] where finite examples optimize task-specific parameters. We instead parameterize task distributions with  $\boldsymbol{\mu}_k$  and  $\boldsymbol{\Sigma}_{\beta_k}$ , sampling unseen test tasks  $\beta_k$  and focusing on distribution-level generalization. We address the meta-learning objective in (4), treating each distribution as a meta-learning problem (c.f. Definition 1).

#### 3.2 Single-layer linear attention

Considering the task-aware multi-task ICL problem as described in Section 3.1, we explore the benefits of using *task-specific prompts* to enhance the performance.

**Definition 3 (Task-specific prompts)** *Recap input sequence  $\mathbf{Z}$  from (3). Given a task index  $k$ ,*

$k \in [K]$ , let  $\mathbf{p}_k \in \mathbb{R}^{d+1}$  represent its corresponding trainable prompt token. Then the input sequence of task  $k$  is denoted by:

$$\mathbf{Z}^{(k)} = [\mathbf{p}_k \ \mathbf{z}_1 \ \dots \ \mathbf{z}_n \ \mathbf{z}]^\top \in \mathbb{R}^{(n+2) \times (d+1)}. \quad (5)$$

Our work focuses on the single-layer linear attention model in solving multi-task ICL problem with data distribution following Definition 2. Given an input sequence  $\mathbf{Z}^{(k)}$  corresponding to task  $k$  as defined in (5), let  $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{(d+1) \times (d+1)}$  denote the query, key, and value parameters. Then the single-layer linear attention model outputs

$$\text{Attn}(\mathbf{Z}^{(k)}) = (\mathbf{Z}^{(k)} \mathbf{W}_q \mathbf{W}_k^\top (\mathbf{Z}^{(k)})^\top) \mathbf{M} \mathbf{Z}^{(k)} \mathbf{W}_v$$

where  $\text{Attn}(\cdot) : \mathbb{R}^{(n+2) \times (d+1)} \rightarrow \mathbb{R}^{(n+2) \times (d+1)}$ . Here, inspired by the previous work [Ahn et al., 2023], we apply mask matrix  $\mathbf{M} = \begin{bmatrix} \mathbf{I}_{n+1} & \mathbf{0}_{n+1} \\ \mathbf{0}_{n+1}^\top & 0 \end{bmatrix}$  to separate the  $(n+1)$ -colomun context and the query  $\mathbf{z}$  in  $\mathbf{Z}^{(k)}$ . Let  $\mathbf{h} \in \mathbb{R}^{d+1}$  be the linear head that enables the single-layer linear attention model to map the input sequence to the prediction. Additionally, for simplification and without loss of generality, let  $\mathbf{A} := \mathbf{W}_q \mathbf{W}_k^\top$  and  $\mathbf{a} := \mathbf{W}_v \mathbf{h}$ . Then the prediction returns

$$\hat{y} := f_{\text{Attn}}(\mathbf{Z}^{(k)}) = (\mathbf{z}^\top \mathbf{A} (\mathbf{Z}^{(k)})^\top) \mathbf{M} \mathbf{Z}^{(k)} \mathbf{a}. \quad (6)$$

This is a more general form than the widely discussed Wu et al. [2023] case, which uses the bottom-right entry of the attention layer output as prediction (and equivalent to a one-hot head  $\mathbf{h} = \mathbf{e}_{d+1}$ ), i.e.,  $f_{\text{Attn}}(\mathbf{Z}^{(k)}) = \text{Attn}(\mathbf{Z}^{(k)})_{n+2, d+1}$ .

### 3.3 Optimizing the attention model

Our goal is to understand how optimizing  $f_{\text{Attn}}$  in (6) results in in-context learning. To this aim, we introduce the following widely applied Wu et al. [2023], Ahn et al. [2023] assumption on the model construction.

**Assumption 1 (Preconditioning)** Given parameters  $\mathbf{A} := \mathbf{W}_q \mathbf{W}_k^\top$  and  $\mathbf{a} := \mathbf{W}_v \mathbf{h}$ , they are constrained by

$$\mathbf{A} = \begin{bmatrix} \mathbf{W}_{d \times d} & \mathbf{0}_{d \times 1} \\ *_{1 \times d} & *_{1 \times 1} \end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix} \mathbf{0}_{d \times 1} \\ 1_{1 \times 1} \end{bmatrix}.$$

Here, we use  $*$  to fill the entries that do not affect the final prediction, with subscripts indicating the dimensions.  $\mathbf{W} \in \mathbb{R}^{d \times d}$  represents the parameter that governs  $\mathbf{A}$ .

Under Assumption 1, let the prompt token for task  $k$  be  $\mathbf{p}_k = \begin{bmatrix} \bar{\mathbf{p}}_k \\ 1 \end{bmatrix}$ , hence  $\mathbf{Z}^{(k)} = \begin{bmatrix} \bar{\mathbf{p}}_k & \mathbf{X}^\top & \mathbf{x} \\ 1 & \mathbf{y}^\top & 0 \end{bmatrix}^\top$ . The

prediction of a single-layer linear attention model in (6) can then be written as:

$$\begin{aligned} f_{\text{Attn}}(\mathbf{Z}^{(k)}) &= \mathbf{x}^\top \mathbf{W} [\bar{\mathbf{p}}_k \quad \mathbf{X}^\top] \begin{bmatrix} 1 \\ \mathbf{y} \end{bmatrix} \\ &= \mathbf{x}^\top \mathbf{W} (\mathbf{X}^\top \mathbf{y} + \bar{\mathbf{p}}_k). \end{aligned} \quad (7)$$

It is worth noting that we set the last entry of  $\mathbf{a}$  and each  $\mathbf{p}_k$  for  $k \in [K]$  to one for simplicity, as any nonzero value yields the same output of (7) due to rescaling invariance of:  $\mathbf{a} \leftarrow \gamma_1 \mathbf{a}$ ,  $\mathbf{A} \leftarrow \gamma_1^{-1} \mathbf{A}$ , and  $\bar{\mathbf{p}}_k \leftarrow \begin{bmatrix} \gamma_2^{-1} \bar{\mathbf{p}}_k \\ \gamma_2 \end{bmatrix}$  for any nonzero scalar  $\gamma_1, \gamma_2$ .

We define the set of tunable prompts as:

$$\mathbf{P} = [\bar{\mathbf{p}}_1 \ \dots \ \bar{\mathbf{p}}_K]^\top \in \mathbb{R}^{K \times d}. \quad (8)$$

Notably, previous research Ahn et al. [2023] has rigorously proven that for single-layer linear attention applied to a single-task linear regression ICL problem with zero-mean features, i.e.,  $\mathbb{E}[\mathbf{x}_i] = \mathbb{E}[\beta] = \mathbf{0}$ , the optimal solution must conform to the structure specified in Assumption 1. This insight motivates our adoption of this assumption when extending the analysis to more complex multi-task ICL settings with non-zero task means. Building on this foundation, our work breaks new ground by exploring task-specific tuning for ICL across multiple tasks with varying means, offering a more general perspective on ICL. This represents a significant advancement in the field, as understanding the loss landscape in the simpler  $\mathbf{W}$ -preconditioned space is an essential step toward tackling the complexities of the full  $\mathbf{W}_k, \mathbf{W}_q, \mathbf{W}_v$  parameter space.

Additionally, in Section 5, we show that it is possible to derive closed-form optimal solutions for  $\mathbf{W}_k, \mathbf{W}_q, \mathbf{W}_v$  with both task-specific prompts and heads, even without relying on Assumption 1, further expanding the scope of our analysis.

Recall the attention predictor from (7) and loss function from (4). Consider the multi-task ICL problem defined in Definition 2 and let  $\mathbf{W} \in \mathbb{R}^{d \times d}$  and  $\mathbf{P} \in \mathbb{R}^{K \times d}$  be the tunable parameters corresponding to attention weights and task-specific prompt tokens. The multi-task ICL loss is given by

$$\mathcal{L}(\mathbf{W}, \mathbf{P}) = \sum_{k=1}^K \pi_k \mathcal{L}_k(\mathbf{W}, \bar{\mathbf{p}}_k) \quad (9)$$

$$\text{where } \mathcal{L}_k(\mathbf{W}, \bar{\mathbf{p}}_k) = \mathbb{E}_{(\mathbf{Z}, y) \sim \mathcal{D}_k} [(f_{\text{Attn}}(\mathbf{Z}^{(k)}) - y)^2].$$

In this work, we address multi-task ICL problems by investigating and comparing three optimization settings: plain training, fine-tuning, and joint training.

**Plain training:** Plain training refers to a standard ICL problem that train an linear attention model without applying the task-specific prompts that are defined in Definition 3. Therefore, following loss function (9), its objective can be defined via

$$\mathbf{W}_{\text{PT}}^* = \arg \min_{\mathbf{W}} \mathcal{L}(\mathbf{W}, \mathbf{P} = \mathbf{0}), \quad (10)$$

and  $\mathcal{L}_{\text{PT}}^* = \mathcal{L}(\mathbf{W}_{\text{PT}}^*, \mathbf{P} = \mathbf{0})$  is the optimal loss.

**Fine-tuning:** Fine-tuning/Prompt-tuning involves training separate prompts for each task while keeping the attention weights fixed. The goal then is to fine-tune the prompt parameters  $\mathbf{P}$  (c.f. (8)) for all the tasks  $k \in [K]$ . Suppose that parameter  $\mathbf{W}$  is given, the the optimal  $\mathbf{P}$  based on  $\mathbf{W}$  is defined by:

$$\mathbf{P}^*(\mathbf{W}) = \arg \min_{\mathbf{P}} \mathcal{L}(\mathbf{W}, \mathbf{P}).$$

In this work, we consider fine-tuning based on the plain pretrained model, that is, by setting  $\mathbf{W} = \mathbf{W}_{\text{PT}}^*$  given in (10), and define the optimal solution by

$$\mathbf{P}_{\text{FT}}^* := \mathbf{P}^*(\mathbf{W}_{\text{PT}}^*) = \arg \min_{\mathbf{P}} \mathcal{L}(\mathbf{W}_{\text{PT}}^*, \mathbf{P}). \quad (11)$$

The optimal loss is given via  $\mathcal{L}_{\text{FT}}^* = \mathcal{L}(\mathbf{W}_{\text{PT}}^*, \mathbf{P}_{\text{FT}}^*)$ .

**Joint training:** In contrast, joint training involves jointly optimizing the attention weights  $\mathbf{W}$  and prompt tokens  $\mathbf{P}$ . Hence, the optimization problem can be formulated as:

$$\mathbf{W}_{\text{JT}}^*, \mathbf{P}_{\text{JT}}^* = \arg \min_{\mathbf{W}, \mathbf{P}} \mathcal{L}(\mathbf{W}, \mathbf{P}). \quad (12)$$

The optimal loss is given via  $\mathcal{L}_{\text{JT}}^* = \mathcal{L}(\mathbf{W}_{\text{JT}}^*, \mathbf{P}_{\text{JT}}^*)$ .

## 4 Main Results

In this section, we train and optimize the single-layer linear attention model in a multi-task linear regression ICL setting with dataset described in Definition 2, and characterize the loss landscape under different settings, i.e.,  $\mathcal{L}_{\text{PT}}^*$ ,  $\mathcal{L}_{\text{FT}}^*$  and  $\mathcal{L}_{\text{JT}}^*$  in Section 3.3.

### 4.1 Optimization landscape

Recap the multi-task ICL dataset from Definition 2 where  $\Sigma_{\mathbf{x}}$  is the shared covariance matrix of the input features and  $\{(\mu_k, \Sigma_{\beta_k})\}_{k=1}^K$  are the task mean vectors and covariance matrices. In the main paper, we consider noiseless data setting where  $\sigma = 0$ . We defer the exact analysis considering noisy labels to Appendix.

Consider any data distribution in Definition 1 and let  $\beta \sim \mathcal{N}(\mu, \Sigma_{\beta})$ .  $\beta$  can be rewritten via  $\beta = \bar{\beta} + \mu$  with  $\bar{\beta} \sim \mathcal{N}(0, \Sigma_{\beta})$ . Then under the noiseless setting ( $\sigma = 0$ ), the associated label  $y_i = \mathbf{x}_i^\top \beta$  is generated via

$$y_i = \underbrace{\mathbf{x}_i^\top \bar{\beta}}_{\text{debiased}} + \mathbf{x}_i^\top \mu. \quad (13)$$

Here we describe  $\mathbf{x}_i^\top \bar{\beta}$  as debiased since  $\mathbb{E}[\mathbf{x}_i^\top \bar{\beta}] = 0$ . In this work, we investigate how task-specific prompts can help to capture individual task means such that learning task means ( $\{\mu_k\}_{k=1}^K$ ) and covariances ( $\{\Sigma_{\beta_k}\}_{k=1}^K$ ) can be decoupled via optimizing prompts  $\mathbf{P}$  and the attention weight  $\mathbf{W}$ . We say the model *fully decouples* mean and covariance if the optimized attention weight  $\mathbf{W}^*$  is only determined by the debiased term as shown in (13), with prompts responsible for capturing the bias introduced by the non-zero means.

To start with, recap from Definition 2 where task  $k$  has probability  $\pi_k$  and its task vector follows distribution  $\beta_k \sim \mathcal{N}(\mu_k, \Sigma_{\beta_k})$ . Following (13), we define the debiased and biased mixed-task covariances (variant with  $\Sigma_{\mathbf{x}}$  prior) as follows:

$$\text{Debiased: } \bar{\Sigma}_{\beta} = \Sigma_{\mathbf{x}} \sum_{k=1}^K \pi_k \mathbb{E}[(\beta_k - \mu_k)(\beta_k - \mu_k)^\top]; \quad (14a)$$

$$\text{Biased: } \tilde{\Sigma}_{\beta} = \Sigma_{\mathbf{x}} \sum_{k=1}^K \pi_k \mathbb{E}[\beta_k \beta_k^\top]. \quad (14b)$$

Note that they satisfy  $\bar{\Sigma}_{\beta} = \Sigma_{\mathbf{x}} \sum_{k=1}^K \pi_k \Sigma_{\beta_k}$  and  $\tilde{\Sigma}_{\beta} = \Sigma_{\mathbf{x}} \sum_{k=1}^K \pi_k (\Sigma_{\beta_k} + \mu_k \mu_k^\top)$ .

We first analyze the plain training setting where no additional task-specific parameters are introduced, and all  $K$  tasks are mixed together.

**Theorem 1 (Plain training)** Consider training a single-layer linear attention model in solving multi-task ICL problem with dataset defined in Definition 2 and model construction as described in Assumption 1. Let the optimal solution  $\mathbf{W}_{\text{PT}}^*$  (c.f. (10)) and the minimal plain training loss  $\mathcal{L}_{\text{PT}}^*$  as defined in Section 3.3. Additionally, let  $\bar{\Sigma}_{\beta}$  be defined in (14) and  $\bar{\mathbf{W}}_{\text{PT}}^* = \Sigma_{\mathbf{x}} \mathbf{W}_{\text{PT}}^*$ . Then the solution  $\bar{\mathbf{W}}_{\text{PT}}^*$  and optimal loss  $\mathcal{L}_{\text{PT}}^*$  satisfy

$$\bar{\mathbf{W}}_{\text{PT}}^* = \bar{\Sigma}_{\beta} \left( (n+1) \bar{\Sigma}_{\beta} + \text{tr}(\bar{\Sigma}_{\beta}) \mathbf{I} \right)^{-1},$$

$$\mathcal{L}_{\text{PT}}^* = \text{tr}(\bar{\Sigma}_{\beta}) - n \text{tr}(\bar{\mathbf{W}}_{\text{PT}}^* \bar{\Sigma}_{\beta}).$$

Note that the above solution and optimal loss are identical to those in previous work [Li et al., 2024] when considering a single-task ICL problem with task vector following distribution  $\beta \sim \mathcal{N}(0, \bar{\Sigma}_{\beta})$ .

**Theorem 2 (Fine-tuning)** Suppose a pretrained model as described in Theorem 1 is given with  $\mathbf{W}_{\text{PT}}^*$  being its optimal solution. Consider fine-tuning this model with task-specific prompts as defined in Definition 3, and let the optimal prompt matrix  $\mathbf{P}_{\text{FT}}^*$  (c.f. (11)) and the minimal fine-tuning loss  $\mathcal{L}_{\text{FT}}^*$  be defined in Section 3.3. Additionally, let  $\bar{\Sigma}_{\beta}, \tilde{\Sigma}_{\beta}$  be defined in (14) and  $\bar{\mathbf{W}}_{\text{PT}}^* = \Sigma_{\mathbf{x}} \mathbf{W}_{\text{PT}}^*$ , and define the

mean matrix

$$\mathbf{M}_\mu = [\boldsymbol{\mu}_1 \ \cdots \ \boldsymbol{\mu}_K]^\top \in \mathbb{R}^{K \times d}.$$

Then the solution  $\mathbf{P}_{\text{FT}}^*$  and optimal loss  $\mathcal{L}_{\text{FT}}^*$  satisfy

$$\begin{aligned} \mathbf{P}_{\text{FT}}^* &= \mathbf{M}_\mu ((\bar{\mathbf{W}}_{\text{PT}}^*)^{-1} - n\mathbf{I}) \boldsymbol{\Sigma}_x, \\ \mathcal{L}_{\text{FT}}^* &= \mathcal{L}_{\text{PT}}^* - \text{tr}((\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} - \bar{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}})(n\bar{\mathbf{W}}_{\text{PT}}^* - \mathbf{I})^\top(n\bar{\mathbf{W}}_{\text{PT}}^* - \mathbf{I})). \end{aligned}$$

Results in Theorem 2 show that, fine-tuning achieves better loss than plain training,  $\mathcal{L}_{\text{FT}}^* \leq \mathcal{L}_{\text{PT}}^*$ , and our results provably quantize the loss difference.

**Theorem 3 (Joint training)** Consider training a single-layer linear attention model in solving multi-task ICL problem with dataset defined in Definition 2 and model construction as described in Assumption 1. Let  $\mathbf{W}_{\text{JT}}^*$ ,  $\mathbf{P}_{\text{JT}}^*$  (c.f. (12)) be the optimal solutions and  $\mathcal{L}_{\text{JT}}^*$  is the optimal joint training loss defined in Section 3.3. Additionally, let  $\bar{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$ ,  $\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$ ,  $\mathbf{M}_\mu$  follow the same definitions as in Theorem 2 and define  $\bar{\mathbf{W}}_{\text{JT}}^* = \boldsymbol{\Sigma}_x \mathbf{W}_{\text{JT}}^*$ . Then the solution  $(\mathbf{W}_{\text{JT}}^*, \mathbf{P}_{\text{JT}}^*)$  and optimal loss  $\mathcal{L}_{\text{JT}}^*$  satisfy

$$\begin{aligned} \bar{\mathbf{W}}_{\text{JT}}^* &= \bar{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} \left( (n+1)\bar{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} + \text{tr}(\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}})\mathbf{I} + \mathcal{O}(1) \right)^{-1}, \\ \mathbf{P}_{\text{JT}}^* &= \mathbf{M}_\mu ((\bar{\mathbf{W}}_{\text{JT}}^*)^{-1} - n\mathbf{I}) \boldsymbol{\Sigma}_x, \\ \mathcal{L}_{\text{JT}}^* &= \text{tr}(\bar{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}) - n \text{tr}(\bar{\mathbf{W}}_{\text{JT}}^* \bar{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}). \end{aligned}$$

Here,  $\mathcal{O}(1)$  is a  $d \times d$ -sized matrix with entries being bounded by some constant value, regardless of  $n$  and  $d$ .

In Theorem 3, for clarity, we use  $\mathcal{O}(1)$  to represent equality up to a matrix with entries bounded by a constant. The explicit form of  $\bar{\mathbf{W}}_{\text{JT}}^*$  is provided in the Appendix.

The results of Theorem 2 and Theorem 3 highlight an important commonality of the optimal prompt: the optimal prompts  $\bar{\mathbf{p}}_k$  capture the mean of corresponding tasks based on the attention weight  $\mathbf{W}_{\text{PT}}^*$  or  $\mathbf{W}_{\text{JT}}^*$ . For a large context length  $n$ ,  $(\bar{\mathbf{p}}_k)_{\text{FT}}^*$  and  $(\bar{\mathbf{p}}_k)_{\text{JT}}^*$  will be approximately  $-n\boldsymbol{\Sigma}_x \boldsymbol{\mu}_k$ . Additionally, in a finite-dimensional setting where  $d < \infty$ , as  $n \rightarrow \infty$ , the solutions  $\bar{\mathbf{W}}_{\text{PT}}^*$  and  $\bar{\mathbf{W}}_{\text{JT}}^*$  converge to  $\mathbf{I}/n$  and all the optimal losses  $\mathcal{L}_{\text{PT}}^*$ ,  $\mathcal{L}_{\text{FT}}^*$  and  $\mathcal{L}_{\text{JT}}^*$  approach 0. Therefore, the benefits of prompt tuning are more apparent for finite  $n$ .

**Corollary 1** Let  $\mathcal{L}_{\text{PT}}^*$ ,  $\mathcal{L}_{\text{FT}}^*$ , and  $\mathcal{L}_{\text{JT}}^*$  denote the optimal losses for plain training, fine-tuning, and joint training, as described in Theorems 1, 2 and 3, respectively. These losses satisfy:

$$\mathcal{L}_{\text{JT}}^* \leq \mathcal{L}_{\text{FT}}^* \leq \mathcal{L}_{\text{PT}}^*. \quad (15)$$

The equalities hold if and only if  $\bar{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} = \tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$  (c.f. (14)), which occurs when all task means  $\boldsymbol{\mu}_k = \mathbf{0}$  for  $k \in [K]$ . Furthermore, the loss gaps satisfy the following:

1. The loss gaps scale quadratically with task mean:

$$\mathcal{L}_{\text{PT}}^* - \mathcal{L}_{\text{FT}}^* \sim \mathcal{O}\left(\frac{1}{n^2}\right) \|\Delta\|_F, \quad \mathcal{L}_{\text{FT}}^* - \mathcal{L}_{\text{JT}}^* \sim \mathcal{O}\left(\frac{1}{n}\right) \|\Delta\|_F,$$

$$\text{where } \Delta := \tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} - \bar{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} = \boldsymbol{\Sigma}_x \sum_{k=1}^K \pi_k \boldsymbol{\mu}_k \boldsymbol{\mu}_k^\top.$$

2. The ratio between gaps is:  $\frac{\mathcal{L}_{\text{PT}}^* - \mathcal{L}_{\text{FT}}^*}{\mathcal{L}_{\text{FT}}^* - \mathcal{L}_{\text{JT}}^*} \sim \mathcal{O}\left(\frac{1}{n}\right)$ , indicating that fine-tuning provides most of the benefit in few-shot regimes (small  $n$ ), while joint training benefits more for larger  $n$ .

## 4.2 Covariance-mean decoupling

For a single-layer linear attention model under Assumption 1, where  $\mathbf{W} \in \mathbb{R}^{d \times d}$  captures all the statistics, the plain training loss  $\mathcal{L}_{\text{PT}}^*$  is determined by the second-order moment of the task parameters  $\boldsymbol{\beta}_k$ :

$$\mathbb{E}[\boldsymbol{\beta}_k \boldsymbol{\beta}_k^\top] = \boldsymbol{\Sigma}_{\boldsymbol{\beta}_k} + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^\top,$$

which represents the biased covariance of task  $k$ . Consequently,  $\mathcal{L}_{\text{PT}}^*$  can be viewed as a function of this biased variable, defined as  $\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$  (see Theorem 1), which affects both the terms in  $\bar{\mathbf{W}}_{\text{PT}}^*$  and those that appear directly in  $\mathcal{L}_{\text{PT}}^*$  but outside of  $\bar{\mathbf{W}}_{\text{PT}}^*$ .

When all tasks have zero mean, i.e.,  $\mathbb{E}[\boldsymbol{\beta}_k] = \boldsymbol{\mu}_k = \mathbf{0}$  for all  $k \in [K]$ , leading to  $\bar{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} = \mathbf{0}_{d \times d}$ , the optimal losses for pretraining, fine-tuning, and joint training become identical:

$$\boldsymbol{\mu}_k = \mathbf{0}, \quad k \in [K] \Rightarrow \mathcal{L}_{\text{PT}}^* = \mathcal{L}_{\text{FT}}^* = \mathcal{L}_{\text{JT}}^*. \quad (16)$$

In this case, the fine-tuned prompts remain as zero vectors, learning nothing during fine-tuning (as shown in Theorem 2 and Theorem 3). This implies that any differences in loss arise from the ability of the trainable prompts to **decouple** (i.e., remove task-mean related bias terms) from  $\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$  to obtain  $\bar{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$ . Specifically, when all task means are zero, this decoupling is nullified, resulting in no difference in losses across the training methods.

When the task means  $\mathbb{E}[\boldsymbol{\beta}_k] = \boldsymbol{\mu}_k$  are non-zero, the decoupling effect varies between different training settings. In joint training, the simultaneous optimization of prompts and attention weights enables decoupling in both  $\mathcal{L}_{\text{JT}}^*$  and  $\mathbf{W}_{\text{JT}}^*$ , while in fine-tuning, only the biased terms in  $\mathbf{W}_{\text{FT}}^*$  are decoupled. As a result, joint training achieves greater decoupling than fine-tuning. According to Corollary 1, when a loss is more directly influenced by the biased covariance  $\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$ , it tends to be higher, whereas reducing the influence of  $\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$  through decoupling generally leads to a lower loss.

These findings suggest that introducing additional task-specific trainable parameters into the single-layer linear attention model, combined with joint optimization, can effectively reduce the bias in mixed-task covariance, thereby improving performance.

## 5 Fully-decoupled Loss

In Section 4, we focus on cases where model parameters are constructed according to Assumption 1 and analyze the loss landscapes for plain training, finetuning, and joint training. However, our results indicate that with a shared head  $\mathbf{a} := \mathbf{W}_v \mathbf{h}$ , none of these methods fully decouple the mean and covariance. To address this, we introduce an alternative approach that allows each task to have its own specific linear prediction head  $\mathbf{h}_k \in \mathbb{R}^{d+1}$ . It is worth noting that using separate heads for different tasks is a common practice in the general multi-task learning literature [Caruana, 1997, Zhang and Yang, 2021, Li and Oymak, 2023]. Our results demonstrate that optimizing task-specific prompts, heads, and attention weights leads to a fully decoupled loss (Theorem 4).

**Definition 4 (Task-specific heads)** *Given  $K$  tasks, let  $\{\mathbf{h}_k\}_{k=1}^K \subset \mathbb{R}^{d+1}$  represent their corresponding trainable linear prediction heads. Recalling the input sequence and prediction from (5) and (6), the prediction for task  $k$  returns*

$$\tilde{f}_{\text{Attn}}(\mathbf{Z}^{(k)}) = (\mathbf{z}^\top \mathbf{W}_q \mathbf{W}_k^\top (\mathbf{Z}^{(k)})^\top) \mathbf{M} \mathbf{Z}^{(k)} \mathbf{W}_v \mathbf{h}_k. \quad (17)$$

Recap ICL problem from Definition 2,  $\mathbf{Z}^{(k)}$  from Definition 3 and loss function from (4), the ICL objective considering task-specific prompts and heads is:

$$\tilde{\mathcal{L}}_{\text{Attn}}^* = \min_{(\mathbf{p}_k, \mathbf{h}_k)_{k=1}^K, \mathbf{W}_{k,q,v}} \mathcal{L}(\tilde{f}_{\text{Attn}}) \quad (18)$$

$$\text{where } \mathcal{L}(\tilde{f}_{\text{Attn}}) = \sum_{k=1}^K \pi_k \mathbb{E}_{\mathbf{Z}, y \sim \mathcal{D}_k} [(\tilde{f}_{\text{Attn}}(\mathbf{Z}^{(k)}) - y)^2].$$

Here, the search space for  $\mathbf{p}_k, \mathbf{h}_k$  is  $\mathbb{R}^{d+1}$  and the search space for  $\mathbf{W}_{k,q,v}$  is  $\mathbb{R}^{(d+1) \times (d+1)}$ .

Next, given task  $k$  with mean  $\boldsymbol{\mu}_k$ , we introduce debiased preconditioned gradient descent (PGD) predictor as follows:

$$\tilde{f}_{\text{PGD}}(\mathbf{Z}^{(k)}) = \mathbf{x}^\top \mathbf{W} \mathbf{X}^\top (\mathbf{y} - \mathbf{X} \boldsymbol{\mu}_k) + \mathbf{x}^\top \boldsymbol{\mu}_k.$$

Note that for any task  $k$ , we have  $\mathbb{E}[y_i - \mathbf{x}_i^\top \boldsymbol{\mu}_k] = 0$ , and therefore, we expect  $\tilde{f}_{\text{PGD}}(\mathbf{Z}^{(k)}) - \mathbf{x}^\top \boldsymbol{\mu}_k$  to predict unbiased label  $y - \mathbf{x}^\top \boldsymbol{\mu}_k$ . The corresponding PGD objective is defined as:

$$\tilde{\mathcal{L}}_{\text{PGD}}^* = \min_{\mathbf{W} \in \mathbb{R}^{d \times d}} \mathcal{L}(\tilde{f}_{\text{PGD}}) \quad (19)$$

$$\text{where } \mathcal{L}(\tilde{f}_{\text{PGD}}) := \sum_{k=1}^K \pi_k \mathbb{E}_{\mathbf{Z}, y \sim \mathcal{D}_k} [(\tilde{f}_{\text{PGD}}(\mathbf{Z}^{(k)}) - y)^2].$$

The following proposition establishes the equivalence between optimizing single layer linear attention (c.f. (18)) and one step of PGD predictor (c.f. (19)).

**Proposition 1** *Consider the multi-task ICL data as described in Definition 2 and let  $\tilde{\mathcal{L}}_{\text{Attn}}^*$  and  $\tilde{\mathcal{L}}_{\text{PGD}}^*$  be the optimal linear attention and debiased preconditioned gradient descent losses as presented in (18) and (19), respectively. Then,  $\tilde{\mathcal{L}}_{\text{Attn}}^* = \tilde{\mathcal{L}}_{\text{PGD}}^*$ .*

Considering the single-task and zero-mean setting, Proposition 1 aligns with the findings of previous work [Ahn et al., 2023, Li et al., 2024]. Our results further confirm the necessity of both task-specific prompts and heads in effectively decoupling the influence of non-zero means from the data.

Then, based on the Proposition 1, we are able to analyze the optimization landscape of (18) via studying (19).

**Theorem 4** *Consider the multi-task ICL problem with dataset defined in Definition 2. Let  $\mathbf{W}_{\text{PGD}}^* := \arg \min_{\mathbf{W}} \mathcal{L}(\tilde{f}_{\text{PGD}})$  following (19). Define  $\bar{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}$  in (14) and let  $\bar{\mathbf{W}}_{\text{PGD}}^* = \sum_{\mathbf{x}} \mathbf{W}_{\text{PGD}}^*$ . Then the solution  $\bar{\mathbf{W}}_{\text{PGD}}^*$  and optimal loss  $\tilde{\mathcal{L}}_{\text{PGD}}^*$  (c.f. (19)) satisfy*

$$\bar{\mathbf{W}}_{\text{PGD}}^* = \bar{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} ((n+1) \bar{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} + \text{tr}(\bar{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}) \mathbf{I})^{-1},$$

$$\tilde{\mathcal{L}}_{\text{PGD}}^* = \text{tr}(\bar{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}) - n \text{tr}(\bar{\mathbf{W}}_{\text{PGD}}^* \bar{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}).$$

See Appendix for a proof.

Comparing Theorem 4 with Theorem 1, it is evident that  $\tilde{\mathcal{L}}_{\text{PGD}}^*$  represents a fully decoupled loss, demonstrating the clear benefit of adding task-specific heads. While Theorem 1 provides an upper bound  $\mathcal{L}_{\text{PT}}^*$  for the multi-task ICL loss, the proof of Theorem 4 establishes that  $\tilde{\mathcal{L}}_{\text{PGD}}^*$  serves as the lower bound for a multi-task ICL loss in a single-layer linear attention model.

## 6 Experiments

We conduct experiments on synthetic datasets to validate our theoretical assumptions and explore the behavior of single-layer linear attention models with various trainable parameters under different training settings.

**Experimental Setting.** We train single-layer attention models to solve  $K$ -task,  $d$ -dimensional linear regression ICL in a noise-free meta-learning setup for consistency with the main paper’s theorems, deferring noisy results to the Appendix. For each context length  $n$ , an independent model is trained for 20,000 iterations with a batch size of 8192 using the Adam optimizer (learning rate  $10^{-3}$ ).

To ensure robustness, each training process is repeated 50 times with independent initializations, and the mini-

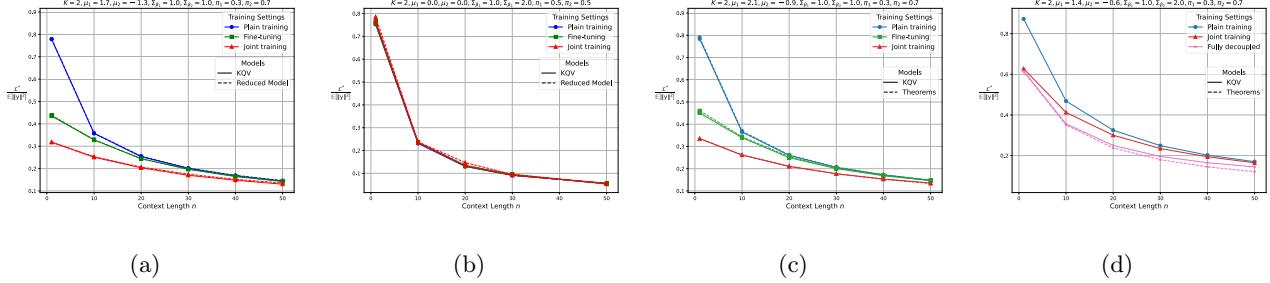


Figure 2: Experimental results across various settings: (a) Performance of unconstrained  $\mathbf{W}_k, \mathbf{W}_q, \mathbf{W}_v$ -parameterized linear attention model and reduced model, with non-zero task mean. (b) Performance of unconstrained  $\mathbf{W}_k, \mathbf{W}_q, \mathbf{W}_v$ -parameterized linear attention model and reduced model, with zero task mean. (c) Performance of unconstrained  $\mathbf{W}_k, \mathbf{W}_q, \mathbf{W}_v$ -parameterized linear attention model and theoretical prediction, with non-zero task mean. (d) Performance of unconstrained  $\mathbf{W}_k, \mathbf{W}_q, \mathbf{W}_v$ -parameterized linear attention model, with different numbers of task-specific trainable parameters.

mal test risk among these trials is reported. Theoretical predictions in the plots are based on the theorems in Section 4, and all results are normalized by  $\mathbb{E}[\|y\|^2]$ .

**Validation of the preconditioning.** In order to support Assumption 1 in Section 3.3, we train an unconstrained  $\mathbf{W}_k, \mathbf{W}_q, \mathbf{W}_v$ -parameterized model (with a shared head  $\mathbf{h}$ ) which is of an alternative form 6, as well as a reduced model 7 derived from Assumption 1, in order to check the alignment of their loss landscape. We configure the experiment with the following choice:

$$d = 10, K = 2, \mathbf{M}_\mu = [1.7 \cdot \mathbf{1}_{10} \quad -1.3 \cdot \mathbf{1}_{10}], \quad \Sigma_{\beta_1} = \Sigma_{\beta_2} = \mathbf{I}_{10}, \quad \pi_1 = 0.3, \quad \pi_2 = 0.7.$$

$$d = 10, K = 2, \mathbf{M}_\mu = [\mathbf{0}_{10} \quad \mathbf{0}_{10}], \quad \Sigma_{\beta_1} = \mathbf{I}_{10}, \Sigma_{\beta_2} = 2 \cdot \mathbf{I}_{10}, \quad \pi_1 = \pi_2 = 0.5.$$

As shown in Figure 2a, 2b, the alignment of the dashed and solid lines across the plain training, fine-tuning, and joint training settings suggests that Assumption 1 results in a simple yet effective reduced model. The performance of this reduced model aligns closely with that of optimizing a single-layer linear attention model without the constraint imposed by this assumption, indicating its reasonableness. We also validated that the performance of different training settings of the unconstrained model and the reduced model will be aligned given a zero task mean scenario, which is stated in Section 4.2.

Note that the joint optimization of the task-specific head, prompt, and model attention weights is not included in this experiment, although it could potentially support Theorem 4. When the task-specific head is included in the joint training, Assumption 1 is no longer needed to derive the reduced form, making validation of Assumption 1 under such settings unnecessary.

**Validation of the theorems.** Given the previous experimental results supporting Assumption 1, which in

turn validate Theorems 1, 2, and 3, these theorems are primarily supported by empirical evidence. Similarly, we train an unconstrained  $\mathbf{W}_k, \mathbf{W}_q, \mathbf{W}_v$ -parameterized model (with a shared head  $\mathbf{h}$ ) to assess the alignment between its loss landscape and our theoretical predictions from Theorems 1, 2, and 3. The experiments are configured with the following settings:

$$d = 10, K = 2, \mathbf{M}_\mu = [2.1 \cdot \mathbf{1}_{10} \quad -0.9 \cdot \mathbf{1}_{10}], \quad \Sigma_{\beta_1} = \Sigma_{\beta_2} = \mathbf{I}_{10}, \quad \pi_1 = 0.3, \pi_2 = 0.7.$$

As shown in Figure 2c, the alignment of the dashed and solid lines across the plain training, fine-tuning, and joint training settings suggests that our theoretical result can predict the multi-task linear regression ICL loss accurately.

**Benefits of Additional Task-Specific Parameters.** We assess the impact of adding task-specific heads to a single-layer linear attention model with task-specific prompts. Specifically, we compare the loss  $\tilde{\mathcal{L}}_{\text{Attn}}^*$  from jointly training  $\mathbf{W}, \mathbf{P}, \mathbf{H}$  to the loss  $\mathcal{L}_{\text{JT}}^*$  from jointly training  $\mathbf{W}, \mathbf{P}$ , using  $\mathcal{L}_{\text{PT}}^*$  as a baseline, where only  $\mathbf{W}$  is optimized. Theoretical curves are shown only for Theorem 4, as the others have been validated in Theorems 1, 2, and 3. The experiments are configured as follows:

$$d = 10, K = 2, \mathbf{M}_\mu = [1.4 \cdot \mathbf{1}_{10} \quad -0.6 \cdot \mathbf{1}_{10}], \quad \Sigma_{\beta_1} = \mathbf{I}_{10}, \Sigma_{\beta_2} = 2 \cdot \mathbf{I}_{10}, \quad \pi_1 = 0.3, \pi_2 = 0.7.$$

There is a clear, gradual improvement from adding more task-specific parameters, as shown in Figure 2d.

**Discussions.** We analyze the impact of task-specific parameters in multi-task ICL settings. Our work provides theoretical guarantees for joint training and pretrain  $\rightarrow$  finetune approaches. We introduce a covariance-mean decoupling mechanism for optimal ICL: Task-specific parameters learn the task mean prediction and attention weights learn the variance. Experimental results support our theoretical analysis.

## Acknowledgments

This work was partially supported under NSF grants CCF2046816, CCF-2403075, CCF-2008020, and the Office of Naval Research grants N000142412289 and N000141812252. Additionally, research was sponsored by the OUSD (R&E)/RT&L and was accomplished under Cooperative Agreement Number W911NF-20-2-0267. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the ONR, ARL and OUSD(R&E)/RT&L or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. This work was also partially supported by gifts from Open Philanthropy, Amazon Research, and Google Research.

## References

Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. *Advances in Neural Information Processing Systems*, 36, 2023.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=0g0X4H8yN4I>.

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

Rich Caruana. Multitask learning. *Machine learning*, 28:41–75, 1997.

GeminiTeam, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Yingcong Li and Samet Oymak. Provable pathways: Learning multiple tasks over multiple paths. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 8701–8710, 2023.

Yingcong Li, Muhammed Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. Transformers as algorithms: Generalization and stability in in-context learning. In *International Conference on Machine Learning*, pages 19565–19594. PMLR, 2023a.

Yingcong Li, Kartik Sreenivasan, Angeliki Giannou, Dimitris Papailiopoulos, and Samet Oymak. Dissecting chain-of-thought: Compositionality through in-context filtering and learning. *Advances in Neural Information Processing Systems*, 36:22021–22046, 2023b.

Yingcong Li, Ankit Singh Rawat, and Samet Oymak. Fine-grained analysis of in-context linear estimation: Data, architecture, and beyond. *arXiv preprint arXiv:2407.10005*, 2024.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

Arvind V. Mahankali, Tatsunori Hashimoto, and Tengyu Ma. One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=8p3fu561Kc>.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susanah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Johannes Von Oswald, Eyyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR, 2023.

Jingfeng Wu, Difan Zou, Zixiang Chen, Vladimir Braverman, Quanquan Gu, and Peter L Bartlett. How many pretraining tasks are needed for in-context learning of linear regression? *arXiv preprint arXiv:2310.08391*, 2023.

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

Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE transactions on knowledge and data engineering*, 34(12):5586–5609, 2021.

## Checklist

1. For all models and algorithms presented, check if you include:
  - (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
  - (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable]
  - (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]
2. For any theoretical claim, check if you include:
  - (a) Statements of the full set of assumptions of all theoretical results. [Yes]
  - (b) Complete proofs of all theoretical results. [Yes]
  - (c) Clear explanations of any assumptions. [Yes]
3. For all figures and tables that present empirical results, check if you include:
  - (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
  - (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
  - (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
  - (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Not Applicable]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
  - (a) Citations of the creator If your work uses existing assets. [Not Applicable]
  - (b) The license information of the assets, if applicable. [Not Applicable]
  - (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
  - (d) Information about consent from data providers/curators. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
  - (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

---

## Appendix

---

### Contents

---

<b>A Lemmas</b>	<b>1</b>
<b>B Proofs for Section 4</b>	<b>4</b>
B.1 Proof of Theorem 1 . . . . .	5
B.2 Proof of Theorem 2 . . . . .	7
B.3 Proof of Theorem 3 . . . . .	9
B.4 Proof of Corollary 1 . . . . .	10
<b>C Proofs for Section 5</b>	<b>14</b>
C.1 Proof of Proposition 1 . . . . .	14
C.2 Proof of Theorem 4 . . . . .	16
<b>D Additional experiments: noisy label and non-isotropic covariance</b>	<b>16</b>
D.1 Noisy labels . . . . .	16
D.2 Non-isotropic covariance . . . . .	16
<b>E Additional experiments: multi-layer linear attention models</b>	<b>17</b>
E.1 Multi-layer linear attention model . . . . .	17
E.2 Experiments . . . . .	17

---

## A Lemmas

**Lemma 1** Suppose  $\mathbf{X}$  is a  $n \times d$  matrix, each column of which is independently drawn from a  $d$ -variate Gaussian distribution with zero mean:

$$\mathbf{X} = [\mathbf{x}_1 \ \dots \ \mathbf{x}_n]^\top, \text{ where } \mathbf{x}_i \sim \mathcal{N}(0, \Sigma_{\mathbf{x}}) \in \mathbb{R}^d, i \in [n].$$

For a constant matrix  $\mathbf{A} \in \mathbb{R}^{d \times d}$ , the following expectation can be determined by:

$$\mathbb{E}[\mathbf{X}^\top \mathbf{X} \mathbf{A} \mathbf{X}^\top \mathbf{X}] = n \text{tr}(\Sigma_{\mathbf{x}} \mathbf{A}) \Sigma_{\mathbf{x}} + n(n+1) \Sigma_{\mathbf{x}} \mathbf{A} \Sigma_{\mathbf{x}}.$$

**Proof.** We begin by expressing  $\mathbf{X}^\top \mathbf{X}$  as a sum over its columns:

$$\mathbf{X}^\top \mathbf{X} = \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top.$$

Therefore,

$$\mathbf{X}^\top \mathbf{X} \mathbf{A} \mathbf{X}^\top \mathbf{X} = \left( \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top \right) \mathbf{A} \left( \sum_{j=1}^n \mathbf{x}_j \mathbf{x}_j^\top \right) = \sum_{i=1}^n \sum_{j=1}^n \mathbf{x}_i \mathbf{x}_i^\top \mathbf{A} \mathbf{x}_j \mathbf{x}_j^\top.$$

Taking expectations on both sides, we have:

$$\mathbb{E}[\mathbf{X}^\top \mathbf{X} \mathbf{A} \mathbf{X}^\top \mathbf{X}] = \sum_{i=1}^n \sum_{j=1}^n \mathbb{E}[\mathbf{x}_i \mathbf{x}_i^\top \mathbf{A} \mathbf{x}_j \mathbf{x}_j^\top].$$

Since the vectors  $\mathbf{x}_i$  are independent and identically distributed, we can split the sum into terms where  $i = j$  and  $i \neq j$ :

$$\mathbb{E}[\mathbf{X}^\top \mathbf{X} \mathbf{A} \mathbf{X}^\top \mathbf{X}] = \sum_{i=1}^n \mathbb{E}[\mathbf{x}_i \mathbf{x}_i^\top \mathbf{A} \mathbf{x}_i \mathbf{x}_i^\top] + \sum_{i \neq j} \mathbb{E}[\mathbf{x}_i \mathbf{x}_i^\top \mathbf{A} \mathbf{x}_j \mathbf{x}_j^\top].$$

For  $i \neq j$ , independence implies:

$$\mathbb{E}[\mathbf{x}_i \mathbf{x}_i^\top \mathbf{A} \mathbf{x}_j \mathbf{x}_j^\top] = \mathbb{E}[\mathbf{x}_i \mathbf{x}_i^\top] \mathbf{A} \mathbb{E}[\mathbf{x}_j \mathbf{x}_j^\top] = \Sigma_{\mathbf{x}} \mathbf{A} \Sigma_{\mathbf{x}}.$$

There are  $n(n-1)$  such terms. For  $i = j$ , we compute:

$$\mathbb{E}[\mathbf{x}_i \mathbf{x}_i^\top \mathbf{A} \mathbf{x}_i \mathbf{x}_i^\top] = \mathbb{E}[(\mathbf{x}_i^\top \mathbf{A} \mathbf{x}_i) \mathbf{x}_i \mathbf{x}_i^\top].$$

Using Isserlis' theorem for zero-mean Gaussian vectors, we have:

$$\mathbb{E}[(\mathbf{x}^\top \mathbf{A} \mathbf{x}) \mathbf{x} \mathbf{x}^\top] = \text{tr}(\mathbf{A} \Sigma_{\mathbf{x}}) \Sigma_{\mathbf{x}} + 2 \Sigma_{\mathbf{x}} \mathbf{A} \Sigma_{\mathbf{x}}.$$

Thus, summing over  $n$  terms:

$$\sum_{i=1}^n \mathbb{E}[\mathbf{x}_i \mathbf{x}_i^\top \mathbf{A} \mathbf{x}_i \mathbf{x}_i^\top] = n (\text{tr}(\mathbf{A} \Sigma_{\mathbf{x}}) \Sigma_{\mathbf{x}} + 2 \Sigma_{\mathbf{x}} \mathbf{A} \Sigma_{\mathbf{x}}).$$

Adding all terms together:

$$\begin{aligned} \mathbb{E}[\mathbf{X}^\top \mathbf{X} \mathbf{A} \mathbf{X}^\top \mathbf{X}] &= n (\text{tr}(\mathbf{A} \Sigma_{\mathbf{x}}) \Sigma_{\mathbf{x}} + 2 \Sigma_{\mathbf{x}} \mathbf{A} \Sigma_{\mathbf{x}}) + n(n-1) \Sigma_{\mathbf{x}} \mathbf{A} \Sigma_{\mathbf{x}} \\ &= n \text{tr}(\mathbf{A} \Sigma_{\mathbf{x}}) \Sigma_{\mathbf{x}} + (2n + n(n-1)) \Sigma_{\mathbf{x}} \mathbf{A} \Sigma_{\mathbf{x}} \\ &= n \text{tr}(\mathbf{A} \Sigma_{\mathbf{x}}) \Sigma_{\mathbf{x}} + n(n+1) \Sigma_{\mathbf{x}} \mathbf{A} \Sigma_{\mathbf{x}}. \end{aligned}$$

This completes the proof. ■

**Lemma 2** Suppose  $\mathbf{X}$  is a  $n \times d$  matrix, each column of which is independently drawn from a  $d$ -variate Gaussian distribution with zero mean:

$$\mathbf{X} = [\mathbf{x}_1 \ \dots \ \mathbf{x}_n]^\top, \text{ where } \mathbf{x}_i \sim \mathcal{N}(0, \Sigma_{\mathbf{x}}) \in \mathbb{R}^d, i \in [n].$$

For a zero-mean Gaussian variable sampled independently  $\boldsymbol{\xi} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_n) \in \mathbb{R}^n$ , the following expectation moment can be determined by:

$$\mathbb{E} [\mathbf{X}^\top \boldsymbol{\xi} \boldsymbol{\xi}^\top \mathbf{X}] = n\sigma^2 \Sigma_{\mathbf{x}}.$$

**Proof.** Using the independence of  $\mathbf{X}$  and the entries of  $\boldsymbol{\xi} = [\xi_1 \ \dots \ \xi_n]^\top$ ,  $\mathbb{E}[\xi_i \xi_j] = \sigma^2 \delta_{ij}$  (where  $\delta_{ij}$  is the Kronecker delta, equal to 1 if  $i = j$  and 0 otherwise):

$$\begin{aligned} \mathbb{E} [\mathbf{X}^\top \boldsymbol{\xi} \boldsymbol{\xi}^\top \mathbf{X}] &= \mathbb{E} \left[ \sum_{i=1}^n \sum_{j=1}^n \xi_i \xi_j \mathbf{x}_i \mathbf{x}_j^\top \right] \\ &= \sum_{i=1}^n \mathbb{E}[\xi_i \xi_i] \mathbb{E}[\mathbf{x}_i \mathbf{x}_i^\top] \\ &= n\sigma^2 \Sigma_{\mathbf{x}}. \end{aligned}$$

■

**Lemma 3** Let  $\mathbf{W} \in \mathbb{R}^{d \times d}$ , and  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{d \times d}$  be constant matrices. Then,

$$\frac{\partial}{\partial \mathbf{W}} \text{tr}(\mathbf{W} \mathbf{A} \mathbf{W}^\top \mathbf{B}) = \mathbf{B}^\top \mathbf{W} \mathbf{A}^\top + \mathbf{B} \mathbf{W} \mathbf{A}.$$

**Proof.** We will compute the derivative  $\frac{\partial}{\partial \mathbf{W}} \text{tr}(\mathbf{W} \mathbf{A} \mathbf{W}^\top \mathbf{B})$  using an element-wise approach.

First, expand the trace function:

$$\text{tr}(\mathbf{W} \mathbf{A} \mathbf{W}^\top \mathbf{B}) = \sum_{i=1}^d (\mathbf{W} \mathbf{A} \mathbf{W}^\top \mathbf{B})_{ii}.$$

Using the definition of matrix multiplication, we have:

$$(\mathbf{W} \mathbf{A})_{ij} = \sum_{k=1}^d W_{ik} A_{kj},$$

$$(\mathbf{W}^\top \mathbf{B})_{ji} = \sum_{l=1}^d W_{lj} B_{li}.$$

Therefore,

$$(\mathbf{W} \mathbf{A} \mathbf{W}^\top \mathbf{B})_{ii} = \sum_{j=1}^d (\mathbf{W} \mathbf{A})_{ij} (\mathbf{W}^\top \mathbf{B})_{ji} = \sum_{j=1}^d \left( \sum_{k=1}^d W_{ik} A_{kj} \right) \left( \sum_{l=1}^d W_{lj} B_{li} \right).$$

Thus, the trace becomes:

$$\text{tr}(\mathbf{W} \mathbf{A} \mathbf{W}^\top \mathbf{B}) = \sum_{i=1}^d \sum_{j=1}^d \sum_{k=1}^d \sum_{l=1}^d W_{ik} A_{kj} W_{lj} B_{li}.$$

We need to compute the derivative with respect to  $W_{pq}$ :

$$\frac{\partial}{\partial W_{pq}} \text{tr}(\mathbf{W} \mathbf{A} \mathbf{W}^\top \mathbf{B}) = \sum_{i=1}^d \sum_{j=1}^d \sum_{k=1}^d \sum_{l=1}^d \frac{\partial}{\partial W_{pq}} (W_{ik} A_{kj} W_{lj} B_{li}).$$

Note that  $A_{kj}$  and  $B_{li}$  are constants.

We have:

$$\begin{aligned}\frac{\partial}{\partial W_{pq}} W_{ik} &= \delta_{ip}\delta_{kq}, \\ \frac{\partial}{\partial W_{pq}} W_{lj} &= \delta_{lp}\delta_{jq},\end{aligned}$$

where  $\delta_{ij}$  is the Kronecker delta, equal to 1 if  $i = j$  and 0 otherwise.

Therefore,

$$\frac{\partial}{\partial W_{pq}} (W_{ik} A_{kj} W_{lj} B_{li}) = (\delta_{ip}\delta_{kq} W_{lj} + W_{ik}\delta_{lp}\delta_{jq}) A_{kj} B_{li}.$$

Then:

$$\begin{aligned}\frac{\partial}{\partial W_{pq}} \text{tr}(\mathbf{W} \mathbf{A} \mathbf{W}^\top \mathbf{B}) &= \sum_{i=1}^d \sum_{j=1}^d \sum_{k=1}^d \sum_{l=1}^d \frac{\partial}{\partial W_{pq}} (W_{ik} A_{kj} W_{lj} B_{li}) \\ &= \sum_{i=1}^d \sum_{j=1}^d \sum_{k=1}^d \sum_{l=1}^d (\delta_{ip}\delta_{kq} W_{lj} + W_{ik}\delta_{lp}\delta_{jq}) A_{kj} B_{li} \\ &= \sum_{j=1}^d \sum_{l=1}^d W_{lj} A_{qj} B_{lp} + \sum_{i=1}^d \sum_{k=1}^d W_{ik} A_{kj} B_{pi} \\ &= (\mathbf{A} \mathbf{W}^\top \mathbf{B})_{qp} + (\mathbf{B} \mathbf{W} \mathbf{A})_{pq} \\ &= (\mathbf{B}^\top \mathbf{W} \mathbf{A}^\top + \mathbf{B} \mathbf{W} \mathbf{A})_{pq}\end{aligned}$$

Since this holds for all elements  $(p, q)$ , in matrix form, we have:

$$\frac{\partial}{\partial \mathbf{W}} \text{tr}(\mathbf{W} \mathbf{A} \mathbf{W}^\top \mathbf{B}) = \mathbf{B}^\top \mathbf{W} \mathbf{A}^\top + \mathbf{B} \mathbf{W} \mathbf{A}.$$

■

**Lemma 4 (Reduced form)** Denote the output sequence of the attention layer as  $\text{Attn}(\mathbf{Z})$ . Then under Assumption 1, the output becomes

**Proof.**

$$\begin{aligned}\hat{y} &= \text{Attn}(\mathbf{Z})_{(n+1, d+1)} = \mathbf{e}_{n+1}^\top \text{Attn}(\mathbf{Z}) \mathbf{e}_{d+1} \\ &= \mathbf{e}_{n+1}^\top (\mathbf{Z} \mathbf{W}_q \mathbf{W}_k^\top (\mathbf{Z}^\top)^\top) \mathbf{M} \mathbf{Z} \mathbf{W}_v \mathbf{e}_{d+1} \\ &= (\mathbf{e}_{n+1}^\top \mathbf{Z}) \underbrace{(\mathbf{W}_q \mathbf{W}_k^\top)}_{=\mathbf{A}} (\mathbf{Z}^\top \mathbf{M} \mathbf{Z}) \underbrace{\mathbf{W}_v \mathbf{e}_{d+1}}_{=\mathbf{a}} \\ &= (\mathbf{e}_{n+1}^\top \mathbf{Z}) \mathbf{A} (\mathbf{Z}^\top \mathbf{M} \mathbf{Z}) \mathbf{a} \\ &= \begin{bmatrix} \mathbf{x} \\ 0 \end{bmatrix}^\top \mathbf{A} \begin{bmatrix} \mathbf{X}^\top \mathbf{X} & \mathbf{X}^\top \mathbf{y} \\ \mathbf{y}^\top \mathbf{X} & \mathbf{y}^\top \mathbf{y} \end{bmatrix} \mathbf{a}\end{aligned}$$

Under Assumption 1

$$\mathbf{A} = \begin{bmatrix} \mathbf{W}_{d \times d} & \mathbf{0}_{d \times 1} \\ *_{1 \times d} & *_{1 \times 1} \end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix} \mathbf{0}_{d \times 1} \\ 1_{1 \times 1} \end{bmatrix},$$

The output finally reduced to

$$\begin{aligned}\hat{y} &= \begin{bmatrix} \mathbf{x} \\ 0 \end{bmatrix}^\top \begin{bmatrix} \mathbf{W} & \mathbf{0} \\ * & * \end{bmatrix} \begin{bmatrix} \mathbf{X}^\top \mathbf{X} & \mathbf{X}^\top \mathbf{y} \\ \mathbf{y}^\top \mathbf{X} & \mathbf{y}^\top \mathbf{y} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \\ &= \mathbf{x}^\top \mathbf{W} \mathbf{X}^\top \mathbf{y}.\end{aligned}$$

■

## B Proofs for Section 4

We consider an in-context learning (ICL) problem with demonstrations  $(\mathbf{x}_i, y_i)_{i=1}^{n+1}$ , and the input sequence  $\mathbf{Z}$  is defined by removing  $y_{n+1}$  as follows:

$$\mathbf{Z} = [z_1 \dots z_n \mathbf{z}]^\top = \begin{bmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_n & \mathbf{x} \\ y_1 & \dots & y_n & 0 \end{bmatrix}^\top = \begin{bmatrix} \mathbf{X}^\top & \mathbf{x} \\ \mathbf{y}^\top & 0 \end{bmatrix}^\top \in \mathbb{R}^{(n+1) \times (d+1)}. \quad (\text{A1})$$

Here,  $\mathbf{z} = [\mathbf{x}^\top \ 0]^\top$  is the query token where  $\mathbf{x} := \mathbf{x}_{n+1}$ , and  $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_n]^\top \in \mathbb{R}^{n \times d}$ ,  $\mathbf{y} = [y_1 \dots y_n]^\top \in \mathbb{R}^n$ . Then, we aim for a sequence model to predict the associated label  $y := y_{n+1}$  of the given input sequence  $\mathbf{Z}$ . In this work, we consider the following data generation of  $(\mathbf{Z}, y)$ . We will refer to  $(\mathbf{X}, \mathbf{y})$ ,  $\mathbf{x}$ , and  $y$  as contexts, query feature and the label to predict, respectively.

**Definition 1 (Single-task ICL)** *Given a task mean  $\boldsymbol{\mu} \in \mathbb{R}^d$ , and covariances  $\boldsymbol{\Sigma}_{\boldsymbol{x}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}} \succ 0 \in \mathbb{R}^{d \times d}$ . The input sequence and its associated label, i.e.,  $(\mathbf{Z}, y)$  with  $\mathbf{Z}$  denoted in (A1), are generated as follows:*

- A task parameter  $\boldsymbol{\beta}$  is generated from a Gaussian prior  $\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$ .
- Conditioned on  $\boldsymbol{\beta}$ , for  $i \in [n+1]$ ,  $(\mathbf{x}_i, y_i)$  is generated by  $\mathbf{x}_i \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{x}})$  and  $y_i \sim \mathcal{N}(\mathbf{x}_i^\top \boldsymbol{\beta}, \sigma^2)$ .

Here,  $\sigma \geq 0$  is the noise level.

In a noisy label setting, the labels  $y_i$  in the input sequence  $\mathbf{Z}$  can be obtained by

$$y_i = \mathbf{x}_i^\top \boldsymbol{\beta} + \xi_i, \text{ where } \xi_i \sim \mathcal{N}(0, \sigma^2), i \in [n+1].$$

Thus, the labels in the contexts and the label to predict can be obtained by:

$$\begin{aligned} \mathbf{y} &= \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\xi}, \text{ where } \boldsymbol{\xi} = [\xi_1 \dots \xi_n]^\top \in \mathbb{R}^n, \\ y &= \mathbf{x}^\top \boldsymbol{\beta} + \xi_{n+1}. \end{aligned}$$

**Definition 2 (Multi-task ICL)** *Consider a multi-task ICL problem with  $K$  different tasks. Each task generates  $(\mathbf{Z}, y) \sim \mathcal{D}_k$  following Definition 1 using shared feature distribution  $\mathbf{x}_i \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{x}})$ ,  $i \in [n+1]$  but distinct task distributions  $\boldsymbol{\beta}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_{\boldsymbol{\beta}_k})$  with mean  $\boldsymbol{\mu}_k$  and covariance  $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_k}$  for  $k \in [K]$ .*

Additionally, let  $\{\pi_k\}_{k=1}^K$  be the probabilities of each task, satisfying  $\sum_{k=1}^K \pi_k = 1$  and  $\pi_k \geq 0$ .

We consider a task-aware multi-task ICL setting. Specifically, when a task is selected according to  $\pi_k$ , its task index  $k$  is known. Let  $\bar{\mathcal{D}} := \sum_{k=1}^K \pi_k \mathcal{D}_k$  be the mixture of distributions and given sequence model  $f : \mathbb{R}^{(n+1) \times (d+1)} \rightarrow \mathbb{R}$ , we define the multi-task ICL objective as follows:

$$\mathcal{L}(f) = \mathbb{E}_{(\mathbf{Z}, y) \sim \bar{\mathcal{D}}}[(y - f(\mathbf{Z}))^2]. \quad (\text{A2})$$

To start with, recap from Definition 2 where task  $k$  has probability  $\pi_k$  and its task vector follows distribution  $\boldsymbol{\beta}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_{\boldsymbol{\beta}_k})$ . Following (13) in the main paper, we define the debiased and biased mixed-task covariances (variant with  $\boldsymbol{\Sigma}_{\boldsymbol{x}}$  prior) as follows:

$$\text{Debiased: } \bar{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} = \boldsymbol{\Sigma}_{\boldsymbol{x}} \sum_{k=1}^K \pi_k \mathbb{E}[(\boldsymbol{\beta}_k - \boldsymbol{\mu}_k)(\boldsymbol{\beta}_k - \boldsymbol{\mu}_k)^\top]; \quad (\text{A3a})$$

$$\text{Biased: } \tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} = \boldsymbol{\Sigma}_{\boldsymbol{x}} \sum_{k=1}^K \pi_k \mathbb{E}[\boldsymbol{\beta}_k \boldsymbol{\beta}_k^\top]. \quad (\text{A3b})$$

Note that they satisfy  $\bar{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} = \boldsymbol{\Sigma}_{\boldsymbol{x}} \sum_{k=1}^K \pi_k \boldsymbol{\Sigma}_{\boldsymbol{\beta}_k}$  and  $\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} = \boldsymbol{\Sigma}_{\boldsymbol{x}} \sum_{k=1}^K \pi_k (\boldsymbol{\Sigma}_{\boldsymbol{\beta}_k} + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^\top)$ .

We first analyze the plain training setting where no additional task-specific parameters are introduced, and all  $K$  tasks are mixed together.

Under Assumption 1 in the main paper, let the prompt token for task  $k$  be  $\mathbf{p}_k = \begin{bmatrix} \bar{\mathbf{p}}_k \\ 1 \end{bmatrix}$ . The prediction of a single-layer linear attention model can then be written as:

$$\begin{aligned} f(\mathbf{Z}^{(k)}) &= \mathbf{x}^\top \mathbf{W} [\bar{\mathbf{p}}_k \quad \mathbf{X}^\top] \begin{bmatrix} 1 \\ \mathbf{y} \end{bmatrix} \\ &= \mathbf{x}^\top \mathbf{W} (\mathbf{X}^\top \mathbf{y} + \bar{\mathbf{p}}_k) := g(\mathbf{x}, \mathbf{X}, \mathbf{y}; \mathbf{W}, \bar{\mathbf{p}}_k). \end{aligned} \quad (\text{A4})$$

Note that for the multi-task ICL with task-specific prompting, the optimization object (A2) can be denoted as:

$$\begin{aligned} \mathcal{L}(f) &= \sum_{k=1}^K \pi_k \underbrace{\mathbb{E}_{(\mathbf{Z}, \mathbf{y}) \sim \mathcal{D}_k} [(y - f(\mathbf{Z}^{(k)}))^2]}_{\text{Denoted as } \mathcal{L}_k(f)} \\ &= \sum_{k=1}^K \pi_k \mathcal{L}_k(f) = \sum_{k=1}^K \pi_k \mathbb{E}_{(\mathbf{Z}, \mathbf{y}) \sim \mathcal{D}_k} [(y - g(\mathbf{x}, \mathbf{X}, \mathbf{y}; \mathbf{W}, \bar{\mathbf{p}}_k))^2] := \sum_{k=1}^K \pi_k \mathcal{L}_k(\mathbf{W}, \bar{\mathbf{p}}_k), \end{aligned}$$

$$\text{where } \mathbf{Z} = \begin{bmatrix} \mathbf{X}^\top & \mathbf{x} \\ \mathbf{y}^\top & 0 \end{bmatrix}^\top \in \mathbb{R}^{(n+1) \times (d+1)}, \mathbf{Z}^{(k)} = \begin{bmatrix} \bar{\mathbf{p}}_k & \mathbf{X}^\top & \mathbf{x} \\ 1 & \mathbf{y}^\top & 0 \end{bmatrix}^\top \in \mathbb{R}^{(n+2) \times (d+1)}.$$

Note that the multi-task ICL loss is equivalent to calculating the weighted sum of the task- $k$  ICL losses over all tasks  $k \in [K]$ .

We begin by deriving the single-task ICL loss  $\mathcal{L}_k(f)$  for task  $k$ , and then generalize it to the multi-task ICL loss by taking a weighted sum. Since the derivation is similar for all tasks, the task index  $k$  is omitted in the following derivation for simplicity. Unless otherwise specified,  $\mathcal{L}_k(\mathbf{W}, \bar{\mathbf{p}}_k)$  and  $\mathcal{L}(\mathbf{W}, \bar{\mathbf{p}})$  will represent the same meaning. This convention similarly applies to other task-specific parameters, e.g.,  $\Sigma_\beta \leftrightarrow \Sigma_{\beta_k}$ ,  $\mu \leftrightarrow \mu_k$ , etc.

The loss on a certain task with a trainable prompt can be determined by:

$$\begin{aligned} \mathcal{L}(\mathbf{W}, \bar{\mathbf{p}}) &= \mathbb{E} [(y - g(\mathbf{x}, \mathbf{X}, \mathbf{y}; \mathbf{W}, \bar{\mathbf{p}}))^2] \\ &= \mathbb{E} [(\mathbf{x}^\top \mathbf{W} \mathbf{X}^\top \mathbf{y} + \mathbf{x}^\top \mathbf{W} \bar{\mathbf{p}} - \mathbf{x}^\top \beta - \xi_{n+1})^2] \\ &= \mathbb{E} [(\mathbf{x}^\top \mathbf{W} \mathbf{X}^\top (\mathbf{X} \beta + \xi) + \mathbf{x}^\top \mathbf{W} \bar{\mathbf{p}} - \mathbf{x}^\top \beta - \xi_{n+1})^2] \\ &= \mathbb{E} \left[ \left( \mathbf{x}^\top \underbrace{((\mathbf{W} \mathbf{X}^\top \mathbf{X} - \mathbf{I})(\bar{\beta} + \mu) + \mathbf{W} \mathbf{X}^\top \xi + \mathbf{W} \bar{\mathbf{p}})}_{\text{Denoted as a (task-specific) vector } \mathbf{c}} \right)^2 \right] + \sigma^2 \\ &= \mathbb{E} [(\mathbf{x}^\top \mathbf{c})^2] + \sigma^2 = \text{tr}(\mathbb{E}[\mathbf{c} \mathbf{c}^\top] \Sigma_{\mathbf{x}}) + \sigma^2, \end{aligned} \quad (\text{A5})$$

where  $\bar{\beta} = \beta - \mu \sim \mathcal{N}(\mathbf{0}, \Sigma_\beta)$  is a centralized variable.

## B.1 Proof of Theorem 1

**Theorem 1 (Plain training)** Consider training a single-layer linear attention model in solving multi-task ICL problem with dataset defined in Definition 2 and model construction as described in Assumption 1. Let the optimal solution  $\mathbf{W}_{\text{PT}}^*$  (c.f. (10) in the main paper) and the minimal plain training loss  $\mathcal{L}_{\text{PT}}^*$  as defined in Section 3.3. Additionally, let  $\tilde{\Sigma}_\beta$  be defined in (14) in the main paper and  $\bar{\mathbf{W}}_{\text{PT}}^* = \Sigma_{\mathbf{x}} \mathbf{W}_{\text{PT}}^*$ .

Then the solution  $\bar{\mathbf{W}}_{\text{PT}}^*$  and optimal loss  $\mathcal{L}_{\text{PT}}^*$  satisfy

$$\begin{aligned} \bar{\mathbf{W}}_{\text{PT}}^* &= \tilde{\Sigma}_\beta \left( (n+1) \tilde{\Sigma}_\beta + (\text{tr}(\tilde{\Sigma}_\beta) + \sigma^2) \mathbf{I} \right)^{-1}, \\ \mathcal{L}_{\text{PT}}^* &= \text{tr}(\tilde{\Sigma}_\beta) + \sigma^2 - n \text{tr}(\bar{\mathbf{W}}_{\text{PT}}^* \tilde{\Sigma}_\beta). \end{aligned}$$

**Proof.** As previously stated, the following derivation applies to all tasks  $k \in [K]$ . Therefore, for simplicity, we omit the index  $k$  in the notation unless otherwise specified.

In the plain training setting,  $\bar{\mathbf{p}} = \mathbf{0}$  for all tasks, and only the attention model, parameterized by  $\mathbf{W}$  under Assumption 1, is updated. Hence, the loss in (A5) is:

$$\mathcal{L}(\mathbf{W}, \bar{\mathbf{p}} = \mathbf{0}) = \text{tr}(\mathbb{E}[cc^\top] \Sigma_x) + \sigma^2, \text{ where } \mathbf{c} = ((\mathbf{W} \mathbf{X}^\top \mathbf{X} - \mathbf{I})(\bar{\boldsymbol{\beta}} + \boldsymbol{\mu}) + \mathbf{W} \mathbf{X}^\top \boldsymbol{\xi}).$$

The expansion of  $cc^\top$  is (there are  $3 \times 3 = 9$  terms in total):

$$\begin{aligned} cc^\top &= ((\mathbf{W} \mathbf{X}^\top \mathbf{X} - \mathbf{I})(\bar{\boldsymbol{\beta}} + \boldsymbol{\mu}) + \mathbf{W} \mathbf{X}^\top \boldsymbol{\xi}) ((\mathbf{W} \mathbf{X}^\top \mathbf{X} - \mathbf{I})(\bar{\boldsymbol{\beta}} + \boldsymbol{\mu}) + \mathbf{W} \mathbf{X}^\top \boldsymbol{\xi})^\top \\ &= ((\mathbf{W} \mathbf{X}^\top \mathbf{X})(\bar{\boldsymbol{\beta}} + \boldsymbol{\mu}) - (\bar{\boldsymbol{\beta}} + \boldsymbol{\mu}) + \mathbf{W} \mathbf{X}^\top \boldsymbol{\xi}) ((\mathbf{W} \mathbf{X}^\top \mathbf{X})(\bar{\boldsymbol{\beta}} + \boldsymbol{\mu}) - (\bar{\boldsymbol{\beta}} + \boldsymbol{\mu}) + \mathbf{W} \mathbf{X}^\top \boldsymbol{\xi})^\top \\ &= [(\mathbf{W} \mathbf{X}^\top \mathbf{X})(\bar{\boldsymbol{\beta}} + \boldsymbol{\mu})] [(\mathbf{W} \mathbf{X}^\top \mathbf{X})(\bar{\boldsymbol{\beta}} + \boldsymbol{\mu})]^\top + [(\mathbf{W} \mathbf{X}^\top \mathbf{X})(\bar{\boldsymbol{\beta}} + \boldsymbol{\mu})] [-(\bar{\boldsymbol{\beta}} + \boldsymbol{\mu})]^\top + [(\mathbf{W} \mathbf{X}^\top \mathbf{X})(\bar{\boldsymbol{\beta}} + \boldsymbol{\mu})] [\mathbf{W} \mathbf{X}^\top \boldsymbol{\xi}]^\top \\ &\quad + [-(\bar{\boldsymbol{\beta}} + \boldsymbol{\mu})] [(\mathbf{W} \mathbf{X}^\top \mathbf{X})(\bar{\boldsymbol{\beta}} + \boldsymbol{\mu})]^\top + [-(\bar{\boldsymbol{\beta}} + \boldsymbol{\mu})] [-(\bar{\boldsymbol{\beta}} + \boldsymbol{\mu})]^\top + [-(\bar{\boldsymbol{\beta}} + \boldsymbol{\mu})] [\mathbf{W} \mathbf{X}^\top \boldsymbol{\xi}]^\top \\ &\quad + [\mathbf{W} \mathbf{X}^\top \boldsymbol{\xi}] [(\mathbf{W} \mathbf{X}^\top \mathbf{X})(\bar{\boldsymbol{\beta}} + \boldsymbol{\mu})]^\top + [\mathbf{W} \mathbf{X}^\top \boldsymbol{\xi}] [-(\bar{\boldsymbol{\beta}} + \boldsymbol{\mu})]^\top + [\mathbf{W} \mathbf{X}^\top \boldsymbol{\xi}] [\mathbf{W} \mathbf{X}^\top \boldsymbol{\xi}]^\top. \end{aligned}$$

Take expectation of it,

$$\begin{aligned} \mathbb{E}[cc^\top] &= \left[ \mathbf{W} \underbrace{\mathbb{E}[\mathbf{X}^\top \mathbf{X}(\bar{\boldsymbol{\beta}} + \boldsymbol{\mu})(\bar{\boldsymbol{\beta}} + \boldsymbol{\mu})^\top \mathbf{X}^\top \mathbf{X}]}_{\text{Lemma 1, denoted as a (task-specific) matrix } \mathbf{C}} \mathbf{W}^\top \right] + [-n\mathbf{W} \Sigma_x (\boldsymbol{\Sigma}_\beta + \boldsymbol{\mu} \boldsymbol{\mu}^\top)] + 0 \\ &\quad + [-n(\boldsymbol{\Sigma}_\beta + \boldsymbol{\mu} \boldsymbol{\mu}^\top) \Sigma_x \mathbf{W}^\top] + (\boldsymbol{\Sigma}_\beta + \boldsymbol{\mu} \boldsymbol{\mu}^\top) + 0 \\ &\quad + 0 + 0 + \underbrace{n\sigma^2 \mathbf{W} \Sigma_x \mathbf{W}^\top}_{\text{Lemma 2}} \\ &= \mathbf{W}(\mathbf{C} + n\sigma^2 \Sigma_x) \mathbf{W}^\top - n\mathbf{W} \Sigma_x (\boldsymbol{\Sigma}_\beta + \boldsymbol{\mu} \boldsymbol{\mu}^\top) - n(\boldsymbol{\Sigma}_\beta + \boldsymbol{\mu} \boldsymbol{\mu}^\top) \Sigma_x \mathbf{W}^\top + (\boldsymbol{\Sigma}_\beta + \boldsymbol{\mu} \boldsymbol{\mu}^\top), \end{aligned}$$

where  $\mathbf{C} = n\text{tr}(\Sigma_x(\boldsymbol{\Sigma}_\beta + \boldsymbol{\mu} \boldsymbol{\mu}^\top)) \Sigma_x + n(n+1) \Sigma_x (\boldsymbol{\Sigma}_\beta + \boldsymbol{\mu} \boldsymbol{\mu}^\top) \Sigma_x$ .

Substitute back into the loss  $\mathcal{L}(\mathbf{W}, \bar{\mathbf{p}} = \mathbf{0})$ :

$$\mathcal{L}(\mathbf{W}, \bar{\mathbf{p}} = \mathbf{0}) = \text{tr}(\mathbf{W}(\mathbf{C} + n\sigma^2 \Sigma_x) \mathbf{W}^\top \Sigma_x) - 2n\text{tr}(\Sigma_x \mathbf{W} \Sigma_x (\boldsymbol{\Sigma}_\beta + \boldsymbol{\mu} \boldsymbol{\mu}^\top)) + \text{tr}(\Sigma_x (\boldsymbol{\Sigma}_\beta + \boldsymbol{\mu} \boldsymbol{\mu}^\top)) + \sigma^2.$$

Use the following definition:

$$\begin{aligned} \text{Debiased: } \bar{\boldsymbol{\Sigma}}_\beta &= \Sigma_x \sum_{k=1}^K \pi_k \mathbb{E}[(\boldsymbol{\beta}_k - \boldsymbol{\mu}_k)(\boldsymbol{\beta}_k - \boldsymbol{\mu}_k)^\top] = \Sigma_x \sum_{k=1}^K \pi_k \boldsymbol{\Sigma}_{\beta_k}; \\ \text{Biased: } \tilde{\boldsymbol{\Sigma}}_\beta &= \Sigma_x \sum_{k=1}^K \pi_k \mathbb{E}[\boldsymbol{\beta}_k \boldsymbol{\beta}_k^\top] = \Sigma_x \sum_{k=1}^K \pi_k (\boldsymbol{\Sigma}_{\beta_k} + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^\top). \end{aligned}$$

Denote

$$\bar{\mathbf{C}} = \sum_{k=1}^K \pi_k \mathbf{C}_k = n\text{tr}(\tilde{\boldsymbol{\Sigma}}_\beta) \Sigma_x + n(n+1) \tilde{\boldsymbol{\Sigma}}_\beta \Sigma_x.$$

The weighted sum of the  $k$ -th task plain training loss over all tasks  $k \in [K]$  is:

$$\begin{aligned} \mathcal{L}_{\text{PT}}(\mathbf{W}, \mathbf{P} = \mathbf{0}) &= \sum_{k=1}^K \pi_k \mathcal{L}_k(\mathbf{W}, \bar{\mathbf{p}}_k = \mathbf{0}) \\ &= \text{tr}(\mathbf{W}(\bar{\mathbf{C}} + n\sigma^2 \Sigma_x) \mathbf{W}^\top \Sigma_x) - 2n\text{tr}(\mathbf{W} \tilde{\boldsymbol{\Sigma}}_\beta \Sigma_x) + \text{tr}(\tilde{\boldsymbol{\Sigma}}_\beta) + \sigma^2. \end{aligned}$$

Take derivative w.r.t.  $\mathbf{W}$ , using Lemma 3. Since  $\mathbf{C}$  is symmetric

$$\frac{\partial \mathcal{L}_{\text{PT}}(\mathbf{W}, \mathbf{P} = \mathbf{0})}{\partial \mathbf{W}} = 2\Sigma_x \mathbf{W}(\bar{\mathbf{C}} + n\sigma^2 \Sigma_x) - 2n\tilde{\boldsymbol{\Sigma}}_\beta \Sigma_x$$

Let  $\bar{\mathbf{W}}_{\text{PT}}^* = \Sigma_x \mathbf{W}_{\text{PT}}^*$ , and set the derivative to zero:

$$\begin{aligned}\bar{\mathbf{W}}_{\text{PT}}^* &= n \tilde{\Sigma}_{\beta} \Sigma_x (\bar{C} + n\sigma^2 \Sigma_x)^{-1} \\ &= \tilde{\Sigma}_{\beta} \Sigma_x \left( (\text{tr}(\tilde{\Sigma}_{\beta}) + \sigma^2) \Sigma_x + (n+1) \tilde{\Sigma}_{\beta} \Sigma_x \right)^{-1} \\ &= \tilde{\Sigma}_{\beta} \left( (n+1) \tilde{\Sigma}_{\beta} + (\text{tr}(\tilde{\Sigma}_{\beta}) + \sigma^2) \mathbf{I} \right)^{-1}.\end{aligned}$$

Substitute back into  $\mathcal{L}_{\text{PT}}(\mathbf{W}, \mathbf{P} = \mathbf{0})$ :

$$\mathcal{L}_{\text{PT}}^* = \text{tr}(\tilde{\Sigma}_{\beta}) + \sigma^2 - n \text{tr}(\bar{\mathbf{W}}_{\text{PT}}^* \tilde{\Sigma}_{\beta}).$$

■

## B.2 Proof of Theorem 2

**Theorem 2 (Fine-tuning)** Suppose a pretrained model as described in Theorem 1 is given with  $\mathbf{W}_{\text{PT}}^*$  being its optimal solution. Consider fine-tuning this model with task-specific prompts as defined in Definition 3, and let the optimal prompt matrix  $\mathbf{P}_{\text{FT}}^*$  (c.f. (11) in the main paper) and the minimal fine-tuning loss  $\mathcal{L}_{\text{FT}}^*$  be defined in Section 3.3. Additionally, let  $\tilde{\Sigma}_{\beta}, \tilde{\Sigma}_{\beta}$  be defined in (14) in the main paper and  $\bar{\mathbf{W}}_{\text{PT}}^* = \Sigma_x \mathbf{W}_{\text{PT}}^*$ , and define the mean matrix

$$\mathbf{M}_{\mu} = [\mu_1 \ \cdots \ \mu_K]^{\top} \in \mathbb{R}^{K \times d}.$$

Then the solution  $\mathbf{P}_{\text{FT}}^*$  and optimal loss  $\mathcal{L}_{\text{FT}}^*$  satisfy

$$\begin{aligned}\mathbf{P}_{\text{FT}}^* &= \mathbf{M}_{\mu} ((\bar{\mathbf{W}}_{\text{PT}}^*)^{-1} - n\mathbf{I}) \Sigma_x, \\ \mathcal{L}_{\text{FT}}^* &= \mathcal{L}_{\text{PT}}^* - \text{tr}((\tilde{\Sigma}_{\beta} - \tilde{\Sigma}_{\beta})(n\bar{\mathbf{W}}_{\text{PT}}^* - \mathbf{I})^{\top}(n\bar{\mathbf{W}}_{\text{PT}}^* - \mathbf{I})).\end{aligned}$$

**Proof.** As previously stated, the following derivation applies to all tasks  $k \in [K]$ . Therefore, for simplicity, we omit the index  $k$  in the notation unless otherwise specified.

### 1. Determining the optimal task-specific prompts

In the fine-tuning setting, the attention model is pretrained and parameterized by a fixed  $\mathbf{W} = \mathbf{W}_{\text{PT}}^*$ , and only the task-specific prompts are fine-tuned. Then recap from (A5):

$$\begin{aligned}\mathcal{L}(\mathbf{W}, \bar{\mathbf{p}}) &= \text{tr}(\mathbb{E}[cc^{\top}] \Sigma_x) + \sigma^2, \\ \text{where } \mathbf{c} &= ((\mathbf{W}\mathbf{X}^{\top}\mathbf{X} - \mathbf{I})(\bar{\beta} + \mu) + \mathbf{W}\mathbf{X}^{\top}\xi + \mathbf{W}\bar{\mathbf{p}}).\end{aligned}$$

The optimal task-specific prompt is determined by taking the derivative and setting it to zero:

$$\begin{aligned}\frac{\partial \mathcal{L}(\mathbf{W}, \bar{\mathbf{p}})}{\partial \bar{\mathbf{p}}} &= 2 \mathbb{E} \left[ \frac{\partial \mathbf{c}^{\top}}{\partial \bar{\mathbf{p}}} \Sigma_x \mathbf{c} \right] \\ &= 2\mathbf{W}^{\top} \Sigma_x \mathbb{E}[\mathbf{c}] = 2\mathbf{W}^{\top} \Sigma_x [(n\mathbf{W}\Sigma_x - \mathbf{I})\mu + \mathbf{W}\bar{\mathbf{p}}] = \mathbf{0}, \\ \Rightarrow \bar{\mathbf{p}}^* &= (\mathbf{W}^{-1} - n\Sigma_x) \mu,\end{aligned}$$

which is equivalent to (by substituting  $\mathbf{W} = \mathbf{W}_{\text{PT}}^*$ ):

$$\mathbf{P}_{\text{FT}}^* = \mathbf{M}_{\mu} ((\bar{\mathbf{W}}_{\text{PT}}^*)^{-1} - n\Sigma_x).$$

### 2. Determining the fine-tuning loss

Substituting back  $\bar{\mathbf{p}}^* = (\mathbf{W}^{-1} - n\Sigma_x) \mu$ :

$$\begin{aligned}\mathcal{L}(\mathbf{W}, \bar{\mathbf{p}}) &= \text{tr}(\mathbb{E}[cc^{\top}] \Sigma_x) + \sigma^2, \\ \text{where } \mathbf{c} &= ((\mathbf{W}\mathbf{X}^{\top}\mathbf{X} - \mathbf{I})(\bar{\beta} + \mu) + \mathbf{W}\mathbf{X}^{\top}\xi + \mathbf{W}\bar{\mathbf{p}}) \\ &= \mathbf{W}\mathbf{X}^{\top}\mathbf{X}(\bar{\beta} + \mu) - \bar{\beta} + \mathbf{W}\mathbf{X}^{\top}\xi - n\mathbf{W}\Sigma_x\mu.\end{aligned}$$

The expansion of  $cc^\top$  is (there are  $4 \times 4 = 16$  terms in total):

$$\begin{aligned}
 cc^\top &= [\mathbf{W} \mathbf{X}^\top \mathbf{X} (\bar{\beta} + \boldsymbol{\mu}) - \bar{\beta} + \mathbf{W} \mathbf{X}^\top \boldsymbol{\xi} - n \mathbf{W} \boldsymbol{\Sigma}_x \boldsymbol{\mu}] [\mathbf{W} \mathbf{X}^\top \mathbf{X} (\bar{\beta} + \boldsymbol{\mu}) - \bar{\beta} + \mathbf{W} \mathbf{X}^\top \boldsymbol{\xi} - n \mathbf{W} \boldsymbol{\Sigma}_x \boldsymbol{\mu}]^\top \\
 &= [\mathbf{W} \mathbf{X}^\top \mathbf{X} (\bar{\beta} + \boldsymbol{\mu})] [\mathbf{W} \mathbf{X}^\top \mathbf{X} (\bar{\beta} + \boldsymbol{\mu})]^\top + [\mathbf{W} \mathbf{X}^\top \mathbf{X} (\bar{\beta} + \boldsymbol{\mu})] [-\bar{\beta}]^\top + [\mathbf{W} \mathbf{X}^\top \mathbf{X} (\bar{\beta} + \boldsymbol{\mu})] [\mathbf{W} \mathbf{X}^\top \boldsymbol{\xi}]^\top + [\mathbf{W} \mathbf{X}^\top \mathbf{X} (\bar{\beta} + \boldsymbol{\mu})] [-n \mathbf{W} \boldsymbol{\Sigma}_x \boldsymbol{\mu}]^\top \\
 &\quad + [-\bar{\beta}] [\mathbf{W} \mathbf{X}^\top \mathbf{X} (\bar{\beta} + \boldsymbol{\mu})]^\top + [-\bar{\beta}] [-\bar{\beta}]^\top + [-\bar{\beta}] [\mathbf{W} \mathbf{X}^\top \boldsymbol{\xi}]^\top + [-\bar{\beta}] [-n \mathbf{W} \boldsymbol{\Sigma}_x \boldsymbol{\mu}]^\top \\
 &\quad + [\mathbf{W} \mathbf{X}^\top \boldsymbol{\xi}] [\mathbf{W} \mathbf{X}^\top \mathbf{X} (\bar{\beta} + \boldsymbol{\mu})]^\top + [\mathbf{W} \mathbf{X}^\top \boldsymbol{\xi}] [-\bar{\beta}]^\top + [\mathbf{W} \mathbf{X}^\top \boldsymbol{\xi}] [\mathbf{W} \mathbf{X}^\top \boldsymbol{\xi}]^\top + [\mathbf{W} \mathbf{X}^\top \boldsymbol{\xi}] [-n \mathbf{W} \boldsymbol{\Sigma}_x \boldsymbol{\mu}]^\top \\
 &\quad + [-n \mathbf{W} \boldsymbol{\Sigma}_x \boldsymbol{\mu}] [\mathbf{W} \mathbf{X}^\top \mathbf{X} (\bar{\beta} + \boldsymbol{\mu})]^\top + [-n \mathbf{W} \boldsymbol{\Sigma}_x \boldsymbol{\mu}] [-\bar{\beta}]^\top + [-n \mathbf{W} \boldsymbol{\Sigma}_x \boldsymbol{\mu}] [\mathbf{W} \mathbf{X}^\top \boldsymbol{\xi}]^\top + [-n \mathbf{W} \boldsymbol{\Sigma}_x \boldsymbol{\mu}] [-n \mathbf{W} \boldsymbol{\Sigma}_x \boldsymbol{\mu}]^\top.
 \end{aligned}$$

Take expectation of it,

$$\begin{aligned}
 \mathbb{E}[cc^\top] &= \left[ \mathbf{W} \underbrace{\mathbb{E}[\mathbf{X}^\top \mathbf{X} (\bar{\beta} + \boldsymbol{\mu}) (\bar{\beta} + \boldsymbol{\mu})^\top \mathbf{X}^\top \mathbf{X}]}_{\text{Lemma 1, denoted as a (task-specific) matrix } \mathbf{C}} \mathbf{W}^\top \right] + [-n \mathbf{W} \boldsymbol{\Sigma}_x \boldsymbol{\Sigma}_\beta] + 0 + [-n^2 \mathbf{W} \boldsymbol{\Sigma}_x \boldsymbol{\mu} \boldsymbol{\mu}^\top \boldsymbol{\Sigma}_x \mathbf{W}^\top] \\
 &\quad + [-n \boldsymbol{\Sigma}_\beta \boldsymbol{\Sigma}_x \mathbf{W}^\top] + \boldsymbol{\Sigma}_\beta + 0 + 0 \\
 &\quad + 0 + 0 + \underbrace{n \sigma^2 \mathbf{W} \boldsymbol{\Sigma}_x \mathbf{W}^\top}_{\text{Lemma 2}} + 0 \\
 &\quad + [-n^2 \mathbf{W} \boldsymbol{\Sigma}_x \boldsymbol{\mu} \boldsymbol{\mu}^\top \boldsymbol{\Sigma}_x \mathbf{W}^\top] + 0 + 0 + [n^2 \mathbf{W} \boldsymbol{\Sigma}_x \boldsymbol{\mu} \boldsymbol{\mu}^\top \boldsymbol{\Sigma}_x \mathbf{W}^\top] \\
 &= \mathbf{W} (\mathbf{C} + n \sigma^2 \boldsymbol{\Sigma}_x - n^2 \boldsymbol{\Sigma}_x \boldsymbol{\mu} \boldsymbol{\mu}^\top \boldsymbol{\Sigma}_x) \mathbf{W}^\top - n \mathbf{W} \boldsymbol{\Sigma}_x \boldsymbol{\Sigma}_\beta - n \boldsymbol{\Sigma}_\beta \boldsymbol{\Sigma}_x \mathbf{W}^\top + \boldsymbol{\Sigma}_\beta
 \end{aligned}$$

where  $\mathbf{C} = n \text{tr}(\boldsymbol{\Sigma}_x (\boldsymbol{\Sigma}_\beta + \boldsymbol{\mu} \boldsymbol{\mu}^\top)) \boldsymbol{\Sigma}_x + n(n+1) \boldsymbol{\Sigma}_x (\boldsymbol{\Sigma}_\beta + \boldsymbol{\mu} \boldsymbol{\mu}^\top) \boldsymbol{\Sigma}_x$ .

Substitute back into the loss  $\mathcal{L}(\mathbf{W}, \bar{\mathbf{p}}^*(\mathbf{W}))$ :

$$\begin{aligned}
 \mathcal{L}(\mathbf{W}, \bar{\mathbf{p}}^*(\mathbf{W})) &= \text{tr}(\mathbb{E}[cc^\top] \boldsymbol{\Sigma}_x) + \sigma^2 \\
 &= \text{tr}(\mathbf{W} (\mathbf{C} + n \sigma^2 \boldsymbol{\Sigma}_x - n^2 \boldsymbol{\Sigma}_x \boldsymbol{\mu} \boldsymbol{\mu}^\top \boldsymbol{\Sigma}_x) \mathbf{W}^\top \boldsymbol{\Sigma}_x) - 2n \text{tr}(\mathbf{W} \boldsymbol{\Sigma}_x \boldsymbol{\Sigma}_\beta \boldsymbol{\Sigma}_x) + \text{tr}(\boldsymbol{\Sigma}_\beta \boldsymbol{\Sigma}_x) + \sigma^2
 \end{aligned}$$

Use the following definition:

$$\begin{aligned}
 \text{Debiased: } \bar{\boldsymbol{\Sigma}}_\beta &= \boldsymbol{\Sigma}_x \sum_{k=1}^K \pi_k \mathbb{E}[(\boldsymbol{\beta}_k - \boldsymbol{\mu}_k)(\boldsymbol{\beta}_k - \boldsymbol{\mu}_k)^\top] = \boldsymbol{\Sigma}_x \sum_{k=1}^K \pi_k \boldsymbol{\Sigma}_{\beta_k}; \\
 \text{Biased: } \tilde{\boldsymbol{\Sigma}}_\beta &= \boldsymbol{\Sigma}_x \sum_{k=1}^K \pi_k \mathbb{E}[\boldsymbol{\beta}_k \boldsymbol{\beta}_k^\top] = \boldsymbol{\Sigma}_x \sum_{k=1}^K \pi_k (\boldsymbol{\Sigma}_{\beta_k} + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^\top).
 \end{aligned}$$

Denote:

$$\bar{\mathbf{C}} = \sum_{k=1}^K \pi_k \mathbf{C}_k = n \text{tr}(\tilde{\boldsymbol{\Sigma}}_\beta) \boldsymbol{\Sigma}_x + n(n+1) \tilde{\boldsymbol{\Sigma}}_\beta \boldsymbol{\Sigma}_x.$$

The weighted sum of the  $k$ -th task fine-tuning loss over all tasks  $k \in [K]$  is:

$$\mathcal{L}_{\text{FT}}(\mathbf{W}, \mathbf{P}_{\text{FT}}^*(\mathbf{W})) = \text{tr}(\mathbf{W} (\bar{\mathbf{C}} + n \sigma^2 \boldsymbol{\Sigma}_x - n^2 (\tilde{\boldsymbol{\Sigma}}_\beta - \bar{\boldsymbol{\Sigma}}_\beta) \boldsymbol{\Sigma}_x) \mathbf{W}^\top \boldsymbol{\Sigma}_x) - 2n \text{tr}(\mathbf{W} \bar{\boldsymbol{\Sigma}}_\beta \boldsymbol{\Sigma}_x) + \text{tr}(\bar{\boldsymbol{\Sigma}}_\beta) + \sigma^2$$

Note that for plain training,

$$\mathcal{L}_{\text{PT}}(\mathbf{W}, \mathbf{P} = \mathbf{0}) = \text{tr}(\mathbf{W} (\bar{\mathbf{C}} + n \sigma^2 \boldsymbol{\Sigma}_x) \mathbf{W}^\top \boldsymbol{\Sigma}_x) - 2n \text{tr}(\mathbf{W} \tilde{\boldsymbol{\Sigma}}_\beta \boldsymbol{\Sigma}_x) + \text{tr}(\tilde{\boldsymbol{\Sigma}}_\beta) + \sigma^2,$$

and their optimal loss share the same attention weight parameterization  $\mathbf{W} = \mathbf{W}_{\text{PT}}^*$ . Let  $\bar{\mathbf{W}}_{\text{PT}}^* = \boldsymbol{\Sigma}_x \mathbf{W}_{\text{PT}}^*$ ,

$$\begin{aligned}
 \mathcal{L}_{\text{FT}}^* &= \mathcal{L}_{\text{PT}}^* - \text{tr}(\tilde{\boldsymbol{\Sigma}}_\beta - \bar{\boldsymbol{\Sigma}}_\beta) + 2n \text{tr}(\bar{\mathbf{W}}_{\text{PT}}^* (\tilde{\boldsymbol{\Sigma}}_\beta - \bar{\boldsymbol{\Sigma}}_\beta)) - n^2 \text{tr}(\bar{\mathbf{W}}_{\text{PT}}^* (\tilde{\boldsymbol{\Sigma}}_\beta - \bar{\boldsymbol{\Sigma}}_\beta) \bar{\mathbf{W}}_{\text{PT}}^{*\top}) \\
 &= \mathcal{L}_{\text{PT}}^* - \text{tr}((\tilde{\boldsymbol{\Sigma}}_\beta - \bar{\boldsymbol{\Sigma}}_\beta) (n \bar{\mathbf{W}}_{\text{PT}}^* - \mathbf{I})^\top (n \bar{\mathbf{W}}_{\text{PT}}^* - \mathbf{I})).
 \end{aligned}$$

■

### B.3 Proof of Theorem 3

**Theorem 3 (Joint training)** Consider training a single-layer linear attention model in solving multi-task ICL problem with dataset defined in Definition 2 and model construction as described in Assumption 1 in the main paper. Let  $\mathbf{W}_{\text{JT}}^*, \mathbf{P}_{\text{JT}}^*$  (c.f. (12) in the main paper) be the optimal solutions and  $\mathcal{L}_{\text{JT}}^*$  is the optimal joint training loss defined in Section 3.3. Additionally, let  $\bar{\Sigma}_{\beta}, \tilde{\Sigma}_{\beta}, \mathbf{M}_{\mu}$  follow the same definitions as in Theorem 2 and define  $\bar{\mathbf{W}}_{\text{JT}}^* = \Sigma_x \mathbf{W}_{\text{JT}}^*$ . Then the solution  $(\mathbf{W}_{\text{JT}}^*, \mathbf{P}_{\text{JT}}^*)$  and optimal loss  $\mathcal{L}_{\text{JT}}^*$  satisfy

$$\begin{aligned}\bar{\mathbf{W}}_{\text{JT}}^* &= \bar{\Sigma}_{\beta} \left( (n+1) \bar{\Sigma}_{\beta} + (tr(\tilde{\Sigma}_{\beta}) + \sigma^2) \mathbf{I} + \Sigma_x \sum_{k=1}^K \pi_k \mu_k \mu_k^\top \right)^{-1}, \\ \mathbf{P}_{\text{JT}}^* &= \mathbf{M}_{\mu} ((\bar{\mathbf{W}}_{\text{JT}}^*)^{-1} - n \mathbf{I}) \Sigma_x, \\ \mathcal{L}_{\text{JT}}^* &= tr(\bar{\Sigma}_{\beta}) + \sigma^2 - n tr(\bar{\mathbf{W}}_{\text{JT}}^* \bar{\Sigma}_{\beta}).\end{aligned}$$

Here,  $\Sigma_x \sum_{k=1}^K \pi_k \mu_k \mu_k^\top \sim \mathcal{O}(1)$  is a  $d \times d$ -sized constant matrix.

**Proof.** As previously stated, the following derivation applies to all tasks  $k \in [K]$ . Therefore, for simplicity, we omit the index  $k$  in the notation unless otherwise specified.

#### 1. Determining the optimal task-specific prompts

In the joint training setting, the attention model is pretrained and parameterized by a trainable  $\mathbf{W}$ , and the task-specific prompts are fine-tuned accordingly (see (A5)):

$$\begin{aligned}\mathcal{L}(\mathbf{W}, \bar{\mathbf{p}}) &= \text{tr}(\mathbb{E}[\mathbf{c}\mathbf{c}^\top] \Sigma_x) + \sigma^2, \\ \text{where } \mathbf{c} &= (\mathbf{W}\mathbf{X}^\top \mathbf{X} - \mathbf{I})(\bar{\beta} + \mu) + \mathbf{W}\mathbf{X}^\top \xi + \mathbf{W}\bar{\mathbf{p}}.\end{aligned}$$

The optimal task-specific prompt is determined by taking the derivative and setting it to zero:

$$\begin{aligned}\frac{\partial \mathcal{L}(\mathbf{W}, \bar{\mathbf{p}})}{\partial \bar{\mathbf{p}}} &= 2 \mathbb{E} \left[ \frac{\partial \mathbf{c}^\top}{\partial \bar{\mathbf{p}}} \Sigma_x \mathbf{c} \right] \\ &= 2 \mathbf{W}^\top \Sigma_x \mathbb{E}[\mathbf{c}] = 2 \mathbf{W}^\top \Sigma_x [(n \mathbf{W} \Sigma_x - \mathbf{I}) \mu + \mathbf{W} \bar{\mathbf{p}}] = \mathbf{0}, \\ \Rightarrow \bar{\mathbf{p}}^* &= (\mathbf{W}^{-1} - n \Sigma_x) \mu,\end{aligned}$$

which is equivalent to:

$$\mathbf{P}_{\text{JT}}^* = \mathbf{M}_{\mu} ((\bar{\mathbf{W}}_{\text{JT}}^*)^{-1} - n \mathbf{I}) \Sigma_x.$$

#### 2. Determining the fine-tuning loss

It is worth noting that fine-tuning and joint training share a functional relationship between the tuned prompts  $\bar{\mathbf{p}}^*$  and the current attention model parameterization  $\mathbf{W}$ . Thus, substituting the tuned prompt back into the joint training loss will result in the same expression as the fine-tuning loss:

$$\begin{aligned}\mathcal{L}(\mathbf{W}, \bar{\mathbf{p}}^*(\mathbf{W})) &= \text{tr}(\mathbb{E}[\mathbf{c}\mathbf{c}^\top] \Sigma_x) + \sigma^2 \\ &= \text{tr}(\mathbf{W}(\mathbf{C} + n\sigma^2 \Sigma_x - n^2 \Sigma_x \mu \mu^\top \Sigma_x) \mathbf{W}^\top \Sigma_x) - 2n \text{tr}(\mathbf{W} \Sigma_x \Sigma_{\beta} \Sigma_x) + \text{tr}(\Sigma_{\beta} \Sigma_x)\end{aligned}$$

Use the following definition:

$$\begin{aligned}\text{Debiased: } \bar{\Sigma}_{\beta} &= \Sigma_x \sum_{k=1}^K \pi_k \mathbb{E}[(\beta_k - \mu_k)(\beta_k - \mu_k)^\top] = \Sigma_x \sum_{k=1}^K \pi_k \Sigma_{\beta_k}; \\ \text{Biased: } \tilde{\Sigma}_{\beta} &= \Sigma_x \sum_{k=1}^K \pi_k \mathbb{E}[\beta_k \beta_k^\top] = \Sigma_x \sum_{k=1}^K \pi_k (\Sigma_{\beta_k} + \mu_k \mu_k^\top).\end{aligned}$$

Denote:

$$\bar{C} = \sum_{k=1}^K \pi_k C_k = n \text{tr}(\tilde{\Sigma}_{\beta}) \Sigma_x + n(n+1) \tilde{\Sigma}_{\beta} \Sigma_x.$$

The weighted sum of the  $k$ -th task fine-tuning loss over all tasks  $k \in [K]$  is:

$$\mathcal{L}_{\text{JT}}(\mathbf{W}, \mathbf{P}_{\text{JT}}^*(\mathbf{W})) = \text{tr}(\mathbf{W}(\bar{\mathbf{C}} + n\sigma^2 \Sigma_{\mathbf{x}} - n^2(\tilde{\Sigma}_{\beta} - \bar{\Sigma}_{\beta})\Sigma_{\mathbf{x}})\mathbf{W}^\top \Sigma_{\mathbf{x}}) - 2n\text{tr}(\mathbf{W}\bar{\Sigma}_{\beta}\Sigma_{\mathbf{x}}) + \text{tr}(\bar{\Sigma}_{\beta}) + \sigma^2.$$

In joint training, the attention model parameterization  $\mathbf{W}$  is no longer fixed (as it is in the fine-tuning setting), but is instead optimized. Due to the functional relationship between the tuned prompts and parameterization  $\bar{\mathbf{p}}^* = \bar{\mathbf{p}}^*(\mathbf{W})$ , they will be optimized jointly until reaching their optimal values.

Take derivative w.r.t.  $\mathbf{W}$ , using Lemma 3:

$$\frac{\partial \mathcal{L}_{\text{JT}}(\mathbf{W}, \mathbf{P}_{\text{JT}}^*(\mathbf{W}))}{\partial \mathbf{W}} = 2\Sigma_{\mathbf{x}}\mathbf{W}(\bar{\mathbf{C}} + n\sigma^2 \Sigma_{\mathbf{x}} - n^2(\tilde{\Sigma}_{\beta} - \bar{\Sigma}_{\beta})\Sigma_{\mathbf{x}}) - 2n\bar{\Sigma}_{\beta}\Sigma_{\mathbf{x}}$$

Let  $\bar{\mathbf{W}}_{\text{JT}}^* = \Sigma_{\mathbf{x}}\mathbf{W}_{\text{JT}}^*$ , and set the derivative to zero (note that  $\sum_{k=1}^K \mu_k \mu_k^\top \sim \mathcal{O}(1)$ ):

$$\begin{aligned} \bar{\mathbf{W}}_{\text{JT}}^* &= n\bar{\Sigma}_{\beta}\Sigma_{\mathbf{x}} \left( \bar{\mathbf{C}} + n\sigma^2 \Sigma_{\mathbf{x}} - n^2(\tilde{\Sigma}_{\beta} - \bar{\Sigma}_{\beta})\Sigma_{\mathbf{x}} \right)^{-1} \\ &= \bar{\Sigma}_{\beta}\Sigma_{\mathbf{x}} \left( (\text{tr}(\tilde{\Sigma}_{\beta}) + \sigma^2)\Sigma_{\mathbf{x}} - n(\tilde{\Sigma}_{\beta} - \bar{\Sigma}_{\beta})\Sigma_{\mathbf{x}} + (n+1)\tilde{\Sigma}_{\beta}\Sigma_{\mathbf{x}} \right)^{-1} \\ &= \bar{\Sigma}_{\beta} \left( (\text{tr}(\tilde{\Sigma}_{\beta}) + \sigma^2)\mathbf{I} + (n+1)\bar{\Sigma}_{\beta} + (\tilde{\Sigma}_{\beta} - \bar{\Sigma}_{\beta}) \right)^{-1} \\ &= \boxed{\bar{\Sigma}_{\beta} \left( (n+1)\bar{\Sigma}_{\beta} + (\text{tr}(\tilde{\Sigma}_{\beta}) + \sigma^2)\mathbf{I} + \Sigma_{\mathbf{x}} \sum_{k=1}^K \pi_k \mu_k \mu_k^\top \right)^{-1}}. \end{aligned}$$

Substitute back into  $\mathcal{L}_{\text{JT}}(\mathbf{W}, \mathbf{P}^*(\mathbf{W}))$ :

$$\boxed{\mathcal{L}_{\text{JT}}^* = \text{tr}(\bar{\Sigma}_{\beta}) + \sigma^2 - n\text{tr}(\bar{\mathbf{W}}_{\text{JT}}^* \bar{\Sigma}_{\beta})}.$$

■

#### B.4 Proof of Corollary 1

**Corollary 1** Let  $\mathcal{L}_{\text{PT}}^*$ ,  $\mathcal{L}_{\text{FT}}^*$ , and  $\mathcal{L}_{\text{JT}}^*$  denote the optimal losses for plain training, fine-tuning, and joint training, as described in Theorems 1, 2 and 3, respectively. These losses satisfy:

$$\mathcal{L}_{\text{JT}}^* \leq \mathcal{L}_{\text{FT}}^* \leq \mathcal{L}_{\text{PT}}^*. \quad (\text{A6})$$

The equalities hold if and only if  $\bar{\Sigma}_{\beta} = \tilde{\Sigma}_{\beta}$  (c.f. (14)), which occurs when all task means  $\mu_k = \mathbf{0}$  for  $k \in [K]$ . Furthermore, the loss gaps satisfy the following:

1. The loss gaps scale quadratically with task mean:  $\mathcal{L}_{\text{PT}}^* - \mathcal{L}_{\text{FT}}^* \sim \mathcal{O}(\frac{1}{n^2}) \|\Delta\|_F$ ,  $\mathcal{L}_{\text{FT}}^* - \mathcal{L}_{\text{JT}}^* \sim \mathcal{O}(\frac{1}{n}) \|\Delta\|_F$ , where  $\Delta := \tilde{\Sigma}_{\beta} - \bar{\Sigma}_{\beta} = \Sigma_{\mathbf{x}} \sum_{k=1}^K \pi_k \mu_k \mu_k^\top$ .
2. The ratio between gaps is:  $\frac{\mathcal{L}_{\text{PT}}^* - \mathcal{L}_{\text{FT}}^*}{\mathcal{L}_{\text{FT}}^* - \mathcal{L}_{\text{JT}}^*} \sim \mathcal{O}(\frac{1}{n})$ , indicating that fine-tuning provides most of the benefit in few-shot regimes (small  $n$ ), while joint training benefits more for larger  $n$ .

**Proof.** 1.  $\mathcal{L}_{\text{FT}}^* \leq \mathcal{L}_{\text{PT}}^*$ :

From Theorem 2, we have:

$$\mathcal{L}_{\text{FT}}^* = \mathcal{L}_{\text{PT}}^* - \text{tr}((\tilde{\Sigma}_{\beta} - \bar{\Sigma}_{\beta})(n\bar{\mathbf{W}}_{\text{PT}}^* - \mathbf{I})^\top (n\bar{\mathbf{W}}_{\text{PT}}^* - \mathbf{I})).$$

Use the following definition:

$$\begin{aligned} \text{Debiased: } \bar{\Sigma}_{\beta} &= \Sigma_{\mathbf{x}} \sum_{k=1}^K \pi_k \mathbb{E}[(\beta_k - \mu_k)(\beta_k - \mu_k)^\top] = \Sigma_{\mathbf{x}} \sum_{k=1}^K \pi_k \Sigma_{\beta_k}; \\ \text{Biased: } \tilde{\Sigma}_{\beta} &= \Sigma_{\mathbf{x}} \sum_{k=1}^K \pi_k \mathbb{E}[\beta_k \beta_k^\top] = \Sigma_{\mathbf{x}} \sum_{k=1}^K \pi_k (\Sigma_{\beta_k} + \mu_k \mu_k^\top). \end{aligned}$$

We define an auxiliray variable:

$$\tilde{\Sigma}_{\beta} - \bar{\Sigma}_{\beta} = \Sigma_x \sum_{k=1}^K \pi_k \mu_k \mu_k^\top = \Delta$$

It can be seen that  $(\tilde{\Sigma}_{\beta} - \bar{\Sigma}_{\beta}) = \Sigma_x (\sum_{k=1}^K \pi_k \mu_k \mu_k^\top) \succeq \mathbf{0}$ ,  $(n \bar{W}_{\text{PT}}^* - \mathbf{I})^\top (n \bar{W}_{\text{PT}}^* - \mathbf{I}) \succ \mathbf{0}$ , which leads to

$$\text{tr}((\tilde{\Sigma}_{\beta} - \bar{\Sigma}_{\beta})(n \bar{W}_{\text{PT}}^* - \mathbf{I})^\top (n \bar{W}_{\text{PT}}^* - \mathbf{I})) \geq 0 \Rightarrow \mathcal{L}_{\text{FT}}^* \leq \mathcal{L}_{\text{PT}}^*.$$

The equality holds if and only if  $\boxed{\mu_k = \mathbf{0}, k \in [K] \iff \bar{\Sigma}_{\beta} = \tilde{\Sigma}_{\beta}}.$

The loss gap between  $\mathcal{L}_{\text{FT}}^*$  and  $\mathcal{L}_{\text{PT}}^*$  can be written as:

$$\mathcal{L}_{\text{PT}}^* - \mathcal{L}_{\text{FT}}^* = \text{tr}(\Delta(n \bar{W}_{\text{PT}}^* - \mathbf{I})^\top (n \bar{W}_{\text{PT}}^* - \mathbf{I}))$$

To analyze the asymptotic behavior of the gap, we need to analyze the asymptotic behavior of:

$$\bar{W}_{\text{PT}}^* = \tilde{\Sigma}_{\beta}((n+1)\tilde{\Sigma}_{\beta} + \text{tr}(\tilde{\Sigma}_{\beta})\mathbf{I})^{-1}.$$

First, for large  $n$ , we can factor out  $n$  from the inverse term:

$$\bar{W}_{\text{PT}}^* = \frac{1}{n} \tilde{\Sigma}_{\beta} \left( \tilde{\Sigma}_{\beta} \left( 1 + \frac{1}{n} \right) + \frac{\text{tr}(\tilde{\Sigma}_{\beta})}{n} \mathbf{I} \right)^{-1}$$

Using a matrix Taylor expansion, with  $\mathbf{A} = \tilde{\Sigma}_{\beta}$  and  $\mathbf{B} = \frac{\text{tr}(\tilde{\Sigma}_{\beta})}{n} \mathbf{I} + \frac{1}{n} \tilde{\Sigma}_{\beta}$ :

$$(\mathbf{A} + \mathbf{B})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1} \mathbf{B} \mathbf{A}^{-1} + \mathcal{O}(\|\mathbf{B}\|^2)$$

Applying this to our expression:

$$\left( \tilde{\Sigma}_{\beta} \left( 1 + \frac{1}{n} \right) + \frac{\text{tr}(\tilde{\Sigma}_{\beta})}{n} \mathbf{I} \right)^{-1} = \tilde{\Sigma}_{\beta}^{-1} - \frac{1}{n} \tilde{\Sigma}_{\beta}^{-1} \left( \tilde{\Sigma}_{\beta} + \text{tr}(\tilde{\Sigma}_{\beta}) \mathbf{I} \right) \tilde{\Sigma}_{\beta}^{-1} + \mathcal{O}\left(\frac{1}{n^2}\right)$$

Therefore:

$$\bar{W}_{\text{PT}}^* = \frac{1}{n} \mathbf{I} - \frac{1}{n^2} \left( \mathbf{I} + \text{tr}(\tilde{\Sigma}_{\beta}) \tilde{\Sigma}_{\beta}^{-1} \right) + \mathcal{O}\left(\frac{1}{n^3}\right)$$

The omitted  $\mathcal{O}(\frac{1}{n^3})$  terms include higher-order expansion terms from the matrix Taylor series. These terms involve powers of  $\tilde{\Sigma}_{\beta}$  and its inverse. They grow increasingly small as  $n$  increases and don't affect the dominant  $\mathcal{O}(\frac{1}{n^2})$  behavior of the loss gap.

Using the result above:

$$n \bar{W}_{\text{PT}}^* - \mathbf{I} = n \left( \frac{1}{n} \mathbf{I} - \frac{1}{n^2} \left( \mathbf{I} + \text{tr}(\tilde{\Sigma}_{\beta}) \tilde{\Sigma}_{\beta}^{-1} \right) + \mathcal{O}\left(\frac{1}{n^3}\right) \right) - \mathbf{I}$$

$$n \bar{W}_{\text{PT}}^* - \mathbf{I} = \mathbf{I} - \frac{1}{n} \left( \mathbf{I} + \text{tr}(\tilde{\Sigma}_{\beta}) \tilde{\Sigma}_{\beta}^{-1} \right) + \mathcal{O}\left(\frac{1}{n^2}\right) - \mathbf{I}$$

$$n \bar{W}_{\text{PT}}^* - \mathbf{I} = -\frac{1}{n} \left( \mathbf{I} + \text{tr}(\tilde{\Sigma}_{\beta}) \tilde{\Sigma}_{\beta}^{-1} \right) + \mathcal{O}\left(\frac{1}{n^2}\right)$$

The omitted terms at order  $\mathcal{O}(\frac{1}{n^2})$  include additional matrix products that become negligible for large  $n$ .

Therefore:

$$\|n \bar{W}_{\text{PT}}^* - \mathbf{I}\|_F \sim \mathcal{O}\left(\frac{1}{n}\right)$$

Now, substituting this into the expression for the gap:

$$\mathcal{L}_{\text{PT}}^* - \mathcal{L}_{\text{FT}}^* = \text{tr}(\Delta(n\bar{\mathbf{W}}_{\text{PT}}^* - \mathbf{I})^\top(n\bar{\mathbf{W}}_{\text{PT}}^* - \mathbf{I}))$$

$$\mathcal{L}_{\text{PT}}^* - \mathcal{L}_{\text{FT}}^* \sim \text{tr}(\Delta \cdot \mathcal{O}\left(\frac{1}{n^2}\right))$$

$$\mathcal{L}_{\text{PT}}^* - \mathcal{L}_{\text{FT}}^* \sim \mathcal{O}\left(\frac{1}{n^2}\right) \|\Delta\|_F$$

Given that  $\|\Delta\|_F \leq M^2 \sum_{k=1}^K \pi_k^2 \cdot \lambda_{\max}(\Sigma_x)$ , we have:

$$\boxed{\mathcal{L}_{\text{PT}}^* - \mathcal{L}_{\text{FT}}^* \sim \mathcal{O}\left(\frac{1}{n^2}\right) M^2 \sum_{k=1}^K \pi_k^2 \cdot \lambda_{\max}(\Sigma_x)}$$

## 2. $\mathcal{L}_{\text{JT}}^* \leq \mathcal{L}_{\text{FT}}^*$ :

From the proof of Theorem 3, it can be seen that joint training and fine-tuning shares a functional relationship between  $\mathcal{L}$  and  $\mathbf{W}$ :

$$\mathcal{L}_{\text{JT}}(\mathbf{W}, \mathbf{P}_{\text{JT}}^*(\mathbf{W})) = \mathcal{L}_{\text{FT}}(\mathbf{W}, \mathbf{P}_{\text{FT}}^*(\mathbf{W}))$$

However, the only minimizer of this function is derived from  $\frac{\partial \mathcal{L}_{\text{JT}}(\mathbf{W}, \mathbf{P}_{\text{JT}}^*(\mathbf{W}))}{\partial \mathbf{W}} = 0 \Rightarrow \mathbf{W}_{\text{JT}}^*$ , which leads to:

$$\mathcal{L}_{\text{JT}}^* \leq \mathcal{L}_{\text{FT}}^*.$$

The equality holds if and only if  $\boxed{\mathbf{W}_{\text{PT}}^* = \mathbf{W}_{\text{JT}}^* \iff \boldsymbol{\mu}_k = \mathbf{0}, k \in [K] \iff \bar{\Sigma}_{\boldsymbol{\beta}} = \tilde{\Sigma}_{\boldsymbol{\beta}}}$ .

Recall that  $\bar{\mathbf{W}}_{\text{JT}}^*$  is given by:

$$\bar{\mathbf{W}}_{\text{JT}}^* = \bar{\Sigma}_{\boldsymbol{\beta}}((n+1)\bar{\Sigma}_{\boldsymbol{\beta}} + \text{tr}(\tilde{\Sigma}_{\boldsymbol{\beta}})\mathbf{I} + \Sigma_x \sum_{k=1}^K \pi_k \boldsymbol{\mu}_k \boldsymbol{\mu}_k^\top)^{-1}$$

Note that we denote  $\Delta = \Sigma_x \sum_{k=1}^K \pi_k \boldsymbol{\mu}_k \boldsymbol{\mu}_k^\top$ , so:

$$\begin{aligned} \bar{\mathbf{W}}_{\text{JT}}^* &= \bar{\Sigma}_{\boldsymbol{\beta}}((n+1)\bar{\Sigma}_{\boldsymbol{\beta}} + \text{tr}(\bar{\Sigma}_{\boldsymbol{\beta}} + \Delta)\mathbf{I} + \Delta)^{-1} \\ &= \bar{\Sigma}_{\boldsymbol{\beta}}((n+1)\bar{\Sigma}_{\boldsymbol{\beta}} + \text{tr}(\bar{\Sigma}_{\boldsymbol{\beta}})\mathbf{I} + \text{tr}(\Delta)\mathbf{I} + \Delta)^{-1} \end{aligned}$$

First, we rewrite this for large  $n$ :

$$\bar{\mathbf{W}}_{\text{JT}}^* = \frac{1}{n} \bar{\Sigma}_{\boldsymbol{\beta}} \left( \bar{\Sigma}_{\boldsymbol{\beta}} \left( 1 + \frac{1}{n} \right) + \frac{\text{tr}(\bar{\Sigma}_{\boldsymbol{\beta}})}{n} \mathbf{I} + \frac{\text{tr}(\Delta)}{n} \mathbf{I} + \frac{\Delta}{n} \right)^{-1}$$

Let  $\mathbf{A} = \bar{\Sigma}_{\boldsymbol{\beta}}$  and  $\mathbf{B} = \frac{1}{n} \bar{\Sigma}_{\boldsymbol{\beta}} + \frac{\text{tr}(\bar{\Sigma}_{\boldsymbol{\beta}})}{n} \mathbf{I} + \frac{\text{tr}(\Delta)}{n} \mathbf{I} + \frac{\Delta}{n}$ .

Using the matrix Taylor expansion for  $(\mathbf{A} + \mathbf{B})^{-1}$ :

$$(\mathbf{A} + \mathbf{B})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1} \mathbf{B} \mathbf{A}^{-1} + \mathbf{A}^{-1} \mathbf{B} \mathbf{A}^{-1} \mathbf{B} \mathbf{A}^{-1} + \mathcal{O}(\|\mathbf{B}\|^3)$$

Applying this to our expression and noting that  $\|\mathbf{B}\| = \mathcal{O}(1/n)$ :

$$\left( \bar{\Sigma}_{\boldsymbol{\beta}} \left( 1 + \frac{1}{n} \right) + \frac{\text{tr}(\bar{\Sigma}_{\boldsymbol{\beta}})}{n} \mathbf{I} + \frac{\text{tr}(\Delta)}{n} \mathbf{I} + \frac{\Delta}{n} \right)^{-1} = \bar{\Sigma}_{\boldsymbol{\beta}}^{-1} - \frac{1}{n} \bar{\Sigma}_{\boldsymbol{\beta}}^{-1} (\bar{\Sigma}_{\boldsymbol{\beta}} + \text{tr}(\bar{\Sigma}_{\boldsymbol{\beta}})\mathbf{I} + \text{tr}(\Delta)\mathbf{I} + \Delta) \bar{\Sigma}_{\boldsymbol{\beta}}^{-1} + \mathcal{O}\left(\frac{1}{n^2}\right)$$

Therefore:

$$\bar{W}_{\text{JT}}^* = \frac{1}{n} \mathbf{I} - \frac{1}{n^2} \left( \mathbf{I} + \text{tr}(\bar{\Sigma}_{\beta}) \bar{\Sigma}_{\beta}^{-1} + \text{tr}(\Delta) \bar{\Sigma}_{\beta}^{-1} + \bar{\Sigma}_{\beta}^{-1} \Delta \right) + \mathcal{O} \left( \frac{1}{n^3} \right)$$

The omitted  $\mathcal{O}(\frac{1}{n^3})$  terms include higher-order matrix products involving powers of  $\bar{\Sigma}_{\beta}$ ,  $\Delta$ , and their inverses. These become negligible as  $n$  grows.

Using the result above:

$$\begin{aligned} n \bar{W}_{\text{JT}}^* - \mathbf{I} &= n \left( \frac{1}{n} \mathbf{I} - \frac{1}{n^2} \left( \mathbf{I} + \text{tr}(\bar{\Sigma}_{\beta}) \bar{\Sigma}_{\beta}^{-1} + \text{tr}(\Delta) \bar{\Sigma}_{\beta}^{-1} + \bar{\Sigma}_{\beta}^{-1} \Delta \right) + \mathcal{O} \left( \frac{1}{n^3} \right) \right) - \mathbf{I} \\ &= -\frac{1}{n} \left( \mathbf{I} + \text{tr}(\bar{\Sigma}_{\beta}) \bar{\Sigma}_{\beta}^{-1} + \text{tr}(\Delta) \bar{\Sigma}_{\beta}^{-1} + \bar{\Sigma}_{\beta}^{-1} \Delta \right) + \mathcal{O} \left( \frac{1}{n^2} \right) \end{aligned}$$

Therefore:

$$\|n \bar{W}_{\text{JT}}^* - \mathbf{I}\|_F = \mathcal{O} \left( \frac{1}{n} \right)$$

Recall that:

$$\bar{W}_{\text{PT}}^* = \frac{1}{n} \mathbf{I} - \frac{1}{n^2} \left( \mathbf{I} + \text{tr}(\bar{\Sigma}_{\beta}) \bar{\Sigma}_{\beta}^{-1} \right) + \mathcal{O} \left( \frac{1}{n^3} \right)$$

$$\bar{W}_{\text{JT}}^* = \frac{1}{n} \mathbf{I} - \frac{1}{n^2} \left( \mathbf{I} + \text{tr}(\bar{\Sigma}_{\beta}) \bar{\Sigma}_{\beta}^{-1} + \text{tr}(\Delta) \bar{\Sigma}_{\beta}^{-1} + \bar{\Sigma}_{\beta}^{-1} \Delta \right) + \mathcal{O} \left( \frac{1}{n^3} \right)$$

The leading terms  $(1/n)\mathbf{I}$  cancel, and the difference appears in the second-order terms:

$$\bar{W}_{\text{PT}}^* - \bar{W}_{\text{JT}}^* = \frac{1}{n^2} \mathbf{C} + \mathcal{O} \left( \frac{1}{n^3} \right)$$

Where  $\mathbf{C}$  is a matrix that depends on  $\bar{\Sigma}_{\beta}$  and  $\Delta$ . Therefore:

$$\|\bar{W}_{\text{PT}}^* - \bar{W}_{\text{JT}}^*\|_F \sim \mathcal{O} \left( \frac{1}{n^2} \right) \|\Delta\|_F$$

Starting from the loss function definition:

$$\mathcal{L}(\mathbf{W}) = \text{tr}(\bar{\Sigma}_{\beta}) - n \text{tr}(\mathbf{W} \bar{\Sigma}_{\beta})$$

The loss gap becomes:

$$\mathcal{L}_{\text{FT}}^* - \mathcal{L}_{\text{JT}}^* = n \|\text{tr}((\bar{W}_{\text{PT}}^* - \bar{W}_{\text{JT}}^*) \bar{\Sigma}_{\beta})\|$$

This gives us:

$$\mathcal{L}_{\text{FT}}^* - \mathcal{L}_{\text{JT}}^* \sim \mathcal{O} \left( \frac{1}{n} \right) M^2 \sum_{k=1}^K \pi_k^2 \cdot \lambda_{\max}(\Sigma_x)$$

### 3. Comparative Analysis

1. Plain Training vs Fine-tuning:

$$\mathcal{L}_{\text{PT}}^* - \mathcal{L}_{\text{FT}}^* \sim \mathcal{O} \left( \frac{1}{n^2} \right) M^2 \sum_{k=1}^K \pi_k^2 \cdot \lambda_{\max}(\Sigma_x)$$

2. Fine-tuning vs Joint Training:

$$\mathcal{L}_{\text{FT}}^* - \mathcal{L}_{\text{JT}}^* \sim \mathcal{O} \left( \frac{1}{n} \right) M^2 \sum_{k=1}^K \pi_k^2 \cdot \lambda_{\max}(\Sigma_x)$$

3. Ratio of gaps:

$$\frac{\mathcal{L}_{\text{PT}}^* - \mathcal{L}_{\text{FT}}^*}{\mathcal{L}_{\text{FT}}^* - \mathcal{L}_{\text{JT}}^*} \sim \mathcal{O}\left(\frac{1}{n}\right)$$

This aligns with our experimental results in Section 6, indicating that for few-shot multi-task in-context learning settings (e.g., when  $n = 1$ ), fine-tuning task-specific prompts alone is sufficiently effective at improving performance. However, for many-shot (large  $n$ ) settings, joint training of both the attention weight and task-specific prompts is necessary to achieve further performance improvements.  $\blacksquare$

## C Proofs for Section 5

### C.1 Proof of Proposition 1

**Proposition 1** Consider the multi-task ICL data as described in Definition 2 and let  $\tilde{\mathcal{L}}_{\text{Attn}}^*$  and  $\tilde{\mathcal{L}}_{\text{PGD}}^*$  be the optimal linear attention and debiased preconditioned gradient descent losses as presented in (18) and (19) in the main paper, respectively. Then,  $\tilde{\mathcal{L}}_{\text{Attn}}^* = \tilde{\mathcal{L}}_{\text{PGD}}^*$ .

**Proof.** To begin with, let attention weights be

$$\mathbf{W}_q \mathbf{W}_k^\top = \begin{bmatrix} \bar{\mathbf{W}}_1 & \mathbf{w}_1 \\ * & * \end{bmatrix} \quad \text{and} \quad \mathbf{W}_v = \begin{bmatrix} \bar{\mathbf{W}}_2 & \mathbf{w}_2 \\ \mathbf{w}_3^\top & w \end{bmatrix},$$

where  $\bar{\mathbf{W}}_{1,2} \in \mathbb{R}^{d \times d}$ ,  $\mathbf{w}_{1,2,3} \in \mathbb{R}^d$  and  $w \in \mathbb{R}$ . Additionally, let task-specific prompts and heads be

$$\mathbf{p}_k = \begin{bmatrix} \bar{\mathbf{p}}_k \\ p_k \end{bmatrix} \quad \text{and} \quad \mathbf{h}_k = \begin{bmatrix} \bar{\mathbf{h}}_k \\ h_k \end{bmatrix} \quad \text{for } k \in [K],$$

where  $\bar{\mathbf{p}}_k, \bar{\mathbf{h}}_k \in \mathbb{R}^d$  and  $p_k, h_k \in \mathbb{R}$ . Recapping the prediction from (17) in the main paper and input sequence  $\mathbf{Z}^{(k)}$  from (5) in the main paper, we obtain

$$\begin{aligned} \tilde{f}_{\text{Attn}}(\mathbf{Z}^{(k)}) &= (\mathbf{z}^\top \mathbf{W}_q \mathbf{W}_k^\top (\mathbf{Z}^{(k)})^\top) \mathbf{M} \mathbf{Z}^{(k)} \mathbf{W}_v \mathbf{h}_k \\ &= [\mathbf{x}^\top \bar{\mathbf{W}}_1 \quad \mathbf{x}^\top \mathbf{w}_1] \left( \begin{bmatrix} \bar{\mathbf{p}}_k \bar{\mathbf{p}}_k^\top & p_k \bar{\mathbf{p}}_k \\ p_k \bar{\mathbf{p}}_k^\top & p_k^2 \end{bmatrix} + \begin{bmatrix} \mathbf{X}^\top \mathbf{X} & \mathbf{X}^\top \mathbf{y} \\ \mathbf{y}^\top \mathbf{X} & \|\mathbf{y}\|_{\ell_2}^2 \end{bmatrix} \right) \begin{bmatrix} \bar{\mathbf{W}}_2 \bar{\mathbf{h}}_k + h_k \mathbf{w}_2 \\ \mathbf{w}_3^\top \mathbf{h}_k + h_k w \end{bmatrix}. \end{aligned}$$

Recap the task distribution from Definition 2. Let  $\mathbf{y}_0 = \mathbf{y} - \mathbf{X}\boldsymbol{\mu}_k$ . For cleaner notation and without loss of generality, we remove the subscription  $k$  and set

$$\begin{bmatrix} \bar{\mathbf{W}}_2 \bar{\mathbf{h}}_k + h_k \mathbf{w}_2 \\ \mathbf{w}_3^\top \mathbf{h}_k + h_k w \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ v \end{bmatrix}$$

where  $\mathbf{v} \in \mathbb{R}^d$  and  $v \in \mathbb{R}$ .

$$\begin{aligned} \tilde{f}_{\text{Attn}}(\mathbf{Z}^{(k)}) &= [\mathbf{x}^\top \bar{\mathbf{W}}_1 \quad \mathbf{x}^\top \mathbf{w}_1] \left( \begin{bmatrix} \bar{\mathbf{p}} \bar{\mathbf{p}}^\top & p \bar{\mathbf{p}} \\ p \bar{\mathbf{p}}^\top & p^2 \end{bmatrix} + \begin{bmatrix} \mathbf{X}^\top \mathbf{X} & \mathbf{X}^\top \mathbf{y} \\ \mathbf{y}^\top \mathbf{X} & \|\mathbf{y}\|_{\ell_2}^2 \end{bmatrix} \right) \begin{bmatrix} \mathbf{v} \\ v \end{bmatrix} \\ &= [\mathbf{x}^\top \bar{\mathbf{W}}_1 \quad \mathbf{x}^\top \mathbf{w}_1] \begin{bmatrix} \bar{\mathbf{p}} \bar{\mathbf{p}}^\top & p \bar{\mathbf{p}} \\ p \bar{\mathbf{p}}^\top & p^2 \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ v \end{bmatrix} + [\mathbf{x}^\top \bar{\mathbf{W}}_1 \quad \mathbf{x}^\top \mathbf{w}_1] \begin{bmatrix} \mathbf{X}^\top \mathbf{X} & \mathbf{X}^\top \mathbf{y} \\ \mathbf{y}^\top \mathbf{X} & \|\mathbf{y}\|_{\ell_2}^2 \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ v \end{bmatrix} \\ &= \mathbf{x}^\top (\bar{\mathbf{W}}_1 \bar{\mathbf{p}} \bar{\mathbf{p}}^\top \mathbf{v} + p \mathbf{w}_1 \bar{\mathbf{p}}^\top \mathbf{v} + p v \bar{\mathbf{W}}_1 \bar{\mathbf{p}} + p^2 v \mathbf{w}_1) + \mathbf{x}^\top (\bar{\mathbf{W}}_1 \mathbf{X}^\top \mathbf{X} \mathbf{v} + \mathbf{w}_1 \mathbf{y}^\top \mathbf{X} \mathbf{v} + v \bar{\mathbf{W}}_1 \mathbf{X}^\top \mathbf{y} + v \|\mathbf{y}\|_{\ell_2}^2 \mathbf{w}_1) \\ &= \mathbf{x}^\top \tilde{\mathbf{p}} + \mathbf{x}^\top (\bar{\mathbf{W}}_1 \mathbf{X}^\top \mathbf{X} \mathbf{v} + (\mathbf{w}_1 \mathbf{v}^\top + v \bar{\mathbf{W}}_1) \mathbf{X}^\top \mathbf{y} + v \|\mathbf{y}\|_{\ell_2}^2 \mathbf{w}_1) \\ &= \mathbf{x}^\top (\tilde{\mathbf{p}} + \bar{\mathbf{W}}_1 \mathbf{X}^\top \mathbf{X} \mathbf{v} + (\mathbf{w}_1 \mathbf{v}^\top + v \bar{\mathbf{W}}_1) \mathbf{X}^\top (\mathbf{y}_0 + \mathbf{X}\boldsymbol{\mu}) + v \mathbf{w}_1 (\|\mathbf{y}_0\|_{\ell_2}^2 + \boldsymbol{\mu}^\top \mathbf{X}^\top \mathbf{X} \boldsymbol{\mu} + 2 \boldsymbol{\mu}^\top \mathbf{X}^\top \mathbf{y}_0)) \\ &= \mathbf{x}^\top \tilde{\mathbf{W}} \mathbf{X}^\top \mathbf{y}_0 + \mathbf{x}^\top \tilde{\mathbf{p}} + \mathbf{x}^\top \underbrace{(\bar{\mathbf{W}}_1 \mathbf{X}^\top \mathbf{X} \mathbf{v} + (\tilde{\mathbf{W}} - v \mathbf{w}_1 \boldsymbol{\mu}^\top) \mathbf{X}^\top \mathbf{X} \boldsymbol{\mu} + v \mathbf{w}_1 \|\mathbf{y}_0\|_{\ell_2}^2)}_{\varepsilon(\mathbf{X}, \mathbf{y}_0)} \end{aligned}$$

where

$$\begin{aligned}\tilde{\mathbf{p}} &:= \bar{\mathbf{W}}_1 \bar{\mathbf{p}} \bar{\mathbf{p}}^\top \mathbf{v} + p \mathbf{w}_1 \bar{\mathbf{p}}^\top \mathbf{v} + p v \bar{\mathbf{W}}_1 \bar{\mathbf{p}} + p^2 v \mathbf{w}_1 \\ \tilde{\mathbf{W}} &:= \mathbf{w}_1 \mathbf{v}^\top + v \mathbf{W}_1 + 2 v \mathbf{w}_1 \boldsymbol{\mu}^\top.\end{aligned}$$

Then letting  $y_0 = y - \mathbf{x}^\top \boldsymbol{\mu}_k$ , the expected risk of task  $k$  obeys

$$\begin{aligned}\mathbb{E}_{\mathbf{Z}, y \sim \mathcal{D}_k} [(\tilde{f}_{\text{Attn}}(\mathbf{Z}^{(k)}) - y)^2] &= \mathbb{E} \left[ \left( \mathbf{x}^\top \tilde{\mathbf{W}} \mathbf{X}^\top \mathbf{y}_0 - y_0 + \mathbf{x}^\top \tilde{\mathbf{p}} - \mathbf{x}^\top \boldsymbol{\mu} + \mathbf{x}^\top \varepsilon(\mathbf{X}, \mathbf{y}_0) \right)^2 \right] \\ &= \mathbb{E} \left[ \left( \mathbf{x}^\top \tilde{\mathbf{W}} \mathbf{X}^\top \mathbf{y}_0 - y_0 \right)^2 \right] + \mathbb{E} \left[ \left( \mathbf{x}^\top \tilde{\mathbf{p}} - \mathbf{x}^\top \boldsymbol{\mu} + \mathbf{x}^\top \varepsilon(\mathbf{X}, \mathbf{y}_0) \right)^2 \right] \\ &\quad + 2 \mathbb{E} \left[ \left( \mathbf{x}^\top \tilde{\mathbf{W}} \mathbf{X}^\top \mathbf{y}_0 - y_0 \right) \left( \mathbf{x}^\top \tilde{\mathbf{p}} - \mathbf{x}^\top \boldsymbol{\mu} + \mathbf{x}^\top \varepsilon(\mathbf{X}, \mathbf{y}_0) \right) \right]\end{aligned}$$

Note that, letting  $\boldsymbol{\beta}_0 = \boldsymbol{\beta} - \boldsymbol{\mu}_k$ , we have  $\mathbf{y}_0 = \mathbf{X} \boldsymbol{\beta}_0 + \boldsymbol{\xi}$ ,  $y = \mathbf{x}^\top \boldsymbol{\beta}_0 + \xi_{n+1}$  and  $\boldsymbol{\beta}_0 \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{\beta}_k})$ . Therefore,

$$\begin{aligned}&\mathbb{E} \left[ \left( \mathbf{x}^\top \tilde{\mathbf{W}} \mathbf{X}^\top \mathbf{y}_0 - y_0 \right) \left( \mathbf{x}^\top \tilde{\mathbf{p}} - \mathbf{x}^\top \boldsymbol{\mu} + \mathbf{x}^\top \varepsilon(\mathbf{X}, \mathbf{y}_0) \right) \right] \\ &= \mathbb{E} \left[ \mathbf{x}^\top \left( \tilde{\mathbf{W}} \mathbf{X}^\top \mathbf{X} \boldsymbol{\beta}_0 - \boldsymbol{\beta}_0 \right) \left( \tilde{\mathbf{p}} - \boldsymbol{\mu} + \varepsilon(\mathbf{X}, \mathbf{y}_0) \right)^\top \mathbf{x} \right] \\ &= \mathbb{E} \left[ \mathbf{x}^\top \left( \tilde{\mathbf{W}} \mathbf{X}^\top \mathbf{X} - \mathbf{I} \right) \boldsymbol{\beta}_0 \varepsilon(\mathbf{X}, \mathbf{y}_0)^\top \mathbf{x} \right] \\ &= \mathbb{E} \left[ \mathbf{x}^\top \left( \tilde{\mathbf{W}} \mathbf{X}^\top \mathbf{X} - \mathbf{I} \right) \boldsymbol{\beta}_0 \left( v \|\mathbf{X} \boldsymbol{\beta}_0\|_{\ell_2}^2 \right) \mathbf{w}_1^\top \mathbf{x} \right] \\ &= 0.\end{aligned}$$

Then the risk satisfies

$$\begin{aligned}\mathbb{E}_{\mathbf{Z}, y \sim \mathcal{D}_k} [(\tilde{f}_{\text{Attn}}(\mathbf{Z}^{(k)}) - y)^2] &= \mathbb{E} \left[ \left( \mathbf{x}^\top \tilde{\mathbf{W}} \mathbf{X}^\top \mathbf{y}_0 - y_0 \right)^2 \right] + \mathbb{E} \left[ \left( \mathbf{x}^\top \tilde{\mathbf{p}} - \mathbf{x}^\top \boldsymbol{\mu} + \mathbf{x}^\top \varepsilon(\mathbf{X}, \mathbf{y}_0) \right)^2 \right] \\ &\geq \mathbb{E} \left[ \left( \mathbf{x}^\top \tilde{\mathbf{W}} \mathbf{X}^\top \mathbf{y}_0 - y_0 \right)^2 \right].\end{aligned}$$

We next prove that the equality is achievable for any  $\tilde{\mathbf{W}}$ . Consider the following constructions:

$$\mathbf{W}_q \mathbf{W}_k^\top = \begin{bmatrix} \mathbf{W} & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{bmatrix}, \quad \mathbf{W}_v = \mathbf{I}, \quad \mathbf{p}_k = \begin{bmatrix} \mathbf{W}^{-1} \boldsymbol{\mu}_k \\ \boldsymbol{\mu}_k^\top \mathbf{W}^{-\top} \boldsymbol{\mu}_k + 1 \end{bmatrix} \quad \text{and} \quad \mathbf{h}_k = \begin{bmatrix} -\boldsymbol{\mu}_k \\ 1 \end{bmatrix}.$$

Then

$$\begin{bmatrix} \mathbf{v} \\ v \end{bmatrix} = \begin{bmatrix} -\boldsymbol{\mu}_k \\ 1 \end{bmatrix}, \quad \tilde{\mathbf{p}} = \boldsymbol{\mu}_k, \quad \text{and} \quad \tilde{\mathbf{W}} = \mathbf{W}.$$

Using above construction, we obtain that for any  $\mathbf{W} \in \mathbb{R}^{d \times d}$ , there exist  $\mathbf{p}_k$ 's and  $\mathbf{h}_k$ 's such that

$$\mathbb{E}_{\mathbf{Z}, y \sim \mathcal{D}_k} \left[ \left( \mathbf{x}^\top \tilde{\mathbf{p}} - \mathbf{x}^\top \boldsymbol{\mu} + \mathbf{x}^\top \varepsilon(\mathbf{X}, \mathbf{y}_0) \right)^2 \right] = 0$$

and hence,

$$\mathbb{E}_{\mathbf{Z}, y \sim \mathcal{D}_k} [(\tilde{f}_{\text{Attn}}(\mathbf{Z}^{(k)}) - y)^2] = \mathbb{E} \left[ \left( \mathbf{x}^\top \tilde{\mathbf{W}} \mathbf{X}^\top \mathbf{y}_0 - y_0 \right)^2 \right].$$

Next, consider the preconditioned gradient descent problem defined in (19) in the main paper. Recapping the PGD prediction where we have

$$\tilde{f}_{\text{PGD}}(\mathbf{Z}^{(k)}) = \mathbf{x}^\top \mathbf{W} \mathbf{X}^\top (\mathbf{y} - \mathbf{X} \boldsymbol{\mu}_k) + \mathbf{x}^\top \boldsymbol{\mu}_k.$$

Then

$$\begin{aligned}\mathbb{E}_{\mathbf{Z}, y \sim \mathcal{D}_k} [(\tilde{f}_{\text{PGD}}(\mathbf{Z}^{(k)}) - y)^2] &= \mathbb{E}_{\mathbf{Z}, y \sim \mathcal{D}_k} [(\mathbf{x}^\top \mathbf{W} \mathbf{X}^\top (\mathbf{y} - \mathbf{X} \boldsymbol{\mu}_k) + \mathbf{x}^\top \boldsymbol{\mu}_k - y)^2] \\ &= \mathbb{E}_{\mathbf{Z}, y \sim \mathcal{D}_k} [(\mathbf{x}^\top \mathbf{W} \mathbf{X}^\top \mathbf{y}_0 - y_0)^2].\end{aligned}$$

Combining the results together completes the proof. ■

## C.2 Proof of Theorem 4

**Theorem 4** Consider the multi-task ICL problem with dataset defined in Definition 2. Let  $\mathbf{W}_{\text{PGD}}^* := \arg \min_{\mathbf{W}} \mathcal{L}(\tilde{f}_{\text{PGD}})$  following (19) in the main paper. Define  $\bar{\Sigma}_{\beta}$  in (14) in the main paper and let  $\bar{\mathbf{W}}_{\text{PGD}}^* = \Sigma_{\alpha} \mathbf{W}_{\text{PGD}}^*$ . Then the solution  $\bar{\mathbf{W}}_{\text{PGD}}^*$  and optimal loss  $\tilde{\mathcal{L}}_{\text{PGD}}^*$  (c.f. (19) in the main paper) satisfy

$$\bar{\mathbf{W}}_{\text{PGD}}^* = \bar{\Sigma}_{\beta} ((n+1)\bar{\Sigma}_{\beta} + (tr(\bar{\Sigma}_{\beta}) + \sigma^2)\mathbf{I})^{-1},$$

$$\tilde{\mathcal{L}}_{\text{PGD}}^* = tr(\bar{\Sigma}_{\beta}) - n tr(\bar{\mathbf{W}}_{\text{PGD}}^* \bar{\Sigma}_{\beta}).$$

**Proof.** From the proof of Proposition 1, it can be seen that the optimal multi-task ICL learning performance of a 1-layer linear attention model can be rigorously calculated as:  $\tilde{\mathcal{L}}_{\text{Attn}}^* = \tilde{\mathcal{L}}_{\text{PGD}}^*$ . Moreover, in this case, with the help of task-specific heads  $\mathbf{h}_k$ , the optimal loss  $\tilde{\mathcal{L}}_{\text{PGD}}^*$  is equivalent to the optimal plain training loss in a **zero task mean** multi-task ICL setting.

By applying Theorem 1 with all task means  $\boldsymbol{\mu}_k = \mathbf{0}$  for  $k \in [K]$ , Theorem 4 can be proven.  $\blacksquare$

## D Additional experiments: noisy label and non-isotropic covariance

We conduct experiments on synthetic datasets to validate our theoretical assumptions and explore the behavior of single-layer linear attention models with various trainable parameters under different training settings.

**Experimental Setting.** We train single-layer attention models to solve  $K$ -task,  $d$ -dimensional linear regression ICL with noise level  $\sigma^2 = 5$ . For each context length  $n$ , an independent model is trained for 20,000 iterations with a batch size of 8192 using the Adam optimizer (learning rate  $10^{-3}$ ).

To ensure robustness, each training process is repeated 50 times with independent initializations, and the minimal test risk among these trials is reported. Theoretical predictions in the plots are based on the theorems in Section 4, and all results are normalized by  $\mathbb{E}[\|y\|^2]$ .

### D.1 Noisy labels

We validate our theoretical assumptions and predictions based on a noise-free setting. To test these assumptions in a noisy label setting, where  $\sigma^2 > 0$ , we repeat the experiments from the main paper under the noisy setting, using the same experimental configurations as in Figure 2(a) and Figure 2(c), to validate Assumption 1 and Theorems 1, 2, 3, and 4.

### D.2 Non-isotropic covariance

We also validate our experiments under a non-isotropic covariance setting. At noise level  $\sigma^2 = 5$ , we repeat the experiments from the main paper under the noisy setting, using the same experimental configurations as in Figure 2(a) and Figure 2(d), except that the isotropic covariance multiplier  $\mathbf{I}_{10}$  is replaced with a non-isotropic one:

$$\mathbf{I}_{10} \longrightarrow \text{diag}\{0.9, 0.9^{-1}, \dots, 0.9^{-9}\}$$

This is done to validate Assumption 1 and Theorems 1, 2, 3, and 4.

As seen from Figure A1 (a)(c), the performance of the reduced model derived from Assumption 1 aligns perfectly with the performance of the exact linear attention model, indicating that it serves as a good proxy. Furthermore, from Figure A1 (b)(d), our theoretical predictions align perfectly with the performance of the linear attention model.

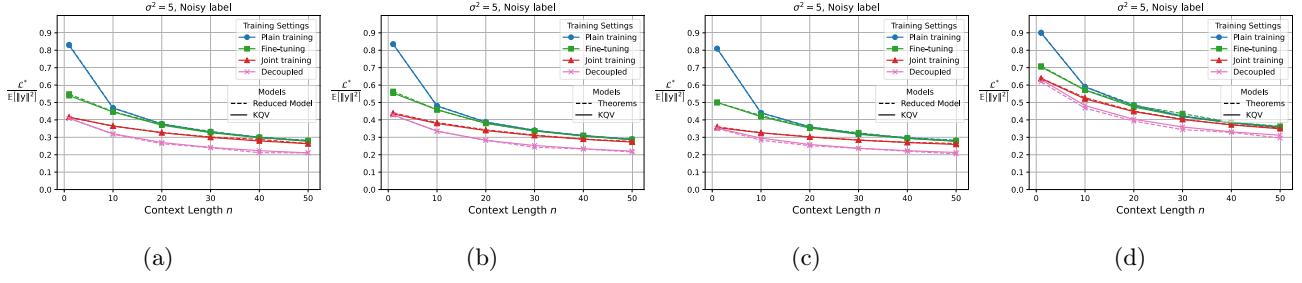


Figure A1: Experimental results across various settings: (a)(b) Noisy labels; (c)(d) Non-isotropic covariance. (a)(c) validate Assumption 1 in the main paper, and (b)(d) validate Theorems 1-4 in the main paper.

## E Additional experiments: multi-layer linear attention models

### E.1 Multi-layer linear attention model

In the main paper, Section 3, the output of a single-layer linear attention model is defined as:

$$\text{Attn}(\mathbf{Z}^{(k)}) = (\mathbf{Z}^{(k)} \mathbf{W}_q \mathbf{W}_k^\top (\mathbf{Z}^{(k)})^\top) \mathbf{M} \mathbf{Z}^{(k)} \mathbf{W}_v.$$

If task-specific heads  $\mathbf{h}_k, k \in [K]$  (see Section 5) are used, the prediction for  $\mathbf{Z}^{(k)}$  is (where  $\mathbf{e}_i$  is a one-hot indicator vector with 1 at the  $i$ -th position and 0 elsewhere):

$$\hat{y} = \mathbf{e}_{n+1}^\top \text{Attn}(\mathbf{Z}^{(k)}) \mathbf{h}_k,$$

otherwise,

$$\hat{y} = \mathbf{e}_{n+1}^\top \text{Attn}(\mathbf{Z}^{(k)}) \mathbf{e}_{d+1}.$$

Next, we extend the network architecture we used throughout this section. For a model with  $L \geq 2$  layers, we define an  **$L$ -layer linear attention model** as a stack of  $L$  single-layer attention models. Formally, denoting by  $\mathbf{Z}_l^{(k)}$  the output of the  $l$ -th layer of attention, we define

$$\mathbf{Z}_{l+1}^{(k)} = \mathbf{Z}_l^{(k)} + \text{Attn}(\mathbf{Z}_l^{(k)}), \quad l = 1, \dots, L-1.$$

If task-specific heads  $\mathbf{h}_k, k \in [K]$  (see Section 5) are used, the prediction for  $\mathbf{Z}^{(k)}$  is :

$$\hat{y} = \mathbf{e}_{n+1}^\top \mathbf{Z}_L^{(k)} \mathbf{h}_k,$$

otherwise,

$$\hat{y} = \mathbf{e}_{n+1}^\top \mathbf{Z}_L^{(k)} \mathbf{e}_{d+1}.$$

### E.2 Experiments

We only conduct empirical study to explore the impact of **linear attention model depth** and **label noise** in this section. Similarly, We conduct experiments on synthetic datasets to explore the behavior of multi-layer linear attention models with various trainable parameters under a joint training setting.

**Experimental Setting.** For each context length  $n$ , an independent model is trained for 20,000 iterations with a batch size of 8192 using the Adam optimizer (learning rate  $10^{-3}$ ). To ensure robustness, each training process is repeated 50 times with independent initializations, and the minimal test risk among these trials is reported. All the results are normalized by  $\mathbb{E}[||y||^2]$ .

We use a same synthetic dataset configuration across all the multi-layer attention model experiments:

$$\begin{aligned} \sigma^2 &= 5 \text{ (Noisy), or } \sigma^2 = 0 \text{ (Noise-free)} \\ d &= 10, \quad K = 2, \quad \Sigma_{\text{non-iso}} = \text{diag}\{0.9, 0.9^{-1}, \dots, 0.9^{-9}\} \\ \mathbf{M}_\mu &= [1.7 \cdot \mathbf{1}_{10} \quad -1.3 \cdot \mathbf{1}_{10}], \\ \Sigma_{\beta_1} &= \frac{1}{2} \Sigma_{\beta_2} = \Sigma_{\text{non-iso}}, \quad \pi_1 = 0.3, \quad \pi_2 = 0.7 \end{aligned}$$

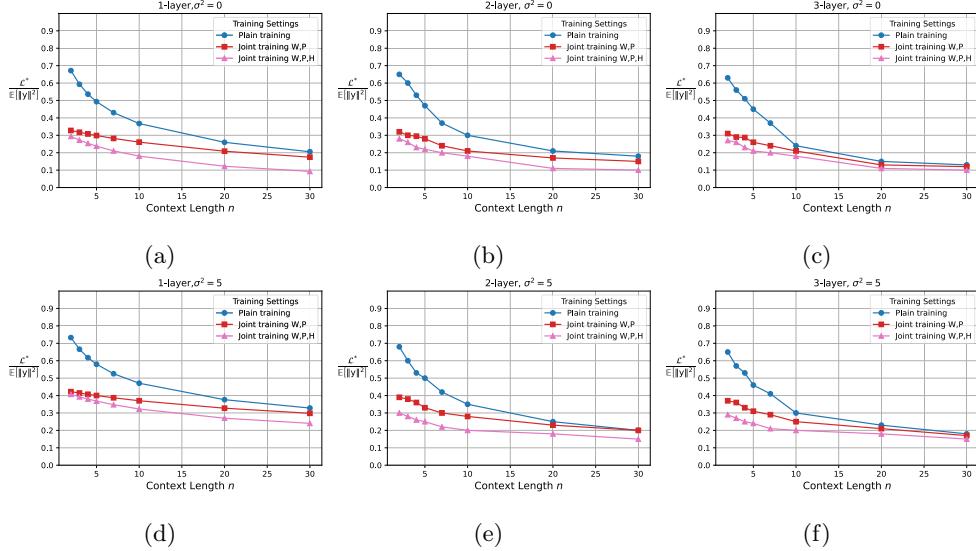


Figure A2: Performance of  $L$ -layer linear attention models ( $L = 1, 2, 3$ ) on (a-c) clean and (d-f) noisy datasets.

As seen in Figure A2: (1) For a clean dataset, task-specific parameters significantly improve performance in the few-shot context region (where the context length  $n < d$ ), but this benefit diminishes as  $n$  increases. (2) Task-specific parameters help mitigate the impact of label noise. (3) Although increasing the depth of the attention model can narrow the performance gaps between different numbers of trainable task-specific parameters, adding task-specific parameters remains beneficial.