Privacy-preserving Prompt Personalization in Federated Learning for Multimodal Large Language Models

Sizai Hou
shouac@connect.ust.hk
Hong Kong University of Science and
Technology
Hong Kong, China

Songze Li
songzeli@seu.edu.cn
Southeast University
Nanjing, China

Baturalp Buyukates
b.buyukates@bham.ac.uk
University of Birmingham
Birmingham, United Kingdom

Abstract

Prompt learning is a crucial technique for adapting pre-trained multimodal language models (MLLMs) to user tasks. Federated prompt personalization (FPP) is further developed to address data heterogeneity and local overfitting, however, it exposes personalized prompts – valuable intellectual assets – to privacy risks like prompt stealing or membership inference attacks. Widely-adopted techniques like differential privacy add noise to prompts, whereas degrading personalization performance. We propose SecFPP, a secure FPP protocol harmonizing generalization, personalization, and privacy guarantees. SecFPP employs hierarchical prompt adaptation with domain-level and class-level components to handle multigranular data imbalance. For privacy, it uses a novel secret-sharing-based adaptive clustering algorithm for domain-level adaptation while keeping class-level components private. While theoretically and empirically secure, SecFPP achieves state-of-the-art accuracy under severe heterogeneity in data distribution. Extensive experiments show it significantly outperforms both non-private and privacy-preserving baselines, offering a superior privacy-performance trade-off.

Keywords

Federated Learning, Prompt Personalization, Privacy Protection

1 Introduction

Multimodal large language models (MLLMs) have received significant attention in recent years due to their remarkable generalization capabilities and strong performance in downstream tasks across diverse applications. However, the effectiveness of pre-trained MLLMs in specific downstream tasks is often limited by task-specific data distributions and the need for localized optimization. Prompt tuning has emerged as one of the most effective techniques for improving the performance of pre-trained models [25, 29, 34, 50]. Subsequent works [6, 35, 69, 70] extended prompt tuning to vision-language models (VLMs) by introducing learnable prompt mechanisms, allowing for more flexible and data-efficient adaptation to downstream tasks. As a lightweight adaptation strategy, prompt learning enables efficient and effective customization of pre-trained models for user-specific downstream tasks. To mitigate local data overfitting and preserve data privacy, researchers have integrated prompt learning into federated learning (FL) frameworks [22, 68, 68]. This integration allows users to collaboratively train prompt parameters while benefiting from the global data distribution. Later works progressively developed the concept of *federated prompt personalization (FPP)*, which has emerged as a promising approach for adapting pre-trained models to individual user tasks through various personalization techniques [8, 21, 31, 32, 63].

As in conventional FL, merely keeping data on local devices does not guarantee privacy in FPP. Numerous studies have demonstrated that gradient information leakage in FL can lead to successful privacy attacks (e.g., [10, 14, 18, 20, 33, 41, 56, 64, 66, 67, 71]), and these threats are equally applicable to FPP. To make matters worse, learned prompts usually represent high-value assets, as they include both task-specific knowledge and potentially sensitive user information [17, 48, 60]. As in FL (e.g., [1, 3, 11, 15, 16, 49, 58, 62]), the most prevalent countermeasure against such attacks in FPP is the application of differential privacy (DP). Consequently, recent FPP studies have also adopted DP as a standard privacy-preserving solution [21, 55]. Unfortunately, unlike full machine learning models, the lightweight nature of prompts makes them especially vulnerable to even minor perturbations introduced by DP noise. Under stringent privacy constraints, this sensitivity often results in significant performance degradation. As empirically demonstrated by [55], DP noise can lead to performance drops of up to 25% under a strict privacy budget, even under low-rank adaption techniques designed to mitigate this effect.

To address the fundamental trade-off between prompt personalization performance and data privacy guarantees, we propose SecFPP, a novel FPP protocol that achieves strong privacy protection without sacrificing personalization performance. Our protocol leverages a secret-sharing primitive, *Lagrange coded computation (LCC)* [65], and introduces a privacy-preserving prompt clustering mechanism, SecPC. In SecFPP, each user decomposes their prompt into two components: a local prompt retained on-device, and a global prompt collaboratively learned across users. This decomposition enables granularity-aware prompt adaptation that effectively handles two levels of data heterogeneity. Specifically,

Table 1: Evolution of prompt learning. Hard prompts refer to manually engineered prompts, such as 'a photo of <class>'. Soft prompts, introduced by CoOp [69, 70], are learnable embeddings that were later integrated with the FL framework by PromptFL [22] and Fed-Prompt [68]. Federated prompt personalization (FPP) [8, 21, 32] incorporates personalization techniques to adapt prompts and handle data heterogeneity. DP-FPL [55] further enhances the privacy of FPP by introducing differential privacy.

	Single Dataset	Multi-Domain Dataset	Privacy
Hard Prompt	68.2	61.9	✗
Soft Prompt	91.4	84.2	✗
FPP	91.6	85.5	✗
DP-FPL	77.4	65.7	✓
SecFPP (ours)	91.6	91.2	✓

global prompts capture coarse-grained distribution shifts (e.g., cross-domain heterogeneity) that groups of users may have, while local prompts accommodate fine-grained distribution shifts (e.g., inter-label heterogeneity) unique to individual users.

Our contributions are summarized as follows:

- We propose SecFPP, a privacy-preserving protocol for FPP that achieves user-level adaptation through a hierarchical prompt structure built upon two key components: LCC and SecPC. This hierarchical prompt strategy utilizes a granularity-aware adaptation scheme that balances generalization and personalization across the user federation. The granularity-aware components of the prompt globally adapt to the domain-level heterogeneity while locally accommodating the class-level heterogeneity.

- We develop SecPC, a novel privacy-preserving adaptive clustering algorithm designed for prompt clustering, leveraged to enable effective domain-level adaptation in SecFPP.

- We provide both theoretical and empirical security analyses of SecFPP, demonstrating that the protocol offers strong privacy guarantees with negligible computational and communicational overheads.

- We conduct extensive experiments under diverse data heterogeneity scenarios. Results show that SecFPP achieves state-of-the-art personalization performance, matching or even surpassing existing non-private methods, while significantly outperforming privacy-preserving baselines.

2 Related Works

2.1 Federated Prompt Personalization (FPP)

First introduced for vision-language models by [69, 70], prompt learning is a lightweight and effective method to enhance the generalization capabilities of foundation models [6, 25, 29, 34, 35, 50, 69, 70]. Rather than relying on manually engineered prompts, prompt learning introduces parametric prompts, learnable continuous vectors trained on user data. Compared to computationally intensive full-model fine-tuning, prompt learning offers a more efficient alternative with notable gains in downstream performance as shown

in Table 1. However, prompts trained solely on local data are prone to overfitting, especially in scenarios with limited or skewed user data distributions. To mitigate overfitting, users can employ federated collaborative prompt learning [22, 68], jointly training prompt parameters while retaining data locally. The core challenge in this setting is to balance the foundation model's generalization capacity with user-specific adaptation. Building on the concept of personalization in FL, recent studies have incorporated FL personalization techniques into prompt learning to better navigate this generalization and localization trade-off [8, 12, 21, 22, 31, 32, 63, 68].

Termed *federated prompt personalization (FPP)*, this approach enables efficient and effective local task adaptation for federated users while maintaining low computational and communication overhead. FedPrompt [68] and PromptFL [22] are the first to integrate FL into prompt learning through different FL paradigms. Subsequent works, such as pFedPrompt [21] and pFedPG [63], introduce more practical FPP methods to adapt frozen pre-trained models to local data heterogeneity. pFedPrompt uses a global prompt and an additional texture encoder for personalized attention to improve local task performance. pFedPG exploits a prompt generator at the server to provide personalized prompts for downstream users. Apart from textual prompts, [31] proposes visual prompts that are attached to image inputs to represent local data distribution. [12] integrates FL model personalization with prompt selection techniques to resolve data heterogeneity effectively. FedOTP [32] further improves the global prompt by user consensus knowledge extraction and uses the local prompt for capturing data features in severe data heterogeneity settings, such as label shifts and domain shifts. Addressing a similar challenge, FedPGP [8] adopts a low-rank prompt decomposition and additional contrastive loss to balance personalization and generalization.

2.2 Privacy Preservation in FPP

A personalized prompt is designed to guide a user's downstream task to fully exploit the capabilities of a pre-trained model. As user-specific customization becomes increasingly desirable, personalized prompts for commercially deployed LLMs are now widely recognized as valuable digital assets on various platforms such as OpenAI GPT Store [39], PromptBase [42], SnackPrompt [51]. This growing ecosystem of prompt sharing and reuse introduces significant privacy risks. As high-value assets, customized user prompts for different model architectures have shown vulnerability to various attacks. Notably, prompt stealing attacks [48] and membership inference attacks [60] have demonstrated the potential threats of unauthorized prompt usage, reproduction, or even leakage of proprietary or sensitive user data. In FL paradigms, a series of gradient-based privacy attacks have been demonstrated to successfully threaten user prompt information in FPP [10, 14, 18, 20, 41, 64, 66, 71].

Although FPP has drawn considerable attention in recent research, its security risks remain rather under-explored, with only a few works addressing privacy concerns. PromptFL [22] establishes a connection between differential privacy (DP) in FL and prompt learning. However, it does not effectively adapt DP to the prompt learning task and neglects the need for user-level personalization. A recent work, DP-FPL [55], is the first to systematically propose a
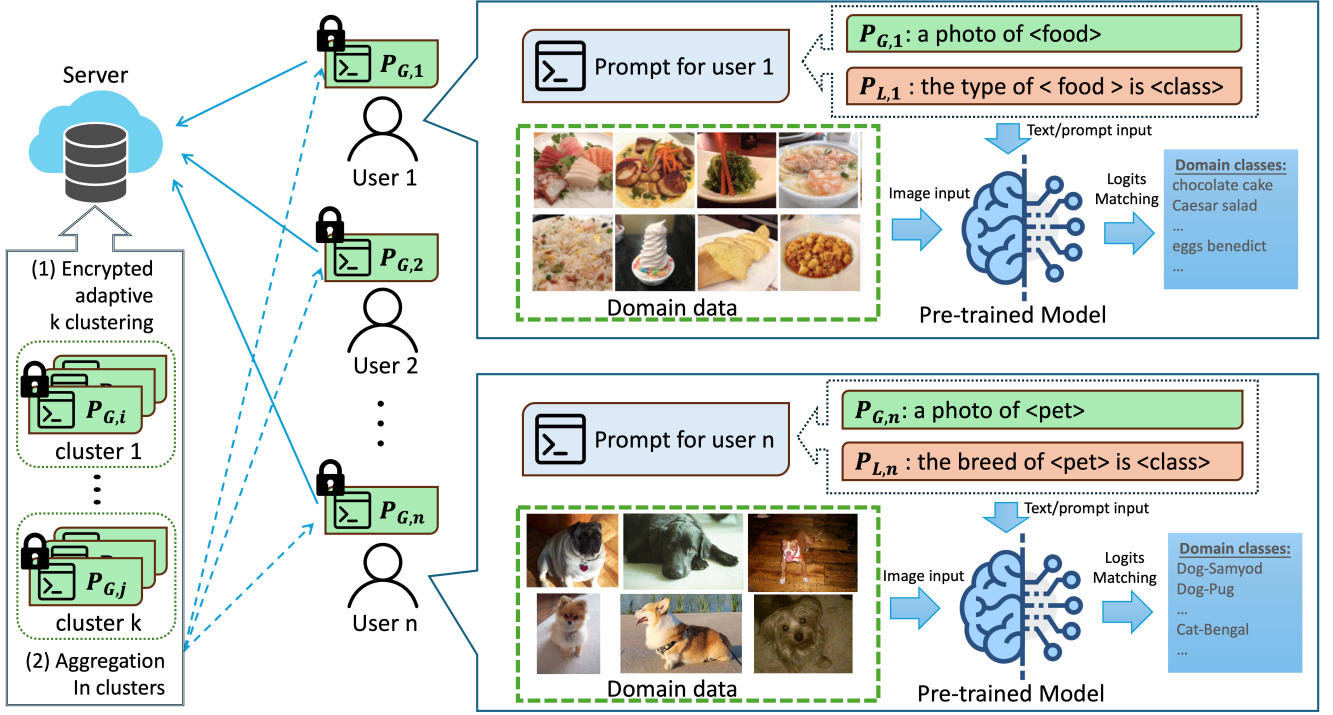
Figure 1: Workflow of SecFPP. On the right, users decompose prompts into global and local components; the global prompt adapts to the dataset domain while the local prompt accommodates the local tasks. On the left, the global prompts are coded by Lagrange coded computation (LCC) and communicated among parties to enable adaptive clustering; then they are aggregated group-wise and distributed to users for the next training round.

DP-based solution in FPP to seek a potential trade-off among personalization, generalization, and privacy. It implements global and local DP to protect prompts and leverages a low-rank adaption strategy to mitigate performance degradation induced by DP. Following prior work [8], DP-FPL factorizes prompts into low-rank components to accommodate different data distributions while applying DP noise to low-rank components during training. Nevertheless, empirical results in Table 1 present significant performance degradation (up to 25% loss in certain cases) when a strict privacy budget is in place (e.g., $\epsilon \leq 0.01$). Additionally, the personalization design of DP-FPL falls short of adapting to domain-level data heterogeneity, as only the local prompt is factorized for personalization while the global prompt is learned universally.

3 Problem Overview

In this section, we present the problem formulation, followed by the threat model and overview of the proposed protocol, SecFPP.

3.1 Problem Formulation

Let $F(\cdot)$ denote an upstream pre-trained model. In FPP, orchestrated by a central server, n end users collaboratively train personalized prompts to adapt the pre-trained model to their datasets while fully utilizing the model's generalization capacity without overfitting. Denote user local datasets by $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n\}$, where user i has a distinct dataset $\mathcal{D}_i$. Instead of using pre-defined prompts (e.g., "a photo of <label>"), users train their own prompts as learnable

embeddings (or soft prompts), $\mathbf{P}_i \in \mathbb{R}^{d \times k}$, where d is the dimension of the word embedding and k is the number of tokens. We only keep the learnable parts in prompt and omit the hard prompt tokens with masked label positions for simplicity in formulations.

Consider data heterogeneity caused by different domains and an unbalanced split within a single domain. We denote personalized prompts by $\mathbf{P}_i$ and decompose it into global prompts and local prompts by $\mathbf{P}_i = P_{G,i} + P_{L,i}$. In the early training phase, the global prompt is shared across all users, i.e., $P_{G,i} = P_G$ for all i. During the later training stages, $P_{G,i}$ is trained group-wise according to the clustering assignment and aggregated inside the designated cluster (see Section 4.2 for details). As a downstream task, each user i performs classification as $\hat{y} = F(x, \mathbf{P}_i)$, for $\forall (x, y) \in \mathcal{D}_i$. Hence, the optimization objective for each user is given by:

$$\mathcal{L}_{\mathcal{D}_i} = \sum_{(x,y) \in \mathcal{D}_i} \ell_{ce}(y, F(x, \mathbf{P}_i)), \tag{1}$$

where ℓ_{ce} stands for the cross entropy loss of the prediction $\hat{y}$ and y. The overall optimization problem for user prompts is:

$$\arg\min_{\{\mathbf{P}_i | i \in [n]\}} \sum_{i \in [n]} \frac{1}{|\mathcal{D}_i|} \mathcal{L}_{\mathcal{D}_i}, \tag{2}$$

where $[n]$ denotes the user index set $\{1, \ldots, n\}$ and $|\cdot|$ denotes the set cardinality.

3.2 Threat Model

The server and the users are *honest-but-curious*, which is commonly used in evaluating the privacy of FL protocols [3, 5, 20, 54, 55]. That is, all parties in the system follow the prescribed protocol faithfully, but the curious server and colluding users attempt to reveal private information including private prompts or membership, from both their local states and received messages during the protocol execution. We further assume that the server does not collude with the clients, as commonly considered in the secure federated learning literature [3, 47, 52].

3.3 Solution Overview

Prior works in FPP mostly focus on personalization performance but overlook the security of the problem. Existing differentially private FPP works heavily suffer from model degradation and poor domain adaptation. To address the critical challenges in performance degradation and privacy preservation in FPP, we propose a secure federated prompt personalization protocol, SecFPP, that integrates a privacy-preserving adaptive clustering algorithm and hierarchical prompt personalization scheme. In the ensuing sections, we first introduce the cryptographic building blocks including the coding primitive, LCC, and the privacy-preserving adaptive clustering algorithm, SecPC; then present the SecFPP workflow in detail; and at last, provide the theoretical analysis for the security of the protocol. The overview of SecFPP is presented in Figure 1.

4 Proposed Secure FPP: SecFPP

This section presents the building blocks, workflow, and theoretical analysis of the proposed SecFPP scheme.

4.1 Cryptographic Building Blocks

SecFPP involves an innovative privacy-preserving clustering algorithm, SecPC. Both clustering and aggregation are built upon the coding primitive LCC.

LCC. The key preliminary to achieve secure clustering is the secret-sharing scheme named Lagrange coded computing (LCC) [65]. LCC is a multi-secret sharing primitive, which generates shares of ℓ secrets $X_1, \ldots, X_\ell$ using Lagrange polynomial interpolation, where the shares are denoted by $[X_1, \ldots, X_\ell]_j$ for any shareholder j. LCC supports computations of $f(X_1), \ldots, f(X_\ell)$, for any multi-variate polynomial $f(\cdot)$. The decryption of LCC is robust to missing and erroneous computation results with Reed-Solomon decoding. Compared with traditional Shamir secret sharing [46] that encrypts one secret at a time, LCC reduces the share size and, thus, the load of computation on each party by ℓ times. Finally, we denote the usages of the sharing and reconstruction algorithms by $LCC.share(\cdot)$, $LCC.recon(\cdot)$, respectively.

SecPC. We propose a secure adaptive prompt clustering algorithm presented in Algorithm 1, SecPC, which is developed upon the well-known k-means clustering algorithm and its adaptive variants [2, 9, 36, 61]. Since the prompt clustering is agnostic to the overall data distribution, the number of clusters is unknown initially. We build upon a simple yet effective adaptive k-means algorithm [2] to perform clustering on user prompts federatively. During this clustering, only the relative distances to the cluster centers are communicated and revealed to the server, for which we present the

---

Algorithm 1: SecPC: Secure Adaptive Clustering

Data: Reduced local prompts $\overline{\mathbf{P}}_i$ of n users, previous round cluster assignment $\mathcal{S}$.
Output: Updated k-cluster assignment $\mathcal{S} : \{s_1, \ldots, s_k\}$.
1 Server broadcasts previous clustering $\mathcal{S}$;
2 *Phase 1: Secure secret sharing*;
3 **for** *user $i \in [n]$ in parallel* **do**
4   Slice its reduced local prompt $\overline{\mathbf{P}}_i$ by length ℓ and generate secret shares by LCC sharing scheme, $\left\{ \left[\overline{\mathbf{P}}_i \right]_1, \ldots, \left[\overline{\mathbf{P}}_i \right]_n \right\} \leftarrow LCC.Share(\overline{\mathbf{P}}_i)$;
5   Share $\left[\overline{\mathbf{P}}_i \right]_j$ with user $j \in [n]$;
6 **end**
7 *Phase 2: Homomorphic distance update*;
8 **for** *user $j \in [n]$ in parallel* **do**
9   Update local coded center, $[\mu_{\mathbf{s}}]_j \leftarrow \sum_{i \in s} \left[\overline{\mathbf{P}}_i \right]_j, \forall s \in \mathcal{S}$;
10   Compute coded distances for all users to each center $[d_{i,s}]_j \leftarrow \left\| [\mu_{\mathbf{s}}]_j - |s| \cdot \left[\overline{\mathbf{P}}_i \right]_j \right\|_2^2, \quad \forall(i,s) \in [n] \times \mathcal{S}$;
11   Transmit all coded distances to the server;
12 **end**
13 *Phase 3: Cluster over revealed distances*;
14 Server reconstructs each $d_{i,s}$ by Reed-Solomon decoding, $d_{i,s} \leftarrow LCC.recon(\{[d_{i,s}]_j \mid j \in [n]\}), \forall(i,s) \in [n] \times \mathcal{S}$;
15 Server recovers the real distances by $d_{i,s} \leftarrow \frac{d_{i,s}}{|s|^2}, \forall(i,s) \in [n] \times \mathcal{S}$;
16 Server performs one-shot adaptive clustering and updates cluster assignment $\mathcal{S}$.

---

security analysis in Section 4.3. Different from [2], we do not update the centers whenever a data point is assigned. Instead, SecPC only performs cluster merging and center updating once at the end of each communication round. In this way, the clustering iterations correspond to the FL communication rounds (which can be considered as a one-shot clustering).

Secure Aggregation. SecFPP involves an aggregation step that group-wise aggregates global prompt components according to the cluster assignments $\mathcal{S}$. We employ the existing off-the-shelf secure aggregation protocols based on the same primitive of LCC, such as [5, 23, 24, 54].

4.2 SecFPP Workflow

In the proposed SecFPP scheme, we introduce an effective federated prompt personalization protocol that provides privacy guarantees while addressing multi-level data heterogeneity, presented in Figure 1. In prior FPP works, various approaches are proposed to balance generality and local adaptation [8, 32, 55], but these methods all adopt a split structure of personalized prompt comprising a global prompt (shared across clients) and a local prompt (customized for individual users). In practice, data heterogeneity may come from diverse sources. It may arise from pathological class distributions

within a single dataset but also may fundamentally come from distinct types of datasets. Motivated by such split structure, we propose to use the splitting decomposition to achieve prompt adaptation for multi-level data heterogeneity: a global prompt component is dynamically adjusted to domain-level heterogeneity, and a local prompt component adapts local heterogeneity while remaining private on users to ensure privacy. To provide privacy preservation to domain-level adaption, we develop SecPC to achieve a simple yet effective approach to adaptively cluster the coarse-grained prompt adaptation of heterogeneity.

At the beginning of training, the server initializes a universal prompt with global and local components for all users. In each communication round, each user computes the gradients for each component using loss $\mathcal{L}_{\mathcal{D}_i}$ measured on the local dataset $\mathcal{D}_i$. After finishing local training epochs, each user updates its local prompt. After updating the personalized prompt $\mathbf{P}_i$ by the local prompt, it performs dimensional reduction on the personalized prompt using a pre-defined method, such as the truncated SVD or PCA [59]. The dimensional reduction preserves the most important components of the prompt while greatly reducing the transmitted data. The adaptive clustering, SecPC, is then performed federatively on the reduced $\overline{\mathbf{P}}_i$ and results in a cluster assignment, denoted by $\mathcal{S}$. According to the cluster assignment, the server invokes secure aggregation to group-wise aggregate the gradients for global components and back-propagates for the domain-level prompts. Finally, the global prompt component is updated into the personalized prompt for the next round or the downstream tasks. We summarize the protocol of SecFPP in Algorithm 2. The detailed privacy-preserving clustering algorithm and aggregation schemes are explained in Sec.4.1.

4.3 Theoretical Analysis

As shown in Algorithms 1 and 2, $P_{L,i}$ is always privately kept by users while $\overline{\mathbf{P}}_i, P_{G,s}$ join the secret-sharing based algorithms. LCC preserves perfect secrecy for the coded information within the security threshold of shareholders. Hence, we are interested in the theoretical analysis for the revealed information in the protocol, specifically, the reconstructed distances $d_{i,s}$, cluster assignment $\mathcal{S}$ and global prompts. Since cluster assignment is non-parametric and global prompts are considered public, we focus on the analysis of the reconstructed distances revealed to the honest-but-curious server.

We assume that the prompt vector of each user has independent and identically distributed (i.i.d.) entries and satisfies a normal distribution of $\mathcal{N}(\mu_i, \sigma_i)$. We denote the vector space by $\mathcal{P}$. Without loss of generality, we analyze a single user prompt $\overline{\mathbf{P}}_i$ against a cluster of user prompts denoted by $\{\overline{\mathbf{P}}_1, \ldots, \overline{\mathbf{P}}_n\}$, such that $\overline{\mathbf{P}}_j \sim \mathcal{N}^d(\mu_j, \sigma_j), \forall j \in [n]$, where i may or may not in $[n]$. Prompts are independently sampled. We denote the average over the cluster by $\overline{\mathbf{P}}_{avg}$, which is the cluster center. To quantify the information leakage by the distance to the center, we analyze the mutual information between a prompt and the distance in the following theorem.

Theorem 1. *Given a cluster of prompts as normal random vectors by $\overline{\mathbf{P}}_j \sim \mathcal{N}^d(\mu_j, \sigma_j), j \in [n]$, the distance is the ℓ_2-norm between a prompt $\overline{\mathbf{P}}_i$ and the cluster center $\overline{\mathbf{P}}_{avg}$, i.e., $D^2 = \left\| \overline{\mathbf{P}}_i - \overline{\mathbf{P}}_{avg} \right\|_2^2$. The*

Algorithm 2: SecFPP: Secure Fed-Prompt Personalization

Data: Local datasets $\{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$ distributed on n users
Input: Communication round T, learning rate η, decomposition rank k.
Output: Personalized prompts $\mathbf{P}_i$ for users

1 Initialize cluster assignment $\mathcal{S} = \{[n]\}$;
2 Initialize personalized prompt parameters
$\mathbf{P}_i^{(0)} = P_{G,s}^{(0)} + P_{L,i}^{(0)}, s \in \mathcal{S}$;
3 **for** *iteration* $t \leftarrow 0$ **to** $T-1$ **do**
4   **for** *user* $i \in [n]$ *in parallel* **do**
5   Compute loss by Eq.1 and calculate gradients as $\nabla_{G,i}\mathcal{L}$ and $\nabla_{L,i}\mathcal{L}$ correspondingly;
6   Update local prompt by $p_{L,i}^{(t+1)} \leftarrow p_{L,i}^{(t)} - \eta\nabla_{L,i}\mathcal{L}$;
7   Use truncated SVD to reduce dimension for the personalized prompt $\overline{\mathbf{P}}_i \leftarrow PCA(\mathbf{P}_i^{(t+1)}, k)$;
8   **end**
9   Perform privacy-preserving adaptive k-means SecPC on reduced personalized prompts, $\mathcal{S} \leftarrow SecPC(\mathcal{S}, \{\overline{\mathbf{P}}_i \mid i \in [n]\})$;
10   Use secure aggregation for global prompt gradients according to cluster assignment, equivalently as, $\nabla_{G,s}\mathcal{L} \leftarrow \frac{1}{|s|} \sum_{i \in s} \nabla_{G,i}\mathcal{L}, \quad \forall s \in \mathcal{S}$;
11   Update clustered global prompts by $P_{G,s}^{(t+1)} \leftarrow P_{G,s}^{(t)} - \eta\nabla_{G,s}\mathcal{L}, \quad \forall s \in \mathcal{S}$;
12   Users update personalized prompts by $\mathbf{P}_i^{(t+1)} \leftarrow P_{G,s|i \in s}^{(t+1)} + P_{L,i}^{(t+1)}, \quad \forall i \in [n]$.
13 **end**

---

mutual information between $\overline{\mathbf{P}}_i$ and D^2 is given by:

$$MI\left(\overline{\mathbf{P}}_i; D^2\right) = \log 2\Gamma\left(\frac{d}{2}\right) + \left(1 - \frac{d}{2}\right)\psi\left(\frac{d}{2}\right) + \frac{d}{2}$$

$$+ \int \cdots \int_{\mathbf{P} \in \mathcal{P}} f_{\overline{\mathbf{P}}_i}\left(\overline{\mathbf{P}}_i = \mathbf{p}\right) \cdot \begin{cases} (\ln(2) + h_d\left(\frac{\tau}{2}\right) + c) \cdot d\mathbf{P} & if\, d \in \mathbb{N}^{odd}, \\ (\ln(2) + g_{d/2}\left(\frac{\tau}{2}\right) + c) \cdot d\mathbf{P} & if\, d \in \mathbb{N}^{even}. \end{cases}$$

$$(3)$$

where Γ, ψ represent gamma function and digamma function, respectively; $f_{\overline{\mathbf{P}}_i}$ is the probability density function of $\overline{\mathbf{P}}_i$; c is $2\log\left(\frac{n-1}{n}\right)$; h_n and $g_{n/2}$ are two families of functions expanded in Appendix B.

PROOF. (Part I) We are given reduced prompts $\{\overline{\mathbf{P}}_1, \ldots, \overline{\mathbf{P}}_n\}$ as d-dimensional continuous random vectors in a cluster, with sample mean $\overline{\mathbf{P}}_{avg} = \frac{1}{n}\sum_{j=1}^n \overline{\mathbf{P}}_j$. As all prompts satisfy $\overline{\mathbf{P}}_j \sim \mathcal{N}^d(\mu_j, \sigma_j)$, their average is also normally distributed, $\overline{\mathbf{P}}_{avg} \sim \mathcal{N}^d(\mu_{avg}, \sigma_{avg})$, where $\mu_{avg}, \sigma_{avg}^2$ are linear combinations of prompts' μ and σ^2. For a prompt $\overline{\mathbf{P}}_i$ inside or outside the given cluster, the distance between the prompt and the cluster center is $D^2 = \left\| \overline{\mathbf{P}}_i - \overline{\mathbf{P}}_{avg} \right\|_2^2$. We discuss the inside case in this part of the proof and the outside case in Appendix B as Part II of proof.

Expand the mutual information in the form of differential entropy:

$$MI\left(\overline{\mathbf{P}}_i; D^2\right) = h\left(D^2\right) - h\left(D^2 \mid \overline{\mathbf{P}}_i\right). \tag{4}$$

For the entropy $h(D^2)$ we make the following observation: D^2 is simply a χ^2-distribution with d degrees of freedom, since it is the summation of the squares of all entries with centered normal distribution. Then:

$$h(D^2) = \log 2\Gamma\left(\frac{d}{2}\right) + \left(1 - \frac{d}{2}\right)\psi\left(\frac{d}{2}\right) + \frac{d}{2}, \tag{5}$$

where Γ and ψ represent gamma function and digamma function, respectively. For the conditional entropy:

$$h\left(D^2 \mid \overline{\mathbf{P}}_i\right) = h\left(\left\|\overline{\mathbf{P}}_i - \overline{\mathbf{P}}_{avg}\right\|_2^2 \mid \overline{\mathbf{P}}_i\right), \tag{6}$$

when $\overline{\mathbf{P}}_{avg}$ contains a term of $\overline{\mathbf{P}}_i$, by isolating it, we have:

$$h\left(D^2 \mid \overline{\mathbf{P}}_i\right) = h\left(\left\|\frac{n-1}{n}\overline{\mathbf{P}}_i - \frac{1}{n}\sum_{j=1, j\neq i}^{n} \overline{\mathbf{P}}_j\right\|_2^2 \mid \overline{\mathbf{P}}_i\right)$$

$$= h\left(\left\|\overline{\mathbf{P}}_i - \frac{1}{n-1}\sum_{j=1, j\neq i}^{n} \overline{\mathbf{P}}_j\right\|_2^2 \mid \overline{\mathbf{P}}_i\right) + 2\log\left(\frac{n-1}{n}\right). \tag{7}$$

Notice $\overline{\mathbf{P}}_i$ and the normally distributed average, $\frac{1}{n-1}\sum_{j=1, j\neq i}^{n} \overline{\mathbf{P}}_j$, are independent. By taking the reduced prompt $\overline{\mathbf{P}}_i = \mathbf{p}$ as the conditional term, the first term of equation (7) is a non-central χ^2-distribution with d degrees of freedom and the non-centrality parameter τ is related to $\overline{\mathbf{P}}_i$ such that:

$$\tau \triangleq \sum_{k=1}^{d} v_k^2. \tag{8}$$

We denote each entry of $\frac{1}{n-1}\sum_{j=1, j\neq i}^{n} \overline{\mathbf{P}}_j$ as X_k, such that the random variable D^2 is:

$$D^2 \triangleq \sum_{k=1}^{d} (X_k + v_k)^2. \tag{9}$$

Then, D^2 is a non-central χ^2-distribution with d degrees of freedom and non-centrality parameter τ. Apply the Theorem 1 in [37] using two families of function $g_m(\cdot), h_n(\cdot)$. We can reach the close form expression by:

$$h\left(\left\|\frac{1}{n-1}\sum_{j=1, j\neq i}^{n} \overline{\mathbf{P}}_j - \mathbf{p}\right\|_2^2 \mid \overline{\mathbf{P}}_i = \mathbf{p}\right) = f_{\mathbf{P}}\left(\overline{\mathbf{P}}_i = \mathbf{p}\right)\mathbb{E}[-\log f(D^2)]$$

$$= -f_{\mathbf{P}}\left(\overline{\mathbf{P}}_i = \mathbf{p}\right) \cdot \begin{cases} \ln(2) + h_d\left(\frac{\tau}{2}\right) + c & \text{if } d \in \mathbb{N}^{\text{odd}}, \\ \ln(2) + g_{d/2}\left(\frac{\tau}{2}\right) + c & \text{if } d \in \mathbb{N}^{\text{even}}. \end{cases} \tag{10}$$

Hence, the overall conditional entropy is:

$$h\left(\left\|\frac{1}{n-1}\sum_{j=1, j\neq i}^{n} \overline{\mathbf{P}}_j - \overline{\mathbf{P}}_i\right\|_2^2 \mid \overline{\mathbf{P}}_i\right) = \int \cdots \int_{\mathbf{p}\in\mathcal{P}} -f_{\overline{\mathbf{P}}_i}\left(\overline{\mathbf{P}}_i = \mathbf{p}\right)$$

$$\cdot \begin{cases} \left(\ln(2) + h_d\left(\frac{\tau}{2}\right) + c\right) \cdot d\mathbf{P} & \text{if } d \in \mathbb{N}^{\text{odd}}, \\ \left(\ln(2) + g_{d/2}\left(\frac{\tau}{2}\right) + c\right) \cdot d\mathbf{P} & \text{if } d \in \mathbb{N}^{\text{even}}. \end{cases} \tag{11}$$

$f_{\overline{\mathbf{P}}_i}(\cdot)$ is the probability density function of $\overline{\mathbf{P}}_i$, i.e. a normal distribution; constant c is $2\log\left(\frac{n-1}{n}\right)$. Combining two terms of $h\left(D^2\right)$ and $h\left(D^2 \mid \overline{\mathbf{P}}_i\right)$, we have equation (3).

$\square$

Remark 1: For rare edge cases, there is an infinitesimal possibility that one could reconstruct a user's entire prompt solely from distance measurements. However, when the prompts are randomly distributed, such reconstruction becomes statistically infeasible. Our focus is on typical scenarios and we aim to statistically answer the following question: *To what extent can an honest-but-curious server infer information about a user's prompt from reconstructed distances?* In Theorem 1, we provide an analytical formulation that characterizes the mutual information between a user's reduced prompt representation $\overline{\mathbf{P}}_i$ and the squared distance D^2, which quantifies the information leakage from the distance in a rigorous, statistical sense. Although deriving a tight upper bound on this mutual information is intractable, the expression serves as a theoretical foundation for analyzing privacy leakage. For instance, if $MI\left(\overline{\mathbf{P}}_i; D^2\right) << h\left(\overline{\mathbf{P}}_i\right)$, the distance provides negligible information about the prompt, i.e., observing D^2 reduces only an infinitesimal amount of uncertainty in the prompt. In such cases, the system satisfies information-theoretic privacy guarantees. To demonstrate that, we present empirical mutual information estimations in Section 5.3.

Remark 2: Equation (3) has two main variables: d and n. $MI(\overline{\mathbf{P}}_i; D^2)$ is dominated by the prompt dimension d as the degrees of freedom. Though n is a variable in the integral of constant, $\log\frac{n-1}{n}$ approaches zero when n is large. The function family h_n and g_m exhibits logarithmic characteristics with strictly increasing monotonicity and is increasingly monotonic in d. In Section 5.3, we numerically demonstrate these characteristics.

5 Experiments

In this section, we evaluate the FPP performance of SecFPP and compare it with private and non-private baselines under different levels of data heterogeneity. Moreover, we present empirical results for the mutual information estimations in the security analysis in Section 4.3. Finally, we perform an evaluation of the computational overheads to demonstrate the cost of privacy in SecFPP.

5.1 Experimental Settings

Following previous works [8, 21, 32, 55], we perform the prompt-based image classification tasks on the pre-trained CLIP model [43] using ViT-B/16 as backbone [13].

Datasets. We consider various datasets from different domains to evaluate the FPP tasks. We use general-domain datasets: CIFAR-10, CIFAR-100 [28] and Caltech-101 [30]; along with specific-domain datasets: Oxford-Pet [40], Oxford-Flowers [38], Food-101 [4], and a texture database DTD [7]. To comprehensively simulate data heterogeneity, we consider multi-granular heterogeneity for data distribution. For single dataset allocation, we split the dataset evenly across users by Dirichlet distribution. Then, we allocate two domains of datasets to each half of the users. Within each dataset, we also apply Dirichlet distribution as the non-i.i.d. partition. This

Table 2: FPP performance with non-private and privacy-preserving protocols (accuracy in %).

Personalization Methods	Single Datasets						Multi-Domain Datasets							
	CIFAR-10	Caltech-101	Oxford-Pet	Oxford-Flowers	Food-101	DTD	CIFAR-10+Caltech-101	CIFAR-100+Caltech-101	Caltech-101+Oxford-Pet	Caltech-101+Oxford-Flowers	Caltech-101+Food-101	Caltech-101+DTD	Oxford-Pet+Oxford-Flowers	Food-101+DTD
PromptFL	89.0	91.4	82.8	69.8	84.6	42.4	89.1	74.8	84.1	78.3	74.9	65.1	77.2	54.1
FedOTP	**89.6**	**91.6**	86.7	66.4	**85.4**	**44.6**	**89.8**	74.7	85.4	**79.4**	84.1	**68.9**	78.1	**59.3**
FedGPG	88.9	91.1	86.8	70.1	84.7	42.6	88.6	74.0	**91.4**	76.3	75.5	67.8	**81.2**	57.8
DP-FPL w.o. privacy	89.3	90.9	84.9	**70.2**	85.1	41.9	89.2	76.0	85.3	77.6	80.6	67.1	80.8	56.7
DP-FPL w. loose privacy	88.5	85.6	82.2	66.2	82.7	35.5	83.4	70.6	78.5	74.5	72.5	63.0	75.5	53.1
DP-FPL w. default privacy	86.1	77.4	77.6	61.2	72.3	32.6	82.9	62.2	73.1	53.1	66.3	45.4	64.0	49.1
DP-FPL w. strict privacy	82.6	74.7	44.8	27.4	56.5	22.0	71.5	50.8	65.7	56.7	54.9	41.3	35.4	45.4
SecFPP (ours)	**89.4**	**91.6**	86.3	70.6	**85.4**	**44.7**	90.6	77.8	91.2	79.6	87.6	69.8	82.7	59.4

dual-level heterogeneity is denoted as datasetA + datasetB in the following section, e.g., Caltech101+OxfordPets.

Baselines. For non-private baselines, we consider PromptFL [22], FedOTP [32], and FedGPG [8]. PromptFL is a federated version of CoOp [69]. FedOTP and FedGPG are the existing state-of-the-art FPP protocols. Regarding privacy-preserving schemes, DP-enabled PromptFL is not capable of training personalized prompts, resulting in poor performance. Hence, we consider the only existing privacy-preserving FPP scheme, DP-FPL [55], for comparisons.

Implementation details. The default number of users, maximum number of epochs, number of local epochs, and learning rate are 20, 100, 10, and 0.001, respectively. For protocols involving low-rank decomposition, we set the default rank to 8. In SecFPP, we set the first-k principal components to 8 as well. For DP-FPL, we choose DP parameter ϵ from $\{0.0, 0.4, 0.1, 0.01\}$ as no privacy, loose privacy, default privacy, and strict privacy constraints, respectively. We set the Dirichlet distribution parameter to $\beta = 0.3$.

### 5.2 FPP Performance Comparisons

The overall performance is presented in Table 2. In general domain datasets, such as CIFAR-10, Caltech-101, all baselines have marginal differences. In contrast, for single specific-domain datasets, FPP-based methods consistently achieve higher accuracy compared to non-personalized PromptFL. SecFPP also achieves comparably strong accuracy results in all single-domain datasets. In multi-domain datasets, the three non-private FPP approaches demonstrate distinct advantages on different dataset combinations while SecFPP presents consistent advantages in accuracy. When the combined datasets are general (such as CIFAR-10+Caltech-101), all methods demonstrate comparable accuracy levels, although our solution exhibits a marginal but consistent performance advantage. Notably, in scenarios with high domain discrepancy, SecFPP's advantage becomes particularly pronounced. While FedOTP demonstrates

competitive robustness under severe data heterogeneity compared to other FPP approaches, SecFPP consistently surpasses all existing FPP solutions in these challenging scenarios.

On the other hand, for the private FPP baselines, while DP-FPL can achieve comparable performance to non-private solutions without DP noise, its accuracy significantly degrades when DP is applied. Furthermore, under tighter privacy constraints, DP-FPL's performance deteriorates proportionally. In sharp contrast, our SecFPP provides rigorous privacy guarantees without compromising any model performance, while demonstrating superior robustness in personalization for both single-domain and multi-domain heterogeneous data scenarios.

5.3 Empirical Studies for Security

In this subsection, we present numerical analysis for mutual information (MI) following the analytical results in Theorem 1. To evaluate it, we exploit widely used KSG algorithm [27] for estimations. Referring to earlier studies [19, 44, 57], we apply the MI estimation on independently sampled user prompts. Recall in Section 4.3, the main focus is on '*how much information about the prompt an honest-but-curious server can infer from the reconstructed distances?*'. We aim to demonstrate the distance's statistical insignificance with respect to a reduced prompt. Hence, we consider three variables for estimating MI as comparisons: entropy of a user prompt (self MI), $h(\overline{\mathbf{P}}_i)$; MI between a user prompt and cluster center, $MI(\overline{\mathbf{P}}_i; \overline{\mathbf{P}}_{avg})$; MI between a user prompt and the distance, $MI(\overline{\mathbf{P}}_i; D^2)$. For $MI(\overline{\mathbf{P}}_i; \overline{\mathbf{P}}_{avg})$, we consider both cases such that the prompt can be inside or outside the given cluster. As in [19], numerical analysis shows that a sample size over 1000 reduces mean squared residuals to 10^{-3}, ensuring stable estimation. So, we use 1000 as sample size and sample user prompts as i.i.d. random vectors. Specifically, the prompt has 15 tokens and the dimension is 512 by default. The dimension is then reduced to 8 as part of the

(a) MI vs. user number   (b) MI vs. prompt dimension
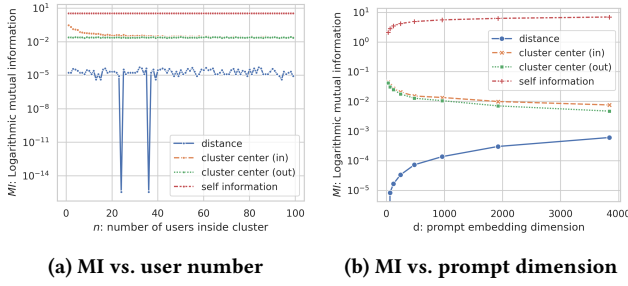
Figure 2: The mutual information comparisons: distance stands for $MI(\overline{\mathbf{P}}_i; D^2)$; cluster center is $MI(\overline{\mathbf{P}}_i; \overline{\mathbf{P}}_{avg})$, which has two cases of the prompt being inside/outside the cluster; self-information is $h(\overline{\mathbf{P}}_i)$. To present the scale, mutual information is plotted in log-scale.

SecFPP, and (μ_i, σ_i)s are simplified to $(0, 1)$. For the distance as a number, we directly use the deterministic scalar-to-vector mapping (replicating D^2 across dimensions) to form a vector for MI estimation, such that the entropy is preserved by the deterministic function. The evaluation results are presented in Figure 2.

As demonstrated by the blue trajectory in Figures 2a and 2b, the prompt dimension d is the predominant factor and exhibits a positive correlation with MI, while the user number n has negligible influence (where n is formally $|\mathcal{S}|$). The empirical results correctly align with the analytical result in Equation (3). Across both evaluations, the mutual information $MI(\overline{\mathbf{P}}_i; D^2)$ between the distance and the prompt (blue) remains exponentially lower than the prompt's entropy (red). Specifically, we have $\log_{10} \frac{h(\overline{\mathbf{P}}_i)}{MI(\overline{\mathbf{P}}_i; D^2)} > r$, with r serving as a lower bound. In our experiments, we observe $r = 4$ such that $MI(\overline{\mathbf{P}}_i; D^2) << h(\overline{\mathbf{P}}_i)$. This substantial gap provides a rigorous information-theoretic constraint on any adversarial reconstruction attempts. Thus, *SecFPP offers strong information-theoretic privacy guarantees for user prompts in practical settings.*

We also present mutual information $MI(\overline{\mathbf{P}}_i; \overline{\mathbf{P}}_{avg})$ between a given prompt and the cluster center for further illustrations. When a prompt resides within a cluster and the cluster cardinality is small, the cluster center exhibits higher statistical dependence on the prompt, inducing larger $MI(\overline{\mathbf{P}}_i; \overline{\mathbf{P}}_{avg})$. The dependency diminishes asymptotically by increasing cluster cardinality $|\mathcal{S}|$ and dimension d. Generally, the observation agrees with the expectation that the cluster center vector conveys more information than the scalar distance while diluting the original prompt entropy by averaging over more prompts. Thus, the fact that SecFPP only reveals relative distances, rather than explicit cluster centers, offers inherently stronger security guarantees, as it significantly limits the server's ability to infer sensitive prompt information.

5.4 Complexity Evaluations

Here, we focus on the additional computation and communication costs of achieving privacy guarantees with SecFPP. All experiments are conducted on a machine using Intel(R) Xeon(R) Gold 5118 CPU @ 2.30GHz, with 12 cores of 48 threads. The computational capability of a user is constrained to one-quarter of the server by parallelization. We denote the LCC parameters by ℓ and the privacy

Table 3: Communication/Computation Overheads of SecFPP

Communication	Computation	
User	User	Server
$O(\frac{nd}{\ell} + kn)$	$O(\frac{d}{\ell} n \log^2 n + \frac{ndk+nd}{\ell})$	$O(kn^2 \log^2 n)$

threshold by $t = \alpha \cdot n$, where α is a constant allowing privacy against $\lfloor \alpha \cdot n \rfloor$ colluding users. The complexity is summarized in Table 3.

(a) time vs. user number   (b) time vs. prompt dimension
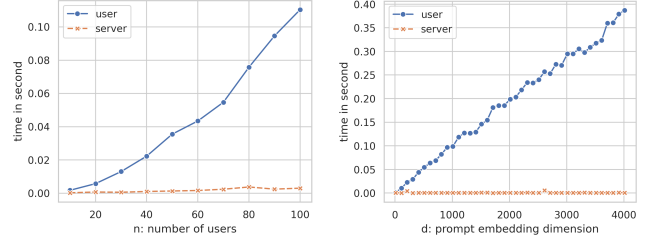
Figure 3: The computational cost in seconds per round.

User communication. The communication overhead of each user consists of two parts: 1) the communication cost of secure secret sharing is $O(\frac{nd}{\ell})$; 2) each client sends $O(kn)$ distance shares to the server. The total communication cost of each client is $O(\frac{nd}{\ell} + kn)$, which is dominated by nd. Here, k is the number of cluster centers.

User computation. The computation performed by each user consists of three parts: 1) utilizing fast polynomial interpolation and evaluation [26], each user generates the secret shares of its local prompt with complexity $O(\frac{d}{\ell} n \log^2 n)$; 2) Within each round, each user first updates the secret shares of k centers, and computes the distances from each data point to each center in the finite domain, taking $O(\frac{ndk}{\ell})$ operations. 3) Secure aggregation by $O(\frac{nd}{\ell})$ The total computation complexity of each client is $O(\frac{d}{\ell} n \log^2 n + \frac{ndk+nd}{\ell})$, which is dominated by $nd \log^2 n$.

Server computation. Server's computation complexity consists of two parts: 1) The decoding of the actual pair-wise distances takes $O(kn^2 \log^2 n)$ operations; 2) The cost of running cluster assignment on each prompt is $O(kn)$. Thus, the total computation complexity of the server over s iterations is $O(kn^2 \log^2 n)$.

Next, we present the empirical overheads of performing SecPC in Figure 3 with dominant factors: n and d. Privacy threshold α is set to $1/3$ consistently and $\ell = \lfloor \frac{n-t}{2} \rfloor$ accordingly. The large-enough prime q is sampled in the scale of 10^{10} and the quantization parameter λ is 10^3 (see Appendix A for quantization details). As shown in Figure 3, the computational cost is almost linearly predominated by nd and the user's overhead is the primary cost. Though the server computation complexity has $n^2 \log^2 n$ term, its overhead is negligible in practice. The user's computation time reaches 0.4s when the dimension is 4000, however, when executing SecFPP, SecPC uses reduced prompts with rank less than 10, multiplying the number of tokens, the total dimension is only $d = 150$. For communication costs, consider a 4G network with 98 Mbps bandwidth and the largest parameter settings in experiments. It takes 0.05s for a user to share and less than 10^{-4}s to communicate distances, which is even faster with networks like LAN or 5G. Overall, the communication

and computation overheads additionally for privacy preservation are negligible.

6 Conclusion

In this paper, we present a novel secure federated prompt personalization protocol, SecFPP, that addresses multi-granular data heterogeneity via the hierarchical prompt adaption design and a privacy-preserving adaptive clustering mechanism, SecPC. Extensive experiments validate SecFPP's robust performance across varying data unbalance, demonstrating consistent and superior performance over SOTA baselines. Moreover, SecFPP theoretically guarantees and empirically validates strong privacy preservation of the user prompts, bridging the critical gap in the performance-privacy trade-off of existing FPP schemes.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 308–318.

[2] Sanjiv K Bhatia et al. 2004. Adaptive K-Means Clustering.. In *FLAIRS*. 695–699.

[3] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 1175–1191.

[4] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101 – Mining Discriminative Components with Random Forests. In *European Conference on Computer Vision*.

[5] Baturalp Buyukates, Jinhyun So, Hessam Mahdavifar, and Salman Avestimehr. 2024. LightVeriFL: A lightweight and verifiable secure aggregation for federated learning. *IEEE Journal on Selected Areas in Information Theory* (2024).

[6] Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. 2023. Unleashing the potential of prompt engineering in large language models: a comprehensive review. *arXiv preprint arXiv:2310.14735* (2023).

[7] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. 2014. Describing Textures in the Wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[8] Tianyu Cui, Hongxia Li, Jingya Wang, and Ye Shi. 2024. Harmonizing Generalization and Personalization in Federated Prompt Learning. In *International Conference on Machine Learning*. PMLR, 9646–9661.

[9] Christian Darken and John Moody. 1990. Fast adaptive k-means clustering: some empirical results. In *1990 IJCNN international joint conference on neural networks*. IEEE, 233–238.

[10] Badhan Chandra Das, M Hadi Amini, and Yanzhao Wu. 2025. Security and privacy challenges of large language models: A survey. *Comput. Surveys* 57, 6 (2025), 1–39.

[11] Lea Demelius, Roman Kern, and Andreas Trügler. 2025. Recent advances of differential privacy in centralized deep learning: A systematic survey. *Comput. Surveys* 57, 6 (2025), 1–28.

[12] Wenlong Deng, Christos Thrampoulidis, and Xiaoxiao Li. 2024. Unlocking the potential of prompt-tuning in bridging generalized and personalized federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6087–6097.

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[14] Jiacheng Du, Jiahui Hu, Zhibo Wang, Peng Sun, Neil Zhenqiang Gong, Kui Ren, and Chun Chen. 2024. SoK: On Gradient Leakage in Federated Learning. *arXiv preprint arXiv:2404.05403* (2024).

[15] Minxin Du, Xiang Yue, Sherman SM Chow, Tianhao Wang, Chenyu Huang, and Huan Sun. 2023. Dp-forward: Fine-tuning and inference on language models with differential privacy in forward pass. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. 2665–2679.

[16] Cynthia Dwork. 2006. Differential privacy. In *International colloquium on automata, languages, and programming*. Springer, 1–12.

[17] Kennedy Edemacu and Xintao Wu. 2024. Privacy preserving prompt engineering: A survey. *Comput. Surveys* (2024).

[18] Xinguo Feng, Zhongkui Ma, Zihan Wang, Eu Joe Chegne, Mengyao Ma, Alsharif Abuadbba, and Guangdong Bai. 2024. Uncovering Gradient Inversion Risks in

[19] Practical Language Model Training. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*. 3525–3539.

[19] Weihao Gao, Sewoong Oh, and Pramod Viswanath. 2018. Demystifying fixed k-nearest neighbor information estimators. *IEEE Transactions on Information Theory* 64, 8 (2018), 5629–5661.

[20] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. 2020. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in neural information processing systems* 33 (2020), 16937–16947.

[21] Tao Guo, Song Guo, and Junxiao Wang. 2023. Pfedprompt: Learning personalized prompt for vision-language models in federated learning. In *Proceedings of the ACM Web Conference 2023*. 1364–1374.

[22] Tao Guo, Song Guo, Junxiao Wang, Xueyang Tang, and Wenchao Xu. 2023. Promptfl: Let federated participants cooperatively learn prompts instead of models–federated learning in age of foundation model. *IEEE Transactions on Mobile Computing* 23, 5 (2023), 5179–5194.

[23] Sizai Hou, Songze Li, Tayyebeh Jahani-Nezhad, and Giuseppe Caire. 2024. PriRoAgg: Achieving Robust Model Aggregation with Minimum Privacy Leakage for Federated Learning. *arXiv preprint arXiv:2407.08954* (2024).

[24] Tayyebeh Jahani-Nezhad, Mohammad Ali Maddah-Ali, Songze Li, and Giuseppe Caire. 2023. SwiftAgg+: Achieving asymptotically optimal communication loads in secure aggregation for federated learning. *IEEE Journal on Selected Areas in Communications* 41, 4 (2023), 977–989.

[25] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. 2022. Visual prompt tuning. In *European conference on computer vision*. Springer, 709–727.

[26] Kiran S. Kedlaya and Christopher Umans. 2011. Fast polynomial factorization and modular composition. *SIAM J. Comput.* 40, 6 (2011), 1767–1802.

[27] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. 2004. Estimating mutual information. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* 69, 6 (2004), 066138.

[28] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[29] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691* (2021).

[30] Fei-Fei Li, Marco Andreeto, Marc'Aurelio Ranzato, and Pietro Perona. 2022. Caltech 101. doi:10.22002/D1.20086

[31] Guanghao Li, Wansen Wu, Yan Sun, Li Shen, Baoyuan Wu, and Dacheng Tao. 2023. Visual prompt based personalized federated learning. *arXiv preprint arXiv:2303.08678* (2023).

[32] Hongxia Li, Wei Huang, Jingya Wang, and Ye Shi. 2024. Global and local prompts cooperation via optimal transport for federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12151–12161.

[33] Zhuohang Li, Jiaxin Zhang, Luyang Liu, and Jian Liu. 2022. Auditing privacy defenses in federated learning via generative gradient leakage. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10132–10142.

[34] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602* (2021).

[35] Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. 2022. Prompt distribution learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5206–5215.

[36] James MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, Vol. 5. University of California press, 281–298.

[37] Stefan M Moser. 2020. Expected logarithm and negative integer moments of a noncentral χ 2-distributed random variable. *Entropy* 22, 9 (2020), 1048.

[38] Maria-Elena Nilsback and Andrew Zisserman. 2008. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*. IEEE, 722–729.

[39] OpenAI. [n. d.]. *OpenAI GPT Store*. https://openai.com/index/introducing-the-gpt-store/

[40] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. 2012. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 3498–3505.

[41] Ivo Petrov, Dimitar I Dimitrov, Maximilian Baader, Mark Müller, and Martin Vechev. 2024. Dager: Exact gradient inversion for large language models. *Advances in Neural Information Processing Systems* 37 (2024), 87801–87830.

[42] PromptBase. [n. d.]. *PromptBase*. https://promptbase.com

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PmLR, 8748–8763.

[44] Brian C Ross. 2014. Mutual information between discrete and continuous data sets. *PloS one* 9, 2 (2014), e87357.

[45] Reent Schlegel, Siddhartha Kumar, Eirik Rosnes, and Alexandre Graell i Amat. 2023. CodedPaddedFL and CodedSecAgg: Straggler mitigation and secure aggregation in federated learning. *IEEE Transactions on Communications* (2023).

[46] Adi Shamir. 1979. How to share a secret. *Commun. ACM* 22, 11 (1979), 612–613.

[47] Jiawei Shao, Yuchang Sun, Songze Li, and Jun Zhang. 2022. Dres-fl: Dropout-resilient secure federated learning for non-iid clients via secret data sharing. *Advances in Neural Information Processing Systems* 35 (2022), 10533–10545.

[48] Xinyue Shen, Yiting Qu, Michael Backes, and Yang Zhang. 2024. Prompt Stealing Attacks Against {Text-to-Image} Generation Models. In *33rd USENIX Security Symposium (USENIX Security 24)*. 5823–5840.

[49] Weiyan Shi, Ryan Shea, Si Chen, Chiyuan Zhang, Ruoxi Jia, and Zhou Yu. 2022. Just fine-tune twice: Selective differential privacy for large language models. *arXiv preprint arXiv:2204.07667* (2022).

[50] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. 2022. Test-time prompt tuning for zero-shot generalization in vision-language models. *Advances in Neural Information Processing Systems* 35 (2022), 14274–14289.

[51] snackprompt. [n. d.]. *snackprompt.* https://snackprompt.com

[52] Jinhyun So, Başak Güler, and A Salman Avestimehr. 2020. Byzantine-resilient secure federated learning. *IEEE Journal on Selected Areas in Communications* 39, 7 (2020), 2168–2181.

[53] Jinhyun So, Basak Guler, and Salman Avestimehr. 2020. A scalable approach for privacy-preserving collaborative machine learning. *Advances in Neural Information Processing Systems* 33 (2020), 8054–8066.

[54] Jinhyun So, Chaoyang He, Chien-Sheng Yang, Songze Li, Qian Yu, Ramy E Ali, Basak Guler, and Salman Avestimehr. 2022. Lightsecagg: a lightweight and versatile design for secure aggregation in federated learning. *Proceedings of Machine Learning and Systems* 4 (2022), 694–720.

[55] Linh Tran, Wei Sun, Stacy Patterson, and Ana Milanova. 2025. Privacy-Preserving Personalized Federated Prompt Learning for Multimodal Large Language Models. *arXiv preprint arXiv:2501.13904* (2025).

[56] Minh Vu, Truc Nguyen, My T Thai, et al. 2024. Analysis of privacy leakage in federated large language models. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 1423–1431.

[57] Binghui Wang, Jiayi Guo, Ang Li, Yiran Chen, and Hai Li. 2021. Privacy-preserving representation learning on graphs: A mutual information perspective. In *Proceedings of the 27th acm sigkdd conference on knowledge discovery & data mining*. 1667–1676.

[58] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. 2020. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE transactions on information forensics and security* 15 (2020), 3454–3469.

[59] Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems* 2, 1-3 (1987), 37–52.

[60] Yixin Wu, Rui Wen, Michael Backes, Pascal Berrang, Mathias Humbert, Yun Shen, and Yang Zhang. 2024. Quantifying privacy risks of prompts in visual prompt learning. In *33rd USENIX Security Symposium (USENIX Security 24)*. 5841–5858.

[61] Shuyin Xia, Daowan Peng, Deyu Meng, Changqing Zhang, Guoyin Wang, Elisabeth Giem, Wei Wei, and Zizhong Chen. 2020. A fast adaptive k-means with no bounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[62] Xingpeng Xiao, Yaomin Zhang, Heyao Chen, Wenkun Ren, Junyi Zhang, and Jian Xu. 2025. A Differential Privacy-Based Mechanism for Preventing Data Leakage in Large Language Model Training. *Academic Journal of Sociology and Management* 3, 2 (2025), 33–42.

[63] Fu-En Yang, Chien-Yi Wang, and Yu-Chiang Frank Wang. 2023. Efficient model personalization in federated learning via client-specific prompt generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 19159–19168.

[64] Haomiao Yang, Mengyu Ge, Dongyun Xue, Kunlan Xiang, Hongwei Li, and Rongxing Lu. 2023. Gradient leakage attacks in federated learning: Research frontiers, taxonomy and future directions. *IEEE Network* (2023).

[65] Qian Yu, Songze Li, Netanel Raviv, Seyed Mohammadreza Mousavi Kalan, Mahdi Soltanolkotabi, and Salman A. Avestimehr. 2019. Lagrange coded computing: Optimal design for resiliency, security, and privacy. In *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 1215–1225.

[66] Xi Sheryl Zhang, Weifan Guan, Jiahao Lu, Zhaopeng Qiu, Jian Cheng, Xian Wu, and Yefeng Zheng. 2024. GraphLeak: Patient Record Leakage through Gradients with Knowledge Graph. In *Proceedings of the ACM Web Conference 2024*. 4706–4716.

[67] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. 2020. idlg: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610* (2020).

[68] Haodong Zhao, Wei Du, Fangqi Li, Peixuan Li, and Gongshen Liu. 2023. Fedprompt: Communication-efficient and privacy-preserving prompt tuning in federated learning. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.

[69] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 16816–16825.

[70] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022. Learning to prompt for vision-language models. *International Journal of Computer Vision* 130, 9 (2022), 2337–2348.

[71] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. *Advances in neural information processing systems* 32 (2019).

Appendix

A Additional Preliminaries

In SecFPP, the operations of data sharing and distance computation are carried out in a finite field $\mathbb{F}_q$, for some large prime q. Hence, for a data point $\mathbf{x}_i$ from the real field, one needs to first quantize it onto $\mathbb{F}_q$. We abuse the symbols only in this section.

Data Quantization. The quantization technique in SecFPP has been utilized in [52] [54]. To quantize a data point $\mathbf{x}_i$, we first scale it by λ, and embed the scaled value onto $\mathbb{F}_q$ such that

$$\bar{\mathbf{x}}_i = Q(\mathbf{x}_i, \lambda) = \begin{cases} \lfloor \lambda \mathbf{x}_i \rfloor, & \text{if } \mathbf{x}_i \geq 0 \\ \lfloor q + \lambda \mathbf{x}_i \rfloor, & \text{if } \mathbf{x}_i < 0 \end{cases}. \tag{12}$$

Here $\lfloor x \rfloor$ denotes the largest integer less than or equal to x, and the quantization function Q is applied element-wise. Assuming each element of $\lambda \mathbf{x}_i$ is within $[-\eta, \eta)$ for some $\eta > 0$, then on the range of Q, $[0, \eta)$ is the image of the positive part, and $[q - \eta, q)$ is the image of the negative part. While λ controls the precision loss of quantization, it also needs to be chosen so that overflow does not occur during computations of SecFPP. A larger η requires a larger finite field size q to avoid computation overflow, and a smaller η leads to a higher precision loss between $\mathbf{x}_i$ and $\bar{\mathbf{x}}_i$. We can choose a proper scaling factor λ to preserve enough precision while keeping the field size practical. To avoid computation overflow, we should choose q such that all the intermediate computation results on the scaled data $\lambda \mathbf{x}_i$ are within the range $\left(-\frac{q}{2}, \frac{q}{2}\right)$. In the worst case, the largest distance across data points and cluster centers $D \triangleq \max\limits_{i \in [m], h \in [k]} \left\| \boldsymbol{\mu}_h - |\mathcal{S}_h| \cdot \mathbf{x}_i \right\|_2^2$ results in the largest output value. Therefore, we should choose q that is at least $2\lambda^2 D$.

Lagrange Coded Computing (LCC). The technique used when each client secret shares its data with the other clients is Lagrange Coded Computing (LCC). As a result, each client gets a hold of the entire dataset in a coded manner, thereby being able to compute the secret shares of the distances between each pair of datapoints and cluster centers. Similar secret-sharing operations between clients have been widely utilized in FL literature to improve performance, privacy, and robustness (see e.g., [47], [45], [52], [3], [53]). In [47], authors tackle the issue of non-iid data distribution of local client datasets, and use LCC-based secret sharing of client data to improve the convergence of FL models. In Reference [45], authors use Shamir's secret sharing to encode and share their local datasets with the other clients. The added redundancy in the encoding process helps mitigate stragglers in FL, without sacrificing local data privacy. In Reference [52], clients exchange secret shares of their local model updates in a verifiable manner during FL iterations, so that each client can make sure that others have sent valid secret shares, which are then utilized by the server to decode the pairwise distances among client models for outlier detection and removal before aggregation. In [3], secret shares of the client models are utilized to handle client dropouts during secure model aggregation. Authors of [53] study the setting of decentralized FL, and utilize secret shares of private datasets among multiple parties to perform distributed training, while still preserving privacy.

B Supplementary to Proof

$g_m(\cdot)$ and $h_n(\cdot)$ are two family of functions introduced in [37]. $g_m(\cdot) : \mathbb{R}^+ \to \mathbb{R}$ with $m \in \mathbb{N}$, which has the following expression:

$$g_m(\xi) \triangleq \begin{cases} \ln(\xi) - \text{Ei}(-\xi) + \sum_{j=1}^{m-1} (-1)' \left[e^{-\xi}(j-1)! - \frac{(m-1)!}{j(m-1-j)!} \right] \left(\frac{1}{\xi} \right)' & \text{if } \xi > 0, \\ \psi(m) & \text{if } \xi = 0. \end{cases} \tag{13}$$

where Ei is the exponential integral and ψ is digamma function. $h_n(\cdot) : \mathbb{R}^+ \to \mathbb{R}$ with $n \in \mathbb{N}^{odd}$, which has the following expression:

$$h_n(\xi) \triangleq \begin{cases} -\gamma - 2\ln(2) + 2\xi \cdot {}_2F_2\left(1, 1; \frac{3}{2}, 2; -\xi\right) \\ + \sum_{j=1}^{\frac{n-1}{2}} (-1)^{j-1} \Gamma\left(j - \frac{1}{2}\right) \cdot \left[\sqrt{\xi} e^{-\xi} \text{erfi}(\sqrt{\xi}) + \sum_{i=1}^{j-1} \frac{(-1)^i \xi^i}{\Gamma\left(i+\frac{1}{2}\right)} \right] \left(\frac{1}{\xi} \right)^j & \text{if } \xi > 0, \\ \psi\left(\frac{n}{2}\right) & \text{if } \xi = 0. \end{cases} \tag{14}$$

where ${}_2F_2$ is a generalized hypergeometric function and erfi is the imaginary error function. For the analysis of the function families, please refer to [37].

Proof. (Part II) In addition to Part I of the proof, we continue to discuss the other case when the prompt is outside of the given cluster. We continue from equation (4). When prompt $\overline{\mathbf{P}}_i$ is outside of the cluster, $h(D^2)$ is non-central χ^2-distribution with d degrees of freedom. The non-centrality parameter τ is defined by μ_i. Hence, similar to equation (8), $\tau_i = d \cdot \mu_i^2$. Then:

$$h(D^2) = \begin{cases} \ln(2) + h_d\left(\frac{\tau_i}{2}\right) & \text{if } d \in \mathbb{N}^{\text{odd}}, \\ \ln(2) + g_{d/2}\left(\frac{\tau_i}{2}\right) & \text{if } d \in \mathbb{N}^{\text{even}}. \end{cases} \tag{15}$$

For the other term of $h\left(D^2 \mid \overline{\mathbf{P}}_i\right)$, different from equation (7), we do not need to isolate $\overline{\mathbf{P}}_i$. Therefore, there is no constant term c in the final expression of the conditional entropy. Combining equation (15) and equation (11) without c yields the counterpart of the theorem. Note that if the prompt distribution has $\mu_i = 0$, equation (15) still reduces to the first part of the proof. □