Probabilistic Federated Prompt-Tuning with Non-IID and Imbalanced Data

Pei-Yau Weng
Washington State University
pei-yau.weng@wsu.edu

Minh Hoang
Princeton University
minhhoang@princeton.edu

Lam M. Nguyen
IBM Research
lamnguyen.mltd@ibm.com

My T. Thai
University of Florida
mythai@cise.uf.edu

Tsui-Wei Weng
University of California San Diego
lweng@ucsd.edu

Trong Nghia Hoang[*]
Washington State University
trongnghia.hoang@wsu.edu

Abstract

Fine-tuning pre-trained models is a popular approach in machine learning for solving complex tasks with moderate data. However, fine-tuning the entire pre-trained model is ineffective in federated data scenarios where local data distributions are diversely skewed. To address this, we explore integrating federated learning with a more effective prompt-tuning method, optimizing for a small set of input prefixes to reprogram the pre-trained model's behavior. Our approach transforms federated learning into a distributed set modeling task, aggregating diverse sets of prompts to globally fine-tune the pre-trained model. We benchmark various baselines based on direct adaptations of existing federated model aggregation techniques and introduce a new probabilistic prompt aggregation method that substantially outperforms these baselines. Our reported results on a variety of computer vision datasets confirm that the proposed method is most effective to combat extreme data heterogeneity in federated learning.

1 Introduction

The proliferation of personal devices has transformed our data landscape into numerous federated systems with distinct resource constraints, data representations, distributions, and ownership. This has motivated the development of new machine learning (ML) paradigms that enable collaborative learning across different systems while respecting their data privacy. A prominent framework to substantiate this collaborative scheme is federated learning (FL), which allows multiple systems to train a common model without sharing their private data [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12].

Existing FL methods often assume that learning begins from scratch and does not build on prior expertise. On the other hand, fine-tuning pre-trained or foundation models [13] is an emerging paradigm for efficient generation of ML solutions. Almost all state-of-the-art models in natural language processing (NLP) are now fine-tuned versions of foundation models, such as BERT [14], BART [15], RoBERTa [16], and T5 [17]. Likewise, many top performing vision models have also benefited from the generalization capability of foundation models, such as the vision transformer model [18], which was trained on large-scale and generic datasets such as ImageNet [19]. Nonetheless, this fine-tuning practice has only recently been investigated in federated learning [20, 21].

Current research in this direction demonstrated various benefits of integrating fine-tuning into FL. For example, fine-tuning can utilize information better in decentralized data scenarios and thus improves the performance in various FL scenarios, mostly in NLP [22, 23, 24, 25]. Interestingly, it has been pointed out that even a simple initialization of local clients with a foundation model can prevent solution drift to some extent in large-scale scenarios with heterogeneous local data distributions [20, 21, 23]. This is also consistently observed in the results of our case studies in Fig. 1.

[*] Corresponding authors: Pei-Yau Weng, Minh Hoang, Trong Nghia Hoang.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

However, in FL environments that depend on frequent communication and synchronization of model updates across multiple devices, fine-tuning the entire pre-trained model is often infeasible due to limited local storage and communication bandwidth. Our study also reveals that full-model fine-tuning approaches fall short when the federated data are not only *heterogeneous* but also *imbalanced*. Our case studies in Fig. 1 in particular show that when the local data are both heterogeneous and imbalance, federated learning with full-model adaptation suffers a huge performance drop, highlighting its instability in extreme data settings.

To address the resource bottleneck, several works in the recent federated learning literature have turned to a new class of parameter-efficient tuning method called prompt tuning. This method focuses on engineering cues, such as extra tokens which are appended to the input embedding in a transformer architecture. Such tokens or prompts provide beneficial context to performing the computational task, similar to how hints can be provided to assist puzzle solving. Local sets of prompts can then be aggregated or personalized using existing federated learning strategies. For example, the prior works of [26] and [27] use FedAvg [2] to aggregate the local prompts while [28] integrates prompt-tuning into personalized federated learning, which helps generate client-specific sets of prompts that are well-customized to the corresponding local data distribution.

Although federated prompt-tuning approaches eliminate the need to update and communicate hundreds of millions of network weights, there is a substantial gap between the performance of these approaches in highly heterogeneous, data-imbalanced settings, and the upper-bound performance of fine-tuning in the centralized data setting, as shown in Fig. 1. This gap can be attributed to the fact that locally learned prompt sets are learned in arbitrary orders, which are generally unaligned across clients. That is, the same prompt position across different clients might encode different contextual information about the local data. A simple aggregation that disregards prompt alignment might attempt to combine prompts from different contexts and collapse into less informative prompts. This has not been explicitly modeled in prior federated prompt-tuning work.

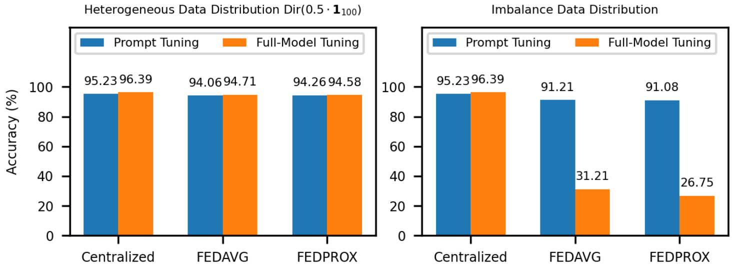

Figure 1: Test Accuracy (%) achieved on the CIFAR-10 dataset by solving Eq. (3) via centralizing data, using FedAvg, and using FedProx on (orange) full-model (FM) and (blue) prompt-tuning (PT) setups. The evaluation is performed under (left) a standard (non-extreme) heterogeneous data partition; and (right) an extremely imbalanced data partitioning scheme (see Section 4).

To address this issue, we adopt a hierarchical probabilistic modeling approach to characterize both the generation and alignment of local prompts in this paper. We cast the server aggregation step as an alignment of local prompt sets. This alignment discovers and aggregates local prompts encoding similar contextual information into summarizing prompts. On the client side, we view each local prompt set as an independent sample drawn from a generative model parameterized by the global summarizing prompts. In this view, each local prompt can be seen as a probabilistic exploration initialized by a randomly selected summarizing prompt. The alignment between prompts can be characterized in terms of their association with the summarizing prompts, which can be inferred via learning the parameters of the above generative model. We summarize our main contributions below:

C1. We formulate the prompt summarizing procedure as a probabilistic set modeling task in which each local set is assumed to be an independent sample of a random point process. The alignment of similar prompts across different local sets can be set as part of the modeling parameterization, whose optimization can be interleaved with the optimization of the point process's parameters (Section 3.2).

C2. We develop an algorithm to find the most probable association between the local and summarizing prompts. This association is viewed as a latent variable in the above generative model. Specifically, we cast its inference as a classical weighted bipartite matching task via an interesting observation that the association between the summarizing prompt and each local set of prompts participate linearly to the loss function of our proposed generative model (Section 3.3).

C3. We compare the performance of our method against various federated prompt-tuning baselines based on existing heterogeneous federated learning techniques to demonstrate its effectiveness. Our reported results on a variety of experiments and baselines demonstrate consistently that our method is most effective to combat data imbalance in extreme heterogeneous scenarios (Section 4).

2 Related Work

Federated learning (FL) [1, 2] is a collaborative framework that allows multiple parties to collaborate on training a common model without sharing their private data. In FL, the training data are distributed across m clients. The t-th client owns a local, private dataset $D_t = \{(\mathbf{x}_{tm}, y_{tm})\}_{m=1}^{n_t}$ comprising n_t data points. The goal is to fit a model $\mathbf{w}_*$ using all private datasets without centralizing them. That is,

$$\mathbf{w}_* \;=\; \arg\min_{\mathbf{w}} \left\{ L(\mathbf{w}) \triangleq \sum_{t=1}^{m} \left(\frac{n_t}{n}\right) \cdot L_t(\mathbf{w}) \right\} \;,\; \text{where} \;\; L_t(\mathbf{w}) \;\triangleq\; \frac{1}{n_t} \sum_{m=1}^{n_t} \ell\big(\mathbf{x}_{tm}, y_{tm}; \mathbf{w}\big) \qquad (1)$$

and $n = n_1 + n_2 + \ldots n_m$ denotes the total number of data points while $\ell(\mathbf{x}_{tm}, y_{tm}; \mathbf{w})$ denotes some loss function of choice. Clients collaborate via sharing their local models instead of data. Each client can run multiple local updates before sharing its local model for aggregation. This helps reduce the number of communication rounds while still preserving the convergence guarantee. [2] names this the FEDAVG algorithm, which iterates between local optimization and global aggregation:

$$\mathbf{w}_t^{(r)} \;=\; U\Big(\mathfrak{L}_t, \mathbf{w}_*^{(r-1)}\Big) \; \forall t \in [m] \;, \quad \mathbf{w}_*^{(r)} \;=\; \sum_{t=1}^{m} \left(\frac{n_t}{n}\right) \cdot \mathbf{w}_t^{(r)} \;. \qquad (2)$$

The local update routine $U(\mathfrak{L}_t, .)$ is typically standard gradient updates such as SGD or Adam. At the beginning of an iteration, the local weight is set to be the global estimate from the previous communication round. In practice, local data distributions tend to diverge which consequently causes the local updates in Eq. (2) to have different convergence points across clients. This is known as the solution drift phenomenon, which often decreases the performance of the federated model. Numerous approaches had been proposed to mitigate this phenomenon, which includes the following directions:

Client Regularization. These methods focus on augmenting the local update strategies to prevent clients from drifting apart. For example, FEDPROX [29] introduces an ℓ_2-regularization term to penalize updates that diverge the local model from the global model. FEDDYN [30] uses a FL-adapted version of [31] on distributed optimization as the regularizer. SCAFFOLD [32] attempts to correct the direction of local gradient updates at each client using their sum of gradients. [33] utilizes the similarity between model representations to correct local training via contrastive learning.

Server Regularization. Another approach to prevent the drifting effect caused by heterogeneous local distributions is to replace the model average in Eq. (2) with a different mechanism of model aggregation. For example, [7, 6] decomposes client models into bags of neurons and performs a non-parametric clustering of neurons. Cluster centroids are used to synthesize an aggregated model for the next communication round. This approach bypasses the drifting phenomenon as the number of cluster is non-parametric and can be adjusted to accommodate new information patterns. In a similar vein, [34] adopts a probabilistic perspective of model aggregation, sampling higher-quality global models and combining them via Bayesian model ensemble, leading to a more robust aggregation. [35] presents a data-free knowledge distillation methods for FL that help to train generators without compromising clients' data. [36] redistributes each client's shared model to others for b consecutive communication iterations, exposing it to various heterogeneous data sources before performing aggregation, curving their solution divergence.

Personalization. Instead of learning a single, universal model, [29, 37, 38, 39, 40] seek to learn personalized models for all clients. [3, 39] formulates this problem as multi-task learning, whereas [37] and [38] respectively ground it in meta-learning and regularization frameworks. Several recent works impose a shared data representation across clients [40, 41, 42, 43, 44], which is optimized using existing FL techniques, and locally fine-tune a small classification head to achieve personalization. However, personalized models trained in this manner tend to only perform well on its corresponding local test set, and not on a comprehensive test set combining all local test data (see Section 4).

Finally, most previous work assume that each FL client has sufficient data to adequately train its local model. This often contradicts real-world situations where data are scarce [45, 36, 46, 47, 48]. While data scarcity is not a new challenge of machine learning, it has not been thoroughly investigated in the context of FL. [36] points out that with limited, scarce data, local models often have bad qualities, and aggregating such models tend to result in poor global performance. Although fine-tuning pre-trained large models is an increasingly popular technique to combat data shortage, most existing FL works (including [36] which aims to address data scarcity) have not tried to leverage this resource. This motivates us to investigate prompt-tuning as a new FL paradigm.

3 Probabilistic Federated Prompt-Tuning

First, Section 3.1 provides a concise background on the standard prompt tuning technique in FL setting. Motivated by the result of our case study (see Fig. 1), Section 3.2 further introduces a probabilistic federated prompt-tuning framework that aims to close up the performance gap between federated prompt-tuning and centralized full-model fine-tuning in data imbalance settings. Our framework characterizes each local model as a random set of prompts distributed by a hierarchical generative model. An effective optimization algorithm to learn these is then detailed in Section 3.3. An overall diagram featuring a bird-view of our framework is also provided in Fig. 2.

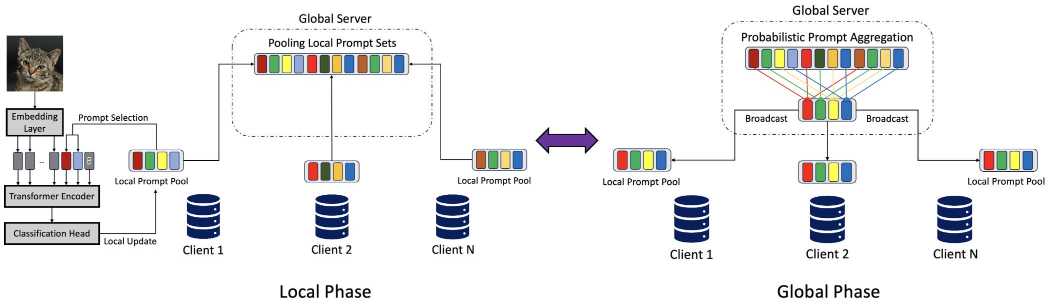

Figure 2: Workflow of Probabilistic Federated Prompt Aggregation: (left) each client selects a subset of prompts from the global set of summarizing prompts using the prompt-selection mechanism adapted from [49], and fine-tune them using local data; and (right) the server collects all local prompt sets and updates the global summarizing prompts that aggregate similar local prompts. This is achieved by our proposed probabilistic federated prompt aggregation (PFPT) algorithm.

3.1 Prompt-Tuning with Pre-Trained Model

We use a pre-trained Vision Transformer [18] in all our experiments. The pre-trained model is a composition $F_c \circ F_a \circ F_e$ where F_c is the classification head, F_a is the stack of attention blocks, and F_e is the input embedding block. Keeping both F_a and F_e frozen, the local solution for each client t can be represented as $F_{c,t} \circ F_a \circ (F_e \cup \boldsymbol{\omega})$ which applies a personalized prediction head on the output of the frozen attention block F_a whose (set) inputs are the union of F_e and a set $\boldsymbol{\omega}$ of reprogramming prompts [49]. $\boldsymbol{\omega}$ is in turn a union of k individual prompts $\boldsymbol{\omega} = \boldsymbol{\omega}_1 \cup \boldsymbol{\omega}_2 \cup \ldots \cup \boldsymbol{\omega}_k$. Each local solution is characterized by a tuple of $(F_{c,t}, \boldsymbol{\omega})$ comprising a personalized head $F_{c,t}$ and common prompt set $\boldsymbol{\omega}$. The FL framework in Eq. (1) is now re-cast as

$$\boldsymbol{\omega}_* = \arg\min_{\boldsymbol{\omega}} \left\{ \sum_{t=1}^{m} \left(\frac{n_t}{n}\right) \cdot L_t(\boldsymbol{\omega}) \right\} \text{ with } L_t(\boldsymbol{\omega}) \triangleq \max_{F_{c,t}} \left\{ \left(\frac{1}{n_t}\right) \cdot \sum_{m=1}^{n_t} \ell\Big(\mathbf{x}_{tm}, y_{tm}; F_{c,t}, \boldsymbol{\omega}\Big) \right\} \quad (3)$$

where $\ell(\mathbf{x}_{tm}, y_{tm}; F_{c,t}, \boldsymbol{\omega}) = \ell(\mathbf{x}_{tm}, y_{tm}; F_{c,t} \circ F_a \circ (F_e \cup \boldsymbol{\omega}))$ which makes explicit the leverage of the existing (pre-trained) expertises (F_a, F_e). The optimal solution $\boldsymbol{\omega}_*$ of Eq. (3) can be approximated using any of the existing federated learning algorithms, such as FEDAVG [2] and FEDPROX [29].

Remark. There is an alternative approach to enabling light-weight federated fine-tuning. Instead of using prompts, approaches in this direction use a (learnable) adapter network that adapts the output of an intermediate (frozen) neural network segment before passing it to the rest of the (frozen) neural network. Local adapters across client can then be aggregated using FEDAVG [21]. This direction is however orthogonal to federated prompt-tuning, which is our main focus here. Further investigation into probabilistic methods for adapter aggregation would be a potential follow-up of our current work.

3.2 Probabilistic Prompt Model

Intuitively, our federated prompt tuning framework is an analog to traditional FL on the space of prompts. Prior to each communication round, we suppose that each local client t, via prompt-tuning, has obtained a set of n_t prompts $\boldsymbol{\omega}_t \triangleq \{\boldsymbol{\omega}_{t1}, \boldsymbol{\omega}_{t2}, \ldots, \boldsymbol{\omega}_{tn_t}\}$. As the participating clients upload their local prompt sets, the server aggregates the combined set and subsequently returns a set of n summary prompts, denoted as $\boldsymbol{\Phi} \triangleq \{\boldsymbol{\phi}_i\}_{i=1}^n$. We will elaborate on this aggregation step in Section 3.2.2. Similar to standard FL, this aggregated set is also distributed to all clients at the beginning of the next communication round. Each client then selects the most relevant prompts from this set for further fine-tuning. This sampling step follows a generative process described in Section 3.2.1.

3.2.1 Generative Model

At the beginning of each communication round, each client constructs a set of n_t prompt initializations given the server-broadcast summary $\boldsymbol{\phi}$. We model this construction by a random generative process. First, each local prompt initialization, $\boldsymbol{\omega}_{tk}$, is modeled as a sample drawn from a Gaussian with learnable neural parameterization,

$$\boldsymbol{\omega}_{tk} \quad \sim \quad \mathbb{N}\Big(\boldsymbol{\psi}_{tk}, \mathrm{diag}\Big(\alpha(\boldsymbol{\psi}_{tk};\gamma)\Big)\Big) \ , \tag{4}$$

where the diagonal covariance matrix are parameterized as the output of a neural net α (with weight γ) on the mean parameter $\boldsymbol{\psi}_{tk}$. The set of mean parameters $\boldsymbol{\psi}_t \triangleq \{\boldsymbol{\psi}_{tk}\}_{k=1}^{n_t}$ is in turn modeled as a random subset of the (server-broadcast) summarizing prompts $\boldsymbol{\phi}$, following a Bernoulli point process prior with a finite mixture as the base measure:

$$Q_t(\boldsymbol{\psi}) \quad \sim \quad \mathrm{BeP}\left(\sum_{i=1}^{n} \sigma\Big(g(\boldsymbol{\phi}_i;\mathbf{w})\Big) \cdot \mathbb{I}\Big(\boldsymbol{\psi} = \boldsymbol{\phi}_i\Big)\right) \ . \tag{5}$$

Here, σ denotes the sigmoid activation function, and g is another deep-parameterized function parameterized by weight $\mathbf{w}$. By definition of the Bernoulli process, the sampled measure is given by:

$$Q_t(\boldsymbol{\psi}) \ = \ \sum_{i=1}^{n} c_i \cdot \mathbb{I}\Big(\boldsymbol{\psi} = \boldsymbol{\phi}_i\Big) \ , \tag{6}$$

where $c_i \in \{0, 1\}$ is the outcome of a Bernoulli trial with bias $\sigma(g(\boldsymbol{\phi}_i;\mathbf{w}))$, that is:

$$\mathbb{P}_i(c_i) \quad = \quad \sigma\Big(g(\boldsymbol{\phi}_i;\mathbf{w})\Big)^{c_i} \cdot \Big(1 - \sigma\Big(g(\boldsymbol{\phi}_i;\mathbf{w})\Big)\Big)^{1-c_i} \ . \tag{7}$$

In layman term, this process simply means we will observe the result of a coin toss c_i for every summarizing prompt $\boldsymbol{\phi}_i$. If this coin lands on head ($c_i = 1$), the summarizing prompt $\boldsymbol{\phi}_i$ will be included in the set of mean parameters. Vice versa, if this coin lands on tail, we will skip $\boldsymbol{\phi}_i$. Because of this, we remark that n_t could be different from the number of prompts in the previous round.

3.2.2 Prompt Aggregation

Given the above generative story, we can now describe our algorithm to find the set of summarizing prompts $\{\boldsymbol{\phi}_i\}_{i=1}^n$, which maximize the likelihood of observing local prompts $\{\boldsymbol{\omega}_t\}_{t=1}^m$. The resulting optimal set of summarizing prompts can then be used to re-program the pre-trained model, which correspond to the global fine-tuned solution. Assuming that each prompt captures a particular fine-tuning pattern or concept, summarizing the prompt sets characterizing the local solutions will allow local models to be aggregated on the concept level, naturally mitigating the solution drift effect caused by compounding impact of data heterogeneity and pre-trained weight interference. This is substantiated in the overall computation process below.

Let $z_{tk}^i \in \{0, 1\}$ denotes whether the local prompt $\boldsymbol{\omega}_{tk}$ was sampled from a Gaussian centered at the i-th summarizing prompt $\boldsymbol{\phi}_i$. That is, the assignment variable $z_{tk}^i = 1$ if and only if there exists $k \in [n_t]$ such that $\boldsymbol{\psi}_{tk} = \boldsymbol{\phi}_i$. Each client t can now be represented as $(\boldsymbol{\omega}_t, \mathbf{z}_t)$ where $\mathbf{z}_t = \{\mathbf{z}_t^i\}_{i=1}^n$, $\mathbf{z}_t^i = (z_{t1}^i, z_{t2}^i, \ldots, z_{tn_t}^i)$ and $\boldsymbol{\omega}_t = (\boldsymbol{\omega}_{t1}, \boldsymbol{\omega}_{t2}, \ldots, \boldsymbol{\omega}_{tn_t})$.

Now, using the generative process specified in the previous section and all its defining parameters $\boldsymbol{\varphi} = (\mathbf{w}, \gamma, \{\boldsymbol{\phi}_{i=1}^n\})$, the log likelihood of each client can be derived as follow,

$$\log \mathbb{P}\left(\boldsymbol{\omega}_t, \mathbf{z}_t \mid \boldsymbol{\varphi}\right) \quad = \quad \log \mathbb{P}\left(\boldsymbol{\omega}_t \mid \mathbf{z}_t, \boldsymbol{\varphi}\right) \ + \ \log \mathbb{P}\left(\mathbf{z}_t \mid \boldsymbol{\varphi}\right) \ , \tag{8}$$

where each of the summand on the right-hand side is computed below. First, given $(\mathbf{z}_t, \boldsymbol{\varphi})$,

$$\mathbb{P}\Big(\boldsymbol{\omega}_t \mid \mathbf{z}_t, \boldsymbol{\varphi}\Big) = \prod_{k=1}^{n_t} \mathbb{N}\Big(\boldsymbol{\omega}_{tk} \mid \boldsymbol{\psi}_{tk}, \mathrm{diag}\Big(\alpha\Big(\boldsymbol{\psi}_{tk};\gamma\Big)\Big)\Big) \ \text{ with } \boldsymbol{\psi}_{tk} = \Big(z_{tk}^1 \cdot \boldsymbol{\phi}_1 + \ldots + z_{tk}^n \cdot \boldsymbol{\phi}_n\Big) \tag{9}$$

Algorithm 1 Probabilistic Federated Prompt Tuning (PFPT)

```
input: pre-trained model $F$, no. $\tau$ of iterations, no. $m$ of sampled clients per iteration
output: optimized set of prompts $\boldsymbol{\Phi}$
 1: initialize global summarizing prompts $\boldsymbol{\Phi}$
 2: for $s = 1$ to $\tau$ do
 3:     sample $m$ clients with private datasets $\{D_t\}_{t=1}^{m}$
 4:     for $t = 1$ to $m$ do
 5:         $\boldsymbol{\omega}_t \leftarrow \mathrm{client}(F, D_t, \boldsymbol{\Phi})$ // local prompt-tuning – optimizing $L_t(\omega_t)$ in Eq. (3)
 6:     end for
 7:     $\boldsymbol{\Phi}, \mathbf{z} \leftarrow \mathrm{server}(\{\boldsymbol{\omega}_t\}_{t=1}^{m})$ // solving Eq. (11)
 8:     $\boldsymbol{\Phi} \leftarrow \left\{\boldsymbol{\phi}_i \in \boldsymbol{\Phi} \mid \sum_{t=1}^{m}\sum_{k=1}^{n_t} z_{tk}^{i} > 0\right\}$ // remove inactive prompts
 9: end for
10: return the set $\boldsymbol{\Phi}$ of optimal prompts
```

Then, given $\boldsymbol{\varphi}$ and let $c_{ti} = z_{t1}^{i} + z_{t2}^{i} + \ldots + z_{tn_t}^{i}$,

$$\mathbb{P}\Big(\mathbf{z}_t \mid \boldsymbol{\varphi}\Big) = \prod_{i=1}^{n}\left(\sigma\Big(g(\boldsymbol{\phi}_i;\mathbf{w})\Big)^{c_{ti}}\right) \times \prod_{i=1}^{n}\left(\Big(1-\sigma\Big(g(\boldsymbol{\phi}_i;\mathbf{w})\Big)\Big)^{\left(1-c_{ti}\right)}\right) \tag{10}$$

Eq. (9) follows from the Gaussian likelihood of local prompt in Eq. (4) with the additional identification form for $\boldsymbol{\psi}_{tk}$ which is computable if $\mathbf{z}_t$ is given. Eq. (10) on the other hand is the consequence of the Bernoulli process in Eq. (6), which might not be immediately straight-forward. Its detailed derivation will be given in Appendix B.

3.2.3 Overall Workflow

Using Eq. (9) and (10), we have the following prompt aggregation formulation,

$$\boldsymbol{\Phi}^* = \operatorname*{argmax}_{\boldsymbol{\Phi}} \left\{ \max_{\mathbf{w},\boldsymbol{\gamma},\mathbf{z}} \sum_{t=1}^{m} \log \mathbb{P}\Big(\boldsymbol{\omega}_t, \mathbf{z}_t \mid \mathbf{w}, \boldsymbol{\gamma}, \boldsymbol{\Phi}\Big) \right\} \tag{11}$$

where $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_i\}_{i=1}^{n}$, $\mathbf{z} = \{\mathbf{z}_t\}_{t=1}^{m}$, and $\boldsymbol{\gamma}$ is previously defined in Eq. (4). The probability term on the right-hand side can be further expanded into computable terms using Eq. (8), Eq. (9), and Eq. (10). Solving Eq. (11) is therefore the key step in our federated fine-tuning framework, whose overall flow is summarized in Alg. 1. Our algorithm proceeds in multiple iterations. At each iteration, a set of m clients is sampled. Each client selects the most relevant subset of summarizing prompts from the server using a mechanism adapted from continual visual prompt-tuning [49]. The selected subset of summarizing prompts are then fine-tuned using local data (see lines 3-4). The sets of fine-tuned prompts are subsequently uploaded to the server, which performs the aggregation via solving Eq. (11) (see line 6). The aggregation step has two subtleties: (1) the total number of global prompts, $n = n_1 + \ldots + n_t$, is proportional to the total number of prompts across clients, making the entire algorithm non-parametric; and (2) once the optimal assignment parameters $\mathbf{z}$ are found via solving Eq. (11), global prompts that were not assigned to any client will be removed (see lines 7 and 8). Intuitively, if the sampled clients are similar, their prompt sets will substantially overlap, which reduces the total number of distinct prompts selected by all clients. Otherwise, if the sampled clients are dissimilar, there will be less overlapping and the union set of selected global prompts will expand, increasing the complexity of the resulting model to match the increased data heterogeneity.

3.3 Optimization

Solving Eq. (11) above is however not trivial due to its mixed set of discrete/continuous variables. To sidestep this intractability, we instead formulate it as a bi-level optimization that alternates between two sub-tasks: (1) optimizing $\mathbf{z}$ given $\mathbf{w}, \boldsymbol{\gamma}$ and $\boldsymbol{\Phi}$; and (2) optimizing $\boldsymbol{\Phi}, \mathbf{w}, \boldsymbol{\gamma}$ given $\mathbf{z}$. Among which, solving sub-task (2) is straight-forward since it only requires being able to differentiate the expression in Eq. (9) and Eq. (10), which is trivial if we use the current estimation of the (discrete) assignment parameters $\mathbf{z}$. On the other hand, solving sub-task (1) is seemingly prohibitive expensive because it involves optimizing the (discrete) assignment parameters $\mathbf{z}$. Fortunately, this can be mitigated by casting both Eq. (9) and Eq. (10) into a linear form with respect to the assignment parameter $\mathbf{z}$. Intuitively, it might be confusing why such linear form can be achieved at all given that $\mathbf{z}$ is input to a non-linear log probability function in Eq. (9) and Eq. (10). However, we must recognize that for each client t, there is at most one local prompt from t which can be associated with the i^{th} summarizing prompt. This implies $\mathbf{z}_t^i \triangleq (z_{t1}^i, z_{t2}^i, \ldots, z_{tn_t}^i)$ is either a zero or one-hot vector. This is an important observation because a linear function of such zero or one-hot vector will remain linear regardless of any (non-linear) post-processing transformation. It is shown in Lemma 3.1 below.

Lemma 3.1. *For any scalar function $g(\mathbf{r})$ and a binary vector $\boldsymbol{\zeta} = [\zeta_1, \zeta_2, \ldots, \zeta_n]$ such that $\zeta_i \in \{0, 1\}$ and $\boldsymbol{\zeta}$ has at most one non-zero component, we have*

$$g\left(\sum_{i=1}^{n} \zeta_i \cdot \mathbf{r}_i\right) \;=\; \sum_{i=1}^{n}\left(\zeta_i \cdot g(\mathbf{r}_i)\right) \tag{12}$$

with respect to any set $\{\mathbf{r}_i\}_{i=1}^n$ of valid inputs to $g(\mathbf{r})$.

The proof of Lemma 3.1 is straight-forward. First, if there is no non-zero component, both sides of Eq. (12) evaluate to $g(0)$. Otherwise, suppose the only non-zero component appears at position κ, both sides of Eq. (12) will evaluate to $g(\mathbf{r}_\kappa)$. In both cases, Eq. (12) holds. Using Lemma 3.1, we can establish the desired linear forms for Eq. (10) and Eq. (11) which are its immediate consequences.

Lemma 3.2. *Let $\mathbb{P}(\boldsymbol{\omega}_t \mid \mathbf{z}_t, \boldsymbol{\varphi})$ defined as in Eq. (9). Let $L_1(\mathbf{z}) = \sum_{t=1}^m \log \mathbb{P}(\boldsymbol{\omega}_t \mid \mathbf{z}_t, \boldsymbol{\varphi})$, considering $(\boldsymbol{\omega}_t, \boldsymbol{\varphi})$ as constants. We have*

$$L_1(\mathbf{z}) \;=\; \sum_{i=1}^{n}\sum_{t=1}^{m}\sum_{k=1}^{n_t} z_{tk}^i \cdot \log \mathbb{N}\left(\boldsymbol{\omega}_{tk} | \boldsymbol{\phi}_i, \operatorname{diag}\left(\alpha\left(\boldsymbol{\phi}_i; \boldsymbol{\gamma}\right)\right)\right) \tag{13}$$

which is linear in terms of the assignment parameter $\mathbf{z}$.

Lemma 3.3. *Let $\mathbb{P}(\mathbf{z}_t \mid \boldsymbol{\varphi})$ defined as in Eq. (10). Let $L_2(\mathbf{z}) = \sum_{t=1}^m \log \mathbb{P}(\mathbf{z}_t \mid \boldsymbol{\varphi})$, considering $\boldsymbol{\varphi}$ as constants. Then, we have*

$$L_2(\mathbf{z}) \;=\; \sum_{i=1}^{n}\sum_{t=1}^{m}\sum_{k=1}^{n_t} z_{tk}^i \cdot \log\left(\frac{\sigma\Big(g(\boldsymbol{\phi}_i; \mathbf{w})\Big)}{1 - \sigma\Big(g(\boldsymbol{\phi}_i; \mathbf{w})\Big)}\right) \;+\; \sum_{i=1}^{n}\sum_{t=1}^{m} \log\left(1 - \sigma\Big(g(\boldsymbol{\phi}_i; \mathbf{w})\Big)\right) \tag{14}$$

which is linear in terms of the assignment parameter $\mathbf{z}$.

The detailed proof for Lemma 3.2 and Lemma 3.3 are deferred to Appendix C. Using these results, we can put together the overall optimization task for $\mathbf{z}$ while fixing the rest of the parameterization

$$\mathbf{z}^* \;=\; \operatorname*{argmax}_{\mathbf{z}}\left\{\sum_{t=1}^{m} \log \mathbb{P}\Big(\boldsymbol{\omega}_t, \mathbf{z}_t \mid \boldsymbol{\varphi}\Big)\right\} \;=\; \operatorname*{argmax}_{\mathbf{z}}\left\{L_1(\mathbf{z}) \;+\; L_2(\mathbf{z})\right\} \tag{15}$$

which is a weighted linear optimization task. The second equality follows from Eq. 8 and the results of Lemma 3.2 and Lemma 3.3. Now, if we further choose to optimize $\mathbf{z}_t$ iteratively while fixing in addition $\mathbf{z}_{-t}$, Eq. (15) reduces to a weighted bipartite matching task, which can be solved effectively in $\mathbb{O}((\max_{t=1}^m n_t)^3)$ processing time using the Hungarian algorithm [50]. A detailed pseudo-code implementing the above scheme is deferred to Appendix D.

4 Empirical Results

This section presents our empirical studies on a variety of computer vision datasets, including CIFAR-10 and CIFAR-100 [51], TinyImageNet [52] and a synthetic, diverse dataset created by pooling together the MNIST-M [53], Fashion-MNIST [54], CINIC-10 [55] and MMAFEDB (available on Kaggle) datasets, which is referred to as the 4-dataset.

Our experiments are conducted on two data settings: (a) Dirichlet-based heterogeneous partition following a previous setup in [6]; and (b) manual imbalanced data partition, as detailed below.

Heterogeneous Partition. We partition the training data into m subsets (for m clients). Each client has observations of all classes but the distributions of classes across clients are different. We simulate this using a $\text{Dirichlet}(\alpha \cdot \mathbf{1}_s)$ distribution over an s-dimensional simplex where s is the number of classes and α is the concentration parameter.

Table 1: Accuracy ($\%$) achieved on the CIFAR-10 dataset by PFPT and other baselines.

	FedAvg-PT	FedProx-PT	Scaffold-PT	FedOpt-PT	PFedPG	GMM-PT	PFPT (ours)
$\alpha = 0.5$	94.06 ± 0.45	94.26 ± 0.34	92.26 ± 0.18	90.70 ± 0.54	88.00 ± 0.22	92.63 ± 0.16	$\mathbf{94.39 \pm 0.51}$
$\alpha = 0.1$	93.05 ± 0.03	93.05 ± 0.23	91.82 ± 0.34	88.20 ± 0.39	77.25 ± 0.23	92.68 ± 0.28	$\mathbf{93.39 \pm 0.22}$
imbalance	91.21 ± 0.16	91.08 ± 0.25	87.06 ± 1.27	79.32 ± 1.73	43.16 ± 0.27	90.50 ± 0.17	$\mathbf{91.45 \pm 0.08}$

Table 2: Accuracy ($\%$) achieved on the CIFAR-100 dataset by PFPT and other baselines.

	FedAvg-PT	FedProx-PT	Scaffold-PT	FedOpt-PT	PFedPG-PT	GMM-PT	PFPT (ours)
$\alpha = 0.5$	79.40 ± 0.27	79.21 ± 0.31	70.83 ± 0.18	59.58 ± 1.13	43.47 ± 0.39	77.50 ± 0.12	$\mathbf{80.24 \pm 0.24}$
$\alpha = 0.1$	74.50 ± 0.46	73.60 ± 0.38	69.94 ± 0.63	60.31 ± 0.39	29.46 ± 0.80	70.49 ± 0.46	$\mathbf{75.08 \pm 0.51}$
imbalance	70.07 ± 0.35	70.67 ± 0.25	69.35 ± 0.96	58.86 ± 0.77	11.57 ± 0.25	66.95 ± 0.46	$\mathbf{72.05 \pm 0.93}$

Table 3: Accuracy ($\%$) achieved on the TinyImageNet dataset by PFPT and other baselines.

	FedAvg-PT	FedProx-PT	Scaffold-PT	FedOpt-PT	PFedPG	GMM-PT	PFPT (ours)
$\alpha = 0.5$	86.38 ± 0.18	86.05 ± 0.47	78.70 ± 0.33	61.55 ± 0.28	50.92 ± 0.44	84.73 ± 0.20	$\mathbf{86.91 \pm 0.14}$
$\alpha = 0.1$	78.58 ± 0.57	79.19 ± 0.28	78.02 ± 0.33	63.81 ± 0.79	34.34 ± 0.10	76.90 ± 0.28	$\mathbf{82.31 \pm 0.26}$
imbalance	75.21 ± 0.73	75.49 ± 0.38	76.88 ± 0.45	64.33 ± 0.61	11.92 ± 0.13	73.85 ± 0.36	$\mathbf{78.21 \pm 1.25}$

Our experiments are conducted with $\alpha = 0.1$ and $\alpha = 0.5$. For each client r, we drawn a sample $\mathbf{p}_r \sim \mathrm{Dirichlet}(\alpha \cdot \mathbf{1}_s)$ where $\mathbf{p}_{r,c} \in (0, 1)$ specifies the percentage of examples in class c to be assigned to client r. For the CIFAR-10, CIFAR-100, and TinyImageNet datasets, we set $m = 100$. For the 4-dataset, we simulate 20 partitions for each of the 4 sub-dataset using $\mathrm{Dirichlet}(\alpha \cdot \mathbf{1}_s)$ with $s = 10, 10, 10$, and 7 for MNIST-M, FASHION-MNIST, CINIC-10, and MMAFEDB. This amounts to a total of 80 clients with diverse and heterogeneous data distribution.

Imbalance Partition. For the CIFAR-10 (10-class), CIFAR-100 (100-class) and TinyImageNet (200-class) datasets, the training data of each client is set to be dominated by a particular subset of classes, which amounts to 10% of the total number of classes. 99% of the local dataset of each client is set to belong to a certain 10% of the total number of classes. For our synthetic 4-dataset, we partition each of the 4 sub-dataset into 20 subsets. Each subset is set so that 99% of its data points belong to a single class. The remaining 1% data of each class are then pooled together and evenly distributed among all clients. This amounts to 80 clients with extremely imbalance local datasets.

The above schemes are applied only on the train partition of each dataset. We use the default train/test partition for CIFAR-10, CIFAR-100, and TinyImageNet. For the 4-dataset, we sample 30K data points from the default train partition of each of its 4 sub-datasets. We also sample 2.5K data points from the default test partition of each sub-dataset. This amounts to a synthetic dataset with 120K data points in the train partition and 10K data points in the test partition.

For each data setup on each dataset, we compare the performance of our probabilistic federated prompt-tuning (PFPT) algorithm with those of a representative set of state-of-the-art federated learning algorithms adapted to the prompt-tuning setting in Eq. (3), which include FedAvg [2], FedProx [29], Scaffold [32], FedOpt [56] and PFedPG [28]. PFedPG is a personalization method that is originally measured based on how well its personalized models perform on their corresponding local test sets. In this context, the performance target is however set on a global test set so we use PFedPG's common model for evaluation. We will refer to these adapted baselines as FedAvg-PT, FedProx-PT, Scaffold-PT, FedOpt-PT, and PFedPG. In addition, we also compare PFPT against a simple prompt clustering baseline, which replaces the prompt averaging of FedAvg by a Gaussian Mixture Model (GMM) clustering in which the cluster centroids are returned as the aggregated prompts. We refer to this baseline as GMM-PT. All results are averaged over 5 independent runs and reported in Tables 1 to 4 below. The result of each run is evaluated on a global test set, which comprises the entire test partition.

4.1 Non-IID Data with Locally Skewed Class Distributions

Table 1 reports the performance of PFPT and various federated prompt-tuning baselines on the CIFAR-10 dataset. In all three settings (i.e., heterogeneous partitions with $\alpha = 0.5$ and 0.1, as well as the imbalance partition), PFPT consistently achieves the best performance. On average, the classification accuracy of our method improves by 0.3% over the closest competitors, FedAvg-PT and FedProx-PT. While this gain seems modest, we note that the prompt tuning performance tends to fall off more significantly on several FL frameworks that were devised to counteract the effect of imbalanced data, such as Scaffold-PT and FedOpt-PT. In fact, the accuracy our method

Table 4: Accuracy (%) achieved on the synthetic 4-dataset, which combines data from MNIST-M [53], Fashion-MNIST [54], CINIC-10 [55], and MMAFEDB[1], by PFPT and other baselines.

	FedAvg-PT	FedProx-PT	Scaffold-PT	FedOpt-PT	PFedPG	GMM-PT	PFPT (ours)
$\alpha = 0.5$	59.48 ± 0.59	59.50 ± 0.70	54.51 ± 1.56	40.69 ± 2.29	30.54 ± 0.50	57.05 ± 1.04	$\mathbf{76.89 \pm 0.17}$
$\alpha = 0.1$	48.74 ± 0.63	48.71 ± 2.13	46.65 ± 3.17	43.51 ± 0.78	25.40 ± 0.17	46.82 ± 2.80	$\mathbf{70.29 \pm 0.32}$
imbalance	39.28 ± 0.55	39.65 ± 1.89	23.79 ± 1.24	44.43 ± 1.78	22.54 ± 0.32	41.37 ± 1.12	$\mathbf{62.23 \pm 1.02}$

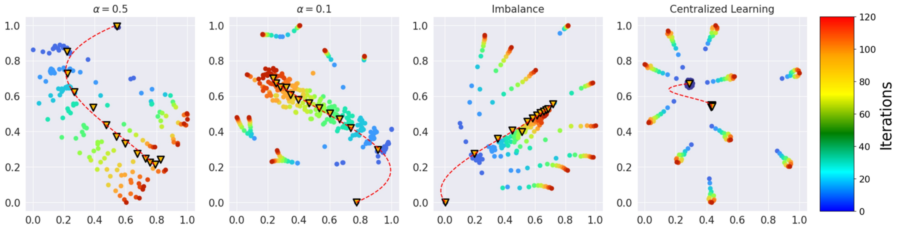

Figure 3: t-SNE plots of the (learned) summarizing prompts of PFPT on CIFAR-100 over 120 communication iterations with different heterogeneity settings. Yellow triangles denote the centroids of the t-SNE embeddings of the prompts. The dashed red line visualizes their trajectories.

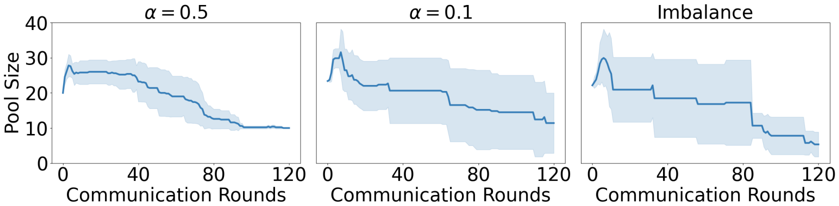

Figure 4: Variations in CIFAR-100 global prompt pool size across 120 communication rounds under different heterogeneity settings.

respectively improves by 2.7% and 7.0% over these baselines. This result clearly suggests that prompt tuning performance is more vulnerable to the heterogeneity challenge in FL, and thus requires a more specialized treatment. Finally, we observe that PFedPG achieves the worst performance among all baselines (23.6% worse than PFPT on average). This performance gap is especially pronounced on the imbalance setting. While PFedPG claimed to deal with both heterogeneity and prompt-tuning, this result is not surprising since it assumes that the local test sets are partitioned similarly to the local training sets, whereas our results are obtained on a holistic global test set.

We further repeat this experiment on the CIFAR-100 and TinyImageNet datasets, which are both more challenging than CIFAR-10 due to the increased number of classes (i.e., 100 and 200 classes respectively). The results of these experiments are reported in Table 2 and 3. As expected, our method still consistently outperforms other baselines. However, the performance gap between Pfpt and other methods tends to widen proportionately to the increased difficulty of the dataset. For example, in the CIFAR-100 experiment, our method is 1.2% better than the second best baseline (FedAvg-PT) and 47.6% better than the worst baseline (PFedPG).

In the TinyImageNet experiment, our method is 2.4% better than FedAvg-PT and 50.1% better than PFedPG. These results are therefore consistent with our findings in Table 1. Finally, Table 4 records the prompt tuning performance of the same baselines on the most challenging scenario, which combines 4 datasets from completely different vision domains. We observe a significant performance gap between PFPT and baselines that have been keeping up with its performance in earlier experiments, such as FedAvg-PT (now 20.6% worse) and FedProx-PT (now 20.5% worse). More interestingly, GMM-PT, which has performed on par with our method in previous experiments, observes a significant drop in performance (now 21.4% worse). These extremely wide performance gaps are due to the fact that local prompts are more diverse (to account for highly heterogeneous tasks), which require a more refined alignment technique to facilitate effective aggregation.

4.2 Non-IID Data with Globally Skewed Class Distributions

In addition to the above setting which features non-IID data and locally skewed class distribution, we also evaluated the performance of our framework in a more adverse setting where the class distribution will remain skewed even if all local datasets are aggregated. That is, local datasets in this setting are non-IID draws from a long-tailed data distribution parameterized by an imbalance factor IF $= (\max_c n_c)/(\min_c n_c)$ characterizing the ration between sample sizes (n_c) of the most frequent and least frequent classes [57, 58]. Such (global) long-tailed datasets are synthetically created from the original CIFAR-100 and ImageNet datasets following the data simulation in [59]. Following previous experiment setup in previous (non-ViT and no prompt-tuning) baseline methods FEDIC [57] and CReFF [58] [2], the resulting long-tailed datasets (CIFAR-100-LT and ImageNet-LT) are configured to have IF $= 10, 50, 100$ for CIFAR-100 and IF $= 1280/5$ for ImageNet. The reported results in Table 5 show that our method is also highly effective in this setting, achieving significantly higher accuracy than the current SOTA [57, 58] across all datasets and imbalance setups.

Table 5: Accuracy (%) achieved on long-tailed datasets by PFPT, CREFF [58] and FEDIC [57].

Non-IID ($\alpha = 0.1$)	CIFAR-100-LT			ImageNet-LT (IF $= 1280/5$)
	IF $=$ 100	IF $=$ 50	IF $=$ 10	
FEDIC	33.67	34.74	41.93	28.93
CREFF	26.19	28.32	35.49	26.31
PFPT (ours)	**60.74**	**65.54**	**71.66**	**75.54**

4.3 Prompt Convergence and Diversity

To generate insights regarding the learned prompts, we plot the 2-dimensional t-SNE embeddings of all CIFAR-100 global prompts discovered by our method across 120 communication rounds. This is repeated for the two heterogeneity and one data imbalance FL settings as well as the centralized data setting (Fig. 3). The yellow triangles in each plot mark the corresponding centroids of the prompt embeddings (one per 10 communication iterations). We also plot a best-fit spline curve (dashed and colored in red) through these centroids to visualize their update trajectory. In all settings, the distance between successive centroids consistently gets smaller as training progresses, which suggests that the learned prompts generally converge well. This is also corroborated by Fig. 4, which shows that the number of global prompts also converges over 120 communication iterations. It is also observed that in the centralized learning scenario in which no heterogeneity is present, the convergence happens much faster. The prompts quickly converge within 10 communication iterations. We also observe a gradual increase in the spread of the prompts with respect to their centroid. This suggests that the learned prompts are optimized by our method to capture the data diversity across participating clients. More interestingly, as we move from the standard heterogeneity settings (with $\alpha = 0.1$ and $\alpha = 0.5$) to the extremely heterogeneous setting with imbalance (local) data, the magnitude of this spread becomes larger, confirming the strong correlation between task heterogeneity and prompt diversity.

5 Conclusion

Our paper presents a new and effective approach to address the challenges of prompt-tuning pre-trained models in federated data scenarios with diverse local data distributions. Our proposed approach is a hierarchical probabilistic framework that models server aggregation as a non-parametric alignment of locally learned prompt sets into summarizing prompts. In each subsequent communication round, local clients sample and perform local updates on relevant prompts from the previous set of summarizing prompts. In this manner, our approach bypasses the effect of solution drift and outperforms existing federated learning techniques (applied on prompt-tuning) on various computer vision datasets. The reported results emphasize the cost-efficiency and efficacy of our approach in combating data heterogeneity within extremely diverse federated scenarios.

Acknowledgements and Disclosure of Funding. This work used GPU compute resource at SDSC through allocation CIS230391 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services and Support (ACCESS) program [60], which is supported by U.S. National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296. My T. Thai acknowledges the support of National Science Foundation grants SCH-2123809 and III-2416606. T.-W. Weng is supported by National Science Foundation awards CCF-2107189, IIS-2313105, IIS-2430539, the Hellman Fellowship, and Intel Rising Star Faculty Award.

[2] These federated fine-tuning baselines use full fine-tuning with ResNet as their backbone. Both baselines were not evaluated in our main setting in Section 4.1.

References

[1] Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated Optimization: Distributed Machine Learning for On-Device Intelligence. *CoRR*, abs/1610.02527, 2016.

[2] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proc. AISTATS*, pages 1273–1282, 2017.

[3] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated Multi-Task Learning. *Advances in Neural Information Processing Systems*, 30, 2017.

[4] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista A. Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and Open Problems in Federated Learning. *CoRR*, abs/1912.04977, 2019.

[5] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic Federated Learning. *CoRR*, abs/1902.00146, 2019.

[6] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian Nonparametric Federated Learning of Neural Networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7252–7261. PMLR, 09–15 Jun 2019.

[7] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, and Nghia Hoang. Statistical Model Aggregation via Parameter Matching. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[8] Tengfei Ma, Trong Nghia Hoang, and Jie Chen. Federated learning of models pre-trained on different features with consensus graphs. In *Uncertainty in Artificial Intelligence*, 2023.

[9] Ziwei Fan, Hao Ding, Anoop Deoras, and Trong Nghia Hoang. Personalized federated domain adaptation for item-to-item recommendation. In *Uncertainty in Artificial Intelligence*, 2023.

[10] T. N. Hoang, Q. M. Hoang, K. H. Low, and J. P. How. Collective online learning of Gaussian processes in massive multi-agent systems. In *Proc. AAAI*, 2019.

[11] Minh Hoang and Trong Nghia Hoang. Few-shot learning via repurposing ensemble of black-box models. In *AAAI Conference on Artificial Intelligence*, 2024.

[12] Trong Nghia Hoang. Effective knowledge representation and utilization for sustainable collaborative learning across heterogeneous systems. *AI Magazine*, 45(3):404–410, 2024.

[13] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen A. Creel, Jared Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas F. Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling,

Fereshte Khani, O. Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir P. Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Benjamin Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, J. F. Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Robert Reich, Hongyu Ren, Frieda Rong, Yusuf H. Roohani, Camilo Ruiz, Jack Ryan, Christopher R'e, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishna Parasuram Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei A. Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the Opportunities and Risks of Foundation Models. *ArXiv*, 2021.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[15] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *CoRR*, abs/1910.13461, 2019.

[16] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692, 2019.

[17] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *JMLR*, 21(1):5485–5551, 2020.

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv preprint arXiv:2010.11929*, 2020.

[19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, pages 248–255. IEEE, 2009.

[20] John Nguyen, Jianyu Wang, Kshitiz Malik, Maziar Sanjabi, and Michael Rabbat. Where to Begin? On the Impact of Pre-Training and Initialization in Federated Learning, 2023.

[21] Wang Lu, Xixu Hu, Jindong Wang, and Xing Xie. FedCLIP: Fast Generalization and Personalization for CLIP in Federated Learning, 2023.

[22] Yuanyishu Tian, Yao Wan, Lingjuan Lyu, Dezhong Yao, Hai Jin, and Lichao Sun. FedBERT: When Federated Learning Meets Pre-Training. *ACM Transaction of Intelligent Systems and Technology*, 13(4), aug 2022.

[23] Tuo Zhang, Tiantian Feng, Samiul Alam, Dimitrios Dimitriadis, Mi Zhang, Shrikanth S. Narayanan, and Salman Avestimehr. GPT-FL: Generative Pre-trained Model-Assisted Federated Learning, 2023.

[24] Yeachan Kim, Junho Kim, Wing-Lam Mok, Jun-Hyung Park, and SangKeun Lee. Client-Customized Adaptation for Parameter-Efficient Federated Learning. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1159–1172, Toronto, Canada, July 2023. Association for Computational Linguistics.

[25] Ankur Agarwal, Mehdi Rezagholizadeh, and Prasanna Parthasarathi. Practical Takes on Federated Learning with Pretrained Language Models. In Andreas Vlachos and Isabelle Augenstein, editors, *EACL*, pages 454–471, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

[26] Tao Guo, Song Guo, Junxiao Wang, Xueyang Tang, and Wenchao Xu. PromptFL: Let Federated Participants Cooperatively Learn Prompts Instead of Models-Federated Learning in Age of Foundation Model. *IEEE Transactions on Mobile Computing*, 2023.

[27] Haodong Zhao, Wei Du, Fangqi Li, Peixuan Li, and Gongshen Liu. FedPrompt: Communication-Efficient and Privacy Preserving Prompt Tuning in Federated Learning. In *Proc. ICASSP*, 2023.

[28] Fu-En Yang, Chien-Yi Wang, and Yu-Chiang Frank Wang. Efficient Model Personalization in Federated Learning via Client-Specific Prompt Generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19159–19168, October 2023.

[29] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and Robust Federated Learning Through Personalization. In Marina Meila and Tong Zhang, editors, *International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6357–6368. PMLR, 18–24 Jul 2021.

[30] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated Learning based on Dynamic Regularization. In *International Conference on Learning Representations*, 2021.

[31] Ohad Shamir, Nati Srebro, and Tong Zhang. Communication-Efficient Distributed Optimization using an Approximate Newton-type Method. In *International Conference on Machine Learning*, pages 1000–1008. PMLR, 2014.

[32] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5132–5143. PMLR, 13–18 Jul 2020.

[33] Qinbin Li, Bingsheng He, and Dawn Song. Model-Contrastive Federated Learning. In *CVPR*, 2021.

[34] Hong-You Chen and Wei-Lun Chao. FedDistill: Making Bayesian Model Ensemble Applicable to Federated Learning. In *ICLR*, 2021.

[35] Lin Zhang, Li Shen, Liang Ding, Dacheng Tao, and Ling-Yu Duan. Fine-Tuning Global Model via Data-Free Knowledge Distillation for Non-IID Federated Learning. In *CVPR*, pages 10174–10183, June 2022.

[36] Michael Kamp, Jonas Fischer, and Jilles Vreeken. Federated Learning from Small Datasets. *arXiv preprint arXiv:2110.03469*, 2021.

[37] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized Federated Learning with Theoretical Guarantees: A Model-Agnostic Meta-Learning Approach. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3557–3568. Curran Associates, Inc., 2020.

[38] Canh T. Dinh, Nguyen Hoang Tran, and Tuan Dung Nguyen. Personalized Federated Learning with Moreau Envelopes. In *NeurIPS*, 2020.

[39] Qiying Yu, Yang Liu, Yimu Wang, Ke Xu, and Jingjing Liu. Multimodal Federated Learning via Contrastive Representation Ensemble. In *ICLR*, 2023.

[40] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting Shared Representations for Personalized Federated Learning. In Marina Meila and Tong Zhang, editors, *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 2089–2099. PMLR, 18–24 Jul 2021.

[41] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive Personalized Federated Learning. *CoRR*, abs/2003.13461, 2020.

[42] Paul Pu Liang, Terrance Liu, Ziyin Liu, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think Locally, Act Globally: Federated Learning with Local and Global Representations. *CoRR*, abs/2001.01523, 2020.

[43] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated Learning with Personalization Layers. *CoRR*, abs/1912.00818, 2019.

[44] Filip Hanzely and Peter Richtárik. Federated Learning of a Mixture of Global and Local Models. *CoRR*, abs/2002.05516, 2020.

[45] Dianwen Ng, Xiang Lan, Melissa Min-Szu Yao, Wing P Chan, and Mengling Feng. Federated Learning: A Collaborative Effort to Achieve Better Medical Imaging Models for Individual Sites that Have Small Labelled Datasets. *Quantitative Imaging in Medicine and Surgery*, 11(2):852, 2021.

[46] Ying Zhao, Junjun Chen, Di Wu, Jian Teng, and Shui Yu. Multi-Task Network Anomaly Detection using Federated Learning. In *Proceedings of the 10th International Symposium on Information and Communication Technology*, pages 273–279, 2019.

[47] Yu Zhang, Guoming Tang, Qianyi Huang, Yi Wang, Kui Wu, Keping Yu, and Xun Shao. FedNILM: Applying Federated Learning to NILM Applications at the Edge. *IEEE Transactions on Green Communications and Networking*, 2022.

[48] Ying Zhao, Junjun Chen, Qianling Guo, Jian Teng, and Di Wu. Network Anomaly Detection using Federated Learning and Transfer Learning. In *Security and Privacy in Digital Economy: First International Conference, SPDE 2020, Quzhou, China, October 30–November 1, 2020, Proceedings*, pages 219–231. Springer, 2020.

[49] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to Prompt for Continual Learning. In *CVPR*, pages 139–149, 2022.

[50] Harold W Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[51] Alex Krizhevsky, Geoffrey Hinton, et al. Learning Multiple Layers of Features from Tiny Images, 2009.

[52] Ya Le and Xuan Yang. Tiny Imagenet Visual Recognition Challenge. *CS 231N*, 7(7):3, 2015.

[53] Seunghun Lee, Sunghyun Cho, and Sunghoon Im. DRANet: Disentangling Representation and Adaptation Networks for Unsupervised Cross-Domain Adaptation. In *CVPR*, pages 15252–15261, June 2021.

[54] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. *CoRR*, abs/1708.07747, 2017.

[55] Luke N. Darlow, Elliot J. Crowley, Antreas Antoniou, and Amos J. Storkey. CINIC-10 is not ImageNet or CIFAR-10, 2018.

[56] Muhammad Asad, Ahmed Moustafa, and Takayuki Ito. FedOpt: Towards Communication Efficiency and Privacy Preservation in Federated Learning. *Applied Sciences*, 10(8):2864, 2020.

[57] Xinyi Shang, Yang Lu, Yiu-Ming Cheung, and Hanzi Wang. FEDIC: Federated Learning on Non-IID and Long-Tailed Data via Calibrated Distillation . In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, Los Alamitos, CA, USA, July 2022. IEEE Computer Society.

[58] Xinyi Shang, Yang Lu, Gang Huang, and Hanzi Wang. Federated learning on heterogeneous and long-tailed data via classifier re-training with federated features. *arXiv preprint arXiv:2204.13399*, 2022.

[59] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Aréchiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Neural Information Processing Systems*, 2019.

[60] Timothy J. Boerner, Stephen Deems, Thomas R. Furlani, Shelley L. Knuth, and John Towns. Access: Advancing innovation: Nsf's advanced cyberinfrastructure coordination ecosystem: Services & support. In *Practice and Experience in Advanced Research Computing 2023: Computing for the Common Good*, PEARC '23, page 173–176, New York, NY, USA, 2023. Association for Computing Machinery.

Code Release. Our experimental code is released and maintained at

https://github.com/PeiYauWeng/PFPT.

A Broader Statement of Impact

This research focuses on developing an effective prompt-tuning and summarizing algorithm for federated learning scenarios featuring a set of clients with diversely skewed local data. The mathematical approaches and insights developed in this paper will help bridge the gap between large, pre-trained models and their applications in private data settings with extremely skewed data distribution. While applications of our work to real data could result in ethical considerations, this is an indirect (and unpredictable) side-effect of our work. Our experimental work uses publicly available datasets to evaluate the performance of our algorithms; no ethical considerations are raised.

B Derivation of Eq. (10)

Note that $\mathbf{z}_t = \{\mathbf{z}_t^i\}_{i=1}^n$ which is a set of independent random vector given $\boldsymbol{\varphi}$. Hence,

$$\mathbb{P}\left(\mathbf{z}_t \mid \boldsymbol{\varphi}\right) \;=\; \prod_{i=1}^{n} \mathbb{P}\left(\mathbf{z}_t^i \mid \boldsymbol{\varphi}\right) \;=\; \prod_{i=1}^{n}\prod_{k=1}^{n_t} \mathbb{P}\left(z_{tk}^i \mid \boldsymbol{\varphi}\right) \tag{16}$$

where the last equality follows from the fact that $\mathbf{z}_t^i = (z_{t1}^i, z_{t2}^i, \ldots, z_{tn_t}^i)$ which is in turn a set of independent Bernoulli variables $z_{tk}^i \in \{0,1\}$ indicating whether the local prompt $\boldsymbol{\omega}_{tk}$ was sampled from a Gaussian centered at the i-th summarizing prompt ϕ_i. Furthermore, following the Bernoulli process definition in Eq. (6), the i-th summarizing prompt ϕ_i is selected independently with probability $\sigma(g(\phi_i;\mathbf{w}))$ to sample the k-th local prompt $\boldsymbol{\omega}_{tk}$ via setting $\boldsymbol{\psi}_{tk} = \phi_i$ in Eq. (4). Hence,

$$\mathbb{P}\left(z_{tk}^i \mid \boldsymbol{\varphi}\right) \;=\; \sigma\Big(g(\phi_i;\mathbf{w})\Big)^{z_{tk}^i} \cdot \Big(1 - \sigma\Big(g(\phi_i;\mathbf{w})\Big)\Big)^{1-z_{tk}^i} \tag{17}$$

Plugging Eq. (17) into Eq. (16) and using $c_{ti} = z_{t1}^i + z_{t2}^i + \ldots + z_{tn_t}^i$ result in Eq. (10).

C Proof of Lemmas 3.2 and 3.3

Lemma 3.2. To derive the result of Lemma 3.2, note that Eq. (9) implies

$$\begin{aligned}
\log \mathbb{P}\left(\boldsymbol{\omega}_t \mid \mathbf{z}_t, \boldsymbol{\varphi}\right) &= \sum_{k=1}^{n_t} \log \mathbb{N}\left(\boldsymbol{\omega}_{tk} \,\middle|\, \sum_{i=1}^{n} z_{tk}^i \cdot \boldsymbol{\phi}_i, \mathrm{diag}\left(\alpha\left(\sum_{i=1}^{n} z_{tk}^i \cdot \boldsymbol{\phi}_i; \boldsymbol{\gamma}\right)\right)\right) \\
&= \sum_{k=1}^{n_t} g_k\left(\sum_{i=1}^{n} z_{tk}^i \cdot \boldsymbol{\phi}_i\right) \ ,
\end{aligned} \tag{18}$$

where we define

$$g_k\left(\mathbf{r}\right) \;\triangleq\; \log \mathbb{N}\left(\boldsymbol{\omega}_{tk} \,\middle|\, \mathbf{r}, \mathrm{diag}\Big(\mathbf{r}; \boldsymbol{\gamma}\Big)\right) \ . \tag{19}$$

In addition, since $\sum_i z_{tk}^i = 1$ with $z_{tk}^i \in \{0,1\}$, Lemma 3.1 implies

$$g_k\left(\sum_{i=1}^{n} z_{tk}^i \cdot \boldsymbol{\phi}_i\right) \;=\; \sum_{i=1}^{n} z_{tk}^i \cdot g_k(\boldsymbol{\phi}_i) \ , \tag{20}$$

which can be plugged into Eq. (18) to arrive at

$$\log \mathbb{P}\left(\boldsymbol{\omega}_t \mid \mathbf{z}_t, \boldsymbol{\varphi}\right) \;=\; \sum_{k=1}^{n_t} g_k\left(\sum_{i=1}^{n} z_{tk}^i \cdot \boldsymbol{\phi}_i\right) \;=\; \sum_{k=1}^{n_t}\sum_{i=1}^{n} z_{tk}^i \cdot g_k(\boldsymbol{\phi}_i) \tag{21}$$

$$\;=\; \sum_{k=1}^{n_t}\sum_{i=1}^{n} z_{tk}^i \log \mathbb{N}\left(\boldsymbol{\omega}_{tk} \,\middle|\, \boldsymbol{\phi}_i, \mathrm{diag}\left(\boldsymbol{\phi}_i; \boldsymbol{\gamma}\right)\right) \ , \tag{22}$$

where the last equality follows from the definition of g_k in Eq. 20. Finally, taking summation over $t = 1, 2, \ldots, n$ on both sides of Eq. (22) and using the definition of $\mathfrak{L}_1$, we arrive at Lemma 3.2.

Lemma 3.3. To derive the result of Lemma 3.3, we start with taking log on both sides of Eq. (10), which results in

$$\begin{aligned}
\log \mathbb{P}\left(\mathbf{z}_t \mid \boldsymbol{\varphi}\right) \;&=\; \sum_{i=1}^{n} c_{ti} \cdot \log \sigma\Big(g\left(\boldsymbol{\phi}_i; \mathbf{w}\right)\Big) + \sum_{i=1}^{n} \Big(1 - c_{ti}\Big) \cdot \log \Big(1 - \sigma\Big(g\left(\boldsymbol{\phi}_i; \mathbf{w}\right)\Big)\Big) && (23)\\
&=\; \sum_{i=1}^{n} c_{ti} \cdot \log \left(\frac{\sigma\Big(g\left(\boldsymbol{\phi}_i; \mathbf{w}\right)\Big)}{1 - \sigma\Big(g\left(\boldsymbol{\phi}_i; \mathbf{w}\right)\Big)}\right) + \sum_{i=1}^{n} \log \Big(1 - \sigma\Big(g\left(\boldsymbol{\phi}_i; \mathbf{w}\right)\Big)\Big) && (24)\\
&=\; \sum_{i=1}^{n} \sum_{k=1}^{n_t} z_{tk}^i \cdot \log \left(\frac{\sigma\Big(g\left(\boldsymbol{\phi}_i; \mathbf{w}\right)\Big)}{1 - \sigma\Big(g\left(\boldsymbol{\phi}_i; \mathbf{w}\right)\Big)}\right) + \sum_{i=1}^{n} \log \Big(1 - \sigma\Big(g\left(\boldsymbol{\phi}_i; \mathbf{w}\right)\Big)\Big) && (25)
\end{aligned}$$

where the last equality is due to the fact that $c_{ti} = z_{t1}^i + z_{t2}^i + \ldots + z_{tn_t}^i$. Hence, taking the summation over $t = 1, 2, \ldots, m$ on both sides of Eq. (25) and using the definition of $\mathfrak{L}_2$, we arrive at Lemma 3.3.

D Optimizing Eq. (15) via Iterative Weighted Bipartite Matching

For any given index t, Eq. (15) can be rewritten as

$$\begin{aligned}
F\Big(\mathbf{z}_t; \mathbf{z}_{-t}\Big) \;=\; \sum_{i=1}^{n} \sum_{k=1}^{n_t} z_{tk}^i \cdot \left(\log \mathbb{N}\left(\boldsymbol{\omega}_{tk} \;\middle|\; \boldsymbol{\phi}_i, \operatorname{diag}\Big(\alpha\Big(\boldsymbol{\phi}_i; \boldsymbol{\gamma}\Big)\Big)\right) + \log \left(\frac{\sigma\Big(g\left(\boldsymbol{\phi}_i; \mathbf{w}\right)\Big)}{1 - \sigma\Big(g\left(\boldsymbol{\phi}_i; \mathbf{w}\right)\Big)}\right)\right) \\
+\; G(\mathbf{z}_{-t}) \;=\; \sum_{i=1}^{n} \sum_{k=1}^{n_t} z_{tk}^i \cdot C_{tk}^i \;+\; G(\mathbf{z}_{-t}) \qquad (26)
\end{aligned}$$

where $G(\mathbf{z}_{-t})$ denotes the summation over the remaining terms in the definition of L_1 and L_2 that do not depend on $\mathbf{z}_t$, and

$$C_{tk}^i \;\triangleq\; \log \mathbb{N}\left(\boldsymbol{\omega}_{tk} \;\middle|\; \boldsymbol{\phi}_i, \operatorname{diag}\Big(\alpha\Big(\boldsymbol{\phi}_i; \boldsymbol{\gamma}\Big)\Big)\right) \;+\; \log \left(\frac{\sigma\Big(g\left(\boldsymbol{\phi}_i; \mathbf{w}\right)\Big)}{1 - \sigma\Big(g\left(\boldsymbol{\phi}_i; \mathbf{w}\right)\Big)}\right) . \qquad (27)$$

Thus, treating all but $\mathbf{z}_t$ as constants, we can solve for the optimal value of $\mathbf{z}_{-t}$ via solving the corresponding weighted bipartite matching. As we iterate over t, optimizing for $\mathbf{z}_t$ one at a time, this results in an iterative weighted bipartite matching approach, which is described below:

Algorithm 2 Iterative Weighted Bipartite Matching

```
input: generative parameters $\boldsymbol{\gamma}$, $\mathbf{w}$, $\boldsymbol{\omega}$ and $\boldsymbol{\phi}$
output: optimized set of values for $\mathbf{z}$
initialize $\mathbf{z}$ randomly
for $t = 1$ to $m$ do
    compute the cost matrix $C_{tk}^i$ using Eq. (27) and fixing $\mathbf{z}_{-t}$ as constant
    $\mathbf{z}_t \leftarrow \text{argmax}_{\mathbf{z}_t} F(\mathbf{z}_t; \mathbf{z}_{-t})$ // solve the weighted bipartite matching algorithm in Eq. (27)
end for
return the optimal set of alignment $\mathbf{z}$
```

E Additional Ablation Studies

Fig. 5 shows the learning trajectories of the summarizing prompts over 120 communication iterations of our method PFPT and GMM-PT, which is the vanilla baseline for modeling prompt alignment via clustering. Overall, the visual plots show that the learning trajectories of GMM-PT appear less

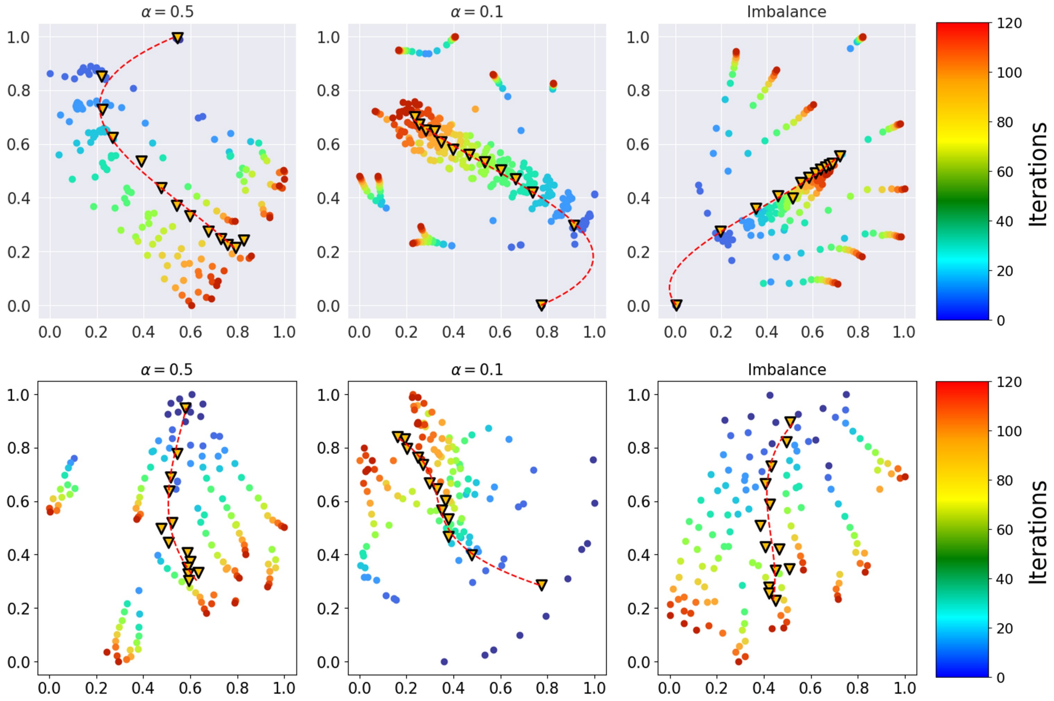

Figure 5: t-SNE plots of the (learned) summarizing prompts of (top) our method PFPT and (bottom) GMM-PT over 120 communication iterations with different heterogeneity settings. Yellow triangles denote the centroids of the t-SNE embeddings of the prompts. The dashed red lines visualize the trajectories of the centroids. The figure is best viewed with color.

diverse than those of PFPT, hinting that the better performance of PFPT can be attributed to its more diverse exploration over the prompt space, allowing it to contextualize information better into its prompt set. This appears consistent with the reported performance gap between GMM-PT and PFPT across a variety of datasets (see Table 1-Table 4 in the main text).

F Additional Experiment Results

In addition to the results reported in the main text, we have also conducted extra experiments comparing the performance of PFPT (ours) with a more challenging set of baselines. These are the improved variants of the original baselines we used in the main text experiment. Each baseline is now improved with a client-specific prompt selection mechanism [49], which enables each client to selectively contextualize its local data into a subset of most relevant prompts. This prevents different information contexts from being collapsed into the same prompt. This modification improves the performance of the baselines substantially on the diverse 4-dataset (Table 6) while preserving their performance on the TinyImageNet dataset (Table 7). However, it is observed that even in this more challenging setting, our PFPT still outperforms all baselines on both datasets substantially, featuring an accuracy improvement up to 4.52%. Similar to our observations in the main text, the improvement is most pronouncing when client data becomes more diverse, which is the case with the 4-dataset.

Remark. We also consider this improvement to be a part of this work's contribution (although it is not the main contribution) since the incorporation of the prompt selection mechanism in [49] has also never been investigated in the existing literature of federated fine-tuning. The comparison results in this section also helps enrich our ablation studies of the key component of PFPT.

We also run experiments to compare the performance of our PFPT algorithm against another set of federated fine-tuning baselines that incorporate adapter-tuning approach into the FedAvg and FedProx backbones. The results are reported in Table 8, which shows that the performance achieved by incorporating the adapter approach to fine-tuning in FedAvg and FedProx is only comparable

Table 6: Accuracy (%) achieved on our synthetic 4-dataset by our proposed PFPT algorithm and other baselines improved with a similar client-specific prompt selection mechanism. The symbol + indicates the improved variant of the original baseline.

	FEDAVG-PT+	FEDPROX-PT+	FEDOPT-PT+	GMM-PT+	PFPT
$\alpha = 0.5$	75.92	76.85	57.37	74.03	**76.89**
$\alpha = 0.1$	66.21	67.77	61.97	66.47	**70.29**
imbalance	61.00	61.75	57.15	61.66	**62.23**

Table 7: Accuracy (%) achieved on the TinyImageNet dataset by our proposed PFPT algorithm and other baselines improved with a similar client-specific prompt selection mechanism. The symbol + indicates the improved variant of the original baseline.

	FEDAVG-PT+	FEDPROX-PT+	FEDOPT-PT+	GMM-PT+	PFPT
$\alpha = 0.5$	86.30	85.78	60.52	84.71	**86.91**
$\alpha = 0.1$	77.79	75.46	64.60	76.35	**82.31**
imbalance	74.25	74.15	64.24	74.17	**78.21**

to that of PFPT on CIFAR10. Since FedAvg and FedProx were consistently the best performing baselines (after PFPT) across all datasets, we believe that the comparison of FedAvg and FedProx configured with adapter-tuning is sufficient to demonstrate the advantage of PFPT over potential adapter-based FL approaches.

Table 8: Accuracy (%) achieved on the all dataset by FEDAVG with Adapter-Tuning, FEDPROX with Adapter-Tuning, and our proposed PFPT algorithm.

Method	Setting	CIFAR10	CIFAR100	TinyImageNet	synthetic 4-datasets
FEDAVG-Adapter	$\alpha = 0.5$	93.86±0.17	75.95±0.40	78.88±0.23	55.55±0.79
	$\alpha = 0.1$	92.66±0.26	65.04±0.68	57.62±0.80	30.58±4.67
	imbalance	**92.33±0.26**	49.8±0.79	40.90±1.34	10.86±8.94
FEDPROX-Adapter	$\alpha = 0.5$	93.69±0.20	75.75±0.16	79.01±0.56	58.39±1.15
	$\alpha = 0.1$	93.04±0.33	64.59±0.82	58.62±0.56	32.92±1.34
	imbalance	92.13±0.13	50.75±1.71	37.65±2.19	13.19±7.92
PFPT	$\alpha = 0.5$	**94.39±0.51**	**80.24±0.24**	**86.91±0.14**	**76.89±0.17**
	$\alpha = 0.1$	**93.39±0.22**	**75.08±0.51**	**82.31±0.26**	**70.29±0.32**
	imbalance	91.45±0.08	**72.05±0.93**	**78.21±1.25**	**62.23±1.02**

G Hyperparameter Settings

All experimented baselines use a batch size of 16 and are fine-tuned with 10 learnable prompts. All experiments are performed on a V100 GPU with 32GB GPU RAM. We employ the same hyperparameter setting in all baselines (except for SCAFFOLD-PT). The hyperparameter settings are all presented in Table 9. SCAFFOLD-PT requires a larger learning rate since SCAFFOLD's variate control mechanism is designed to work with SGD. Hence, unlike other baselines, SCAFFOLD-PT is configured with SGD instead of Adam. Empirically, we find that a larger learning rate is required for SCAFFOLD-PT with SGD to reach its best performance in our setting.

H Limitations

Despite the clear advantage of PFPT in terms of communication bandwidth, as opposed to full-model FT and other forms of FL that typically have to communicate the entire architecture weights per round of communication, we foresee that PFPT will face some challenges when the pre-trained and downstream task domains differ (which goes even beyond the distribution shift setting that we investigate in this paper). Intuitively, a significant domain shift will likely require using more trainable prompts. Can we quantify this shift and bound the minimum number of prompts required to enable accurate adaptation? Although this question is beyond the scope of this paper, we will investigate

Table 9: Hyperparameter setting for all baselines and our PFPT

Method	Setting	Batch size	Communication round	Eps. in local training	Optimizer & learning rate	Total clients	Sampled clients
FEDAVG-PT FEDPROX-PT FEDAVG-Adapter FEDPROX-Adapter FEDOPT-PT PFEDPG-PT GMM-PT PFPT	$\alpha = 0.5$	16	120	5	Adam: beta=(0.9, 0.98), eps=1e-6 lr: 5e-4	CIFAR10: 100 CIFAR100: 100 TinyImageNet: 100 Synthetic 4-dataset: 80	10
	$\alpha = 0.1$	16	120	5	Adam: beta=(0.9, 0.98), eps=1e-6 lr: 1e-4	CIFAR10: 100 CIFAR100: 100 TinyImageNet: 100 Synthetic 4-dataset: 80	10
	imbalance	16	120	5	Adam: beta=(0.9, 0.98), eps=1e-6 lr: 1e-4	CIFAR10: 100 CIFAR100: 100 TinyImageNet: 100 Synthetic 4-dataset: 80	10
SCAFFOLD-PT	$\alpha = 0.5$	16	200	5	SGD lr: 5e-4	CIFAR10: 100 CIFAR100: 100 TinyImageNet: 100 Synthetic 4-dataset: 80	10
	$\alpha = 0.1$	16	200	5	SGD lr: 1e-4	CIFAR10: 100 CIFAR100: 100 TinyImageNet: 100 Synthetic 4-dataset: 80	10
	imbalance	16	200	5	SGD lr: 1e-4	CIFAR10: 100 CIFAR100: 100 TinyImageNet: 100 Synthetic 4-dataset: 80	10

and address it in future works, especially in the context of heterogeneous FL where each client operates in a different domain. We have also raised the issue of prompt tuning having to compute all intermediate gradients since the prompt tokes are appended at the top level of the pre-trained model. Going forward, we can explore new prompt placement strategies to reduce this memory consumption.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Our contributions can be found in Sections 3 and 4

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Please refer to Appendix H

   Guidelines:

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

3. **Theory Assumptions and Proofs**

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

   Answer: [Yes]

Justification: Our results mostly establish useful probabilistic equalities which do not require any assumptions. Complete proofs can be found in Appendices B, C, and D.

Guidelines:

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

4. **Experimental Result Reproducibility**

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

   Answer: [Yes]

   Justification: Our experiment protocols are described in Section 4. The detailed hyperparameter settings are fully specified in Appendix G.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: All the used datasets are publicly available. Upon acceptance, we will release our code.

Guidelines:

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

6. **Experimental Setting/Details**

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

Answer: [Yes]

Justification: Such details can be found in Section 4 and Appendix G

Guidelines:

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

7. **Experiment Statistical Significance**

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

Answer: [Yes]

Justification: We have provided error bars for all reported results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

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

8. **Experiments Compute Resources**

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

   Answer: [Yes]

   Justification: The detail regarding our compute resources are provided in Appendix G

   Guidelines:

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

9. **Code Of Ethics**

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

   Answer: [Yes]

   Justification: We have read the NeurIPS Code of Ethics and do not find our work violate any aspects of the code

   Guidelines:

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

10. **Broader Impacts**

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

    Answer: [Yes]

    Justification: We provide a statement of impact in Appendix A.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

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

11. **Safeguards**

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

    Answer: [NA]

    Justification: Our work do not create new dataset. We only use existing, publicly available datasets and their combination. We also do not create any new pre-trained NLP or vision models. We only used existing, publicly available pre-trained models.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We cite the source of all datasets and pre-trained models used in our experiments.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

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

    Answer: [NA]

    Justification: Our work does not release any new assets.

    Guidelines:

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

14. **Crowdsourcing and Research with Human Subjects**

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

    Answer: [NA]

    Justification: Our work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

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

    Answer: [NA]

    Justification: Our work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.