BLEND: Balancing Personalization vs. Generalization in Federated Vision–Language Models

Payam Abdisarabshali
University at Buffalo–SUNY
payamabd@buffalo.edu

Fardis Nadimi
University at Buffalo–SUNY
Fardisna@buffalo.edu

Kasra Borazjani
University at Buffalo–SUNY
Kasrabor@buffalo.edu

Naji Khosravan
Adobe
naji.khosravan@gmail.com

Richeng Jin
Zhejiang University
richengjin@zju.edu.cn

Seyyedali Hosseinalipour
University at Buffalo–SUNY
alipour@buffalo.edu

Abstract

Federated vision–language models (FedVLMs) represent one of the emerging frontiers of machine learning, aiming to integrate federated learning within the fine-tuning pipelines of vision–language models (VLMs). In this work, we introduce BLEND, a new framework designed to jointly enhance the personalization and generalization capabilities of FedVLMs. BLEND proposes a selective VLM parameter fine-tuning and aggregation strategy that consists of (i) global vision and text adapters shared across clients, and (ii) a personalized vision projection adapter tailored to each client. BLEND further introduces a novel loss function that *blends* the outputs of the global and personalized adapters to achieve personalization, while simultaneously promoting generalization through a principled anchor design implemented via the projection heads of the zero-shot VLM. Extensive experiments demonstrate that BLEND consistently outperforms state-of-the-art baselines, highlighting its ability to effectively balance personalization and generalization across diverse system settings. (Code is available at: https://github.com/payamsiabd/BLEND.git)

1 Introduction

Vision–language models (VLMs) such as *contrastive language-image pretraining* (CLIP) [1] have rapidly become a major part of modern intelligence [2, 3]. In many applications, the data that fuel VLMs are inherently *distributed* across numerous entities, which necessitates developing frameworks that enable the *collaborative fine-tuning of VLMs across distributed clients*. This has given rise to a rapidly emerging research area, termed *federated vision–language models (FedVLMs)* [4–9].

In this work, we focus on *balancing model generalization vs. personalization in FedVLMs*, a highly recent field with only a few relevant studies [5, 10]. Our approach is built upon revisiting a powerful observation made in [11], a work dedicated to *centralized* fine-tuning of VLMs. In particular, revisiting *Fig. 1 in [11]*, we obtain two intuitions, which point to an effective approach for balancing model generalization vs. personalization in the *distributed* fine-tuning setting of FedVLMs:

(Intuition 1) The *vision* branch of CLIP yields lower dataset-level recognition performance compared to the text branch, indicating that it captures more generalizable representations. However, such generality lacks client-specific adaptation, necessitating personalization to client data in FedVLMs.

(Intuition 2) Compared to the vision branch, the *text* branch of CLIP exhibits a higher degree of dataset-level recognition, and thus further personalization may result in overfitting in FedVLMs.

Based on the above intuitions (concretized in Sec. 3.2.2), we formalize our research question below:

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

How can we design adaptation mechanisms for the vision/text branches of CLIP in FedVLM settings so as to preserve the generalizable representations of the vision branch, avoid over-personalization in the text branch, and jointly enable effective client-specific personalization?

To address this, we systematically (i) determine where to apply model personalization, (ii) decide which components should be aggregated to capture global knowledge and which should remain local to capture client-specific adaptations, and (iii) obtain a carefully crafted loss function that can leverage such local-global model fine-tuning decomposition in FedVLMs to balance model generalization vs. personalization. In doing so, we modify the CLIP architecture to produce *three complementary sources of representations*. In the vision branch, we generate a *generalized vision representation* and a *personalized vision representation*, while in the text branch, we maintain a *generalized text representation*. To effectively combine these representations, our introduced dual-objective training entails a *fusion loss* and an *anchor loss*. This design enables the model to *blend* the generalized and personalized vision representations into a fusion representation that captures both general and client-specific knowledge, while simultaneously preserving the generalized knowledge of CLIP encoded in its text branch.

Compared to a few comparable works [5, 10] and other FedVLM works that are mostly focused on sole model personalization [4, 6, 2, 3] and use techniques such as optimal transport and singular value decomposition, an advantage of our method is its *simplicity*: it offers a systematic global-local decomposition of CLIP fine-tuning elements combined with a modified loss function that effectively utilizes this decomposition and outperforms all baseline methods. Our contributions are as follows:

- We introduce BLEND, a systematically designed framework to balance personalization and generalization in FedVLMs. BLEND is a fine-tuning strategy that augments CLIP with trainable modules: global vision and text adapters, and a personalized vision projection adapter.
- BLEND establishes a selective aggregation scheme, where only the global adapters are aggregated across clients while the personalized projection adapter remains local. It further introduces a novel dual loss function that is designed to leverage the output of these adapters to ensure a balanced model personalization vs. generalization during distributed model fine-tuning in FedVLMs.
- Extensive experiments confirm that BLEND consistently outperforms state-of-the-art FedVLM baselines, achieving superior balance between model personalization and generalization.

$\star$ Following the comparable works [3, 2, 6, 5, 4, 12], we adopt CLIP as the deployed VLM.

2 Related Work

2.1 Centralized VLM Fine-Tuning Techniques

Works on the fine-tuning of VLMs can be broadly categorized into *(i) prompt-based learning techniques* and *(ii) adapter-based learning techniques*. On the former front, early works such as CoOp [13] introduced learnable text prompts, while subsequent studies, such as ProVP [14], MetaPrompt [15], TCP [16], MoP [17], and DPC [18] focused on improving prompt optimization. Also, researchers have explored Bayesian regularization [19], margin-based tuning [20], test-time adaptation [21], integrating LLMs for dynamic prompt routing [22], self-distillation for zero-shot generalization [23], and semantic-guided prompt optimization [24, 25]. Overall, while prompt-learning serves as a highly lightweight adaptation technique for VLMs, it often suffers from limited performance compared to more expressive adaptation approaches such as adapter-based learning.

Adapter-based learning introduces lightweight trainable modules into transformer architectures to capture task-specific information while keeping most parameters frozen. Early works such as Tip-Adapter [26] and CLIP-Adapter [27] proposed leveraging cached feature representations and multi-layer perceptron adapters within the CLIP backbone for few-shot classification. Recently, MMA [11] developed multi-modal adapters for CLIP that jointly refine vision–language alignment, which as a byproduct helps balance generalization and personalization in centralized VLMs.

Compared to the aforementioned studies that focus on *centralized* fine-tuning strategies, BLEND targets an underexplored research direction: the joint design of parameter-efficient adapters, architectural adjustments to CLIP, and a novel loss function that explicitly balances model personalization and generalization in *distributed/federated* fine-tuning settings for VLMs.

2.2 Federated Vision-Language Models (FedVLMs)

Recent works on FedVLMs such as PromptFL [6] and FedOTP [4] introduced federated prompt optimization for VLMs, which was followed by PromptFolio [28], FedPGP [5], FedPHA [3], and

pFedMoAP [2]. In parallel, works such as FedDat [29] and TriplePlay [30] introduced adapter-based fine-tuning of VLMs to distributed clients with non-independent and identically distributed (non-IID) data, which was advanced by pFedMMA [10]. Further, PromptFolio [28] proposed fusing global and local text *prompts* for FedVLMs. Among these studies, FedPGP, pFedMMA, and PromptFolio are among the very few that aim to improve both personalization and generalization in FedVLMs.

Compared to the above works, BLEND provides a systematic design explicitly tailored to balance generalization and personalization in FedVLMs. This objective is directly encoded in its architectural modifications to CLIP, its local training and aggregation procedures, and its loss formulation. Extensive experiments demonstrate that BLEND consistently outperforms all existing relevant baselines, establishing a new state-of-the-art in the FedVLM literature.

3 Method

3.1 Background: CLIP in a Nutshell

CLIP [1] adopts a dual-encoder architecture consisting of a *vision encoder* $\mathcal{V}$ and a *text encoder* $\mathcal{T}$, each implemented as a stack of L transformers. The model is *pre-trained* on large numbers of image–text pairs using a contrastive learning objective, which is often followed by a *fine-tuning* phase on downstream tasks. Below, we provide a background on CLIP's components.

Vision encoder: In the vision encoder $\mathcal{V}$, an input image $\mathbf{I}$ is first partitioned into multiple patches $\mathsf{Patch}(\mathbf{I})$. These patches are then projected into feature embeddings and concatenated with a learnable token ζ_0, referred to as the *classification token*, to form the initial feature matrix $\boldsymbol{x}_0$ as follows:

$$\boldsymbol{x}_0 = [\zeta_0, \mathsf{Patch}(\mathbf{I})]. \tag{1}$$

Feature matrix $\boldsymbol{x}_0$ is then sequentially processed by L transformer blocks of the vision encoder, referred to by $\{\mathcal{V}_\ell\}_{\ell=1}^{L}$, during which the subsequent classification tokens $\{\zeta_\ell\}_{\ell=1}^{L}$ aggregate visual information from all other tokens. Mathematically, the output of each of the subsequent transformer blocks $\ell \in \{1, 2, \cdots, L\}$, denoted by $\boldsymbol{x}_\ell$, can be presented as the concatenation of a classification token ζ_ℓ along with the nominal outputs of the transformer block o_ℓ as follows:

$$\boldsymbol{x}_\ell \triangleq [\zeta_\ell, o_\ell] = \mathcal{V}_\ell(\boldsymbol{x}_{\ell-1}), \qquad \ell \in \{1, 2, \cdots, L\}. \tag{2}$$

After the final transformer $\mathcal{V}_L$, the classification token ζ_L is projected into a shared vision–language embedding space using the *vision projection head* π^{Vis}, producing the final visual representation as:

$$\mathcal{R} = \zeta_L \pi^{\mathsf{Vis}}. \tag{3}$$

Text encoder: The text encoder follows a similar transformer-based architecture to the vision encoder but operates on tokenized text sequences. Specifically, for a classification task with C distinct classes, each class label $c \in \mathcal{C} \triangleq \{1, \cdots, C\}$ is first represented with a *class description* $T^{(c)}$ using a predefined prompt template, e.g., $T^{(c)}$=`an image of a {class_label}`. Each $T^{(c)}$ is then tokenized into multiple tokens via a tokenizer as:

$$\boldsymbol{w}_0^{(c)} = \mathsf{Tokenizer}\big(T^{(c)}\big). \tag{4}$$

The initial token embeddings $\boldsymbol{w}_0^{(c)}$ are subsequently passed through L transformer blocks of the text encoder, referred to by $\{\mathcal{T}_\ell\}_{\ell=1}^{L}$, to produce the subsequent text token embeddings $\{\boldsymbol{w}_\ell^{(c)}\}_{\ell=1}^{L}$, where each $\boldsymbol{w}_\ell^{(c)}$ can be presented as the concatenation of the nominal outputs of the transformer block $\hat{o}_\ell$ and the *end-of-text (EoT)* token ξ_ℓ as follows:

$$\boldsymbol{w}_\ell^{(c)} \triangleq [\hat{o}_\ell, \xi_\ell] = \mathcal{T}_\ell\big(\boldsymbol{w}_{\ell-1}^{(c)}\big), \ \ell \in \{1, 2, \cdots, L\}. \tag{5}$$

Specifically, the last token of each embedding $\boldsymbol{w}_\ell^{(c)}$, i.e., the EoT token ξ_ℓ, captures the contextual dependencies across the entire sequence. After the final transformer block $\mathcal{T}_L$, the resulting EoT token of $\boldsymbol{w}_L^{(c)}$, referred to as $\xi_{_L}$, is projected into the shared vision–language embedding space via the *text projection head* π^{Txt} to produce the final text feature $\mathcal{W}^{(c)}$ as follows:

$$\mathcal{W}^{(c)} = \xi_{_L} \pi^{\mathsf{Txt}}. \tag{6}$$

Classification in CLIP: To perform a prediction, given the visual embedding $\mathcal{R}$ defined in (3) and the textual embeddings of classes $\{\mathcal{W}^{(c)}\}_{c\in\mathcal{C}}$ defined in (6), the similarity between the image and each class is computed using the cosine similarity (i.e., normalized inner product) as follows:

$$S^{(c)} = \langle \mathcal{R}, \mathcal{W}^{(c)} \rangle / \|R\|_{_2} . \|\mathcal{W}^{(c)}\|_{_2}, \quad c \in \mathcal{C}. \tag{7}$$

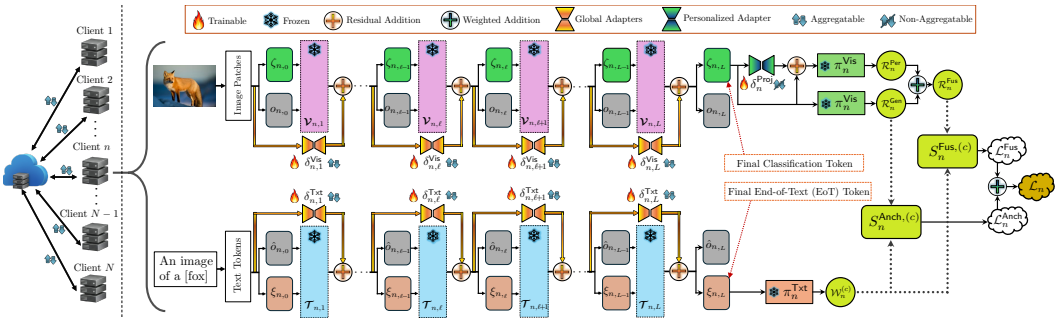


Figure 1: **Left:** A FedVLM system with N clients and a central server. **Right:** Schematic of the model introduced by BLEND for each client n, encompassing (i) a set of trainable and aggregatable global vision and text adapters, which are residually added to the transformer layers (i.e., $\{\delta^{\mathsf{Vis}}_{n,\ell}, \delta^{\mathsf{Txt}}_{n,\ell}\}_{\ell=1}^{L}$), (ii) a trainable personalized vision projection adapter, which is integrated in a new branch and kept local during the training (i.e., $\delta^{\mathsf{Proj}}_{n}$), and (iii) dual loss functions (i.e., $\mathcal{L}^{\mathsf{Fus}}_{n}$ and $\mathcal{L}^{\mathsf{Anch}}_{n}$), which are integrated into the final loss $\mathcal{L}_n$.

The set of similarities $\{S^{(c)}\}_{c\in\mathcal{C}}$ is then used to compute the class probabilities for the input image $\mathbf{I}$ via a softmax function with temperature τ as: $\mathsf{Pr}(c \mid \mathbf{I}) = \exp(S^{(c)}/\tau)/\sum_{c'\in\mathcal{C}} \exp(S^{(c')}/\tau)$.

3.2 Proposed Method (BLEND)

We consider a FedVLM system, comprising a set of clients $\mathcal{N} = \{1, \ldots, N\}$, where $|\mathcal{N}| = N$. Each client $n \in \mathcal{N}$ possesses a local dataset $\mathcal{D}_n$, and maintains a local copy of the CLIP model. The goal of our proposed method, i.e., BLEND, is to achieve two potentially competing objectives: (i) enhanced *personalization*, by adapting the local CLIP model to each client's data through fine-tuning, and (ii) enhanced *generalization*, by improving the local CLIP model's ability to recognize previously unseen classes/labels (i.e., those that are not part of clients' data) that may be encountered during deployment. A schematic of BLEND is depicted in Fig. 1 and its operations are summarized below.

3.2.1 BLEND in a Nutshell

BLEND establishes a federated fine-tuning methodology for CLIP based on ***Intuitions 1&2*** presented in Sec. 1. Specifically, BLEND employs two learnable components, each implementing a distinct but complementary mechanism in the learning process to achieve the aforementioned two competing objectives. The first component is a set of adapters (called *global vision/text adapters*), aggregated across clients, that are designed to learn *generalizable representations* through collaborative learning across clients, while the second component is an adapter (called *personalized projection adapter*), kept local at each client, that is crafted to learn *client-specific personalized representations*. These two components are then systematically used in a novel loss function to balance personalization and generalization. The realization of these two complementary components stems from a two-step design process. First, we inject the global vision and text adapters into the transformer blocks of both the vision and text encoders of CLIP. These adapters are integrated into the transformer pipelines, producing a vision output that is passed through the pre-trained (frozen) CLIP vision projection head, referred to as the G-head, and a text output that is processed by the pre-trained (frozen) CLIP text projection head. These projection heads are intentionally kept frozen to preserve the generalizability of the pre-trained CLIP model while accompanying the learnable global adapters. We then enable the local training of the global adapters on each client, followed by periodic aggregation at the server during federated training. As shown in Fig. 2 (blue square), compared to the pre-trained CLIP model, incorporating the global adapters improves generalization to unseen classes by capturing cross-device knowledge. However, while these adapters also yield gains in personalization, we hypothesize that there remains room for improvement, since global adapters alone cannot fully preserve client-specific representations due to their periodic aggregation.

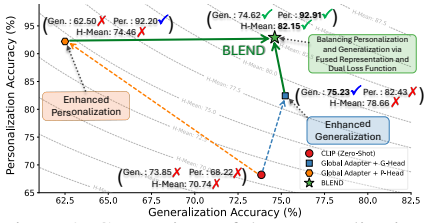


Figure 2: Comparison of the personalization (Per.) and generalization (Gen.) of various fine-tuning strategies along with their corresponding harmonic mean (H-Mean). Results are averaged over the 10 datasets in Sec. 4, where the system configuration is explained.

To address this limitation and enhance personalization, BLEND injects a personalized projection adapter into the vision projection head of the pre-trained CLIP, referred to as the P-head. This adapter module is learnable but is not transmitted to the server and remains on the client to learn personalized

representations. As shown in Fig. 2 (orange circle), when this module is used in conjunction with the global adapters, it can significantly improve model personalization. Nevertheless, the results show that relying solely on this module leads to a significant degradation in the model's generalization.

Specifically, given the individual generalization and personalization gains achieved using the global adapters with the G-head and the P-head, respectively, the remaining challenge is as follows: *how to integrate all of these components into a unified model architecture to take advantage of their complementary benefits?* BLEND achieves this goal through fusing the outputs of G-head and P-head to construct a combined representation, referred to as the *projected representation fusion*, which is used as the final visual embedding. Afterwards, by defining a novel loss function, called *representation fusion loss*, BLEND enables the model to fuse these two representations in an optimal manner to balance personalization and generalization of the model. As shown in Fig. 2, BLEND achieves the highest level of personalization while simultaneously maintaining superior generalization, jointly captured by the reported harmonic mean (H-Mean) value. Below, we provide a theoretical motivation for BLEND's asymmetric personalization of the vision and text branches of CLIP.

3.2.2 Theoretical Motivations

BLEND's asymmetric personalization of encoder branches of CLIP is rooted in the mutual information between the input scene and the visual and textual observations as concretized below:

Proposition 3.1. *Let S denote the latent semantic information of a scene, V its visual observation, and T a textual description generated from V. The information flow satisfies the Markov chain $S \to V \to T$, and therefore the mutual information $I(\cdot;\cdot)$ of $(S;V)$ and $(S;T)$ holds in*

$$I(S;T) \leq I(S;V).$$

Proof. The proof follows from Theorem 2.8.1 (Data-Processing Inequality) in [31]. □

The inequality $I(S;T) \leq I(S;V)$ establishes that a textual description generated from an image *cannot* contain more information about the underlying scene than the visual observation from which it is derived. For instance, an image may depict a brown dog running on grass, facing left, with a red ball nearby, while its textual description may simply state "an image of a dog." Thus, the visual observation can preserve multiple semantic attributes that are omitted from the textual description.

The above information-theoretic result can be further complemented with two empirical observations from prior work on centralized fine-tuning of VLMs. First, Schrodi et al. [32] identified an information imbalance in CLIP-like models, showing that textual captions often provide sparse descriptions while images contain richer information. Second, Yang et al. [11] observed that textual representations can exhibit stronger dataset-specific behavior compared to vision representations. Together, these observations further justify ***Intuitions 1&2*** presented in Sec. 1, and subsequently motivate BLEND's asymmetric personalization of CLIP's encoder branches, with stronger client-specific adaptation applied to visual representations (through introducing P-head).

3.2.3 Local Model Architecture

We next focus on each client n and detail the model architecture in BLEND depicted in Fig. 1.

Global vision adapters: Let $\boldsymbol{x}_{n,0} \in \mathbb{R}^{M_x \times d_x}$ be the initial token embedding of an image $\mathbf{I}$ of client n (corresponding to $\boldsymbol{x}_0$ calculated using (1)), where M_x is the number of visual tokens and d_x is the visual embedding dimension. Further, let $\boldsymbol{x}_{n,\ell-1}$ be the input to the ℓ-th vision transformer block at client n denoted by $\mathcal{V}_{n,\ell}$. BLEND establishes an adapter branch in each vision transformer block $\mathcal{V}_{n,\ell}$ whose output is residually added to the block's output as follows:

$$\boldsymbol{x}_{n,\ell} = \underbrace{\mathcal{V}_{n,\ell}\left(\boldsymbol{x}_{n,\ell-1}\right)}_{\text{Pre-Trained Transformer Output}} + \underbrace{\delta^{\mathsf{Vis}}_{n,\ell}\left(\boldsymbol{x}_{n,\ell-1}\right)}_{\text{Residually Added Adapter Output}}, \ \ell \in \{1, \cdots, L\}, \tag{8}$$

where $\mathcal{V}_{n,\ell}(\cdot)$ is the ℓ-th transformer, used in (2), while the residual $\delta^{\mathsf{Vis}}_{n,\ell}$ is realized as follows:

$$\delta^{\mathsf{Vis}}_{n,\ell}\left(\boldsymbol{x}_{n,\ell-1}\right) = \mathbf{B}^{\mathsf{Vis}}_{n,\ell}\Big(\sigma\big(\mathbf{A}^{\mathsf{Vis}}_{n,\ell}(\boldsymbol{x}_{n,\ell-1})\big)\Big). \tag{9}$$

Here, $\mathbf{A}^{\mathsf{Vis}}_{n,\ell} \in \mathbb{R}^{r_x \times d_x}$ and $\mathbf{B}^{\mathsf{Vis}}_{n,\ell} \in \mathbb{R}^{d_x \times r_x}$ are the down- and up-projection matrices, respectively, with $r_x \ll d_x$ denoting the rank of the adaptation (various choices of r_x are studied in *Ablation Study 4* in Sec. 4). Further, σ denotes the non-linear ReLU activation function. In *Ablation Study 6* in Sec. 4, we analyze the performance gains obtained by inserting adapters at different transformer blocks (when no adapter is used at transformer layer ℓ, the residual branch is disabled and $\delta^{\mathsf{Vis}}_{n,\ell}(\cdot) = 0$).

Global text adapters: The text encoder follows the same architectural design as the vision encoder. Specifically, let $\boldsymbol{w}^{(c)}_{n,0} \in \mathbb{R}^{M_w \times d_w}$ denote the initial token embeddings of class description $T^{(c)}$ at

client n (corresponding to $\boldsymbol{w}_0^{(c)}$ calculated using (4)), where M_w is the number of text tokens and d_w is the text embedding dimension. Further, let $\boldsymbol{w}_{n,\ell-1}^{(c)}$ denote the input to the ℓ-th transformer block of the text encoder for client n denoted by $\mathcal{T}_{n,\ell}$. BLEND integrates an adapter branch in each text transformer block ℓ, whose output is residually added to the original block output as:

$$\boldsymbol{w}_{n,\ell}^{(c)} = \mathcal{T}_{n,\ell}\big(\boldsymbol{w}_{n,\ell-1}^{(c)}\big) + \delta_{n,\ell}^{\mathsf{Txt}}\big(\boldsymbol{w}_{n,\ell-1}^{(c)}\big),\ \ell\in\{1,\cdots,L\}, \tag{10}$$

where $\mathcal{T}_{n,\ell}(\cdot)$ is the transformer block, used in (5), while the residual $\delta_{n,\ell}^{\mathsf{Txt}}$ is realized as follows:

$$\delta_{n,\ell}^{\mathsf{Txt}}\big(\boldsymbol{w}_{n,\ell-1}^{(c)}\big) = \mathbf{B}_{n,\ell}^{\mathsf{Txt}}\Big(\sigma\big(\mathbf{A}_{n,\ell}^{\mathsf{Txt}}(\boldsymbol{w}_{n,\ell-1}^{(c)})\big)\Big). \tag{11}$$

Here, $\mathbf{A}_{n,\ell}^{\mathsf{Txt}} \in \mathbb{R}^{r_w\times d_w}$ and $\mathbf{B}_{n,\ell}^{\mathsf{Txt}} \in \mathbb{R}^{d_w\times r_w}$ are the down- and up-projection matrices, respectively, with $r_w \ll d_w$ (the effect of r_w on the performance is studied in *Ablation Study 4* in Sec. 4).

Text projection head: BLEND utilizes the pre-trained text projection head of CLIP $\pi_n^{\mathsf{Txt}} \in \mathbb{R}^{d_w\times d'}$ (the effect of fine-tuning π_n^{Txt} is studied in *Ablation Study 10* in Sec. 4), where d' is the final shared vision-language embedding dimension. Specifically, at each client n holding the subset of labels $\mathcal{C}_n \subseteq \mathcal{C}$, the final projected representation of a class description $T^{(c)}, \forall c \in \mathcal{C}_n$, is computed similarly to (6) based on the latest text transformer output $\boldsymbol{w}_{n,L}^{(c)} = [\hat{o}_{n,L}, \xi_{n,L}]$, obtained in (10), as follows:

$$\mathcal{W}_n^{(c)} = \xi_{n,L}\pi_n^{\mathsf{Txt}}, \quad \mathcal{W}_n^{(c)} \in \mathbb{R}^{d'}, \tag{12}$$

where ξ_{L} is the respective EoT token of $\boldsymbol{w}_{n,L}^{(c)}$ and $\hat{o}_{n,L}$ is the nominal outputs of the transformer.

Two-branch vision projection head: Let $\boldsymbol{x}_{n,L} = [\zeta_{n,L}, o_{n,L}]$, obtained following (8), denote the output of the last vision transformer block of client n, consisting of the classification token $\zeta_{n,L}$ and nominal outputs of the transformer $o_{n,L}$. To enhance both generalization and personalization, BLEND establishes a two-branch vision projection head, including a G-head and a P-head, as follows.

Realization of G-head: G-head maintains the model's generalization and projects the classification token $\zeta_{n,L}$ using the pre-trained CLIP vision projection $\pi_n^{\mathsf{Vis}} \in \mathbb{R}^{d_x\times d'}$ in a similar fashion to (3) as:

$$\mathcal{R}_n^{\mathsf{Gen}} = \zeta_{n,L}\pi_n^{\mathsf{Vis}}, \quad \mathcal{R}_n^{\mathsf{Gen}} \in \mathbb{R}^{d'}. \tag{13}$$

Realization of P-head: P-head introduces a new branch to the vision projection head, containing a learnable *personalized projection adapter* responsible for improving the model's personalization. Specifically, in this branch, the *personalized classification token embedding* $\hat{\zeta}_{n,L}$ is obtained through residually adding the output of the personalized projection adapter to the classification token as:

$$\hat{\zeta}_{n,L} = \zeta_{n,L} + \delta_n^{\mathsf{Proj}}(\zeta_{n,L}), \tag{14}$$

where the adapter output δ_n^{Proj} is defined as:

$$\delta_n^{\mathsf{Proj}}(\zeta_{n,L}) = \mathbf{B}_n^{\mathsf{Proj}}\Big(\sigma\big(\mathbf{A}_n^{\mathsf{Proj}}\big(\zeta_{n,L}\big)\big)\Big). \tag{15}$$

Here, $\mathbf{A}_n^{\mathsf{Proj}}\in\mathbb{R}^{r_p\times d_x}$ and $\mathbf{B}_n^{\mathsf{Proj}}\in\mathbb{R}^{d_x\times r_p}$ are learnable low-rank matrices with $r_p \ll d_x$ (the impact of r_p is studied in *Ablation Study 5* in Sec. 4). The *personalized vision embedding* is then obtained as:

$$\mathcal{R}_n^{\mathsf{Per}} = \hat{\zeta}_{n,L}\pi_n^{\mathsf{Vis}}, \quad \mathcal{R}_n^{\mathsf{Per}} \in \mathbb{R}^{d'}. \tag{16}$$

Fusion of the embeddings of G-head and P-head: To construct the final visual representation, we combine the generalized and personalized embeddings in (13) and (16) via a weighted average as:

$$\mathcal{R}_n^{\mathsf{Fus}} = (1-\gamma)\mathcal{R}_n^{\mathsf{Gen}} + \gamma\mathcal{R}_n^{\mathsf{Per}}, \quad \mathcal{R}_n^{\mathsf{Fus}} \in \mathbb{R}^{d'}, \tag{17}$$

where $\gamma \in [0,1]$ controls the trade-off between generalization and personalization (the impact of γ is studied in *Ablation Study 7* in Sec. 4).

Through introducing a novel loss function, defined in the subsequent section, BLEND constructs a learning paradigm in which all the trainable components (i.e., $\mathbf{A}_{n,\ell}^{\mathsf{Vis}}$, $\mathbf{B}_{n,\ell}^{\mathsf{Vis}}$, $\mathbf{A}_{n,\ell}^{\mathsf{Txt}}$, $\mathbf{B}_{n,\ell}^{\mathsf{Txt}}$, $\mathbf{A}_n^{\mathsf{Proj}}$, $\mathbf{B}_n^{\mathsf{Proj}}$) are realized in such a way to ensure that the produced text representations $\{\mathcal{W}_n^{(c)}\}_{c\in\mathcal{C}_n}$ and the fusion of visual representations $\mathcal{R}_n^{\mathsf{Gen}}$ and $\mathcal{R}_n^{\mathsf{Per}}$ enhance both model personalization and generalization.

3.2.4 Local Model Fine-Tuning

The trainable parameters of each client n are collected by

$$\boldsymbol{\mathcal{P}}_n^{\mathsf{Train}} = \Big[[\mathbf{A}_{n,\ell}^{\mathsf{Vis}}, \mathbf{B}_{n,\ell}^{\mathsf{Vis}}, \mathbf{A}_{n,\ell}^{\mathsf{Txt}}, \mathbf{B}_{n,\ell}^{\mathsf{Txt}}]_{\ell^{\mathsf{S}}\le\ell\le\ell^{\mathsf{E}}}, \mathbf{A}_n^{\mathsf{Proj}}, \mathbf{B}_n^{\mathsf{Proj}}\Big], \tag{18}$$

where ℓ^{S} and ℓ^{E} denote the start and end indices of the transformer layers into which adapters are injected. Federated learning operates through a sequence of global rounds, each indexed by k. At global round k, each client $n \in \mathcal{N}$ fine-tunes its local model through minimizing a local loss function $\mathcal{L}_n(\cdot)$ by performing stochastic gradient descent (SGD) iterations on its trainable parameters using its

local dataset $\mathcal{D}_n$. In particular, at the j-th local SGD iteration ($j = 1, \ldots, J$) of global round k, the parameter update of client n is given by

$$\boldsymbol{\mathcal{P}}_n^{\mathsf{Train},(k,j)} = \boldsymbol{\mathcal{P}}_n^{\mathsf{Train},(k,j-1)} - \eta \nabla \mathcal{L}_n(\boldsymbol{\mathcal{P}}_n^{\mathsf{Train},(k,j-1)}), \tag{19}$$

where η is the learning-rate and the iterations start from $\boldsymbol{\mathcal{P}}_n^{\mathsf{Train},(k,0)}$, which is the latest received global model from the server (described in Sec. 3.2.5). Next, we detail our proposed loss function $\mathcal{L}_n(\cdot)$.

Realization of loss function: Given an input image $\mathbf{I}$, let $\mathcal{R}_n^{\mathsf{Fus}} \in \mathbb{R}^{d'}$ denote the fused representation of the two-branch vision projection head in (17) and $\{\mathcal{W}_n^{(c)}\}_{c \in \mathcal{C}_n}$ denote the text embeddings of all class descriptions obtained from (12). We first compute the similarity logits $\{S_n^{\mathsf{Fus},(c)}\}_{c \in \mathcal{C}_n}$ as:

$$S_n^{\mathsf{Fus},(c)} = \langle \mathcal{R}_n^{\mathsf{Fus}}, \mathcal{W}_n^{(c)} \rangle / \|\mathcal{R}_n^{\mathsf{Fus}}\|_2 . \|\mathcal{W}_n^{(c)}\|_2, \quad c \in \mathcal{C}_n. \tag{20}$$

We then introduce *the representation fusion loss* $\mathcal{L}_n^{\mathsf{Fus}}(\boldsymbol{\mathcal{P}}_n^{\mathsf{Train}})$ as the cross-entropy loss [4] with the following conditional probabilities: $\mathsf{Pr}(c|\mathbf{I}) = \exp\big(S_n^{\mathsf{Fus},(c)}/\tau\big) / \sum_{c' \in \mathcal{C}_n} \exp\big(S_n^{\mathsf{Fus},(c')}/\tau\big)$. The representation fusion loss encourages the optimal fusion of the general representation $\mathcal{R}_n^{\mathsf{Gen}}$ and the personalized representation $\mathcal{R}_n^{\mathsf{Per}}$ introduced in (17) by optimizing $\boldsymbol{\mathcal{P}}_n^{\mathsf{Train}}$.

Although $\mathcal{L}_n^{\mathsf{Fus}}(\boldsymbol{\mathcal{P}}_n^{\mathsf{Train}})$ integrates both the general and personalized representations (i.e., $\mathcal{R}_n^{\mathsf{Gen}}$ and $\mathcal{R}_n^{\mathsf{Per}}$), its sole minimization is *not* enough: minimizing this loss function alone can negatively affect the generalization of the model as it biases both the global vision adapters and personalized projection adapter to the clients' local datasets. This calls for the introduction of a complementary loss function that prevents the bias of the global vision adapters to the local datasets, promoting the generalizability of the model. To this end, we introduce a *running anchor loss* that constrains the general vision representation $\mathcal{R}_n^{\mathsf{Gen}}$ (i.e., the G-head's output) to remain aligned with the corresponding text representation $\mathcal{W}_n^{(c)}$: since both the G-head and the text encoder branch employ the pre-trained (zero-shot) CLIP projection heads, enforcing this alignment establishes a stable anchoring that promotes generalization. The term *'running'* reflects the fact that this anchor is not fixed; rather, it evolves as the global vision and text adapters update the underlying representations in both branches.

To realize the running anchor loss, we first obtain the similarity logits between $\mathcal{R}_n^{\mathsf{Gen}}$ and $\{\mathcal{W}_n^{(c)}\}_{c \in \mathcal{C}_n}$:

$$S_n^{\mathsf{Anch},(c)} = \langle \mathcal{R}_n^{\mathsf{Gen}}, \mathcal{W}_n^{(c)} \rangle / \|\mathcal{R}_n^{\mathsf{Gen}}\|_2 . \|\mathcal{W}_n^{(c)}\|_2, \quad c \in \mathcal{C}_n. \tag{21}$$

Then, we define the running anchor loss $\mathcal{L}_n^{\mathsf{Anch}}(\boldsymbol{\mathcal{P}}_n^{\mathsf{Train}})$ as the cross-entropy loss with the following conditional probabilities: $\mathsf{Pr}(c|\mathbf{I}) = \exp\big(S_n^{\mathsf{Anch},(c)}/\tau\big) / \sum_{c' \in \mathcal{C}_n} \exp\big(S_n^{\mathsf{Anch},(c')}/\tau\big)$. Finally, we define the local training loss as a weighted combination of the above two losses:

$$\mathcal{L}_n(\boldsymbol{\mathcal{P}}_n^{\mathsf{Train}}) = (1-\lambda)\mathcal{L}_n^{\mathsf{Anch}}(\boldsymbol{\mathcal{P}}_n^{\mathsf{Train}}) + \lambda \mathcal{L}_n^{\mathsf{Fus}}(\boldsymbol{\mathcal{P}}_n^{\mathsf{Train}}), \tag{22}$$

where $\lambda \in [0, 1]$ controls the influence of anchoring (see *Ablation Study 8* in Sec. 4 for more details).

3.2.5 Global Aggregation

In BLEND, parameter aggregation is applied selectively rather than to all trainable parameters. Specifically, while the global vision/text adapters are aggregated, the personalized projection adapter remains locally optimized at each client n. This design choice stems from the intuition that visual data across clients tends to exhibit significantly higher diversity than text data (which are typically label- or prompt-based), making localized adaptation of the vision pathway more effective for client-specific personalization. To formalize this distinction, the vector of trainable parameters, i.e., $\boldsymbol{\mathcal{P}}_n^{\mathsf{Train}}$ in (18), is divided into aggregatable $\boldsymbol{\mathcal{P}}_n^{\mathsf{Agg}}$ and personalized/non-aggregatable components $\boldsymbol{\mathcal{P}}_n^{\mathsf{Loc}}$ as:

$$\boldsymbol{\mathcal{P}}_n^{\mathsf{Agg}} = \Big[\mathbf{A}_{n,\ell}^{\mathsf{Vis}}, \mathbf{B}_{n,\ell}^{\mathsf{Vis}}, \mathbf{A}_{n,\ell}^{\mathsf{Txt}}, \mathbf{B}_{n,\ell}^{\mathsf{Txt}}\Big]_{\ell^{\mathsf{S}} \le \ell \le \ell^{\mathsf{E}}}, \qquad \boldsymbol{\mathcal{P}}_n^{\mathsf{Loc}} = \Big[\mathbf{A}_n^{\mathsf{Proj}}, \mathbf{B}_n^{\mathsf{Proj}}\Big]. \tag{23}$$

After every J local SGD iterations, client n obtains its latest model parameters:

$$\boldsymbol{\mathcal{P}}_n^{\mathsf{Train},(k,J)} = \big[\boldsymbol{\mathcal{P}}_n^{\mathsf{Agg},(k,J)},\ \boldsymbol{\mathcal{P}}_n^{\mathsf{Loc},(k,J)}\big]. \tag{24}$$

Afterwards, the client transmits only its aggregatable parameters $\boldsymbol{\mathcal{P}}_n^{\mathsf{Agg},(k,J)}$ to the central server, while retaining the localized parameters $\boldsymbol{\mathcal{P}}_n^{\mathsf{Loc},(k,J)}$ for personalization. The server then performs a global aggregation step to update the shared/global parameters as follows:

$$\boldsymbol{\mathcal{P}}^{\mathsf{Agg},(k+1)} = \sum_{n \in \mathcal{N}} d_n \boldsymbol{\mathcal{P}}_n^{\mathsf{Agg},(k,J)}, \tag{25}$$

where $d_n = |\mathcal{D}_n| / \sum_{n' \in \mathcal{N}} |\mathcal{D}_{n'}|$. The updated global parameters $\boldsymbol{\mathcal{P}}^{\mathsf{Agg},(k+1)}$ are then redistributed to the clients in the subsequent communication round to synchronize their aggregatable parameters, while each client retains its personalized localized parameters $\boldsymbol{\mathcal{P}}_n^{\mathsf{Loc},(k,J)}$ for subsequent refinement.

Table 1: Performance results (accuracy, %). The row Δ denotes the performance gap between BLEND and the best-performing baseline. Standard deviations across three random seeds are shown via sub-indices. In the bottom rows, statistical significance is assessed using a two-sided paired t-test on seed-level H-Mean scores between BLEND and the strongest baseline, with improvements statistically significant at $p < 0.05$.

	UCF101			Oxford Pets			Stanford Cars			Flowers			Imagenet			FGVC Aircraft		
	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean
Zero-shot CLIP	68.10$_{\pm 0.00}$	75.18$_{\pm 0.00}$	71.46$_{\pm 0.00}$	89.47$_{\pm 0.00}$	96.81$_{\pm 0.00}$	93.00$_{\pm 0.00}$	63.77$_{\pm 0.00}$	**74.94$_{\pm 0.00}$**	68.91$_{\pm 0.00}$	69.42$_{\pm 0.00}$	**76.74$_{\pm 0.00}$**	72.90$_{\pm 0.00}$	72.40$_{\pm 0.00}$	68.08$_{\pm 0.00}$	70.17$_{\pm 0.00}$	27.49$_{\pm 0.00}$	33.05$_{\pm 0.00}$	30.02$_{\pm 0.00}$
MMA+FedAvg	85.40$_{\pm 0.93}$	75.95$_{\pm 1.20}$	80.40$_{\pm 1.08}$	95.12$_{\pm 0.07}$	**97.65$_{\pm 0.08}$**	96.37$_{\pm 0.05}$	76.23$_{\pm 1.63}$	69.60$_{\pm 0.40}$	72.75$_{\pm 0.82}$	96.93$_{\pm 0.22}$	70.87$_{\pm 1.19}$	81.87$_{\pm 0.87}$	75.52$_{\pm 2.03}$	**69.18$_{\pm 0.98}$**	72.21$_{\pm 0.32}$	38.92$_{\pm 2.03}$	**33.77$_{\pm 0.98}$**	36.10$_{\pm 0.32}$
pFedMMA	89.95$_{\pm 1.16}$	70.51$_{\pm 0.66}$	79.04$_{\pm 0.22}$	96.81$_{\pm 0.81}$	95.08$_{\pm 0.25}$	95.93$_{\pm 0.33}$	67.45$_{\pm 2.01}$	64.52$_{\pm 0.72}$	65.95$_{\pm 1.33}$	84.61$_{\pm 2.42}$	69.89$_{\pm 0.73}$	76.52$_{\pm 0.62}$	49.15$_{\pm 1.42}$	50.93$_{\pm 1.02}$	50.03$_{\pm 1.22}$	40.37$_{\pm 5.66}$	29.14$_{\pm 1.69}$	33.78$_{\pm 3.04}$
pFedMoAP	93.36$_{\pm 0.12}$	67.27$_{\pm 0.02}$	78.19$_{\pm 0.04}$	99.18$_{\pm 0.02}$	84.84$_{\pm 0.10}$	91.45$_{\pm 0.01}$	92.00$_{\pm 0.20}$	68.50$_{\pm 0.05}$	78.53$_{\pm 0.07}$	98.83$_{\pm 0.15}$	71.49$_{\pm 0.00}$	82.96$_{\pm 0.05}$	86.65$_{\pm 0.00}$	62.56$_{\pm 0.03}$	72.66$_{\pm 0.02}$	65.51$_{\pm 0.08}$	18.58$_{\pm 0.11}$	28.95$_{\pm 0.01}$
FedPHA	92.61$_{\pm 0.70}$	50.29$_{\pm 2.31}$	65.14$_{\pm 1.82}$	98.81$_{\pm 0.63}$	70.06$_{\pm 2.75}$	81.95$_{\pm 1.89}$	91.90$_{\pm 0.55}$	44.52$_{\pm 1.31}$	59.96$_{\pm 1.07}$	98.73$_{\pm 0.21}$	49.77$_{\pm 2.86}$	66.12$_{\pm 2.53}$	**90.49$_{\pm 0.14}$**	39.71$_{\pm 2.26}$	55.20$_{\pm 2.33}$	66.62$_{\pm 0.44}$	13.00$_{\pm 2.97}$	21.59$_{\pm 4.12}$
FedOTP	82.79$_{\pm 1.05}$	27.04$_{\pm 2.36}$	40.70$_{\pm 2.58}$	95.20$_{\pm 1.13}$	51.47$_{\pm 4.08}$	66.75$_{\pm 3.74}$	78.52$_{\pm 1.64}$	14.67$_{\pm 1.36}$	24.70$_{\pm 1.95}$	95.84$_{\pm 0.39}$	35.53$_{\pm 4.86}$	51.64$_{\pm 5.00}$	81.23$_{\pm 0.91}$	14.23$_{\pm 1.50}$	24.22$_{\pm 1.62}$	51.95$_{\pm 1.23}$	5.98$_{\pm 1.11}$	10.68$_{\pm 1.77}$
PromptFL	80.60$_{\pm 1.26}$	70.83$_{\pm 2.15}$	75.36$_{\pm 0.75}$	95.37$_{\pm 0.61}$	96.33$_{\pm 0.89}$	95.84$_{\pm 0.71}$	67.89$_{\pm 2.24}$	70.14$_{\pm 0.13}$	69.94$_{\pm 1.24}$	96.24$_{\pm 0.32}$	71.23$_{\pm 2.91}$	81.83$_{\pm 1.82}$	73.38$_{\pm 0.40}$	67.75$_{\pm 2.00}$	70.45$_{\pm 1.30}$	27.56$_{\pm 7.64}$	20.44$_{\pm 10.99}$	22.79$_{\pm 10.69}$
FedPGP	90.88$_{\pm 1.02}$	55.51$_{\pm 0.15}$	68.92$_{\pm 0.20}$	95.59$_{\pm 0.35}$	97.29$_{\pm 0.35}$	96.43$_{\pm 0.21}$	87.82$_{\pm 0.73}$	57.87$_{\pm 3.16}$	69.71$_{\pm 2.23}$	98.40$_{\pm 0.78}$	65.43$_{\pm 2.11}$	78.58$_{\pm 1.71}$	89.21$_{\pm 0.89}$	45.93$_{\pm 1.01}$	60.64$_{\pm 1.00}$	61.37$_{\pm 1.33}$	21.03$_{\pm 0.62}$	31.31$_{\pm 0.55}$
PromptFolio	90.19$_{\pm 0.59}$	52.97$_{\pm 3.98}$	66.64$_{\pm 3.00}$	97.94$_{\pm 0.13}$	71.87$_{\pm 4.42}$	82.83$_{\pm 2.99}$	73.42$_{\pm 3.79}$	54.88$_{\pm 5.98}$	62.76$_{\pm 5.31}$	97.94$_{\pm 0.22}$	27.59$_{\pm 6.74}$	42.40$_{\pm 8.25}$	85.74$_{\pm 1.28}$	50.81$_{\pm 4.93}$	62.30$_{\pm 4.15}$	45.12$_{\pm 2.50}$	7.74$_{\pm 0.48}$	13.21$_{\pm 0.81}$
BLEND	**95.03$_{\pm 0.16}$**	**78.20$_{\pm 0.21}$**	**85.80$_{\pm 0.19}$**	**99.21$_{\pm 0.24}$**	96.99$_{\pm 0.03}$	**98.09$_{\pm 0.11}$**	**93.72$_{\pm 0.20}$**	70.20$_{\pm 0.26}$	**80.27$_{\pm 0.23}$**	**98.94$_{\pm 0.06}$**	75.21$_{\pm 0.36}$	**85.45$_{\pm 0.21}$**	87.83$_{\pm 0.13}$	66.31$_{\pm 0.22}$	**75.56$_{\pm 0.19}$**	**70.05$_{\pm 0.61}$**	33.24$_{\pm 0.47}$	**45.09$_{\pm 0.55}$**
Δ ($p-$value)			**+5.40** (0.025)			**+1.66** (0.009)			**+1.74** (0.014)			**+2.49** (0.003)			**+2.90** (0.001)			**+8.99** (0.003)
	Food101			DTD			EuroSAT			Caltech			**Cross-Dataset Avg. (↑)**			**Cross-Dataset Std. (↓)**		
	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean
Zero-shot CLIP	89.44$_{\pm 0.00}$	90.67$_{\pm 0.00}$	90.05$_{\pm 0.00}$	54.75$_{\pm 0.00}$	58.94$_{\pm 0.00}$	56.76$_{\pm 0.00}$	50.21$_{\pm 0.00}$	69.87$_{\pm 0.00}$	58.43$_{\pm 0.00}$	97.16$_{\pm 0.00}$	**94.21$_{\pm 0.00}$**	95.66$_{\pm 0.00}$	68.22$_{\pm 0.00}$	73.85$_{\pm 0.00}$	70.74$_{\pm 0.00}$	19.86$_{\pm 0.00}$	17.81$_{\pm 0.00}$	18.72$_{\pm 0.00}$
MMA+FedAvg	89.85$_{\pm 0.33}$	90.21$_{\pm 0.67}$	90.03$_{\pm 0.50}$	79.30$_{\pm 1.11}$	60.39$_{\pm 1.80}$	68.56$_{\pm 1.50}$	92.05$_{\pm 1.96}$	70.49$_{\pm 5.21}$	79.67$_{\pm 2.61}$	97.60$_{\pm 0.30}$	93.30$_{\pm 0.52}$	95.40$_{\pm 0.17}$	82.69$_{\pm 0.86}$	73.14$_{\pm 1.21}$	77.34$_{\pm 0.79}$	16.57$_{\pm 0.75}$	17.49$_{\pm 1.44}$	16.46$_{\pm 0.76}$
pFedMMA	95.80$_{\pm 0.19}$	86.83$_{\pm 0.40}$	91.10$_{\pm 0.21}$	77.87$_{\pm 2.04}$	57.81$_{\pm 1.34}$	66.34$_{\pm 1.13}$	80.06$_{\pm 15.74}$	49.22$_{\pm 5.28}$	60.81$_{\pm 8.69}$	97.79$_{\pm 0.31}$	92.95$_{\pm 0.11}$	95.31$_{\pm 0.20}$	77.99$_{\pm 3.03}$	66.69$_{\pm 1.12}$	71.48$_{\pm 1.58}$	19.01$_{\pm 4.52}$	19.93$_{\pm 1.48}$	19.20$_{\pm 2.52}$
pFedMoAP	**96.28$_{\pm 0.08}$**	86.68$_{\pm 0.00}$	91.23$_{\pm 0.04}$	**92.43$_{\pm 0.42}$**	47.72$_{\pm 0.00}$	62.94$_{\pm 0.10}$	98.05$_{\pm 1.25}$	63.47$_{\pm 0.00}$	77.05$_{\pm 0.39}$	97.98$_{\pm 0.18}$	93.69$_{\pm 0.00}$	95.79$_{\pm 0.08}$	92.03$_{\pm 0.25}$	66.48$_{\pm 0.00}$	75.98$_{\pm 0.08}$	9.60$_{\pm 0.35}$	20.46$_{\pm 0.00}$	18.22$_{\pm 0.11}$
FedPHA	95.87$_{\pm 0.40}$	67.45$_{\pm 3.86}$	79.08$_{\pm 3.33}$	90.41$_{\pm 0.44}$	34.01$_{\pm 2.07}$	49.39$_{\pm 2.17}$	**100.00$_{\pm 0.00}$**	28.33$_{\pm 5.81}$	43.84$_{\pm 6.88}$	**98.09$_{\pm 0.38}$**	81.83$_{\pm 1.89}$	89.21$_{\pm 0.98}$	92.35$_{\pm 0.37}$	47.90$_{\pm 2.68}$	61.15$_{\pm 2.48}$	9.25$_{\pm 0.23}$	19.76$_{\pm 1.58}$	19.03$_{\pm 1.85}$
FedOTP	90.33$_{\pm 0.14}$	44.62$_{\pm 1.31}$	59.73$_{\pm 1.21}$	88.52$_{\pm 0.64}$	34.69$_{\pm 4.02}$	49.72$_{\pm 4.30}$	**100.00$_{\pm 0.00}$**	31.75$_{\pm 2.04}$	48.17$_{\pm 2.37}$	94.79$_{\pm 0.54}$	65.70$_{\pm 2.78}$	77.57$_{\pm 1.91}$	85.92$_{\pm 0.67}$	32.57$_{\pm 2.39}$	45.39$_{\pm 2.48}$	13.15$_{\pm 0.54}$	**17.35$_{\pm 1.46}$**	19.66$_{\pm 1.42}$
PromptFL	90.54$_{\pm 0.22}$	90.52$_{\pm 0.54}$	90.53$_{\pm 0.31}$	77.01$_{\pm 0.25}$	51.17$_{\pm 2.55}$	61.44$_{\pm 1.75}$	81.72$_{\pm 4.28}$	65.91$_{\pm 4.69}$	72.76$_{\pm 2.62}$	97.17$_{\pm 0.38}$	93.23$_{\pm 0.53}$	95.16$_{\pm 0.18}$	78.75$_{\pm 1.72}$	69.95$_{\pm 2.54}$	73.61$_{\pm 2.01}$	19.60$_{\pm 2.33}$	21.20$_{\pm 3.15}$	20.18$_{\pm 3.00}$
FedPGP	91.58$_{\pm 0.46}$	89.81$_{\pm 0.22}$	90.69$_{\pm 0.34}$	81.99$_{\pm 2.52}$	50.76$_{\pm 3.56}$	62.62$_{\pm 2.78}$	84.73$_{\pm 3.08}$	67.67$_{\pm 8.88}$	74.90$_{\pm 5.96}$	97.21$_{\pm 0.38}$	92.55$_{\pm 0.34}$	94.83$_{\pm 0.26}$	87.88$_{\pm 1.06}$	64.38$_{\pm 1.94}$	72.86$_{\pm 1.42}$	10.14$_{\pm 0.95}$	22.50$_{\pm 2.63}$	18.44$_{\pm 1.77}$
PromptFolio	93.05$_{\pm 0.38}$	53.00$_{\pm 6.18}$	67.33$_{\pm 5.12}$	84.86$_{\pm 0.36}$	42.30$_{\pm 4.44}$	56.33$_{\pm 4.04}$	92.24$_{\pm 1.83}$	65.03$_{\pm 6.62}$	75.79$_{\pm 3.94}$	97.00$_{\pm 0.12}$	82.21$_{\pm 5.02}$	88.91$_{\pm 2.89}$	85.75$_{\pm 1.10}$	50.84$_{\pm 4.87}$	61.85$_{\pm 4.04}$	16.13$_{\pm 1.23}$	21.41$_{\pm 1.82}$	21.58$_{\pm 1.96}$
BLEND	95.24$_{\pm 0.16}$	**91.25$_{\pm 0.11}$**	**93.20$_{\pm 0.04}$**	91.64$_{\pm 1.07}$	**64.32$_{\pm 0.70}$**	**75.59$_{\pm 0.84}$**	99.36$_{\pm 0.28}$	**76.50$_{\pm 0.90}$**	**86.44$_{\pm 0.65}$**	**98.09$_{\pm 0.09}$**	94.01$_{\pm 0.24}$	**96.01$_{\pm 0.09}$**	**92.91$_{\pm 0.30}$**	**74.62$_{\pm 0.35}$**	**82.15$_{\pm 0.31}$**	**8.40$_{\pm 0.30}$**	17.54$_{\pm 0.25}$	**14.40$_{\pm 0.26}$**
Δ ($p-$value)			**+1.97** (0.001)			**+7.03** (0.005)			**+6.77** (0.033)			**+0.22** (0.011)			**+4.81**			**-2.06**

In particular, each client n instantiates its aggregatable and localized parameters as:

$$\underbrace{\boldsymbol{\mathcal{P}}_n^{\mathsf{Agg},(k+1,0)} \leftarrow \boldsymbol{\mathcal{P}}^{\mathsf{Agg},(k+1)}}_{\text{Synchronized by the Global Parameters}}, \quad \underbrace{\boldsymbol{\mathcal{P}}_n^{\mathsf{Loc},(k+1,0)} \leftarrow \boldsymbol{\mathcal{P}}_n^{\mathsf{Loc},(k,J)}}_{\text{Carried Over from the Previous Round}}, \tag{26}$$

and then starts the next local model training round by following (19), with k replaced by $k+1$ and the initialization given by $\boldsymbol{\mathcal{P}}_n^{\mathsf{Train},(k+1,0)} = \left[\boldsymbol{\mathcal{P}}_n^{\mathsf{Agg},(k+1,0)}, \boldsymbol{\mathcal{P}}_n^{\mathsf{Loc},(k+1,0)}\right]$.

4 Experiments

We next study the performance of BLEND (code: https://github.com/payamsiabd/BLEND.git). Experiments were run on a server with 3×L40S GPUs, 1TB of RAM, and 2×Intel 8562Y CPUs.

System Setup: Following [3] and baselines listed below, we consider a system comprising 10 clients (higher numbers of clients are studied in *Ablation Study 2*). We inject the global adapters to the vision and text transformer layers from $\ell^{\mathsf{S}} = 1$ to $\ell^{\mathsf{E}} = 12$ (see *Ablation Study 6* for other injection-layer choices), consider the ranks $r_x = r_w = 16$ and $r_p = 256$, and set the balancing coefficients as $\gamma = 0.09$, $\lambda = 0.2$. Each client performs one epoch of local training (with mini-batch size of 16 and $\eta = 0.0015$) before participating in global aggregation.

Baselines: We compare BLEND with several baselines, including zero-shot CLIP [1], FedOTP [4], PromptFolio [28], FedPGP [5], PromptFL [6], pFedMMA [10], pFedMoAP [2], FedPHA [3], and MMA + FedAvg [11]. Here, MMA + FedAvg is our federated extension of a state-of-the-art centralized CLIP fine-tuning, i.e., MMA [11], where clients perform local MMA-based fine-tuning and models are aggregated via averaging as in (25). For all baselines, we use the optimal hyperparameters reported in their papers.

Datasets: We consider 10 datasets: ImageNet [33], Caltech [34], OxfordPets [35], StanfordCars [36], Flowers [37], Food101 [38], DTD [39], EuroSAT [40], FGVC Aircraft [41], and UCF101 [42].

Data Distribution and Evaluation Method: Each dataset is evenly divided into *base* and *novel* subsets, where only the base portion is used for model training. To simulate non-IID data across clients, we adopt the Dirichlet distribution $\mathrm{Dir}(\alpha^{\mathsf{Dir}})$, where α^{Dir} denotes the concentration parameter controlling the degree of data heterogeneity [3]. Following [3], we sample 16 shots per class from the base subset for training across clients (the effect of varying this few-shot configuration is analyzed in *Ablation Study 3*). Following [11, 5], we consider two performance metrics:

1. ***Personalization (Per.) Accuracy:*** measures the accuracy of the final model of each client on its local test data from base classes (10% of local dataset is used as test data), thereby capturing the model's *personalization capability*.
2. ***Generalization (Gen.) Accuracy:*** measures the accuracy of the final model of each client tested on the novel classes, thereby capturing the *generalization capability*.

We report the **harmonic mean (H-Mean)** of the above metrics as a measure of the balance between personalization and generalization [11, 5]. Since the H-Mean emphasizes the smaller of the two values, it is sensitive to imbalances and reflects how well both metrics are simultaneously achieved.

Main Results: In our main experiments, we set $\alpha^{\mathsf{Dir}} = 0.01$, reflecting highly non-IID data across clients (other choices of α^{Dir} are studied in *Ablation Study 1*). Table 1 compares BLEND with the baselines after 50 global rounds (convergence across all methods is reached). Across all datasets, BLEND consistently achieves the best H-Mean (i.e., the best balance of personalization and generalization), with a notable average improvement of $+4.81$, while also exhibiting the lowest performance variation

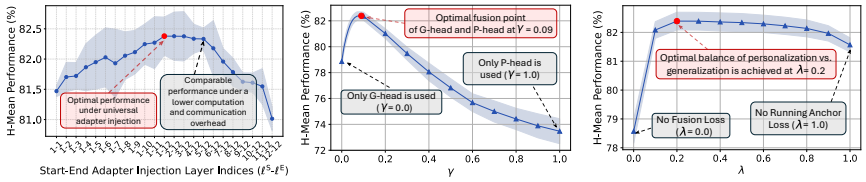


Figure 3: Effect of injecting adapters at various transformer layers (left plot). Effect of varying γ (middle plot) and λ (right plot), which are used in (17) and (22), respectively, on the performance.

across the datasets (i.e., higher consistency), as indicated by its smallest standard deviation (Std.) of accuracy across the datasets, which is 2.06% lower than the best baseline. Further, Appendix A reports the **communication/computation overhead** of BLEND compared to the baselines, showing that BLEND achieves higher H-Mean performance with comparable/lower resource overhead.

We summarize BLEND's performance variation under increasing task difficulty in Appendix M. The results show larger improvements over the best baseline as data heterogeneity, the number of clients, and local data scarcity increase. Below, we present a set of ablation studies, where all results are averaged over the 10 datasets. Detailed ablation studies for each dataset are in the Appendix sections.

(Ablation Study 1) **Data heterogeneity variations:** In addition to the $\alpha^{\text{Dir}} = 0.01$ used above, we further evaluate the impact of data heterogeneity by varying $\alpha^{\text{Dir}} \in \{0.1, 10\}$ in Appendix B. Further, Appendix D evaluates BLEND on iNaturalist [43], a distributed dataset with *real-world data distributions* exhibiting extreme non-IID characteristics. Across all settings, BLEND maintains superior H-Mean performance, showing robustness to diverse data distributions.

(Ablation Study 2) **Varying the number of clients:** In addition to the $N = 10$ setting used above, we examine BLEND under different system sizes $N \in \{20, 30\}$ in Appendix C. Also, Appendix D evaluates BLEND in real-world large-scale settings with $N \in \{100, 200\}$. The results in Appendix C and Appendix D show that BLEND outperforms the baselines across various system scales, demonstrating its scalability.

(Ablation Study 3) **Different few-shot settings:** In Appendix E, we evaluate the few-shot learning capability of BLEND under varying numbers of shots per class, showing that BLEND consistently outperforms baselines across different data scarcity configurations.

(Ablation Study 4) **Different bottleneck sizes for global adapters:** Table 2 analyzes the impact of the global adapter bottleneck rank $r{=}r_x{=}r_w$ on average performance across datasets (Appendix F contains the per-dataset results). As shown, incorporating global adapters consistently outperforms the 'No adapter' setting and exhibits robustness across different adapter sizes. Moreover, the H-Mean increases with r up to 16, showing that strong performance can be achieved even under low parameter-update budgets in the federated learning process.

Table 2: Effect of the rank $r{=}r_x{=}r_w$ on performance.

Rank (r)	Per.	Gen.	H-Mean
No adapter	89.50	72.31	79.95
4	92.25	74.69	82.05
8	92.45	**74.70**	82.10
16	92.91	74.62	**82.15**
32	**92.93**	74.39	81.97
64	92.91	74.09	81.74
128	92.80	73.70	81.43
256	92.68	73.47	81.20
Full-rank	92.48	73.63	81.26

(Ablation Study 5) **Different bottleneck sizes for the personalized projection adapter:** Table 3 analyzes the impact of the personalized adapter projection rank r_p on average performance across datasets (Appendix G contains the per-dataset results). As r_p increases, the personalization accuracy improves (i.e., higher-capacity personalized adapters can capture client-specific characteristics), while the generalization gradually degrades. Overall, the H-Mean improves progressively and peaks at $r_p = 256$, reflecting the best personalization–generalization trade-off.

(Ablation Study 6) **Adapter injection at different layers:** In the left plot of Fig. 3, we evaluate the effect of injecting the global adapters at different transformer layers on average performance across datasets (Appendix H contains the per-dataset results). As shown, the placement of adapters across transformer layers has a notable impact on performance. As expected, adding adapters to all layers ($1{-}12$, i.e., $\ell^{\text{S}} = 1$ and $\ell^{\text{E}} = 12$) yields the highest accuracy; however, the H-Mean achieved by the $5 - 12$ configuration is very close to that of $1 - 12$ setting, indicating that adapter injection in all layers is not strictly necessary to obtain good performance.

Table 3: Effect of the rank r_p on performance.

Rank (r_p)	Per.	Gen.	H-Mean
4	89.99	**75.00**	81.44
8	90.45	74.83	81.73
16	91.00	74.86	81.73
32	91.59	74.91	81.96
64	92.12	74.64	81.92
128	92.57	74.71	82.12
256	92.91	74.62	**82.15**
Full-rank	**92.95**	74.46	82.08

(Ablation Study 7) **Effect of γ on performance:** In the middle plot of Fig. 3, we study the effect of varying γ in (17) on the average performance across datasets (per-dataset results and individual personalization/generalization accuracies are provided in Appendix I). We see that $\gamma = 0.09$ yields the highest H-Mean, while larger values lead to a worse personalization–generalization trade-off.

(Ablation Study 8) **Effect of λ on performance:** In the right plot of Fig. 3, we examine the effect of varying λ in (22) on the average performance across datasets (per-dataset results and individual personalization/ generalization accuracies are provided in Appendix J). We observe that $\lambda = 0.2$ achieves the highest H-Mean. Further, the results in Appendix J (see Fig. 48) reveal that choosing λ too high weakens the anchoring to the zero-shot model, yielding the worst generalization performance.

(Ablation Study 9) **Effect of aggregation of various components:** Table 4 examines how aggregating different trainable components (i.e., global vision/text adapters and personalized adapter projection) affects average performance across datasets (Appendix K contains the per-dataset results). BLEND (i.e., aggregating all global vision/text adapters and keeping the personalized projection adapter at each client for local adaptation) delivers the best overall H-Mean performance. Also, aggregating all trainable parameters ('Full') improves generalization but substantially harms personalization, while keeping all global adapters at clients ('NoAgg') preserves personalization to some extent but leads to a notable drop in generalization. Interestingly, even though 'NoAgg' uses all adapters for personalization, it does *not* achieve the highest personalization performance, highlighting the importance of effective cross-client knowledge sharing during model aggregation.

Table 4: Comparison of different aggregation strategies.

Agg. Strategy	Per.	Gen.	H-Mean
BLEND	92.91	74.62	**82.15**
Full	83.35	**75.41**	79.01
NoAgg	89.12	69.71	77.71
Only Vision	74.64	70.02	71.82
Only Text	**93.84**	67.31	77.04

(Ablation Study 10) **Effect of trainable text projection head:** Appendix L shows that unfreezing the text projection head mostly degrades the H-Mean performance, as training the text projection head causes forgetting of the pre-trained CLIP knowledge.

(Ablation Study 11) **Effect of running anchor:** We evaluate BLEND's running anchor loss against two regularization strategies: a frozen zero-shot G-head and a KL-distilled G-head. The running anchor achieves a better personalization–generalization balance, as detailed in Appendix N.

(Ablation Study 12) **Effect of adaptive fusion weight:** We study the effect of the fusion weight γ under varying levels of data heterogeneity and numbers of clients. As shown in Appendix O, the optimal value varies with the degree of data heterogeneity and the number of clients.

(Ablation Study 13) **Effect of explicit personalization:** In Appendix P, we study whether supervising the personalized representation with a separate cross-entropy loss, rather than relying on the fusion loss, improves BLEND's performance. The results show that explicit personalization slightly improves personalization accuracy but reduces generalization and the overall H-mean.

(Ablation Study 14) **Hyperparameter selection and validation:** To avoid dataset-specific tuning, for BLEND, we used a unified hyperparameter configuration across all experimental settings. To further address potential test-set-informed selection, we conduct additional experiments on an independent held-out validation split in Appendix Q. The validation results identify the same optimal values of γ, λ, and adapter ranks as those reported in the above sensitivity analysis (Fig. 3, Table 2, and Table 3).

5 Limitations

BLEND is built on CLIP and focuses on text–image modalities. Extending it to broader multi-modal settings (e.g., audio or video), retrieval and generative tasks, and multi-task learning remains an open direction (see Appendix R). Also, BLEND's performance depends on hyperparameters (e.g., γ and λ) that govern the personalization–generalization trade-off, and their adaptive selection in new environments remains an open problem. Moreover, extension of BLEND to fully-decentralized/serverless FedVLMs remains open. Finally, the integration of privacy-preserving parameter aggregation mechanisms, such as differential privacy, within BLEND is worth investigating.

6 Conclusion

We introduced BLEND, a FedVLM framework that integrates globally aggregated adapters into the vision and text encoders of CLIP, together with a locally optimized vision projection head. Through a novel loss formulation, BLEND leverages the output of these components to enhance the balance of personalization vs. generalization. Extensive experiments validated each of the introduced model components and showed that `BLEND` consistently outperforms state-of-the-art baselines.

7 Acknowledgments

This work was supported in part by the U.S. National Science Foundation (NSF) under Grants ECCS-2543754, SaTC-2513164, and ECCS-2512911; and in part by the National Natural Science Foundation of China under Grants No. 62301487 and No. 62671554.

References

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

[2] Jun Luo, Chen Chen, and Shandong Wu. Mixture of experts made personalized: Federated prompt learning for vision-language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xiDJaTim3P.

[3] Chengying Fang, Wenke Huang, Guancheng Wan, Yihao Yang, and Mang Ye. FedPHA: Federated prompt learning for heterogeneous client adaptation. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=y7pDvbi9xz.

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

[5] Tianyu Cui, Hongxia Li, Jingya Wang, and Ye Shi. Harmonizing generalization and personalization in federated prompt learning. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=YYwERRXsJW.

[6] 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*, 23(5):5179–5194, 2023.

[7] Fardis Nadimi, Payam Abdisarabshali, Kasra Borazjani, Jacob Chakareski, and Seyyedali Hosseinalipour. Multi-modal multi-task federated foundation models for next-generation extended reality systems: Towards privacy-preserving distributed intelligence in AR/VR/MR. *npj Wireless Technology*, 1(1):6, 2025.

[8] Kasra Borazjani, Payam Abdisarabshali, Fardis Nadimi, Naji Khosravan, Minghui Liwang, Xianbin Wang, Yiguang Hong, and Seyyedali Hosseinalipour. Multi-modal multi-task (M3T) federated foundation models for embodied AI: Potentials and challenges for edge integration. *IEEE Internet of Things Magazine*, 9(1):20–30, 2026. doi: 10.1109/MIOT.2025.3604330.

[9] Payam Abdisarabshali, Fardis Nadimi, Kasra Borazjani, Naji Khosravan, Minghui Liwang, Wei Ni, Dusit Niyato, Michael Langberg, and Seyyedali Hosseinalipour. Hierarchical federated foundation models over wireless networks for multi-modal multi-task intelligence: Integration of edge learning with D2D/P2P-enabled fog learning architectures. *IEEE Communications Magazine*, 64(4):66–72, 2026. doi: 10.1109/MCOM.001.2500502.

[10] Sajjad Ghiasvand, Mahnoosh Alizadeh, and Ramtin Pedarsani. pfedmma: Personalized federated fine-tuning with multi-modal adapter for vision-language models. *arXiv preprint arXiv:2507.05394*, 2025.

[11] Lingxiao Yang, Ru-Yuan Zhang, Yanchen Wang, and Xiaohua Xie. Mma: Multi-modal adapter for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23826–23837, 2024.

[12] Yuncheng Guo and Xiaodong Gu. Mmrl: Multi-modal representation learning for vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 25015–25025, 2025.

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

[14] Chen Xu, Yuhan Zhu, Haocheng Shen, Boheng Chen, Yixuan Liao, Xiaoxin Chen, and Limin Wang. Progressive visual prompt learning with contrastive feature re-formation. *International Journal of Computer Vision*, 133(2):511–526, 2025.

[15] Cairong Zhao, Yubin Wang, Xinyang Jiang, Yifei Shen, Kaitao Song, Dongsheng Li, and Duoqian Miao. Learning domain invariant prompt for vision-language models. *IEEE Transactions on Image Processing*, 33:1348–1360, 2024.

[16] Hantao Yao, Rui Zhang, and Changsheng Xu. Tcp: Textual-based class-aware prompt tuning for visual-language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23438–23448, 2024.

[17] Yu Du, Tong Niu, and Rong Zhao. Mixture of prompts learning for vision-language models. *Frontiers in Artificial Intelligence*, 8:1580973, 2025.

[18] Haoyang Li, Liang Wang, Chao Wang, Jing Jiang, Yan Peng, and Guodong Long. Dpc: Dual-prompt collaboration for tuning vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 25623–25632, 2025.

[19] Mingyu Kim, Jongwoo Ko, and Mijung Park. Bayesian principles improve prompt learning in vision-language models. *arXiv preprint arXiv:2504.14123*, 2025.

[20] Debarshi Brahma, Anuska Roy, and Soma Biswas. Prompt tuning vision language models with margin regularizer for few-shot learning under distribution shifts. *arXiv preprint arXiv:2505.15506*, 2025.

[21] Yuhan Zhu, Guozhen Zhang, Chen Xu, Haocheng Shen, Xiaoxin Chen, Gangshan Wu, and Limin Wang. Efficient test-time prompt tuning for vision-language models. *arXiv preprint arXiv:2408.05775*, 2024.

[22] Yongju Jia, Jiarui Ma, Xiangxian Li, Baiqiao Zhang, Xianhui Cao, Juan Liu, and Yulong Bian. Llm-empowered dynamic prompt routing for vision-language models tuning under long-tailed distributions. *arXiv preprint arXiv:2508.15688*, 2025.

[23] Mengwei Li, Zilei Wang, and Yixin Zhang. Improving zero-shot generalization for clip with prompt ensemble self-distillation. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.

[24] Xiao Shi, Yangjun Ou, and Zhenzhong Chen. Sempt: Semantic prompt tuning for vision-language models. *arXiv preprint arXiv:2508.10645*, 2025.

[25] Biao Chen, Kunbin He, Zhikun Zheng, Mengmeng Jing, and Lin Zuo. Chain-of-thought guided semantic debiasing for low-shot vision-language tasks. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 4600–4609, 2025.

[26] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *European conference on computer vision*, pages 493–510. Springer, 2022.

[27] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, 132(2):581–595, 2024.

[28] Bikang Pan, Wei Huang, and Ye Shi. Federated learning from vision-language foundation models: Theoretical analysis and method. *Advances in Neural Information Processing Systems*, 37:30590–30623, 2024.

[29] Haokun Chen, Yao Zhang, Denis Krompass, Jindong Gu, and Volker Tresp. Feddat: An approach for foundation model finetuning in multi-modal heterogeneous federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11285–11293, 2024.

[30] Ahmed Imteaj, Md Zarif Hossain, Saika Zaman, and Abdur R Shahid. Tripleplay: Enhancing federated learning with clip for non-iid data and resource efficiency. In *2024 International Conference on Machine Learning and Applications (ICMLA)*, pages 1474–1480. IEEE, 2024.

[31] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, July 2006. ISBN 0471241954.

[32] Simon Schrodi, David T Hoffmann, Max Argus, Volker Fischer, and Thomas Brox. Two effects, one trigger: On the modality gap, object bias, and information imbalance in contrastive vision-language models. In *International Conference on Learning Representations*, volume 2025, pages 27836–27864, 2025.

[33] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[34] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004.

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

[36] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

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

[38] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pages 446–461. Springer, 2014.

[39] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014.

[40] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

[41] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[42] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[43] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Federated visual classification with real-world data distribution. In *European Conference on Computer Vision*, pages 76–92. Springer, 2020.

Appendix Table of Contents

A Communication and Computation Overhead **2**

B Data Heterogeneity Variations **3**

C Varying the Number of Clients **6**

D Large-Scale Setting with Real-World Data Distribution **7**

E Different Few-shot Settings **7**

F Different Bottleneck Sizes for Global Adapters **11**

G Different Bottleneck Sizes for Personalized Projection Adapter **12**

H Adapter Injection at Different Layers **13**

I Effect of γ on Performance **16**

J Effect of λ on Performance **19**

K Effect of Aggregation on Various Components **22**

L Effect of Trainable Text Projection Head **23**

M Performance Variation Across Different Settings **24**

N Effect of the Running Anchor Loss **25**

O Effect of the Adaptive Fusion Weight **26**

P Effect of Explicit Personalization **27**

Q Hyperparameter Selection and Held-Out Validation **28**

R Limitations and Generalization Beyond CLIP **28**

A Communication and Computation Overhead

In this section, we demonstrate that BLEND achieves superior performance while using comparable or fewer trainable parameters during local training and incurring lower communication overhead. To this end, we compare against both *prompt-based* and *adapter-based* baselines under controlled resource budgets. Specifically, for each setting, we adjust BLEND (e.g., by reducing the number of adapters or their bottleneck size) to match the parameter or communication budget of the baselines, and then evaluate performance under these constraints. All reported results are averaged across all 10 datasets used in the paper.

Tables 5 and 6 report results under comparable communication overhead. Table 5 corresponds to a high-budget regime (matching the resource footprint of MMA+FedAVG, which is a rather heavy adapter-based fine-tuning method), where BLEND achieves the highest H-Mean while transmitting fewer bits than MMA+FedAVG. Table 6 presents the low-budget regime, showing that BLEND maintains strong performance even with significantly reduced communication, outperforming prompt-based and lightweight adapter-based methods.

Table 5: Comparable communicated parameters (high budget). Communication overhead is measured as the total number of bits transmitted to the server over 50 global rounds, reported in MB.

Method	Total Number of Exchanged Params.	Communication Overhead (MB)	Per.	Gen.	H-Mean
MMA+FedAVG	675K	64.33	82.69	73.14	77.34
BLEND	507K	48.38	92.92	74.61	**82.15**

Table 6: Comparable communicated parameters (low budget). Communication overhead is measured as the total number of bits transmitted to the server over 50 global rounds, reported in MB.

Method	Total Number of Exchanged Params.	Communication Overhead (MB)	Per.	Gen.	H-Mean
pFedMMA	8.4K	0.81	77.99	66.69	71.48
pFedMOAP	8.2K	0.78	92.03	66.48	75.98
FedPHA	8.2K	0.78	92.35	47.90	61.15
FedOTP	8.2K	0.78	85.92	32.57	45.39
PromptFL	8.2K	0.78	78.75	69.95	73.61
PromptFolio	8.2K	0.78	85.75	50.84	61.85
FedPGP	16.4K	1.56	87.88	64.38	72.86
BLEND	6.4K	0.61	91.01	73.37	**80.75**

Also, Tables 7 and 8 report results under comparable trainable parameter budgets. In Table 7 (high-budget setting), BLEND again achieves the best balance between personalization and generalization. In Table 8 (low-budget setting), BLEND consistently outperforms competing methods despite using fewer trainable parameters, highlighting its efficiency and effectiveness under constrained resources.

Table 7: Comparable trainable parameters (high budget).

Method	Total Number of Trainable Params.	Per.	Gen.	H-Mean
MMA+FedAVG	675K	82.69	73.14	77.34
pFedMMA	675K	77.99	66.69	71.48
BLEND	536K	92.67	74.15	**81.79**

Table 8: Comparable trainable parameters (low budget).

Method	Total Number of Trainable Params.	Per.	Gen.	H-Mean
pFedMOAP	74.2K	92.03	66.48	75.98
FedPHA	16.4K	92.35	47.90	61.15
FedOTP	16.4K	85.92	32.57	45.39
PromptFL	8.2K	78.75	69.95	73.61
PromptFolio	8.2K	85.75	50.84	61.85
FedPGP	20.6K	87.88	64.38	72.86
BLEND	13.3K	85.32	73.99	**79.01**

We further investigate the computational efficiency of BLEND by comparing its per-round wall-clock time and average GPU memory usage with baselines. As shown in Table 9, BLEND requires 11.18 seconds per round and 0.62 GB of GPU memory. These results demonstrate that computational requirements and GPU memory usage of BLEND are comparable to those of existing methods.

Table 9: Comparison of per-round wall-clock time and average GPU memory usage across FL methods.

Metric	BLEND	pFedMMA	MMA+FedAvg	FedPHA	FedOPT	FedPGP	PromptFL	pFedMOAP	PromptFolio
Time (s)	11.18	11.53	11.59	13.35	9.70	15.19	10.30	10.01	10.11
Mean GPU (GB)	0.62	0.60	0.61	0.63	0.59	1.28	0.71	0.76	0.74

B Data Heterogeneity Variations

To evaluate the performance of BLEND under various levels of data heterogeneity, here we change the Dirichlet distribution parameter α^{Dir} to include the values $\{10, 0.1, 0.01\}$ **over which the data heterogeneity increases** (smaller values of α^{Dir} correspond to higher levels of data heterogeneity, and thus the x-axis in all plots is shown in descending order of α^{Dir}). We plot the personalization and generalization performances and the H-Mean of the two under each level of data heterogeneity across datasets in Figs. 5-14. This general performance overview is depicted in Fig. 4 where the averaged performances under all datasets are shown.

Across all experiments, BLEND consistently achieves the highest H-Mean, indicating a superior balance between personalization and generalization. More importantly, the performance gap between BLEND and the baselines widens as data heterogeneity becomes more severe, as evidenced by the H-Mean curves. This trend highlights the robustness of BLEND under harsh non-IID data conditions (most baseline methods exhibit a sharp decline in H-Mean as data heterogeneity increases, whereas BLEND maintains stable performance even in highly data heterogeneous settings).

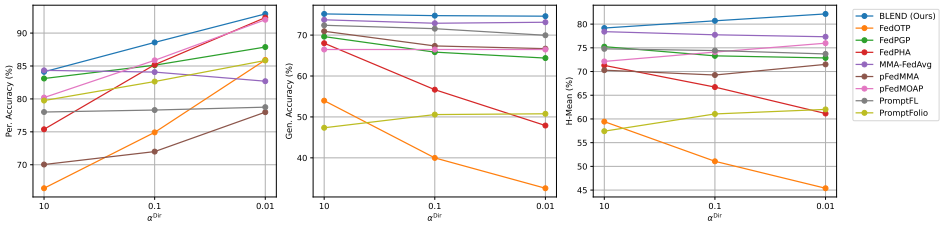


Figure 4: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of α^{Dir} (data heterogeneity) **averaged across all datasets**.

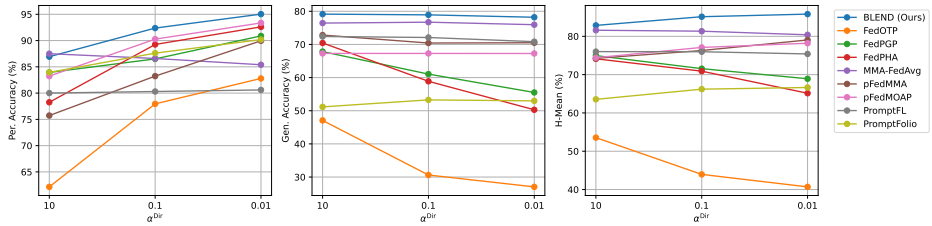


Figure 5: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of α^{Dir} (data heterogeneity) for the ***UCF101*** dataset.

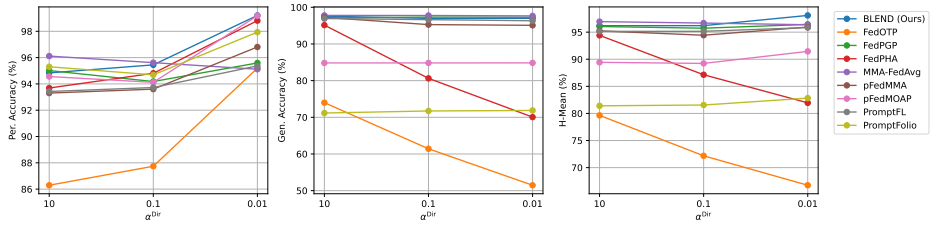


Figure 6: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of α^{Dir} (data heterogeneity) for the ***OxfordPets*** dataset.

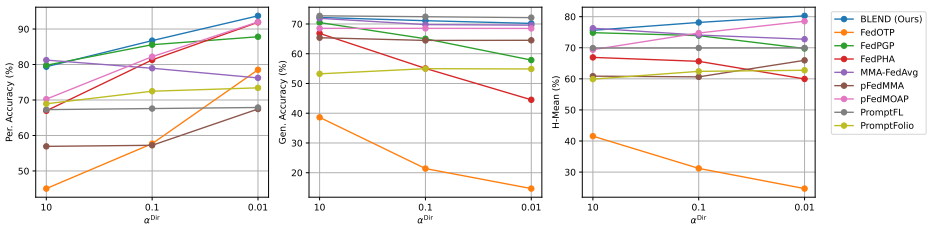


Figure 7: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of α^{Dir} (data heterogeneity) for the ***Stanford-Cars*** dataset.

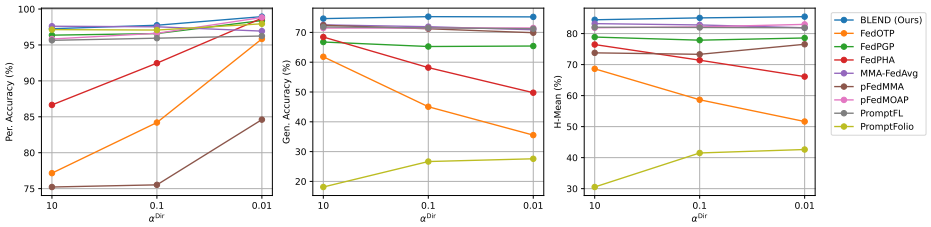


Figure 8: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of α^{Dir} (data heterogeneity) for the ***Flowers*** dataset.

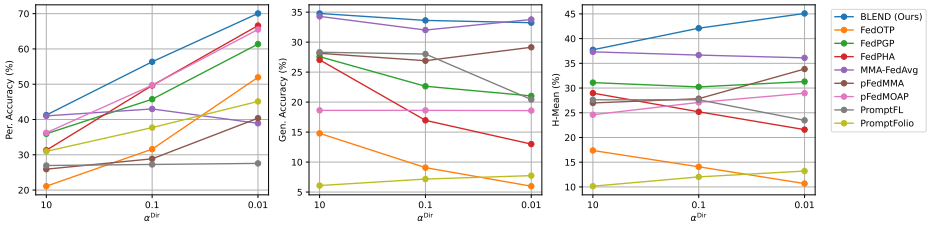


Figure 9: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of α^{Dir} (data heterogeneity) for the ***FGVC Aircraft*** dataset.

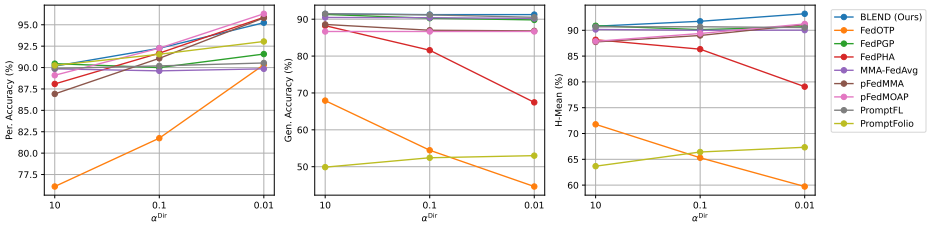


Figure 10: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of α^{Dir} (data heterogeneity) for the ***Food101*** dataset.

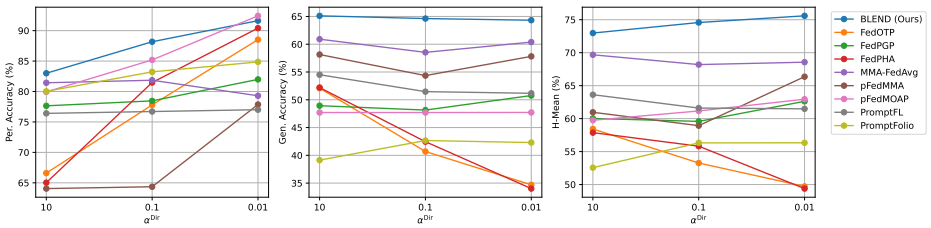


Figure 11: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of α^{Dir} (data heterogeneity) for the ***DTD*** dataset.

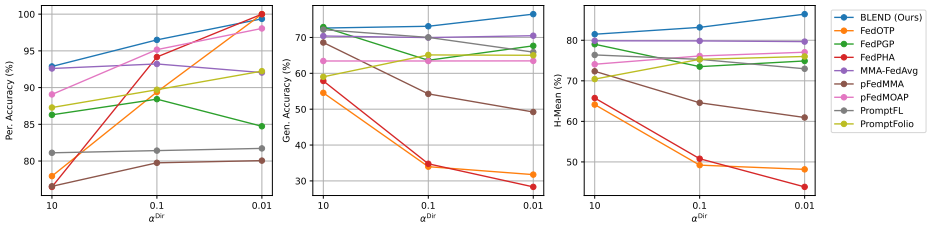


Figure 12: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of α^{Dir} (data heterogeneity) for the ***EuroSAT*** dataset.

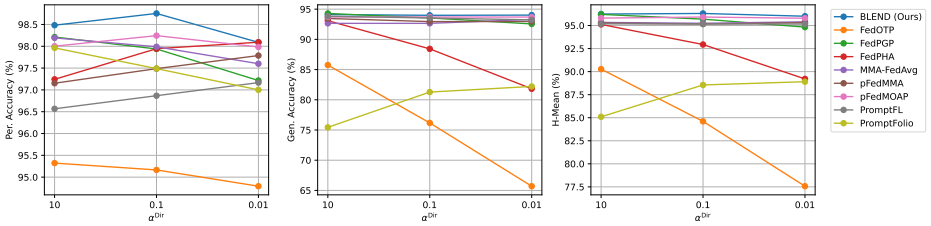


Figure 13: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of α^{Dir} (data heterogeneity) for the ***Caltech*** dataset.

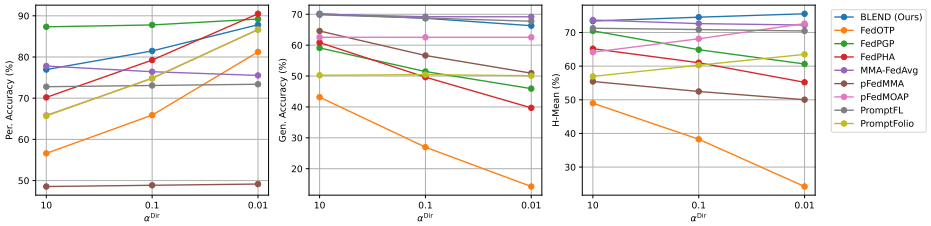


Figure 14: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of α^{Dir} (data heterogeneity) for the ***ImageNet*** dataset.

C Varying the Number of Clients

To extend our experiments regarding the number of clients that collectively train the global model, we repeated the experiments as in Table 1 with larger numbers of clients $N \in \{20, 30\}$, the results of which we include in Tables 10 and 11, respectively. Inspecting the results in these Tables along with Table 1 in the main text, we can observe that the H-Mean performance of BLEND is consistently better than all the baselines in all the datasets.

More importantly, the performance margin between BLEND and the strongest baseline increases as the number of clients grows: the average gain rises from 4.81% for $N = 10$ (Table 1), to 5.64% for $N = 20$ (Table 10), and further to 6.11% for $N = 30$ (Table 11). This trend can be attributed to the fact that increasing the number of clients typically amplifies local data bias and heterogeneity: by jointly leveraging globally aggregated adapters and locally personalized projection heads, BLEND is better able to mitigate this increased heterogeneity, resulting in improved robustness and scalability as the number of clients increases.

Table 10: Results for a scenario with $N = 20$ clients. Performance comparisons (measured in accuracy, %) between BLEND and baselines. The row Δ denotes the performance gap between BLEND and the best-performing baseline. Standard deviations across three random seeds are shown through sub-indices. Statistical significance is assessed using a two-sided paired t-test on seed-level H-Mean scores between BLEND and the strongest baseline, with improvements statistically significant at $p < 0.05$.

	UCF101			Oxford Pets			Stanford Cars			Flowers			Imagenet			FGVC Aircraft		
	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean
Zero-shot CLIP	68.10±0.00	75.18±0.00	71.46±0.00	89.47±0.00	96.81±0.00	93.00±0.00	63.77±0.00	**74.94±0.00**	68.91±0.00	69.42±0.00	**76.74±0.00**	72.90±0.00	72.40±0.00	68.08±0.00	70.17±0.00	27.49±0.00	33.05±0.00	30.02±0.00
MMA+FedAvg	79.22±1.17	77.27±0.98	78.23±0.83	95.96±0.10	97.50±0.11	96.73±0.04	75.86±0.71	69.71±0.79	72.66±0.74	94.31±0.29	72.29±1.72	81.83±1.02	77.15±0.22	**69.10±0.20**	72.90±0.07	35.34±0.13	34.67±0.45	35.00±0.26
pFedMMA	92.85±0.51	69.35±0.70	79.40±0.53	97.19±0.96	94.97±0.24	96.06±0.37	77.73±2.77	62.93±0.54	69.52±0.92	86.38±1.12	72.15±0.29	78.62±0.46	70.39±2.26	58.47±0.85	63.87±1.44	40.40±1.40	28.13±0.45	33.16±0.60
pFedMoAP	93.94±0.35	67.22±0.01	78.36±0.12	99.63±0.09	84.78±0.01	91.61±0.04	92.49±0.54	68.45±0.01	78.67±0.20	96.73±0.63	71.44±0.01	82.18±0.22	90.21±0.00	62.49±0.00	73.84±0.00	**73.23±2.16**	18.53±0.00	29.58±0.18
FedPHA	93.37±0.56	51.90±3.30	66.64±2.58	**99.82±0.06**	56.42±3.71	72.02±3.02	93.80±0.70	41.06±0.10	57.12±0.03	95.07±0.46	43.03±1.25	59.23±1.28	**91.60±0.01**	34.09±0.50	49.69±0.53	71.86±0.00	8.03±0.00	14.44±0.00
FedOTP	79.39±0.65	24.91±1.17	37.91±1.44	95.80±1.39	44.30±7.16	60.17±6.68	76.06±1.12	13.87±1.33	23.43±1.90	88.41±1.60	36.05±3.59	51.11±3.74	84.25±0.13	13.53±0.07	23.31±0.11	57.40±2.63	5.82±1.09	10.55±1.85
PromptFL	81.56±1.36	71.84±2.49	76.36±1.33	96.74±0.04	96.85±0.65	96.79±0.35	71.06±0.43	70.53±1.37	70.78±0.52	96.29±0.62	70.14±0.42	81.16±0.40	75.13±0.17	68.96±0.04	71.91±0.10	34.46±1.51	24.86±6.81	28.46±5.12
FedPGP	82.64±0.16	68.44±2.20	74.85±1.28	96.88±0.05	**97.53±0.03**	97.20±0.01	85.11±1.15	60.90±2.11	70.96±1.03	91.79±1.75	69.20±0.59	78.91±1.03	88.11±0.09	46.39±0.06	60.78±0.10	43.49±1.84	25.16±2.17	31.86±2.23
PromptFolio	87.92±0.57	53.14±3.92	66.14±2.89	97.91±0.16	71.90±4.39	82.83±2.87	77.11±2.96	54.88±5.78	64.05±4.97	94.48±0.96	27.70±6.70	42.36±7.94	86.97±1.33	50.86±3.96	61.68±3.76	51.52±2.86	7.74±0.42	13.44±0.54
BLEND	**94.59±0.09**	**78.84±0.16**	**86.00±0.13**	99.65±0.03	96.58±0.09	**98.09±0.06**	**94.30±0.21**	73.71±0.02	**82.75±0.09**	**98.15±0.14**	75.74±0.23	**85.50±0.17**	90.49±0.13	68.57±0.16	**78.02±0.15**	70.08±0.61	**34.92±0.68**	**46.61±0.46**
Δ ($p-$value)			+6.60 (0.017)			+0.89 (0.008)			+10.09 (0.0.013)			+3.32 (0.003)			+4.18 (0.002)			+11.61 (0.003)

	Food101			DTD			EuroSAT			Caltech			Cross-Dataset Avg. (↑)			Cross-Dataset Std. (↓)		
	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean
Zero-shot CLIP	89.44±0.00	90.67±0.00	90.05±0.00	54.75±0.00	58.94±0.00	56.76±0.00	50.21±0.00	69.87±0.00	58.43±0.00	97.16±0.00	94.21±0.00	95.66±0.00	68.22±0.00	73.85±0.00	70.74±0.00	19.86±0.00	17.81±0.00	18.72±0.00
MMA+FedAvg	89.83±0.07	91.33±0.03	90.57±0.05	78.90±1.53	**62.48±1.87**	69.72±1.48	92.01±0.91	68.89±6.26	78.64±4.15	97.53±0.33	93.78±0.32	95.62±0.33	81.61±0.55	73.70±1.28	77.19±0.90	17.33±0.48	17.33±1.77	16.73±1.18
pFedMMA	**96.99±0.14**	88.32±0.39	92.45±0.27	86.40±1.31	58.11±1.05	69.47±0.37	74.33±17.75	49.26±6.61	58.98±10.22	98.08±0.41	93.24±0.11	95.60±0.24	82.07±2.86	67.49±1.12	73.71±1.54	16.77±5.02	19.87±1.85	18.38±2.91
pFedMoAP	96.09±0.10	86.62±0.01	91.11±0.04	94.33±0.50	47.66±0.01	63.32±0.12	98.26±1.11	63.43±0.00	77.09±0.34	99.42±0.31	93.64±0.01	96.44±0.14	93.43±0.58	66.43±0.01	76.22±0.14	**7.32±0.61**	20.46±0.00	18.06±0.10
FedPHA	96.28±0.27	60.36±1.78	74.18±1.27	93.58±0.50	29.94±1.22	45.36±1.46	99.91±0.09	32.08±4.97	48.34±5.69	98.82±0.00	83.07±0.00	90.27±0.00	93.41±0.27	44.00±1.68	57.73±1.59	7.69±0.25	19.37±1.66	19.48±1.69
FedOTP	88.04±1.04	43.88±1.44	58.55±1.13	88.78±0.19	26.71±1.12	41.06±1.33	**100.00±0.00**	32.31±3.50	48.73±4.07	96.19±1.43	67.10±2.23	79.05±2.03	85.43±1.02	30.85±2.27	43.39±2.43	11.72±0.77	17.21±1.93	19.46±1.80
PromptFL	90.01±0.06	90.94±0.10	90.48±0.04	80.21±0.48	52.74±2.44	63.61±1.94	86.70±2.21	64.51±6.56	73.70±3.49	97.49±0.39	92.10±0.89	94.71±0.35	80.96±0.73	70.35±2.18	74.80±1.36	17.80±0.68	20.04±2.40	18.62±1.61
FedPGP	90.68±0.20	**91.50±0.13**	91.09±0.17	79.80±1.21	51.30±2.32	62.40±1.35	84.39±1.13	70.24±10.55	76.16±5.85	97.05±0.17	**94.57±0.25**	95.79±0.21	83.99±0.76	67.52±2.04	74.00±1.32	14.56±0.69	21.83±3.00	18.52±1.66
PromptFolio	92.94±0.21	52.98±6.14	67.29±5.05	89.00±1.10	42.21±4.35	57.10±3.78	94.78±0.82	64.94±6.65	76.84±4.42	96.57±0.72	82.15±5.08	88.67±2.66	86.92±1.15	50.85±4.83	62.04±3.90	13.84±0.99	21.39±1.83	21.56±1.94
BLEND	95.89±0.06	91.46±0.06	**93.62±0.03**	**94.71±0.37**	62.47±0.49	**75.28±0.46**	99.07±0.33	**75.30±2.19**	**85.54±1.36**	**99.62±0.06**	94.25±0.01	**96.86±0.03**	**93.65±0.20**	**75.18±0.41**	**82.83±0.29**	8.33±0.17	**17.11±0.63**	**14.01±0.39**
Δ ($p-$value)			+1.17 (0.001)			+5.56 (0.005)			+8.45 (0.028)			+0.42 (0.012)			+5.64			-2.72

Table 11: Results for a scenario with $N = 30$ clients. Performance comparisons (measured in accuracy, %) between BLEND and baselines. The row Δ denotes the performance gap between BLEND and the best-performing baseline. Standard deviations across three random seeds are shown through sub-indices. Statistical significance is assessed using a two-sided paired t-test on seed-level H-Mean scores between BLEND and the strongest baseline, with improvements statistically significant at $p < 0.05$.

	UCF101			Oxford Pets			Stanford Cars			Flowers			Imagenet			FGVC Aircraft		
	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean
Zero-shot CLIP	68.10±0.00	75.18±0.00	71.46±0.00	89.47±0.00	96.81±0.00	93.00±0.00	63.77±0.00	**74.94±0.00**	68.91±0.00	69.42±0.00	**76.74±0.00**	72.90±0.00	72.40±0.00	68.08±0.00	70.17±0.00	27.49±0.00	33.05±0.00	30.02±0.00
MMA+FedAvg	77.84±0.74	77.54±1.06	77.68±0.55	94.78±0.23	**97.50±0.05**	96.12±0.10	72.45±1.07	71.57±0.40	72.00±0.49	92.89±0.49	73.19±1.61	81.86±0.82	77.57±0.25	69.20±0.30	73.15±0.28	32.25±0.96	35.15±1.04	33.64±0.98
pFedMMA	95.70±0.45	71.16±0.56	81.63±0.39	92.72±2.16	95.14±0.22	93.90±1.02	78.41±2.96	58.46±1.80	66.94±1.61	92.62±1.16	71.28±0.81	80.55±0.28	71.74±1.02	55.29±1.04	62.45±1.05	64.05±3.77	27.65±0.43	38.61±1.10
pFedMoAP	94.82±0.09	67.21±0.00	78.66±0.03	98.66±0.11	84.78±0.00	91.20±0.05	90.56±0.63	68.44±0.00	77.96±0.23	98.46±0.66	71.43±0.00	82.80±0.24	90.83±0.00	62.50±0.00	74.05±0.00	81.52±1.10	18.53±0.00	30.19±0.08
FedPHA	**96.38±0.56**	47.72±2.62	63.78±2.22	98.86±0.20	49.07±2.02	65.56±1.76	**94.59±0.18**	39.27±0.35	55.50±0.38	98.83±0.57	33.60±0.66	50.15±0.66	**91.68±0.02**	36.76±0.36	52.48±0.37	**83.14±0.65**	7.11±1.30	13.06±2.20
FedOTP	89.03±2.76	23.42±2.80	36.95±3.18	96.83±0.69	39.18±6.72	55.41±6.99	74.52±0.52	16.43±1.26	26.91±1.73	92.48±0.83	32.76±3.04	48.32±3.38	84.50±0.47	12.47±0.61	21.72±0.94	68.05±2.58	5.42±1.65	9.99±2.88
PromptFL	81.25±0.61	72.15±2.31	76.40±1.08	95.84±0.15	95.34±1.68	95.58±0.92	70.92±1.73	71.61±1.37	71.25±1.33	94.68±0.35	68.23±2.08	79.29±1.42	75.37±0.09	69.33±0.04	72.22±0.06	28.18±9.06	20.54±10.59	23.38±10.48
FedPGP	79.50±2.05	70.87±0.05	74.92±0.88	95.75±0.29	96.51±1.10	96.13±0.69	77.31±0.04	68.15±0.64	72.44±0.38	91.11±1.57	72.36±0.18	80.66±0.73	87.05±0.10	46.79±0.08	60.86±0.11	40.39±2.37	26.65±3.00	31.89±1.43
PromptFolio	88.63±1.07	53.14±3.92	66.32±2.77	96.74±0.03	71.90±4.39	82.41±2.90	75.55±2.35	54.88±5.78	63.49±4.72	96.11±0.39	27.70±6.70	42.55±8.06	87.19±0.32	50.86±5.70	63.00±7.46	53.79±0.07	7.74±0.42	13.53±0.64
BLEND	95.95±0.18	**78.20±0.14**	**86.17±0.03**	**99.03±0.03**	96.74±0.08	**97.87±0.04**	93.87±0.28	74.34±0.10	**82.97±0.09**	**98.97±0.08**	76.49±0.19	**86.29±0.15**	90.56±0.04	**70.24±0.21**	**79.12±0.12**	73.34±0.31	**35.36±0.26**	**47.71±0.17**
Δ ($p-$value)			+4.54 (0.021)			+1.74 (0.007)			+5.01 (0.012)			+3.49 (0.004)			+5.07 (0.001)			+9.10 (0.003)

	Food101			DTD			EuroSAT			Caltech			Cross-Dataset Avg. (↑)			Cross-Dataset Std. (↓)		
	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean	Per.	Gen.	H-Mean
Zero-shot CLIP	89.44±0.00	90.67±0.00	90.05±0.00	54.75±0.00	58.94±0.00	56.76±0.00	50.21±0.00	69.87±0.00	58.43±0.00	97.16±0.00	94.21±0.00	95.66±0.00	68.22±0.00	73.85±0.00	70.74±0.00	19.86±0.00	17.81±0.00	18.72±0.00
MMA+FedAvg	89.56±0.31	**91.45±0.06**	90.50±0.17	72.85±1.62	**64.94±0.80**	68.65±0.51	93.05±0.84	72.08±5.64	81.10±3.34	95.95±0.10	94.40±0.22	95.17±0.08	79.92±0.66	74.70±1.12	76.99±0.73	18.17±0.45	17.04±1.58	17.05±0.92
pFedMMA	**97.45±0.19**	88.81±0.34	92.93±0.25	85.46±1.87	59.55±0.63	70.18±0.35	79.53±11.47	50.52±6.17	61.69±7.79	99.45±0.09	93.67±0.11	96.48±0.10	85.71±2.51	67.15±1.21	74.54±1.39	11.31±3.19	20.25±1.71	17.20±2.18
pFedMoAP	96.58±0.17	86.62±0.00	91.33±0.08	96.21±0.32	47.66±0.00	63.74±0.07	96.40±1.29	63.41±0.00	76.50±0.41	99.25±0.13	93.64±0.00	96.36±0.06	94.33±0.45	66.42±0.00	76.28±0.12	5.13±0.43	20.46±0.00	17.86±0.12
FedPHA	**97.45±0.08**	50.01±0.67	66.10±0.57	**97.26±0.12**	27.78±1.74	43.19±2.09	99.86±0.14	31.07±1.62	47.37±1.91	**99.72±0.02**	77.24±3.31	87.01±2.11	**95.78±0.25**	39.96±1.47	54.42±1.43	**4.84±0.23**	17.30±0.94	18.22±0.78
FedOTP	91.31±0.90	38.40±1.38	54.05±1.48	96.03±0.29	23.42±0.72	37.66±0.95	**100.00±0.00**	31.36±1.86	47.72±2.16	96.67±0.23	63.29±4.86	76.39±3.47	88.94±0.93	28.62±2.49	41.51±2.72	9.88±0.91	**15.61±1.85**	18.13±1.69
PromptFL	90.06±0.25	91.04±0.67	90.55±0.35	79.45±0.36	50.24±2.94	61.51±2.33	88.32±0.74	68.45±2.98	77.09±1.92	97.14±0.07	93.74±0.73	95.41±0.36	80.12±1.34	70.07±2.54	74.27±2.03	19.29±2.61	21.21±2.84	19.92±2.90
FedPGP	90.47±0.06	91.40±0.01	90.93±0.03	76.60±3.49	50.51±1.86	60.78±0.25	85.60±2.48	**75.64±7.08**	80.23±5.09	97.10±0.33	**94.42±1.42**	95.73±0.57	82.09±1.27	69.33±1.53	74.46±1.01	15.49±1.21	21.43±2.07	18.58±1.42
PromptFolio	93.66±0.32	52.98±6.14	67.48±5.10	89.18±0.35	42.21±4.35	57.16±3.94	91.86±0.40	64.94±6.65	75.87±4.44	98.69±0.31	82.15±5.08	89.59±3.15	87.14±0.59	50.85±4.83	62.14±3.97	13.43±0.68	21.39±1.83	21.53±1.92
BLEND	96.58±0.19	91.32±0.11	**93.88±0.06**	95.49±0.57	61.50±0.32	**74.81±0.17**	98.90±0.34	75.01±1.57	**85.31±1.09**	99.61±0.06	94.17±0.04	**96.82±0.05**	94.23±0.21	**75.34±0.30**	**83.10±0.20**	7.46±0.16	17.01±0.43	**13.72±0.30**
Δ ($p-$value)			+2.55 (0.002)			+4.63 (0.004)			+4.21 (0.032)			+0.34 (0.011)			+6.11			-3.33

D Large-Scale Setting with Real-World Data Distribution

We further evaluate BLEND under realistic, large-scale client data heterogeneity to better reflect practical deployment scenarios. To this end, we use the real-world iNaturalist-User-120k dataset [43], and conduct experiments with $N = 100$ and $N = 200$ sampled clients, where each client follows naturally occurring data distributions reported by the dataset. We use the same network and learning parameters (e.g., λ and γ) as the simulations presented in the main text.

The results are summarized in Table 12 and Table 13. For $N = 100$ clients (Table 12), BLEND achieves the highest H-Mean of 63.37, outperforming all baselines by a clear margin. This improvement stems from simultaneously maintaining strong personalization (Base = 86.96) and better generalization to unseen classes (New = 49.84). In contrast, while some methods such as pFedMOAP achieve high personalization, they suffer from significantly lower generalization, leading to inferior H-Mean.

As the number of clients increases to $N = 200$ (Table 13), all methods experience a drop in performance due to increased data heterogeneity and reduced per-client data. Nevertheless, BLEND achieve the best performance, with an H-Mean of 57.39, maintaining a consistent gap over baselines.

Overall, these results demonstrate that BLEND scales effectively to large client populations and remains robust under realistic, highly non-IID data distributions, consistently achieving a superior balance between personalization and generalization compared to existing methods.

Table 12: $N = 100$ clients

Method	Per.	Gen.	H-Mean
FedOTP	63.31	11.46	19.32
FedPGP	81.44	36.76	50.64
FedPHA	81.72	21.75	34.33
MMA+FedAVG	69.99	50.13	58.41
pFedMMA	67.06	48.15	56.05
pFedMOAP	89.55	37.89	53.25
PromptFL	77.78	43.96	56.17
PromptFolio	82.36	32.96	47.07
BLEND	86.96	49.84	**63.37**

Table 13: $N = 200$ clients

Method	Per.	Gen.	H-Mean
FedOTP	68.35	10.72	18.54
FedPGP	82.03	34.68	48.63
FedPHA	86.78	17.11	28.58
MMA+FedAVG	64.06	43.50	51.82
pFedMMA	56.48	42.38	48.41
pFedMOAP	92.43	22.49	36.18
PromptFL	76.75	34.15	47.23
PromptFolio	82.03	25.12	38.46
BLEND	85.62	43.17	**57.39**

E Different Few-shot Settings

To further explore the effect of the number of shots seen from each class during the model training across the clients, we performed an ablation study sweeping the number of shots on the values $\{2, 4, 8, 16, 32\}$. We have performed this experiment for all of the datasets mentioned in the paper and included the results in Figs. 16-25. Also, Fig. 15 depicts the average performance across all the datasets.

Intuitively, as the number of shots increase, we would expect each participating client's local model to become more adapted to the local data distribution on that specific client, hence increasing personalization (this is reflected in Fig. 15-left and in most of the datasets shown in Figs. 16-25). Simultaneously, increasing the number of shots can result in the local models being more biased to their corresponding client's data distribution, hence damaging the generalization performance. This hypothesis is also confirmed in the generalization performance subplot of Fig. 15-middle and in most of the datasets shown in Figs. 16-25. One key takeaway is that BLEND's average performance across the datasets is better than all other methods when H-Mean performance is considered (Fig. 15-right). Further, across all individual datasets, BLEND's H-Mean performance is the best when the number of shots is less than or equal to 16, while being marginally outperformed in two datasets when the number of shots is 32. This demonstrates the exceptional capability of BLEND in limited few-shot learning settings.

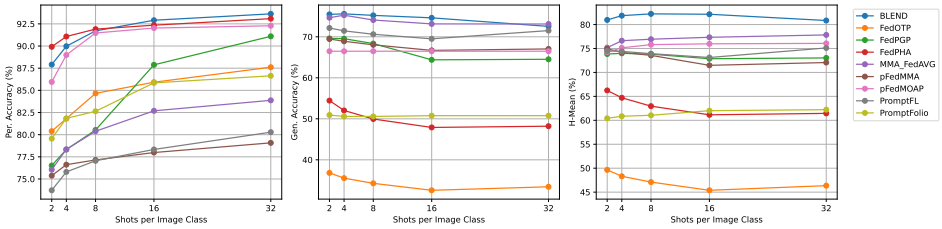


Figure 15: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing number of shots per class **averaged across all datasets**.

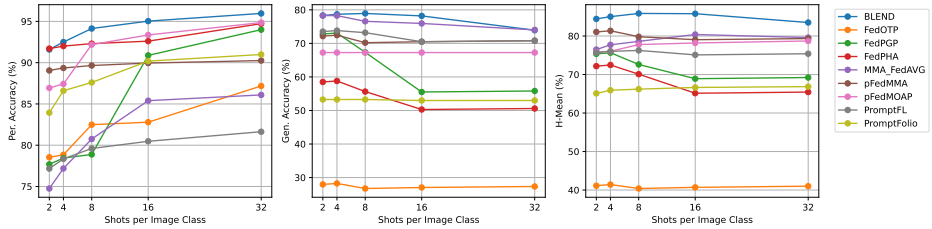


Figure 16: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing number of shots per class for the ***UCF101*** dataset.

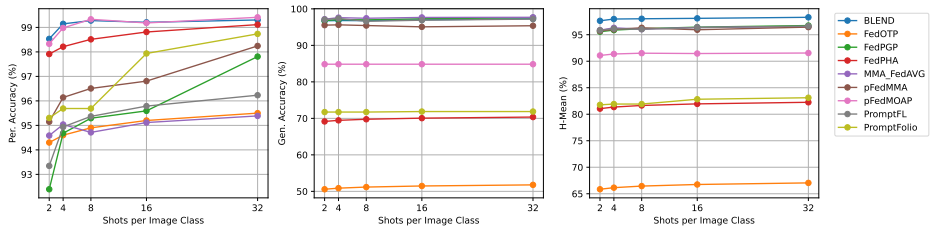


Figure 17: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing number of shots per class for the ***OxfordPets*** dataset.

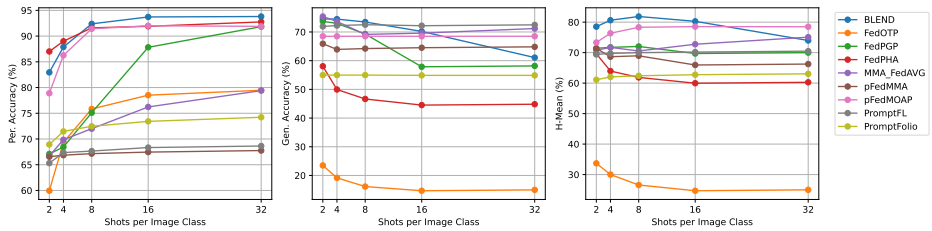


Figure 18: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing number of shots per class for the ***Stanford-Cars*** dataset.

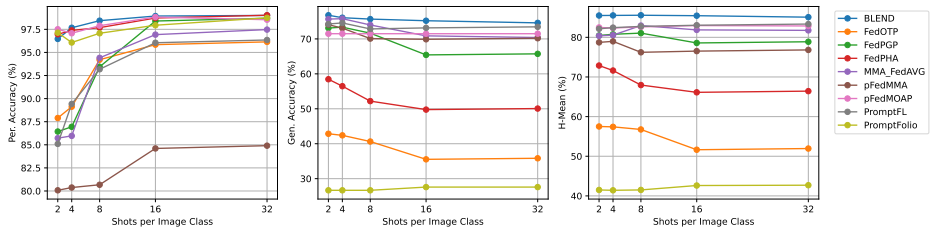


Figure 19: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing number of shots per class for the ***Flowers*** dataset.

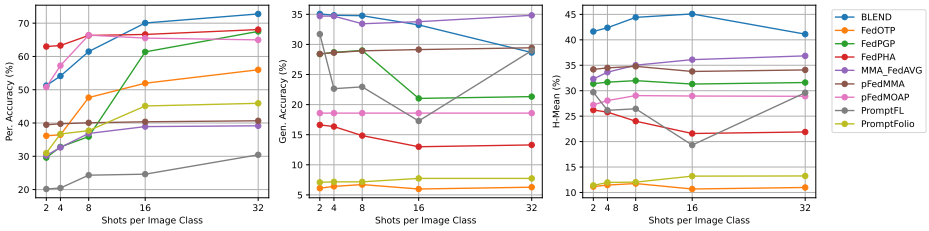


Figure 20: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing number of shots per class for the ***FGVC Aircraft*** dataset.

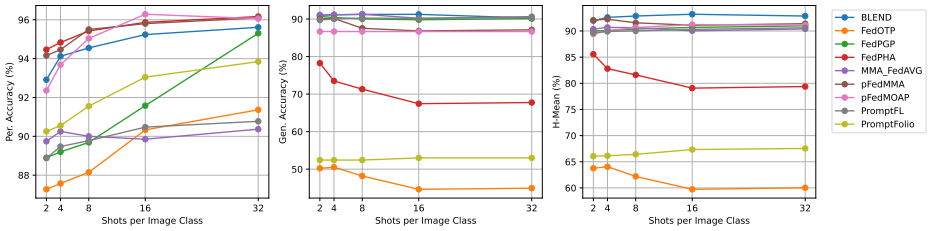


Figure 21: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing number of shots per class for the ***Food101*** dataset.

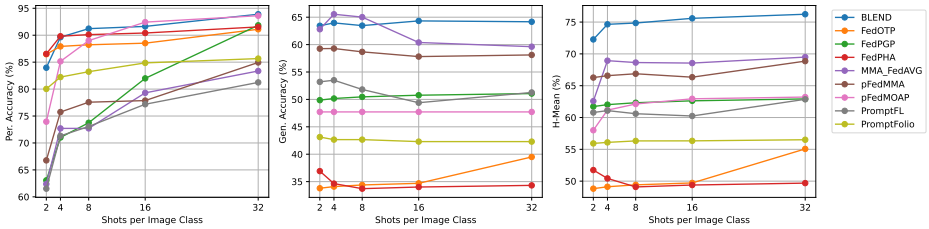


Figure 22: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing number of shots per class for the ***DTD*** dataset.

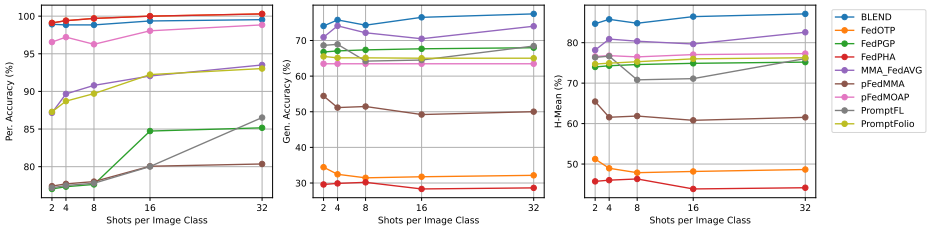


Figure 23: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing number of shots per class for the ***EuroSAT*** dataset.

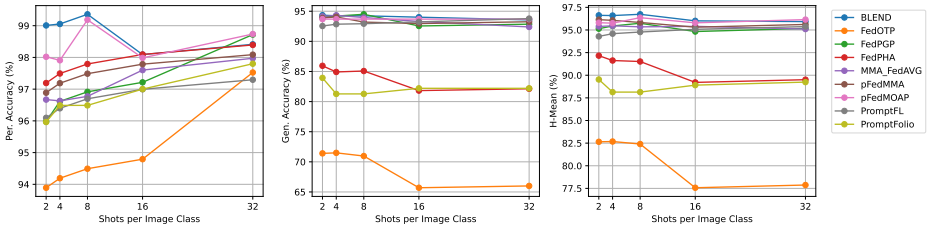


Figure 24: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing number of shots per class for the ***Caltech*** dataset.

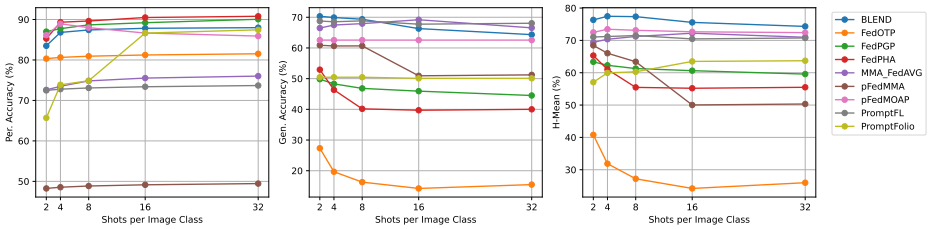


Figure 25: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing number of shots per class for the ***ImageNet*** dataset.

F Different Bottleneck Sizes for Global Adapters

We study the impact of global adapter bottleneck rank $r = r_x = r_w$ across all datasets by sweeping the rank $r \in \{\emptyset, 4, 8, 16, 32, 64, 128, 256, \text{Full-rank}\}$, where $\emptyset$ corresponds to the case where no adapter is used. We have summarized the result for each dataset in Tables 15-24, while the average performance across all the datasets is shown in Table 14.

Based on the complexity of the dataset under test, the best personalization and generalization performance can be achieved by employing different global adapter bottleneck sizes. Taking Flowers (Table 19) as an example, the best generalization performance occurs when no global adapter is employed while the best generalization performance is realized under full-rank bottleneck size for DTD dataset (Table 15). A similar phenomenon can be observed when focusing on the personalization performance. However, based on the average behavior observed in Table 14, we can infer that the performance that maintains a balance between personalization and generalization is realized under $r = 16$ which requires low computation and communication budget.

Table 14: **Avg across datasets.**

Rank (r)	Per.	Gen.	H-Mean
No Adapter	90.12	73.81	80.66
4	92.25	74.69	82.05
8	92.45	**74.70**	82.10
16	92.91	74.62	**82.15**
32	**92.93**	74.39	81.97
64	92.91	74.09	81.74
128	92.80	73.70	81.43
256	92.68	73.47	81.20
Full-rank	92.48	73.63	81.26

Table 15: DTD

Rank (r)	Per.	Gen.	H-Mean
No Adapter	85.93	59.66	70.42
4	91.32	64.21	75.40
8	91.46	64.26	75.49
16	91.64	64.32	75.59
32	**91.92**	65.12	76.23
64	91.89	64.69	75.93
128	91.51	64.98	76.00
256	91.59	65.33	76.26
Full-rank	91.49	**65.78**	**76.53**

Table 16: EuroSAT

Rank (r)	Per.	Gen.	H-Mean
No Adapter	98.31	66.50	79.33
4	**99.54**	72.16	83.66
8	99.39	74.73	85.31
16	99.36	**76.50**	**86.44**
32	99.48	76.16	86.27
64	99.34	75.42	85.74
128	99.20	73.74	84.60
256	99.02	72.14	83.47
Full-rank	98.59	71.04	82.57

Table 17: FGVC Aircraft

Rank (r)	Per.	Gen.	H-Mean
No Adapter	58.17	32.92	42.05
4	64.39	**34.54**	44.96
8	66.51	33.71	44.74
16	70.05	33.24	**45.09**
32	71.04	32.65	44.74
64	71.45	32.04	44.24
128	**72.47**	31.83	44.23
256	72.16	30.86	43.24
Full-rank	71.83	31.74	44.03

Table 18: Food101

Rank (r)	Per.	Gen.	H-Mean
No Adapter	95.24	90.51	92.82
4	**95.57**	**91.39**	**93.43**
8	95.44	91.34	93.34
16	95.24	91.25	93.20
32	95.10	91.02	93.02
64	94.70	90.64	92.62
128	94.29	90.33	92.27
256	94.23	90.35	92.25
Full-rank	93.82	90.15	91.95

Table 19: Flowers

Rank (r)	Per.	Gen.	H-Mean
No Adapter	94.23	**75.80**	84.01
4	98.99	74.92	85.29
8	98.72	75.32	85.45
16	98.94	75.21	85.46
32	99.11	74.99	85.38
64	**99.33**	75.09	**85.52**
128	99.10	74.91	85.32
256	99.11	74.07	84.78
Full-rank	99.08	74.33	84.94

Table 20: OxfordPets

Rank (r)	Per.	Gen.	H-Mean
No Adapter	98.19	95.86	97.01
4	99.14	96.79	97.95
8	99.14	96.81	97.96
16	**99.21**	96.99	98.09
32	99.13	97.21	98.16
64	98.98	97.21	98.09
128	98.99	97.35	**98.16**
256	98.84	97.35	98.09
Full-rank	98.85	**97.47**	98.15

Table 21: Stanford Cars

Rank (r)	Per.	Gen.	H-Mean
No Adapter	90.88	**73.39**	81.21
4	93.04	72.66	**81.60**
8	93.48	71.50	81.03
16	**93.72**	70.20	80.27
32	93.58	68.14	78.86
64	93.63	67.46	78.42
128	93.20	66.56	77.66
256	93.06	67.36	78.15
Full-rank	92.95	68.54	78.90

Table 22: UCF101

Rank (r)	Per.	Gen.	H-Mean
No Adapter	92.19	75.64	83.10
4	94.63	78.61	85.88
8	94.59	**78.68**	**85.90**
16	95.03	78.20	85.80
32	94.69	77.70	85.36
64	**95.21**	77.59	85.50
128	94.83	75.92	84.33
256	94.93	76.16	84.52
Full-rank	94.43	75.76	84.07

Table 23: ImageNet

Rank (r)	Per.	Gen.	H-Mean
4	87.81	67.28	**76.19**
8	87.73	66.51	75.66
16	**87.83**	66.31	75.56
32	87.11	66.96	75.72
64	86.53	66.79	75.39
128	86.37	67.37	75.70
256	85.85	67.37	75.50
Full-rank	85.81	**67.47**	75.54

Table 24: Caltech

Rank (r)	Per.	Gen.	H-Mean
No Adapter	97.97	94.04	95.97
4	**98.12**	**94.32**	**96.18**
8	98.08	94.18	96.09
16	98.09	94.01	96.01
32	98.11	93.92	95.97
64	98.07	93.98	95.98
128	98.05	94.04	96.00
256	97.96	93.72	95.79
Full-rank	97.96	94.02	95.95

G Different Bottleneck Sizes for Personalized Projection Adapter

To study the effect of the personalized projection adapter's bottleneck size, we have conducted a separate set of experiments, sweeping the bottleneck rank r_p over the range $r_p \in \{4, 8, 16, 32, 64, 128, 256, \text{Full-rank}\}$. The results for individual datasets are presented in Tables 26-35, while the average performance across all datasets is reported in Table 25.

Given the utility of the personalized projection adapter in our method (discussed in Sec. 3.2), it is expected that a more sophisticated structure for this adapter will result in better personalization performance. This hypothesis is verified as in 7 out of 10 cases in our experiments (Tables 26, 27, 29, 32, 33, 34, 35) the best personalization performance occurs under full-rank settings. On the other hand, the generalization performance often peaks under the two shallowest ranks $r_p \in \{4, 8\}$ (Tables 26, 27, 30, 31, 32), showing how dedication of relatively larger number of parameters to personalization in the trained global model can negatively affect the generalization performance.

Focusing on Table 25 that contains the average personalization and generalization performances alongside their H-Mean for all of the tested r_p values across all datasets, we observe that the best personalization performance is achieved under the Full-rank setting and the best balance between generalization and personalization (i.e., the H-Mean) is realized under one of the larger bottleneck ranks ($r_p = 256$). We can thus conclude that sophisticated personalized projection adapters, even though harmful to the generalization performance in some cases, have proven to be significantly beneficial in improving personalization and consequently the balance between personalization and generalization.

Table 25: **Avg across datasets.**

Rank (r_p)	Per.	Gen.	H-Mean
4	89.99	**75.00**	81.44
8	90.45	74.83	81.50
16	91.00	74.86	81.73
32	91.59	74.91	81.96
64	92.12	74.64	81.92
128	92.57	74.71	82.12
256	92.91	74.62	**82.15**
Full-rank	**92.95**	74.46	82.08

Table 26: DTD

Rank (r_p)	Per.	Gen.	H-Mean
4	88.15	**65.00**	74.83
8	89.21	64.56	74.91
16	89.79	64.71	75.22
32	90.13	64.77	75.38
64	90.81	64.80	75.63
128	91.40	64.67	75.75
256	91.64	64.32	75.59
Full-rank	**92.93**	63.93	**75.75**

Table 27: EuroSAT

Rank (r_p)	Per.	Gen.	H-Mean
4	97.32	**78.69**	**87.02**
8	98.08	77.86	86.81
16	98.40	77.39	86.64
32	98.86	77.55	86.92
64	98.96	75.57	85.70
128	99.22	76.38	86.31
256	99.36	76.50	86.44
Full-rank	**99.53**	76.19	86.31

Table 28: FGVC Aircraft

Rank (r_p)	Per.	Gen.	H-Mean
4	57.93	33.28	42.27
8	58.47	32.67	41.92
16	60.25	33.46	43.03
32	62.78	**33.78**	43.92
64	66.02	32.85	43.87
128	67.60	33.08	44.42
256	**70.05**	33.24	**45.09**
Full-rank	68.19	32.80	44.29

Table 29: Food101

Rank (r_p)	Per.	Gen.	H-Mean
4	94.75	91.26	92.97
8	94.79	91.22	92.97
16	94.87	91.28	93.04
32	94.97	91.21	93.06
64	95.15	91.13	93.10
128	95.22	**91.31**	93.22
256	95.24	91.25	93.20
Full-rank	**95.43**	91.16	**93.24**

Table 30: Flowers

Rank (r_p)	Per.	Gen.	H-Mean
4	98.71	75.17	85.35
8	98.83	**75.69**	**85.73**
16	98.65	75.52	85.55
32	99.01	75.54	85.70
64	99.00	75.34	85.56
128	**99.17**	75.31	85.61
256	98.94	75.21	85.46
Full-rank	99.05	75.48	85.68

Table 31: OxfordPets

Rank (r_p)	Per.	Gen.	H-Mean
4	98.44	**97.54**	97.99
8	98.87	97.44	**98.15**
16	99.02	97.23	98.12
32	99.08	97.21	98.14
64	99.12	96.93	98.02
128	99.19	96.99	98.08
256	**99.21**	96.99	98.09
Full-rank	99.20	96.99	98.08

Table 32: Stanford Cars

Rank (r_p)	Per.	Gen.	H-Mean
4	89.14	70.52	78.74
8	90.29	**70.87**	79.41
16	91.13	70.70	79.62
32	92.21	70.75	80.07
64	92.85	70.45	80.11
128	93.47	70.55	**80.41**
256	93.72	70.20	80.27
Full-rank	**93.80**	69.53	79.86

Table 33: UCF101

Rank (r_p)	Per.	Gen.	H-Mean
4	93.27	77.55	84.69
8	92.94	77.77	84.68
16	93.82	77.62	84.96
32	94.22	78.17	85.45
64	94.17	**78.29**	85.50
128	94.87	78.18	85.72
256	95.03	78.20	**85.80**
Full-rank	**95.22**	77.87	85.68

Table 34: ImageNet

Rank (r_p)	Per.	Gen.	H-Mean
4	84.23	66.76	74.48
8	85.04	66.15	74.41
16	86.11	66.51	75.05
32	86.61	66.28	75.09
64	87.02	**66.78**	75.57
128	87.42	66.38	75.46
256	87.83	66.31	75.56
Full-rank	**88.00**	66.49	**75.75**

Table 35: Caltech

Rank (r_p)	Per.	Gen.	H-Mean
4	97.97	94.20	96.05
8	98.00	94.10	96.01
16	97.92	94.19	96.02
32	97.98	93.85	95.87
64	98.06	94.23	96.11
128	98.15	**94.28**	**96.17**
256	98.09	94.01	96.01
Full-rank	**98.16**	94.19	96.13

H Adapter Injection at Different Layers

We further explore the effect of injecting adapters throughout the backbone. To this end, we conduct a series of experiments through which we sweep different depths and placements of the adapters between the transformer blocks of the backbone. Our experiments include the following start $-$ end (i.e., $\ell^{\mathsf{S}} - \ell^{\mathsf{E}}$) combinations for transformer blocks in which we have used adapters: $\{1-1, 1-2, 1-3, \cdots, 1-11, 1-12, 2-12, 3-12, \cdots, 11-12, 12-12\}$. Our results for these experiments regarding the personalization, generalization, and H-Mean of these two are included in Figs. 27-35, and the average performance across all the datasets is reported in Fig. 26.

The H-Mean curves (i.e., the right sub-plots) in Figs. 27–35 indicate that the optimal placement of the adapters depends on dataset-specific characteristics. In some datasets, the most important features are captured by the early transformer layers, in which case inserting adapters in those layers yields the best performance (e.g., Fig. 29-right). In contrast, some datasets may benefit more from adapters in the later transformer layers, suggesting that their key features emerge deeper in the network (e.g., Fig. 30-right). Nevertheless, when performance is averaged across all datasets (Fig. 26-right), the peak H-Mean is achieved when adapters are distributed across a broader portion of the backbone (layers 1 to 12), highlighting the advantage of multi-layer adaptation. Further, from Fig. 26-right, it can be seen that a comparable H-Mean performance can be achieved when inserting the adapters from layer 5 to 12 (i.e., under a lower computation/communication overhead).

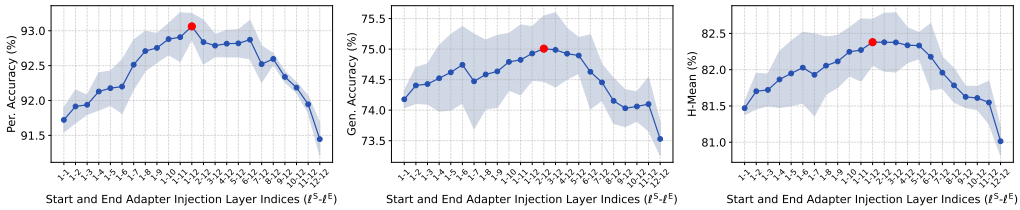


Figure 26: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing adapter injection settings **averaged across all datasets**.

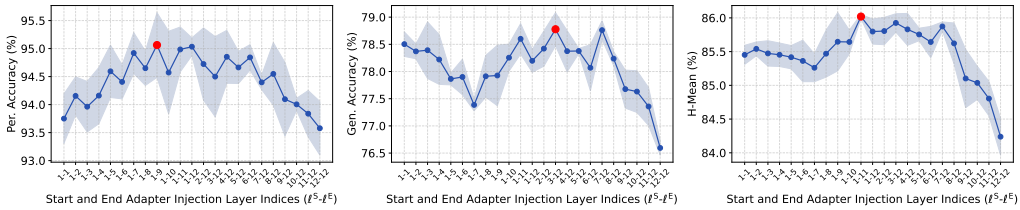


Figure 27: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing adapter injection settings for the ***UCF101*** dataset.

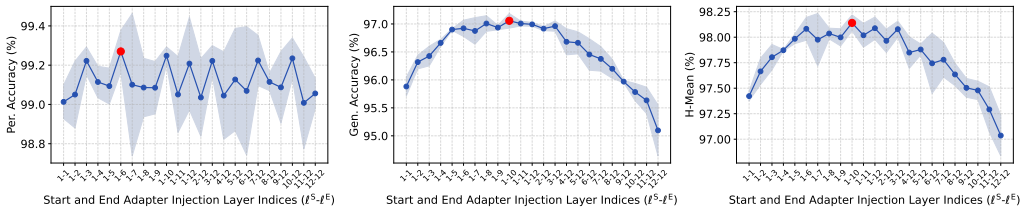


Figure 28: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing adapter injection settings for the ***OxfordPets*** dataset.

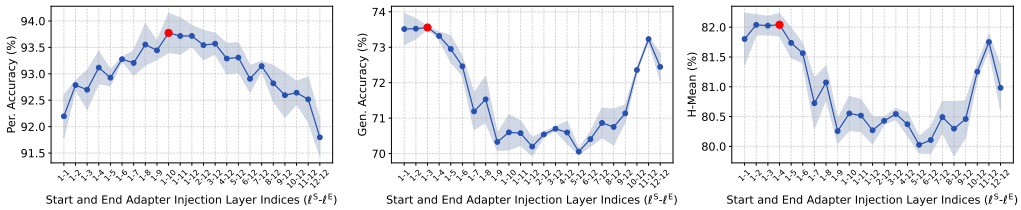


Figure 29: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing adapter injection settings for the ***Stanford-Cars*** dataset.

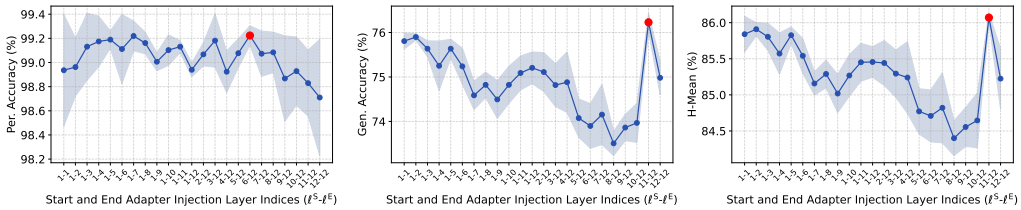


Figure 30: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing adapter injection settings for the ***Flowers*** dataset.

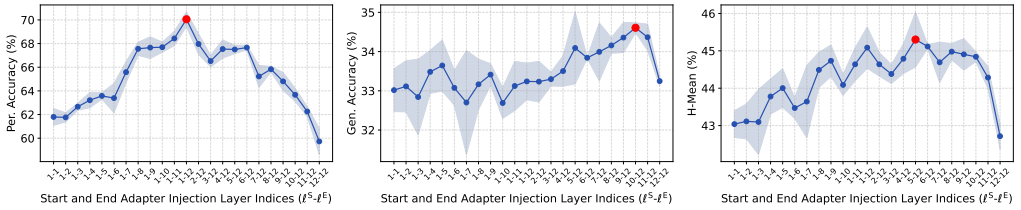


Figure 31: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing adapter injection settings for the ***FGVC Aircraft*** dataset.

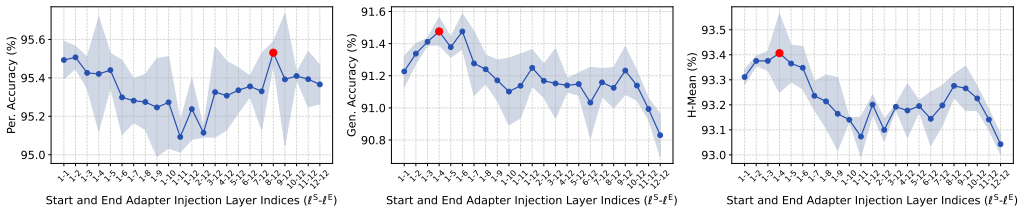


Figure 32: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing adapter injection settings for the ***Food101*** dataset.

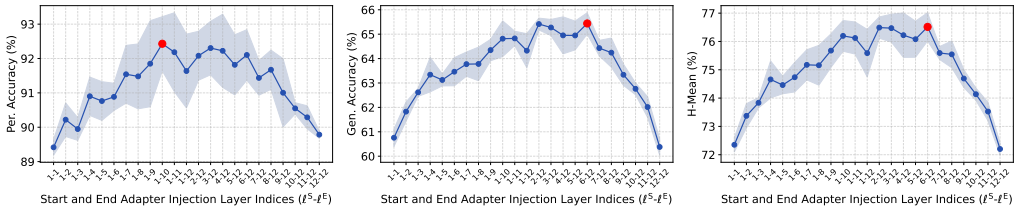


Figure 33: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing adapter injection settings for the ***DTD*** dataset.

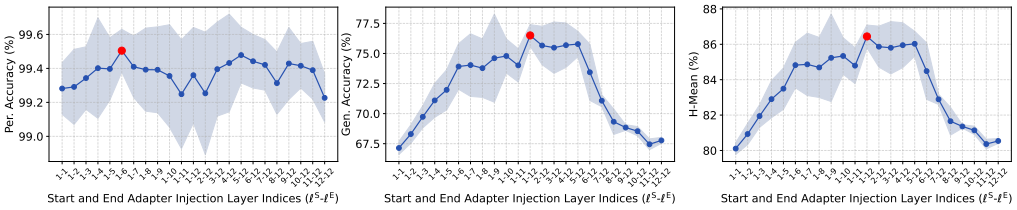


Figure 34: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing adapter injection settings for the ***EuroSAT*** dataset.

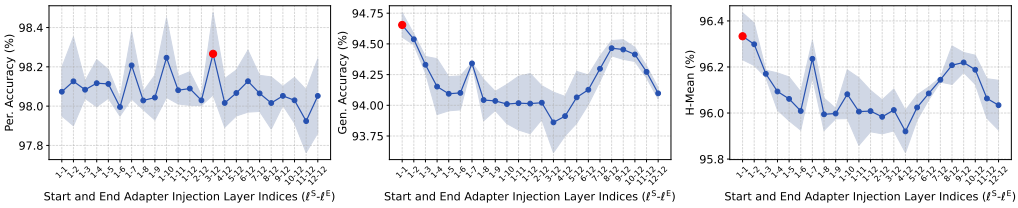


Figure 35: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing adapter injection settings for the ***Caltech*** dataset.

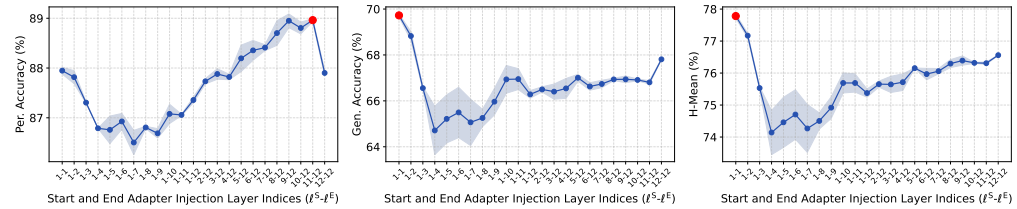


Figure 36: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing adapter injection settings for the ***ImageNet*** dataset.

I Effect of γ on Performance

To evaluate the effect of γ on the model performance, we reported the personalization, generalization, and the H-Mean of the two across the datasets in Figs. 38-47, while Fig. 37 reports the average performance across the datasets.

As explained in Eq. (17), the parameter γ corresponds to the emphasis of personalization vs. generalization in training. This is captured in our results as the personalization performance (left sub-plot in all result sets) improves for $\gamma > 0$ for all datasets compared to the case of $\gamma = 0$. Also, we observe a decline in the generalization performance for higher values of γ (middle sub-plot in all result sets). Focusing on the left sub-plot in the result set for the *UCF101* dataset (Fig. 38), we can see that the gap between the best ($\gamma = 0.4$) and worst ($\gamma = 0$) personalization performances is almost 10%, where the sharp increase in the personalization performance with γ increasing from 0 to a slightly higher value ($\gamma = 0.09$) shows the immediate effectiveness of adding the personalization term. Also, the right sub-plot of Fig. 38 shows the H-Mean performance where the optimal balance of personalization vs. generalization is reached at $\gamma = 0.09$.

Focusing on Fig. 37 that reports the average performance across the datasets, it can be observed that the best test accuracy considering both the generalization and personalization is achieved with small values of γ, indicating the equilibrium point of $\gamma = 0.09$ when the H-Mean performance is measured. The same trends can also be observed with minor fluctuations in the experiments on other datasets (Figs. 38-47).

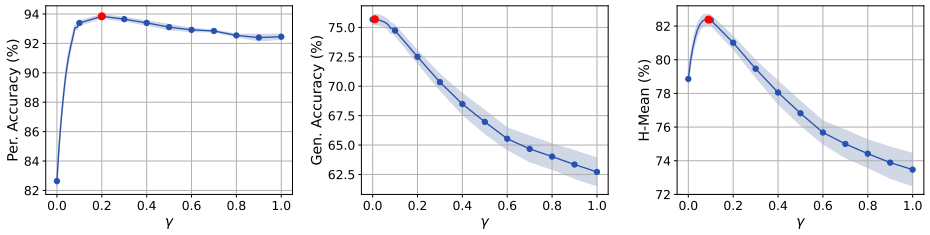


Figure 37: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of parameter γ **averaged across all datasets**.

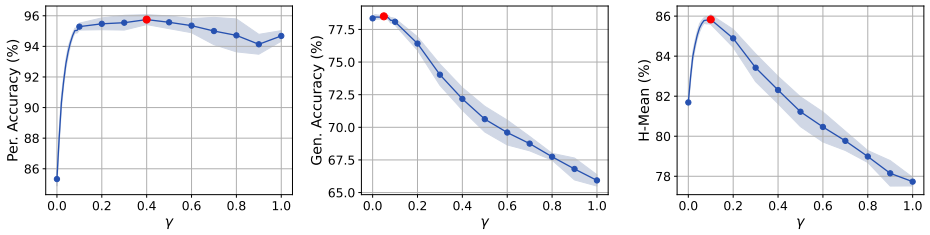


Figure 38: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of parameter γ for the ***UCF101*** dataset.

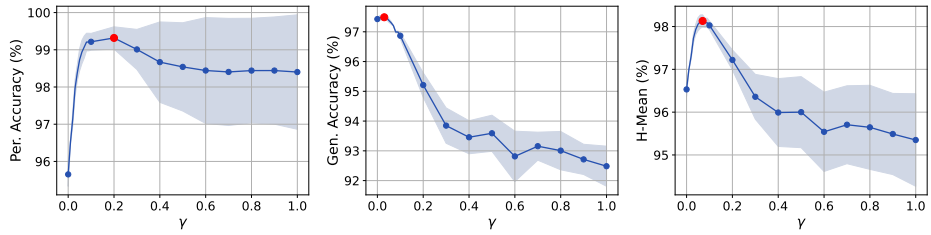


Figure 39: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of parameter γ for the ***Oxford Pets*** dataset.

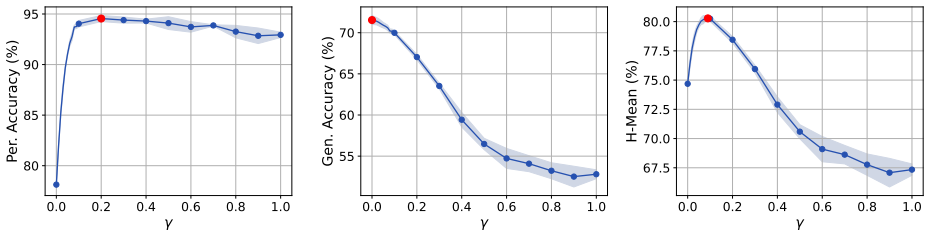


Figure 40: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of parameter γ for the ***Stanford Cars*** dataset.

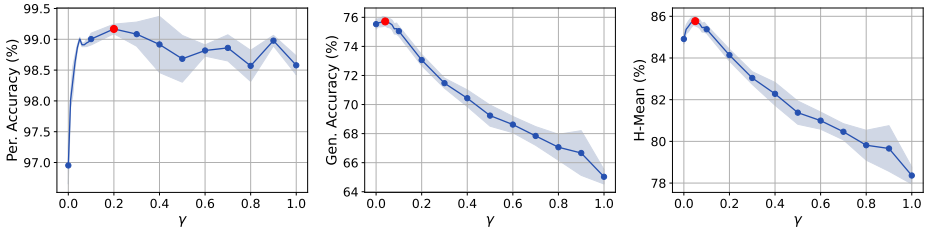


Figure 41: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of parameter γ for the ***Flowers*** dataset.

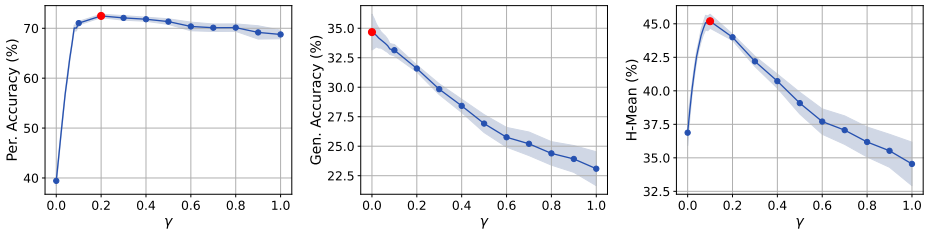


Figure 42: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of parameter γ for the ***FGVC Aircraft*** dataset.

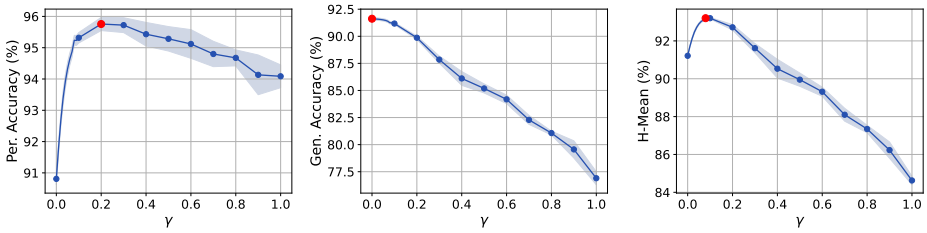


Figure 43: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of parameter γ for the ***Food101*** dataset.

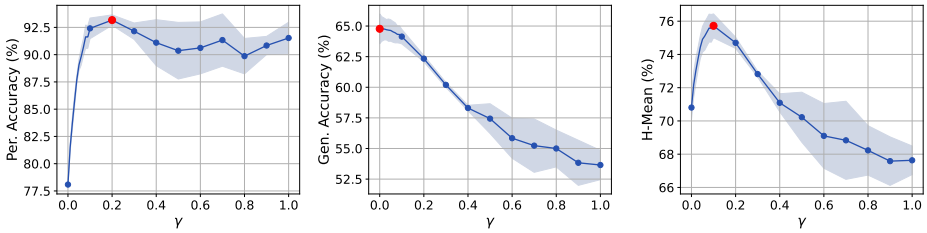


Figure 44: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of parameter γ for the ***DTD*** dataset.

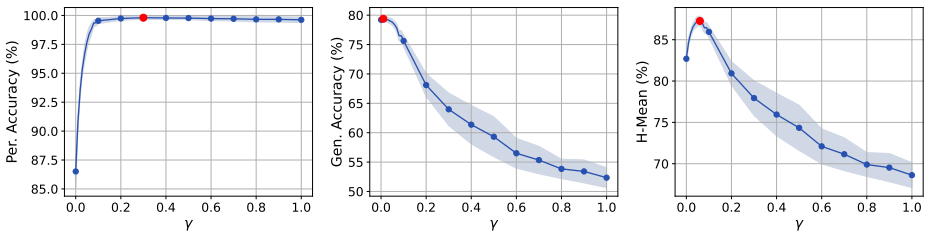


Figure 45: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of parameter γ for the ***EuroSAT*** dataset.

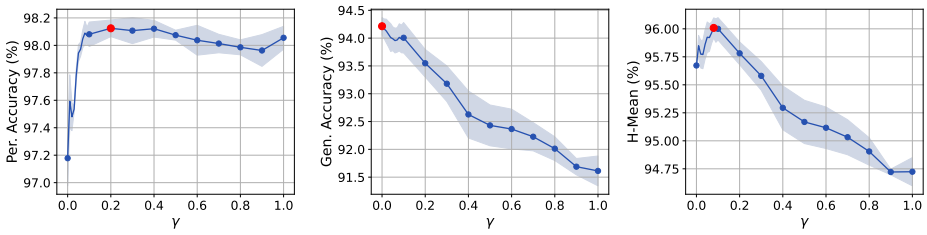


Figure 46: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of parameter γ for the ***Caltech*** dataset.

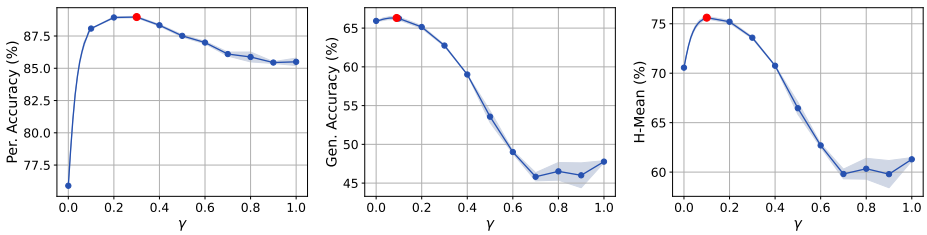


Figure 47: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of parameter γ for the ***ImageNet*** dataset.

J Effect of λ on Performance

We study the effect of the parameter λ which involves the tradeoff between the running anchor loss $\left(\mathcal{L}_n^{\mathsf{Anch}}\right)$ and the fusion loss $\left(\mathcal{L}_n^{\mathsf{Fus}}\right)$ in the loss function (refer to Eq. (22)). In Figs. 49-57, we have shown the personalization and generalization performances of the trained global models on various datasets alongside their H-Mean with the best performance for each one shown in a red marker/dot for each corresponding sub-plot.

In Eq. (22), we can see that increasing λ will result in more emphasis on the fusion term ($\mathcal{L}_n^{\mathsf{Fus}}$). Doing so, as pointed out in Sec. 3.2.4, will improve the personalization performance while it may sacrifice the generalization performance of the model which can be generally observed in Figs. 49- 57. In rare cases, such as the Flowers dataset (Fig. 52), we can observe that the generalization performance may improve as λ increases. Such behaviors can be attributed to dataset-specific characteristics. Nevertheless, in most scenarios (Fig. 49-51 and 53-57), a decline in generalization and improvement in personalization can be observed upon increasing the value of λ. Collectively, the optimal choice of λ, resulting in the best H-Mean of personalization and generalization, is obtained at $\lambda = 0.2$ (Fig. 48-right).

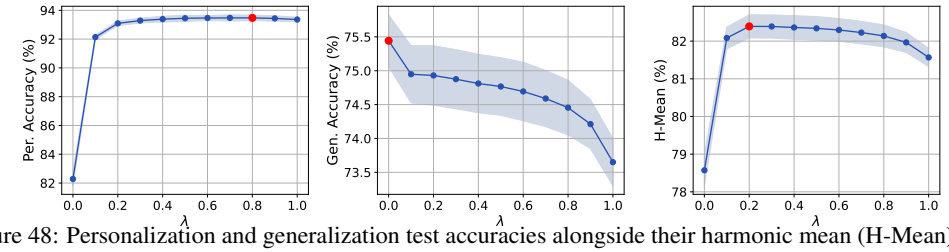


Figure 48: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of parameter λ **averaged across all datasets**.

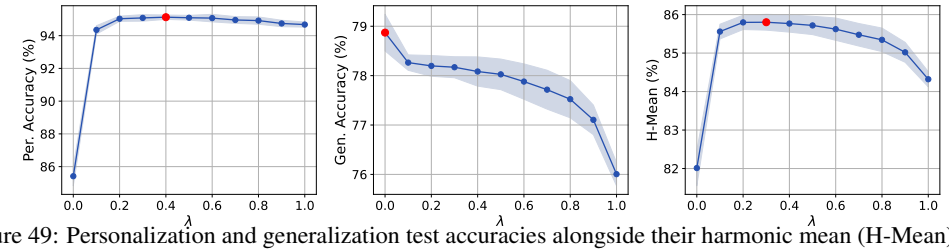


Figure 49: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of parameter λ for the ***UCF101*** dataset.

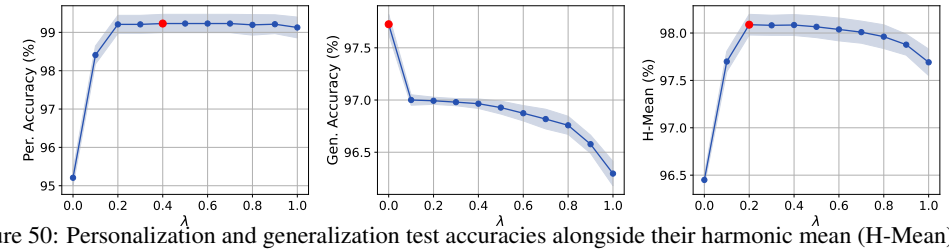


Figure 50: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of parameter λ for the ***OxfordPets*** dataset.

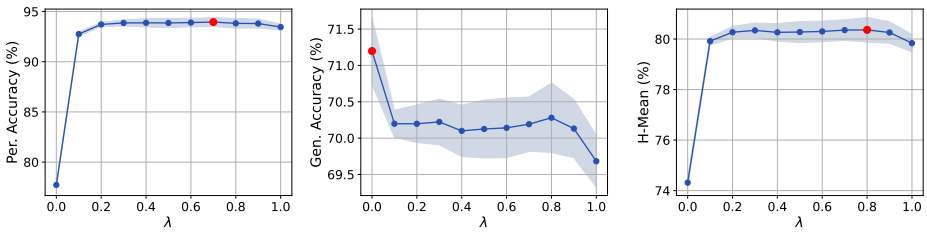


Figure 51: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of parameter λ for the ***Stanford-Cars*** dataset.

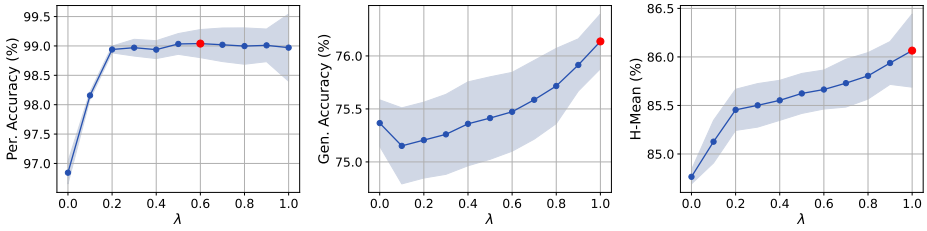


Figure 52: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of parameter λ for the ***Flowers*** dataset.

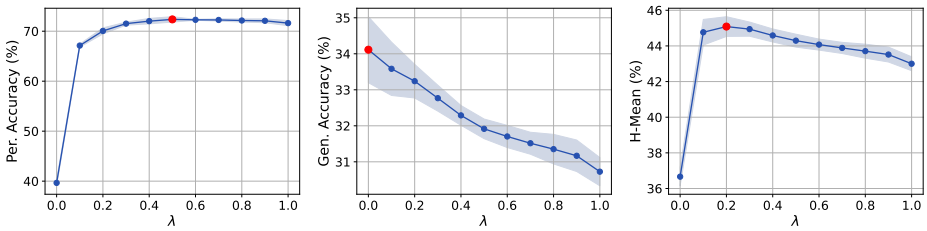


Figure 53: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of parameter λ for the ***FGVC Aircraft*** dataset.

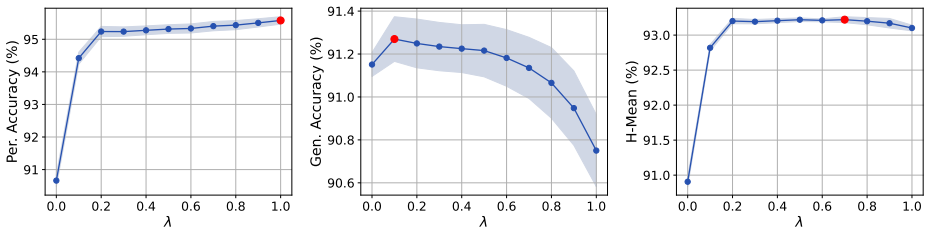


Figure 54: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of parameter λ for the ***Food101*** dataset.

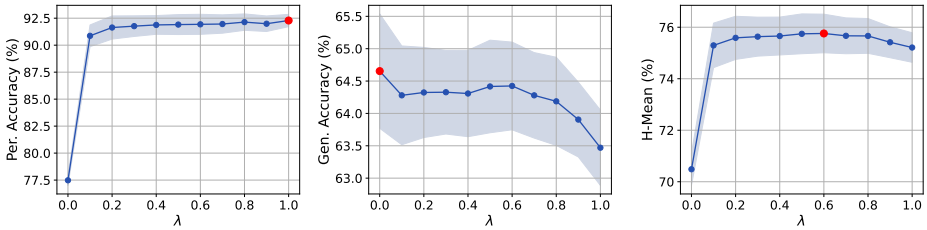


Figure 55: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of parameter λ for the ***DTD*** dataset.

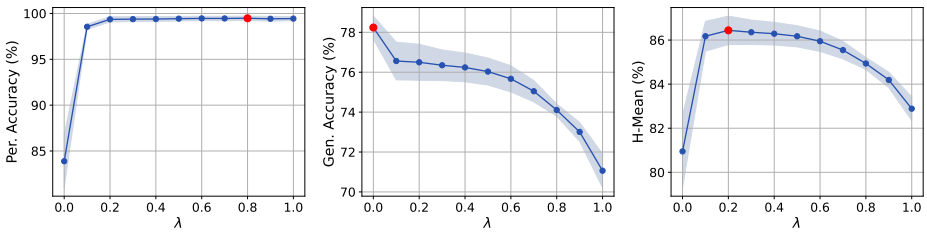


Figure 56: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of parameter λ for the *EuroSAT* dataset.

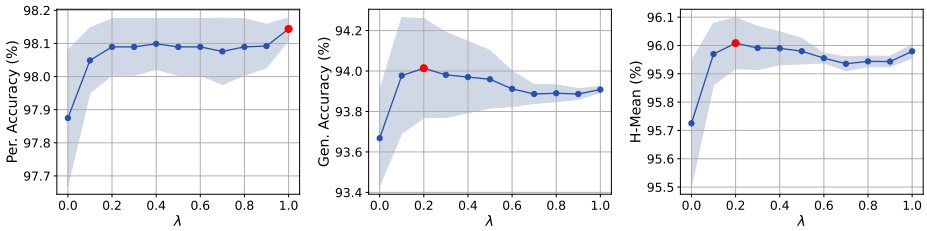


Figure 57: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of parameter λ for the ***Caltech*** dataset.

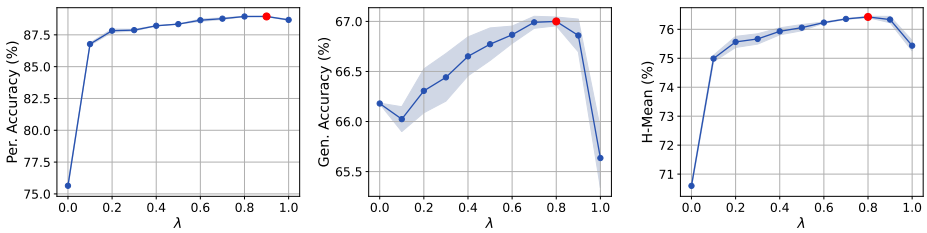


Figure 58: Personalization and generalization test accuracies alongside their harmonic mean (H-Mean) under changing values of parameter λ for the ***ImageNet*** dataset.

K Effect of Aggregation on Various Components

We devised a set of experiments to localize the effect of aggregating each trainable component in the model. We consider the following scenarios:

- Our proposed method (BLEND) where the vision and text adapters are aggregated while the personalized projection adapter is preserved locally.
- Aggregation of all the trainable components (corresponding to “Full" in the tables)
- No aggregation for any component where all the trained components are kept local (corresponding to “NoAgg" in the tables)
- Only vision adapters are aggregated (corresponding to “Only Vision" in the tables)
- Only text adapters are aggregated (corresponding to “Only Text" in the tables)

Our results indicate that for 8 out of 10 scenarios (Tables 37-39, 41, 43-46), aggregating only the text adapters results in the best personalization performance. This comes naturally, as the information about the classes is primarily encoded into the text modality. Hence, aggregating only the text encoders, given the use of pretrained CLIP models for the vision backbones, can enable the models to swiftly adapt to the local data distributions, making personalization less effortful. Intuitively, since the (Full) scenario aggregates all components, we expect it to result in better generalization performance. Our results verify this hypothesis as 9 out of 10 experiments (Tables 37-40, 42-46) have their best generalization test accuracy under the full aggregation scenario. However, in all datasets (Tables 37-46) along with the case that the average performance across the datasets is considered (Table 36), considering the balance between the generalization and personalization (i.e., the H-Mean performance), BLEND results in the best performance.

Table 36: **Avg across datasets.**

Agg. Strategy	Per.	Gen.	H-Mean
BLEND	92.91	74.62	**82.15**
Full	83.35	**75.41**	79.01
NoAgg	89.12	69.71	77.71
Only Vision	74.64	70.02	71.82
Only Text	**93.84**	67.31	77.04

Table 37: DTD

Agg. Strategy	Per.	Gen.	H-Mean
BLEND	91.64	64.32	**75.59**
Full	81.83	**65.62**	72.84
NoAgg	92.76	62.96	75.01
Only Vision	86.15	61.92	72.05
Only Text	**94.56**	62.77	75.46

Table 38: EuroSAT

Agg. Strategy	Per.	Gen.	H-Mean
BLEND	99.36	76.50	**86.44**
Full	89.50	**78.88**	83.85
NoAgg	99.56	72.15	83.66
Only Vision	99.43	72.85	84.09
Only Text	**99.71**	73.89	84.88

Table 39: FGVC Aircraft

Agg. Strategy	Per.	Gen.	H-Mean
BLEND	70.05	33.24	**45.09**
Full	40.22	**35.17**	37.53
NoAgg	52.46	26.49	35.20
Only Vision	25.28	30.35	27.58
Only Text	**72.04**	23.09	34.97

Table 40: Food101

Agg. Strategy	Per.	Gen.	H-Mean
BLEND	95.24	91.25	**93.20**
Full	90.61	**91.27**	90.94
NoAgg	**96.00**	87.10	91.33
Only Vision	86.58	88.00	87.28
Only Text	95.59	84.33	89.61

Table 41: Flowers

Agg. Strategy	Per.	Gen.	H-Mean
BLEND	98.94	**75.21**	**85.46**
Full	97.84	74.11	84.34
NoAgg	96.64	69.52	80.87
Only Vision	80.51	74.09	77.17
Only Text	**99.07**	65.18	78.63

Table 42: OxfordPets

Agg. Strategy	Per.	Gen.	H-Mean
BLEND	99.21	96.99	**98.09**
Full	94.87	**97.45**	96.14
NoAgg	**99.27**	96.28	97.75
Only Vision	99.02	96.62	97.81
Only Text	99.23	96.06	97.62

Table 43: Stanford Cars

Agg. Strategy	Per.	Gen.	H-Mean
BLEND	93.72	70.20	**80.27**
Full	78.95	**71.11**	74.82
NoAgg	81.11	56.14	66.35
Only Vision	47.83	57.44	52.20
Only Text	**94.47**	48.10	63.75

Table 44: UCF101

Agg. Strategy	Per.	Gen.	H-Mean
BLEND	95.03	78.20	**85.80**
Full	86.51	**79.65**	82.94
NoAgg	94.04	71.37	81.15
Only Vision	73.16	68.97	71.00
Only Text	**95.32**	71.45	81.67

Table 45: ImageNet

Agg. Strategy	Per.	Gen.	H-Mean
BLEND	87.83	66.31	**75.56**
Full	76.08	**67.15**	71.34
NoAgg	81.57	61.38	70.05
Only Vision	51.66	56.73	54.08
Only Text	**90.30**	54.57	68.03

Table 46: Caltech

Agg. Strategy	Per.	Gen.	H-Mean
BLEND	98.09	**94.01**	**96.01**
Full	97.11	93.67	95.36
NoAgg	97.80	93.68	95.69
Only Vision	96.75	93.20	94.94
Only Text	**98.12**	93.65	95.83

L Effect of Trainable Text Projection Head

To examine the effect of frozen vs. trainable text projection head, we experiment with different text projection head adapter ranks for the values $\{\emptyset, 4, 8, 16, 32, 64, 128, 256, \text{Full-rank}\}$, where $\emptyset$ corresponds to the case where no adapter is used (i.e., the case where the text projection head is frozen). We include the results for all datasets in Tables 48-57 with the average performance across the datasets included in Table 47. The best personalization result in 6 out of 10 scenarios (Tables 49, 50, 53, 55-57) is achieved without adapting the text projection head to the new data (i.e., by keeping it frozen). The same situation holds for the generalization performance in 5 out of 10 scenarios (Tables 51-55). Given that the best average H-Mean across all datasets is also observed in the absence of the text projection head training (Table 47, right column), we conclude that fine-tuning the text projection head over new sets of data can be redundant on average as it introduces training overhead with no performance gains in terms of H-Mean.

Table 47: **Avg across datasets.**

Text Proj. Rank	Per.	Gen.	H-Mean
No Adapter	92.91	**74.62**	**82.15**
4	92.92	73.96	81.74
8	**92.98**	73.88	81.72
16	92.56	73.76	81.48
32	92.60	73.76	81.47
64	92.49	73.72	81.40
128	92.11	73.56	81.19
256	91.70	73.18	80.77
Full-rank	91.40	72.84	80.46

Table 48: DTD

Text Proj. Rank	Per.	Gen.	H-Mean
No Adapter	91.64	64.32	75.59
4	92.93	**64.84**	**76.38**
8	**92.99**	64.76	76.35
16	92.50	64.46	75.98
32	92.89	63.80	75.65
64	92.20	62.92	74.80
128	92.25	62.25	74.33
256	91.39	61.17	73.28
Full-rank	91.12	60.83	72.95

Table 49: EuroSAT

Text Proj. Rank	Per.	Gen.	H-Mean
No Adapter	**99.36**	76.50	**86.44**
4	99.01	73.43	84.33
8	99.07	73.35	84.29
16	98.78	75.01	85.27
32	98.62	75.44	85.49
64	98.60	**76.53**	86.17
128	98.27	75.95	85.68
256	98.58	75.13	85.27
Full-rank	98.07	71.33	82.59

Table 50: FGVC Aircraft

Text Proj. Rank	Per.	Gen.	H-Mean
No Adapter	**70.05**	33.24	45.09
4	69.65	**33.37**	**45.12**
8	69.71	33.29	45.06
16	68.48	32.92	44.46
32	69.25	32.48	44.22
64	68.97	32.04	43.75
128	68.05	32.62	44.10
256	66.91	31.86	43.16
Full-rank	67.02	33.31	44.50

Table 51: Food101

Text Proj. Rank	Per.	Gen.	H-Mean
No Adapter	95.24	**91.25**	**93.20**
4	95.36	91.13	**93.20**
8	**95.42**	91.05	93.18
16	94.93	90.72	92.78
32	94.98	90.37	92.62
64	94.86	90.10	92.42
128	94.66	90.18	92.36
256	93.91	90.06	91.95
Full-rank	93.13	89.88	91.48

Table 52: Flowers

Text Proj. Rank	Per.	Gen.	H-Mean
No Adapter	98.94	**75.21**	**85.46**
4	99.35	74.35	85.05
8	**99.41**	74.27	85.02
16	99.37	73.39	84.43
32	99.32	73.47	84.46
64	99.30	73.34	84.37
128	99.30	73.17	84.25
256	99.33	72.76	84.00
Full-rank	99.27	72.48	83.79

Table 53: OxfordPets

Text Proj. Rank	Per.	Gen.	H-Mean
No Adapter	**99.21**	**96.99**	**98.09**
4	99.11	96.05	97.56
8	99.17	95.97	97.55
16	99.13	96.29	97.69
32	98.24	96.07	97.14
64	98.41	95.76	97.07
128	98.38	95.20	96.76
256	98.08	95.77	96.91
Full-rank	98.49	96.16	97.31

Table 54: Stanford Cars

Text Proj. Rank	Per.	Gen.	H-Mean
No Adapter	93.72	**70.20**	**80.27**
4	93.75	68.51	79.17
8	**93.81**	68.43	79.14
16	92.90	66.61	77.59
32	92.89	67.92	78.47
64	93.76	68.79	79.35
128	92.74	69.49	79.45
256	92.73	68.75	78.96
Full-rank	92.72	69.30	79.32

Table 55: UCF101

Text Proj. Rank	Per.	Gen.	H-Mean
No Adapter	**95.03**	**78.20**	**85.80**
4	94.76	76.67	84.76
8	94.82	76.59	84.73
16	94.37	76.24	84.35
32	94.85	76.58	84.74
64	94.60	76.28	84.45
128	93.99	76.41	84.29
256	94.02	76.44	84.33
Full-rank	92.69	75.13	82.99

Table 56: ImageNet

Text Proj. Rank	Per.	Gen.	H-Mean
No Adapter	**87.83**	66.31	75.56
4	87.25	66.93	75.75
8	87.31	66.85	75.72
16	87.14	**67.49**	**76.07**
32	86.86	67.46	75.94
64	86.15	67.36	75.61
128	85.46	66.15	74.58
256	84.07	65.70	73.76
Full-rank	83.67	65.84	73.69

Table 57: Caltech

Text Proj. Rank	Per.	Gen.	H-Mean
No Adapter	**98.09**	94.01	96.01
4	97.99	94.32	96.12
8	98.05	94.24	96.11
16	98.01	**94.47**	**96.21**
32	98.05	94.02	95.99
64	98.02	94.09	96.02
128	97.99	94.17	96.04
256	97.96	94.18	96.03
Full-rank	97.82	94.14	95.95

M Performance Variation Across Different Settings

Table 58 summarizes BLEND's performance under varying levels of data heterogeneity, number of clients, and local data scarcity. Across all settings, BLEND achieves larger H-mean gains over the best baseline as task difficulty increases. Specifically, the gains increase with stronger data heterogeneity, more clients, and fewer local training samples. Similar improvements are observed on the real-world iNaturalist-User-120k dataset as the number of clients increases. These results highlight BLEND's effectiveness in challenging federated/distributed fine-tuning scenarios.

Table 58: Performance variation of BLEND across different settings. H-mean gains are reported relative to the best baseline.

Setting	**Increasing Difficulty**	**H-Mean Gain vs. Best Baseline**
Data heterogeneity (across 10 datasets in Sec. 4)	$\alpha^{\text{Dir}}: 10 \rightarrow 0.1 \rightarrow 0.01$	$+0.9 \rightarrow +2.9 \rightarrow +4.8$
Number of clients (across 10 datasets in Sec. 4)	$N: 10 \rightarrow 20 \rightarrow 30$	$+4.8 \rightarrow +5.6 \rightarrow +6.1$
Real-World iNaturalist-User-120k Dataset (Appendix D)	$N: 100 \rightarrow 200$	$+4.9 \rightarrow +5.6$
Local data scarcity (across 10 datasets in Sec. 4)	#Shots: $32 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2$	$+3.0 \rightarrow +4.8 \rightarrow +5.2 \rightarrow +5.3 \rightarrow +5.7$

N Effect of the Running Anchor Loss

To further investigate the effectiveness of BLEND's running anchor loss, we compare it against two alternative zero-shot CLIP regularization strategies: (i) **Zero-Shot G-Head**, which uses the frozen zero-shot CLIP vision representation for the G-head, and (ii) **Zero-Shot Distilled G-Head**, which employs KL divergence to distill zero-shot knowledge from frozen CLIP's vision projection into the G-head.

All experiments follow the experimental setting of Table I in the main paper. As shown in Table 59, our proposed running anchor loss achieves the highest H-mean of 82.15%, outperforming the zero-shot G-head and zero-shot distilled G-head by 2.20 and 1.61 percentage points, respectively. These results demonstrate the effectiveness of our running anchor loss in balancing personalization and generalization.

Table 59: Comparison of BLEND's running anchor with alternative zero-shot CLIP regularization strategies.

Method	Per.	Gen.	H-Mean
BLEND with Zero-Shot G-Head	89.50	72.31	79.95
BLEND with Zero-Shot Distilled G-Head	92.83	72.35	80.54
BLEND with Running Anchor Loss	**92.91**	**74.62**	**82.15**

O Effect of the Adaptive Fusion Weight

To further investigate the sensitivity of BLEND to fusion weight γ, we conduct additional ablation studies over different values of γ while varying the degree of data heterogeneity and the number of clients. Table 60 summarizes the optimal fusion weights and their corresponding H-mean performance. As the Dirichlet concentration parameter decreases from 10 to 0.01, indicating increasingly heterogeneous client data, the optimal γ increases from 0.03 to 0.09 (this is because under more heterogeneous data, the need for model personalization increases, which is achieved under a larger value of γ). Similarly, increasing the number of clients from 10 to 30 results in a small increase in the optimal γ from 0.09 to 0.10. These observations suggest that setting-dependent fusion weights can further improve BLEND's performance.

Table 60: Effect of fusion weight γ under varying levels of data heterogeneity and different numbers of clients.

α^{Dir}	Optimal γ	H-Mean
10	0.03	79.66
0.1	0.06	80.86
0.01	0.09	82.15

N	Optimal γ	H-Mean
10	0.09	82.15
20	0.10	82.83
30	0.10	83.10

P Effect of Explicit Personalization

To investigate the effect of explicitly supervising the personalized branch (i.e., P-head), we introduce an additional cross-entropy (CE) loss that directly optimizes the personalized representation, rather than relying solely on its contribution through the fusion CE loss. We refer to this variant as BLEND (Sep-Per). As shown in Table 61, introducing explicit personalization increases personalization accuracy from 92.91% to 93.56%, but decreases generalization accuracy from 74.62% to 70.96%. Consequently, the H-mean decreases from 82.15% to 79.78%. These results suggest that directly optimizing the personalized branch can improve client-specific performance at the expense of generalization, supporting BLEND's original formulation, which optimizes the personalized representation through the fusion objective.

Table 61: Effect of explicitly supervising the personalized branch with a separate cross-entropy loss.

Method	Per.	Gen.	H-Mean
BLEND (Sep-Per)	**93.56**	70.96	79.78
BLEND (Original)	92.91	**74.62**	**82.15**

Q Hyperparameter Selection and Held-Out Validation

In this ablation, we evaluate the robustness of BLEND's hyperparameter selection using a held-out validation split that is independent of the training and test sets. As shown in Tables 62 and 63, the highest validation H-mean is obtained with $\gamma = 0.09$, $\lambda = 0.2$, a global adapter dimension of 16, and a projection adapter dimension of 256. These values coincide with the configuration identified in our original test-set sensitivity analyses. This agreement provides independent validation of the selected configuration and indicates that the reported hyperparameter choices are not specific to the original test split.

Table 62: Hyperparameter sensitivity of the fusion weight γ and regularization coefficient λ on the held-out validation split. The highest H-mean for each hyperparameter is highlighted in bold.

γ	0.0	0.06	0.08	0.09	0.1	0.3	1
H-Mean	78.86	82.25	82.38	**82.39**	82.36	79.47	73.47
λ	0.0	0.1	0.19	0.2	0.21	0.5	1
H-Mean	77.32	82.08	82.24	**82.39**	82.36	82.21	81.16

Table 63: Sensitivity to the global and projection adapter dimensions on the held-out validation split. The highest H-mean for each adapter is highlighted in bold.

Global adapter dim.	0	4	8	16	32	64	128	256	512
H-Mean	80.84	82.11	82.18	**82.39**	82.06	81.97	81.75	81.31	81.32

Projection adapter dim.	4	8	16	32	64	128	256	768
H-Mean	81.78	81.92	82.08	82.14	82.20	82.23	**82.39**	82.18

R Limitations and Generalization Beyond CLIP

Extension to Other Multimodal Architectures. BLEND currently focuses on CLIP-based image–text models, consistent with prior work on personalized federated learning with vision-language models. Extending BLEND to other multimodal architectures primarily requires reconsidering where global and personalized modules are inserted into the model pipeline. For instance, in Qwen2.5-VL, one possible extension is to maintain a global vision merger while introducing a separate personalized vision merger. Their resulting visual representations could then be fused and passed to the language model as visual tokens for generation.

Extension to Other Downstream Tasks. While BLEND currently focuses on classification, extending it to image–text retrieval would require adapting its fusion mechanism and training objectives. Unlike classification, retrieval involves ranking image–text pairs rather than selecting from a fixed set of classes. Therefore, the fusion of global and personalized visual representations must preserve cross-modal alignment while balancing generalization and client-specific personalization. For generative tasks such as visual question answering (VQA), BLEND's global–local decomposition could be retained, while its fusion mechanism and anchor objectives would need to be adapted to autoregressive generation.

Extension to Multi-Task and Multi-Modal Settings. Extending BLEND to more complex multi-task and multi-modal settings introduces two important challenges. First, different tasks may rely on different modalities to varying degrees, making the information imbalance across modalities task-dependent. This introduces an additional form of heterogeneity, which we refer to as *heterogeneity in information imbalance across tasks*. Addressing this challenge may require task-dependent placement of personalized modules and adaptive fusion weights. Second, globally shared and aggregatable components must account for task similarity, modality relevance, and potential negative cross-task transfer under heterogeneous client data distributions. Investigating these challenges represents an important direction for extending BLEND to broader multimodal federated learning scenarios.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims presented in the abstract and introduction accurately reflect the contributions and scope of the paper.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

Justification: The paper explicitly discusses the limitations of the proposed approach.

Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: This work does not present formal theoretical results or proofs. Instead, we introduce mathematical formulations to define the model components and the proposed loss functions. These formulations are fully specified in the paper, but they do not constitute theoretical claims requiring formal assumptions or proofs.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

Justification: The paper provides all necessary details to reproduce the main experimental results. This includes a complete description of the model architecture, training procedure, loss functions, datasets, evaluation metrics, and hyperparameters.

Guidelines:

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

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

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

   Answer: [Yes]

   Justification: We provide open access to the implementation of our method along with detailed instructions to reproduce the main experimental results. The codebase includes scripts for data preprocessing, training, and evaluation, as well as configuration files specifying all relevant hyperparameters and settings. All datasets used in this work are publicly available, and links or references are provided to ensure reproducibility.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The paper specifies all necessary training and evaluation details required to understand and reproduce the results. This includes descriptions of datasets and data splits, model architecture, loss functions, hyperparameters and their selection, optimization methods, and evaluation protocols.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The paper reports variability in the experimental results using appropriate statistical measures, such as mean and standard deviation across multiple runs or seeds. Error bars or ± values are included to reflect performance variability and ensure that the reported improvements are statistically meaningful.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The paper reports the hardware setup used for experiments (e.g., GPU, CPU, and memory specifications). Instead of relying on execution time, we characterize computation, communication, and memory requirements using the number of trainable parameters and communication overhead, providing a consistent and hardware-independent measure of resource usage.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research presented in this paper complies with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This work focuses on methodological improvements for federated vision–language models and does not introduce new applications, datasets, or systems with direct societal impact. As such, we do not identify significant positive or negative societal impacts specific to this work beyond those already associated with existing VLMs and federated learning frameworks.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate

to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This work does not introduce or release new datasets or pre-trained models that pose a high risk of misuse. Our method focuses on improving training strategies for federated vision–language models. All experiments are conducted on publicly available datasets and standard model architectures (e.g., CLIP). Therefore, we do not identify significant risks that require additional safeguards beyond standard research practices.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All assets used in this work, including datasets, pre-trained models (e.g., CLIP), and any external codebases, are publicly available and properly credited. We explicitly cite the original sources in the paper and adhere to their respective licenses and terms of use.

    Guidelines:

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

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: This work does not introduce new datasets, models, or code as standalone assets. We build upon existing publicly available resources and focus on methodological contributions. Therefore, no additional asset documentation is required.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing, human subjects, or user studies. All experiments are conducted on publicly available datasets and do not require human participation.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This work does not involve human subjects, user studies, or any form of data collection from participants. All experiments are conducted using publicly available datasets. Therefore, IRB approval is not required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs are not used as part of the methodology or experimental pipeline in this work. Any use of LLMs, if present, was limited to writing or editing assistance and does not affect the scientific content, methodology, or results. Therefore, no declaration is required.

    Guidelines:

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