FEDAURA: FEDERATED VISUAL PROMPTING FOR CROSS-MODEL ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning (FL) faces the issue of model heterogeneity in the era of large models. Clients with diverse computational capacities often run models of different sizes. This renders traditional model aggregation methods ineffective. Existing approaches try to address this issue through knowledge distillation or partial training. The former relies on public data and often yields suboptimal performance, while the latter limits model architecture and often leads to uneven optimization. This paper presents FedAura, a federated learning scheme that accommodates model heterogeneity from the input space based on visual prompting. It enables cross-model adaptation by learning and aggregating lightweight visual prompts, which significantly reduces computation and communication overhead. FedAura introduces a model-contrastive loss to alleviate feature corruption in visual prompting caused by model heterogeneity. It aligns cross-model representation based on local consistency between pre-trained and prompted representations. In addition, it leverages a prompt-contrastive loss to alleviate data heterogeneity by driving semantic alignment and improving generalization across diverse data distributions. Extensive experiments across eight heterogeneous architectures (ResNet and ViT variants) on four image recognition datasets show the effectiveness of FedAura in accommodating model heterogeneity. Compared to state-of-the-art knowledge-distillation and partial-training baselines, FedAura improves model accuracy by up to 18.40%, reduces communication cost by up to $912\times$, and reduces training FLOPs by up to $7.8\times$. These results demonstrate that input-level prompt collaboration is a promising and resource-efficient direction for model-heterogeneous federated learning.

1 INTRODUCTION

Federated learning (FL) McMahan et al. (2017); Shi et al. (2023); Wang et al. (2024b); Li et al. (2021b) has emerged as a powerful paradigm for enabling collaborative model training across distributed clients without sharing their raw data. It preserves privacy and adheres to data protection regulations like the General Data Protection Regulation (GDPR) Wikipedia (2018b) and the California Consumer Privacy Act (CCPA) Wikipedia (2018a). In a typical FL process, each client trains a local model on its private dataset and periodically uploads its local model to a central server. The server then aggregates these local models to produce a global model, which is subsequently redistributed to all clients for training in the next training round. Through this iterative process, FL enables global knowledge sharing across clients while maintaining local data privacy.

However, the rapid proliferation of large-scale visual models (e.g., ViT Dosovitskiy et al. (2021); Touvron et al. (2021a); Liu et al. (2021) and CLIP Radford et al. (2021); Li et al. (2022)) brings new challenges to FL. In practice, clients differ widely in computational and memory capacities, ranging from powerful edge servers to resource-constrained edge devices like mobile phones and smart vehicles. Their performance heterogeneity makes it impractical to deploy a uniform vision model across all clients Diao et al. (2020); Wang et al. (2023b); Dai et al. (2022). As model architectures continue to grow in depth and parameter size, the disparity among clients in training models that match their resources becomes increasingly pronounced Diao et al. (2020); Alam et al. (2022). Consequently, clients often run models of varied sizes based on their computing capacities and application needs. This raises the issue of *model heterogeneity*.

Model heterogeneity breaks the core assumption of traditional FL algorithms, which rely on parameter alignment across identical models to perform global aggregation McMahan et al. (2017); Li et al. (2020b). When clients adopt models with different depths, layer types, or parameter sizes, direct aggregation of their weights becomes infeasible, as illustrated in Fig. 1. Knowledge distillation (KD) Li & Wang (2019); Lin et al. (2020) has been widely explored as a solution to model heterogeneity in FL. In KD-based FL, clients transmit the soft logits produced by their models to the server, which aggregates these logits to produce global logits for redistribution to the clients. In this way, knowledge is shared across clients through the exchange of soft logits rather than model parameters. However, KD-based methods

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

typically depend on a public or auxiliary dataset at clients and the server to secure competitive performance. This is often impractical in privacy-sensitive settings. Moreover, since these methods rely solely on shared logits rather than full parameter updates, their performance is generally inferior to parameter-based aggregation approaches Zhang et al. (2021); Liang et al. (2025).

Another solution to model heterogeneity is partial training (PT) Diao et al. (2020); Alam et al. (2022); Li et al. (2021a), where each client trains a sub-model extracted from the global model. Representative methods include random sub-model extraction Liang et al. (2025); Alam et al. (2022) and static sub-model extraction Diao et al. (2020). In these approaches, the server aggregates sub-models trained by individual clients to update the global model. However, PT-based methods do not allow clients to obtain a full model adapted to their local computational capacity. This limits model expressiveness on capable devices. Moreover, since clients train different parts of the global model, the global model often experiences uneven optimization Wang et al. (2024a); Alam et al. (2022); Zhou et al. (2023), where some parameters receive substantially more updates than others. This undermines the overall training performance.

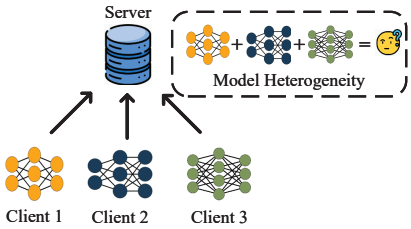


Figure 1: Illustration of model heterogeneity in federated learning. Clients possess models with different architectures and capacities, making direct parameter aggregation infeasible for the server.

As the first attempt to accommodate visual model heterogeneity through input-level adaptation in vision, this paper presents FedAura, a new FL scheme for vision models that learns and aggregates visual prompts to enable cross-model collaboration with minimal computation and communication cost. In FedAura, each client runs a pre-trained model (e.g., ViT) that matches its computational and storage capacity. These pre-trained models remain frozen while clients only train the visual prompts (VP) Huang et al. (2023); Bahng et al. (2022) on their local data. Since prompt training involves significantly fewer parameters and all clients share the same input dimensionality, the server can efficiently aggregate the prompts with the same architecture using standard methods such as FedAvg McMahan et al. (2017); Lim et al. (2020); Collins et al. (2021); Karimireddy et al. (2020). However, we observe that although visual prompting improves the performance of the model used during prompt training, the learned prompts often fail to generalize across different model architectures. Even prompts trained on the same backbone model diverge under non-Independent and Identically Distributed (non-IID) data. These observations highlight that prompt aggregation is challenged by both model heterogeneity and data heterogeneity. This motivates the need for explicit representation alignment.

To tackle the above challenges, FedAura introduces two contrastive losses during local prompt training to preserve generalizable knowledge representations. First, it employs a model-contrastive loss to mitigate feature corruption caused by model heterogeneity. It aligns prompted representations with those of the original backbone to preserve essential general knowledge. Second, FedAura introduces a prompt-contrastive loss to mitigate prompt drifts caused by data heterogeneity. It encourages each client's prompt to remain aligned with the global prompt representation. This alignment reduces the divergence among client prompts and improves the generalization of the aggregated prompt. FedAura's key contributions include:

- To our best knowledge, FedAura is the first FL scheme that addresses model heterogeneity from the input space. Compared with existing FL schemes that aggregate model parameters or logits, FedAura enables collaboration across heterogeneous models through learning and sharing their visual prompts in a federated manner.
- FedAura introduces a model-contrastive loss to maintain consistent feature modulation across heterogeneous backbones. This loss aligns the representations of prompted and unprompted models on each client, preserving shared semantic structure and mitigating feature corruption caused by model heterogeneity.
- FedAura introduces a prompt-contrastive loss to mitigate prompt drift under non-IID data settings. By encouraging local prompts to remain semantically aligned with the global prompt representation, prompt-contrastive loss stabilizes cross-client prompt semantics and improves the generalization of the global prompt.
- Extensive experiments on CIFAR-10 Krizhevsky et al. (2009), CIFAR-100 Krizhevsky et al. (2009), Tiny-ImageNet Le & Yang (2015), Flower-102 Nilsback & Zisserman (2008) with heterogeneous-model configurations validate the effectiveness of FedAura. The results show that FedAura consistently improves cross-model collaboration and global accuracy compared to knowledge-distillation-based and partial-training-based baselines.

2 Related Work

Federated Learning. Federated learning (FL) McMahan et al. (2017); Mohri et al. (2019); Panchal et al. (2023) enables multiple clients to collaboratively train a model under data locality constraints, without sharing their raw data. A central

server coordinates the training process by aggregating client models, typically using parameter averaging strategies such as FedAvg McMahan et al. (2017) or its variants Li et al. (2020b). Extensive research in recent years has focused on improving stability and convergence in real-world conditions, including communication-efficient optimization Konečnỳ et al. (2016); Wang et al. (2019), robustness to non-IID data distributions Li et al. (2020a); Karimireddy et al. (2020), and personalization to accommodate heterogeneous client objectives Fallah et al. (2020); Collins et al. (2021). These advances have significantly broadened the applicability of FL across domains such as mobile intelligence Feng et al. (2020), healthcare Dayan et al. (2021), and finance Zhang et al. (2024a). However, most classical FL schemes assume that all clients share the same model architecture, which allows direct parameter aggregation. This assumption becomes restrictive in modern deployment environments where client devices differ widely in computational capacity Diao et al. (2020); Wang et al. (2023b); Li & Wang (2019), leading to model heterogeneity.

Model Heterogeneity in Federated Learning. Knowledge Distillation (KD)-based FL methods Li & Wang (2019); Lin et al. (2020); Wang et al. (2023a); Wu et al. (2022) address model heterogeneity by exchanging soft predictions instead of model parameters. FedMD Li & Wang (2019) aligns heterogeneous clients through distillation on a shared public dataset, while FedDF Lin et al. (2020) aggregates ensemble predictions on the server as global targets. However, these approaches rely on public or auxiliary data, which is often infeasible in privacy-sensitive scenarios, and their knowledge transfer via logits is less expressive than parameter-based aggregation, leading to suboptimal performance.

Partial training methods Diao et al. (2020); Alam et al. (2022); Li et al. (2021a) mitigate heterogeneity by allowing each client to train only a subset of the global model. HeteroFL Diao et al. (2020) assigns model widths based on client capacity, while FedRolex Alam et al. (2022) rotates and redistributes sub-models across rounds to ensure all parameters are updated. Although these approaches allow clients to use models that match local resource constraints, they do not provide each client with a complete model suited to its capacity. The reliance on sub-model extraction leads to uneven parameter optimization, which can degrade global performance.

Visual Prompting in Vision Models. Visual prompting Huang et al. (2023); Bahng et al. (2022) extends the idea of prompting from natural language models to vision models, particularly vision transformers (ViTs). Instead of updating backbone parameters, visual prompting learns a set of input-level embeddings that are prepended or fused with image patches to guide the pretrained model toward the downstream task. This approach enables parameter-efficient adaptation while preserving the generalizable representations learned during large-scale pre-training. Recent studies have attempted to improve the transferability Zhang et al. (2024b) and robustness Li et al. (2024b) of visual prompts across diverse tasks and domains.

Federated Prompt Learning. Recent studies propose to adapt pretrained models in federated learning by tuning prompts rather than full model parameters Guo et al. (2023b;a); Deng et al. (2024); Li et al. (2024a). PromptFL Guo et al. (2023b) shows that clients can collaboratively learn a shared prompt with the backbone frozen. This reduces the communication overhead incurred and preserves model consistency. PfedPrompt Guo et al. (2023a) and PLAN Gong et al. (2026) introduces personalized prompts to strike a balance between global knowledge sharing and client-specific specialization. Li et al. Li et al. (2024a) formulate global-local prompt cooperation based on the optimal transport theory to improve generalization on heterogeneous client data.

However, *all* these methods assume that every client uses the *same backbone architecture*, and inserts prompts into the same layers of the model. This architectural homogeneity enables direct prompt aggregation but is unrealistic in practical federated environments where clients may hold models with different architectures. Furthermore, existing federated prompt learning methods are inherently restricted to *Transformer-based architectures*, since their prompts are implemented as learnable token embeddings whose dimension is tightly coupled to the hidden dimension of the specific backbone (e.g., d=768 for ViT-B/16), making them structurally incompatible with CNN-based models such as ResNet that lack a token-based input interface. In contrast, FedAura adopts pixel-level visual prompts applied in the input image space, prior to and independently of the backbone, making it compatible with any vision architecture, whether CNN-based (e.g., ResNet) or Transformer-based (e.g., ViT, DeiT, CaiT).

3 METHOD

Visual prompting Jia et al. (2022) offers a parameter-efficient way to adapt pre-trained vision models by learning lightweight input-level embeddings without updating backbone parameters. In federated settings, however, clients train prompts locally on private data and with heterogeneous model architectures. This causes inconsistent feature modulation and prompt drift. As a result, directly aggregating local prompts often leads to performance degradation. To address this, this paper presents FedAura, a federated visual prompting scheme that enables effective prompt sharing across heterogeneous client models and non-IID data. It incorporates a model-contrastive loss to maintain feature consistency across different backbones and a prompt-contrastive loss to improve prompt generalization across clients. This section

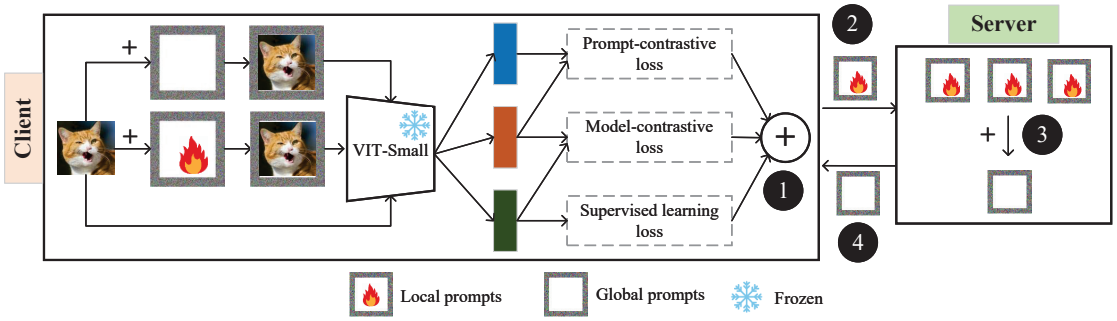


Figure 2: Overview of FedAura. Each training round consists of: ① Local prompt training using supervised loss, model-contrastive loss, and prompt-contrastive loss; ② Uploading local prompts to the server; ③ Server-side aggregation to obtain the global prompt; ④ Broadcasting the updated global prompt back to clients. For clarity, only one representative client is shown, while in practice, clients often run different backbone models according to their computational capacity.

first presents preliminaries on federated visual prompting, then formulates the problem of prompt aggregation across heterogeneous models, and finally describes FedAura in detail. An overview of the scheme is illustrated in Fig. 2.

3.1 Preliminaries

We consider a federated learning setting consisting of a central server and K clients. Each client $k \in \{1, \dots, K\}$ holds a private dataset $\mathcal{D}_k$ drawn from a local distribution $p_k(x, y)$, where the distributions across clients may be non-IID. Due to differences in computational and storage capacities, clients may deploy *heterogeneous pre-trained models*, denoted as $f_k(\cdot)$. These models share similar representational objectives but vary in architecture.

Visual prompting introduces a set of learnable input-level parameters, termed *visual prompts*, denoted as P_k. Given an input image x, the prompted input is expressed as:

$$x' = \mathrm{Prompt}(x, P_k), \tag{1}$$

where $\mathrm{Prompt}(\cdot)$ is a deterministic operation such as patch prepending, pixel-level embedding, or channel-wise modulation. The prompted input is then processed by the client's pre-trained model:

$$z_k = f_k(x'). \tag{2}$$

Throughout the training process, the backbone $f_k(\cdot)$ remains *frozen*, and only the prompt parameters P_k are optimized. Client k's local task objective is given by:

$$\mathcal{L}(P_k; \mathcal{D}_k) = \mathbb{E}_{(x,y)\sim\mathcal{D}_k}\, \ell\big(f_k(\mathrm{Prompt}(x, P_k)), y\big), \tag{3}$$

where $\ell(\cdot)$ is a client loss, such as cross-entropy. The goal is to learn prompts that are both effective locally and *compatible across clients* for enabling federated knowledge sharing across heterogeneous client models under non-IID data.

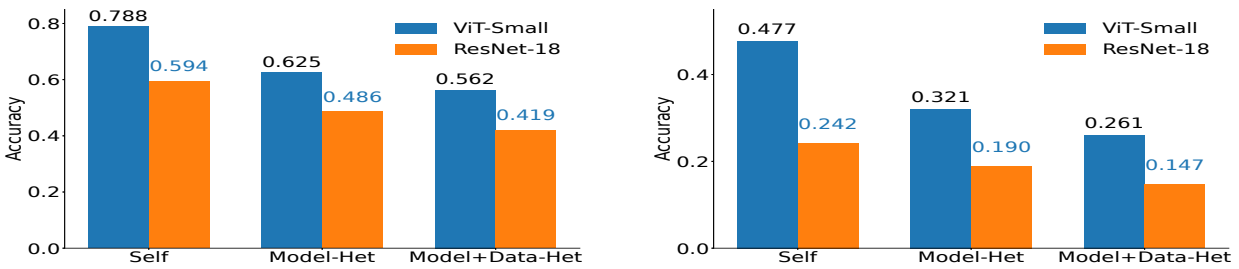
3.2 Problem Formulation

In federated learning based on visual prompting, each client k holds a private dataset $\mathcal{D}_k$ and a pre-trained backbone model $f_k(\cdot)$. Clients' models may differ in architecture and size due to their heterogeneous computational capacities. A learnable visual prompt P_k is optimized locally while keeping $f_k(\cdot)$ frozen. The local learning objective is:

$$\min_{P_k}\ \mathcal{L}(P_k; \mathcal{D}_k), \tag{4}$$

where P_k is initialised from the global prompt P_{global} at the beginning of each round and updated locally via gradient descent. The global prompt P_{global} serves as a fixed reference during local training and is not updated by the client; it is only updated by the server through aggregation at the end of each round. After local training, clients upload their prompt updates to the server for aggregation:

$$P_{\text{global}} = \mathrm{Aggregate}(\{P_k\}_{k=1}^{K}). \tag{5}$$



(a) CIFAR-10 (b) Tiny-ImageNet

Figure 3: Top-1 accuracy comparison of visual prompt performance on CIFAR-10 and Tiny-ImageNet. The x-axis denotes three evaluation settings: **Self** (prompt applied to its own model and data), **Model-Het** (prompt applied across different model architectures), and **Model+Data-Het** (prompt applied across both different models and non-IID data).

Motivation. Directly aggregating prompts is challenging for two reasons. **(1) Model Heterogeneity.** Since $f_k(\cdot)$ varies in architecture and size, the same prompt may induce inconsistent feature modulation across clients. **(2) Data Heterogeneity.** Non-IID local distributions $\{p_k(x, y)\}$ cause locally learned prompts to drift toward client-specific semantics, degrading their aggregated performance. To further validate this observation, we conduct a two-client study and examine how visual prompts perform when used under different levels of model heterogeneity. Specifically, we compare three settings corresponding to model heterogeneity: 1) **Self**. Each client uses its own prompt with its own model and data; 2) **Model-Het**. The prompt learned on one client is applied to a different model architecture but with the same data; and 3) **Model+Data-Het**. The prompt is used across different model architectures and non-IID data distributions. Fig. 3 presents the results. It can be observed that both clients achieve the highest accuracy when using their own prompts, while performance decreases notably when prompts are used with different model architectures. The degradation becomes even more profound when both model architectures and data distributions differ. This trend is consistent across CIFAR-10 and Tiny-ImageNet, which demonstrates that visual prompts are highly dependent on their trained models and local data. Consequently, directly aggregating or reusing prompts across heterogeneous client models results in significant performance loss. Thus, the goal is to learn prompts P_k that remain effective for each local client while maintaining global consistency across heterogeneous models and data distributions.

3.3 FedAura

FedAura enables collaborative prompt learning across heterogeneous client models without updating their backbone parameters. In each training round, clients optimize their local visual prompts on their private data, and the server aggregates the updated prompts to form a global prompt. This design allows clients to benefit from shared knowledge while maintaining computational efficiency and model flexibility.

However, naive prompt aggregation is prone to failure due to inconsistent feature modulation across heterogeneous client models and prompt drift under non-IID data. To address these issues, FedAura introduces two contrastive regularizations during local prompt training. The *model-contrastive loss* encourages the prompted representation to remain aligned with the representation of the original backbone and preserve feature consistency across different architectures. The *prompt-contrastive loss* aligns each local prompt with the global prompt representation maintained on the server to prevent prompt divergence across clients.

Through alternating local prompt optimization and global prompt aggregation, FedAura learns prompts that are both locally adaptive and globally coherent to enable effective collaboration across heterogeneous models and data distributions. An overview of FedAura is illustrated in Fig. 2.

Model-Contrastive Loss. To address feature inconsistency caused by model heterogeneity, FedAura introduces a model-contrastive loss that aligns the representations of the prompted model with those of the original pre-trained backbone. Given an input x, we denote the backbone representation as:

$$z_k = f_k(x), \tag{6}$$

the prompted representation and the representation produced by the backbone with a prompt from the last round as:

$$z'_k = f_k(\text{Prompt}(x, P_k)), z_k^{\text{pre}} = f_k(\text{Prompt}(x, P_k^{\text{pre}})). \tag{7}$$

We apply an InfoNCE contrastive objective Chen et al. (2020) that forces z'_k toward z_k and pushes it away from z_k^{pre}:

$$\mathcal{L}_{\text{m}} = -\log \frac{\exp(\text{sim}(z'_k, z_k)/\tau)}{\exp(\text{sim}(z'_k, z_k)/\tau) + \exp(\text{sim}(z'_k, z_k^{\text{pre}})/\tau)}, \tag{8}$$

where $\text{sim}(\cdot, \cdot)$ denotes cosine similarity and τ is a temperature parameter. This loss preserves the semantic structure of the backbone feature space and stabilizes prompt behavior across heterogeneous client models.

Prompt-Contrastive Loss. To mitigate prompt divergence under non-IID settings, FedAura maintains a global prompt P_{global} and aligns local prompts toward it. We have:

$$\mathcal{L}_{\text{p}} = -\log \frac{\exp(\text{sim}(z'_k, z^{\text{global}})/\tau)}{\exp(\text{sim}(z'_k, z^{\text{global}})/\tau) + \exp(\text{sim}(z'_k, z_k^{\text{pre}})/\tau)}, \tag{9}$$

where $z^{\text{global}} = f_k(\text{Prompt}(x, P_{\text{global}}))$. This loss reduces prompt drift among clients and improves the generalization of the aggregated prompt. Here, z^{global} in $\mathcal{L}_p$ is treated as a fixed anchor during local optimisation and does not receive gradients; only P_k is updated by the local optimizer.

Final Local Training Objective. The overall training loss for clients' local models combines the supervised loss with the two contrastive regularization terms:

$$\mathcal{L}_k = \mathcal{L}_{\text{sup}} + \lambda_{\text{m}}\mathcal{L}_{\text{m}} + \lambda_{\text{p}}\mathcal{L}_{\text{p}}. \tag{10}$$

Here, $\mathcal{L}_{\text{sup}}$ supervises the visual prompt to adapt the pre-trained model to the client's local data. Coefficient λ_{m} controls the strength of the model-contrastive loss $\mathcal{L}_{\text{m}}$, which preserves the semantic structure of the backbone representation and stabilizes the modulation effect of the prompt across heterogeneous architectures. Coefficient λ_{p} scales the prompt-contrastive loss $\mathcal{L}_{\text{p}}$, which aligns local prompts with the global prompt representation to mitigate prompt drift on non-IID data. By jointly optimizing these terms, FedAura learns prompts that are both locally effective and globally coherent to enable federated learning across heterogeneous client models and data distributions. Algorithm 1 of Appendix A.1 presents the pseudocode of the training process. And we also provide the convergence analysis of FedAura in Appendix A.2

4 Experiments

4.1 Experimental Settings

Datasets. We evaluate FedAura on four standard image classification benchmarks: CIFAR-10 & 100 Krizhevsky & Hinton (2009), Tiny-ImageNet Le & Yang (2015), and Flower-102 Nilsback & Zisserman (2008).

Models. To explicitly model the heterogeneity in computational resources across clients in federated learning, we assign each client a model whose architectural complexity matches its device capacity. The backbone architectures used in our experiments include ResNet-18 He et al. (2016), ResNet-50 He et al. (2016), ViT-Small Dosovitskiy et al. (2021), ViT-Base Dosovitskiy et al. (2021), DeiT-Small Touvron et al. (2021a), DeiT-Base Touvron et al. (2021a), CaiT-Tiny Touvron et al. (2021b), and CaiT-Small Touvron et al. (2021b). These cover a broad spectrum of CNN- and Transformer-based vision models.

Baselines. We compare FedAura with prompt-based parameter aggregation methods (PLAN Gong et al. (2026)), knowledge distillation-based methods, including FedMD Li & Wang (2019), FedDF Lin et al. (2020), and FedDFA Wang et al. (2025) (SOTA FL based on knowledge distillation), and partial-training approaches, including HeteroFL Diao et al. (2020) and FedEcover Liang et al. (2025) (SOTA FL based on partial learning). Since PLAN requires all clients to share the same backbone architecture and is restricted to Transformer-based models, it cannot be directly compared with FedAura under the heterogeneous backbone setting. A dedicated comparison between FedAura and PLAN under an identical ViT-B/16 backbone is provided in Appendix B.2.

Implementation Details. All images are resized to 224×224, and visual prompts are implemented as learnable padding tokens appended to the input image. For data partitioning, each dataset is split into 100 client subsets. IID partitions are obtained by uniform random sampling, while non-IID partitions are generated using a Dirichlet distribution $\text{Dir}(\beta)$ over label proportions, where a smaller β indicates higher heterogeneity. For convolutional backbones (e.g., ResNet families), we use SGD with a learning rate of 0.01 and momentum 0.9; for ViT-based backbones, we use AdamW with weight decay 0.01. The backbone parameters remain frozen during training, and only the visual prompts are optimized. We set the local training epoch to local_epoch $= 10$, and use default contrastive hyperparameters $\lambda_{\text{p}} = 1$, $\lambda_{\text{m}} = 1$, and temperature $\tau = 0.5$. Full details of implementation are provided in Appendix C.

Table 1: Comparison of top-1 accuracy (%) under different data distributions, where $\beta = 1$ and 10 denotes that each client has a proportion of the labels following a Dirichlet distribution with a parameter of $\beta = 1$ and $\beta = 10$.

Method	CIFAR-10			CIFAR-100			Tiny-ImageNet			Flower-102		
	IID	$\beta = 1$	$\beta = 10$	IID	$\beta = 1$	$\beta = 10$	IID	$\beta = 1$	$\beta = 10$	IID	$\beta = 1$	$\beta = 10$
FedMD	71.57	64.85	68.84	60.11	50.21	57.14	46.61	35.41	44.79	51.19	39.03	47.51
FedDF	72.79	66.68	70.71	60.32	50.75	57.39	46.94	35.37	43.83	52.48	43.15	50.08
Fed-DFA	74.38	63.92	71.95	59.83	50.54	55.03	46.08	34.77	42.15	51.73	40.53	50.70
HeteroFL	53.31	46.84	50.05	45.81	40.37	44.59	31.18	18.92	29.41	55.49	43.92	52.61
FedEcover	55.15	48.79	52.51	50.92	41.18	47.61	33.57	19.31	32.13	56.71	45.11	53.05
FedAura	**87.18**	**82.86**	**86.25**	**69.63**	**64.62**	**68.67**	**55.77**	**49.48**	**55.42**	**64.96**	**58.12**	**63.31**

4.2 Overall Comparison

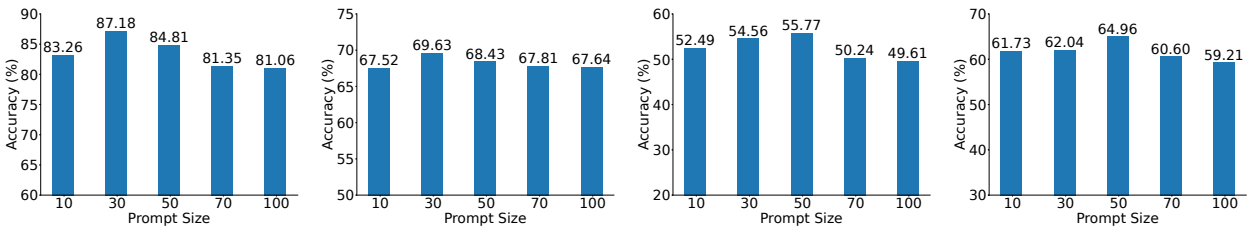
Top-1 Accuracy. Table 1 compares FedAura with representative baselines under both IID and non-IID data distributions across CIFAR-10, CIFAR-100, Tiny-ImageNet, and Flower-102. Four key observations are as follows.

- Compared to FedMD and FedDF, FedAura consistently achieves a higher accuracy across all datasets and heterogeneity settings. For instance, on CIFAR-10 with $\beta = 1$, FedAura improves accuracy by 18.01% over FedMD and 16.18% over FedDF. This suggests that logit-level alignment in KD-based methods is insufficient for transferring fine-grained semantic knowledge across heterogeneous models, whereas FedAura explicitly aligns feature spaces via model-contrastive learning.
- HeteroFL and FedEcover mitigate architectural heterogeneity by training sub-networks or modular components. When data heterogeneity increases from IID to $\beta = 1$, HeteroFL experiences a performance drop of 9.93%, while FedEcover drops by 10.5%. This indicates that they suffer from uneven optimization and degraded global representation quality under non-IID data. In contrast, FedAura reduces the corresponding performance drop to only 5.62%. This shows that prompt-contrastive alignment effectively stabilizes prompt semantics across heterogeneous client models.
- Across all datasets, FedAura consistently gains the best top-1 accuracy under both IID and non-IID settings. On the Flower-102 dataset, FedAura improves accuracy by 8.07% over the best baseline in comparison. This demonstrates its ability to enable knowledge sharing across diverse model architectures and data distributions. These results verify the effectiveness of Federated Visual Prompting with representation alignment in addressing model heterogeneity.
- FedAura also exhibits stronger robustness to data heterogeneity. When shifting from IID to non-IID distributions ($\beta = 1$ and $\beta = 10$), existing methods show notable accuracy drops. For instance, on CIFAR-10, PLAN degrades by 9.35% at $\beta = 1$ and 3.71% at $\beta = 10$, and Fed-DFA drops by 10.46% and 2.43%, respectively. In contrast, FedAura reduces the corresponding degradation to only 4.32% and 0.93%. This demonstrates that prompt-contrastive alignment effectively stabilizes cross-client prompt semantics under highly skewed data. Similar behavior is observed on CIFAR-100, Tiny-ImageNet, and Flower-102, which confirms that FedAura effectively mitigates data heterogeneity and maintains stable performance across diverse data distributions.

4.3 In-Depth Evaluation

Impact of Prompt Size. We investigate the effect of prompt size on the performance of FedAura. Following prior work on visual prompting, we implement prompts via learnable padding and vary the prompt size within $\{10, 30, 50, 70, 100\}$ pixels. Fig. 4 shows the results on CIFAR-10 & 100, Tiny-ImageNet, and Flower-102. We observe a consistent trend across all datasets: model accuracy improves initially as the prompt size increases, but begins to decline when the prompt becomes excessively large. This phenomenon can be attributed to two main reasons. First, small prompts provide insufficient modulation capability and thus limit adaptation to client-specific distributions. Second, overly large prompts dominate the visual input and distort the feature representations learned by the backbone. This degrades generalization ability and reduces semantic consistency across clients. Overall, prompt sizes in the mid-range (e.g., 30 - 50) strike a favorable balance between expressiveness and stability, and are adopted in our experiments.

Impact of Negative Prompt Memory Size. In contrastive learning, negative samples are essential for preventing representation collapse and encouraging separation among semantically different instances. A more diverse negative set generally provides stronger contrastive signals and leads to more informative, stable representations. To evaluate this effect in FedAura, we vary the number of stored negative prompts M with $\{1, 3, 5\}$. As shown in Fig. 5, increasing the number of negatives from 1 to 3 yields a clear performance improvement. For example, on CIFAR-10, accuracy increases from 79.16% to 82.03%, corresponding to a gain of 2.87%. This is because additional negatives provide



(a) CIFAR-10 (b) CIFAR-100 (c) Tiny-ImageNet (d) Flower-102

Figure 4: Top-1 accuracy with varying sizes of visual prompts across four datasets, i.e., CIFAR-10, CIFAR-100, Tiny-ImageNet, and Flower-102.

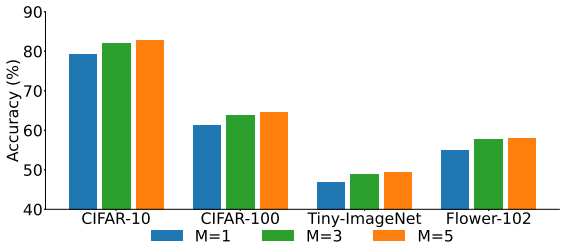


Figure 5: Top-1 accuracy of FedAura under different numbers of negative visual prompts, where there are 10 clients in the FL system with $\beta = 1$.

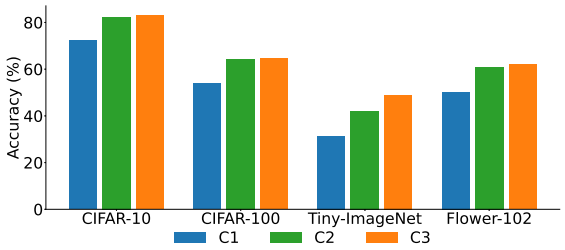


Figure 6: Top-1 accuracy of FedAura under different client configurations, where there are 10 clients in the FL system with $\beta = 1$.

stronger contrastive constraints, which help the model better distinguish prompt semantics across clients and rreduce local overfitting. However, when the memory size increases further from 3 to 5, the performance gain becomes marginal. On CIFAR-10, accuracy improves from 82.03% to 82.86%, by only 0.83%. Once the negative set sufficiently captures prompt diversity, new negatives tend to be semantically redundant and contribute a limited additional alignment signal. Overall, a moderate negative memory size (e.g., 3) provides a proper balance between representational discrimination and memory efficiency.

Impact of Client Capacities. As discussed in Section 1, model heterogeneity can arise from differences in clients' computational capacities. For instance, high-capacity clients may deploy larger ViT-base backbones, while low-capacity clients may only afford small backbones like ViT-tiny. This experiment evaluates how the difference in client capacities affects the performance of FedAura. We consider three capacity configurations: **(C1)** 60% ViT-tiny clients, 20% ViT-small clients, and 20% ViT-base clients; **(C2)** 20% ViT-tiny, 60% ViT-small, and 20% ViT-base; and **(C3)** 20% ViT-tiny, 20% ViT-small, and 60% ViT-base. The results, shown in Fig. 6, indicate that performance improves as the proportion of high-capacity clients increases. For example, FedAura achieves top-1 accuracies of 31.38%, 42.11%, and 48.84% on Tiny-ImageNet under configurations C1, C2, and C3, respectively. This suggests that high-capacity clients extract richer semantic features that can be effectively shared and aligned through our model- and prompt-contrastive objectives. More importantly, these results indicate that FedAura is able to *leverage* the knowledge contributed by high-capacity clients without overfitting to their feature space.

Client Participation Rate. We further investigate how the number of participating clients affects the performance of FedAura. We vary the number of active clients per round from 10 to 40 in steps of 10, while keeping the total number of clients fixed at 100. As shown in Table 2, the performance steadily improves as more clients participate in each communication round. For example, on CIFAR-100 with $\beta = 10$, FedAura achieves 68.67%, 69.15%, 69.93%, and 70.23% accuracy with 10, 20, 30, and 40 participating clients, respectively. This trend is expected: a larger participation rate increases the diversity and coverage of local data observed by the global prompt, leading to more consistent prompt semantics and better model-level alignment across clients. These results indicate that FedAura makes effective use of the information contributed by participating clients to secure steady performance improvements as more clients join the training process.

Table 2: Comparison of top-1 accuracy under different client participation rates across four datasets in FedAura. All experiments are conducted under a non-IID client data partition with $\beta = 10$.

Dataset	Number of Clients			
	10	20	30	40
CIFAR-10	86.25	86.96	87.88	**88.26**
CIFAR-100	68.67	69.15	69.93	**70.23**
Tiny-ImageNet	55.42	55.97	56.42	**57.12**
Flower-102	63.31	64.03	64.87	**65.14**

Table 3: Ablation study of FedAura's top-1 accuracy (%) with different losses on four datasets, where $\mathcal{L}_{\text{sup}}$, $\mathcal{L}_{\text{sup}}+\mathcal{L}_{\text{m}}$, $\mathcal{L}_{\text{sup}}+\mathcal{L}_{\text{p}}$, and FedAura denote the use of only supervised loss, supervised & model-contrastive loss, supervised & prompt-contrastive loss, and all three losses combined, respectively. All experiments are conducted under a non-IID client data partition with $\beta = 1$.

Dataset	Loss			
	$\mathcal{L}_{\text{sup}}$	$\mathcal{L}_{\text{sup}}+\mathcal{L}_{\text{m}}$	$\mathcal{L}_{\text{sup}}+\mathcal{L}_{\text{p}}$	FedAura
CIFAR-10	72.19	76.76	78.88	**82.86**
CIFAR-100	57.58	59.39	61.63	**64.62**
Tiny-ImageNet	44.55	46.94	47.92	**49.48**
Flower-102	52.64	55.36	56.22	**58.12**

4.4 Ablation Study

To evaluate the contribution of each of the three loss components in FedAura, we conduct ablation experiments on four datasets under the non-IID setting with $\beta = 1$ (Table 3). We compare the use of only the supervised loss $\mathcal{L}_{\text{sup}}$, supervised combined with model-contrastive loss $\mathcal{L}_{\text{sup}}+\mathcal{L}_m$, supervised combined with prompt-contrastive loss $\mathcal{L}_{\text{sup}}+\mathcal{L}_p$, and the full FedAura objective that combines all three loss components. On CIFAR-10, introducing $\mathcal{L}_m$ improves top-1 accuracy by 4.57% over the supervised baseline. This shows that model-level contrastive alignment helps mitigate model heterogeneity by aligning feature representations across different backbone architectures. In contrast, adding $\mathcal{L}_p$ yields a further gain of 6.69%, as prompt-level contrastive alignment encourages cross-client semantic consistency and thus alleviates data heterogeneity under non-IID settings. Combining both losses leads to the best result, outperforming the baseline by 10.67%. We observe consistent trends on CIFAR-100, Tiny-ImageNet, and Flower-102. $\mathcal{L}_m$ and $\mathcal{L}_p$ play complementary roles: $\mathcal{L}_m$ reduces the representational gap across heterogeneous model architectures, while $\mathcal{L}_p$ reduces the semantic gap induced by heterogeneous local data. FedAura, which jointly enforces both forms of alignment, achieves stable accuracy improvements of $4.93\% - 10.67\%$ across all datasets. The ablation study shows that both $\mathcal{L}_m$ and $\mathcal{L}_p$ are indispensable, and their joint optimization is a key to achieving stable performance across heterogeneous models and non-IID data distributions.

4.5 Efficiency Analysis

Beyond accuracy, we evaluate FedAura's resource efficiency in terms of trainable parameters, total upload volume, and total training FLOPs required to reach a target accuracy. A full quantitative comparison is provided in Appendix B. The key findings are as follows. In the heterogeneous backbone setting (Table 6), FedAura trains only 0.091 M prompt parameters per client and uploads 0.37 MB per round, reducing total upload by up to $912\times$ and total FLOPs by up to $7.8\times$ compared to the strongest baselines. In the single-backbone setting (Table 8), FedAura reduces FLOPs per step by $3.25\times$ compared to PromptFL (34.00 vs. 110.52 GFLOPs) because its input-level visual prompt avoids the deep backward propagation required by PromptFL's layer-wise token embeddings, and further reduces total upload by $2.16\times$ through faster convergence.

5 Conclusion

This paper presented FedAura, a federated visual prompting scheme that enables federated learning across heterogeneous client models without sharing model parameters. Instead of aggregating full model weights or logits, FedAura learns and exchanges lightweight visual prompts among clients, allowing clients with different backbone architectures to share their knowledge efficiently. To address the challenges of model heterogeneity and non-IID data, FedAura incorporates a model-contrastive loss to maintain consistent feature modulation across diverse backbones, and a prompt-contrastive loss to stabilize prompt semantics across clients. Compared to state-of-the-art knowledge-distillation and partial-training baselines, FedAura improves model accuracy up to 18.40%. In terms of resource efficiency, FedAura reduces total upload by up to $912\times$ and total training FLOPs by up to $7.8\times$ compared to the baselines. These results demonstrate that input-level prompt collaboration is a promising and resource-efficient direction for model-heterogeneous federated learning.

REFERENCES

Samiul Alam, Luyang Liu, Ming Yan, and Mi Zhang. Fedrolex: Model-heterogeneous federated learning with rolling sub-model extraction. *Advances in Neural Information Processing Systems*, 35:29677–29690, 2022.

Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pp. 1597–1607, 2020.

Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International Conference on Machine Learning*, pp. 2089–2099, 2021.

Rong Dai, Li Shen, Fengxiang He, Xinmei Tian, and Dacheng Tao. Dispfl: Towards communication-efficient personalized federated learning via decentralized sparse training. In *International Conference on Machine Learning*, pp. 4587–4604, 2022.

Ittai Dayan, Holger R Roth, Aoxiao Zhong, Ahmed Harouni, Amilcare Gentili, Anas Z Abidin, Andrew Liu, Anthony Beardsworth Costa, Bradford J Wood, Chien-Sung Tsai, et al. Federated learning for predicting clinical outcomes in patients with COVID-19. *Nature Medicine*, 27(10):1735–1743, 2021.

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

Enmao Diao, Jie Ding, and Vahid Tarokh. HeteroFL: Computation and communication efficient federated learning for heterogeneous clients. *arXiv preprint arXiv:2010.01264*, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*, 2021.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33:3557–3568, 2020.

Jie Feng, Can Rong, Funing Sun, Diansheng Guo, and Yong Li. Pmf: A privacy-preserving human mobility prediction framework via federated learning. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(1):1–21, 2020.

Shuai Gong, Chaoran Cui, Chunyun Zhang, Wenna Wang, Xiushan Nie, and Lei Zhu. Federated domain generalization via prompt learning and aggregation. *IEEE Transactions on Information Forensics and Security*, 2026.

Tao Guo, Song Guo, and Junxiao Wang. pFedPrompt: Learning personalized prompt for vision-language models in federated learning. In *The Web Conference*, pp. 1364–1374, 2023a.

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, 2023b.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Qidong Huang, Xiaoyi Dong, Dongdong Chen, Weiming Zhang, Feifei Wang, Gang Hua, and Nenghai Yu. Diversity-aware meta visual prompting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10878–10887, 2023.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pp. 709–727. Springer, 2022.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143, 2020.

Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

Alex Krizhevsky and Geoffrey Hinton. CIFAR-10 dataset, 2009. URL `https://www.cs.toronto.edu/~kriz/cifar.html`.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical Report, Toronto, ON, Canada*, 2009.

Yann Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Ang Li, Jingwei Sun, Pengcheng Li, Yu Pu, Hai Li, and Yiran Chen. Hermes: An efficient federated learning framework for heterogeneous mobile clients. In *27th Annual International Conference on Mobile Computing and Networking*, pp. 420–437, 2021a.

Daliang Li and Junpu Wang. FedMD: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

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

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pp. 12888–12900. PMLR, 2022.

Lin Li, Haoyan Guan, Jianing Qiu, and Michael Spratling. One prompt word is enough to boost adversarial robustness for pre-trained vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24408–24419, 2024b.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020a.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In I. Dhillon, D. Papailiopoulos, and V. Sze (eds.), *Proceedings of Machine Learning and Systems*, volume 2, pp. 429–450, 2020b.

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pp. 6357–6368, 2021b.

Juntao Liang, Lan Zhang, Xiangmou Qu, and Jun Wang. FedEcover: Fast and stable converging model-heterogeneous federated learning with efficient-coverage submodel extraction. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, pp. 2575–2587. IEEE, 2025.

Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(3):2031–2063, 2020.

Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33:2351–2363, 2020.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*, pp. 1273–1282, 2017.

Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *International Conference on Machine Learning*, pp. 4615–4625, 2019.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729, 2008.

Kunjal Panchal, Sunav Choudhary, Subrata Mitra, Koyel Mukherjee, Somdeb Sarkhel, Saayan Mitra, and Hui Guan. Flash: concept drift adaptation in federated learning. In *International Conference on Machine Learning*, pp. 26931–26962, 2023.

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*, pp. 8748–8763. PmLR, 2021.

Yifan Shi, Li Shen, Kang Wei, Yan Sun, Bo Yuan, Xueqian Wang, and Dacheng Tao. Improving the model consistency of decentralized federated learning. In *International conference on machine learning*, pp. 31269–31291. PMLR, 2023.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357, 2021a.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *IEEE/CVF International Conference on Computer Vision*, pp. 32–42, 2021b.

Haozhao Wang, Yichen Li, Wenchao Xu, Ruixuan Li, Yufeng Zhan, and Zhigang Zeng. DaFKD: Domain-aware federated knowledge distillation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20412–20421, 2023a.

Haozhao Wang, Yabo Jia, Meng Zhang, Qinghao Hu, Hao Ren, Peng Sun, Yonggang Wen, and Tianwei Zhang. FedDSE: Distribution-aware sub-model extraction for federated learning over resource-constrained devices. In *The Web Conference*, pp. 2902–2913, 2024a.

Haozhao Wang, Haoran Xu, Yichen Li, Yuan Xu, Ruixuan Li, and Tianwei Zhang. Fedcda: Federated learning with cross-rounds divergence-aware aggregation. In *The Twelfth International Conference on Learning Representations*, 2024b.

Kaibin Wang, Qiang He, Feifei Chen, Chunyang Chen, Faliang Huang, Hai Jin, and Yun Yang. FlexiFed: Personalized federated learning for edge clients with heterogeneous model architectures. In *ACM Web Conference*, pp. 2979–2990, 2023b.

Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 37(6):1205–1221, 2019.

Zichen Wang, Feng Yan, Tianyi Wang, Cong Wang, Yuanchao Shu, Peng Cheng, and Jiming Chen. Fed-DFA: Federated distillation for heterogeneous model fusion through the adversarial lens. In *AAAI Conference on Artificial Intelligence*, volume 39, pp. 21429–21437, 2025.

Wikipedia. California consumer privacy act, 2018a. URL `https://en.wikipedia.org/wiki/California_Consumer_Privacy_Act`.

Wikipedia. General data protection regulation, 2018b. URL `https://en.wikipedia.org/wiki/General_Data_Protection_Regulation`.

Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. Communication-efficient federated learning via knowledge distillation. *Nature Communications*, 13(1):2032, 2022.

Chunxu Zhang, Guodong Long, Tianyi Zhou, Zijian Zhang, Peng Yan, and Bo Yang. GpFedRec: Graph-guided personalization for federated recommendation. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4131–4142, 2024a.

Jie Zhang, Song Guo, Xiaosong Ma, Haozhao Wang, Wenchao Xu, and Feijie Wu. Parameterized knowledge transfer for personalized federated learning. *Advances in Neural Information Processing Systems*, 34:10092–10104, 2021.

Yichi Zhang, Yinpeng Dong, Siyuan Zhang, Tianzan Min, Hang Su, and Jun Zhu. Exploring the transferability of visual prompting for multimodal large language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26562–26572, 2024b.

Hanhan Zhou, Tian Lan, Guru Prasadh Venkataramani, and Wenbo Ding. Every parameter matters: Ensuring the convergence of federated learning with dynamic heterogeneous models reduction. *Advances in Neural Information Processing Systems*, 36:25991–26002, 2023.

A APPENDIX

A.1 PSEUDOCODE

Algorithm 1 presents the training procedure in FedAura. The server initializes the global prompt (Line 2) and coordinates client selection (Line 4). Clients perform local prompt training by optimizing the task loss, model-contrastive loss, and prompt-contrastive loss (Lines 5 - 15), then upload their updated prompts to the server (Line 18). The server aggregates these prompts into a new global prompt (Line 19) and sends it to clients for the next round (Line 20).

Algorithm 1 Training Procedure of FedAura

1: **Input:** Clients $\{1, \ldots, K\}$ with private datasets $\{\mathcal{D}_k\}$ and frozen backbones $\{f_k\}$; rounds R; local epochs E; batch size B; learning rate η; temperature τ; loss weights $\lambda_{\text{m}}, \lambda_{\text{p}}$
2: **Server init:** initialize P_{global}.
3: **for** $r = 1$ **to** R **do**
4: Sample participating clients $\mathcal{S}_r \subseteq \{1, \ldots, K\}$
5: **for** $k \in \mathcal{S}_r$ **do**
6: $P_k \leftarrow P_{\text{global}}$
7: **for** $e = 1$ **to** E **do**
8: $z_k \leftarrow f_k(x_i)$;
9: $z_k^{'} \leftarrow f_k(\text{Prompt}(x_i, P_k))$;
10: $z_k^{pre} \leftarrow f_k(\text{Prompt}(x_i, P_k^{pre}))$;
11: $z^{global} \leftarrow f_k(\text{Prompt}(x_i, P_{\text{global}}))$
12: $\mathcal{L}_{\text{sup}} \leftarrow \frac{1}{B} \sum_{i=1}^{B} \ell(z_i', y_i)$
13: $\mathcal{L}_{\text{m}} \leftarrow \text{CL}(z_k', z_k, z_k^{pre})$ in Eq. 8
14: $\mathcal{L}_{\text{p}} \leftarrow \text{CL}(z_k', z_{\text{global}}, z_k^{pre})$ in Eq. 9
15: $\mathcal{L}_k \leftarrow \mathcal{L}_{\text{sup}} + \lambda_{\text{m}}\mathcal{L}_{\text{m}} + \lambda_{\text{p}}\mathcal{L}_{\text{p}}$
$P_k \leftarrow P_k - \eta \nabla_{P_k} \mathcal{L}_k$
16: **end for**
17: **end for**
18: Send P_k to server
19: $P_{\text{global}} \leftarrow \sum_{k \in \mathcal{S}_r} \frac{|\mathcal{D}_k|}{\sum_{j \in \mathcal{S}_r} |\mathcal{D}_j|} P_k$
20: Broadcast updated P_{global} to clients
21: **end for**

A.2 CONVERGENCE ANALYSIS

A.2.1 PROBLEM SETUP

Let $P \in \mathbb{R}^d$ denote the shared visual prompt. There are K clients; client k holds a private dataset $\mathcal{D}_k$ of size $|\mathcal{D}_k|$ and a *frozen* backbone $f_k(\cdot)$. Define data-weight $w_k = |\mathcal{D}_k|/|\mathcal{D}|$ where $|\mathcal{D}| = \sum_k |\mathcal{D}_k|$. The global loss is

$$F(P) \;=\; \sum_{k=1}^{K} w_k \, \mathcal{L}_k(P), \qquad \mathcal{L}_k(P) \;=\; \mathcal{L}_{\text{sup}}(P) + \lambda_m \mathcal{L}_m(P) + \lambda_p \mathcal{L}_p(P),$$

where $\mathcal{L}_{\text{sup}}$ is the supervised (cross-entropy) loss, $\mathcal{L}_m$ is the model-contrastive loss (Eq. 9 of the main paper), and $\mathcal{L}_p$ is the prompt-contrastive loss (Eq. 11), with non-negative coefficients $\lambda_m, \lambda_p \geq 0$. Let $P^* = \arg\min_P F(P)$ and $F^* = F(P^*)$.

At the beginning of round r, the server broadcasts $P_{\text{global}}^{(r)}$ to a sampled subset S_r of $m = |S_r|$ clients. Each client $k \in S_r$ initialises $P_k^{(r,0)} = P_{\text{global}}^{(r)}$ and runs E steps of local SGD. The server then aggregates

$$P_{\text{global}}^{(r+1)} \;=\; \sum_{k \in S_r} w_k \, P_k^{(r,E)}. \tag{11}$$

Assumption A.1 (L-Smoothness)**.** For every client k and all $P, Q \in \mathbb{R}^d$,

$$\mathcal{L}_k(P) - \mathcal{L}_k(Q) - \langle \nabla \mathcal{L}_k(Q),\, P - Q \rangle \;\leq\; \frac{L}{2} \, \|P - Q\|^2 ,$$

where the effective smoothness constant $L = L_{\text{sup}} + \lambda_m L_m + \lambda_p L_p$ absorbs the contributions of all three loss terms.

Assumption A.2 (Unbiased Stochastic Gradients with Bounded Variance)**.** Each client computes a stochastic gradient $\tilde{g}_k(P)$ satisfying

$$\mathbb{E}\big[\tilde{g}_k(P)\big] = \nabla\mathcal{L}_k(P), \qquad \mathbb{E}\big[\|\tilde{g}_k(P) - \nabla\mathcal{L}_k(P)\|^2\big] \;\le\; \sigma^2.$$

Assumption A.3 (Bounded Inter-Client Gradient Variance)**.** For all P,

$$\frac{1}{K}\sum_{k=1}^{K} \|\nabla\mathcal{L}_k(P) - \nabla F(P)\|^2 \;\le\; \sigma_g^2.$$

Assumption A.4 (Architectural Heterogeneity Bound)**.** For any two backbone networks f_k and $f_{k'}$ and any prompted input x',

$$\|f_k(x') - f_{k'}(x')\|^2 \;\le\; \delta_{\text{arch}}^2,$$

where δ_{arch} quantifies the degree of model heterogeneity.

Assumption A.5 (Data Heterogeneity Bound)**.** The local optima satisfy

$$\frac{1}{K}\sum_{k=1}^{K} \|P_k^* - P^*\|^2 \;\le\; \zeta^2.$$

Assumption A.6 (Representation Alignment by $\lambda_m\mathcal{L}_m$)**.** The gradient of the model-contrastive loss satisfies, for all P_k and all k,

$$\langle \nabla_{P_k}\mathcal{L}_m(P_k),\, P_k - P^*\rangle \;\ge\; \mu_m \|P_k - P^*\|^2 - \frac{\delta_{\text{arch}}^2}{2\lambda_m + 2}, \qquad \mu_m > 0.$$

Geometrically, $\lambda_m\mathcal{L}_m$ acts as an implicit regulariser that aligns the prompted representation with the frozen backbone, thereby suppressing feature corruption caused by architectural heterogeneity and contributing curvature $\lambda_m\mu_m$ to the total objective.

Assumption A.7 (Proximal Effect of $\lambda_p\mathcal{L}_p$)**.** The gradient of the prompt-contrastive loss satisfies, for all P_k and all k,

$$\langle \nabla_{P_k}\mathcal{L}_p(P_k),\, P_k - P_{\text{global}}\rangle \;\ge\; \mu_p \|P_k - P_{\text{global}}\|^2 - \frac{\zeta^2}{2\lambda_p + 2}, \qquad \mu_p > 0.$$

Geometrically, $\lambda_p\mathcal{L}_p$ acts as a proximal term anchored at P_{global}, preventing local prompts from drifting toward client-specific semantics and contributing curvature $\lambda_p\mu_p$ to the total objective.

Assumption A.8 (μ-Strong Convexity (for Section A.2 only))**.** $F(P)$ is μ-strongly convex: for all $P, Q \in \mathbb{R}^d$,

$$F(Q) \;\ge\; F(P) + \langle \nabla F(P),\, Q - P\rangle + \frac{\mu}{2}\|Q - P\|^2.$$

Assumption A.9 (General Convexity (for Section A.3 only))**.** $F(P)$ is convex (Assumption A.8 with $\mu = 0$).

By Assumptions A.6–A.7, both contrastive losses contribute additional curvature, yielding an *effective* strong-convexity constant

$$\mu_{\text{eff}} \;:=\; \mu + \lambda_m\mu_m + \lambda_p\mu_p \;>\; \mu. \tag{12}$$

Throughout, G^2 denotes a uniform upper bound on $\mathbb{E}\,\|\nabla\mathcal{L}_k(P)\|^2$ over all k and P.

A.2.2 Strongly Convex Case

Lemma A.10 (One-Step SGD Descent)**.** *Under Assumptions A.1–A.7 and A.8, for the local update* $P_k^{(r,t+1)} = P_k^{(r,t)} - \eta\,\tilde{g}_k^{(r,t)}$,

$$\mathbb{E}\left\|P_k^{(r,t+1)} - P^*\right\|^2 \;\le\; (1 - 2\eta\mu_{\text{eff}})\left\|P_k^{(r,t)} - P^*\right\|^2 - 2\eta\big(F(P_k^{(r,t)}) - F^*\big) + \eta^2\sigma^2. \tag{13}$$

Proof. Expand the squared norm:

$$\left\|P_k^{(r,t+1)} - P^*\right\|^2 = \left\|P_k^{(r,t)} - P^*\right\|^2 - 2\eta\left\langle \tilde{g}_k^{(r,t)},\, P_k^{(r,t)} - P^*\right\rangle + \eta^2\left\|\tilde{g}_k^{(r,t)}\right\|^2.$$

Taking expectation over the stochastic gradient noise and applying Assumption A.2:

$$\mathbb{E}\left\|\tilde{g}_k^{(r,t)}\right\|^2 \;\le\; \left\|\nabla\mathcal{L}_k(P_k^{(r,t)})\right\|^2 + \sigma^2.$$

By the μ_{eff}-strong convexity of $\mathcal{L}_k$ (which follows from Assumption A.8 combined with the extra curvature $\lambda_m\mu_m + \lambda_p\mu_p$ from Assumptions A.6–A.7):

$$\langle\nabla\mathcal{L}_k(P),\, P - P^*\rangle \;\ge\; \mu_{\text{eff}}\,\|P - P^*\|^2 + F(P) - F^*.$$

By L-smoothness and μ_{eff}-strong convexity, $\|\nabla\mathcal{L}_k(P)\|^2 \le 2L\big(\mathcal{L}_k(P) - \mathcal{L}_k^*\big)$. Substituting both bounds into the expanded squared norm:

$$\mathbb{E}\left\|P_k^{(r,t+1)} - P^*\right\|^2 \le \left\|P_k^{(r,t)} - P^*\right\|^2 - 2\eta\mu_{\text{eff}}\left\|P_k^{(r,t)} - P^*\right\|^2 - 2\eta\big(F(P_k^{(r,t)}) - F^*\big) + \eta^2\sigma^2,$$

which gives equation 13. □

Lemma A.11 (E-Step Drift Bound under Strong Convexity)**.** *Under the same assumptions, after E local SGD steps starting from $P_k^{(r,0)} = P_{\text{global}}^{(r)}$,*

$$\frac{1}{|S_r|}\sum_{k\in S_r}\mathbb{E}\left\|P_k^{(r,E)} - P^*\right\|^2 \;\le\; (1-\eta\mu_{\text{eff}})^E\left\|P_{\text{global}}^{(r)} - P^*\right\|^2 + \frac{\eta\sigma^2}{\mu_{\text{eff}}} + \frac{\lambda_m\delta_{\text{arch}}^2 + \lambda_p\zeta^2}{2\mu_{\text{eff}}}. \tag{14}$$

Proof. Apply Lemma A.10 recursively for $t = 0, 1, \ldots, E-1$:

$$\mathbb{E}\left\|P_k^{(r,E)} - P^*\right\|^2 \le (1-2\eta\mu_{\text{eff}})^E\left\|P_{\text{global}}^{(r)} - P^*\right\|^2 + \eta^2\sigma^2\sum_{t=0}^{E-1}(1-2\eta\mu_{\text{eff}})^{E-1-t}.$$

The geometric series satisfies $\sum_{t=0}^{E-1}(1-2\eta\mu_{\text{eff}})^t \le \frac{1}{2\eta\mu_{\text{eff}}}$, so the variance term contributes at most $\frac{\eta\sigma^2}{2\mu_{\text{eff}}}$. The residual heterogeneity from Assumptions A.6 and A.7 accumulates additively over E steps: each step incurs a bias of at most $\eta\big(\frac{\lambda_m\delta_{\text{arch}}^2}{2(\lambda_m+1)} + \frac{\lambda_p\zeta^2}{2(\lambda_p+1)}\big)$, and summing over the geometric series gives a total of $\frac{\lambda_m\delta_{\text{arch}}^2+\lambda_p\zeta^2}{2\mu_{\text{eff}}}$. Using $(1-2\eta\mu_{\text{eff}})^E \le (1-\eta\mu_{\text{eff}})^E$ for $\eta\mu_{\text{eff}} \le \frac{1}{2}$, and averaging over $k \in S_r$ by Jensen's inequality gives equation 14. □

Theorem A.12 (Linear Convergence under Strong Convexity)**.** *Suppose Assumptions A.1–A.7 and A.8 hold. Set the step size $\eta \le \frac{1}{4L}$ and define*

$$\rho \;\coloneqq\; (1-\eta\mu_{\text{eff}})^E \in (0,1), \qquad B \;\coloneqq\; \frac{\eta\sigma^2}{\mu_{\text{eff}}} + \frac{\lambda_m\delta_{\text{arch}}^2 + \lambda_p\zeta^2}{2\mu_{\text{eff}}}.$$

After R communication rounds,

$$\mathbb{E}\left\|P_{\text{global}}^{(R)} - P^*\right\|^2 \;\le\; \rho^R\left\|P_{\text{global}}^{(0)} - P^*\right\|^2 + \frac{B}{1-\rho}. \tag{15}$$

To achieve $\mathbb{E}\left\|P_{\text{global}}^{(R)} - P^*\right\|^2 \le \varepsilon$, FedAura requires

$$R \;=\; \mathcal{O}\left(\frac{1}{\eta\mu_{\text{eff}}E}\log\frac{\left\|P^{(0)} - P^*\right\|^2}{\varepsilon}\right)$$

communication rounds.

Proof. **Step 1: Aggregation contraction.** By Jensen's inequality applied to equation 11:

$$\mathbb{E}\left\|P_{\text{global}}^{(r+1)} - P^*\right\|^2 \;=\; \mathbb{E}\left\|\sum_{k\in S_r} w_k(P_k^{(r,E)} - P^*)\right\|^2 \;\le\; \sum_{k\in S_r} w_k\,\mathbb{E}\left\|P_k^{(r,E)} - P^*\right\|^2.$$

Step 2: Substituting the drift bound. Let $e^{(r)} = \mathbb{E}\left\|P_{\text{global}}^{(r)} - P^*\right\|^2$. Substituting Lemma A.11:

$$e^{(r+1)} \;\le\; \rho\, e^{(r)} + B.$$

Step 3: Unrolling the linear recurrence. Applying the recurrence $T = R$ times:

$$e^{(R)} \;\le\; \rho^R e^{(0)} + B\sum_{j=0}^{R-1}\rho^j \;=\; \rho^R e^{(0)} + B\cdot\frac{1-\rho^R}{1-\rho} \;\le\; \rho^R e^{(0)} + \frac{B}{1-\rho},$$

which gives equation 15.

Step 4: Sample complexity. Setting $\rho^R e^{(0)} \le \varepsilon/2$ and using $\log\rho^{-1} \ge \eta\mu_{\text{eff}}E$ gives $R \ge \frac{1}{\eta\mu_{\text{eff}}E}\log\frac{2e^{(0)}}{\varepsilon}$, yielding the stated round complexity. □

Remark A.13 (Role of λ_m and λ_p under strong convexity). The irreducible error floor is

$$\frac{B}{1-\rho} \;\approx\; \frac{\sigma^2}{\mu_{\text{eff}}m} + \frac{\lambda_m\delta_{\text{arch}}^2 + \lambda_p\zeta^2}{2\mu_{\text{eff}}^2\eta E}.$$

FedAura *simultaneously* (i) enlarges the denominator $\mu \to \mu_{\text{eff}} > \mu$, accelerating the linear rate $\rho \to \rho_{\text{FL}} < \rho$, and (ii) keeps the numerator controlled because $\lambda_m\delta_{\text{arch}}^2$ and $\lambda_p\zeta^2$ are small when the contrastive losses are effective.

A.2.3 General Convex Case

Lemma A.14 (Per-Step Progress under General Convexity). *Under Assumptions A.1–A.7 and A.9, for $\eta \le \frac{1}{2L}$,*

$$\mathcal{L}_k(P_k^{(r,t)}) - \mathcal{L}_k^* \;\le\; \frac{\left\|P_k^{(r,t)} - P^*\right\|^2 - \mathbb{E}\left\|P_k^{(r,t+1)} - P^*\right\|^2}{2\eta(1-L\eta)} + \frac{\eta\sigma^2}{2(1-L\eta)} + \frac{\lambda_m\delta_{\text{arch}}^2 + \lambda_p\zeta^2}{2(1-L\eta)}. \tag{16}$$

Proof. Expand the squared norm of one SGD step:

$$\mathbb{E}\left\|P_k^{(r,t+1)} - P^*\right\|^2 = \left\|P_k^{(r,t)} - P^*\right\|^2 - 2\eta\left\langle\nabla\mathcal{L}_k(P_k^{(r,t)}),\, P_k^{(r,t)} - P^*\right\rangle + \eta^2\mathbb{E}\left\|\tilde{g}_k^{(r,t)}\right\|^2.$$

By general convexity together with Assumptions A.6 and A.7, the gradient inner product satisfies

$$\langle\nabla\mathcal{L}_k(P),\, P - P^*\rangle \;\ge\; \mathcal{L}_k(P) - \mathcal{L}_k^* - \frac{\lambda_m\delta_{\text{arch}}^2 + \lambda_p\zeta^2}{2}. \tag{17}$$

By L-smoothness and Assumption A.2, $\mathbb{E}\left\|\tilde{g}_k\right\|^2 \le 2L\big(\mathcal{L}_k(P) - \mathcal{L}_k^*\big) + \sigma^2$. Substituting equation 17 and the variance bound into the expanded squared norm and rearranging:

$$2\eta(1-L\eta)\big(\mathcal{L}_k(P_k^{(r,t)}) - \mathcal{L}_k^*\big) \;\le\; \left\|P_k^{(r,t)} - P^*\right\|^2 - \mathbb{E}\left\|P_k^{(r,t+1)} - P^*\right\|^2 + \eta^2\sigma^2 + \eta\big(\lambda_m\delta_{\text{arch}}^2 + \lambda_p\zeta^2\big).$$

Dividing both sides by $2\eta(1-L\eta) > 0$ gives equation 16. □

Lemma A.15 (Client Drift Bound under General Convexity). *Under Assumptions A.1–A.7, since $P_k^{(r,0)} = P_{\text{global}}^{(r)}$,*

$$\frac{1}{E}\sum_{t=0}^{E-1}\mathbb{E}\left\|P_k^{(r,t)} - P_{\text{global}}^{(r)}\right\|^2 \;\le\; \eta^2E^2G^2(1-\lambda_p\mu_p\eta), \tag{18}$$

where G^2 is an upper bound on $\mathbb{E}\left\|\nabla\mathcal{L}_k(P)\right\|^2$.

Proof. Since $P_k^{(r,0)} = P_{\text{global}}^{(r)}$, after t steps, $P_k^{(r,t)} - P_{\text{global}}^{(r)} = -\eta\sum_{s=0}^{t-1}\tilde{g}_k^{(r,s)}$. By the Cauchy–Schwarz inequality:

$$\mathbb{E}\left\|P_k^{(r,t)} - P_{\text{global}}^{(r)}\right\|^2 \;\le\; t\eta^2\sum_{s=0}^{t-1}\mathbb{E}\left\|\tilde{g}_k^{(r,s)}\right\|^2 \;\le\; t^2\eta^2G^2.$$

The proximal pull of $\lambda_p\mathcal{L}_p$ toward P_{global} (Assumption A.7) reduces the effective step size by a factor $(1-\lambda_p\mu_p\eta)$ per step, so $\mathbb{E}\left\|P_k^{(r,t)} - P_{\text{global}}^{(r)}\right\|^2 \le t^2\eta^2G^2(1-\lambda_p\mu_p\eta)$. Averaging over $t = 0,\dots,E-1$ and using $\frac{1}{E}\sum_{t=1}^{E}t^2 \le E^2$ gives equation 18. □

Theorem A.16 ($\mathcal{O}(1/\sqrt{RE})$ Convergence under General Convexity)**.** *Suppose Assumptions A.1–A.7 and A.9 hold. Set* $\eta = \frac{1}{L\sqrt{RE}}$*. Then*

$$\frac{1}{RE}\sum_{r=1}^{R}\sum_{t=0}^{E-1}\mathbb{E}\big[F(P^{(r,t)}) - F^*\big] \;\le\; \frac{L\left\|P^{(0)} - P^*\right\|^2}{\sqrt{RE}} + \frac{\sigma^2}{2L\sqrt{RE}} + \frac{G^2(1-\lambda_p\mu_p/L)}{R} + \frac{\lambda_m\delta_{\text{arch}}^2 + \lambda_p\zeta^2}{2L}. \quad (19)$$

To reach an ε-optimal point, it suffices to take $T = \mathcal{O}\big((\sigma^2+G^2)/\varepsilon^2\big)$ communication rounds.

Proof. **Step 1: Summing the per-step bound over E steps.**

Summing equation 16 for $t = 0, \dots, E-1$ and using the telescoping identity $\sum_{t=0}^{E-1}\big(\left\|P_k^{(r,t)} - P^*\right\|^2 - \mathbb{E}\left\|P_k^{(r,t+1)} - P^*\right\|^2\big) = \left\|P_{\text{global}}^{(r)} - P^*\right\|^2 - \mathbb{E}\left\|P_k^{(r,E)} - P^*\right\|^2$:

$$\sum_{t=0}^{E-1}\big(\mathcal{L}_k(P_k^{(r,t)}) - \mathcal{L}_k^*\big) \;\le\; \frac{\left\|P_{\text{global}}^{(r)} - P^*\right\|^2}{2\eta(1-L\eta)} + \frac{E\eta\sigma^2}{2(1-L\eta)} + \frac{E\big(\lambda_m\delta_{\text{arch}}^2 + \lambda_p\zeta^2\big)}{2(1-L\eta)}.$$

The gap between $F(P^{(r,t)})$ and $\mathcal{L}_k(P_k^{(r,t)})$ is bounded by L times the client drift (Lemma A.15), contributing $L\eta^2E^2G^2(1-\lambda_p\mu_p\eta)$ per round.

Step 2: Summing over R rounds.

Summing over $r = 1, \dots, R$ and averaging by $\frac{1}{RE}$:

$$\frac{1}{RE}\sum_{r,t}\mathbb{E}[F(P^{(r,t)}) - F^*] \;\le\; \frac{\left\|P^{(0)} - P^*\right\|^2}{2\eta(1-L\eta)RE} + \frac{\eta\sigma^2}{2(1-L\eta)} + \frac{L\eta^2E^2G^2(1-\lambda_p\mu_p\eta)}{RE}\cdot R + \frac{\lambda_m\delta_{\text{arch}}^2 + \lambda_p\zeta^2}{2(1-L\eta)}.$$

Step 3: Substituting $\eta = (L\sqrt{RE})^{-1}$.

With this choice, $1 - L\eta \ge \frac{1}{2}$. Bounding each term:

Term 1 (optimisation): $\frac{\left\|P^{(0)}-P^*\right\|^2}{2\eta(1-L\eta)RE} \le \frac{L\left\|P^{(0)}-P^*\right\|^2}{\sqrt{RE}}$.

Term 2 (stochastic noise): $\frac{\eta\sigma^2}{2(1-L\eta)} \le \frac{\sigma^2}{2L\sqrt{RE}}$.

Term 3 (client drift): $\frac{L\eta^2E^2G^2(1-\lambda_p\mu_p\eta)}{E} = \frac{LG^2E\eta^2(1-\lambda_p\mu_p\eta)}{1} \le \frac{G^2(1-\lambda_p\mu_p/L)}{R}$.

Term 4 (heterogeneity residual): $\frac{\lambda_m\delta_{\text{arch}}^2+\lambda_p\zeta^2}{2(1-L\eta)} \le \frac{\lambda_m\delta_{\text{arch}}^2+\lambda_p\zeta^2}{2L}\cdot L = \frac{\lambda_m\delta_{\text{arch}}^2+\lambda_p\zeta^2}{2L}$.

Combining the four terms gives equation 19. □

Remark A.17 (Asymptotic floor under general convexity). As $R \to \infty$, the first three terms vanish and

$$\liminf_{R\to\infty}\frac{1}{RE}\sum_{r,t}\mathbb{E}[F(P^{(r,t)}) - F^*] \;\le\; \frac{\lambda_m\delta_{\text{arch}}^2 + \lambda_p\zeta^2}{2L}.$$

The $\lambda_m\delta_{\text{arch}}^2$ term is driven to zero as architectural heterogeneity $\delta_{\text{arch}} \to 0$ or as $\mathcal{L}_m$ becomes more effective; the $\lambda_p\zeta^2$ term vanishes as data distributions homogenise or as $\mathcal{L}_p$ tightens the prompt anchor.

A.2.4 Non-Convex Case

In the non-convex setting we target an approximate stationary point, measuring progress via $\mathbb{E}\left\|\nabla F(P)\right\|^2$.

Lemma A.18 (Smoothness-Based Descent Lemma)**.** *Under Assumption A.1, for $\eta \le \frac{1}{L}$,*

$$\frac{\eta}{2}\left\|\nabla F(P)\right\|^2 \;\le\; F(P) - \mathbb{E}[F(P - \eta\tilde{g})] + \frac{L\eta^2\sigma^2}{2}. \quad (20)$$

Proof. Apply L-smoothness with $P' = P - \eta\tilde{g}$:

$$\mathbb{E}[F(P')] \;\le\; F(P) + \langle \nabla F(P),\, -\eta \mathbb{E}[\tilde{g}] \rangle + \frac{L\eta^2}{2} \mathbb{E}\,\|\tilde{g}\|^2 .$$

Since $\mathbb{E}[\tilde{g}] = \nabla F(P)$ and $\mathbb{E}\,\|\tilde{g}\|^2 \le \|\nabla F(P)\|^2 + \sigma^2$ (Assumption A.2):

$$\mathbb{E}[F(P')] \;\le\; F(P) - \eta\left(1 - \frac{L\eta}{2}\right)\|\nabla F(P)\|^2 + \frac{L\eta^2\sigma^2}{2}.$$

Since $\eta \le 1/L$ implies $1 - L\eta/2 \ge 1/2$, rearranging gives equation 20. □

Lemma A.19 (Gradient Heterogeneity Bound)**.** *Under Assumptions A.3, A.6, and A.7,*

$$\frac{1}{K}\sum_{k=1}^{K} \|\nabla F(P) - \nabla \mathcal{L}_k(P)\|^2 \;\le\; \sigma_g^2 + \frac{L\big(\lambda_m \delta_{\text{arch}}^2 + \lambda_p \zeta^2\big)}{2}. \tag{21}$$

Proof. Decompose the gradient difference:

$$\nabla F(P) - \nabla \mathcal{L}_k(P) = \big[\nabla F(P) - \nabla \mathcal{L}_{\text{sup}}^{(k)}(P)\big] - \lambda_m \nabla \mathcal{L}_m(P) - \lambda_p \nabla \mathcal{L}_p(P).$$

The first bracket has squared norm bounded by σ_g^2 (Assumption A.3). For the model-contrastive term, L_m-smoothness and Assumption A.6 give $\|\lambda_m \nabla \mathcal{L}_m(P)\|^2 \le \lambda_m^2 L_m^2 \delta_{\text{arch}}^2 / L \le \frac{L\lambda_m \delta_{\text{arch}}^2}{2}$. An analogous bound applies to $\lambda_p \nabla \mathcal{L}_p(P)$ via Assumption A.7. Summing over k and averaging gives equation 21. □

Lemma A.20 (Client Drift under Non-Convexity)**.** *Under Assumption A.7,*

$$\mathbb{E}\left\|\bar{P}^{(r,E)} - P_{\text{global}}^{(r)}\right\|^2 \;\le\; E^2\eta^2(G^2 + \sigma^2)(1 - \lambda_p\mu_p\eta)^E, \tag{22}$$

where $\bar{P}^{(r,E)} = \sum_{k \in S_r} w_k P_k^{(r,E)}$.

Proof. Each client starts at $P_k^{(r,0)} = P_{\text{global}}^{(r)}$. After t steps the drift satisfies $\left\|P_k^{(r,t)} - P_{\text{global}}^{(r)}\right\|^2 \le t^2\eta^2(G^2 + \sigma^2)$ by Cauchy–Schwarz and Assumption A.2. The proximal pull of $\lambda_p \mathcal{L}_p$ (Assumption A.7) contracts this drift by a factor $(1 - \lambda_p\mu_p\eta)$ at every step, so $\left\|P_k^{(r,E)} - P_{\text{global}}^{(r)}\right\|^2 \le E^2\eta^2(G^2 + \sigma^2)(1 - \lambda_p\mu_p\eta)^E$. Averaging over $k \in S_r$ by Jensen's inequality yields equation 22. □

Theorem A.21 ($\mathcal{O}(1/\sqrt{RE})$ Convergence to a Stationary Point)**.** *Suppose Assumptions A.1–A.7 hold and F is possibly non-convex. Set $K = \mathcal{O}(R^{1/4})$ and $\eta = \sqrt{\frac{F^{(0)} - F^*}{L\sigma^2 RE}}$. After R communication rounds,*

$$\frac{1}{RE}\sum_{r=1}^{R}\sum_{t=0}^{E-1} \mathbb{E}\left\|\nabla F(P^{(r,t)})\right\|^2 \;\le\; \underbrace{4\sqrt{\frac{L(F^{(0)} - F^*)\sigma^2}{RE}}}_{\textit{optimisation term}} + \underbrace{\frac{2LG^2E\eta(1 - \lambda_p\mu_p\eta)^E}{1}}_{\textit{client drift residual}} + \underbrace{\frac{L(\lambda_m\delta_{\text{arch}}^2 + \lambda_p\zeta^2)}{2}}_{\textit{heterogeneity floor}}. \tag{23}$$

To reach an ε-stationary point it suffices to take $T = \mathcal{O}(1/\varepsilon^4)$ communication rounds.

Proof. **Step 1: Summing the descent lemma over E local steps.**

Applying Lemma A.18 at each step $t = 0, \ldots, E-1$ and summing (the F values telescope):

$$\frac{\eta}{2}\sum_{t=0}^{E-1} \left\|\nabla F(P^{(r,t)})\right\|^2 \;\le\; F(P^{(r,0)}) - \mathbb{E}[F(P^{(r,E)})] + \frac{LE\eta^2\sigma^2}{2} + \Delta^{(r)}, \tag{24}$$

where $\Delta^{(r)}$ captures the error from replacing the local gradient $\nabla \mathcal{L}_k$ with the global gradient ∇F. By Lemma A.19:

$$\Delta^{(r)} \;\le\; EL\eta\left(\sigma_g^2 + \frac{L(\lambda_m\delta_{\text{arch}}^2 + \lambda_p\zeta^2)}{2}\right).$$

Step 2: Summing over R rounds (telescoping).

Since $F(P^{(r+1,0)}) \le F(P^{(r,E)}) + L \cdot \mathbb{E}\left\|\bar{P}^{(r,E)} - P^{(r)}_{\text{global}}\right\|^2$ by L-smoothness and Lemma A.20, summing equation 24 over $r = 1, \ldots, R$:

$$\frac{\eta}{2}\sum_{r=1}^{R}\sum_{t=0}^{E-1}\left\|\nabla F(P^{(r,t)})\right\|^2 \;\le\; F^{(0)} - F^* + \frac{LRE\eta^2\sigma^2}{2} + RE\cdot L\eta\cdot\frac{\lambda_m\delta^2_{\text{arch}} + \lambda_p\zeta^2}{2} + R\cdot L^2\eta^2E^2G^2(1-\lambda_p\mu_p\eta)^E. \tag{25}$$

Step 3: Dividing by $\frac{\eta RE}{2}$.

Dividing both sides of equation 25 by $\frac{\eta RE}{2}$:

$$\frac{1}{RE}\sum_{r,t}\left\|\nabla F(P^{(r,t)})\right\|^2 \;\le\; \frac{2(F^{(0)} - F^*)}{\eta RE} + L\eta\sigma^2 + L(\lambda_m\delta^2_{\text{arch}} + \lambda_p\zeta^2) + \frac{2LG^2E\eta(1-\lambda_p\mu_p\eta)^E}{1}. \tag{26}$$

Step 4: Optimising the learning rate.

Setting the first two terms in equation 26 equal:

$$\frac{2(F^{(0)} - F^*)}{\eta^* RE} = L\eta^*\sigma^2 \;\;\Longrightarrow\;\; \eta^* = \sqrt{\frac{2(F^{(0)} - F^*)}{L\sigma^2 RE}}.$$

Their sum then equals

$$\frac{2(F^{(0)} - F^*)}{\eta^* RE} + L\eta^*\sigma^2 = 2\sqrt{\frac{2L\sigma^2(F^{(0)} - F^*)}{RE}} \le 4\sqrt{\frac{L(F^{(0)} - F^*)\sigma^2}{RE}},$$

which is the first (optimisation) term in equation 23. The heterogeneity floor $\frac{L(\lambda_m\delta^2_{\text{arch}} + \lambda_p\zeta^2)}{2}$ comes from the third term of equation 26 (the factor of 2 absorbed into the constant). The client drift residual is the fourth term of equation 26, unchanged. Combining the three contributions gives equation 23. □

Remark A.22 (Suppression of client drift by $\lambda_p\mathcal{L}_p$). When $\lambda_p > 0$ and $\mu_p > 0$, the drift term in equation 23 satisfies

$$G^2E\eta(1-\lambda_p\mu_p\eta)^E \;\xrightarrow{E\to\infty}\; 0$$

because the exponential factor $(1-\lambda_p\mu_p\eta)^E$ decays faster than the polynomial E grows. There exists a critical step count $E^* = \mathcal{O}\big(1/(\lambda_p\mu_p\eta)\big)$ such that for $E < E^*$ the drift is negligible. Without $\mathcal{L}_p$ ($\lambda_p = 0$), the drift grows as $\mathcal{O}(E^2\eta^2G^2)$, severely limiting the number of local steps that can be performed without hurting convergence.

B Efficiency Analysis

We quantify FedAura's resource efficiency along three dimensions: (1) **trainable parameters per client**, (2) **total upload volume to reach a target accuracy**, and (3) **total training FLOPs across all participating clients to reach a target accuracy**. All communication measurements consider upload only (client → server). Because PLAN requires all clients to share the same Transformer backbone and cannot be deployed under the heterogeneous backbone configuration of our main experiments, we organise the analysis into two parts: Section 1 compares FedAura with all baselines under the heterogeneous backbone setting, and Section 2 provides a dedicated comparison between FedAura and PLAN under an identical single-backbone setting.

B.1 Heterogeneous Backbone Setting

Trainable Parameters and Per-Round Upload. All methods below are evaluated under the same heterogeneous backbone configuration as in the main paper: each client runs one of eight architectures (ResNet-18/50, ViT-Small/Base, DeiT-Small/Base, CaiT-Tiny/Small), ranging from 6.4 M to 86.6 M parameters (25.6–346.4 MB), averaging 41.2 M (164.8 MB). Table. 4 shows the results.

Method-specific upload:

- FedMD: uploads soft logits on a shared public dataset (2,000 samples × 100 classes × 4 B ≈ 0.80 MB per round). No model weights are transmitted, but a public dataset is required.

- FedDF / Fed-DFA: uploads the *full* local model for server-side ensemble distillation (25.6–346.4 MB per round, 164.8 MB on average).
- HeteroFL / FedEcover: uploads a width-scaled sub-model (25.6–346.4 MB), averaging ∼82.4 MB (estimated at 50% of backbone size).
- FedAura: uploads a pixel-level visual prompt, learnable padding applied to the input image (s=30, 224×224):

$$2(224{+}2s)s{\times}3 + 2{\times}224s{\times}3 = 91{,}440 \text{ params} \approx 0.37 \text{ MB per round.}$$

  Because the prompt operates in the input image space, its dimension depends only on image resolution and is fully decoupled from backbone architecture, enabling aggregation across any mixture of CNN-based (ResNet-18/50) and Transformer-based models (ViT, DeiT, CaiT).

Table 4: Trainable parameters and upload per round. †FedMD uploads logits only; no model weights transmitted. ‡FedDF/Fed-DFA upload the full local model each round. §FedAura: pixel-level padding, s=30.

Method	Type	Train. Params	Upload / Round
FedMD	KD	6.4–86.6 M	0.80 MB†
FedDF	KD	6.4–86.6 M	164.8 MB‡
Fed-DFA	KD	6.4–86.6 M	164.8 MB‡
HeteroFL	PT	3.2–86.6 M	82.4 MB
FedEcover	PT	3.2–86.6 M	82.4 MB
FedAura	VP	**0.091 M**	**0.37 MB**§

FLOPs per Training Step. The average forward FLOPs across the eight backbone architectures is $\bar{\Phi}_{\text{fwd}}$=7.63 GFLOPs. For methods that train the full local model, $\Phi_{\text{step}} = 3\bar{\Phi}_{\text{fwd}} = 22.90$ GFLOPs (forward + backward, where backward ≈ 2× forward). For sub-model methods training ∼50% of the backbone, $\Phi_{\text{step}} = 2\bar{\Phi}_{\text{fwd}} = 15.27$ GFLOPs. FedAura freezes the backbone and places the visual prompt at the *input image level*; gradients reach the prompt directly from the first backbone layer without propagating through backbone weight matrices, so $\Phi_{\text{step}} \approx \bar{\Phi}_{\text{fwd}} = 7.63$ GFLOPs, the forward pass alone dominates. Table. 5 shows the results.

Table 5: FLOPs per gradient step. $\bar{\Phi}_{\text{fwd}}$=7.63 GFLOPs is the average forward FLOPs across the eight backbones.

Method	Forward FLOPs	Multiplier	FLOPs / Step
FedMD / FedDF / Fed-DFA	$\bar{\Phi}_{\text{fwd}} = 7.63$ G	3× (full model)	22.90 GFLOPs
HeteroFL / FedEcover	$\bar{\Phi}_{\text{fwd}} = 7.63$ G	2× (sub-model)	15.27 GFLOPs
FedAura	$\bar{\Phi}_{\text{fwd}} = 7.63$ G	**1× (frozen, input prompt)**	**7.63 GFLOPs**

Total Upload and FLOPs to Reach Target. Total FLOPs = $R \times m \times S \times \Phi_{\text{step}}$, where R is rounds to target, m=10 clients per round, S is gradient steps per client per round. Targets: 71.57% (CIFAR-10), 59.83% (CIFAR-100), 46.08% (Tiny-ImageNet), 51.19% (Flower-102), with batch size 32, 10 local epochs.

Table. 6 shows the results. FedAura achieves 3× lower FLOPs/step than full-model methods and 2× lower than sub-model methods by freezing the backbone and confining computation to the forward pass. Combined with 2.9×–4.1× faster convergence, FedAura reduces average total upload by 912× and average total FLOPs by 7.8× compared to KD-based methods. HeteroFL and FedEcover fail to reach CIFAR-100 and Tiny-ImageNet targets within 200 rounds, confirming that large per-round upload does not substitute for effective training.

B.2 FedAura vs. PLAN (ViT-B/16)

PLAN inserts VPT-style token embeddings into the internal feature space of a Vision Transformer, imposing two constraints that make it inapplicable to the heterogeneous setting: (1) **architectural homogeneity**, the prompt dimension

Table 6: Part 1: rounds to reach target accuracy, total upload per client (MB), and total FLOPs across m=10 clients. Targets: CIFAR-10 71.57%, CIFAR-100 59.83%, Tiny-ImageNet 46.08%, Flower-102 51.19%. *N/A*: target not reached within 200 rounds.

	CIFAR-10		**CIFAR-100**		**Tiny-ImageNet**		**Flower-102**	
Method	Upload	FLOPs	Upload	FLOPs	Upload	FLOPs	Upload	FLOPs
FedMD	57.6	1,649	78.4	2,244	71.2	2,038	62.4	1,786
FedDF	9,888	1,374	1,5326	2,130	13,513	1,878	1,1371	1,580
Fed-DFA	8,735	1,214	17,633	2,451	15,656	2,176	12,030	1,672
HeteroFL	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	5,026	932
FedEcover	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	4,779	885
FedAura	**6.7**	**132**	**15.2**	**313**	**22.2**	**297**	**10.8**	**221**

equals the backbone hidden dimension (d=768 for ViT-B/16), requiring all clients to use the *identical* model; and (2) **Transformer-only restriction**, token-level prompting requires a patch-embedding interface absent from CNN-based architectures such as ResNet. We therefore evaluate both methods under the same single-backbone setting (ViT-B/16) for a fair comparison.

FLOPs per Training Step. PLAN inserts n_{ctx}=16 learnable tokens into each of the 12 transformer layers, increasing the self-attention sequence length from 196 to 212 tokens per layer. Because the prompt parameters reside inside the network, a full backward pass through all 12 layers is required to compute their gradients:

$$\Phi_{\text{step}}^{\text{PFL}} = 3 \times \underbrace{12\,\mathcal{F}(212, 768, 3072)}_{\Phi_{\text{fwd}}^{\text{PFL}}=36.84\,\text{G}} = 110.52\,\text{GFLOPs}.$$

FedAura's prompt is applied at the input image level; gradients flow directly to the prompt without traversing backbone weight matrices, so only a single forward pass is needed:

$$\Phi_{\text{step}}^{\text{FA}} = 1 \times \underbrace{12\,\mathcal{F}(196, 768, 3072)}_{\Phi_{\text{fwd}}^{\text{FA}}=34.00\,\text{G}} = 34.00\,\text{GFLOPs}.$$

Here $\mathcal{F}(n, d, d_f)$ is the FLOPs of one transformer layer with sequence length n, hidden dimension d, and FFN width d_f. FedAura is $\mathbf{3.25}\times$ cheaper per step because its input-level design avoids the deep backward propagation mandated by PLAN's per-layer token embeddings. Table 7 shows the results.

Table 7: Part 2: FLOPs per gradient step on ViT-B/16.

Method	**Train. Params**	**Forward FLOPs**	**Multiplier**	**FLOPs / Step**
PLAN	0.147 M	$\Phi_{\text{fwd}}^{\text{PFL}} = 36.84\,\text{G}$	3× (deep backward)	110.52 GFLOPs
FedAura	**0.091 M**	$\Phi_{\text{fwd}}^{\text{FA}} = 34.00\,\text{G}$	**1× (input-level prompt)**	**34.00 GFLOPs**

Per-Round Upload Cost.

- PLAN: uploads a VPT-style visual prompt, learnable token embeddings inserted into every transformer layer of the ViT-B/16 backbone (n_{ctx}=16, depth= 12, hidden dim d=768):

  $$n_{\text{ctx}} \times d \times \text{depth} = 16 \times 768 \times 12 = 147{,}456 \text{ params} \approx 0.59\,\text{MB per round.}$$

  The prompt dimension is tightly coupled to the backbone hidden dimension d, so PLAN is restricted to a single fixed architecture (ViT-B/16) and cannot accommodate clients with different backbones.
- FedAura: uploads a pixel-level visual prompt, learnable padding appended to the input image (s=30, 224×224):

  $$2(224{+}2s)s \times 3 + 2 \times 224s \times 3 = 91{,}440 \text{ params} \approx 0.37\,\text{MB per round.}$$

Because the prompt operates in the input image space, its dimension depends only on image resolution (224×224) and is fully decoupled from the backbone architecture. This makes FedAura compatible with any vision model, CNN-based or Transformer-based, while PLAN is restricted to Transformer architectures with the same hidden dimension.

Total Upload and FLOPs to Reach Target. Under the same ViT-B/16 backbone, FedAura reduces FLOPs/round by $\mathbf{3.25}\times$ (34.00 vs. 110.52 GFLOPs) and upload/round by $\mathbf{1.59}\times$ (0.37 vs. 0.49 MB on average). Combined with $1.58\times$–$2.16\times$ faster convergence, FedAura reduces total FLOPs up to $\mathbf{7.1}\times$. FedAura's input-level prompt design extends support to heterogeneous and CNN-based architectures that PLAN structurally cannot accommodate. Table. 8 shows the results.

Table 8: Part 2: rounds to reach target accuracy, total upload per client (MB), and total FLOPs across m=10 clients on ViT-B/16. Targets: CIFAR-10 85.35%, CIFAR-100 73.72%, Tiny-ImageNet 59.26%, Flower-102 71.94%. PLAN upload: 0.59 MB/round. FedAura upload: 0.37 MB/round.

	CIFAR-10		**CIFAR-100**		**Tiny-ImageNet**		**Flower-102**	
Method	Upload	FLOPs	Upload	FLOPs	Upload	FLOPs	Upload	FLOPs
PLAN	24.8	4,641	30.7	5,747	39.5	7,404	43.1	8,067
FedAura	**9.2**	**816**	**12.6**	**1,156**	**11.5**	**1,054**	**15.6**	**1,428**

Accuracy Comparison with PLAN. To ensure a fair comparison, we evaluate both FedAura and PLAN under the same single-backbone setting (ViT-B/16), where all clients use an identical model architecture. Table 9 reports the top-1 accuracy of both methods across four datasets. FedAura consistently outperforms PLAN, demonstrating that input-level visual prompting with contrastive alignment is more effective than token-level prompting with aggregation under federated settings. Furthermore, unlike PLAN, FedAura supports heterogeneous and CNN-based backbone architectures, making it applicable to a broader range of real-world federated environments.

Table 9: Top-1 accuracy (%) comparison between FedAura and PLAN under the same ViT-B/16 backbone. All clients use identical model architectures and have IID data to ensure a fair comparison.

Method	**CIFAR-10**	**CIFAR-100**	**Tiny-ImageNet**	**Flower-102**
PLAN	85.35	73.72	59.26	71.94
FedAura	**89.13**	**76.25**	**63.89**	**75.03**

C Implementation Details

A key implementation detail concerns the classifier head attached to each client's frozen backbone. We clarify its design, initialisation, training procedure, and aggregation behaviour here to ensure reproducibility.

Architecture. Each client k attaches a single linear layer as the classifier head on top of the frozen backbone's output representation. Given the backbone output $z_k = f_k(\mathrm{Prompt}(x, P_k)) \in \mathbb{R}^{d_k}$, the classifier head is parameterised as $W_k \in \mathbb{R}^{C\times d_k}$, where C is the number of target classes and d_k is the output dimension of backbone f_k. The supervised loss in Eq. (12) is therefore:

$$\mathcal{L}_{\mathrm{sup}}(P_k, W_k;\, \mathcal{D}_k) = \mathbb{E}_{(x,y)\sim\mathcal{D}_k}\big[\,\ell_{\mathrm{CE}}(W_k\, z_k,\; y)\big],$$

where ℓ_{CE} denotes the standard cross-entropy loss.

Initialisation. The classifier head W_k is randomly initialised at the beginning of training. No pretrained weights, ImageNet classifier weights, or zero-shot embeddings outputs) are used.

Training. The visual prompt P_k and the classifier head W_k are optimised jointly during local training, while the backbone f_k remains completely frozen. The full local training objective is:

$$\min_{P_k,\, W_k}\; \mathcal{L}_{\mathrm{sup}}(P_k, W_k;\, \mathcal{D}_k) + \lambda_m \mathcal{L}_m(P_k) + \lambda_p \mathcal{L}_p(P_k).$$

The visual prompt P_k and the classifier head W_k are optimised jointly during local training via a single gradient step, while the backbone f_k remains completely frozen throughout.

Aggregation. Only the visual prompt P_k is uploaded to the server and aggregated across clients via FedAvg. The classifier head W_k is *not* aggregated. This is a necessary design choice under model heterogeneity: since d_k varies across backbone architectures (e.g., d=512 for ResNet-18, d=768 for ViT-Base), the classifier heads have different input dimensions and cannot be directly averaged. Each client therefore retains its own classifier head throughout all communication rounds, and only the shared visual prompt carries cross-client knowledge.

Label Space Compatibility. The randomly initialised linear head allows each client to independently adapt to its local label space (CIFAR-10, CIFAR-100, Tiny-ImageNet, Flower-102) without requiring alignment between client label spaces or with the pretraining label space of the backbone. This is consistent with standard practice in parameter-efficient transfer learning, such as linear probing on top of frozen pretrained representations.

D Limitations and Failure Regimes.

While FedAura demonstrates consistent improvements across the evaluated settings, we identify several potential failure regimes. First, when the degree of model heterogeneity is extremely large (e.g., mixing very shallow CNNs with deep ViTs), the model-contrastive loss may struggle to anchor all clients' prompted representations within a shared semantic structure, as the pretrained feature spaces diverge significantly across architectures. Second, under extreme data heterogeneity (e.g., $\beta \to 0$), prompt-contrastive alignment may be insufficient to prevent prompt drift when local data distributions are near-disjoint, as the global prompt becomes a poor anchor for all clients simultaneously. Third, FedAura inherits the general limitation of visual prompting: when the downstream task is substantially different from the pretraining distribution of the backbone, the frozen backbone may not provide useful representations regardless of prompt quality. These failure regimes suggest that FedAura is best suited to settings with moderate heterogeneity and reasonable alignment between pretraining and downstream task domains.