FedIBS: Federated Vision-Language Adaptation via Intrinsic Bias Selection

Xiaoming Wu[1,2,3], **Wei Wenyu**[1,2,3], **Xin Wang**[1,2,3,*], **Ming Yang**[1,2,3]
[1]Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China
[2]Shandong Provincial Key Laboratory of Industrial Network and Information System Security, Shandong Fundamental Research Center for Computer Science, Jinan, China
[3] Quan Cheng Laboratory, Jinan, China

Abstract

Federated adaptation of large vision-language models (VLMs) is appealing for privacy-sensitive applications but must contend with client data heterogeneity, strict communication budgets, and on-device inference constraints. Recent methods predominantly introduce additive adaptation interfaces, such as learnable prompts or adapter modules, which attach extra trainable components to the frozen backbone, inflating both parameter counts and inference latency. We challenge this additive paradigm and ask: can robust federated VLM adaptation be achieved through a compact intrinsic interface alone? We answer this question with **FedIBS** (Federated Intrinsic Bias Selection), a module-free framework that fine-tunes only the biases in feed-forward network projections. FedIBS operates on existing backbone parameters and introduces zero additional inference cost. To make such compact intrinsic adaptation viable under heterogeneous clients, we further propose a Fisher-guided progressive masking mechanism that disentangles shareable and personalized bias blocks. Each client retains personalized blocks locally and communicates only the shareable subset for aggregation, enabling sparse communication while preserving client-specific adaptation. Extensive experiments on diverse datasets and heterogeneity settings demonstrate that FedIBS consistently outperforms recent federated prompt- and adapter-based baselines in both personalization and generalization, all while achieving substantially lower communication cost.

1 Introduction

Adapting large vision-language models (VLMs) in federated settings is increasingly important for privacy-sensitive applications, yet it poses a multifaceted challenge. Foundation models such as CLIP offer highly transferable representations and strong zero-shot generalization Radford et al. [2021], making them attractive for learning from decentralized data that cannot be centrally aggregated due to privacy, ownership, or communication constraints. Such data security and privacy concerns are also prominent in AI-enabled smart manufacturing, where heterogeneous data are exchanged across interconnected systems Wang et al. [2025a]. However, the sheer scale of modern VLMs renders full-parameter fine-tuning impractical in federated environments where clients are often bandwidth- and resource-limited McMahan et al. [2017]. Federated VLM adaptation must therefore simultaneously satisfy three demands: adaptation effectiveness, communication efficiency, and deployment practicality.

Beyond efficiency, a more fundamental difficulty arises from heterogeneous client data, a hallmark of realistic non-IID federated settings Xu et al. [2022], Rieke et al. [2020]. Clients may correspond

*Corresponding author (Email: xinwang@qlu.edu.cn).

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

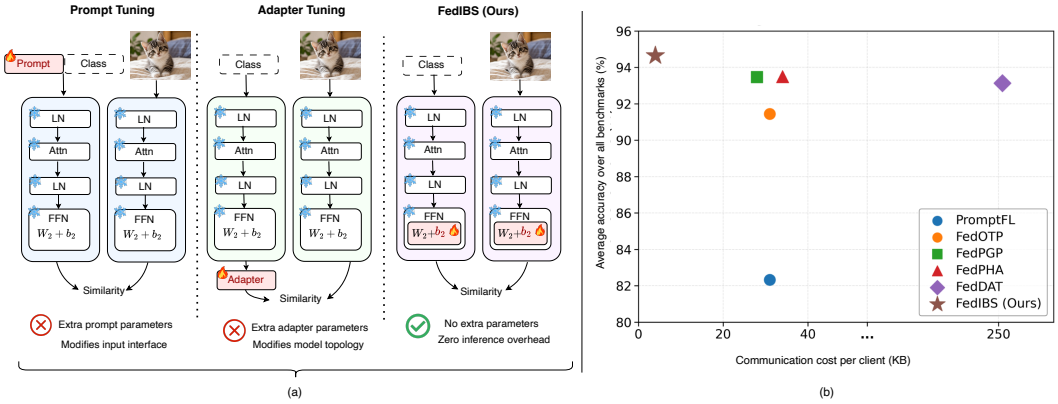

Figure 1: **Motivation and efficiency of FedIBS.** Panel (a) contrasts different adaptation interfaces: prompt tuning adds learnable prompt parameters and modifies the input interface, while adapter tuning inserts additional adapter modules and alters the model topology. In contrast, FedIBS directly tunes intrinsic FFN output biases inside the pretrained CLIP encoders, introducing no additional modules or inference-time overhead. Panel (b) compares average accuracy across all benchmarks and communication cost per client, showing a favorable accuracy–communication trade-off achieved by communicating only the Fisher-selected shareable bias blocks.

to different institutions, devices, acquisition conditions, or user populations, resulting in substantial appearance variation in local visual distributions. Crucially, this heterogeneity tends to manifest far more strongly in visual observations than in semantic label spaces: clients may share the same class vocabulary while exhibiting markedly different image distributions. Hence, effective federated VLM adaptation must be not only parameter-efficient, but also capable of absorbing client-specific shifts.

Most existing federated VLM adaptation methods follow an additive paradigm, introducing learnable prompts or adapter modules Bo et al. [2025], Fang et al. [2025], Cui et al. [2024], Chen et al. [2024]. In federated prompt tuning, trainable tokens are prepended to input embeddings; adapter-based methods insert lightweight modules inside the frozen backbone. While effective, these approaches alter the model interface or topology and typically incur additional inference-time computation, which is undesirable for edge deployment. More importantly, they adapt VLMs primarily through external interfaces, leaving the pretrained backbone itself largely unchanged. Our empirical analysis reveals a deeper mismatch: prompt tuning induces substantial drift in text features but almost no drift in the visual backbone, indicating that it essentially reshapes the text-side interface while leaving internal visual representations untouched. This mismatch is critical because heterogeneous clients usually differ far more in their local visual distributions than in class semantics. The key bottleneck is therefore not simply reducing the number of trainable parameters, but designing an adaptation interface that can directly and efficiently reshape internal visual representations under heterogeneous local domains. Figure 1 illustrates the interface-level distinction between additive adaptation and our method, as well as the resulting accuracy-communication trade-off.

Motivated by these observations, we ask: *can federated VLM adaptation be achieved through a substantially more compact intrinsic interface that incurs zero inference overhead while directly operating on existing backbone coordinates?* Moving from additive modules to intrinsic bias blocks, however, does not by itself solve the sharing problem in federated learning. Even within a compact intrinsic subspace, different clients may rely on different bias blocks to fit their local visual distributions. Compactness alone thus does not determine which updates should be globally shared. Blindly aggregating all intrinsic updates risks overwriting client-specific knowledge and degrading personalization. A robust intrinsic federated adaptation method must answer a second question: *which intrinsic bias blocks are genuinely shareable across clients, and which should remain personalized?*

To address both challenges, we propose **FedIBS** (Federated Intrinsic Bias Selection), a module-free framework that explores intrinsic federated adaptation through a minimal bias-only interface. FedIBS fine-tunes only the biases in the feed-forward network (FFN) projections of CLIP's encoders, directly operating on existing backbone coordinates without introducing new modules or extra inference latency. FedIBS locally adapts a compact FFN output-bias subspace, while Fisher-guided masking communicates only the shareable subset, reducing the communication cost to 4KB per client without

introducing additional inference-time modules. To make such compact intrinsic adaptation compatible with heterogeneous clients, FedIBS introduces a Fisher-guided progressive masking mechanism that disentangles personalized and shareable bias blocks. Each client retains personalized blocks locally and communicates only the shareable subset for aggregation, enabling sparse communication while preserving client-specific adaptability. In this way, FedIBS treats communication efficiency not as an isolated engineering objective, but as a natural consequence of selecting shareable bias blocks within a compact internal adaptation space.

Our contributions are summarized as follows:

- We rethink federated VLM adaptation from the interface-design perspective and propose FedIBS, a module-free intrinsic framework that fine-tunes only FFN projection biases, achieving zero inference overhead and extreme parameter efficiency.
- We introduce Fisher-guided progressive masking to partition intrinsic bias blocks into shareable and personalized subsets, enabling sparse communication while preserving local adaptation under client heterogeneity.
- Through extensive experiments across single-domain, multi-domain, and standard non-IID benchmarks, we demonstrate that FedIBS consistently outperforms recent prompt- and adapter-based baselines in both accuracy and communication efficiency.

2 Related Work

2.1 Parameter-Efficient Adaptation of Vision–Language Models

Pretrained vision–language models such as CLIP Radford et al. [2021] are commonly adapted via parameter-efficient fine-tuning (PEFT), which updates only a small fraction of parameters. Existing PEFT methods for VLMs broadly fall into additive and intrinsic paradigms. Additive methods introduce extra trainable components, such as learnable prompts Zhou et al. [2022] or adapter modules Zhang et al. [2022], Gao et al. [2024], while keeping most backbone parameters frozen. In contrast, intrinsic approaches update existing model parameters without inserting new modules. Representative works like BitFit Zaken et al. [2022] and ClipFit Li et al. [2024b] demonstrate that tuning only existing biases or normalization-related parameters can be highly effective while preserving the original architecture. However, these methods are mainly designed for centralized adaptation and do not address which intrinsic bias blocks should be shared or kept client-specific under federated heterogeneity. FedIBS extends intrinsic PEFT to federated VLM adaptation by coupling bias-only tuning with a principled mechanism for client-aware block selection.

2.2 Federated Adaptation of Vision–Language Models

Federated learning enables decentralized VLM adaptation without sharing raw data, but the sheer size of modern VLMs makes direct tuning communication-intensive. Consequently, recent federated VLM methods overwhelmingly resort to parameter-efficient strategies. Prompt-based approaches jointly optimize learnable tokens across clients and employ grouping, decomposition, alignment, or mixture-based designs to handle heterogeneity Guo et al. [2023], Bo et al. [2025], Fang et al. [2025], Cui et al. [2024], Li et al. [2024a], Singha et al. [2025], Wang et al. [2025b]. Adapter-based methods insert trainable modules into the backbone and often pair them with distillation or regularization to stabilize heterogeneous optimization Chen et al. [2024], Wang et al. [2023]. Despite their effectiveness, these works focus on aggregating and personalizing the newly introduced prompts or adapters; they do not investigate whether a compact set of pre-existing backbone coordinates can itself serve as the primary federated adaptation interface. This gap directly motivates FedIBS: a module-free intrinsic adaptation approach specifically for heterogeneous VLM clients.

2.3 Sparse Communication and Personalized Federated Optimization

FedIBS also connects to sparse communication and personalized federated learning, which aim to reduce transmission costs and mitigate heterogeneity by selectively sharing parameters or decomposing models into shared and local components Guastella et al. [2025], Dinh et al. [2020], Collins et al. [2021], Shamsian et al. [2021]. Related ideas also appear in sparse training and parameter selection, such as the Lottery Ticket Hypothesis Frankle and Carbin [2018] and gradient-based parameter

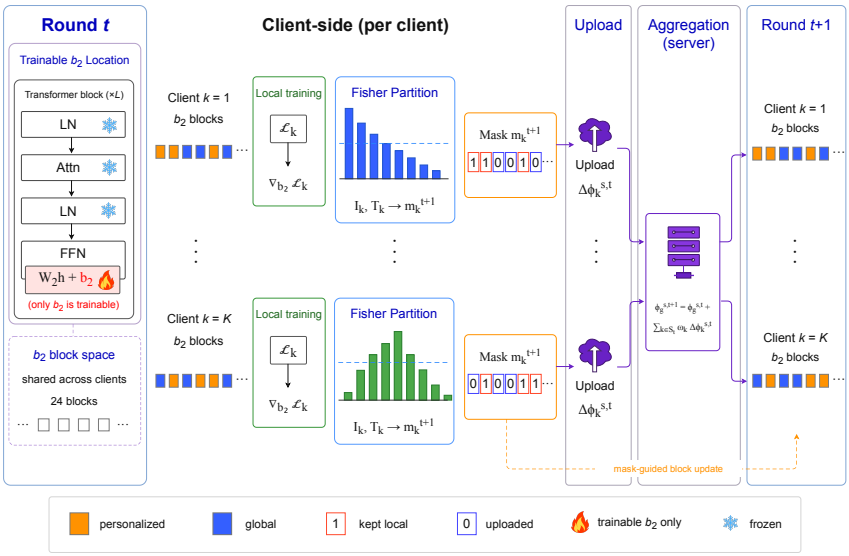

Figure 2: **FedIBS Overview.** FedIBS employs Fisher-guided masking to partition FFN output-bias blocks into shareable and personalized subsets; the server aggregates only the shareable updates.

selection Zhang et al. [2024]. Different from these works, FedIBS studies sparse and personalized optimization inside the compact intrinsic bias subspace of a pretrained VLM. The central question is not just which parameters are important, but which intrinsic bias blocks should be globally shared versus locally retained so that heterogeneous visual domains can be effectively accommodated.

3 Methodology

We propose **FedIBS**, a module-free federated adaptation framework for vision-language models. FedIBS rests on two coupled design principles. First, we confine adaptation to a highly compact intrinsic subspace formed by the FFN output biases in CLIP's encoders, yielding a mergeable interface with zero additional inference cost. Second, to make such compact intrinsic adaptation compatible with heterogeneous clients, we partition the intrinsic bias blocks into *shareable* and *personalized* subsets via Fisher-guided progressive masking. Shareable blocks are regularized toward the global consensus and uploaded for aggregation, while personalized blocks remain local to preserve client-specific adaptation. Figure 2 illustrates the overall pipeline.

3.1 Problem Definition

CLIP with intrinsic bias adaptation. We consider a pretrained vision-language model composed of an image encoder $f_v(\cdot;\theta_v,\phi_v)$ and a text encoder $f_t(\cdot;\theta_t,\phi_t)$. Here $\theta=\{\theta_v,\theta_t\}$ denotes the frozen backbone parameters and $\phi=\{\phi_v,\phi_t\}$ denotes the trainable intrinsic parameters injected into a small subset of FFN projection biases. For an image x and class prompts $\{t_c\}_{c=1}^{C}$, the probability of class y is given by temperature-scaled cosine similarity:

$$p(y \mid x;\theta,\phi)=\frac{\exp\big(\mathrm{sim}(f_v(x;\theta_v,\phi_v),f_t(t_y;\theta_t,\phi_t))/\tau\big)}{\sum_{j=1}^{C}\exp\big(\mathrm{sim}(f_v(x;\theta_v,\phi_v),f_t(t_j;\theta_t,\phi_t))/\tau\big)}, \tag{1}$$

where $\mathrm{sim}(\cdot,\cdot)$ denotes cosine similarity and τ is the temperature. For supervised adaptation, we minimize the cross-entropy loss:

$$\ell_{\mathrm{CE}}(\theta,\phi;\,x,y)=-\log p(y \mid x;\theta,\phi). \tag{2}$$

FedIBS freezes θ and optimizes only a compact intrinsic parameter set ϕ, instantiated as FFN output biases (Sec. 3.2), which preserves the original architecture.

Federated setting. We consider K clients, each of which holds a private dataset $\mathcal{D}_k$ with $n_k=|\mathcal{D}_k|$ samples; the total number of samples is $N=\sum_{k=1}^{K}n_k$. At communication round t, the server selects

a subset of active clients $\mathcal{S}_t$, broadcasts the current global intrinsic parameters $\phi^{(t)}$, and aggregates the uploaded updates. The federated objective over ϕ is

$$\min_{\phi} \mathcal{L}(\phi) = \sum_{k=1}^{K} \frac{n_k}{N} \mathbb{E}_{(x,y)\sim\mathcal{D}_k} \big[\ell_{\mathrm{CE}}(\theta, \phi; x, y)\big]. \tag{3}$$

At each communication round, selected clients initialize shareable bias blocks from the current global model while preserving their locally retained personalized blocks. Each client then performs E local epochs to obtain its updated intrinsic parameters. FedIBS maintains a progressive blockwise partition of the intrinsic parameters and uploads only the updates of shareable bias blocks (Sec. 3.3). Let $\omega_k = \frac{n_k}{\sum_{\ell\in\mathcal{S}_t} n_\ell}$ denote the sample-size weight over active clients. The server updates the global intrinsic parameters as

$$\phi^{(t+1)} = \phi^{(t)} + \sum_{k\in\mathcal{S}_t} \omega_k \, \Delta\phi_k^{(t)}, \tag{4}$$

where $\Delta\phi_k^{(t)}$ denotes the masked shareable update uploaded by client k.

The above formulation specifies the federated adaptation objective over ϕ. We next describe how FedIBS chooses this intrinsic subspace and why FFN output biases provide a suitable adaptation interface under heterogeneous federated optimization.

3.2 Intrinsic Bias Subspace Identification

A central question in federated adaptation is how to select a trainable subspace that is expressive enough for downstream alignment while remaining stable under heterogeneous data. FedIBS restricts the tuning interface to the FFN output projection biases. Specifically, each transformer block contains an FFN:

$$\mathrm{FFN}(h) = W_2 \, \sigma(W_1 h + b_1) + b_2, \tag{5}$$

where W_1, W_2 are weight matrices, b_1, b_2 are bias vectors, and $\sigma(\cdot)$ denotes the nonlinearity. FedIBS freezes all backbone parameters and unlocks only the output projection bias b_2 across FFN blocks in both image and text encoders:

$$\phi \triangleq \{b_2 \mid b_2 \in \mathrm{Bias}(\mathrm{FFN}_{\mathrm{proj}})\}, \qquad |\phi| \ll |\theta|. \tag{6}$$

This choice yields a compact intrinsic adaptation interface with three advantages. First, tuning b_2 preserves the original architecture and incurs zero additional inference cost, since the updated biases are directly merged into the pretrained backbone. Second, b_2 shifts the FFN output before the residual connection, enabling lightweight modulation of intermediate representations. Third, compared with higher-dimensional weight tuning, the bias-only interface reduces client-update variance and serves as an implicit structural regularizer under heterogeneous federated optimization. From the perspective of domain shift, client-specific visual differences often manifest as systematic feature offsets; adjusting b_2 provides a lightweight way to compensate for such shifts while minimally perturbing pretrained semantics.

This compact subspace improves efficiency and stability, but does not determine which intrinsic bias blocks should be globally shared. We next introduce a Fisher-guided partition mechanism to distinguish shareable and personalized bias blocks under heterogeneous local domains.

3.3 Fisher-Guided Shareable and Personalized Partition

Under non-IID data, directly aggregating locally sensitive parameters may interfere with client-specific adaptation. FedIBS therefore partitions the intrinsic bias space into shareable and personalized blocks using Fisher-guided progressive masking. Specifically, CLIP ViT-B/16 contains 12 visual and 12 textual FFN output-projection bias vectors, with 768 and 512 scalars per vector, respectively. These 24 bias blocks, totaling 15,360 trainable scalars, serve as the basic units for Fisher-guided selection and parameter sharing.

Blockwise Fisher Estimation. For client k at round t, let $\mathbf{g}_{k,s}^{(t)}$ denote the cross-entropy gradient of the intrinsic parameters computed on local mini-batch s, with T_k mini-batches used for Fisher

accumulation. The sensitivity of bias block B_j is estimated as

$$I_{k,j}^{(t)} = \frac{1}{T_k} \sum_{s=1}^{T_k} \left\| \mathbf{g}_{k,s,B_j}^{(t)} \right\|_2^2 = \sum_{a \in B_j} I_{k,a}^{(t)}, \tag{7}$$

where $I_{k,a}^{(t)}$ denotes the accumulated squared gradient of scalar a within block B_j. Larger blockwise Fisher scores indicate greater local loss sensitivity, providing a criterion for identifying bias blocks that may benefit from personalized retention. Fisher scores are accumulated during local training and used to update the mask for the following communication round.

Progressive personalization mask. Each client maintains a binary block mask $\mathbf{m}_k^{(t)} \in \{0,1\}^{24}$, where $m_{k,j}^{(t)} = 1$ assigns the entire block B_j to the personalized subset $\mathcal{V}_k$, while $m_{k,j}^{(t)} = 0$ assigns it to the shareable subset $\mathcal{U}_k$. Starting from $\mathbf{m}_k^{(0)} = \mathbf{0}$, the mask is progressively updated as

$$m_{k,j}^{(t+1)} = m_{k,j}^{(t)} \vee \mathbb{I}\left(I_{k,j}^{(t)} > \tau_k^{(t)} \right), \tag{8}$$

subject to $\|\mathbf{m}_k^{(t+1)}\|_0 \leq \lceil 24\alpha \rceil$. Here $\tau_k^{(t)}$ is determined by Otsu thresholding over the blockwise Fisher scores, and newly selected blocks are limited by the personalization budget. The mask is cumulative, so previously personalized blocks remain local. For the default $\alpha = 0.8$, at most 20 of the 24 blocks are personalized.

Local optimization with partition. Given $\mathbf{m}_k^{(t)}$, personalized blocks are adapted locally, while shareable blocks are regularized toward the global model through a Fisher-weighted proximal term:

$$\mathcal{R}_k^{(t)} = \sum_{j=1}^{24} \left(1 - m_{k,j}^{(t)}\right) \left(1 + \widetilde{I}_{k,j}^{(t)}\right) \left\| \phi_{k,B_j} - \phi_{B_j}^{(t)} \right\|_2^2, \tag{9}$$

where $\widetilde{I}_{k,j}^{(t)}$ is the min-max normalized blockwise Fisher score. The overall local objective is

$$\mathcal{L}_k^{(t)} = \mathcal{L}_{\mathrm{CE}} + \lambda_{\mathrm{prox}} \mathcal{R}_k^{(t)} + \lambda_{\mathrm{kd}} \mathcal{L}_{\mathrm{KD}}, \tag{10}$$

with the CLIP-style feature distillation loss

$$\mathcal{L}_{\mathrm{KD}} = \big(1 - \cos(\mathbf{s}_I, \mathbf{t}_I)\big) + \big(1 - \cos(\mathbf{s}_T, \mathbf{t}_T)\big), \tag{11}$$

where $\mathbf{s}_I, \mathbf{s}_T$ and $\mathbf{t}_I, \mathbf{t}_T$ denote student and frozen-teacher image/text features, respectively.

Sparse communication and local retention. Let $\widetilde{\mathbf{m}}_k^{(t)} \in \{0,1\}^{15360}$ denote the scalar-level expansion of the block mask. After local training, client k uploads only the shareable bias updates:

$$\Delta\phi_k^{(t)} = \left(\mathbf{1} - \widetilde{\mathbf{m}}_k^{(t)}\right) \odot \left(\phi_k^{(t,E)} - \phi^{(t)}\right). \tag{12}$$

The server aggregates these masked updates according to Eq. (4). Personalized block values and masks are retained by each client across communication rounds, while only shareable blocks are updated from the server broadcast. The fixed block structure also enables sparse transmission using a compact 24-bit mask without individual scalar indices.

Overall, FedIBS combines compact intrinsic bias adaptation with Fisher-guided blockwise personalization, allowing locally sensitive parameters to be preserved while selectively aggregating shareable knowledge.

3.4 Theoretical Justification

We provide theoretical justification for the two main design choices of FedIBS: compact FFN output-bias adaptation and Fisher-guided personalization. The first result compares dimension-dependent generalization upper bounds for different adaptation spaces, while the second provides a local-quadratic interpretation of Fisher-guided personalization. Detailed proofs are provided in Appendix A.

Bias-only intrinsic tuning reduces adaptation complexity. For any mixture coefficient $\lambda \in \Delta_K$, define the mixed population and empirical distributions as

$$D_\lambda = \sum_{k=1}^{K} \lambda_k D_k, \qquad \widehat{D}_\lambda = \sum_{k=1}^{K} \lambda_k \widehat{D}_k.$$

For a predictor h, we define the worst-case federated generalization gap as

$$\varepsilon(h) \triangleq \sup_{\lambda \in \Delta_K} \left| L_{D_\lambda}(h) - L_{\widehat{D}_\lambda}(h) \right|. \tag{13}$$

Theorem 3.1 (Dimension-dependent generalization bounds)**.** *Let $\mathcal{H}_b$ denote the hypothesis class obtained by tuning only the FFN output bias $b_2 \in \mathbb{R}^d$, and let $\mathcal{H}_{\mathrm{FFN}}$ denote the class obtained by tuning the full FFN output projection (W_2, b_2), where $W_2 \in \mathbb{R}^{d \times r}$.*

Assume that each client has at least $n \geq 2$ independent and identically distributed training samples, the loss is bounded and uniformly Lipschitz with respect to the tunable parameters, and the parameter spaces are contained in bounded Euclidean balls. For a fixed number of clients K, with probability at least $1 - \delta$, the following bounds hold simultaneously:

$$\begin{aligned} \sup_{h \in \mathcal{H}_b} \varepsilon(h) &\leq C \sqrt{\frac{d \log n + \log(1/\delta)}{n}}, \\ \sup_{h \in \mathcal{H}_{\mathrm{FFN}}} \varepsilon(h) &\leq C \sqrt{\frac{(dr + d) \log n + \log(1/\delta)}{n}}, \end{aligned} \tag{14}$$

where C depends on the boundedness, Lipschitz, norm, and fixed-client assumptions, but not on the adaptation dimension or sample size.

Theorem 3.1 indicates that bias-only tuning admits a smaller dimension-dependent generalization upper bound than full FFN output-projection tuning. The leading dimensional terms scale as $\sqrt{d}$ and $\sqrt{dr + d}$, respectively, differing by a factor of $\sqrt{r + 1}$. This provides a complexity-based motivation for compact intrinsic adaptation, without implying a corresponding ratio of realized generalization gaps.

Fisher-guided personalization. We next analyze the local loss incurred when client-adapted bias parameters are synchronized with the global model.

Lemma 3.2 (Local synchronization sensitivity)**.** *Let $L_k(\phi)$ denote the local objective of client k. Suppose that around a local optimum $\phi_k^\star$, the objective admits a second-order approximation with Hessian H_k. For an FFN output-bias block B_j, assume that the corresponding Hessian submatrix can be locally approximated by the diagonal empirical Fisher:*

$$H_{k,B_j B_j} \approx \mathrm{diag}\left(\{I_{k,a}\}_{a \in B_j}\right).$$

Then, for a sufficiently small perturbation that synchronizes the complete block with the global model, the local loss increase is approximately

$$\Delta L_{k,B_j} \approx \frac{1}{2} \sum_{a \in B_j} I_{k,a} \left(\phi_a^{(t)} - \phi_{k,a}^\star\right)^2. \tag{15}$$

Lemma 3.2 provides a local geometric justification for Fisher-guided personalization. Under the local quadratic approximation, the Fisher-weighted deviation between local and global bias parameters characterizes the potential loss incurred by global synchronization. This motivates retaining locally sensitive bias blocks as personalized parameters while aggregating the remaining shareable blocks.

4 Experiments

4.1 Experimental Setup

Datasets. Following prior federated VLM adaptation Guo et al. [2023], Fang et al. [2025], we evaluate FedIBS on three groups of benchmarks. For single-domain CLIP transfer, we use Caltech101 Fei-Fei [2004], Flowers102 Nilsback and Zisserman [2008], UCF101 Soomro et al. [2012],

Table 1: **Comparison with state-of-the-art (SOTA) methods on single-domain datasets across 10 clients.** All methods are evaluated under the same 16-shot setting.

Methods	Caltech101	Flowers102	UCF101	StanfordCars	SUN397
CLIP	92.02 ± 0.23	67.79 ± 0.18	65.06 ± 0.41	65.60 ± 0.36	62.58 ± 0.29
PromptFL	93.22 ± 0.47	88.75 ± 0.68	73.43 ± 1.12	67.31 ± 0.83	68.91 ± 0.61
FedOTP	98.48 ± 0.12	99.42 ± 0.09	90.28 ± 0.54	91.73 ± 0.37	91.89 ± 0.28
FedPGP	98.93 ± 0.31	99.53 ± 0.16	95.68 ± 0.48	97.06 ± 0.26	92.89 ± 0.43
FedPHA	99.04 ± 0.11	99.44 ± 0.08	95.66 ± 0.34	97.21 ± 0.19	93.64 ± 0.27
FedDAT	98.83 ± 0.24	99.43 ± 0.13	95.63 ± 0.39	96.12 ± 0.32	93.57 ± 0.36
FedIBS (Ours)	$\mathbf{99.14 \pm 0.14}$	$\mathbf{99.60 \pm 0.09}$	$\mathbf{97.03 \pm 0.28}$	$\mathbf{97.51 \pm 0.23}$	$\mathbf{94.67 \pm 0.31}$

Table 2: **Trainable parameters and per-client communication cost.** FedIBS transmits only Fisher-selected shareable bias blocks.

Methods	Trainable (K)	Comm. (KB)
PromptFL	16.38	31
FedOTP	16.38	31
FedPGP	16.38	31
FedPHA	16.38	31
FedDAT	132.22	252
FedIBS	15.36	**4**

Table 3: **SOTA comparison on CIFAR-10 and CIFAR-100 across 100 clients.**

Methods	CIFAR-10	CIFAR-100
CLIP	87.87	63.02
PromptFL	92.64	68.41
FedOTP	93.87	77.17
FedPGP	94.31	77.14
FedPHA	94.71	75.44
FedDAT	93.71	76.57
FedIBS	**95.51**	**81.10**

StanfordCars Krause et al. [2013], and SUN397 Xiao et al. [2010], each under a pathological non-IID class partition. For multi-domain shift, we use Office31 Saenko et al. [2010], Office-Home Venkateswara et al. [2017], and DomainNet Peng et al. [2019], treating each domain as a separate client. In particular, we use 10 selected classes from DomainNet for training to construct a tractable multi-domain federated evaluation. For standard federated learning benchmarks, we leverage CIFAR-10 and CIFAR-100 Krizhevsky et al. [2010] with a Dirichlet partition ($\beta = 0.5$) across 100 clients.

Baselines. We compare FedIBS against zero-shot CLIP Radford et al. [2021], the prompt-based federated VLM methods PromptFL Guo et al. [2024], FedOTP Li et al. [2024a], FedPGP Cui et al. [2024], and FedPHA Fang et al. [2025], as well as the adapter-based method FedDAT Chen et al. [2024]. All baselines are evaluated under the same backbone, data partitions, and federated protocol.

Implementation details. All methods employ frozen CLIP encoders with ViT-B/16 Dosovitskiy [2020] as the default backbone. We set the number of local epochs to $E = 3$ and communication rounds to $R = 50$, except on CIFAR-10/100 where $R = 25$. The batch size is 32 for training and 128 for testing. Unless otherwise stated, we use $\alpha = 0.8$, $\lambda_{\text{kd}} = 1.0$, and $\lambda_{\text{prox}} = 0.1$. During inference, each client uses its personalized model, combining the globally aggregated shareable bias blocks with locally retained personalized blocks. We report the mean test accuracy across clients and communication rounds. All experiments are conducted on a single NVIDIA A100 GPU with 40GB memory.

4.2 Performance Comparison

Single-domain CLIP transfer under non-IID data. We first compare FedIBS with representative federated prompt- and adapter-tuning baselines on single-domain CLIP transfer benchmarks under the 16-shot setting. Table 1 reports mean accuracy and standard deviation over three seeds on Caltech101, Flowers102, UCF101, StanfordCars, and SUN397. FedIBS achieves the best performance on all datasets, with particularly clear gains on UCF101 and SUN397, showing that intrinsic bias-only tuning outperforms additive prompt or adapter designs without modifying the backbone. Table 2 further compares parameter efficiency. FedIBS trains only the compact FFN output-bias subspace locally, but communicates exclusively the Fisher-selected shareable blocks, reducing the communicated parameters to 2.05K per client and the communication cost to 4KB per client.

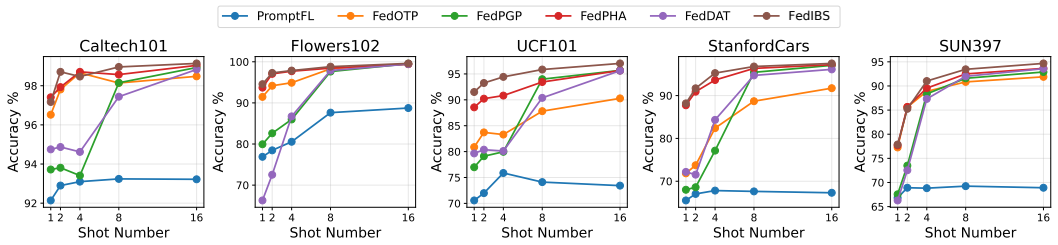

Figure 3: **Impact of the number of shots.** Accuracy comparison under varying numbers of shots on single-domain CLIP transfer datasets.

Table 4: **Comparison with SOTA methods on Office31, OfficeHome, and DomainNet.** Each domain is treated as an individual client. DomainNet is evaluated on 10 selected classes.

Methods	**Office31**				**OfficeHome**					**DomainNet**						
	A	**W**	**D**	**AVG**	**A**	**C**	**P**	**R**	**AVG**	**C**	**I**	**P**	**Q**	**R**	**S**	**AVG**
CLIP	81.14	72.45	73.42	75.67	84.5	66.05	89.51	89.77	82.46	96.32	75.1	96.31	62.53	97.61	97.43	87.55
PromptFL	89.35	90.81	94.3	91.49	86.11	77.11	94.79	93.01	87.76	96.85	80.78	96.18	79.89	97.88	96.65	91.37
FedOTP	86.26	97.75	97.46	93.82	80.7	77.72	92.94	88.41	84.94	97.41	80.63	96.76	87.00	97.90	97.09	92.79
FedPGP	89.71	98.57	99.11	95.79	88.22	78.94	96.05	**94.29**	89.37	97.58	85.04	97.26	87.34	98.42	98.31	93.99
FedPHA	89.35	**99.79**	99.36	96.16	87.1	80.02	**96.18**	93.24	89.13	97.69	85.15	98.02	89.1	**98.62**	98.36	94.49
FedDAT	89.07	98.36	97.59	95.00	87.97	78.05	95.21	93.35	88.64	97.46	84.82	97.52	86.70	98.59	98.16	93.87
FedIBS (Ours)	**89.92**	99.59	**100**	**96.50**	**88.42**	**83.3**	96.05	93.01	**90.20**	**98.02**	**85.37**	**98.48**	**92.72**	98.46	**98.53**	**95.26**

Impact of the number of shots. We further investigate the effect of varying the number of shots per class from 1 to 16 on the single-domain CLIP transfer datasets. As shown in Figure 3, FedIBS consistently delivers strong performance across all shot settings, with the most pronounced advantage in the extreme few-shot regime. The gaps narrow as more local supervision becomes available, while FedIBS remains competitive throughout.

Robustness to label distribution shift. We further examine robustness under label distribution shift using CIFAR-10 and CIFAR-100, partitioned across 100 clients with a Dirichlet distribution ($\beta = 0.5$). As reported in Table 3, FedIBS attains the highest accuracy on both datasets, demonstrating that the proposed bias-only interface remains effective under severe label heterogeneity and a larger federation scale.

Cross-domain robustness. We next evaluate cross-domain robustness on Office31, OfficeHome, and DomainNet, where each domain is treated as an individual client. Table 4 reports both per-domain and average accuracies. FedIBS achieves the highest average accuracy on all three benchmarks, confirming its strong robustness under feature-distribution heterogeneity.

4.3 Ablation Studies and Analysis

Intrinsic adaptation space. Table 5 compares several candidate intrinsic parameterizations before communication masking. Tuning all biases achieves the highest average accuracy, but at the cost of 171,008 trainable parameters. In contrast, restricting tuning to FFN output biases (b_2 only) yields a competitive accuracy of 79.48% with merely 15,360 parameters, while one-sided image- or text-only projection bias variants perform substantially worse. Although tuning both b_1 and b_2 (denoted FFN bias tuning) provides a slight improvement over tuning b_2 alone, the gain is marginal compared to the roughly five-fold increase in trainable parameters. We therefore adopt FFN output bias as the intrinsic adaptation space of FedIBS, striking a favorable accuracy-efficiency trade-off.

Blockwise partition. Table 6 evaluates the effect of the shared/personalized partition on UCF101. The All-Shared baseline, which aggregates all intrinsic bias blocks, reaches only 72.65% accuracy, confirming that naïvely synchronizing every block can severely degrade client-specific adaptation. RandomMask improves the result to 94.29%, indicating that separating shared and personalized parameters is beneficial, even without a principled selection criterion. FedIBS further raises the accuracy to 97.03%, demonstrating that its advantage stems not from sparsity alone, but from

Table 5: **Intrinsic parameterization ablation.** Comparison of local trainable bias subsets before communication masking.

Intrinsic Space	Params	Avg.
All Bias Terms	171,008	80.50
FFN Output Bias	15,360	79.48
FFN Bias	76,800	79.54
Image Proj. Bias	9,216	76.70
Text Proj. Bias	6,144	76.62

Table 6: **Blockwise partition ablation.** Accuracy comparison of shareable–personalized partition strategies.

Partition	UCF101
All-Shared	72.65
RandomMask	94.29
FedIBS	97.03

the Fisher-guided identification of which bias blocks should be shared and which should remain personalized.

5 Conclusion

We presented FedIBS, a module-free framework for federated VLM adaptation that fine-tunes only the FFN output biases of pretrained CLIP encoders. Rather than introducing external prompts or adapters, FedIBS operates on a compact intrinsic parameter space and adds zero inference latency. To make this intrinsic interface robust under client heterogeneity, FedIBS employs Fisher-guided progressive masking to disentangle shareable and personalized bias blocks, enabling sparse communication while preserving local adaptation. Experiments across diverse federated VLM benchmarks demonstrate that FedIBS consistently improves accuracy over recent prompt- and adapter-based baselines while substantially reducing communication cost. These results indicate that intrinsic bias-block selection provides a promising direction for efficient and deployment-friendly federated adaptation of large VLMs.

Limitations

Our experiments primarily focus on CLIP-based adaptation under largely static federated settings. Although FedIBS substantially reduces communication costs, Fisher estimation introduces additional client-side computation. Extending FedIBS to broader VLM architectures and dynamic client populations remains an important direction for future work.

Acknowledgements and Disclosures

Funding. This work was supported by the Shandong Provincial Key Research and Development Program under Grant 2024CXGC010113, the NSFC under Grant 62402256, the Shandong Provincial Natural Science Foundation of China under Grant ZR2024MF100, the Taishan Scholars Program under Grants tsqn202408239 and tsqn202312231, the Pilot Project for the Integration of Science, Education and Industry of Qilu University of Technology (Shandong Academy of Sciences) under Grant 2026ZDCX02, and the Research Project of Quancheng Laboratory, China under Grants QCL20250201 and QCL20250302.

Competing Interests. The authors declare no competing interests.

References

Weihao Bo, Yanpeng Sun, Yu Wang, Xinyu Zhang, and Zechao Li. Fedmgp: Personalized federated learning with multi-group text-visual prompts. *arXiv preprint arXiv:2511.00480*, 2025.

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

Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *ICML*, pages 2089–2099. PMLR, 2021.

Tianyu Cui, Hongxia Li, Jingya Wang, and Ye Shi. Harmonizing generalization and personalization in federated prompt learning. *arXiv preprint arXiv:2405.09771*, 2024.

Canh T. Dinh, Nguyen H. Tran, and Tuan Dung Nguyen. Personalized federated learning with moreau envelopes. In *Advances in Neural Information Processing Systems*, volume 33, pages 21394–21405, 2020.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Chengying Fang, Wenke Huang, Guancheng Wan, Yihao Yang, and Mang Ye. Fedpha: Federated prompt learning for heterogeneous client adaptation. In *Forty-second International Conference on Machine Learning*, 2025.

Li Fei-Fei. Learning generative visual models from few training examples. In *Workshop on Generative-Model Based Vision, IEEE Proc. CVPR, 2004*, 2004.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. CLIP-adapter: Better vision-language models with feature adapters. *IJCV*, 132(2): 581–595, 2024.

Adriano Guastella, Lorenzo Sani, Alex Iacob, Alessio Mora, Paolo Bellavista, and Nicholas D. Lane. Sparsyfed: Sparse adaptive federated training. In *International Conference on Learning Representations (ICLR)*, 2025.

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

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, 2024. doi: 10.1109/TMC.2023.3302410.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013.

Alex Krizhevsky, Geoff Hinton, et al. Convolutional deep belief networks on cifar-10. *Unpublished manuscript*, 40(7):1–9, 2010.

H. Li, W. Huang, J. Wang, and Y. Shi. Global and local prompts cooperation via optimal transport for federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12151–12161, 2024a.

Ming Li, Jike Zhong, Chenxin Li, Liuzhuozheng Li, Nie Lin, and Masashi Sugiyama. Vision-language model fine-tuning via simple parameter-efficient modification. *arXiv preprint arXiv:2409.16718*, 2024b.

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, pages 1273–1282. PMLR, 2017.

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

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021.

Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletarì, Holger R. Roth, et al. The future of digital health with federated learning. *npj Digital Medicine*, 3:119, 2020. doi: 10.1038/s41746-020-00323-1.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.

Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. pfedhn: Personalized federated learning using hypernetworks. In *International Conference on Machine Learning (ICML)*, pages 9494–9504. PMLR, 2021.

Mainak Singha, Subhankar Roy, Sarthak Mehrotra, Ankit Jha, Moloud Abdar, Biplab Banerjee, and Elisa Ricci. Fedmvp: Federated multimodal visual prompt tuning for vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

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

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017.

Lu Wang, Xixu Hu, Jindong Wang, and Xing Xie. Fedclip: Fast generalization and personalization for clip in federated learning. *arXiv preprint arXiv:2302.13485*, 2023.

Xin Wang, Daniel E. Quevedo, Dongrun Li, Peng Cheng, Jiming Chen, and Youxian Sun. Data security and privacy for ai-enabled smart manufacturing. *Engineering*, 52:34–39, 2025a. doi: 10.1016/j.eng.2025.08.008.

Zhihao Wang, Wenke Huang, Tian Chen, Zekun Shi, Guancheng Wan, Yu Qiao, Bin Yang, Jian Wang, Bing Li, and Mang Ye. An empirical study of federated prompt learning for vision-language models. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 10705–10713, 2025b.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010.

Mengqiu Xu, Ming Wu, Kaixin Chen, Chuang Zhang, and Jun Guo. The eyes of the gods: A survey of unsupervised domain adaptation methods based on remote sensing data. *Remote Sensing*, 14 (17):4380, 2022. doi: 10.3390/rs14174380.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *ACL*, 2022.

Renrui Zhang, Ziyu Wei, Rongyao Fang, Peng Gao, Hongsheng Li, and Yu Qiao. Tip-adapter: Training-free CLIP-adapter for better vision-language modeling. In *ECCV*, pages 49–70, 2022.

Zhi Zhang, Qizhe Zhang, Zijun Gao, Renrui Zhang, Ekaterina Shutova, Shiji Zhou, and Shanghang Zhang. Gradient-based parameter selection for efficient fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28566–28577, 2024.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 130(9):2337–2348, 2022.

A Additional Theoretical Analysis

This appendix presents detailed proofs of Theorem 3.1 and Lemma 3.2. The former follows a dimension-dependent generalization analysis, while the latter uses a local second-order approximation.

A.1 Proof of Theorem 3.1

For client k, define

$$\Delta_k(h) = L_{D_k}(h) - L_{\widehat{D}_k}(h).$$

Since λ belongs to the probability simplex Δ_K, we have

$$\begin{aligned} \varepsilon(h) &= \sup_{\lambda \in \Delta_K} \left| \sum_{k=1}^{K} \lambda_k \Delta_k(h) \right| \\ &= \max_{1 \le k \le K} |\Delta_k(h)|. \end{aligned} \tag{16}$$

Therefore, it suffices to establish a uniform generalization bound across all clients.

Consider a hypothesis class $\mathcal{H}_m$ parameterized by an m-dimensional vector $u \in \mathbb{R}^m$ satisfying $\|u\|_2 \le R$. Assume that the loss takes values in $[0, M]$ and is uniformly L-Lipschitz with respect to u.

For $n \ge 2$, an R/n-cover of the parameter ball has cardinality bounded by

$$|\mathcal{N}_m| \le (3n)^m.$$

For any fixed hypothesis and client k with $n_k \ge n$ independent training samples, Hoeffding's inequality yields

$$\Pr\left(|\Delta_k(h)| > t\right) \le 2\exp\left(-\frac{2nt^2}{M^2}\right). \tag{17}$$

Applying a union bound over the covering points, all K clients, and both hypothesis classes, with probability at least $1 - \delta$, we obtain

$$\max_{\substack{u \in \mathcal{N}_m \\ 1 \le k \le K}} |\Delta_k(h_u)| \le M\sqrt{\frac{m\log(3n) + \log(4K/\delta)}{2n}}. \tag{18}$$

For an arbitrary parameter vector u, let $\widetilde{u}$ be its nearest covering point. By the Lipschitz assumption, the population and empirical loss differences introduced by this approximation are each bounded by LR/n.

Consequently,

$$\sup_{h \in \mathcal{H}_m} \varepsilon(h) \le \frac{2LR}{n} + M\sqrt{\frac{m\log(3n) + \log(4K/\delta)}{2n}}. \tag{19}$$

For bias-only tuning, the adaptation dimension is

$$m_b = d.$$

Substituting $m = d$ into Eq. (19) gives

$$\sup_{h \in \mathcal{H}_b} \varepsilon(h) \le \frac{2LR}{n} + M\sqrt{\frac{d\log(3n) + \log(4K/\delta)}{2n}}. \tag{20}$$

For full FFN output-projection tuning, the trainable parameters consist of $W_2 \in \mathbb{R}^{d \times r}$ and $b_2 \in \mathbb{R}^d$. Therefore,

$$m_{\mathrm{FFN}} = dr + d.$$

Applying the same bound yields

$$\sup_{h\in\mathcal{H}_{\mathrm{FFN}}} \varepsilon(h) \le \frac{2LR}{n} + M\sqrt{\frac{(dr+d)\log(3n)+\log(4K/\delta)}{2n}}. \tag{21}$$

For fixed K and common regularity constants, these inequalities yield the simplified dimension-dependent bounds stated in Theorem 3.1.

The leading dimension-dependent terms in the two upper bounds scale as $\sqrt{d}$ and $\sqrt{dr+d}$, respectively. Their ratio is $\sqrt{r+1}$, indicating a lower complexity upper bound for bias-only adaptation.

The above analysis considers a representative FFN output projection. For multiple FFN layers, the corresponding trainable parameter dimensions are summed across the considered layers.

A.2 Proof of Lemma 3.2

Let $\phi_k^\star$ denote a local optimum of the objective $L_k(\phi)$. For a sufficiently small perturbation δ, the second-order Taylor expansion gives

$$\begin{aligned} L_k(\phi_k^\star+\delta) - L_k(\phi_k^\star) = \frac{1}{2}\delta^\top H_k \delta \\ + o(\|\delta\|_2^2), \end{aligned} \tag{22}$$

where the first-order term vanishes because $\nabla L_k(\phi_k^\star) = 0$.

We consider synchronizing an entire FFN output-bias block B_j with the corresponding global parameters. Define the block-restricted perturbation as

$$\delta_a = \begin{cases} \phi_a^{(t)} - \phi_{k,a}^\star, & a \in B_j, \\ 0, & a \notin B_j. \end{cases}$$

Substituting the perturbation into the local quadratic approximation gives

$$\Delta L_{k,B_j} \approx \frac{1}{2}\delta_{B_j}^\top H_{k,B_jB_j}\delta_{B_j}. \tag{23}$$

Under the local diagonal curvature approximation

$$H_{k,B_jB_j} \approx \mathrm{diag}\left(\{I_{k,a}\}_{a\in B_j}\right),$$

the block-level synchronization cost becomes

$$\Delta L_{k,B_j} \approx \frac{1}{2}\sum_{a\in B_j} I_{k,a}\left(\phi_a^{(t)} - \phi_{k,a}^\star\right)^2. \tag{24}$$

The empirical Fisher score of an FFN output-bias block is aggregated as

$$I_{k,B_j} = \sum_{a\in B_j} I_{k,a}.$$

The resulting approximation relates blockwise Fisher sensitivity to the local loss change induced by global synchronization. Thus, Fisher-guided block selection provides a local sensitivity-based criterion for retaining important bias blocks as personalized parameters.

B Additional Experimental Analyses

Unless otherwise specified, all experiments in this section are conducted on UCF101 using CLIP ViT-B/16 under the 16-shot setting with 10 clients and 50 communication rounds.

B.1 Equal-Budget Intrinsic Adaptation Comparison

To investigate whether the effectiveness of FFN output-bias tuning can be attributed solely to its compact parameter budget, we compare three intrinsic adaptation interfaces. Random-All-15.36K randomly selects trainable scalar parameters across the CLIP backbone, while Random-output-weight-15.36K selects scalar parameters from FFN output-projection weights. In contrast, FedAvg-output-bias tunes the complete FFN output-projection bias vectors. All three methods update 15,360 scalar parameters, and accuracy is averaged over communication rounds.

Table 7: Comparison of equal-budget intrinsic adaptation methods.

Method	Trainable (K)	Round-avg. Acc. (%)
Random-All-15.36K	15.36	69.28
Random-output-weight-15.36K	15.36	69.25
FedAvg-output-bias	15.36	**77.02**

As shown in Table 7, FedAvg-output-bias achieves 77.02% round-averaged accuracy, outperforming Random-All-15.36K and Random-output-weight-15.36K by 7.74 and 7.77 percentage points, respectively. These results indicate that intrinsic adaptation depends not only on parameter dimensionality but also on the structural location of trainable parameters, supporting our choice of FFN output biases.

B.2 Personalization Strategy Comparison

To further evaluate Fisher-guided personalization, we compare FedIBS with Local-output-bias, Ditto-output-bias, and RandomMask-output-bias operating on the same FFN output-bias subspace.

Local-output-bias performs independent local adaptation without communication, while Ditto-output-bias applies personalized federated optimization. RandomMask-output-bias replaces Fisher-guided selection with a fixed random block ordering, while matching the personalized visual and textual block counts and communication budget of FedIBS for each client and round.

Accuracy is evaluated at the final communication round. Communication is reported as the nominal uplink cost per client per round in KB, consistent with the main comparison.

Table 8: Comparison of personalization strategies.

Method	Final Acc. (%)	Worst (%)	Comm. (KB)
Local-output-bias	97.01	92.80	0
Ditto-output-bias	97.07	**93.91**	31
RandomMask-output-bias	97.10	93.35	4
FedIBS	**97.24**	93.63	**4**

As shown in Table 8, FedIBS achieves the highest final-round mean personalized accuracy of 97.24%. Compared with Ditto-output-bias, FedIBS improves accuracy by 0.17 percentage points while requiring substantially less communication. Under the matched communication budget, FedIBS outperforms RandomMask-output-bias by 0.14 and 0.28 percentage points in mean and worst-client accuracy, respectively. These results support the effectiveness of Fisher-guided block selection.

B.3 Visual and Text Representation Analysis

To examine the effects of different adaptation interfaces on pretrained representations, we compare PromptFL and FedIBS against frozen CLIP. We measure visual and textual representation similarity using centered kernel alignment (CKA), together with feature drift relative to the pretrained model. The analysis uses the same 505 images and 101 class texts, with representations extracted at the final communication round. We denote image and text feature drift by D_v and D_t, respectively.

As shown in Table 9, PromptFL preserves the original visual representations while substantially modifying textual representations. In contrast, FedIBS produces measurable changes in both visual and textual features through intrinsic FFN output-bias adaptation. These findings highlight the ability of FedIBS to directly adapt internal visual representations under heterogeneous client distributions.

Table 9: Visual and textual representation analysis.

Method	Visual CKA	Text CKA	D_v	D_t
PromptFL	1.0000	0.6726	0.0000	0.5919
FedIBS	0.9834	0.9346	0.0317	0.0320

B.4 Personalization-Budget Sensitivity

We investigate the effect of the personalization budget α, which determines the maximum proportion of FFN output-bias blocks retained locally. We vary α from 0 to 1 and report round-averaged accuracy together with the nominal uplink cost per client per round in KB.

Table 10: Effect of the personalization budget α.

α	Round-avg. Acc. (%)	Comm. (KB)
0.0	73.49	30.06
0.2	96.42	24.91
0.5	96.66	14.84
0.8	**96.84**	4.00
1.0	96.79	**0.29**

As shown in Table 10, increasing α substantially improves personalized accuracy while reducing communication costs. Accuracy increases rapidly at smaller budgets and remains stable as more blocks are personalized. The default $\alpha = 0.8$ achieves the highest round-averaged accuracy of 96.84% with 4.00 KB communication per client per round, providing a favorable accuracy–communication trade-off.

B.5 Per-Client Accuracy Distribution

To further evaluate personalization across heterogeneous clients, we examine the distribution of per-client test accuracy. We compare CLIP, PromptFL, FedOTP, FedPGP, FedPHA, FedDAT, and FedIBS. Each method is evaluated on the same 10 clients at the final communication round.

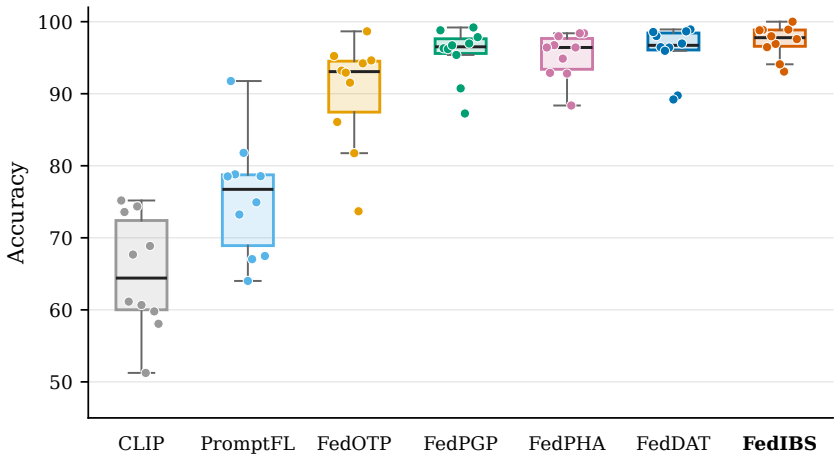

Figure 4: Distribution of per-client test accuracy.

As shown in Fig. 4, FedIBS exhibits a concentrated distribution of final-round client accuracy. Its worst-client accuracy reaches 93.63%, compared with 89.20% for FedDAT, indicating stronger lower-tail performance.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes].

   Justification: The abstract and introduction state that FedIBS fine-tunes FFN projection biases and uses Fisher-guided masking for shareable/personalized blockwise partition, leading to reduced communication cost. These claims are supported by the methodology in Section 3, theoretical justification in Section 3.4 and Appendix A, and experiments in Section 4.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes].

   Justification: The Limitations section discusses the evaluated model architectures, federated settings, and additional client-side computation.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions.
   - The authors should reflect on the scope of the claims made.
   - The authors should reflect on the factors that influence the performance of the approach.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [Yes].

   Justification: Section 3.4 presents Theorem 3.1 and Lemma 3.2 with their assumptions. Appendix A provides their detailed proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material.

- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper?

   Answer: [Yes].

   Justification: Section 4.1 specifies the datasets, partition protocols, baselines, backbone, number of clients, communication rounds, local epochs, batch sizes, and main hyperparameters used in the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Reproducibility can be accomplished in various ways depending on the contribution.

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

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

   Answer: [No].

   Justification: All datasets used in this work are publicly available benchmarks and are properly cited. The implementation will be publicly released after code organization and documentation.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer.
   - The instructions should contain the exact command and environment needed to run to reproduce the results.
   - The authors should provide instructions on data access and preparation.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines.
   - At submission time, to preserve anonymity, the authors should release anonymized versions if applicable.
   - Providing as much information as possible in supplemental material is recommended.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details necessary to understand the results?

   Answer: [Yes].

   Justification: Section 4.1 reports the training and evaluation protocol, including datasets, non-IID partitions, backbone, communication rounds, local epochs, batch size, and hyperparameters.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes].

   Justification: Table 1 reports mean accuracy and standard deviation over three seeds for the main single-domain CLIP transfer experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests.
   - The factors of variability that the error bars are capturing should be clearly stated.
   - The method for calculating the error bars should be explained.
   - The assumptions made should be given.
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it.
   - For asymmetric distributions, the authors should be careful not to show invalid symmetric error bars.
   - If error bars are reported in tables or plots, the authors should explain how they were calculated.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources needed to reproduce the experiments?

   Answer: [Yes].

   Justification: Section 4.1 reports that all experiments are conducted on a single NVIDIA A100 GPU with 40GB memory. The main experimental protocol and hyperparameters are also specified in Section 4.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider.
   - The paper should provide the amount of compute required for each experiment.
   - The paper should disclose whether the full research project required more compute than the experiments reported.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics?

   Answer: [Yes].

   Justification: The work uses public datasets and pretrained models, focuses on privacy-preserving federated adaptation, and does not involve human-subject experiments, private data collection, or unsafe data release.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances.
   - The authors should make sure to preserve anonymity.

10. **Broader impacts**

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

    Answer: [Yes].

Justification: FedIBS may reduce communication and deployment costs for privacy-preserving VLM adaptation. Potential negative impacts are mainly inherited from pretrained VLMs and downstream deployments, including bias, fairness, and misuse risks.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why.
- Examples of negative societal impacts include malicious or unintended uses, fairness considerations, privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications.
- The authors should consider possible harms arising from intended use, incorrect results, and misuse.
- If there are negative societal impacts, the authors could also discuss mitigation strategies.

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse?

Answer: [N/A].

Justification: This work does not release a new high-risk pretrained model, image generator, scraped dataset, or private dataset. It evaluates an adaptation method on public benchmarks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse should be released with necessary safeguards.
- Datasets scraped from the Internet could pose safety risks.
- We encourage authors to make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets used in the paper properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes].

Justification: The paper cites the original works for CLIP and all datasets used in the experiments. All datasets are standard public benchmarks commonly used for academic research.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license should be included for each asset.
- For scraped data, the copyright and terms of service should be provided.
- If assets are released, license and copyright information should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [N/A].

Justification: The paper does not introduce a new dataset or pretrained model asset. If code is released after acceptance, documentation and usage instructions will be provided.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets if applicable.

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

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

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing experiments or human-subject studies.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine.
    - According to the NeurIPS Code of Ethics, workers involved in data collection should be paid at least the minimum wage.

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

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed, and whether IRB approvals were obtained?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing or human-subject research, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country, IRB approval may be required for human-subject research.
    - Authors are expected to adhere to the NeurIPS Code of Ethics and institutional guidelines.
    - For initial submissions, do not include information that would break anonymity.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research?

    Answer: [N/A].

    Justification: The core method does not use LLMs as an important, original, or non-standard component. Any use of writing or editing assistance, if applicable, does not affect the methodology, experiments, or scientific claims.

    Guidelines:

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