OFLBE: DO OUR BEST TO MAKE ONE-SHOT FEDERATED LEARNING BETTER

Anonymous authors
Paper under double-blind review

ABSTRACT

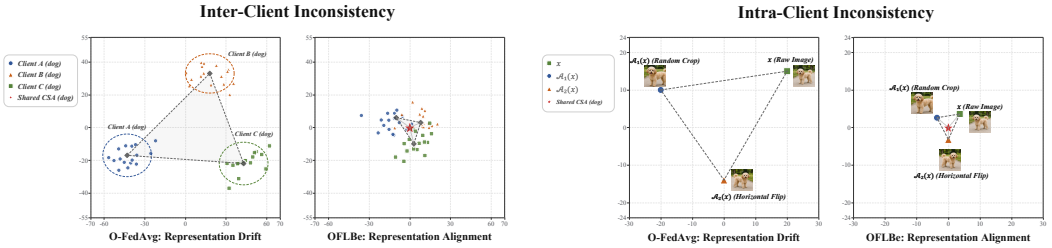
One-shot federated learning (OFL) enables collaborative training across multiple clients through a single round of client-server interaction, substantially reducing communication costs compared to traditional multi-round FL. However, under non-IID data settings, clients perform local training independently without multi-round interactions with the server to align the representation space across clients, leading to severe feature drift. To address this issue, we propose OFLBe, an OFL framework based on consistent anchor semantic space optimization. OFLBe leverages a pre-trained CLIP text encoder to construct shared class-level semantic anchors, providing a unified semantic representation space reference for all clients. During local training, OFLBe introduces an anchor-guided feature alignment loss and a learnable prototype loss to guide cross-client learning toward a consistent representation space with intra-class compactness and inter-class discriminability. Moreover, on the server side, instead of directly averaging the clients' model parameters, OFLBe aggregates the uploaded local prototypes and the features output by the local models during inference based on the similarity between the uploaded local prototypes and the class-level semantic anchors, thereby maximizing the utilization of knowledge from a single client-server interaction. Experiments on three datasets under various data heterogeneity settings demonstrate that OFLBe achieves the best performance compared with 12 OFL baselines.

1 INTRODUCTION

Federated learning (FL) allows multiple clients to collaboratively train a global model without exposing their raw data, making it a widely adopted paradigm in distributed machine learning McMahan et al. (2017); Caldarola et al. (2025); Cheng et al. (2026). In traditional FL, although multiple rounds of client-server interactions help correct local biases across clients and gradually align their representation spaces, this mechanism introduces a significant communication burden (e.g., more than 250GB for a simple VGG19 model cumulatively Wu et al. (2020); Zeng et al. (2024)) and high communication time (e.g., more than 194 hours for one-round transmission of GPT-3 Tang et al. (2024)), making it difficult to apply in practical scenarios Zhang et al. (2022); Zeng et al. (2025a); Berardini et al. (2026); Liu et al. (2026).

One-shot federated learning (OFL) significantly reduces communication overhead by limiting the number of client-server interaction rounds to one Guha et al. (2019); Zhang et al. (2022); Zeng et al. (2025a); Chen et al. (2025); Liu et al. (2026). Hence, the OFL paradigm is well-suited for multi-client scenarios with limited bandwidth and high privacy sensitivity Zhang et al. (2022); Chen et al. (2025); Yao et al. (2026); Cheng et al. (2026). However, in OFL, each client trains its local model independently on private and skewed data, lacking multi-round interactions with the server to align the representation spaces across clients. As a result, the learned representation spaces suffer from severe feature drift (see Figure 1a). Moreover, within each client, samples with the same semantics also exhibit representation inconsistency in the local model's output features (see Figure 1b). Consequently, the global model obtained by aggregating the uploaded local models at the server shows a significant performance gap (over 30%) compared to multi-round FL. To date, most OFL methods focus on improving the server-side aggregation mechanism, which can be broadly categorized into four types: distillation-based methods Li et al. (2021a); Diao et al. (2023); Song et al. (2023), generative methods Heinbaugh et al. (2023); Yang et al. (2024), optimization-based methods Dennis et al. (2021); Su et al. (2023); Jhunjhunwala et al. (2024); Liu et al. (2024) and selective ensemble

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.



(a) Inter-client inconsistency.

(b) Intra-client inconsistency.

Figure 1: **Illustrations of inter-client and intra-client inconsistency.** Under non-IID data distributions, different clients learn distinctly different semantic representation centers for samples of the same class, while samples with the same semantics can also exhibit inconsistent representations within a client. Shared CSA guides heterogeneous clients and semantically equivalent samples toward a consistent representation space.

methods Zhang et al. (2022); Wang et al. (2023); Dai et al. (2024); Zeng et al. (2024). However, improving aggregation mechanism at the server side cannot fundamentally correct the representation bias learned by individual clients, thus leading to limited performance gains. These limitations motivate us to rethink OFL: *how can we provide a shared and unique class-level semantic anchor (CSA) space for all clients before the single communication round, so that each client can learn a consistent representation space?*

To address this issue, we propose OFLBe, a framework that optimizes consistent anchor semantic spaces for OFL. On the client side, we leverage a pre-trained CLIP text encoder to provide a unified CSA representation space for all clients. We then design a semantic consistency constrained training paradigm guided by these CSAs, which pulls the representation spaces learned by individual clients toward the shared CSA space, thereby achieving cross-client representation alignment. On the server side, OFLBe designs an anchor-aware aggregation and inference mechanism to maximize the integration of knowledge from each client's single upload. Specifically, instead of directly averaging client model parameters, the server aggregates local prototypes from each client based on their similarity to the CSAs. Meanwhile, the uploaded local models are used for inference, and their outputs are weighted and fused based on the degree of consistency between the local prototypes and the CSA. The final prediction is obtained by matching the fused features with global prototypes. Our main contributions are summarized as follows:

- We revisit the performance bottleneck of OFL from the perspective of semantic inconsistency across data-heterogeneous clients, and propose OFLBe, a framework based on consistent anchor semantic space optimization. OFLBe aligns the representation spaces learned by individual clients by constructing shared CSAs and designing a semantic consistency constrained local training, thereby effectively avoiding feature drift.
- On the server side, we designed an anchor-aware aggregation and inference mechanism. This mechanism aggregates the uploaded local prototypes and fuses the output features of each client's local model based on the similarity between their uploaded local prototypes and the CSA, thereby maximizing the use of the knowledge from each client's single upload.
- Extensive experiments on CIFAR-10, CIFAR-100 and Tiny-ImageNet under various data heterogeneity settings demonstrate that OFLBe achieves SoTA performance.

2 Methodology

2.1 Problem Formulation

OFLBe considers an OFL system with a central sever and M clients. Each client m holds a private dataset $\mathcal{D}_m = \{(x_i^m, y_i^m)\}_{i=1}^{N_m}$ and a local model $\mathcal{N}_m(\Theta_m)$. Here x_i^m is the input sample, $y_i^m \in$

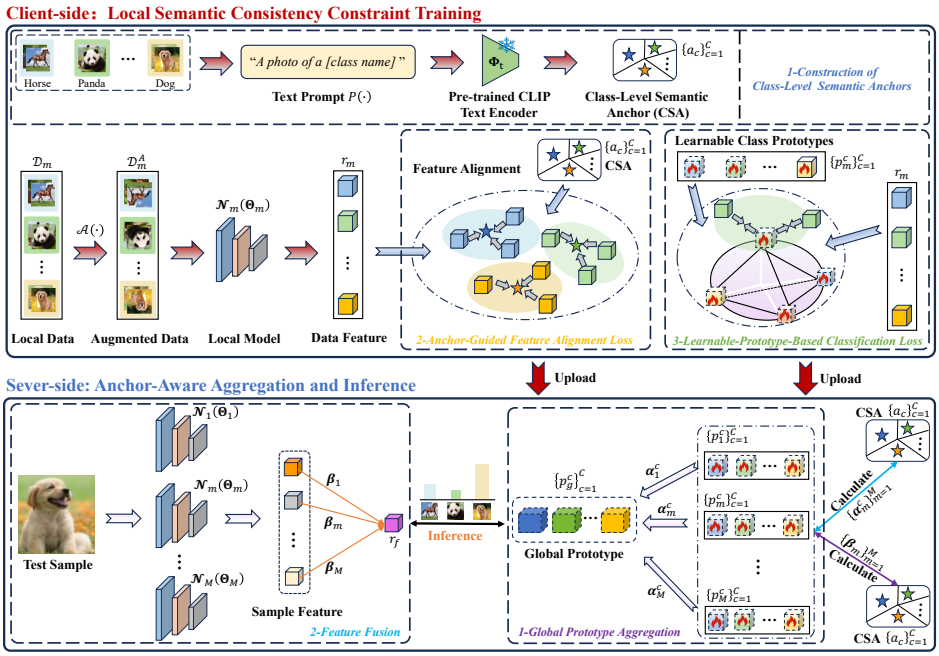


Figure 2: Framework of OFLBe.

$\{1, ..., C^m\}$ is the class label, $N_m = |\mathcal{D}_m|$ is the number of local samples. C^m denotes the number of categories owned by client m. $\mathcal{D} = \bigcup_{m=1}^{M} \mathcal{D}_m$ is the complete dataset. C denotes the number of classes in dataset $\mathcal{D}$. In OFL, all clients independently conduct local training on their own datasets:

$$\Theta_m = \arg\min_{\Theta_m} \mathbb{E}_{(x^m, y^m)\sim\mathcal{D}_m} \ell(\mathcal{N}_m(\Theta_m, x^m), y^m), \tag{1}$$

where $\ell(\cdot)$ is the loss function. After local training, each client uploads its local model $\mathcal{N}_m(\Theta_m^{'})$ and local class prototype $\{p_m^c\}_{c=1}^{C}$ to the server. Unlike multi-round FL, OFLBe aims to achieve higher test accuracy on $\mathcal{D}$ using $\{\mathcal{N}_m(\Theta_m^{'})\}_{m=1}^{M}$ and $\{\{p_m^c\}_{c=1}^{C}\}_{m=1}^{M}$ in a single round of communication:

$$\max_{(x,y)\sim\mathcal{D}} \mathcal{T}((\{\mathcal{N}_m(\Theta_m^{'}, x)\}_{m=1}^{M}, \{\{p_m^c\}_{c=1}^{C}\}_{m=1}^{M}), y). \tag{2}$$

2.2 Solution Overview

As shown in Figure 2, OFLBe comprises two stages: a client-side stage and a server-side stage.

Client-side: Local Semantic Consistency Constraint Training. We use a pre-trained CLIP text encoder to construct a set of globally shared class-level semantic anchors (CSAs) for all classes. CSAs provide a unified semantic reference space for all clients. Each client is then trained under local semantic consistency constraints based on CSAs. After a single round of local training, each client transmits its local model $\mathcal{N}_m(\Theta_m^{'})$ and local class prototypes $\{p_m^c\}_{c=1}^{C}$ to the server.

Server-side: Anchor-Aware Aggregation and Inference. After receiving $\{\mathcal{N}_m(\Theta_m^{'})\}_{m=1}^{M}$ and $\{\{p_m^c\}_{c=1}^{C}\}_{m=1}^{M}$ from all clients, the server performs anchor-aware aggregation and inference. Specifically, the server performs class prototype aggregation based on the similarity between each client's $\{p_m^c\}_{c=1}^{C}$ and the CSA, thereby obtaining the global class prototype $\{p_g^c\}_{c=1}^{C}$. Furthermore, the server does not aggregate the parameters of the local models uploaded by clients. Instead, during inference, the server aggregates the feature outputs from each client's local model and calculates the similarity between the fused feature outputs and $\{p_g^c\}_{c=1}^{C}$ to obtain the final prediction results.

2.3 Client Side: Local Semantic Consistency Constraint Training

In OFL, each client is trained independently and interacts with the server only once. Due to data heterogeneity, such isolated training often leads to inconsistent representations and feature drift

across clients. To address this issue, we introduce a local semantic consistency constraint training paradigm based on shared CSAs and learnable class prototypes.

2.3.1 CONSTRUCTION OF CLASS-LEVEL SEMANTIC ANCHORS.

To enable clients with biased data to learn a consistent feature space for samples belonging to the same class, we use a pre-trained CLIP text encoder $\Phi_t(\cdot)$ to construct a set of CSA. For each class $c \in C$, the CSA is defined as:

$$a_c = \frac{\Phi_{\mathrm{t}}(P(c))}{\|\Phi_{\mathrm{t}}(P(c))\|_2}, \tag{3}$$

where $P(c)$=*"a photo of a [class name]"*. The CSA set $\{a_c\}_{c=1}^{C}$ is generated only once before training and is shared among all clients.

2.3.2 ANCHOR-GUIDED FEATURE ALIGNMENT LOSS.

For client m, we perform data augmentation on dataset $\mathcal{D}_m = \{(x_m^i, y_m^i)\}_{i=1}^{N_m}$ to obtain $\mathcal{D}_m^A = \{(\mathcal{A}(x_m^i), y_m^i)\}_{i=1}^{N_m}$. Here, $\mathcal{A}(\cdot)$ is the augmentation operator. In local training, client m processes each sample $(\mathcal{A}(x_m^i), y_m^i)$ through $\mathcal{N}_m(\Theta_m)$ to extract its feature representation $r_m^i = \mathcal{N}_m(\Theta_m, \mathcal{A}(x_m^i))$. To enable clients to learn consistent class representation spaces within the same class while maintaining distinctiveness from the representation spaces of other classes, we propose the following anchor-guided feature alignment loss:

$$\mathcal{L}_{\mathrm{A}}^{(m)} = -\frac{1}{N_m}\sum_{i=1}^{N_m}\log\frac{\exp(\cos(r_m^i, a_{y_m^i})/\tau_a)}{\sum_{c=1}^{C}\exp(\cos(r_m^i, a_c)/\tau_a)}, \tag{4}$$

where $y_m^i = c$ and τ_a denotes the temperature parameter.

2.3.3 LEARNABLE-PROTOTYPE-BASED CLASSIFICATION LOSS.

To further improve intra-class compactness and inter-class distinguishability, We introduce learnable-prototype-based classification loss. Specifically, client m maintains a set of learnable class prototypes $\{p_m^c\}_{c=1}^{C}$. During local training, $\{p_m^c\}_{c=1}^{C}$ is updated by minimizing the difference between the local model output r_m^i and $\{p_m^c\}_{c=1}^{C}$. The learnable-prototype-based classification loss is defined as:

$$\mathcal{L}_{\mathrm{P}}^{(m)} = -\frac{1}{N_m}\sum_{i=1}^{N_m}\log\frac{\exp\left(r_m^i p_m^c/\tau_p\right)}{\sum_{c=1}^{C}\exp(r_m^i p_m^c/\tau_p)}, \tag{5}$$

where τ_p denotes the temperature parameter. The goal of Loss $\mathcal{L}_{\mathrm{P}}^{(m)}$ is to obtain a set of representative and highly discriminative prototypes $\{p_m^c\}_{c=1}^{C}$.

2.3.4 TOTAL LOCAL LOSS.

The overall training objective for client m is defined as follows:

$$\mathcal{L}^{(m)} = \mathcal{L}_{\mathrm{A}}^{(m)} + \lambda\mathcal{L}_{\mathrm{P}}^{(m)}, \tag{6}$$

where λ is hyperparameter balancing the two loss term. The model parameters are updated via gradient descent:

$$\Theta_m^{'} \leftarrow \Theta_m - \eta\nabla\mathcal{L}^{(m)}, \tag{7}$$

where η is the learning rate. After training, each client uploads its local model $\mathcal{N}_m(\Theta_m^{'})$ and learnable prototype set $\{p_m^c\}_{c=1}^{C}$ to the server for semantic-aware aggregation and inference fusion.

2.4 SERVER SIDE: ANCHOR-AWARE AGGREGATION AND INFERENCE

After receiving the uploaded local models $\{\mathcal{N}_m(\Theta_m^{'})\}_{m=1}^{M}$ and learnable prototype sets $\{\{p_m^c\}_{c=1}^{C}\}_{m=1}^{M}$ from all clients in a single communication round, the server performs Anchor-aware integration to construct global semantic representations and perform inference.

2.4.1 GLOBAL PROTOTYPE AGGREGATION.

To obtain a better global prototype, we measure the reliability of prototypes uploaded by each client. Specifically, for each class $c \in C$, we compute the similarity between each local prototype p_m^c and the shared CSA a_c:

$$R_m^c = \cos(p_m^c, a_c),\ s.t.\ m \in \mathcal{S}_c, \tag{8}$$

where $\mathcal{S}_c = \{m \mid c \in C_m\}_{m=1}^M$ denote the set of clients that contain samples of class c. Therefore, the prototype aggregation weight for client m in class c is:

$$\alpha_m^c = \frac{\exp(R_m^c/\tau_g)}{\sum_{j\in\mathcal{S}_c}\exp(R_j^c/\tau_g)}, \tag{9}$$

where τ_g denotes the temperature parameter. Then the server aggregates the corresponding local prototypes $\{p_m^c\}_{m\in\mathcal{S}_c}$ to obtain the global prototype p_g^c:

$$p_g^c = \sum_{m\in\mathcal{S}_c} \alpha_m^c p_m^c. \tag{10}$$

Since the global aggregation in $\{p_g^c\}_{c=1}^C$ uses shared CSAs to evaluate the credibility of prototypes uploaded by clients, it achieves higher class prototype quality compared to traditional average aggregation.

2.4.2 FEATURE FUSION.

Under the constraint of a single communication round, directly aggregating model parameters is unreliable. This is because each client's local model is optimized independently with respect to data skew, and directly averaging model parameters makes the results susceptible to the influence of clients with missing or weak classes. Therefore, instead of aggregating the local models, we retain all local models uploaded by clients and fuse the output features from each client during the inference process. To determine the fusion coefficient of each client's output features, we evaluate their contribution scores based on the similarity between their prototypes and the CSA.

For each class c, the contribution score for clients possessing a class-c prototype follows Eq. (8), namely $V_m^c = R_m^c$. If a client contains no samples about class c, we adopt the average score of the client set $\mathcal{S}_c$ to compensate for its contribution score on class c:

$$V_{m'}^c = \frac{1}{|\mathcal{S}_c|}\sum_{m\in\mathcal{S}_c} R_m^c, \tag{11}$$

where $|\mathcal{S}_c|$ denotes the number of clients in set $\mathcal{S}_c$, m' denotes a client that does not possess samples of class c. Therefore, for any client m, the formula for calculating the contribution score on class c is as follows:

$$\mathcal{V}_m^c = \begin{cases} V_m^c = \cos(p_m^c, a_c), & m \in \mathcal{S}_c, \\ V_{m'}^c = \frac{1}{|\mathcal{S}_c|}\sum_{m\in\mathcal{S}_c} V_m^c, & m \notin \mathcal{S}_c. \end{cases} \tag{12}$$

The overall contribution score of client m is then computed by averaging its class-wise contribution scores over all classes:

$$\mathcal{U}_m = \frac{1}{C}\sum_{c=1}^{C} \mathcal{V}_m^c. \tag{13}$$

Consequently, for the output features at the inference stage, the fusion coefficient for each client is given by Eq. (14):

$$\beta_m = \frac{\exp(\mathcal{U}_m/\tau_f)}{\sum_{j=1}^M \exp(\mathcal{U}_j/\tau_f)}, \tag{14}$$

where τ_f is the temperature parameter.

2.4.3 INFERENCE WITH GLOBAL PROTOTYPES.

Given a test sample x, the server obtains its inference-stage features by fusing the output features from all participating clients:

$$r_f = \sum_{m=1}^{M} \beta_m \mathcal{N}_m(\Theta_m^{'}, x). \tag{15}$$

The final prediction is obtained by computing the cosine similarity between r_f and $\{p_g^c\}_{c=1}^{C}$, and then selecting the class with the highest score:

$$\hat{y} = \arg\max_{c \in C} \cos(r_f, p_g^c). \tag{16}$$

This prototype-based inference avoids classifier-head dependency and enables semantic-level prediction in a unified shared prototype space.

3 EXPERIMENTS

3.1 EXPERIMENTAL SETUP

Datasets and Data Heterogeneity. Following prior works Tan et al. (2022); Huang et al. (2023); Zeng et al. (2024); Zhang et al. (2022); Tang et al. (2024), we evaluated OFLBe on three benchmark datasets: CIFAR-10 Krizhevsky & Hinton (2009), CIFAR-100 Krizhevsky & Hinton (2009) and Tiny-ImageNet Le & Yang (2015). To simlated non-IID data partitions, we utilize the Dirichlet distribution. The heterogeneity level is controlled by concentration parameter α , where a smaller α indicates greater heterogeneity. Following the seting in FAFI Zeng et al. (2025a), we set $\alpha = \{0.05, 0.1, 0.3, 0.5\}$.

Baselines. We compare OFLBe with twelve OFL methods, including one-shot FedAvg (O-FedAvg) Guha et al. (2019), MA-Echo Yang et al. (2024), FedFisher Jhunjhunwala et al. (2024), DENSE Zhang et al. (2022), FedDF Lin et al. (2020), F-ADI Yin et al. (2018), F-DAFL Chen et al. (2019), directly Ensemble, Co-Boosting Dai et al. (2024), FuseFL Tang et al. (2024), IntactOFL Zeng et al. (2024) and FAFI Zeng et al. (2025a). We also compare with five statistics-based FL methods: FedCGS Guan et al. (2025), GH-OFL Turazza et al. (2026), AFL He et al. (2025), DeepAFL Tang et al. (2026) and SAFLe Bacellar et al. (2026).

Implementation Details. Following FAFI Zeng et al. (2025a), we set the number of clients with $M = 5$, the number of local training epochs to $E = 200$, use SGD as the optimizer with a momentum of 0.9, a batch size of 256 and a learning rate of 0.05. All methods adopt ResNet-18 as the feature extractor. The pre-trained CLIP text encoder is ViT-B/16. More details are provided in Appendix D.

3.2 EFFECTIVENESS

Table 1 presents the experimental results of OFLBe compared with representative OFL baseline methods under non-IID settings. The experimental results show that OFLBe consistently achieves the best performance across all datasets and at all levels of heterogeneity. Notably, in highly heterogeneous scenarios where $\alpha = \{0.05, 0.1\}$, OFLBe exhibits more significant advantages on CIFAR-10, CIFAR-100 and Tiny-ImageNet, respectively. This demonstrates the effectiveness of OFLBe against severe data distribution shifts. Table 2 reports the experimental results of OFLBe compared with statistical FL methods. The results show that OFLBe achieves higher accuracy than these statistical baselines.

3.3 SCALABILITY

To evaluate the scalability of OFLBe under different federated scales, we vary the number of clients on CIFAR-10 with $\alpha = 0.5$, where $M \in \{5, 10, 25, 50, 100\}$. The results are reported in Table 3. As suggested in Lian et al. (2017), increasing client participation usually introduces more severe statistical heterogeneity and makes server-side integration more challenging in federated learning.

Table 1: Main performance comparison on CIFAR-10, CIFAR-100 and Tiny-ImageNet under Dirichlet non-IID partitions with different concentration parameters α. **Bold** numbers indicate the best performance, underlined numbers indicate the second-best performance and Δ denotes the absolute improvement over the strongest baseline.

Methods	CIFAR-10				CIFAR-100				Tiny-ImageNet			
	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$
MA-Echo	$36.77_{\pm 0.91}$	$51.23_{\pm 0.28}$	$60.14_{\pm 0.21}$	$64.21_{\pm 0.23}$	$19.54_{\pm 0.45}$	$29.11_{\pm 0.26}$	$37.77_{\pm 0.24}$	$41.94_{\pm 0.21}$	$15.46_{\pm 0.66}$	$22.23_{\pm 0.56}$	$23.46_{\pm 0.19}$	$28.21_{\pm 0.42}$
O-FedAvg	$12.13_{\pm 2.11}$	$17.43_{\pm 0.51}$	$28.07_{\pm 0.89}$	$35.42_{\pm 0.67}$	$4.77_{\pm 0.21}$	$6.45_{\pm 0.71}$	$10.67_{\pm 0.31}$	$12.13_{\pm 0.05}$	$5.67_{\pm 0.45}$	$8.31_{\pm 0.21}$	$13.61_{\pm 0.10}$	$13.71_{\pm 0.16}$
FedFisher	$40.03_{\pm 1.11}$	$47.01_{\pm 1.81}$	$49.33_{\pm 1.52}$	$50.34_{\pm 1.32}$	$16.56_{\pm 2.67}$	$18.98_{\pm 2.09}$	$27.24_{\pm 1.92}$	$31.44_{\pm 1.87}$	$15.65_{\pm 1.54}$	$17.89_{\pm 1.46}$	$19.54_{\pm 1.31}$	$20.77_{\pm 1.15}$
FedDF	$35.53_{\pm 0.67}$	$41.58_{\pm 0.80}$	$44.78_{\pm 0.60}$	$54.58_{\pm 0.73}$	$15.07_{\pm 0.74}$	$27.17_{\pm 0.55}$	$31.23_{\pm 0.79}$	$35.39_{\pm 0.47}$	$11.45_{\pm 0.40}$	$16.32_{\pm 0.33}$	$17.79_{\pm 0.57}$	$27.55_{\pm 0.66}$
F-ADI	$35.93_{\pm 1.56}$	$48.35_{\pm 1.23}$	$52.66_{\pm 1.44}$	$58.78_{\pm 1.67}$	$14.65_{\pm 0.98}$	$28.13_{\pm 1.24}$	$33.18_{\pm 0.67}$	$39.44_{\pm 1.11}$	$13.92_{\pm 1.99}$	$19.00_{\pm 1.78}$	$26.01_{\pm 1.44}$	$29.98_{\pm 1.34}$
F-DAFL	$38.32_{\pm 1.40}$	$46.34_{\pm 1.12}$	$54.03_{\pm 1.71}$	$59.09_{\pm 2.23}$	$16.31_{\pm 0.33}$	$26.80_{\pm 1.33}$	$34.89_{\pm 1.45}$	$37.88_{\pm 1.34}$	$15.12_{\pm 1.34}$	$19.01_{\pm 1.11}$	$23.78_{\pm 1.23}$	$27.98_{\pm 1.10}$
DENSE	$38.37_{\pm 1.08}$	$50.26_{\pm 0.24}$	$59.76_{\pm 0.45}$	$62.19_{\pm 0.12}$	$18.37_{\pm 2.43}$	$32.03_{\pm 0.44}$	$37.33_{\pm 0.48}$	$38.84_{\pm 0.39}$	$18.77_{\pm 0.67}$	$22.25_{\pm 0.33}$	$28.14_{\pm 0.34}$	$32.34_{\pm 0.32}$
Ensemble	$41.36_{\pm 0.67}$	$45.43_{\pm 0.32}$	$62.18_{\pm 0.34}$	$61.61_{\pm 0.23}$	$20.46_{\pm 0.62}$	$26.23_{\pm 0.55}$	$38.01_{\pm 0.67}$	$41.61_{\pm 0.77}$	$13.28_{\pm 0.67}$	$15.38_{\pm 0.23}$	$17.53_{\pm 0.31}$	$28.50_{\pm 0.46}$
Co-Boosting	$39.20_{\pm 0.81}$	$58.49_{\pm 1.24}$	$67.21_{\pm 1.76}$	$70.24_{\pm 2.34}$	$20.19_{\pm 1.44}$	$27.59_{\pm 1.35}$	$39.30_{\pm 1.30}$	$42.67_{\pm 1.40}$	$19.00_{\pm 1.45}$	$21.90_{\pm 1.20}$	$29.24_{\pm 1.32}$	$30.78_{\pm 2.01}$
FuseFL	$54.42_{\pm 0.41}$	$73.79_{\pm 0.34}$	$84.58_{\pm 0.91}$	$84.34_{\pm 0.88}$	$29.12_{\pm 0.23}$	$36.86_{\pm 0.38}$	$45.12_{\pm 0.51}$	$49.30_{\pm 0.32}$	$22.15_{\pm 2.11}$	$29.28_{\pm 2.04}$	$33.04_{\pm 1.79}$	$34.34_{\pm 1.81}$
IntactOFL	$48.22_{\pm 0.43}$	$61.13_{\pm 0.63}$	$70.21_{\pm 0.60}$	$79.93_{\pm 0.23}$	$27.99_{\pm 0.67}$	$39.15_{\pm 0.46}$	$41.86_{\pm 0.60}$	$46.78_{\pm 0.78}$	$20.45_{\pm 0.34}$	$28.43_{\pm 0.17}$	$30.15_{\pm 0.12}$	$35.09_{\pm 0.14}$
FAFI	$71.84_{\pm 1.53}$	$77.83_{\pm 1.32}$	$84.76_{\pm 0.46}$	$88.74_{\pm 0.11}$	$31.02_{\pm 1.17}$	$45.48_{\pm 1.01}$	$56.65_{\pm 0.91}$	$61.07_{\pm 0.55}$	$36.96_{\pm 0.92}$	$43.62_{\pm 0.77}$	$53.32_{\pm 0.50}$	$56.48_{\pm 0.32}$
OFLBe (Ours)	$\mathbf{80.89}_{\pm 0.98}$	$\underline{\mathbf{84.14}}_{\pm 0.56}$	$\underline{\mathbf{90.32}}_{\pm 0.59}$	$\underline{\mathbf{92.07}}_{\pm 0.14}$	$\underline{\mathbf{55.32}}_{\pm 0.94}$	$\underline{\mathbf{61.57}}_{\pm 1.23}$	$\underline{\mathbf{70.13}}_{\pm 0.20}$	$\underline{\mathbf{71.98}}_{\pm 0.22}$	$\underline{\mathbf{51.81}}_{\pm 0.02}$	$\underline{\mathbf{54.65}}_{\pm 0.42}$	$\underline{\mathbf{60.08}}_{\pm 0.03}$	$\underline{\mathbf{61.44}}_{\pm 0.18}$
Δ	+9.05	+6.31	+5.56	+3.33	+24.30	+16.09	+13.48	+10.91	+14.85	+11.03	+6.76	+4.96

Table 2: Accuracy on CIFAR-10 and CIFAR-100.

Methods	CIFAR-10 ($\alpha = 0.5$)	CIFAR-100 ($\alpha = 0.5$)
FedCGS (AAAI'25)	63.95	39.95
FAFI (ICML'25)	88.74	61.07
GH-NB$_{\text{diag}}$ (ICLR'26)	78.84	55.51
GH-LDA (ICLR'26)	86.05	63.92
GH-QDA$_{\text{full}}$ (ICLR'26)	84.40	66.52
FisherMix (ICLR'26)	84.74	66.99
Proto-Hyper (ICLR'26)	85.74	64.05
AFL (CVPR'25)	80.75	58.56
DeepAFL (ICLR'26)	86.43	66.98
SAFLe (CVPR'26)	90.73	70.61
OFLBe (Ours)	**92.07**$_{\uparrow 1.34}$	**71.98**$_{\uparrow 1.37}$

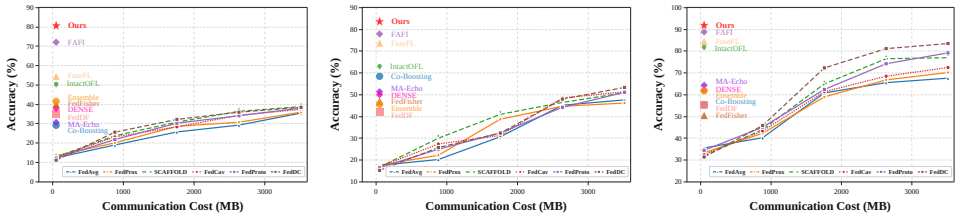
Table 3: Scalability with different numbers of clients on CIFAR-10 with $\alpha = 0.5$.

Methods	$M = 5$	$M = 10$	$M = 25$	$M = 50$	$M = 100$
MA-Echo	64.21	52.64	48.36	45.35	38.54
O-FedAvg	35.42	32.09	28.03	28.24	27.14
FedFisher	50.34	45.67	34.66	29.09	28.89
FedDF	54.58	48.88	35.44	29.91	25.66
F-ADI	59.34	46.33	31.83	27.66	24.89
F-DAFL	58.59	45.45	32.88	29.98	28.91
DENSE	62.19	54.67	49.32	48.67	43.34
Ensemble	61.61	60.44	58.44	52.51	45.72
Co-Boosting	55.34	51.11	49.32	44.56	42.45
FuseFL	84.34	78.28	62.12	42.18	37.11
IntactOFL	79.93	69.11	64.32	59.45	53.21
FAFI	88.74	86.96	85.25	81.32	75.37
OFLBe (Ours)	**92.07**$_{\uparrow 3.33}$	**91.65**$_{\uparrow 4.69}$	**89.48**$_{\uparrow 4.23}$	**86.10**$_{\uparrow 4.78}$	**79.18**$_{\uparrow 3.81}$

We also reach a similar conclusion, as the number of clients M increases, the performance decreases, which is consistent with Zhang et al. (2022); Dai et al. (2024); Lian et al. (2017); Zeng et al. (2024). Despite this challenge, our method consistently achieves the best performance across all client scales.

3.4 Communication Efficiency

We further compare the efficiency of the OFLBe to existing OFL methods and multi-round FL baselines by evaluate the relationship between test accuracy and cumulative communication cost on CIFAR-10 in Figure 3. We select six representative multi-round FL methods, more details are shown in Appendix E. Note that our method achieves higher performance than all other baselines, while incurring a lower communication overhead. Notably, Although multi-round methods can gradually improve their performance as the number of communication rounds increases (Figure 3 only presents the first 80 rounds' results of multi-round FL methods), such improvement requires repeated model transmission and substantially higher cumulative communication cost. In summary, the proposed method provides a more efficient solution for OFL by improving accuracy while preserving the low communication nature of one-shot collaboration.



(a) $\alpha = 0.05$ (b) $\alpha = 0.1$ (c) $\alpha = 0.5$

Figure 3: Test accuracy vs. communication cost on CIFAR-10.

Table 4: Ablation study of $\mathcal{L}_A$, $\mathcal{L}_P$ and AAAI on CIFAR-10 under different Dirichlet settings.

$\mathcal{L}_A$	$\mathcal{L}_P$	AAAI	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$
			12.13	17.43	28.07	35.42
✓			77.68	82.18	88.65	90.04
	✓		75.11	79.46	87.56	88.02
✓	✓		79.77	83.68	88.98	91.17
✓	✓	✓	**80.89**	**84.14**	**90.32**	**92.07**

Table 6: Comparison of different model architectures on Tiny-ImageNet ($\alpha = 0.5$) in terms of accuracy and memory cost.

Architecture	Memory Cost	FAFI	**OFLBe (Ours)**
ResNet-18	11.2 M	57.23	**61.44**$_{\uparrow 4.21}$
ResNet-50	25.6 M	58.12	**62.89**$_{\uparrow 4.77}$
ViT-base	86 M	58.09	**63.05**$_{\uparrow 4.96}$

3.5 Ablation Study

To verify the contribution of each component, we conduct ablation studies on CIFAR-10 under different Dirichlet non-IID settings. As shown in Table 4, we progressively evaluate the effects of *Anchor-Guided Feature Alignment Loss* ($\mathcal{L}_A$), *Learnable-Prototype-Based Classification Loss* ($\mathcal{L}_P$) and *Anchor-Aware Aggregation and Inference* (AAAI). First, both $\mathcal{L}_A$ and $\mathcal{L}_P$ bring substantial improvements over the base model. Second, combining $\mathcal{L}_A$ and $\mathcal{L}_P$ further improves the performance, demonstrating their complementary effects. Specifically, $\mathcal{L}_A$ provides a stable semantic reference for view-invariant representation learning, while $\mathcal{L}_P$ enhances intra-class compactness and inter-class separability through learnable prototypes. Finally, AAAI brings additional gains of 1.12, 0.46, 1.34 and 0.90 percentage points. This shows that improving local representation quality alone is not sufficient for one-shot integration. The server still needs to account for residual cross-client inconsistency. Overall, the ablation results confirm that $\mathcal{L}_A$, $\mathcal{L}_P$ and AAAI are all necessary.

3.6 Model Heterogeneity Scenario

To evaluate the applicability of the OFLBe under model heterogeneity, we further consider the case where different clients use different model architectures. Specifically, under the $\alpha = 0.1$ setting on CIFAR-10, we construct three model-heterogeneous experiments, where client models are selected from different architectures, including LeNet, ResNet-18, VGG, MobileNet and ResNet-50. Each client maps its extracted features into a shared latent space through a lightweight projection layer. The experimental results are shown in Table 5. The results indicate that the OFLBe still achieves strong performance even when client model architectures are inconsistent. This suggests that the OFLBe does not rely on all clients adopting exactly the same model architecture. Instead, through feature-level semantic alignment and the server-side fusion mechanism, it can alleviate the representational discrepancies caused by model heterogeneity to a certain extent.

Table 5: Performance under model heterogeneity with different client model configurations.

Case	**Client Models**					**Accuracy**		
	Client 1	Client 2	Client 3	Client 4	Client 5	IntactOFL	FAFI	**OFLBe (Ours)**
Case #1	LeNet	ResNet-18	VGG	MobileNet	ResNet-50	48.33	71.84	**74.86**$_{\uparrow 3.02}$
Case #2	VGG	MobileNet	ResNet-50	LeNet	ResNet-18	52.55	72.12	**76.24**$_{\uparrow 4.12}$
Case #3	ResNet-50	LeNet	ResNet-18	VGG	MobileNet	54.12	72.45	**77.39**$_{\uparrow 4.94}$

3.7 Effect of Model Architecture.

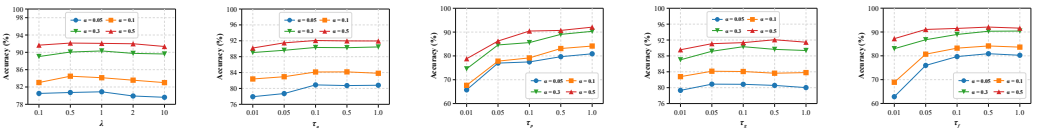
In previous experiments, we used ResNet-18 as the feature extractor by default. To further analyze the impact of different model architectures on OFLBe's performance, we compared OFLBe's performance when using ResNet-18, ResNet-50 and ViT-base as feature extractors, respectively. Specifically, we compared the memory cost and accuracy of different architectures on Tiny-ImageNet ($\alpha = 0.5$). As shown in Table 6, the experimental results indicate that model architecture has a relatively minor impact on OFLBe's performance, but there are significant differences in memory costs. Therefore, the performance improvement is attributable to the designed framework (i.e., OFLBe) rather than the model itself. Furthermore, OFLBe substantially outperforms FAFI.

3.8 Applicability under Resource-Constrained Scenarios

To evaluate the applicability of the OFLBe under resource-constrained scenarios, we further examine its runtime cost under limited computational capacity and storage space. Specifically, we select MobileNet, a representative lightweight model, as the feature extractor, and report its memory cost and computation cost. We use the GPU memory occupied during local training as the metric for memory cost, while computation cost is measured by the time taken per epoch on the GPU (RTX 4090) and the CPU (Intel(R) Xeon(R) Gold 6348 CPU @ 2.60GHz). As shown in Table 7, the experimental results indicate that OFLBe achieves competitive performance even in resource-constrained scenarios.

Table 7: Comparison of computational cost and accuracy among different methods on CIFAR-10 with $\alpha = 0.05$.

Models	Methods	Batch Size	Memory Cost	CPU Time	GPU Time	Accuracy (%)
MobileNet	O-FedAvg	512	1,942 MB	10 s	5 s	10.44
	IntactOFL	512	1,942 MB	10 s	5 s	32.31
	FAFI	512	3,082 MB	22 s	8 s	59.64
	OFLBe (Ours)	512	2,828 MB	13 s	7 s	**71.42**↑11.78
ResNet-18	O-FedAvg	512	4,638 MB	84 s	7 s	12.13
	IntactOFL	512	4,638 MB	85 s	7 s	48.33
	FAFI	512	14,723 MB	214 s	12 s	71.84
	OFLBe (Ours)	512	6,162 MB	91 s	9 s	**80.89**↑9.05



(a) λ (b) τ_a (c) τ_p (d) τ_g (e) τ_f

Figure 4: Effect of different temperature hyper-parameters on accuracy.

3.9 Hyperparameter Analysis

We analyze the impact of λ, τ_a, τ_p, τ_g and τ_f on CIFAR-10. As shown in Figure 4, $\lambda = 1.0$ yields relatively optimal performance. When the anchor-guided feature alignment loss temperature parameter τ_a is less than 0.1, it disrupts training stability, while $\tau_p = 1.0$ achieves the best results for the learnable-prototype-based classification loss. The optimal value of the global prototype aggregation temperature coefficient τ_g increases with the data heterogeneity level α. We set $\tau_g = 0.05$ for $\alpha = \{0.05, 0.1\}$, $\tau_g = 0.1$ for $\alpha = 0.3$ and $\tau_g = 0.5$ for $\alpha = 0.5$. Finally, the global feature fusion temperature parameter $\tau_f = 0.5$ gives the best performance.

4 Conclusion

This paper addresses the model inconsistency problem in OFL and proposes a unified framework driven by shared text anchors. We analyze the major challenges faced by existing methods in one-shot upload scenarios from two perspectives: intra-model inconsistency and inter-model inconsistency. Based on this analysis, we design consistency-enhanced training on the client side and semantic-aware integration on the server side. Specifically, on the client side, anchor-guided alignment on augmented views and learnable prototype-based classification supervision are used to improve the stability of local representations and the compactness of class structures. On the server side, prototype-text similarity-driven global prototype aggregation and static feature fusion are employed to robustly integrate one-shot uploaded knowledge. Experimental results show that the proposed method outperforms existing methods across multiple datasets and heterogeneous settings, validating its effectiveness in mitigating model inconsistency and improving the generalization capability of OFL.

AI USE STATEMENT

In this work, We used generative AI tools solely to polish author-written text and improve the readability and fluency of the manuscript. These tools were not used for any other aspect of this work, including research ideation, methodology development, theoretical analysis, code implementation, data preparation, experimental design or execution, or interpretation of results. The authors reviewed all AI-assisted edits to ensure that they preserved the intended meaning and technical accuracy of the manuscript. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Alan TL Bacellar, Mustafa Munir, Felipe MG França, Priscila Lima, Radu Marculescu, and Lizy K John. Single-round scalable analytic federated learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 39445–39454, 2026.

Daniele Berardini, Vito Paolo Pastore, and Vittorio Murino. Distribution alignment for one-shot federated learning via optimal transport. *arXiv preprint arXiv:2606.16655*, 2026.

Debora Caldarola, Pietro Cagnasso, Barbara Caputo, and Marco Ciccone. Beyond local sharpness: Communication-efficient global sharpness-aware minimization for federated learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 25187–25197, 2025. doi: 10.1109/CVPR52734.2025.02345.

Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3514–3522, 2019.

Haokun Chen, Hang Li, Yao Zhang, Jinhe Bi, Gengyuan Zhang, Yueqi Zhang, Philip Torr, Jindong Gu, Denis Krompass, and Volker Tresp. Fedbip: Heterogeneous one-shot federated learning with personalized latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30440–30450, 2025. doi: 10.1109/CVPR52734.2025.02834.

Guanjie Cheng, Mengzhen Yang, Xinkui Zhao, Shuyi Yu, Tianyu Du, Yangyang Wu, Mengying Zhu, and Shuiguang Deng. Lshfed: Robust and communication-efficient federated learning with locally-sensitive hashing gradient mapping. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 40, pp. 20490–20498, 2026. doi: 10.1609/aaai.v40i25.39184.

Rong Dai, Yonggang Zhang, Ang Li, Tongliang Liu, Xun Yang, and Bo Han. Enhancing one-shot federated learning through data and ensemble co-boosting. *arXiv preprint arXiv:2402.15070*, 2024.

Don Kurian Dennis, Tian Li, and Virginia Smith. Heterogeneity for the win: One-shot federated clustering. In *International Conference on Machine Learning (ICML)*, pp. 2611–2620. PMLR, 2021.

Yiqun Diao, Qinbin Li, and Bingsheng He. Towards addressing label skews in one-shot federated learning. In *International Conference on Learning Representations (ICLR)*, 2023.

Luyang Gai, Shusen Yang, Xuebin Ren, and Zihao Zhou. Biq: Bisection interval quantization for communication-efficient federated learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 40, pp. 21154–21162, 2026. doi: 10.1609/aaai.v40i25.39259.

Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10112–10121, 2022.

Zenghao Guan, Yucan Zhou, and Xiaoyan Gu. Capture global feature statistics for one-shot federated learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, pp. 16942–16950, 2025.

Neel Guha, Ameet Talwalkar, and Virginia Smith. One-shot federated learning. *arXiv preprint arXiv:1902.11175*, 2019.

Run He, Kai Tong, Di Fang, Han Sun, Ziqian Zeng, Haoran Li, Tianyi Chen, and Huiping Zhuang. Afl: A single-round analytic approach for federated learning with pre-trained models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4988–4998, 2025.

Clare Elizabeth Heinbaugh, Emilio Luz-Ricca, and Huajie Shao. Data-free one-shot federated learning under very high statistical heterogeneity. In *International Conference on Learning Representations (ICLR)*, 2023.

Wenke Huang, Mang Ye, Zekun Shi, He Li, and Bo Du. Rethinking federated learning with domain shift: A prototype view. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16312–16322. IEEE, 2023.

Divyansh Jhunjhunwala, Shiqiang Wang, and Gauri Joshi. Fedfisher: Leveraging fisher information for one-shot federated learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1612–1620. PMLR, 2024.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning (ICML)*, volume 139, pp. 4904–4916. PMLR, 2021.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning (ICML)*, volume 119, pp. 5132–5143. PMLR, 2020.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

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

Qinbin Li, Bingsheng He, and Dawn Song. Practical one-shot federated learning for cross-silo setting. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1484–1490, 2021a. doi: 10.24963/ijcai.2021/205.

Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10713–10722, 2021b.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Conference on Machine Learning and Systems (MLSys)*, 2020.

Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

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

Shudong Liu, Hanwen Zhang, Xiuling Wang, Yuesheng Zhu, and Guibo Luo. Feature-aware one-shot federated learning via hierarchical token sequences. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 40, pp. 23819–23827, 2026. doi: 10.1609/aaai.v40i28.39557.

Xiang Liu, Liangxi Liu, Feiyang Ye, Yunheng Shen, Xia Li, Linshan Jiang, and Jialin Li. Fedlpa: One-shot federated learning with layer-wise posterior aggregation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 81510–81548, 2024. doi: 10.52202/079017-2591.

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

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 (AISTATS)*, pp. 1273–1282. PMLR, 2017.

Xutong Mu, Yulong Shen, Ke Cheng, Xueli Geng, Jiaxuan Fu, Tao Zhang, and Zhiwei Zhang. Fedproc: Prototypical contrastive federated learning. *Future Generation Computer Systems*, 143: 93–104, 2023. doi: 10.1016/j.future.2023.01.019.

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 *International Conference on Machine Learning (ICML)*, volume 139, pp. 8748–8763. PMLR, 2021.

Jiangming Shi, Shanshan Zheng, Xiangbo Yin, Yang Lu, Yuan Xie, and Yanyun Qu. Clip-guided federated learning on heterogeneity and long-tailed data. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, pp. 14955–14963, 2024. doi: 10.1609/aaai.v38i13.29416.

Rui Song, Dai Liu, Dave Zhenyu Chen, Andreas Festag, Carsten Trinitis, Martin Schulz, and Alois Knoll. Federated learning via decentralized dataset distillation in resource-constrained edge environments. In *International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10. IEEE, 2023.

Shangchao Su, Bin Li, and Xiangyang Xue. One-shot federated learning without server-side training. *Neural Networks*, 164:203–215, 2023.

Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 36, pp. 8432–8440, 2022.

Jianheng Tang, Yajiang Huang, Kejia Fan, Feijiang Han, Jiaxu Li, Jinfeng Xu, Run He, Anfeng Liu, Houbing Herbert Song, Huiping Zhuang, and Yunhuai Liu. Deepafl: Deep analytic federated learning. In *International Conference on Learning Representations (ICLR)*, 2026.

Zhenheng Tang, Yonggang Zhang, Peijie Dong, Yiu-ming Cheung, Amelie C Zhou, Bo Han, and Xiaowen Chu. Fusefl: One-shot federated learning through the lens of causality with progressive model fusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 28393–28429, 2024. doi: 10.52202/079017-0891.

Fabio Turazza, Marco Picone, and Marco Mamei. The gaussian-head ofl family: One-shot federated learning from client global statistics. In *International Conference on Learning Representations (ICLR)*, 2026.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 20269–20288, 2020.

Naibo Wang, Wenjie Feng, Yuchen Deng, Moming Duan, Fusheng Liu, and See-Kiong Ng. Data-free diversity-based ensemble selection for one-shot federated learning. *Transactions on Machine Learning Research*, 2023.

Xueyu Wu, Xin Yao, and Cho-Li Wang. Fedscr: Structure-based communication reduction for federated learning. *IEEE Transactions on Parallel and Distributed Systems*, 32(7):1565–1577, 2020.

Mingzhao Yang, Shangchao Su, Bin Li, and Xiangyang Xue. Feddeo: Description-enhanced one-shot federated learning with diffusion models. In *ACM International Conference on Multimedia (ACM MM)*, pp. 6666–6675, 2024.

Dezhong Yao, Tongtong Liu, Yuexin Shi, and Zhiqiang Xu. Fedmho: Heterogeneous one-shot federated learning towards resource-constrained clients. In *ACM Web Conference (WWW)*, pp. 5686–5697, 2026. doi: 10.1145/3774904.3792743.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning (ICML)*, pp. 5650–5659. PMLR, 2018.

Hui Zeng, Minrui Xu, Tongqing Zhou, Xinyi Wu, Jiawen Kang, Zhiping Cai, and Dusit Niyato. One-shot-but-not-degraded federated learning. In *ACM International Conference on Multimedia (ACM MM)*, pp. 11070–11079, 2024.

Hui Zeng, Wenke Huang, Tongqing Zhou, Xinyi Wu, Guancheng Wan, Yingwen Chen, and Zhiping Cai. Does one-shot give the best shot? mitigating model inconsistency in one-shot federated learning. In *International Conference on Machine Learning (ICML)*, volume 267, pp. 74080–74097. PMLR, 2025a.

Hui Zeng, Tongqing Zhou, Yeting Guo, Zhiping Cai, and Fang Liu. Towards value-sensitive and poisoning-proof model aggregation for federated learning on heterogeneous data. *Journal of Parallel and Distributed Computing*, 196:104994, 2025b.

Jianqing Zhang, Yang Liu, Yang Hua, and Jian Cao. Fedtgp: Trainable global prototypes with adaptive-margin-enhanced contrastive learning for data and model heterogeneity in federated learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, pp. 16768–16776, 2024. doi: 10.1609/aaai.v38i15.29617.

Jie Zhang, Chen Chen, Bo Li, Lingjuan Lyu, Shuang Wu, Shouhong Ding, Chunhua Shen, and Chao Wu. Dense: Data-free one-shot federated learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 21414–21428, 2022.

Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *International Conference on Machine Learning (ICML)*, pp. 12878–12889. PMLR, 2021.

A Related Work

Federated Learning under Data Heterogeneity. Federated learning (FL) trains a shared model from decentralized data without collecting raw client samples at the server McMahan et al. (2017). In the standard multi-round setting, the server repeatedly aggregates client updates and broadcasts the updated global model. This iterative feedback can partially correct local drift caused by heterogeneous data distribution, but it also introduces substantial communication cost. To improve robustness under non-IID data, existing studies have regularized local optimization, corrected client drift, normalized heterogeneous updates, or introduced representation-level constraints Li et al. (2020); Karimireddy et al. (2020); Wang et al. (2020); Li et al. (2021b); Gao et al. (2022). Recent work has further studied communication-efficient and sharpness-aware optimization for heterogeneous FL, including server-side global sharpness optimization and compact update mappings Caldarola et al. (2025); Gai et al. (2026); Cheng et al. (2026). These methods show that data heterogeneity affects both optimization trajectories and learned representations. However, they usually rely on repeated client-server interaction, which is incompatible with one-shot federated learning where the server receives each client's knowledge only once.

One-Shot Federated Learning. One-shot federated learning (OFL) reduces communication overhead by allowing each client to train locally and communicate with the server only once Guha et al. (2019); Zhang et al. (2022). Existing OFL methods mainly improve the server-side integration stage. Optimization-based methods estimate aggregation weights or model statistics from the uploaded clients Dennis et al. (2021); Su et al. (2023); Jhunjhunwala et al. (2024); Liu et al. (2024). Distillation and data-free synthesis methods train a server-side model using generated or auxiliary representations, thereby transferring knowledge without repeated rounds Lin et al. (2020); Zhu et al. (2021); Zhang et al. (2022); Song et al. (2023). Generative OFL further uses diffusion or description-enhanced synthesis to improve the quality of server-side training signals Yang et al. (2024); Chen et al. (2025). Ensemble and fusion methods retain uploaded local models and combine their predictions or features during inference Wang et al. (2023); Dai et al. (2024); Zeng et al. (2024); Tang et al. (2024). These approaches show that useful global inference is possible under a single communication round. Nevertheless, most of them integrate already trained local models after representation

drift has occurred, and therefore still depend on the implicit assumption that local feature spaces are comparable.

Prototype-Based Federated Learning. Prototype-based federated learning represents each class by compact feature-space statistics instead of exchanging full training model. FedProto introduces class prototypes as communication-efficient knowledge carriers across heterogeneous clients Tan et al. (2022). Subsequent methods use contrastive or trainable prototype objectives to improve class separability and reduce the effect of client-specific distributions Mu et al. (2023); Zhang et al. (2024). Prototype learning has also been used to rethink domain shift in FL, showing that class-level representation structure can expose and mitigate distribution mismatch Huang et al. (2023). These studies are closely related to OFLBe because both use class-level summaries to support federated knowledge transfer. The key difference is that OFLBe does not treat prototypes merely as local summaries for aggregation. Instead, it evaluates and aggregates local prototypes according to their consistency with shared class-level semantic anchors, thereby enabling one-shot integration within a common semantic coordinate system.

Vision-Language Priors and Semantic Anchors. Large-scale vision-language pretraining provides transferable semantic representations that can connect visual features with language-defined class concepts. CLIP, which learns aligned image-text representations from natural language supervision, has become a widely used source of semantic priors for downstream visual recognition Radford et al. (2021). Related vision-language models further show that noisy image-text supervision can still produce broadly transferable representations Jia et al. (2021). In federated settings, such priors have been explored for CLIP adaptation and vision-language knowledge transfer Lu et al. (2023); Shi et al. (2024). OFLBe leverages this line of work in a different manner. Instead of federating a vision-language model itself, it freezes the CLIP text encoder and constructs class-level semantic anchors using class-name prompts before local training. These anchors provide a unified semantic reference for all clients, which helps mitigate intra-client view inconsistency and inter-client class-center drift before the one-shot upload.

B OFLBe Algorithm

Algorithm 1 illustrates the overall procedure of the OFLBe.

C Theoretical Analysis of Consistency Mitigation

This section analyzes the client-side anchor-guided feature alignment used in OFLBe. We show that both intra-client inconsistency and inter-client inconsistency are controlled by the distance from local representations to the shared class-level semantic anchors (CSAs).

C.1 Definitions and Assumptions

Client m holds the local dataset $\mathcal{D}_m = \{(x_m^i, y_m^i)\}_{i=1}^{N_m}$. Let $r_m(x, \mathcal{A}) = \mathcal{N}_m(\Theta_m, \mathcal{A}(x))$ be the feature produced from a random augmentation $\mathcal{A} \sim \mathcal{T}$. Because the training objective uses cosine similarity, we introduce the normalized feature

$$z_m(x, \mathcal{A}) = \frac{r_m(x, \mathcal{A})}{\|r_m(x, \mathcal{A})\|_2}. \tag{17}$$

We assume that the features considered below are nonzero. The CSA a_c of class $c \in \{1, \dots, C\}$ is generated by the frozen CLIP text encoder and satisfies $\|a_c\|_2 = 1$. For nonzero vectors u and v, define

$$d_{\cos}(u, v) = 1 - \cos(u, v). \tag{18}$$

For $(x, y) \in \mathcal{D}_m$, its expected anchor discrepancy is

$$A_m(x) = \mathbb{E}_{\mathcal{A}\sim\mathcal{T}} \left[d_{\cos}\big(z_m(x, \mathcal{A}), a_y\big) \right]. \tag{19}$$

For a class c present at client m, the class-wise discrepancy is

$$A_{m,c} = \mathbb{E}_{(x,y)\sim\mathcal{D}_m|y=c} \mathbb{E}_{\mathcal{A}\sim\mathcal{T}} \left[d_{\cos}\big(z_m(x, \mathcal{A}), a_c\big) \right]. \tag{20}$$

Algorithm 1 OFLBe

Require: M: Number of clients, $\mathcal{D}_m = \{(x_i^m, y_i^m)\}_{i=1}^{N_m}$: local dataset, $\mathcal{N}_m(\Theta_m)$: local model, $\mathcal{C}$: number of classes, C_m: number of classes owned by client m, $\Phi_t(\cdot)$: CLIP text encoder, $\mathcal{A}(\cdot)$: augmentation operator, E: local epochs, η: learning rate, λ: balancing coefficient, temperatures $\tau_a, \tau_p, \tau_g, \tau_f$.
Ensure: prediction $\hat{y}$

Construction of Class-Level Semantic Anchors
for each class $c \in \mathcal{C}$ **do**
  Construct text prompt $P(c) =$ "*a photo of a [class name]*"
  $a_c \leftarrow \Phi_t(P(c))/\|\Phi_t(P(c))\|_2$
end for
Client-Side: Local Semantic Consistency Constraint Training
for each client $m = 1$ to M **in parallel do**
  Initialize local model $\mathcal{N}_m(\Theta_m)$ and learnable prototypes $\{p_m^c\}_{c\in C}$
  for $e = 1$ to E **do**
    Augment local data $\mathcal{D}_m^A \leftarrow \{(\mathcal{A}(x_i^m), y_i^m)\}_{i=1}^{N_m}$
    Extract features $r_m^i \leftarrow \mathcal{N}_m(\Theta_m, \mathcal{A}(x_i^m))$
    Compute Anchor-Guided Feature Alignment Loss $\mathcal{L}_{\mathrm{A}}^{(m)}$ with $\{a_c\}_{c\in\mathcal{C}}$ and temperature τ_a
    Compute Learnable-Prototype-Based Classification Loss $\mathcal{L}_{\mathrm{P}}^{(m)}$ with $\{p_m^c\}_{c\in C}$ and temperature τ_p
    $\mathcal{L}^{(m)} \leftarrow \mathcal{L}_{\mathrm{A}}^{(m)} + \lambda\mathcal{L}_{\mathrm{P}}^{(m)}$
    $\Theta_m^{'} \leftarrow \Theta_m - \eta\nabla_{\Theta_m}\mathcal{L}^{(m)}$
    update $\{p_m^c\}_{c\in C}$
  end for
  Upload local model $\mathcal{N}_m(\Theta_m')$ and learnable prototype set $\{p_m^c\}_{c\in C}$ to the server once
end for
Server-Side: Anchor-Aware Aggregation and Inference
for each class $c \in \mathcal{C}$ **do**
  $\mathcal{S}_c \leftarrow \{m \mid c \in C_m\}$
  for each client $m \in \mathcal{S}_c$ **do**
    $R_m^c \leftarrow \cos(p_m^c, a_c)$
  end for
  For all $m \in \mathcal{S}_c$, set $\alpha_m^c \leftarrow \exp(R_m^c/\tau_g)/\sum_{j\in\mathcal{S}_c}\exp(R_j^c/\tau_g)$
  $p_g^c \leftarrow \sum_{m\in\mathcal{S}_c}\alpha_m^c p_m^c$
end for
for each client $m = 1$ to M **do**
  Compute contribution score $\mathcal{U}_m$ from prototype-anchor consistency
  $\beta_m \leftarrow \exp(\mathcal{U}_m/\tau_f)/\sum_{j=1}^{M}\exp(\mathcal{U}_j/\tau_f)$
end for
For a test sample x, compute $r_f \leftarrow \sum_{m=1}^{M}\beta_m\mathcal{N}_m(\Theta_m', x)$
$\hat{y} \leftarrow \arg\max_{c\in\mathcal{C}}\cos(r_f, p_g^c)$
return $\hat{y}$

Lemma 1 (CSA-mediated cosine-distance bound)**.** *For any nonzero vectors u, v, and a,*

$$d_{\cos}(u, v) \le 2d_{\cos}(u, a) + 2d_{\cos}(v, a). \tag{21}$$

Proof. Let $\bar{u} = u/\|u\|_2$, $\bar{v} = v/\|v\|_2$, and $\bar{a} = a/\|a\|_2$. Then

$$d_{\cos}(u, v) = 1 - \bar{u}^\top\bar{v} = \frac{1}{2}\|\bar{u} - \bar{v}\|_2^2. \tag{22}$$

The triangle inequality and $(p + q)^2 \le 2p^2 + 2q^2$ give

$$\|\bar{u} - \bar{v}\|_2 \le \|\bar{u} - \bar{a}\|_2 + \|\bar{v} - \bar{a}\|_2. \tag{23}$$

Therefore,

$$\begin{aligned} d_{\cos}(u,v) &= \frac{1}{2}\|\bar{u}-\bar{v}\|_2^2 \\ &\le \|\bar{u}-\bar{a}\|_2^2 + \|\bar{v}-\bar{a}\|_2^2 \\ &= 2d_{\cos}(u,a) + 2d_{\cos}(v,a). \end{aligned} \tag{24}$$

□

C.2 Intra-Client Inconsistency Bound

Let $\mathcal{A}_1, \mathcal{A}_2 \overset{\text{i.i.d.}}{\sim} \mathcal{T}$. For $(x,y) \in \mathcal{D}_m$, define the intra-client inconsistency as

$$\Gamma_m(x) = \mathbb{E}_{\mathcal{A}_1,\mathcal{A}_2}\left[d_{\cos}\big(z_m(x,\mathcal{A}_1), z_m(x,\mathcal{A}_2)\big)\right]. \tag{25}$$

Theorem 1 (Intra-client inconsistency bound)**.** *For every $(x,y) \in \mathcal{D}_m$ satisfying the assumptions above,*

$$\Gamma_m(x) \le 4A_m(x). \tag{26}$$

Proof. Applying Lemma 1 with $u = z_m(x,\mathcal{A}_1)$, $v = z_m(x,\mathcal{A}_2)$, and $a = a_y$ gives

$$\begin{aligned} &d_{\cos}\big(z_m(x,\mathcal{A}_1), z_m(x,\mathcal{A}_2)\big) \\ &\le 2d_{\cos}\big(z_m(x,\mathcal{A}_1), a_y\big) + 2d_{\cos}\big(z_m(x,\mathcal{A}_2), a_y\big). \end{aligned} \tag{27}$$

Taking the expectation over the identically distributed augmentations yields $\Gamma_m(x) \le 4A_m(x)$. □

C.3 Inter-Client Inconsistency Bound

For class c, let

$$\mathcal{S}_c = \{m \mid \exists (x,y) \in \mathcal{D}_m,\ y = c\}, \qquad M_c = |\mathcal{S}_c|. \tag{28}$$

This notation matches the client set used for class-wise prototype aggregation in the main paper. Define the class-wise representation center as

$$\mu_{m,c} = \mathbb{E}_{(x,y)\sim\mathcal{D}_m|y=c}\mathbb{E}_{\mathcal{A}\sim\mathcal{T}}\left[z_m(x,\mathcal{A})\right]. \tag{29}$$

We restrict the result to non-degenerate client–class pairs for which $\mu_{m,c} \neq 0$, and set

$$\widehat{\mu}_{m,c} = \frac{\mu_{m,c}}{\|\mu_{m,c}\|_2}. \tag{30}$$

For $M_c \ge 2$, the inter-client inconsistency is the mean pairwise distance

$$\Delta_c = \frac{2}{M_c(M_c-1)} \sum_{\substack{m<n \\ m,n\in\mathcal{S}_c}} d_{\cos}\big(\widehat{\mu}_{m,c}, \widehat{\mu}_{n,c}\big). \tag{31}$$

We set $\Delta_c = 0$ when $M_c < 2$.

Lemma 2 (Class-center anchor discrepancy bound)**.** *For $m \in \mathcal{S}_c$ with $\mu_{m,c} \neq 0$,*

$$d_{\cos}\big(\widehat{\mu}_{m,c}, a_c\big) \le 2A_{m,c}. \tag{32}$$

Proof. Write $\mu = \mu_{m,c}$, $\widehat{\mu} = \mu/\|\mu\|_2$, $a = a_c$, and $A = A_{m,c}$. By linearity of expectation and unit normalization,

$$\mu^\top a = \mathbb{E}_{(x,y)\sim\mathcal{D}_m|y=c}\mathbb{E}_{\mathcal{A}\sim\mathcal{T}}\left[z_m(x,\mathcal{A})^\top a_c\right] = 1 - A. \tag{33}$$

If $A < 1$, then $\mu^\top a > 0$. Moreover, $\|\mu\|_2 \le 1$, so

$$\cos(\mu, a) = \frac{\mu^\top a}{\|\mu\|_2} \ge \mu^\top a = 1 - A. \tag{34}$$

Therefore,

$$d_{\cos}(\widehat{\mu}, a) = 1 - \cos(\mu, a) \le A. \tag{35}$$

If $A \ge 1$, the range of cosine distance gives

$$d_{\cos}(\widehat{\mu}, a) \le 2 \le 2A. \tag{36}$$

Both cases imply the stated bound. □

Theorem 2 (Inter-client inconsistency bound)**.** *For every class c with $M_c \geq 2$ and nonzero class centers,*

$$\Delta_c \leq \frac{8}{M_c} \sum_{m \in \mathcal{S}_c} A_{m,c}. \tag{37}$$

Proof. For $m, n \in \mathcal{S}_c$, Lemma 1 gives

$$d_{\cos}\big(\widehat{\mu}_{m,c}, \widehat{\mu}_{n,c}\big) \leq 2d_{\cos}\big(\widehat{\mu}_{m,c}, a_c\big) + 2d_{\cos}\big(\widehat{\mu}_{n,c}, a_c\big). \tag{38}$$

Let $\delta_{m,c} = d_{\cos}(\widehat{\mu}_{m,c}, a_c)$. Each client occurs in $M_c - 1$ unordered pairs. Thus,

$$\sum_{\substack{m<n \\ m,n \in \mathcal{S}_c}} d_{\cos}\big(\widehat{\mu}_{m,c}, \widehat{\mu}_{n,c}\big) \leq 2(M_c - 1) \sum_{m \in \mathcal{S}_c} \delta_{m,c}. \tag{39}$$

Substitution into the definition of Δ_c yields

$$\Delta_c \leq \frac{4}{M_c} \sum_{m \in \mathcal{S}_c} d_{\cos}\big(\widehat{\mu}_{m,c}, a_c\big). \tag{40}$$

Applying Lemma 2 completes the proof:

$$\Delta_c \leq \frac{8}{M_c} \sum_{m \in \mathcal{S}_c} A_{m,c}. \tag{41}$$

□

C.4 Connection to the Training Objective

The anchor-guided feature alignment loss in the main paper uses all C CSAs. For one normalized feature z with label y, its per-sample form is

$$\ell_{\mathrm{A}}(z, y) = -\log \frac{\exp(\cos(z, a_y)/\tau_a)}{\sum_{c=1}^{C} \exp(\cos(z, a_c)/\tau_a)}. \tag{42}$$

Let $s_c = \cos(z, a_c)$ and define the corresponding softmax probability as

$$q_c = \frac{\exp(s_c/\tau_a)}{\sum_{j=1}^{C} \exp(s_j/\tau_a)}. \tag{43}$$

The logit derivatives are

$$\frac{\partial \ell_{\mathrm{A}}}{\partial s_y} = -\frac{1 - q_y}{\tau_a} \leq 0, \qquad \frac{\partial \ell_{\mathrm{A}}}{\partial s_c} = \frac{q_c}{\tau_a} \geq 0 \quad (c \neq y). \tag{44}$$

Thus, at the logit level, the loss encourages higher similarity to the class-matched CSA and lower similarity to the other CSAs. Because $d_{\cos}(z, a_y) = 1 - s_y$, this direction is consistent with reducing the empirical counterpart of the anchor discrepancy. This derivative argument does not imply that stochastic optimization decreases $A_m(x)$ monotonically.

C.5 Implication and Scope

Theorems 1 and 2 establish

$$\Gamma_m(x) \leq 4A_m(x), \qquad \Delta_c \leq \frac{8}{M_c} \sum_{m \in \mathcal{S}_c} A_{m,c}. \tag{45}$$

If $A_{m,c} \leq \epsilon_c$ for every $m \in \mathcal{S}_c$, then $\Delta_c \leq 8\epsilon_c$. Hence, a shared CSA provides a common reference through which smaller client-side anchor discrepancies imply tighter upper bounds on both forms of representation inconsistency. These bounds explain the role of anchor-guided feature alignment under the stated assumptions.

D EXPERIMENTAL DETAILS

More Implementation Details. Following FAFI Zeng et al. (2025a), We adopt the same experimental settings across all datasets. Specifically, we adopt ResNet-18 as the backbone network, the optimizer is SGD with a momentum coefficient of 0.9, the batch size is set to 256, and the number of local training epochs is set to $E = 200$, the learning rate is set to 0.05. We use the output of the last layer of ResNet-18 as the feature representation, and set the dimensionality of each class prototype to be consistent with the feature dimensionality. In OFL methods, each client performs local training only once. After local training, each client uploads its local model and related statistics to the server once, and the server subsequently performs integration and inference. To further evaluate the communication efficiency gap between the OFLBe and conventional multi-round FL methods, we select six representative multi-round FL methods for comparison, including FedAvg McMahan et al. (2017), FedProx Li et al. (2020), SCAFFOLD Karimireddy et al. (2020), FedCav Zeng et al. (2025b), FedProto Tan et al. (2022) and FedDCGao et al. (2022). To ensure fair comparison, all multi-round FL methods are evaluated under the same experimental conditions as the OFL methods, including the same number of clients, the same heterogeneous data partitioning strategy, and the same local learning-rate setting. We adopt the commonly used setting in the federated learning literature, namely $E = 5$ local epochs per communication round. Furthermore, in the experiments of Table 5 in the main text, LeNet, ResNet-18, VGG, MobileNet, and ResNet-50 are respectively assigned to the five clients, with their classification heads removed. For LeNet, VGG, MobileNet, and ResNet-50, a linear projection layer is added on top of the last-layer outputs to align their output dimensions with that of ResNet-18. And in this paper, the experimental results for MA-Echo Yang et al. (2024), O-FedAvg Guha et al. (2019), FedFisher Jhunjhunwala et al. (2024), FedDF Lin et al. (2020), F-ADI Yin et al. (2018), F-DAFL Chen et al. (2019), DENSE Zhang et al. (2022), Ensemble, Co-Boosting Dai et al. (2024), FuseFL Tang et al. (2024), IntactOFL Zeng et al. (2024) and FAFI Zeng et al. (2025a) are directly taken from the FAFI paper Zeng et al. (2025a). The results for FedCGS Guan et al. (2025), GH-NB_{diag} Turazza et al. (2026), GH-LDA Turazza et al. (2026), GH-QDA_{full} Turazza et al. (2026), FisherMix Turazza et al. (2026) and Proto-Hyper Turazza et al. (2026) are sourced from the GH-OFL paper Turazza et al. (2026), and the results for AFL He et al. (2025), DeepAFL Tang et al. (2026) and SAFLe Bacellar et al. (2026) are obtained from the SAFLe paper Bacellar et al. (2026).

Hardware. The experimental hardware is dual Intel(R) Xeon(R) Gold 5318Y CPU, 2.10 GHz frequency, 1.5Ti RAM, 8 NVIDIA GeForce RTX 4090 GPUs.

Table 8: Effect of the number of local training epochs E on test accuracy across datasets and Dirichlet non-IID settings.

Local Epoch E	CIFAR-10				CIFAR-100				Tiny-ImageNet			
	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$
5	38.10	39.88	52.29	58.92	10.94	11.34	18.52	21.97	8.28	11.50	16.98	20.45
10	47.62	50.27	62.48	69.52	13.81	16.27	26.97	31.43	14.47	18.58	27.21	30.31
50	70.30	74.43	82.96	86.80	34.67	40.69	56.10	60.52	40.11	45.72	54.66	56.79
100	77.51	80.65	88.10	90.58	46.75	52.00	65.43	68.59	48.50	53.00	59.52	60.97
150	79.42	82.10	89.40	91.74	51.86	58.35	68.57	70.93	50.56	54.14	59.47	61.29
200	80.89	84.14	90.32	92.07	55.32	61.57	70.13	71.98	51.81	54.65	60.08	61.44
300	81.79	84.78	91.06	92.45	57.05	61.71	70.59	72.65	51.95	54.89	60.12	61.53

E ADDITIONAL EXPERIMENTAL RESULTS

Effect of the Number of Local Training Epochs. We investigate the impact of different numbers of local training epochs on model performance across the three datasets. As shown in the Table 8, model accuracy generally improves as the number of local training epochs increases. However, after the number of local training epochs reaches 200, the performance gain from further increasing the number of epochs becomes limited. Specifically, increasing the number of local training epochs from 200 to 300 brings only marginal accuracy improvement but significantly increases computational cost. Considering the trade-off between performance improvement and computational cost, we choose 200 local epochs as the default local training setting.