Semantic-Bridge Federated Learning: Bridging CLIP Semantics for Heterogeneous FL

Leixuhai Xu[1*] **Xiangtao Zhang**[1*] **Hailong Yan**[1] **Le Zhang**[1,2 ✉]
[1] University of Electronic Science and Technology of China, Chengdu, China
[2] College of Computer Science and Software Engineering, Hohai University, Nanjing, China
leixuhaixu@163.com, {xiangtaozhang, yanhailong}@std.uestc.edu.cn, lezhang@uestc.edu.cn

Abstract

Federated learning (FL) suffers from severe performance degradation under statistical heterogeneity, where label skew and domain shift induce representation drift and biased decision boundaries. While vision–language models such as CLIP provide transferable semantic priors, raw CLIP embeddings are noisy, highly correlated, and not directly aligned with the latent space of lightweight federated classifiers. We propose **Semantic-Bridge Federated Learning** (SBFL), a framework that transfers pre-trained vision–language semantics into heterogeneous FL without accessing raw client data or requiring online CLIP inference. SBFL constructs text-verified semantic banks, learns a server-side semantic bridge that maps CLIP-derived features into student-compatible teacher spaces, and regularizes local training through mixed semantic prototype alignment and quality-aware semantic replay. Experiments on CIFAR-10, CIFAR-100, Tiny-ImageNet, Office-Caltech 10, and Digit5 show that SBFL consistently improves the average performance of representative FL backbones under both label-skewed and domain-shifted settings. Analysis further shows that the semantic bridge converts highly correlated CLIP priors into more separable teacher-space geometry, explaining its effectiveness in improving representation alignment and global generalization.

1 Introduction

Federated learning (FL) enables multiple clients to collaboratively train a shared model without exposing raw private data [27, 11, 47, 48, 44, 3, 23, 14, 41]. Despite this privacy-preserving training protocol, FL remains highly vulnerable to statistical heterogeneity. In visual recognition tasks, clients may differ in label distributions, visual domains, feature statistics, and class coverage. Such heterogeneity drives local models toward inconsistent objectives, leading to client drift [12], representation drift, biased decision boundaries, and degraded global generalization. This issue becomes particularly severe when some classes are missing or severely underrepresented on local clients, where the global model lacks balanced and reliable class-level supervision.

A large body of work has been proposed to mitigate heterogeneous FL. Client-side methods regularize local optimization, such as FedProx [19], which constrains local updates with a proximal term, and MOON [18], which improves representation consistency through model-contrastive learning. Server-side methods instead refine the aggregation process by adjusting client weights, distribution discrepancies, or update directions [43, 21, 32, 34]. Prototype- and semantics-based methods further introduce class-level guidance to improve cross-client consistency. However, most existing methods still derive their supervisory signals primarily from heterogeneous client data or client-trained representations themselves. As a result, when local class coverage is incomplete or severely imbalanced, the learned class references may still inherit biased semantics from the observed client distributions.

*Equal contribution. ✉ Corresponding author.

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

We visualize learned feature representations using t-SNE [39]. As illustrated in Fig. 1, the representation learned by a standard FL baseline such as FedAvg exhibits mixed class structures and limited separability under heterogeneity, whereas SBFL produces more compact intra-class clusters and clearer inter-class separation. This observation suggests that heterogeneous FL requires not only better optimization or aggregation, but also more reliable class-level semantic guidance that is less tied to biased local observations. Large-scale vision–language models (VLMs), such as CLIP [30], offer a promising external semantic source. Recent CLIP-guided FL methods, such as CLIP2FL [35], and related approaches using federated features, pre-trained representations, or generated semantics [33, 37, 26], suggest that external semantic knowledge can alleviate biased local learning. However, directly injecting CLIP guidance into lightweight FL is not always reliable. Although CLIP maps images and texts into a shared retrieval space, this does not guarantee matched modality distributions or directly usable class-level geometry [22, 42, 38]. Moreover, CLIP representations are sensitive to prompt design and often benefit from task-specific prompts, adapters, or cache-based adaptation [50, 49, 4, 46, 13]. Our prototype analysis further shows that raw CLIP-derived class prototypes can contain strong shared components, which weakens their discriminative value as direct supervision. Thus, directly using CLIP text prototypes, CLIP-derived logits, or raw CLIP embeddings as supervision may impose noisy and geometrically mismatched targets on lightweight federated students.



(a) FedAvg (b) SBFL

Figure 1: t-SNE visualization on CIFAR-10 under label skew with $\beta = 0.1$. Compared with FedAvg, SBFL yields more compact and separable class clusters, indicating improved representation alignment under severe heterogeneity.

These observations suggest that the key challenge is not whether VLM semantics are useful, but how to transform them into reliable and student-compatible teacher signals for heterogeneous FL without online VLM inference or raw data sharing. We answer this question with **Semantic-Bridge Federated Learning** (SBFL). Instead of using raw CLIP embeddings as direct supervision, SBFL constructs text-verified semantic banks to filter unreliable generated features, and learns a server-side semantic bridge that reshapes CLIP-derived priors into student-compatible teacher representation and projection spaces. During local training, clients are regularized by mixed semantic prototype alignment at both feature and projection levels, together with quality-aware semantic replay. CLIP is used only before federated communication to extract cached semantic priors, and the subsequent FL process does not deploy the VLM on clients.

Experiments on label-skewed and domain-shifted FL benchmarks show that SBFL consistently improves representative FL backbones. Further analysis shows that the bridge converts highly correlated CLIP priors into more separable teacher-space geometry, explaining the effectiveness of bridge-based semantic transfer.

Our contributions are summarized as follows:

- We study cached VLM semantics as teacher signals for lightweight heterogeneous FL, motivated by the reliability and geometry mismatch between raw CLIP priors and federated student representations.
- We propose a semantic-bridge framework that verifies generated semantics and adapts CLIP-derived priors into student-compatible teacher spaces without online VLM inference during federated training.
- We introduce dual-level mixed semantic prototype alignment and quality-aware semantic replay, and validate SBFL across label-skewed and domain-shifted FL benchmarks with multiple FL backbones.

2 Related Work

Heterogeneous federated learning. Federated learning trains a shared model from decentralized private data, but statistical heterogeneity often causes client drift, biased local objectives, and degraded global generalization. FedAvg [27] is effective under mild heterogeneity, but can become

unstable when client distributions differ significantly. Client-side methods mitigate this issue by regularizing or correcting local updates, including FedProx [19], SCAFFOLD [12], FedNova [40], and FedDyn [1]. MOON [18] improves representation consistency through model-contrastive learning, while FedNTD [16] reduces global knowledge forgetting during local training through non-true distillation. Server-side methods refine optimization or aggregation by considering adaptive updates, client discrepancy, update direction, or layer-wise divergence [31, 43, 21, 32, 34]. For feature or domain shift, FedBN [20] reduces client-specific feature discrepancy by preserving local batch-normalization statistics. These methods mainly improve optimization, aggregation, or normalization, whereas our work introduces external class-level semantic guidance to improve local representation learning.

Knowledge transfer and prototype learning in FL. Knowledge transfer has been widely used to improve FL under heterogeneous data. Federated distillation transfers global or ensemble knowledge to local models through predictions or intermediate representations [7, 17, 24, 52], while contrastive learning aligns local representations with global or historical references [18]. Prototype-based FL provides a compact way to share class-level information across clients. FedProto [36] uses class prototypes as semantic anchors, and later studies extend prototype learning to domain shift and non-IID data [10, 28, 45]. Recent semantic-anchor methods further decouple prototype construction from local representation learning to improve cross-client representation consistency [51]. Classifier-calibration methods such as CCVR further show that auxiliary feature-level representations can correct biased classifiers under non-IID data [25]. However, these teachers, prototypes, or semantic anchors are usually estimated from client-trained models, local representations, or federated statistics, and may inherit biased semantics when local class coverage is incomplete. SBFL instead derives semantic guidance from external VLM priors and transforms it into student-compatible teacher spaces.

Generative and vision–language priors for FL. Recent studies have explored generative priors, synthetic features, pre-trained representations, and foundation-model semantics to provide external guidance for FL [52, 26, 33, 37, 35]. For example, FedGen [52] performs data-free knowledge distillation, CReFF [33] retrains classifiers with federated features, GGEUR [26] synthesizes semantic features for distribution alignment, FedPCL [37] uses pre-trained models with prototype-wise contrastive learning, and CLIP-guided FL [35] leverages vision–language priors under heterogeneous or long-tailed data. In vision–language learning, prompt-based and lightweight adaptation methods show that raw CLIP representations often benefit from task-specific prompts, feature adapters, cache-based adaptation, or multi-modal prompt tuning [50, 49, 4, 46, 13]. These works suggest that external semantics can alleviate biased local learning. However, directly using generated features, CLIP outputs, or raw CLIP prototypes may be noisy, highly correlated, or geometrically mismatched with lightweight federated students. Instead of treating VLM semantics as direct supervision, SBFL verifies and bridges them before local training, using text-verified semantic banks, a server-side semantic bridge, dual-level semantic alignment, and quality-aware replay without online CLIP inference.

3 Methodology

In this section, we introduce **Semantic-Bridge Federated Learning** (SBFL). We first formulate the heterogeneous FL setting and the construction of cached semantic priors. Then, we describe three key components of SBFL: text-verified semantic banks, server-side semantic bridging, and local semantic alignment with replay. The overall framework is shown in Fig. 2.

3.1 Problem Setup and Semantic Priors

We consider a federated system with K clients. Client k owns a private dataset $\mathcal{D}_k = \{(x_{k,i}, y_{k,i})\}_{i=1}^{n_k}$ sampled from a client-specific distribution. The student model is decomposed as

$$f_S(x;\theta) = C_S(P_S(E_S(x))), \tag{1}$$

where E_S, P_S, and C_S denote the encoder, projector, and classifier, respectively. Given a local mini-batch $\mathcal{B}_k = \{(x_i, y_i)\}_{i=1}^{B}$, the supervised objective is

$$\mathcal{L}_{\text{task}}^{(k)} = \frac{1}{B}\sum_{i=1}^{B} \ell_{\text{ce}}(f_S(x_i;\theta), y_i). \tag{2}$$

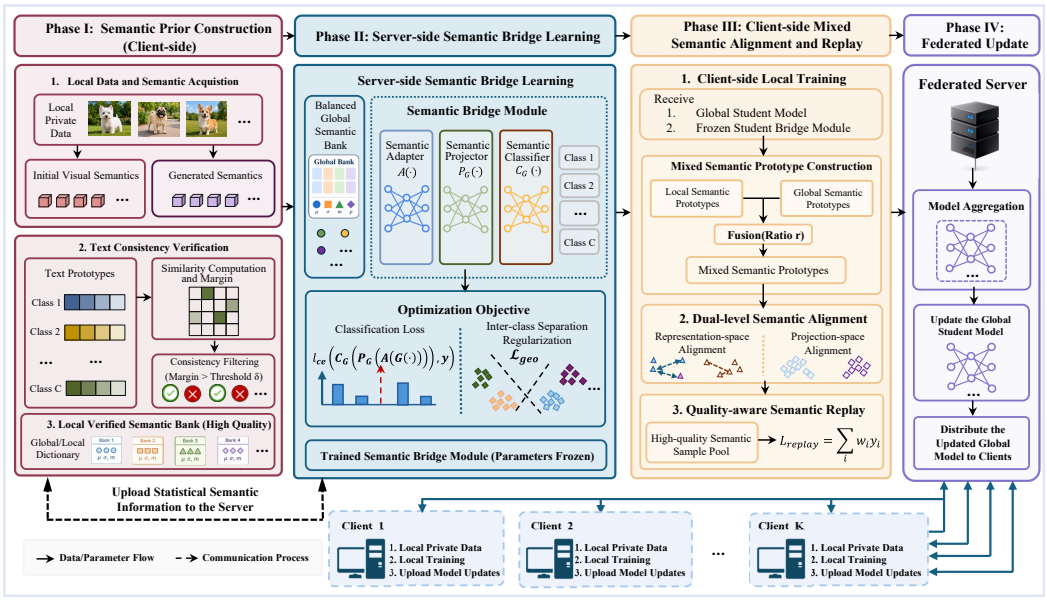


Figure 2: Overview of SBFL. Clients construct cached CLIP-derived semantic priors without uploading raw images, locally verify generated semantic candidates with CLIP text prototypes, and upload the feature–label pairs in the final local semantic bank. The server builds a balanced global semantic bank, learns a semantic bridge to map CLIP priors into student-compatible teacher spaces, and sends bridged guidance for mixed prototype alignment and quality-aware replay.

To provide class-level semantic guidance beyond heterogeneous local data, each client constructs a raw semantic pool

$$\mathcal{G}_k^{\text{raw}} = \mathcal{G}_k^{\text{init}} \cup \mathcal{G}_k^{\text{gen}}, \tag{3}$$

where $\mathcal{G}_k^{\text{init}}$ contains locally extracted CLIP image embeddings and $\mathcal{G}_k^{\text{gen}}$ contains generated semantic features from an existing semantic generation pipeline [26]. We do not modify the generation process; our goal is to verify, transform, and use these cached semantic priors as teacher signals. Raw client images are never uploaded. The proposed SBFL is agnostic to the specific semantic generator: the generator only provides candidate semantic features, while our contribution lies in verifying, bridging, and using these priors as student-compatible teacher signals.

3.2 Text-Verified Semantic Bank

Generated semantic features may contain incorrect class assignments, which can introduce noisy teacher signals.

To reduce such noise, each client locally verifies generated features with CLIP text prototypes before constructing the semantic bank shared with the server. For a generated pair $(g, y) \in \mathcal{G}_k^{\text{gen}}$, we define its text-consistency margin as

$$m(g, y) = \text{sim}(\bar{g}, t_y) - \max_{c \neq y} \text{sim}(\bar{g}, t_c), \tag{4}$$

where $\bar{g} = g/\|g\|_2$. A generated feature is retained only when its CLIP-text prediction agrees with its assigned label and the margin exceeds a threshold:

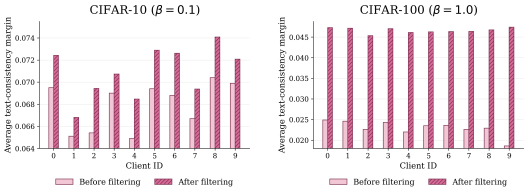


Figure 3: Effect of text verification. Retained generated semantic samples have higher text-consistency margins, indicating cleaner semantic priors.

$$(g, y) \in \mathcal{G}_k^{\text{keep}} \iff \arg\max_c \text{sim}(\bar{g}, t_c) = y \text{ and } m(g, y) \geq \delta. \tag{5}$$

The final local semantic bank is

$$\bar{\mathcal{G}}_k = \mathcal{G}_k^{\text{init}} \cup \mathcal{G}_k^{\text{keep}}. \tag{6}$$

Thus, text verification is applied only to the generated semantic candidates, while the original CLIP image embeddings in $\mathcal{G}_k^{\text{init}}$ are retained. Before federated training, each client uploads the feature–label pairs in $\bar{\mathcal{G}}_k$ to the server once, while raw client images are never transmitted.

Fig. 3 provides a compact visualization of this filtering effect: after text verification, retained generated semantic features exhibit larger text-consistency margins, suggesting that unreliable generated priors are removed before bridge learning and replay. To avoid domination by large or easier clients, the server further constructs a balanced global bank through equalized client sampling:

$$\bar{\mathcal{G}}^{\text{glob}} = \bigcup_{k=1}^{K} \text{Sample}(\bar{\mathcal{G}}_k, M), \qquad M = \min_{u \in \mathcal{K}} |\bar{\mathcal{G}}_u|. \tag{7}$$

3.3 Server-side Semantic Bridge

Raw CLIP embeddings are not naturally aligned with the feature spaces of lightweight federated students. Moreover, raw CLIP prototypes can contain strong shared components, which weakens their discriminative ability as direct supervision. We therefore introduce a server-side semantic bridge consisting of an adapter $A(\cdot;\phi)$, a projector $P_G(\cdot)$, and a classifier $C_G(\cdot)$:

$$h^T = A(g), \qquad p^T = P_G(h^T). \tag{8}$$

The bridge is optimized on $\bar{\mathcal{G}}^{\text{glob}}$ with a classification objective:

$$\mathcal{L}_{\text{cls}} = \mathbb{E}_{(g,y)\sim\bar{\mathcal{G}}^{\text{glob}}}\big[\ell_{\text{ce}}(C_G(P_G(A(g))), y)\big]. \tag{9}$$

To preserve useful CLIP semantic neighborhoods while improving class separability, we add a geometry regularizer. Before computing pairwise similarities, class prototypes are mean-centered and ℓ_2-normalized to remove the dominant shared semantic component. Let s_{ij}^{clip} be the cosine similarity between centered CLIP class prototypes, and let s_{ij}^{q} be the similarity between centered bridged prototypes in space $q \in \{h, p\}$. For $\Omega = \{(i,j) \mid i \neq j\}$ and $w_{ij} = 1 - s_{ij}^{\text{clip}}$, we define

$$\mathcal{L}_{\text{geo}}^{q} = \frac{1}{2|\Omega|} \sum_{(i,j)\in\Omega} (s_{ij}^{q} - s_{ij}^{\text{clip}})^2 + \frac{\sum_{(i,j)\in\Omega} w_{ij} (s_{ij}^{q})^2}{2\sum_{(i,j)\in\Omega} w_{ij}}. \tag{10}$$

The first term preserves CLIP's relative semantic structure, while the second term discourages high similarity between semantically distant classes. The server objective is

$$\mathcal{L}_{\text{server}} = \mathcal{L}_{\text{cls}} + \lambda_{\text{geo}}(\mathcal{L}_{\text{geo}}^{h} + \mathcal{L}_{\text{geo}}^{p}). \tag{11}$$

At the beginning of each communication round, the server warm-starts and updates the semantic bridge on the balanced global bank, then freezes it for client-side training. Thus, the bridge receives no gradients during local updates in the current round, but can be refreshed by the server in the next round to remain compatible with the evolving global student.

3.4 Mixed Semantic Prototype Alignment

The bridged semantic features provide class-level teacher signals in spaces compatible with the student model. For $q \in \{h, p\}$, define the bridge mapping $\Gamma_h(g) = A(g)$ and $\Gamma_p(g) = P_G(A(g))$. For class c, we compute local and global teacher-space prototypes:

$$q_{k,c}^{L} = \text{norm}\big(\tfrac{1}{|\bar{\mathcal{G}}_{k,c}|} \textstyle\sum_{(g,y)\in\bar{\mathcal{G}}_{k,c}} \Gamma_q(g)\big), \qquad q_c^{G} = \text{norm}\big(\tfrac{1}{|\bar{\mathcal{G}}_c^{\text{glob}}|} \textstyle\sum_{(g,y)\in\bar{\mathcal{G}}_c^{\text{glob}}} \Gamma_q(g)\big). \tag{12}$$

We combine local specificity and global completeness by

$$q_{k,c}^{M} = \text{norm}\big((1-r)q_{k,c}^{L} + r q_c^{G}\big), \tag{13}$$

where the global prototype is used as fallback when class c is absent from the local bank.

Given a student feature q_i in either representation or projection space, the semantic alignment loss is

$$\mathcal{L}_{\text{align}}^{q} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(\text{sim}(\bar{q}_i, q_{k,y_i}^{M})/\tau)}{\sum_{c\in\mathcal{C}_k^{v}} \exp(\text{sim}(\bar{q}_i, q_{k,c}^{M})/\tau)}, \tag{14}$$

where $\mathcal{C}_k^{v}$ denotes the valid prototype classes. The losses in representation and projection spaces are denoted as $\mathcal{L}_{\text{rep}}$ and $\mathcal{L}_{\text{proj}}$, respectively.

3.5 Quality-Aware Semantic Replay

Besides aligning image features with semantic prototypes, each client further replays verified generated semantics from $\mathcal{G}_k^{\text{keep}}$ to regularize its local projector and classifier. Given a replay batch

$\{(\tilde{g}_j, \tilde{c}_j)\}_{j=1}^{B_r}$, each sample is weighted by its text-consistency margin:

$$w_j = \frac{\exp(m(\tilde{g}_j, \tilde{c}_j)/\tau_r)}{\sum_{u=1}^{B_r} \exp(m(\tilde{g}_u, \tilde{c}_u)/\tau_r)}. \tag{15}$$

Let $\tilde{h}_j = \mathrm{sg}(A_t(\tilde{g}_j))$ denote the frozen bridged semantic feature in round t, where $\mathrm{sg}(\cdot)$ denotes stop-gradient. The replay prediction is produced by the local student projector and classifier:

$$\tilde{s}_{k,j} = C_S^{(k)}\left(P_S^{(k)}(\tilde{h}_j)\right). \tag{16}$$

The replay loss is

$$\mathcal{L}_{\text{replay}}^{(k)} = \sum_{j=1}^{B_r} w_j \ell_{\text{ce}}(\tilde{s}_{k,j}, \tilde{c}_j). \tag{17}$$

Since $\tilde{h}_j$ is stop-gradient, $\mathcal{L}_{\text{replay}}^{(k)}$ updates only the local student projector and classifier, while the semantic bridge remains frozen during client training. The final local objective of client k in round t is

$$\mathcal{L}_{\text{total}}^{(k)} = \mathcal{L}_{\text{task}}^{(k)} + \lambda_{\text{dist}}\left(\mathcal{L}_{\text{rep}}^{(k)} + \mathcal{L}_{\text{proj}}^{(k)}\right) + \lambda_{\text{replay}}\mathcal{L}_{\text{replay}}^{(k)} + \lambda_b \mathcal{L}_{\text{backbone}}^{(k)}. \tag{18}$$

Here, $\mathcal{L}_{\text{task}}^{(k)}$ is the supervised cross-entropy loss on private images, $\mathcal{L}_{\text{rep}}^{(k)}$ and $\mathcal{L}_{\text{proj}}^{(k)}$ are the semantic alignment losses in the representation and projection spaces, and $\mathcal{L}_{\text{replay}}^{(k)}$ is the quality-aware replay loss. $\mathcal{L}_{\text{backbone}}^{(k)}$ denotes the backbone-specific local regularization term, such as the proximal loss in FedProx or the model-contrastive loss in MOON. For FedAvg and purely server-side aggregation methods, $\mathcal{L}_{\text{backbone}}^{(k)} = 0$.

3.6 Discussion

Why does semantic bridging help? SBFL improves heterogeneous FL by making external VLM semantics more reliable and more compatible with lightweight students. Raw CLIP-derived prototypes provide useful class-level priors, but they can be noisy, highly correlated, and geometrically mismatched with federated student representations. This mismatch becomes more harmful under label skew and domain shift, where local clients often observe biased or incomplete class evidence. Text verification filters unreliable generated semantics, while the semantic bridge reshapes verified CLIP-derived priors into more separable student-compatible teacher spaces. This explains why bridge-based transfer can be more effective than directly injecting raw CLIP guidance.

Relation to existing FL mechanisms. SBFL is designed to be complementary to existing FL improvements. Many heterogeneous FL methods mitigate client drift by refining server-side aggregation or adding local regularization terms, whereas SBFL focuses on improving the semantic supervision available during local training. Thus, SBFL addresses a different source of heterogeneity: the lack of reliable and less client-biased class-level references. The plug-and-play results show that SBFL can be combined with both aggregation-based and local-objective-based FL backbones, suggesting its flexibility as a semantic regularization module.

Complexity, privacy, and limitations. CLIP is used only before FL communication to extract cached semantic priors, and no online VLM inference is required during federated training. Thus, SBFL preserves the same dominant per-round communication order as standard FL, while adding computation for bridge training, semantic alignment, and replay. Detailed complexity, storage, and privacy-boundary discussions are provided in Appendix A.18, Appendix A.19, and Appendix A.8. A limitation is that SBFL depends on class-name prompts and CLIP coverage; when prompts are ambiguous or domains are highly specialized, semantic guidance may become weaker.

3.7 Training Procedure

Algorithm 1 summarizes SBFL. Semantic banks are constructed before FL communication. At each round, the server updates and freezes the semantic bridge, while clients train student models with task supervision, mixed semantic prototype alignment, and quality-aware replay. The uploaded student models are then aggregated using the selected FL backbone.

Algorithm 1 SBFL training procedure

Require: Client datasets $\{\mathcal{D}_k\}_{k=1}^{K}$; initial student model $\theta^{(0)}$; CLIP model; communication rounds T
Ensure: Final global student model $\theta^{(T)}$

Offline semantic-bank construction:
for each client $k = 1, \ldots, K$ **do**
  Extract local CLIP image embeddings $\mathcal{G}_k^{\text{init}}$ and generated semantics $\mathcal{G}_k^{\text{gen}}$.
  Verify generated semantic candidates with CLIP text prototypes to obtain $\mathcal{G}_k^{\text{keep}}$, and form $\bar{\mathcal{G}}_k = \mathcal{G}_k^{\text{init}} \cup \mathcal{G}_k^{\text{keep}}$.
end for
Server collects verified candidates and builds the balanced global bank $\bar{\mathcal{G}}^{\text{glob}}$.
for round $t = 0, \ldots, T-1$ **do**
  Server updates semantic bridge $\Psi_t = \{A_t, P_{G,t}, C_{G,t}\}$ on $\bar{\mathcal{G}}^{\text{glob}}$ using $\mathcal{L}_{\text{server}}$.
  Freeze Ψ_t, construct global teacher prototypes $\{q_c^G\}_{c=1}^{C}$, and broadcast $\theta^{(t)}$ with bridge-derived teacher information.
  Sample participating clients S_t.
  for all client $k \in S_t$ **in parallel do**
    Initialize $\theta_k \leftarrow \theta^{(t)}$ and build local prototypes $\{q_{k,c}^L\}$ from $\bar{\mathcal{G}}_k$ through frozen Ψ_t.
    Construct mixed prototypes $\{q_{k,c}^M\}$ from $\{q_{k,c}^L\}$ and $\{q_c^G\}$.
    for each local mini-batch **do**
      Compute $\mathcal{L}_{\text{task}}^{(k)}$, semantic alignment losses $\mathcal{L}_{\text{rep}}^{(k)}$, $\mathcal{L}_{\text{proj}}^{(k)}$, and replay loss $\mathcal{L}_{\text{replay}}^{(k)}$.
      Update local student parameters θ_k by minimizing $\mathcal{L}_{\text{total}}^{(k)}$.
    end for
    Upload θ_k to the server.
  end for
  Server aggregates local models with the selected FL backbone to obtain $\theta^{(t+1)}$.
end for
return $\theta^{(T)}$

4 Experiments

In this section, we evaluate SBFL under label-skewed and domain-shifted FL settings. We first compare SBFL with representative FL baselines, then examine its plug-and-play behavior with different backbones. Finally, we analyze the contribution of each component and the effect of semantic bridging.

4.1 Experimental Setup

Datasets. We evaluate label-skewed FL on CIFAR-10, CIFAR-100 [15], and Tiny-ImageNet [2]. Following common heterogeneous FL protocols, we partition training data across 10 clients using a Dirichlet distribution with $\beta \in \{0.1, 0.5, 1.0\}$, where a smaller β indicates stronger label skew. We further evaluate domain-shifted FL on Office-Caltech 10 [5] and Digit5 [29], where each domain is treated as one client. We report Top-1 accuracy. For domain-shifted datasets, C.Avg denotes the arithmetic mean over domains. We fix the random seed to ensure reproducibility.

Baselines. We compare with representative heterogeneous FL methods, including FedAvg [27], FedProx [19], MOON [18], FedMut [8], FedMR [9], and FedAWA [34]. For fair comparison, we follow the official implementations and default hyperparameters of each baseline, while keeping data partitions, communication rounds, local epochs, optimizer, learning rate, and backbones consistent unless otherwise specified. All comparisons are conducted under the same online client-side training assumption.

Configurations. For label-skewed FL, we evaluate both CNN and ResNet-50 [6] on CIFAR-10, CIFAR-100, and Tiny-ImageNet. For domain-shifted FL, we use ResNet-50 on Office-Caltech 10 and CNN on Digit5. We train CNN backbones for 100 communication rounds on all label-skewed datasets. For ResNet-50, we train for 100 rounds on CIFAR-10 and CIFAR-100, and 20 rounds on Tiny-ImageNet. For domain-shifted datasets, we train for 30 communication rounds. The batch size is 64, except for Tiny-ImageNet where we use 16. projection dimension is 256, and the optimizer is SGD with momentum 0.9, weight decay 10^{-5}, and learning rate 0.01. We set $\lambda_{\text{dist}} = 2.0$ for CIFAR-10/100 and Office-Caltech 10, $\lambda_{\text{dist}} = 0.1$ for Tiny-ImageNet and Digit5, $\lambda_{\text{replay}} = 0.5$, $r = 0$ for label skew, and $r = 0.4$ for domain shift. Detailed hyperparameters for semantic-bank construction, CLIP prompts, server bridge training, and geometry regularization are provided in Appendix A.6 and Appendix A.7.

Table 4: Performance under domain shift on Office-Caltech 10 and Digit5. C.Avg denotes the arithmetic mean over domains.

Method	Office-Caltech 10						Digit5						
	Amaz	Calt	DSLR	Webc	C.Avg	Gain	SYN	USPS	SVHN	MNTM	MNIST	C.Avg	Gain
FedAvg [27]	83.33	73.78	71.88	66.10	73.77	+0.00	99.52	93.06	89.57	77.33	97.48	91.39	+0.00
FedMut [8]	85.42	74.22	81.25	74.58	78.87	+5.10	98.71	96.72	89.52	88.43	98.76	94.43	+3.04
FedProx [19]	86.98	72.89	68.75	72.88	75.37	+1.60	99.11	93.23	87.37	75.81	96.95	90.49	-0.90
FedMR [9]	80.73	74.22	78.12	74.58	76.91	+3.14	97.91	96.61	86.15	79.45	97.92	91.61	+0.22
MOON [18]	84.90	76.42	68.75	64.41	73.62	-0.15	99.39	93.98	89.74	78.13	97.59	91.76	+0.37
FedAWA [34]	86.98	76.00	75.00	61.02	74.75	+0.98	99.53	92.31	89.54	75.95	97.14	90.89	-0.50
SBFL	85.42	70.67	87.50	84.75	**82.08**	**+8.31**	98.92	97.42	89.95	90.18	98.67	**95.03**	**+3.64**

4.2 Performance Overview

Table 1: Performance under label-skewed non-IID settings. Best results are in bold and second-best results are underlined.

Method	CIFAR-10						CIFAR-100						Tiny-ImageNet					
	CNN			ResNet50			CNN			ResNet50			CNN			ResNet50		
	0.1	0.5	1.0	0.1	0.5	1.0	0.1	0.5	1.0	0.1	0.5	1.0	0.1	0.5	1.0	0.1	0.5	1.0
FedAvg [27]	79.02	85.85	87.98	82.39	91.08	91.04	46.50	47.78	47.97	62.41	64.84	65.44	28.48	32.59	33.97	21.18	24.39	24.99
FedProx [19]	78.67	86.17	87.21	83.50	90.79	91.83	47.36	48.83	48.33	62.89	64.35	65.41	27.95	33.31	34.23	20.75	24.21	25.35
MOON [18]	75.85	86.55	87.91	83.85	91.61	92.55	48.29	49.45	49.78	63.67	67.10	67.30	28.35	33.91	34.72	21.11	23.09	23.81
FedMut [8]	81.12	85.16	86.63	86.41	91.67	92.35	46.39	47.43	47.78	62.67	66.35	67.76	30.30	33.17	33.21	21.59	24.26	24.16
FedMR [9]	80.25	86.11	87.85	86.21	91.56	92.27	45.27	48.51	49.87	62.35	65.75	67.13	28.16	**34.90**	35.05	16.69	19.94	20.50
FedAWA [34]	79.97	85.70	87.22	83.16	91.62	92.26	47.24	48.88	49.08	62.56	65.25	65.38	28.93	33.26	34.35	21.15	23.29	24.93
SBFL	**82.55**	**87.09**	**88.57**	**87.86**	**92.09**	**93.01**	**50.26**	**57.26**	**59.25**	**68.56**	**72.13**	**73.86**	**31.73**	34.08	**35.18**	**22.26**	**27.89**	**30.31**

Label-skewed FL. Table 1 shows that SBFL achieves the best performance in 17 out of 18 label-skewed settings. On average, it improves FedAvg by 4.21 points and the strongest baseline in each setting by 2.76 points. The gains are most pronounced on CIFAR-100, where SBFL improves FedAvg by 7.73 points on average across CNN and ResNet-50, indicating that semantic bridging is especially useful when local class supervision is sparse or biased. Appendix A.11 further compares SBFL with a CLIP2FL-adapted direct CLIP-guided baseline [35], which includes client-side CLIP distillation and server-side CLIP-guided classifier retraining under the same FL settings. SBFL consistently outperforms this baseline on CIFAR-10 and CIFAR-100, suggesting that verification and semantic bridging provide benefits beyond directly injecting CLIP guidance.

Table 2: Plug-and-play gains of SBFL on CIFAR-100.

Backbone	$\beta = 0.1$	$\beta = 0.5$	$\beta = 1.0$	Avg.
FedAvg [27]	+6.15	+7.29	+8.42	7.29
FedProx [19]	+5.90	+8.50	+7.23	7.21
MOON [18]	+0.91	+6.13	+6.01	4.35
FedMut [8]	+4.91	+7.02	+3.78	5.24
FedMR [9]	+7.02	+7.23	+5.81	6.69
FedAWA [34]	+6.33	+6.76	+6.72	6.60
Avg.	**+5.20**	**+7.15**	**+6.32**	**6.22**

Domain-shifted FL. Table 4 shows that SBFL achieves the best average accuracy on both Office-Caltech 10 and Digit5, improving FedAvg by 8.31 and 3.64 points, respectively. Additional comparisons with FedBN and FedHEAL are provided in Appendix A.10; SBFL remains the best method on average, outperforming the strongest baseline by 6.33 points on Office-Caltech 10 and 0.26 points on Digit5. Although SBFL does not achieve the best accuracy on every individual domain, it substantially improves DSLR and Webcam and achieves the best overall average. We further analyze the domain-specific trade-off and the effect of the prototype mixing ratio r in Appendix A.15 and Appendix A.16.

Table 3: Plug-and-play evaluation of SBFL.

Method	Amaz	Calt	DSLR	Webc	Avg.
FedAvg [27]	83.33	73.78	71.88	66.10	73.77
FedAvg+SBFL	85.42	70.67	87.50	84.75	**82.08**
FedProx [19]	86.98	72.89	68.75	72.88	75.37
FedProx+SBFL	85.42	71.56	93.75	83.05	**83.44**
FedAWA [34]	86.98	76.00	75.00	61.02	74.75
FedAWA+SBFL	89.58	74.22	78.12	66.10	**77.01**
Avg. Gain	**+1.04**	-2.07	**+14.58**	**+11.30**	**+6.21**

Plug-and-play ability. Tables 2 and 3 show that SBFL provides consistent average gains when inserted into different FL backbones. It improves CIFAR-100 by 6.22 points on average over 18 backbone–heterogeneity combinations and Office-Caltech 10 by 6.21 points across three representative backbones, suggesting that semantic bridging is complementary to local regularization, representation consistency, and aggregation refinement.

4.3 Ablation and Analysis

We next analyze where the improvement comes from by examining module contributions, bridged prototype geometry, and training robustness.

Ablation study. Table 5 reports a dependency-aware ablation on CIFAR-10 under $\beta = 0.1$, where **A**, **D**, **R**, and **T** denote the semantic adapter, dual-level semantic alignment, semantic replay, and text verification, respectively. The adapter-only variant improves FedAvg by 1.43 points, showing that external semantic priors remain useful after being transferred into the student model. Adding dual-level semantic alignment gives the largest additional gain, increasing the accuracy from 80.45 to 81.81, indicating that semantic alignment is the main source of improvement. Text verification further improves both the adapter-only and alignment variants, while semantic replay raises the full model to 82.55, yielding a total gain of 3.53 points over FedAvg. These results show that the bridge, text verification, semantic alignment, and replay provide complementary benefits. A further matched-resource ablation in Appendix A.12 keeps the semantic bank fixed and progressively introduces the bridge and replay, providing additional evidence for their respective contributions.

Prototype geometry. Table 6 examines whether the semantic bridge improves CLIP-derived prototype geometry. Raw CLIP prototypes show high off-diagonal cosine similarity and many highly similar nearest neighbors, which can provide weak discriminative supervision for small federated classifiers. After bridging, the off-diagonal cosine similarity decreases from 0.6638 to 0.4082 in the representation space and 0.3772 in the projection space. The percentage of highly similar top neighbors also drops from 35.50% to 9.50% and 4.00%, respectively. Meanwhile, the bridged spaces maintain high relation correlation with the original CLIP semantic neighborhoods. This indicates that the semantic bridge does not simply discard CLIP structure; instead, it preserves useful semantic relations while increasing class separability in student-compatible spaces.

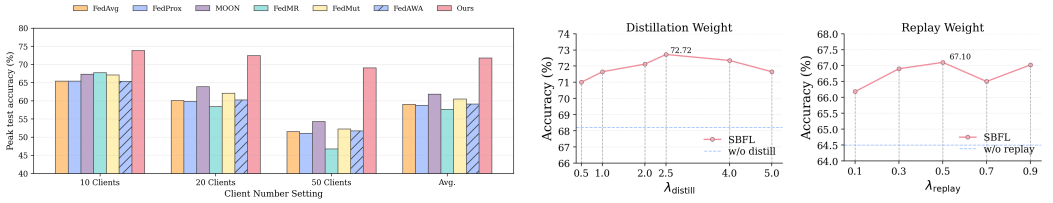
Table 5: Dependency-aware ablation on CIFAR-10 with $\beta = 0.1$.

A	D	R	T	Acc.	Gain
✗	✗	✗	✗	79.02	–
✓	✗	✗	✗	80.45	+1.43
✓	✗	✗	✓	80.92	+1.90
✓	✓	✗	✗	81.81	+2.79
✓	✓	✗	✓	82.03	+3.01
✓	✓	✓	✗	82.22	+3.20
✓	✓	✓	✓	**82.55**	**+3.53**

Robustness and sensitivity. Figure 4 summarizes robustness to client number and sensitivity to semantic loss weights on CIFAR-100. As clients increase from 10 to 50 under $\beta = 1.0$, local data become more fragmented, yet SBFL consistently performs best, indicating stable class-level guidance under sparse supervision. The sensitivity curves show that SBFL is robust within a reasonable range of λ_{dist} and λ_{replay}, with moderate values best balancing supervised learning, semantic alignment, and replay regularization.

Table 6: Prototype geometry analysis on Tiny-ImageNet.

Space	Off-Cos↓	Pair> .9 ↓	Top> .9 ↓	Rank↑	Corr.↑
Raw CLIP	0.6638	0.34%	35.50%	111.85	–
Bridge-h	0.4082	0.05%	9.50%	96.96	0.8703
Bridge-p	0.3772	0.02%	4.00%	32.50	0.7836



(a) Robustness to the number of clients. **(b)** Sensitivity to semantic loss weights.

Figure 4: Robustness and sensitivity analysis on CIFAR-100.

Convergence. Figure 5 compares convergence curves on CIFAR-10 and CIFAR-100. SBFL reaches higher accuracy than representative baselines throughout training and converges to better final solutions. The improvement is more visible on CIFAR-100, where the larger label space makes local training more vulnerable to biased supervision. This supports the role of semantic bridging as stable class-level guidance across communication rounds, improving both final performance and training stability.

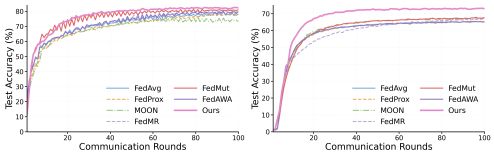


Figure 5: Convergence comparison on CIFAR-10 and CIFAR-100.

4.4 Empirical Privacy Analysis

Although raw client images are never transmitted, the uploaded semantic banks contain CLIP-space feature–label pairs and may still expose information about local data. We therefore evaluate two

Table 7: Client label-distribution inference from uploaded semantic-bank labels. Results are reported as mean ± standard deviation across clients.

Shared information	β	Corr.↓	TV distance↑	Majority acc.↓	Top-5 overlap↓
$\mathcal{G}_k^{\text{init}}$ only	0.1	1.0000 ± 0.0000	0.0000 ± 0.0000	100.00%	100.00%
Final SBFL bank	0.1	0.0000 ± 0.0000	0.7133 ± 0.0305	0.00%	2.00%
$\mathcal{G}_k^{\text{init}}$ only	0.5	1.0000 ± 0.0000	0.0000 ± 0.0000	100.00%	100.00%
Final SBFL bank	0.5	0.0000 ± 0.0000	0.4900 ± 0.0100	0.00%	4.00%
$\mathcal{G}_k^{\text{init}}$ only	1.0	1.0000 ± 0.0000	0.0000 ± 0.0000	100.00%	100.00%
Final SBFL bank	1.0	-0.0094 ± 0.0757	0.3923 ± 0.0266	0.00%	4.00%

Table 8: Effect of Gaussian perturbation on uploaded CLIP features on CIFAR-100. Feature and prototype cosines are computed with respect to the unperturbed representations.

Backbone	Noise σ	Accuracy (%)↑	Feature cosine↑	Prototype cosine↑	Prototype L_2 drift↓
CNN	0	58.06	1.0000 ± 0.0000	1.0000 ± 0.0000	0.0000 ± 0.0000
	0.005	57.60	0.9937 ± 0.0000	0.9988 ± 0.0001	0.0399 ± 0.0016
	0.010	57.25	0.9754 ± 0.0001	0.9954 ± 0.0004	0.0805 ± 0.0031
	0.020	57.09	0.9111 ± 0.0002	0.9822 ± 0.0015	0.1647 ± 0.0056
	0.050	56.88	0.6622 ± 0.0006	0.9061 ± 0.0055	0.3951 ± 0.0092
ResNet-50	0	66.71	1.0000 ± 0.0000	1.0000 ± 0.0000	0.0000 ± 0.0000
	0.005	66.51	0.9937 ± 0.0000	0.9999 ± 0.0000	0.0137 ± 0.0001
	0.010	66.44	0.9754 ± 0.0000	0.9995 ± 0.0000	0.0320 ± 0.0001
	0.020	66.36	0.9111 ± 0.0001	0.9980 ± 0.0000	0.0863 ± 0.0002
	0.050	66.18	0.6622 ± 0.0002	0.9879 ± 0.0002	0.2889 ± 0.0005

CNN uses $\beta = 1.0$ and ResNet-50 uses $\beta = 0.1$; both settings are evaluated for 50 communication rounds.

complementary aspects of empirical privacy exposure: leakage of client label distributions and perturbation of uploaded CLIP features. These experiments characterize empirical exposure rather than provide a formal privacy guarantee.

Client label-distribution leakage. We consider an honest-but-curious server that estimates client k's original class distribution from the labels attached to its uploaded semantic bank. Let p_k denote the true local label distribution and $\hat{p}_k$ the distribution inferred from the uploaded labels. We report Pearson correlation, total-variation (TV) distance, majority-class inference accuracy, and top-5 overlap.

As shown in Table 7, using $\mathcal{G}_k^{\text{init}}$ alone exactly reproduces the local label histogram. In contrast, the final SBFL bank exhibits near-zero correlation with the original class distribution, substantial TV distances, 0% majority-class accuracy, and only 2%–4% top-5 overlap. This indicates that the generated semantics and mixed-bank construction substantially reduce the correspondence between uploaded label composition and the original client class proportions. The lower TV distance at larger β is expected because less-skewed client distributions are closer to the approximately balanced final bank.

Perturbation of uploaded CLIP features. We further perturb each ℓ_2-clipped CLIP feature before upload as $\tilde{g} = \text{clip}(g) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ and $\sigma \in \{0, 0.005, 0.010, 0.020, 0.050\}$. As shown in Table 8, increasing σ progressively reduces individual-feature cosine, reaching 0.6622 at $\sigma = 0.05$ for both backbones. Meanwhile, prototype cosine remains 0.9061 for CNN and 0.9879 for ResNet-50, while accuracy decreases by only 1.18 and 0.53 points, respectively. These results show that class-level semantic structure and downstream utility remain relatively stable under substantial perturbation of individual uploaded embeddings.

Overall, Tables 7 and 8 provide complementary analyses of uploaded label composition and semantic features. SBFL does not provide formal differential-privacy guarantees, but the results characterize the empirical privacy–utility behavior of the shared semantic bank.

5 Conclusion

We proposed SBFL, a semantic-bridge framework that transfers offline VLM semantics into lightweight heterogeneous FL. By verifying semantic banks, learning student-compatible teacher geometry, and using semantic alignment and replay, SBFL improves FL backbones under label skew and domain shift, showing that bridged VLM semantics can serve as practical class-level references.

6 Acknowledgement

The authors gratefully acknowledge the support of the National Key R&D Program of China (No. 2024YFE0100700).

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul N. Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2021. 3

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 7, 15, 24

[3] Lele Fu, Tianchi Liao, Sheng Huang, Bowen Deng, Chuanfu Zhang, Shirui Pan, and Chuan Chen. Unsupervised federated graph learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. 1

[4] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. CLIP-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, 132:581–595, 2024. 2, 3

[5] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073, 2012. 7, 15, 24

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 7, 16, 24

[7] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3

[8] Ming Hu, Yue Cao, Anran Li, Zhiming Li, Chengwei Liu, Tianlin Li, Mingsong Chen, and Yang Liu. FedMut: Generalized federated learning via stochastic mutation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12528–12537, 2024. 7, 8, 19

[9] Ming Hu, Zhihao Yue, Xiaofei Xie, Cheng Chen, Yihao Huang, Xian Wei, Xiang Lian, Yang Liu, and Mingsong Chen. Is aggregation the only choice? federated learning via layer-wise model recombination. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1096–1107, 2024. 7, 8, 19

[10] Wenke Huang, Mang Ye, Zekun Shi, He Li, and Bo Du. Rethinking federated learning with domain shift: A prototype view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16312–16322, 2023. 3

[11] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2):1–210, 2021. 1

[12] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5132–5143. PMLR, 2020. 1, 3

[13] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. MaPLe: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19113–19122, 2023. 2, 3

[14] Zhiqiang Kou, Junxiang Wu, Wenke Huang, Wenwen He, Ming-Kun Xie, Changwei Wang, Yuheng Jia, Di Jiang, Yang Liu, Xin Geng, and Qiang Yang. FedHarmony: Harmonizing heterogeneous label correlations in federated multi-label learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10324–10334, 2026. 1

[15] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 7, 15, 24

[16] Gihun Lee, Minchan Jeong, Yongjin Shin, Sangmin Bae, and Se-Young Yun. Preservation of the global knowledge by not-true distillation in federated learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 38461–38474, 2022. 3

[17] Daliang Li and Junpu Wang. FedMD: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019. 3

[18] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10713–10722, 2021. 1, 3, 7, 8, 19

[19] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems*, volume 2, pages 429–450, 2020. 1, 3, 7, 8, 19

[20] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. FedBN: Federated learning on non-iid features via local batch normalization. In *International Conference on Learning Representations*, 2021. 3, 19

[21] Zexi Li, Tao Lin, Xinyi Shang, and Chao Wu. Revisiting weighted aggregation in federated learning with neural networks. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19767–19788. PMLR, 2023. 1, 3

[22] Victor Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y. Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022. 2

[23] Tianchi Liao, Lele Fu, Sheng Huang, Qing Hu, Hong-Ning Dai, and Chuan Chen. Prototype-guided bilateral alignment multimodal federated learning. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, volume 306 of *Proceedings of Machine Learning Research*, pages 73777–73804. PMLR, 2026. 1

[24] 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*, volume 33, pages 2351–2363, 2020. 3

[25] Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. In *Advances in Neural Information Processing Systems*, volume 34, pages 5972–5984, 2021. 3

[26] Yanbiao Ma, Wei Dai, Wenke Huang, and Jiayi Chen. Geometric knowledge-guided localized global distribution alignment for federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20958–20968, 2025. 2, 3, 4

[27] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 2017. 1, 2, 7, 8, 19

[28] Xutong Mu, Yulong Shen, Ke Cheng, Xueli Geng, Jiaxuan Fu, Tao Zhang, and Zhiwei Zhang. FedProc: Prototypical contrastive federated learning on non-iid data. *Future Generation Computer Systems*, 143:93–104, 2023. 3

[29] 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. 7, 15, 24

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021. 2, 24

[31] Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H. Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021. 3

[32] Yasar Abbas Ur Rehman, Yan Gao, Pedro Porto Buarque de Gusmao, Mina Alibeigi, Jiajun Shen, and Nicholas D. Lane. L-DAWA: Layer-wise divergence aware weight aggregation in federated self-supervised visual representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16464–16473, 2023. 1, 3

[33] Xinyi Shang, Yang Lu, Gang Huang, and Hanzi Wang. Federated learning on heterogeneous and long-tailed data via classifier re-training with federated features. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pages 2218–2224, 2022. 2, 3

[34] Changlong Shi, He Zhao, Bingjie Zhang, Mingyuan Zhou, Dandan Guo, and Yi Chang. FedAWA: Adaptive optimization of aggregation weights in federated learning using client vectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 30651–30660, 2025. 1, 3, 7, 8, 19

[35] Jiangming Shi, Shanshan Zheng, Xiangbo Yin, Yang Lu, Yuan Xie, and Yanyun Qu. CLIP-guided federated learning on heterogeneity and long-tailed data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 14955–14963, 2024. 2, 3, 8, 19

[36] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. FedProto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8432–8440, 2022. 3

[37] Yue Tan, Guodong Long, Jie Ma, Lu Liu, Tianyi Zhou, and Jing Jiang. Federated learning from pre-trained models: A contrastive learning approach. In *Advances in Neural Information Processing Systems*, volume 35, pages 19332–19344, 2022. 2, 3

[38] Kirill Tyshchuk, Polina Karpikova, Andrew Spiridonov, Anastasiia Prutianova, Anton Razzhigaev, and Alexander Panchenko. On isotropy of multimodal embeddings. *Information*, 14(7):392, 2023. 2

[39] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. 2

[40] 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*, volume 33, pages 7611–7623, 2020. 3

[41] Hailong Yan, Ao Li, Xiangtao Zhang, Zhe Liu, Zenglin Shi, Ce Zhu, and Le Zhang. MobileIE: An extremely lightweight and effective ConvNet for real-time image enhancement on mobile devices. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 21949–21960, 2025. 1

[42] Can Yaras, Siyi Chen, Peng Wang, and Qing Qu. Explaining and mitigating the modality gap in contrastive multimodal learning. In *Conference on Parsimony and Learning*, volume 280 of *Proceedings of Machine Learning Research*, pages 1365–1387. PMLR, 2025. 2

[43] Rui Ye, Mingkai Xu, Jianyu Wang, Chenxin Xu, Siheng Chen, and Yanfeng Wang. FedDisco: Federated learning with discrepancy-aware collaboration. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 39879–39902. PMLR, 2023. 1, 3

[44] Hao Yu, Xin Yang, Boyang Fan, Yonghao Li, Hanlin Gu, Junbo Zhang, Tianrui Li, and Qiang Yang. Robust personalized federated continual learning via explainable multi-granularity prompt. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(11):13984–14001, 2026. 1

[45] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 16768–16776, 2024. 3

[46] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of CLIP for few-shot classification. In *European Conference on Computer Vision*, pages 493–510. Springer, 2022. 2, 3

[47] Xiangtao Zhang, Eleftherios Kofidis, Ruituo Wu, Ce Zhu, Le Zhang, and Yipeng Liu. Coupled tensor train decomposition in federated learning. *Pattern Recognition*, 170:112067, 2026. 1

[48] Xiangtao Zhang, Sheng Li, Ao Li, Yipeng Liu, Fan Zhang, Ce Zhu, and Le Zhang. Subspace constraint and contribution estimation for heterogeneous federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20632–20642, 2025. 1

[49] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022. 2, 3

[50] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130:2337–2348, 2022. 2, 3

[51] Yanbing Zhou, Xiangmou Qu, Chenlong You, Jiyang Zhou, Jingyue Tang, Xin Zheng, Chunmao Cai, and Yingbo Wu. FedSA: A unified representation learning via semantic anchors for prototype-based federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23009–23017, 2025. 3

[52] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12878–12889. PMLR, 2021. 3

A Experimental Details

A.1 Datasets

We evaluate SBFL on both label-skewed and domain-shifted federated learning settings. For label-skewed FL, we use CIFAR-10, CIFAR-100 [15], and Tiny-ImageNet, a subset of ImageNet [2]. For domain-shifted FL, we use Office-Caltech 10 [5] and Digit5 [29]. Table 9 summarizes the datasets used in our experiments.

Table 9: Dataset summary for label-skewed and domain-shifted FL experiments.

Dataset	Scenario	Modality	Classes	Clients/Domains	Backbone
CIFAR-10	Label skew	Image	10	10 clients	CNN / ResNet-50
CIFAR-100	Label skew	Image	100	10 clients	CNN / ResNet-50
Tiny-ImageNet	Label skew	Image	200	10 clients	CNN / ResNet-50
Office-Caltech 10	Domain shift	Image	10	4 domains	ResNet-50
Digit5	Domain shift	Image	10	5 domains	CNN

For CIFAR-10, CIFAR-100, and Tiny-ImageNet, all clients share the same label space but have different class proportions. For Office-Caltech 10, the four domains Amazon, Caltech, DSLR, and Webcam are treated as four clients. For Digit5, the five domains SYN, USPS, SVHN, MNIST-M, and MNIST are treated as five clients. All results are reported using Top-1 accuracy. For domain-shifted datasets, C.Avg denotes the arithmetic mean over domains.

A.2 Client Data Partition

For label-skewed FL, we partition the training set across clients using the Dirichlet distribution. Specifically, for each class c, the class samples are split among $K = 10$ clients according to

$$\boldsymbol{\pi}_c \sim \mathrm{Dirichlet}(\beta),$$

where $\beta \in \{0.1, 0.5, 1.0\}$ controls the degree of label skew. A smaller β produces a more imbalanced client label distribution and stronger statistical heterogeneity. After partitioning, each client trains only on its assigned local samples. The test set is kept unchanged and used for evaluating the global model.

Figure 6 visualizes the client-level class distributions under the most heterogeneous label-skew setting, i.e., $\beta = 0.1$. The heatmaps show that samples of each class are unevenly assigned to clients, and many clients observe only a subset of classes with skewed sample counts. This setting creates severe local class imbalance and incomplete class coverage, which is the main heterogeneous FL scenario studied in our label-skewed experiments.

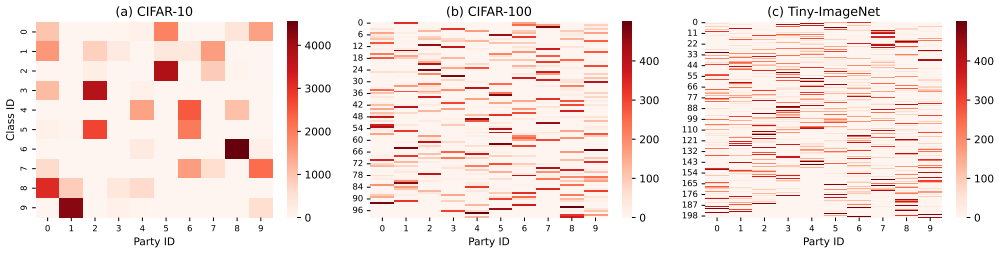


Figure 6: Client-level class distributions on CIFAR-10, CIFAR-100, and Tiny-ImageNet under Dirichlet label-skew partition with $\beta = 0.1$. Each rectangle denotes the number of samples of a class assigned to a client.

For domain-shifted FL, we follow the standard domain-as-client protocol. Each domain corresponds to one client, and clients differ mainly in visual style, feature statistics, and image acquisition conditions. This setting evaluates whether SBFL can provide useful semantic guidance when the label space is shared but feature distributions are shifted across clients.

A.3 Model Architectures

Student CNN. For CIFAR-10, CIFAR-100, Tiny-ImageNet, and Digit5, we use a lightweight CNN student model. The CNN contains three convolutional blocks followed by global average pooling and a two-layer projection MLP. Each convolutional block consists of a 3×3 convolution, batch

normalization, ReLU activation, and 2×2 max pooling. The channel widths of the three convolutional blocks are 64, 128, and 256, respectively. After the last convolutional block, global average pooling produces a 256-dimensional feature vector, which is then passed through a two-layer MLP. The final projection dimension is set to 256.

ResNet backbone. For CIFAR-10, CIFAR-100, Tiny-ImageNet, and Office-Caltech 10, we use ResNet-50 [6] as the student backbone. For all student models, the representation is followed by a projection head and a classifier, consistent with the decomposition $f_S(x;\theta) = C_S(P_S(E_S(x)))$.

A.4 Semantic Bridge Architecture

The semantic bridge maps CLIP-derived semantic features into student-compatible teacher spaces. It consists of an adapter $A(\cdot)$, a projector $P_G(\cdot)$, and a classifier $C_G(\cdot)$. The adapter is implemented as a two-layer MLP with layer normalization:

$$A(g) = \mathrm{LN}\left(W_2\,\mathrm{GELU}\left(W_1\,\mathrm{LN}(g)\right)\right),$$

where g denotes a CLIP-derived semantic feature. The hidden dimension of the adapter is set to 1024, and the output dimension matches the corresponding student representation dimension. This design first normalizes the CLIP feature, then performs nonlinear transformation, and finally normalizes the bridged teacher feature before it is used for prototype construction and semantic replay.

Table 10: Adapter architecture used in the semantic bridge.

Layer	Description
Input normalization	LayerNorm over CLIP feature dimension
Linear 1	Input dimension $\rightarrow$ 1024
Activation	GELU
Linear 2	1024 $\rightarrow$ student-compatible dimension
Output normalization	LayerNorm over output dimension

A.5 Notation and End-to-End Workflow

For clarity, Table 11 summarizes the main notation used in SBFL.

Table 11: Summary of the main notation used in SBFL.

Symbol	Description
$\mathcal{G}_k^{\mathrm{init}}$	CLIP image embeddings extracted from the original data of client k
$\mathcal{G}_k^{\mathrm{gen}}$	Generated semantic candidates of client k
$\mathcal{G}_k^{\mathrm{keep}}$	Generated candidates retained after text verification
$\bar{\mathcal{G}}_k$	Final local semantic bank, $\mathcal{G}_k^{\mathrm{init}} \cup \mathcal{G}_k^{\mathrm{keep}}$
$\bar{\mathcal{G}}^{\mathrm{glob}}$	Balanced global semantic bank constructed by client-wise sampling
(g, y)	A semantic feature and its associated class label
t_c	Normalized CLIP text prototype of class c
A_t	Semantic adapter at communication round t
$P_{G,t}$	Teacher projection head at round t
$C_{G,t}$	Teacher classifier at round t
h^T, p^T	Bridged teacher representations in feature and projection spaces
h^S, p^S	Student representations used for semantic alignment
$q_{k,c}^L, q_c^G, q_{k,c}^M$	Local, global, and mixed teacher-space prototypes of class c

The overall SBFL procedure can be summarized into three stages.
(1) Local semantic-bank construction. Before federated training, client k extracts CLIP image embeddings $\mathcal{G}_k^{\mathrm{init}}$ from its local images and obtains generated semantic candidates $\mathcal{G}_k^{\mathrm{gen}}$. Text verification is applied only to the generated candidates using CLIP text prototypes to check label consistency and the text-consistency margin. The retained subset $\mathcal{G}_k^{\mathrm{keep}}$ is combined with the original embeddings to form

$$\bar{\mathcal{G}}_k = \mathcal{G}_k^{\mathrm{init}} \cup \mathcal{G}_k^{\mathrm{keep}}.$$

(2) Server-side semantic bridging. Each client uploads the feature–label pairs in $\bar{\mathcal{G}}_k$ once before federated training. The server performs client-wise equalized sampling to construct $\bar{\mathcal{G}}^{\mathrm{glob}}$ and learns

the semantic bridge $\Psi_t = \{A_t, P_{G,t}, C_{G,t}\}$. A CLIP-space feature is transformed as

$$g \to h^T = A_t(g) \to p^T = P_{G,t}(h^T),$$

corresponding to the raw-CLIP, Bridge-h, and Bridge-p spaces.
(3) Client-side semantic guidance. At each communication round, the server refreshes the bridge on the fixed global semantic bank and freezes it during local training. Bridge-derived local and global class prototypes are combined into mixed teacher prototypes for representation- and projection-level alignment. Quality-aware semantic replay provides additional feature-level supervision. The updated student models are then returned to the server for standard federated aggregation.

A.6 Training Hyperparameters

Table 12 summarizes the default training configuration. We use SGD with momentum for local optimization. For label-skewed experiments, we set the prototype mixing coefficient to $r = 0$, since each client already follows the same visual domain and local semantic banks provide useful client-specific information. For domain-shifted experiments, we set $r = 0.4$ to incorporate more global semantic prototypes and reduce domain-specific bias.

Table 12: Default training and semantic-bridge hyperparameters used in the main experiments.

Hyperparameter	Value
Communication rounds on CIFAR-10/100	100
Communication rounds on Tiny-ImageNet with CNN	100
Communication rounds on Tiny-ImageNet with ResNet-50	20
Communication rounds on Office-Caltech 10 / Digit5	30
Local epochs	10
Batch size	64
Batch size for Tiny-Imagenet	16
Optimizer	SGD
Learning rate	0.01
Momentum	0.9
Weight decay	10^{-5}
Projection dimension	256
CLIP backbone	ViT-B/16
Server bridge-training epochs E_s	5
Text-filtering margin threshold δ	0
Prototype/replay temperature τ	0.07
Replay weighting temperature τ_r	0.07
Geometry regularization weight λ_{geo}	0.05
λ_{dist} for CIFAR-10/100 and Office-Caltech 10	2.0
λ_{dist} for Tiny-ImageNet and Digit5	0.1
λ_{replay}	0.5
Prototype mixing ratio r for label skew	0
Prototype mixing ratio r for domain shift	0.4

A.7 Semantic Prior Construction

For each client k, we construct a raw semantic pool

$$\mathcal{G}_k^{\text{raw}} = \mathcal{G}_k^{\text{init}} \cup \mathcal{G}_k^{\text{gen}},$$

where $\mathcal{G}_k^{\text{init}}$ contains CLIP ViT-B/16 image embeddings extracted from local images and $\mathcal{G}_k^{\text{gen}}$ contains generated semantic features produced by the same semantic-feature generation pipeline for all compared variants. The generator operates in the cached CLIP feature space and produces class-conditioned semantic features without accessing raw client images during federated training. CLIP is used only before federated communication to extract cached semantic priors and text prototypes. During federated training, clients do not perform online CLIP inference.
CLIP text prototypes are constructed offline using class-name prompts. For CIFAR-10, CIFAR-100, Office-Caltech 10, and Digit5, we use prompt templates such as "a photo of a {class}", "a blurry photo of a {class}", "a close-up photo of a {class}", and "an image of a {class}". For Tiny-ImageNet, we additionally use common image-level templates such as "a cropped photo of a {class}", "a low

resolution photo of a {class}", and "a photo of a {class} in nature". For each class, text embeddings from all templates are averaged and normalized to obtain the class text prototype.
Generated semantic features are verified using CLIP text prototypes. A generated semantic feature is retained only when its CLIP-text prediction agrees with its assigned class label and its text-consistency margin exceeds the threshold δ. Unless otherwise specified, we set $\delta = 0$. The verified local semantic bank is

$$\bar{\mathcal{G}}_k = \mathcal{G}_k^{\text{init}} \cup \mathcal{G}_k^{\text{keep}}.$$

To avoid domination by clients with larger semantic banks, the server constructs a balanced global bank by equalized client sampling:

$$\bar{\mathcal{G}}^{\text{glob}} = \bigcup_{k=1}^{K} \text{Sample}(\bar{\mathcal{G}}_k, M), \qquad M = \min_{u \in K} |\bar{\mathcal{G}}_u|.$$

A.8 Semantic Sharing, Balancing, and Privacy Boundary

What is uploaded. Raw client images are never transmitted. Before federated training, each client k constructs its final local semantic bank

$$\bar{\mathcal{G}}_k = \mathcal{G}_k^{\text{init}} \cup \mathcal{G}_k^{\text{keep}},$$

where $\mathcal{G}_k^{\text{init}}$ contains CLIP image embeddings extracted from original local images and $\mathcal{G}_k^{\text{keep}}$ contains generated semantic features retained after local text verification. Client k uploads only the feature–label pairs (g_i, y_i) in $\bar{\mathcal{G}}_k$, and this upload occurs once before federated training.

Client-wise balancing. Let $N_k = |\bar{\mathcal{G}}_k|$. The server sets

$$M = \min_k N_k$$

and uniformly samples M feature–label pairs from each client. The balanced global semantic bank is therefore

$$\bar{\mathcal{G}}^{\text{glob}} = \bigcup_{k=1}^{K} \text{Sample}(\bar{\mathcal{G}}_k, M).$$

Thus, balancing is performed *client by client*, rather than by jointly pooling all clients or categories before sampling. Each client contributes the same number of semantic entries. Local semantic candidate pools are also constructed to be approximately class-balanced whenever possible.

One-time semantic communication. Let d denote the semantic-feature dimension. The one-time upload of client k requires $\mathcal{O}(dN_k)$ communication, and the total semantic upload is

$$\mathcal{O}\left(d \sum_{k=1}^{K} N_k\right).$$

In contrast, standard bidirectional model communication over T rounds is $\mathcal{O}(2KTP)$, where P denotes the model payload size. Since the semantic bank is uploaded only once before FL training, its amortized contribution decreases with the number of communication rounds.
Under the CIFAR-100/ResNet-50 setting with $K = 10$, $d = 512$, and $T = 100$, the final semantic banks contain approximately $73{,}059$ feature–label pairs in total, corresponding to approximately $7{,}306$ pairs per client on average. The resulting upload is approximately 28.59 MiB per client and 285.94 MiB in total. For comparison, one standard FL communication round with 10 clients requires approximately 2159.13 MiB of bidirectional model communication. Thus, the complete one-time semantic upload corresponds to approximately 13.24% of one standard FL communication round.

Privacy boundary. Although SBFL does not transmit raw images, CLIP-space embeddings may retain information about visual content, domain characteristics, or membership of private samples. Therefore, SBFL does not claim a formal privacy guarantee such as differential privacy. Section 4.4 empirically evaluates label-distribution leakage and perturbation of uploaded CLIP features. Under stricter privacy requirements, SBFL can be combined with explicit privacy mechanisms such as differential privacy, secure aggregation, or feature perturbation.

Table 13: Additional comparisons with FedBN and FedHEAL under domain shift. C.Avg denotes the arithmetic mean over domains. Best results are in bold.

	Office-Caltech 10					Digit5					
Method	Amaz	Calt	DSLR	Webc	C.Avg	SYN	USPS	SVHN	MNTM	MNIST	C.Avg
FedAvg	83.33	**73.78**	71.88	66.10	73.77	**99.52**	93.06	89.57	77.33	97.48	91.39
FedBN	66.67	60.00	**93.75**	72.88	73.33	98.47	91.02	**91.18**	**95.35**	97.85	94.77
FedHEAL	81.25	72.44	78.12	71.19	75.75	98.64	94.35	89.43	86.74	98.03	93.44
SBFL	**85.42**	70.67	87.50	**84.75**	**82.08**	98.92	**97.42**	89.95	90.18	**98.67**	**95.03**

A.9 Baseline Implementation

We compare SBFL with representative heterogeneous FL baselines, including FedAvg [27], FedProx [19], MOON [18], FedMut [8], FedMR [9], and FedAWA [34]. These baselines cover standard aggregation, local regularization, representation consistency, model recombination, stochastic mutation, and adaptive aggregation. All baselines are trained under the same dataset partitions, communication rounds, local epochs, optimizer, learning rate, and backbone architectures as SBFL unless otherwise specified. For plug-and-play experiments, we insert the proposed semantic bridge, prototype alignment, and semantic replay into different FL backbones while keeping the corresponding backbone-specific regularization or aggregation rule unchanged.

A.10 Additional Domain-Shift Baselines

To further strengthen the comparison under domain shift, we additionally evaluate FedBN [20] and FedHEAL under the same data partitions, communication rounds, local epochs, backbone architectures, and optimization settings as the corresponding SBFL experiments. FedBN provides a representative normalization-based approach for feature heterogeneity, while FedHEAL provides an additional recent baseline designed for heterogeneous federated learning.

As shown in Table 13, SBFL achieves the best average accuracy on both domain-shift benchmarks. On Office-Caltech 10, SBFL reaches 82.08%, outperforming FedBN and FedHEAL by 8.75 and 6.33 percentage points, respectively. On Digit5, SBFL achieves 95.03%, exceeding FedBN by 0.26 points and FedHEAL by 1.59 points. Although individual methods may perform better on particular domains, SBFL provides the strongest overall average across both benchmarks. These results further support the effectiveness of semantic bridging under feature and domain heterogeneity.

A.11 Comparison with a Complete CLIP2FL

Table 14: Comparison with a complete CLIP2FL baseline under label-skewed FL. FedAvg denotes the standard baseline without CLIP guidance. CLIP2FL follows the complete direct CLIP-guided pipeline, including client-side CLIP distillation and server-side CLIP-guided classifier retraining, while SBFL uses text verification and semantic bridging before local training.

Dataset	Backbone	β	FedAvg	CLIP2FL-adapted	SBFL
CIFAR-10	CNN	0.1	79.02	81.49	**82.55**
CIFAR-10	CNN	0.5	85.85	86.40	**87.09**
CIFAR-10	CNN	1.0	87.98	88.17	**88.57**
CIFAR-100	ResNet-50	0.1	62.41	64.78	**68.56**
CIFAR-100	ResNet-50	0.5	64.84	70.12	**72.13**
CIFAR-100	ResNet-50	1.0	65.44	71.82	**73.86**

To examine whether the improvement of SBFL comes merely from using CLIP semantics, we further compare with a complete CLIP2FL-adapted baseline under our experimental setting. The original CLIP2FL framework transfers the knowledge of an off-the-shelf CLIP model to federated client–server models: on the client side, CLIP is used as a teacher for knowledge distillation, and on the server side, CLIP text supervision is used to guide federated feature generation and retrain a balanced server classifier [35]. Following this design, our CLIP2FL-adapted baseline includes both client-side CLIP-guided distillation and server-side CLIP-guided classifier retraining with federated semantic features. It uses the same federated data partitions, backbone architectures, communication rounds, local epochs, optimizer, and learning rate as the corresponding FedAvg and SBFL settings. Different from SBFL, this CLIP2FL-adapted baseline directly injects CLIP-derived guidance into the federated training process. It does not use text-verified semantic banks, the proposed server-side semantic bridge, mixed semantic prototype alignment, or quality-aware semantic replay. Therefore,

this comparison isolates the effect of our bridge-based semantic transfer from a complete direct CLIP-guided FL pipeline.
Table 14 reports the comparison under label-skewed FL. The CLIP2FL-adapted baseline consistently improves over FedAvg, confirming that CLIP semantics provide useful external guidance. Nevertheless, SBFL outperforms CLIP2FL-adapted in all six settings, with an average gain of 1.66 points. On CIFAR-100, the gains are 3.78, 2.01, and 2.04 points under $\beta = 0.1$, 0.5, and 1.0, respectively. These results suggest that the improvement of SBFL does not come merely from using CLIP semantics; verifying and bridging CLIP-derived priors provides additional benefits beyond a complete direct CLIP-guided transfer baseline.

A.12 CLIP-Enabled and Matched-Resource Controls

To distinguish the benefit of accessing pretrained CLIP knowledge from the benefit of the proposed semantic-transfer mechanism, we conduct additional controlled comparisons on CIFAR-100 using ResNet-50 under the same federated settings.

Comparison with CLIP-enabled and prototype-based baselines. Table 15 compares SBFL with FedTGP and the CLIP2FL-adapted baseline. FedTGP provides a representative prototype-based FL comparison without CLIP, whereas CLIP2FL-adapted directly introduces pretrained CLIP guidance.

Table 15: Comparison with CLIP-enabled and prototype-based baselines on CIFAR-100/ResNet-50.

Method	Uses CLIP	Class prototypes	$\beta = 0.1$	$\beta = 0.5$	$\beta = 1.0$
FedAvg	✗	✗	62.41	64.84	65.44
FedTGP	✗	✓	64.58	65.40	67.32
CLIP2FL	✓	✗	64.78	70.12	71.82
SBFL	✓	✓	**68.56**	**72.13**	**73.86**

Compared with CLIP2FL-adapted, SBFL improves accuracy by 3.78, 2.01, and 2.04 points under $\beta = 0.1$, 0.5, and 1.0, respectively. These results indicate that the improvement cannot be explained solely by access to pretrained CLIP knowledge or by prototype-based learning.

Same-bank controlled comparison. We further retain exactly the same semantic bank and progressively introduce the bridge and replay mechanisms. SameBank-Direct directly aligns local student representations with semantic features through a linear projection head, without semantic bridging or replay.

Table 16: Matched-resource comparison using the same semantic bank on CIFAR-100/ResNet-50.

Method	Same bank	Bridge	Replay	$\beta = 0.1$	$\beta = 0.5$	$\beta = 1.0$
FedAvg	✗	✗	✗	62.41	64.84	65.44
SameBank-Direct	✓	✗	✗	66.51	71.11	71.48
SBFL-Bridge	✓	✓	✗	68.13	71.94	73.21
Full SBFL	✓	✓	✓	**68.56**	**72.13**	**73.86**

With the semantic resource fixed, introducing the semantic bridge improves SameBank-Direct by 1.62, 0.83, and 1.73 points, respectively. Semantic replay further improves the bridge variant by 0.43, 0.19, and 0.65 points. This controlled progression shows that representation-space adaptation and semantic replay provide gains beyond access to the semantic bank itself.

A.13 Effect of Round-Wise Semantic-Bridge Updating

The semantic bank remains fixed after its one-time construction, whereas the global student representation evolves continuously through local optimization and federated aggregation. The semantic bridge is therefore refreshed to recalibrate the fixed CLIP-space semantics to the current student representation space. At round t, the bridge is warm-started from the previous round, updated on the fixed balanced semantic bank, and then frozen during client training.
To isolate this effect, we compare one-shot bridge training with round-wise bridge updating while keeping the same semantic bank.
Round-wise updating improves accuracy by 2.14 points on CIFAR-100/ResNet-50 and 0.86 points on CIFAR-10/CNN. These results support periodically refreshing the semantic bridge to track the evolving student representation despite using a fixed semantic bank.

Table 17: Effect of round-wise semantic-bridge updating.

Dataset / Backbone	Variant	Bridge update	Accuracy (%)
CIFAR-100 / ResNet-50, $\beta = 0.1$	SBFL-Once	Round 1 only	66.42
	SBFL-Dynamic	Every round	**68.56**
CIFAR-10 / CNN, $\beta = 0.1$	SBFL-Once	Round 1 only	81.69
	SBFL-Dynamic	Every round	**82.55**

A.14 Robustness to CLIP Text Prompts

We evaluate prompt sensitivity on CIFAR-100/ResNet-50 under $\beta = 0.5$ while keeping all federated-learning settings unchanged.
The default prompt ensemble (Prompt-A) contains: "a photo of a {class}", "a blurry photo of a {class}", "a close-up photo of a {class}", and "an image of a {class}".
The alternative ensemble (Prompt-B) contains: "a photograph of {class}", "a visual example of {class}", "an image containing {class}", and "a small low-resolution image of {class}".
For each class, the four text embeddings are averaged and normalized to form the corresponding CLIP text prototype.

Table 18: Sensitivity to CLIP text-prompt ensembles on CIFAR-100/ResNet-50 with $\beta = 0.5$.

Prompt ensemble	Accuracy (%)	Final-bank retention ratio (%)
Prompt-A (Default)	**72.13**	**73.06**
Prompt-B (Alternative)	71.96	71.17

Changing the prompt ensemble alters final accuracy by only 0.17 points, while the final-bank retention ratio changes by 1.89 points. This suggests that prompt formulation influences which generated candidates pass text verification, while downstream performance remains stable in this comparison.

A.15 Failure Analysis on the Caltech Domain

Table 4 in the main text shows that SBFL improves the average accuracy on Office-Caltech 10, but does not improve every individual domain. In particular, the accuracy on the Caltech domain decreases compared with several baselines. We attribute this behavior to the interaction between domain-specific visual statistics and shared semantic guidance.
The Caltech domain is visually cleaner and more canonical than Amazon, DSLR, and Webcam, and its images often have less domain-specific noise. For such a domain, the local visual distribution can already provide relatively reliable class cues. Injecting global semantic prototypes that are mixed across domains may therefore introduce a mild mismatch: the bridged prototypes are optimized to improve cross-domain robustness on average, but they may not be perfectly aligned with the cleaner Caltech-specific feature distribution. As a result, semantic alignment can slightly reduce the advantage of local domain-specific representations on Caltech, while substantially improving harder or more shifted domains such as DSLR and Webcam.
This observation suggests that SBFL mainly improves global robustness rather than guaranteeing monotonic improvement on every domain. A possible improvement is to make the prototype mixing ratio or replay weight domain-adaptive, so that clients whose local features are already well aligned with the class structure receive weaker global semantic regularization. We leave such adaptive semantic weighting for future work.

A.16 Sensitivity to Prototype Mixing Ratio under Domain Shift

The prototype mixing ratio r controls the balance between local semantic prototypes and global semantic prototypes. A smaller r relies more on client-specific semantic banks, while a larger r incorporates stronger global semantic guidance. To study its effect under domain shift, we evaluate SBFL on Office-Caltech 10 with different values of r. All other hyperparameters are kept the same as the main Office-Caltech 10 setting.
Table 19 shows that the prototype mixing ratio r has a clear effect under domain shift. When r increases from 0.3 to 0.5, the average accuracy improves from 80.08% to 86.02%, suggesting that moderate global semantic guidance can better compensate for domain-specific visual shifts. The improvement is especially clear on DSLR and Webcam, where global prototypes provide more stable class-level references across domains. Meanwhile, Caltech improves from 70.22% to 73.78%, indicating that a moderate global-prototype contribution can reduce the mismatch observed with

Table 19: Sensitivity to the prototype mixing ratio r on Office-Caltech 10. C.Avg denotes the arithmetic mean over domains.

Setting	Amazon	Caltech	DSLR	Webcam	C.Avg
SBFL, $r = 0.3$	78.12	70.22	90.62	81.36	80.08
SBFL, $r = 0.4$	85.42	70.67	87.50	84.75	82.08
SBFL, $r = 0.5$	85.16	73.78	95.32	89.84	**86.02**
SBFL, $r = 0.6$	75.52	70.67	**100.00**	86.44	83.16

weaker global mixing. However, further increasing r to 0.6 reduces the average accuracy to 83.16%, mainly due to the drop on Amazon. This suggests that overly strong global mixing may weaken domain-specific adaptation on some clients, even though it can still benefit domains such as DSLR. These results indicate that the optimal local–global prototype balance depends on the severity of domain shift and motivates future work on adaptive prototype mixing.

A.17 Compute Resources

All experiments are conducted on a Linux workstation equipped with NVIDIA RTX 3090 GPUs. Each run uses a single NVIDIA RTX 3090 GPU unless otherwise specified. The CPU is a dual-socket Intel Xeon Gold 5218R @ 2.10GHz system with 80 logical CPUs. The software environment uses CUDA 12.1 and PyTorch 2.4.1. All experiments are conducted using the same implementation framework and configuration files released with the anonymized code package.

Table 20: Compute resources used for the experiments.

Item	Description
GPU type	NVIDIA RTX 3090
GPU memory	24 GB
CPU	Intel Xeon Gold 5218R @ 2.10GHz, 80 logical CPUs
RAM	32 GB
CUDA version	12.1
PyTorch version	2.4.1
Operating system	Linux

A.18 Complexity Analysis

We analyze the computation, memory, and communication overhead introduced by SBFL. Let R denote the number of communication rounds, M the number of participating clients per round, E the number of local epochs, and E_s the number of server-side bridge-training epochs. For client k, let n_k be the number of local image samples, B the image mini-batch size, and $T_k = \lceil n_k/B \rceil$ the number of local mini-batches. Let B_r be the replay batch size. We denote by P the number of student-model parameters, C the number of classes, d_c the CLIP semantic feature dimension, f the student representation dimension, o the projection dimension, and h the hidden dimension of the semantic adapter. Let g_k and g be the sizes of the retained local and global semantic banks, respectively, and let P_A be the number of adapter parameters.

Offline preprocessing. CLIP is used only once before federated communication starts to extract image semantic features and text prototypes. During federated training, SBFL operates on cached semantic features and does not require online CLIP inference. For text verification, each candidate semantic feature is compared with C text prototypes, resulting in a one-time cost

$$\mathcal{O}\left(\sum_{k=1}^{K} g_k C d_c\right). \tag{19}$$

This preprocessing cost is incurred before communication rounds and is not repeated during FL training.

Server-side computation. At the beginning of each communication round, the server updates the semantic bridge on the cached global semantic bank and freezes it for the subsequent client-side local training. For one semantic feature, the adapter–projector–classifier computation costs

$$\mathcal{C}_{\text{bridge}} = \mathcal{O}(d_c h + hf + fo + oC). \tag{20}$$

Therefore, server-side bridge training costs

$$\mathcal{O}(E_s g\, \mathcal{C}_{\text{bridge}}) \tag{21}$$

per communication round.
The CLIP-aware geometry regularizer is computed on class-level prototypes. Its pairwise similarity computation costs

$$\mathcal{O}\big(C^2(d_c + f + o)\big)\,, \tag{22}$$

where the d_c term corresponds to raw CLIP prototype relations and the f, o terms correspond to bridged representation and projection spaces. Model aggregation has the same order as standard FL:

$$\mathcal{O}(MP). \tag{23}$$

Thus, the server-side per-round cost is

$$\mathcal{O}\big(E_s g\, \mathcal{C}_{\text{bridge}} + C^2(d_c + f + o) + MP\big)\,. \tag{24}$$

Client-side computation. For client k, standard image training costs

$$\mathcal{O}(ET_k \mathcal{C}_{\text{img}})\,, \tag{25}$$

where $\mathcal{C}_{\text{img}}$ denotes the per-batch forward and backward cost of the student model on image data.
SBFL introduces three additional costs. First, before local mini-batch training, each selected client constructs and caches teacher-space prototypes using the frozen bridge. This costs

$$\mathcal{O}((g_k + g)d_c + C(d_c h + hf + fo))\,, \tag{26}$$

where $(g_k + g)d_c$ accounts for reading local/global semantic-bank statistics and the second term maps class prototypes through the adapter and teacher projector.
Second, semantic prototype alignment compares image features with class prototypes in both representation and projection spaces. For one image mini-batch, this costs

$$\mathcal{O}(BC(f + o))\,. \tag{27}$$

Third, semantic replay maps a replay batch of cached semantic features through the frozen adapter and updates the local student projector/classifier. For replay batch size B_r, this costs

$$\mathcal{O}(B_r(d_c h + hf + fo + oC))\,. \tag{28}$$

Therefore, the per-round client-side cost for client k is

$$\begin{aligned}\mathcal{O}\Big((g_k + g)d_c + C(d_c h + hf + fo) \\ + ET_k\big[\mathcal{C}_{\text{img}} + BC(f + o) + B_r(d_c h + hf + fo + oC)\big]\Big).\end{aligned} \tag{29}$$

Overall computation. For compactness, define the per-round computation of client k as

$$\mathcal{C}_k^{\text{client}} = (g_k + g)d_c + C(d_c h + hf + fo) + ET_k[\mathcal{C}_{\text{img}} + BC(f + o) + B_r(d_c h + hf + fo + oC)]\,. \tag{30}$$

The total training complexity over R communication rounds is then

$$\mathcal{O}\left(R\left[E_s g \mathcal{C}_{\text{bridge}} + C^2(d_c + f + o) + MP + \sum_{k \in \mathcal{S}_r} \mathcal{C}_k^{\text{client}}\right]\right), \tag{31}$$

where $\mathcal{S}_r$ is the set of selected clients in round r.

Communication complexity. The raw images are never transmitted, and CLIP features are extracted and cached before FL communication. During each communication round, clients upload updated student-model parameters as in standard FL. The server additionally transmits the frozen semantic bridge and class-level teacher prototypes to participating clients. Therefore, the per-round communication cost is

$$\mathcal{O}\big(MP + M\big(P_A + C(f + o)\big)\big)\,, \tag{32}$$

where P_A denotes the size of the semantic adapter and $C(f + o)$ denotes the size of the representation- and projection-level class prototypes. Since $P_A + C(f + o) \ll P$ for typical backbones such as ResNet-50, the dominant per-round communication order remains $\mathcal{O}(MP)$, the same as standard FL.

A.19 Space Complexity

On each client, SBFL stores the student model, the local semantic bank, and compact bridge/prototype information. The persistent client-side memory is

$$\mathcal{O}(P + g_k d_c + P_A + C(f + o)) . \tag{33}$$

The additional temporary memory mainly comes from replay batches, class prototypes, and batchwise similarity scores:

$$\mathcal{O}(B_r d_c + C(f + o) + BC) . \tag{34}$$

Thus, ignoring lower-order temporary tensors, the client-side space complexity is

$$\mathcal{O}(P + g_k d_c + P_A + C(f + o)) . \tag{35}$$

On the server, SBFL stores the global student model, the global semantic bank, and the semantic bridge. The server-side space complexity is

$$\mathcal{O}(P + g d_c + P_A) . \tag{36}$$

Overall, SBFL preserves the same dominant per-round communication order as standard FL, while introducing additional computation and memory overhead from server-side bridge training, semantic-bank based prototype alignment, and replay on cached semantic features.

Table 21: Complexity comparison between FedAvg and SBFL.

Metric	**FedAvg**	**SBFL**
Client computation	$\mathcal{O}(ET_k\mathcal{C}_{\text{img}})$	$\mathcal{O}((g_k + g)d_c + C(d_c h + hf + fo) + ET_k[\mathcal{C}_{\text{img}} + BC(f + o) + B_r(d_c h + hf + fo + oC)])$
Server computation	$\mathcal{O}(MP)$	$\mathcal{O}\big(E_s g\mathcal{C}_{\text{bridge}} + C^2(d_c + f + o) + MP\big)$
Client space	$\mathcal{O}(P)$	$\mathcal{O}(P + g_k d_c + P_A + C(f + o))$
Server space	$\mathcal{O}(P)$	$\mathcal{O}(P + g d_c + P_A)$
Communication	$\mathcal{O}(MP)$	$\mathcal{O}(MP)$ dominant + $\mathcal{O}(M(P_A + C(f + o)))$

A.20 Measured Runtime, Memory, and Communication Overhead

To complement the asymptotic complexity analysis, we profile FedAvg and SBFL under the same CIFAR-100/ResNet-50 setting with 10 clients.

Table 22: Measured resource overhead of SBFL relative to FedAvg.

Method	Runtime / Round	Peak Client GPU	Communication / Round	One-time Semantic Upload
FedAvg	705.63 s	2387 MiB	2159.13 MiB	0
SBFL	763.91 s (+8.26%)	2451 MiB (+2.68%)	2259.44 MiB (+4.65%)	285.94 MiB

SBFL requires 763.91 seconds per round, corresponding to an 8.26% runtime overhead over FedAvg. Peak client GPU memory increases by only 2.68%. Its recurring per-round communication is 2259.44 MiB, corresponding to a 4.65% increase over FedAvg under this implementation. The semantic bank additionally incurs a one-time upload of 285.94 MiB before training rather than a recurring per-round communication cost.

A.21 Existing Assets and Licenses

We use public benchmark datasets and publicly available model architectures. CIFAR-10 and CIFAR-100 are credited to Krizhevsky [15]. Tiny-ImageNet is a subset of ImageNet, and we cite the ImageNet paper [2]. Office-Caltech 10 is credited to the domain adaptation benchmark introduced with Geodesic Flow Kernel [5]. Digit5 follows the multi-source domain adaptation benchmark used in Moment Matching for Multi-Source Domain Adaptation [29]. We use ResNet-50 [6] as one of the student backbones and CLIP [30] for extracting offline vision–language semantic priors. The code release will include instructions for downloading or preparing all public datasets according to their original sources and terms of use.

B Convergence Analysis

We provide a non-convex convergence analysis for SBFL. Let Ψ_t denote the semantic bridge updated by the server at communication round t and frozen during client-side training. For client k, define the expected round-wise local objective as

$$F_k^t(\theta) = F_{\text{task},k}(\theta) + \lambda_{\text{dist}} \left(F_{\text{rep},k}^t(\theta) + F_{\text{proj},k}^t(\theta)\right) + \lambda_{\text{replay}} F_{\text{replay},k}^t(\theta) + \lambda_b F_{\text{backbone},k}(\theta). \tag{37}$$

The corresponding global objective is

$$F^t(\theta) = \sum_{k=1}^{K} p_k F_k^t(\theta), \qquad p_k \geq 0, \qquad \sum_{k=1}^{K} p_k = 1. \tag{38}$$

Since Ψ_t is frozen and all bridge-derived teacher features are stop-gradient during local training, semantic prototypes and replay features are constants with respect to the student parameters within round t.

B.1 Assumptions

We use the following standard assumptions for non-convex federated optimization.

Assumption 1: Smoothness and lower boundedness. On a domain $\mathcal{D}$ containing all iterates, every F_k^t is L-smooth:

$$\left\|\nabla F_k^t(\theta) - \nabla F_k^t(\theta')\right\| \leq L\|\theta - \theta'\|. \tag{39}$$

The round-wise objectives admit a common finite lower bound

$$F^* := \inf_{\substack{0 \leq s \leq T-1 \\ \theta \in \mathcal{D}}} F^s(\theta) > -\infty. \tag{40}$$

Assumption 2: Unbiased stochastic gradients. Let $\mathcal{F}_{t,e}$ denote the history before computing the stochastic gradient $g_{k,e}^t$. Then

$$\mathbb{E}\big[g_{k,e}^t \mid \mathcal{F}_{t,e}\big] = \nabla F_k^t(\theta_{k,e}^t), \tag{41}$$

with conditionally independent mini-batch sampling across clients.

Assumption 3: Bounded variance and second moment. There exist finite constants σ^2 and G^2 such that

$$\mathbb{E}\Big[\left\|g_{k,e}^t - \nabla F_k^t(\theta_{k,e}^t)\right\|^2 \mid \mathcal{F}_{t,e}\Big] \leq \sigma^2, \qquad \mathbb{E}\|g_{k,e}^t\|^2 \leq G^2. \tag{42}$$

Assumption 4: Bounded bridge variation. For $t = 0, \ldots, T-2$, assume there exists $\delta_t \geq 0$ such that

$$F^{t+1}(\theta) \leq F^t(\theta) + \delta_t, \qquad \forall \theta \in \mathcal{D}, \tag{43}$$

and

$$\sum_{t=0}^{T-2} \delta_t \leq V_\Psi < \infty, \tag{44}$$

where V_Ψ is independent of T. If the semantic bridge is fixed, then $V_\Psi = 0$.

B.2 Convergence Result

We analyze full client participation with standard local SGD/FedAvg updates:

$$\theta_{k,0}^t = \theta^t, \qquad \theta_{k,e+1}^t = \theta_{k,e}^t - \eta g_{k,e}^t, \qquad e = 0, \ldots, E-1, \tag{45}$$

followed by

$$\theta^{t+1} = \sum_{k=1}^{K} p_k \theta_{k,E}^t. \tag{46}$$

Let $\gamma = \eta E$.

Theorem 1. Under Assumptions 1–4 and $\gamma \leq 1/(8L)$,

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\nabla F^t(\theta^t)\|^2 \leq \frac{2\left(F^0(\theta^0) - F^* + V_\Psi\right)}{\gamma T} + \frac{2L\gamma\left(\sum_{k=1}^{K} p_k^2\right)\sigma^2}{E} + \frac{5}{6}L^2\gamma^2 G^2. \tag{47}$$

The first term corresponds to optimization progress and cumulative bridge variation, the second term captures stochastic-gradient variance, and the third term captures local-update drift.

Proof. Define

$$\bar{g}_t = \frac{1}{E}\sum_{k=1}^{K} p_k \sum_{e=0}^{E-1} g_{k,e}^t, \qquad \bar{\nabla}_t = \frac{1}{E}\sum_{k=1}^{K} p_k \sum_{e=0}^{E-1} \nabla F_k^t(\theta_{k,e}^t). \tag{48}$$

The global update can be written as

$$\theta^{t+1} = \theta^t - \gamma \bar{g}_t. \tag{49}$$

By conditional unbiasedness and independence of stochastic mini-batch noise,

$$\mathbb{E}\|\bar{g}_t - \bar{\nabla}_t\|^2 \le \frac{\sigma^2}{E}\sum_{k=1}^{K} p_k^2. \tag{50}$$

Furthermore,

$$\theta_{k,e}^t - \theta^t = -\eta \sum_{j=0}^{e-1} g_{k,j}^t. \tag{51}$$

Using the bounded second moment and Cauchy–Schwarz inequality,

$$\mathbb{E}\|\theta_{k,e}^t - \theta^t\|^2 \le \eta^2 e^2 G^2. \tag{52}$$

Averaging over local steps and using L-smoothness gives

$$\mathbb{E}\left\|\bar{\nabla}_t - \nabla F^t(\theta^t)\right\|^2 \le L^2\gamma^2 G^2. \tag{53}$$

By smoothness of F^t,

$$F^t(\theta^{t+1}) \le F^t(\theta^t) - \gamma\langle \nabla F^t(\theta^t), \bar{g}_t\rangle + \frac{L\gamma^2}{2}\|\bar{g}_t\|^2. \tag{54}$$

Taking expectations, applying Eqs. (50)–(53), Young's inequality, and $\gamma \le 1/(8L)$ yields

$$\mathbb{E}F^t(\theta^{t+1}) \le \mathbb{E}F^t(\theta^t) - \frac{\gamma}{2}\mathbb{E}\|\nabla F^t(\theta^t)\|^2 + \frac{L\gamma^2\sigma^2}{E}\sum_{k=1}^{K} p_k^2 + \frac{5}{12}L^2\gamma^3 G^2. \tag{55}$$

For $t = 0, \ldots, T-2$, Assumption 4 gives

$$F^{t+1}(\theta^{t+1}) \le F^t(\theta^{t+1}) + \delta_t. \tag{56}$$

Summing Eq. (55) over communication rounds, using $\sum_{t=0}^{T-2}\delta_t \le V_\Psi$, and lower bounding the final objective by F^* gives

$$\frac{\gamma}{2}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F^t(\theta^t)\|^2 \le F^0(\theta^0) - F^* + V_\Psi + \frac{TL\gamma^2\sigma^2}{E}\sum_{k=1}^{K} p_k^2 + \frac{5T}{12}L^2\gamma^3 G^2. \tag{57}$$

Dividing both sides by $\gamma T/2$ yields Eq. (47). □

B.3 Asymptotic Rate

For balanced client weights $p_k = 1/K$, set

$$\eta = \frac{1}{8LE\sqrt{T}}, \qquad \gamma = \frac{1}{8L\sqrt{T}}. \tag{58}$$

Then Theorem 1 becomes

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F^t(\theta^t)\|^2 \le \frac{16L\left(F^0(\theta^0) - F^* + V_\Psi\right) + \sigma^2/(4KE)}{\sqrt{T}} + \frac{5G^2}{384T}. \tag{59}$$

Therefore,

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F^t(\theta^t)\|^2 = \mathcal{O}(T^{-1/2}). \tag{60}$$

This is the same asymptotic first-order stationarity rate as non-convex FedAvg under full participation. When the semantic terms are removed and $V_\Psi = 0$, the bound reduces to the conventional FedAvg form. Semantic alignment, replay, and bridge refreshing affect the constants but not the dependence on T. Thus, SBFL preserves the asymptotic convergence rate of the underlying FedAvg procedure.

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 SBFL transfers offline VLM semantics into heterogeneous FL through text-verified semantic banks, a server-side semantic bridge, and local semantic alignment/replay. These claims are supported by the method in Section 3 and the experiments under label-skewed and domain-shifted FL 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: Section 3.6 discusses limitations and broader applicability, including reliance on class-name prompts and CLIP coverage, additional computation for bridge training and replay, and privacy boundaries of shared semantic features. Appendix A.7, Appendix A.13, and Appendix A.14 provide further privacy, computation, and storage discussions.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A].

Justification: The paper does not present formal theorem-level theoretical results or proof-based claims. The mathematical expressions in the main paper define the proposed objectives, semantic-bank construction, prototype alignment, replay loss, and training procedure. We additionally provide a complexity analysis in the appendix, including computation, storage, and communication complexity, to clarify the overhead of SBFL relative to standard FL.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes].

Justification: Section 3 provides the full algorithmic pipeline, including semantic-bank construction, semantic bridge learning, mixed prototype alignment, semantic replay, and the local objective. Section 4.1 and the appendix provide the datasets, client data partitions, model backbones, optimizer, hyperparameters, communication rounds, and other implementation details needed to reproduce the main results.

Guidelines:

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

the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes].

   Justification: The paper uses public benchmark datasets. The code will be released upon publication.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes].

   Justification: Section 4.1 specifies the evaluated datasets, Dirichlet label-skew settings, domain-shift setup, model backbones, optimizer, learning rate, local epochs, communication rounds, batch size, and key loss weights. The appendix further provides client data distributions, split details, semantic-bank construction settings, and additional implementation details needed to reproduce the reported experiments.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes].

Justification: Please refer to Sec.4.1 in the main text.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes].

   Justification: The appendix reports the hardware resources used for the experiments, including the GPU type, GPU memory, CPU, RAM, and software environment. These details, together with the released code and configuration files, provide the necessary compute-resource context for reproducing the reported experiments.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes].

   Justification: The research uses standard public image classification benchmarks and studies an algorithmic method for federated learning. It does not involve human subjects, collection of private user data, or deployment in high-stakes decision-making settings.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes].

    Justification: Section 5 discusses the potential benefit of reducing raw-data sharing in federated learning as well as potential privacy concerns of semantic features under stronger threat models. It also mentions possible future combinations with differential privacy, secure aggregation, or feature anonymization.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A].

    Justification: The paper does not release a new high-risk dataset, foundation model, image generator, scraped dataset, or user-facing model. Experiments are conducted on standard public image classification benchmarks.

    Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes].

Justification: The existing assets used in this paper has been adequately cited or credited to.

Guidelines:

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

13. **New assets**

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

Answer: [Yes].

Justification: We have documented the experiment details,In addition, we provide our code with clear details and examples.

Guidelines:

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

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

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

Answer: [N/A].

Justification: The paper does not involve crowdsourcing, human-subject studies, user studies, or participant compensation.

Guidelines:

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

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

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

Answer: [N/A].

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

16. **Declaration of LLM usage**

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

Answer: [N/A].

Justification: The core method uses CLIP-derived vision-language semantic priors but does not use an LLM as an important, original, or non-standard component of the algorithm. Any writing or formatting assistance, if used, does not affect the scientific methodology, experiments, or originality of the research.

Guidelines:

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