GEOPROTOFL: ARE CLASS PROTOTYPES GEOMETRICALLY AGGREGATABLE IN HETEROGENEOUS FEDERATED LEARNING?

Anonymous authors
Paper under double-blind review

ABSTRACT

Prototype-based heterogeneous federated learning (HtFL) enables clients with different model architectures to collaborate through class-prototype exchange instead of parameter aggregation. Existing methods commonly map client features to a shared prototype dimension and directly aggregate the resulting prototype coordinates, implicitly relying on their coordinate systems being comparable across clients. However, independently trained feature extractors across heterogeneous clients may represent equivalent class structures in different client-specific coordinate frames. Consequently, direct aggregation of unregistered prototypes may distort the resulting global prototypes. To address this issue, we introduce GeoProtoFL, a geometry-aware method designed to make heterogeneous class prototypes geometrically aggregatable. Each client learns a local bi-projection with equiangular tight frame (ETF) anchoring to map pooled backbone features onto a shared spherical prototype interface. The server registers each client prototype configuration to a server reference configuration through closed-form Procrustes registration performing gauge registration, then computes each global prototype as a class-wise Fréchet barycenter of the registered prototypes intrinsic aggregation. GeoProtoFL realizes geometric aggregatability through gauge-consistent server-side prototype construction and intrinsic consensus under the spherical metric defined by the shared interface, without requiring public data or homogeneous client architectures. Across standard benchmarks, it consistently improves over representative HtFL baselines, achieving gains of up to 2.61% over the strongest baseline in a given setting while remaining robust across varying statistical heterogeneity, model heterogeneity, and federation-scale settings.

1 INTRODUCTION

Federated learning (FL) enables multiple clients to collaboratively train machine learning models while keeping their data decentralized (McMahan et al., 2017). Most conventional FL algorithms assume that clients share the same model architecture, allowing the server to aggregate model parameters or gradients. In many practical deployments, however, clients may differ in data distributions (Lu et al., 2024; Kundroo et al., 2025), computational resources, and model architectures (Ye et al., 2023; Haider et al., 2026). Heterogeneous federated learning (HtFL) (Zhang et al., 2025a;b) addresses this setting by allowing architecturally diverse clients to participate in collaboration. This model heterogeneity makes parameter aggregation ill-defined, motivating architecture-agnostic forms of knowledge exchange.

Prototype-based HtFL (Tan et al., 2022; Zhu et al., 2023) enables architecture-agnostic collaboration by having each client summarize its local representation space with class prototypes, computed as class-wise feature centroids, in place of model parameters. Each client uploads prototypes only for classes present in its local data, and the server aggregates the prototypes for each class into a global prototype, which is returned to the corresponding clients as semantic guidance for local training. Since prototypes are architecture-independent, this interface is compact and naturally compatible with heterogeneous architectures.

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

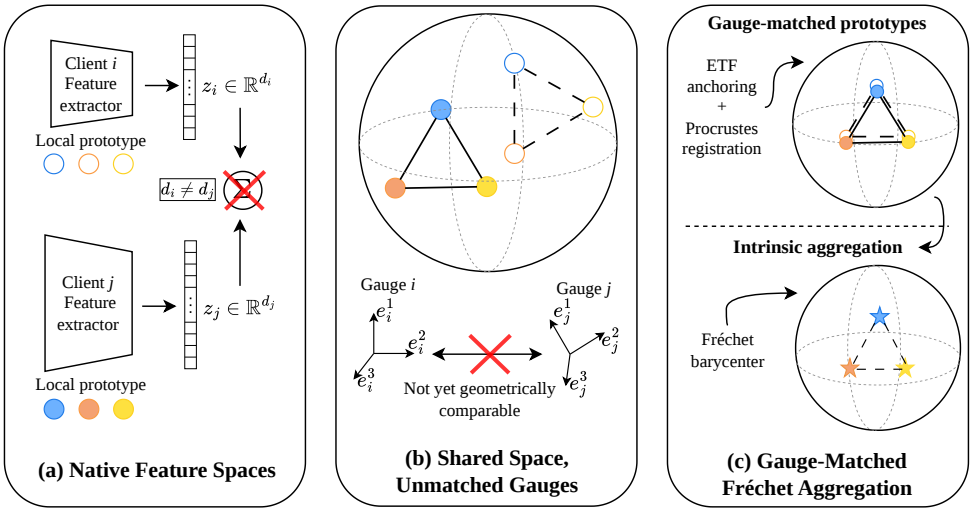

Figure 1: Prototype aggregation in HtFL. (a) Clients i and j extract features in distinct native spaces, so their prototypes cannot be aggregated directly. (b) Projection onto $\mathbb{S}^{d-1}$ provides a common dimension and scale, but each client may express the same class structure under its own gauge. (c) GeoProtoFL registers each client's prototypes to a common gauge and combines them class-wise by their Fréchet barycenter on the hypersphere.

Prototype exchange requires a common feature dimension. As illustrated in Figure 1a, prototypes from incompatible feature spaces cannot be aggregated directly. Existing methods commonly address this interface-level requirement by pooling or mapping client features to a common dimension (Zhang et al., 2024; 2025a), making prototypes communicable across heterogeneous clients. Building on this interface, prior work improves prototype-based collaboration through prototype refinement, inter-class separation, local prototype guidance, and cross-client structural alignment (Zhang et al., 2024; Guo et al., 2025; Wu et al., 2026; Casado-Diez et al., 2026).

However, server-side aggregation ultimately combines prototype coordinates, which requires the prototypes to be expressed in a compatible coordinate frame. A shared dimension and unit scale makes prototypes communicable but not necessarily comparable, because it does not by itself fix such a frame. Independently trained heterogeneous encoders may represent similar class structures under different client-specific orthogonal transformations, which we refer to as *gauges*. As illustrated in Figure 1b, directly combining unregistered prototypes can therefore make global prototypes depend on client-specific coordinates and distort their directions or inter-class geometry. This effect can be aggravated by label skew because each global class prototype is estimated from a different subset of clients. These observations motivate the notion of *geometric aggregatability*. Beyond a common prototype interface, it requires *gauge consistency*, meaning that the global prototype set is invariant, up to a single gauge common to the federation, to the client-specific gauges in which local prototypes are expressed, and *intrinsic consensus*, meaning that client prototypes are combined by minimizing the same geodesic criterion that guides local training.

To achieve geometric aggregatability, we propose GeoProtoFL, a geometry-aware prototype framework for HtFL. GeoProtoFL first constructs a shared spherical prototype interface and then performs gauge-registered intrinsic aggregation of local class prototypes, as illustrated in Figure 1c. Each client learns a bi-projection that maps its pooled backbone features to a shared d-dimensional unit hypersphere. The projection encoder establishes a common prototype dimension and unit-norm features, while the reconstruction decoder encourages information retention from the pooled backbone features. ETF anchoring further encourages a coarse shared class configuration without public data. Residual client-specific gauges may remain after spherical projection. The server therefore estimates a client-specific orthogonal transformation through closed-form Procrustes registration against the current global prototype set, initialized by the ETF anchors. It then computes each global prototype as the class-wise Fréchet barycenter of the registered local prototypes on the hypersphere. The resulting

global prototypes are expressed in the corresponding client frames and used for prototype-guided local training. By separating the construction of a common prototype interface from server-side geometric aggregation, GeoProtoFL enables gauge-consistent global prototypes despite client-specific coordinate frames.

Our contributions are summarized as follows:

- We identify an underexplored requirement in prototype-based HtFL. Matching class labels and feature dimensions makes prototypes communicable but not necessarily comparable, since heterogeneous encoders may express the same class structure in different client-specific gauges. Without gauge registration, coordinate-level aggregation can therefore depend on arbitrary client frames. We formalize this requirement as *geometric aggregatability*, which combines a common prototype interface with gauge consistency and intrinsic consensus.
- We propose GeoProtoFL, a geometry-aware framework for prototype-based HtFL that constructs a public-data-free spherical prototype interface for heterogeneous clients. A spherical bi-projection with ETF anchoring maps heterogeneous representations to a shared hypersphere. The server then performs gauge registration by closed-form orthogonal Procrustes and intrinsic aggregation by computing class-wise Fréchet barycenters of the registered prototypes under the same geodesic metric used for local prototype guidance.
- We evaluate GeoProtoFL against representative prototype-based and heterogeneous FL baselines across three vision benchmarks under varying statistical heterogeneity, model heterogeneity, and client participation. We further conduct ablations to isolate the effects of the spherical interface, gauge registration, and intrinsic aggregation.

2 Related work

2.1 Prototype-Based Heterogeneous Federated Learning

FedProto (Tan et al., 2022) established prototype-based HtFL by representing each locally observed class with a prototype and aggregating same-class prototypes at the server. Subsequent methods improve the exchanged prototype knowledge in different ways. FedTGP (Zhang et al., 2024) learns trainable global prototypes using adaptive-margin contrastive learning, while FedORGP (Guo et al., 2025) and FedDOR (Jia et al., 2026) use orthogonality-based regularization or initialization to encourage discriminative prototype geometry. FedSAF (Wu et al., 2026) instead reduces dependence on the local feature basis by aligning inter-class relations, rather than directly matching prototype coordinates, in its client-side guidance objective. Other approaches reconcile heterogeneous representations through data-free knowledge exchange (Guan et al., 2026) or prototype-based adversarial and collaborative learning (Zhang et al., 2025c). For prototype exchange, heterogeneous client features are typically pooled or mapped to a shared dimension (Zhang et al., 2025a; 2024). This shared-dimensional interface makes prototype vectors communicable and coordinate-level aggregation computationally feasible. It does not, however, ensure that coordinates produced by separately trained feature extractors have a common interpretation at the server. This leaves open how the server should construct global prototypes when uploaded configurations retain client-specific gauges, motivating our formulation of geometric aggregatability.

2.2 Representation Geometry and Geometric Aggregation

Representation-similarity studies compare learned feature configurations using criteria invariant to orthogonal transformations (Kornblith et al., 2019). This perspective motivates modeling the residual mismatch between a client prototype configuration and the server reference as an approximate orthogonal transformation rather than arbitrary coordinate noise. Geometry-aware FL has also considered optimization and aggregation on non-Euclidean parameter spaces. RFedSVRG and related methods perform federated optimization over a shared manifold-valued model (Li & Ma, 2023; Xiao et al., 2025; Huang et al., 2026a), while RFedAGS aggregates transported local gradient streams to handle data heterogeneity and partial participation (Huang et al., 2026b). Procrustes alignment has been used to correct cross-round drift caused by non-unique low-rank factorizations (Meng et al., 2026). These methods assume a shared model parameterization, and their geometric difficulty arises from a non-Euclidean parameter space or a non-unique factorization of corresponding model

states. Prototype-based HtFL differs in that clients use separately trained heterogeneous encoders and share no model parameterization. This raises the question of how client-specific prototype configurations can be registered and aggregated to form intrinsic consensus in the shared spherical prototype interface.

3 METHODOLOGY

3.1 PROBLEM SETTING AND OVERVIEW

We consider a federation of N clients coordinated by a central server, with C classes in total. Client i holds a private dataset $\mathcal{D}_i$ with $n_i = |\mathcal{D}_i|$ samples, observes the classes $\mathcal{C}_i \subseteq \{1, \dots, C\}$, and trains a local model $\theta_i = \{\theta_i^g, \theta_i^h\}$, where $g(\cdot; \theta_i^g)$ is the feature extractor and $h(\cdot; \theta_i^h)$ the classifier head, with no requirement for shared architectures across clients. Client i produces representations $\mathbf{z}_i = g(\mathbf{x}; \theta_i^g) \in \mathbb{R}^{d_i}$, where the dimension d_i may vary across clients. The objective minimizes

$$\min_{\theta_1,\dots,\theta_N} \sum_{i=1}^{N} \frac{n_i}{n} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_i} \left[\ell\big(h(g(\mathbf{x}; \theta_i^g); \theta_i^h),\, y\big)\right], \qquad n = \sum_{i=1}^{N} n_i, \tag{1}$$

where clients collaborate only through class prototypes (Tan et al., 2022), which are exchanged in place of model parameters. For a class $c \in \mathcal{C}_i$, let $\mathbf{p}_i^{c,t}$ denote the prototype uploaded by client i at communication round t and $\mathbf{p}^{c,t}$ the corresponding global prototype. A hat, $\hat{\mathbf{p}}_i^{c,t}$, denotes a registered local prototype, and a bar, $\bar{\mathbf{p}}_i^{c,t}$, a global prototype expressed in client i's coordinate frame. We use $\mathcal{S}^t$ for the clients participating at round t and $\mathcal{S}^{c,t} = \{i \in \mathcal{S}^t : c \in \mathcal{C}_i\}$ for those among them that observe class c.

Geometric aggregatability. Prototype aggregation operates on coordinates and therefore presupposes that uploaded prototypes are comparable in a common geometric space. Independently learned encoders need not provide such a space, and we model the residual mismatch between client i and the server reference as an orthogonal transformation $\mathbf{Q}_i \in O(d)$, which we call the client *gauge*. We say that local prototypes are *geometrically aggregatable* when they are expressed through a common prototype interface and can be combined by a gauge-consistent rule that achieves intrinsic consensus. Gauge consistency makes the resulting global prototypes independent of client-specific coordinate choices, while intrinsic consensus ensures aggregation under the spherical metric used for prototype guidance.

Figure 2 illustrates how GeoProtoFL realizes these requirements. Before federated training begins, each client learns a spherical bi-projection from its local data, establishing the common prototype interface (Section 3.2.1). During each federated round, participating clients use the global prototypes from the previous round for local prototype-guided training and upload class prototypes obtained through their fixed bi-projections (Section 3.2.2). The server then registers each client's uploaded prototypes to the current global reference frame and computes class-wise Fréchet barycenters on the hypersphere (Section 3.2.3). The resulting global prototypes are expressed in each client's coordinate frame and used for guidance in its next local update.

3.2 GEOPROTOFL ALGORITHM

3.2.1 SPHERICAL BI-PROJECTION

Before federated training, each client learns a bi-projection $\Phi_i = \{\Phi_i^{\text{enc}}, \Phi_i^{\text{dec}}\}$ that maps its representations into the shared spherical prototype interface of dimension d,

$$\Phi_i^{\text{enc}} : \mathbb{R}^{d_i} \to \mathbb{S}^{d-1}, \qquad \Phi_i^{\text{dec}} : \mathbb{R}^{d} \to \mathbb{R}^{d_i}. \tag{2}$$

The encoder output $\tilde{\mathbf{z}}_i = \Phi_i^{\text{enc}}(\mathbf{z}_i)$ lies on $\mathbb{S}^{d-1} \subset \mathbb{R}^d$, giving all clients a common prototype dimension and unit-norm features despite potentially different d_i. Both maps are lightweight two-layer MLPs (Appendix A.3). This stage runs once on each client, on local data only, keeps the backbone frozen, and involves no communication with the server.

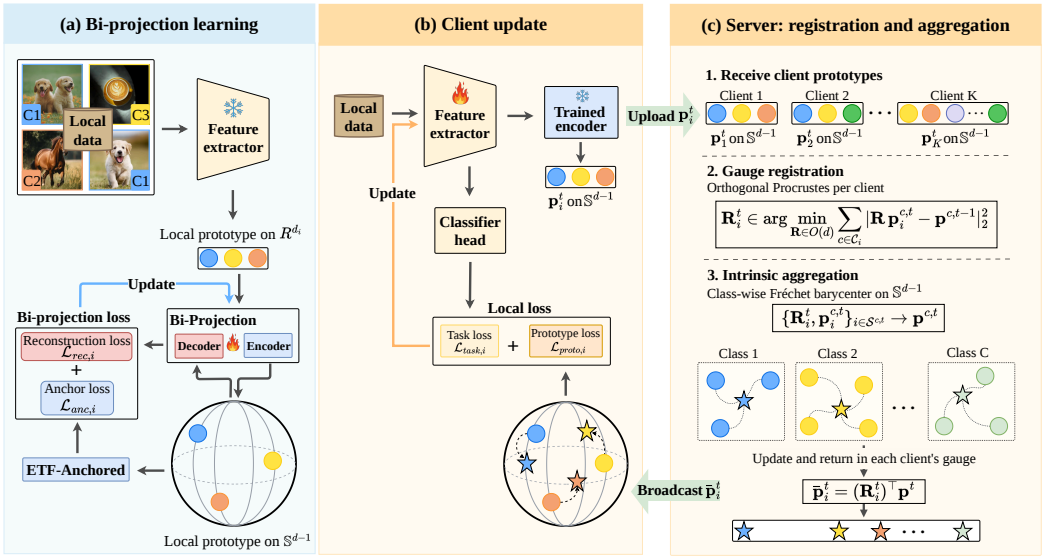

Figure 2: Overview of GeoProtoFL. (a) Each client first learns a spherical bi-projection locally. Its encoder maps client representations to the shared spherical prototype interface, while reconstruction and ETF anchoring encourage information retention and a coarse shared class configuration (Section 3.2.1). (b) In each round, clients train with prototype guidance and upload prototypes $\mathbf{p}_i^t$ computed with the fixed encoder (Section 3.2.2). (c) The server registers each client's prototypes $\mathbf{p}_i^t$ to the current global prototypes $\mathbf{p}^{t-1}$ by orthogonal Procrustes (gauge registration), computes class-wise Fréchet barycenters $\mathbf{p}^t$ of the registered prototypes $\hat{\mathbf{p}}_i^t$ (intrinsic aggregation), and returns them in each client's frame as $\bar{\mathbf{p}}_i^t$ for local guidance in the next round (Section 3.2.3).

To give clients a coarse shared class configuration without public data, the interface is anchored to a fixed class-indexed configuration $\{\boldsymbol{\mu}^c\}_{c=1}^C \subset \mathbb{S}^{d-1}$, constructed as a simplex ETF,

$$[\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^C] = \mathbf{P}\sqrt{\tfrac{C}{C-1}}\left(\mathbf{I}_C - \tfrac{1}{C}\mathbf{1}\mathbf{1}^\top\right), \tag{3}$$

where $\mathbf{P} \in \mathbb{R}^{d\times C}$ is a fixed matrix with orthonormal columns, so that $\langle \boldsymbol{\mu}^c, \boldsymbol{\mu}^{c'}\rangle = -1/(C-1)$ for $c \neq c'$. This construction requires $d \geq C$, and the anchors are identical across clients without communication. Client i trains Φ_i by minimizing

$$\mathcal{L}_i^{\text{proj}} = \lambda_{\text{rec}}\,\mathcal{L}_{\text{rec},i} + \lambda_{\text{anc}}\,\mathcal{L}_{\text{anc},i}, \tag{4}$$

where λ_{rec} and λ_{anc} weight the reconstruction and anchoring terms and, for a mini-batch of size B,

$$\mathcal{L}_{\text{rec},i} = \frac{1}{B}\sum_{b=1}^{B} \left\|\mathbf{z}_{i,b} - \Phi_i^{\text{dec}}(\tilde{\mathbf{z}}_{i,b})\right\|_2^2. \tag{5}$$

The reconstruction term encourages the projected representation to retain information about its input, and the anchoring term is

$$\mathcal{L}_{\text{anc},i} = \frac{1}{|\mathcal{C}_{i,b}|}\sum_{c\in\mathcal{C}_{i,b}} \left(1 - \langle \bar{\tilde{\mathbf{z}}}_i^{\,c}, \boldsymbol{\mu}^c\rangle\right), \tag{6}$$

where $\mathcal{C}_{i,b} \subseteq \mathcal{C}_i$ is the set of classes present in the mini-batch and $\bar{\tilde{\mathbf{z}}}_i^{\,c}$ is the ℓ_2-normalized mean of the projected class-c examples in it. ETF anchoring encourages a coarse shared class configuration, but it is optimized locally over finite and label-skewed client data. Residual client-specific coordinate mismatch can therefore remain, which the server resolves explicitly. After E_{proj} local epochs, the learned projection is fixed and used to construct prototypes during federated training.

3.2.2 PROTOTYPE-GUIDED LOCAL UPDATE

At communication round t, the server selects a set $\mathcal{S}^t$ of participating clients, and each client $i \in \mathcal{S}^t$ receives the previous global prototypes of its local classes expressed in its own coordinate frame, $\bar{\mathbf{p}}_i^{c,t-1}$. Client i then updates its local model θ_i for E epochs by minimizing

$$\mathcal{L}_i^{\text{local}} = \mathcal{L}_{\text{task},i} + \lambda_{\text{proto}} \frac{1}{B} \sum_{b=1}^{B} d_{\mathbb{S}} \left(\Phi_i^{\text{enc}}\big(g(\mathbf{x}_{i,b}; \theta_i^g)\big), \bar{\mathbf{p}}_i^{\,y_{i,b},t-1} \right)^2, \tag{7}$$

where $\mathcal{L}_{\text{task},i}$ is the cross-entropy loss, λ_{proto} weights the guidance term, and $d_{\mathbb{S}}(\mathbf{u}, \mathbf{v}) = \arccos(\mathbf{u}^\top \mathbf{v})$ is the geodesic distance on $\mathbb{S}^{d-1}$. The guidance term compares projected local features and the received target in the same client coordinate frame, under the spherical metric that also defines intrinsic aggregation. In the first round, where no global prototypes have been aggregated yet, only the task loss is used.

After local training, client i applies its fixed encoder to its current representations and computes one prototype per locally observed class as the mean of its projected features,

$$\mathbf{p}_i^{c,t} = \frac{1}{n_i^c} \sum_{\mathbf{x} \in \mathcal{D}_i^c} \Phi_i^{\text{enc}}\big(g(\mathbf{x}; \theta_i^{g,t})\big), \tag{8}$$

where $\mathcal{D}_i^c$ is the local subset with label c and $n_i^c = |\mathcal{D}_i^c|$. The client uploads $\{(\mathbf{p}_i^{c,t},\, n_i^c)\}_{c \in \mathcal{C}_i}$ to the server.

3.2.3 GAUGE REGISTRATION AND INTRINSIC AGGREGATION

After receiving the uploaded client prototypes $\mathbf{p}_i^{c,t}$, the server registers them to the current global reference configuration and performs class-wise intrinsic aggregation on the shared hypersphere. The server maintains this reference as a set of global prototypes $\{\mathbf{p}^{c,t}\}_{c=1}^{C}$, initialized with the ETF anchors as $\mathbf{p}^{c,0} = \boldsymbol{\mu}^c$.

Gauge registration. The server estimates the gauge of each client $i \in \mathcal{S}^t$ by orthogonal Procrustes registration of its uploaded prototypes against the current global prototypes,

$$\mathbf{R}_i^t \in \arg\min_{\mathbf{R} \in O(d)} \sum_{c \in \mathcal{C}_i} \left\| \mathbf{R}\, \mathbf{p}_i^{c,t} - \mathbf{p}^{c,t-1} \right\|_2^2. \tag{9}$$

A solution follows from the singular value decomposition

$$\mathbf{H}_i^t = \sum_{c \in \mathcal{C}_i} \mathbf{p}^{c,t-1} \big(\mathbf{p}_i^{c,t}\big)^\top = \mathbf{U}_i^t \boldsymbol{\Sigma}_i^t \big(\mathbf{V}_i^t\big)^\top, \qquad \mathbf{R}_i^t = \mathbf{U}_i^t \big(\mathbf{V}_i^t\big)^\top. \tag{10}$$

Registration uses all client classes jointly, since a single class is insufficient to identify its gauge (Proposition 1). The single $\mathbf{R}_i^t$, which may include a reflection, is applied to all prototypes of client i, giving the registered prototypes $\hat{\mathbf{p}}_i^{c,t} = \text{Norm}(\mathbf{R}_i^t \mathbf{p}_i^{c,t})$ with $\text{Norm}(\mathbf{u}) = \mathbf{u}/\|\mathbf{u}\|_2$.

Intrinsic aggregation. Since clients are guided by the squared geodesic distance in Eq. 7, the server forms each global prototype as the minimizer of the same criterion over the registered prototypes,

$$\mathbf{p}^{c,t} = \arg\min_{\mathbf{q} \in \mathbb{S}^{d-1}} \sum_{i \in \mathcal{S}^{c,t}} w_i^{c,t}\, d_{\mathbb{S}}\big(\mathbf{q}, \hat{\mathbf{p}}_i^{c,t}\big)^2, \qquad w_i^{c,t} = \frac{n_i^c}{\sum_{j \in \mathcal{S}^{c,t}} n_j^c}. \tag{11}$$

We approximate this weighted Fréchet barycenter with K_{bar} Karcher iterations warm-started at $\mathbf{p}^{c,t-1}$ (Appendix A.1), and classes without participating clients keep their previous prototype. The updated $\mathbf{p}^{c,t}$ and $\mathbf{R}_i$ define the client-frame targets that client i receives at its next participation (Section 3.2.2). Appendix A.2 gives the conditions under which this aggregation is gauge consistent and bounds the error of the global prototypes under an approximate orthogonal-gauge model.

4 EXPERIMENTS

4.1 EXPERIMENT SETUP

We implement GeoProtoFL and all baselines in HtFLlib (Zhang et al., 2025a), so every method uses the same data partitions, client architectures, and training protocol. Appendix A.4 gives the remaining implementation details and all hyperparameters.

Datasets and statistical heterogeneity. We use CIFAR-10, CIFAR-100 (Krizhevsky, 2009), and Tiny-ImageNet (Le & Yang, 2015) under two label-skew protocols. In the pathological setting (McMahan et al., 2017), each client holds 2, 10, or 20 classes on the three datasets, respectively. In the practical setting (Lin et al., 2020), class proportions on each client follow a Dirichlet distribution with $\alpha = 0.1$. Each client splits its local data into 75% training and 25% test data.

Model heterogeneity settings. Main results use the $\text{HtFE}^{\text{img}}8$ pool of HtFLlib, which assigns a 4-layer CNN, GoogLeNet, MobileNetV2, and ResNet-18/34/50/101/152 to clients in round-robin order. The backbones differ in depth, width, and parameter count, and following the HtFLlib protocol, their features are average-pooled to a common dimension $d_i = 512$. GeoProtoFL maps these features to an interface of dimension $d = 128$ on CIFAR-10/100 and $d = 256$ on Tiny-ImageNet. Robustness experiments further vary the architecture pool and the post-pooling dimension.

Baselines and training protocol. We compare with Local training and seven HtFL methods grouped by what clients share: partial model parameters (LG-FedAvg (Liang et al., 2020), FedGH (Yi et al., 2023)), intermediate representations (FedRE (Yao et al., 2026)), and class prototypes (FedProto (Tan et al., 2022), FedTGP (Zhang et al., 2024), FedORGP (Guo et al., 2025), FedSAF (Wu et al., 2026)). All methods use $N = 20$ clients with full participation ($\rho = 1$), 300 communication rounds, and 5 local epochs per round, and GeoProtoFL keeps the same hyperparameters in all experiments apart from d. All results are averaged over three random seeds and reported as mean $\pm$ standard deviation.

4.2 MAIN RESULTS

Table 1 reports test accuracy across three vision datasets under pathological and practical non-IID partitions. GeoProtoFL achieves the highest mean accuracy in all six settings, with the largest gains over FedSAF on CIFAR-100, where it exceeds FedSAF by 4.16% and 4.15%, reaching 62.85% and 47.90%, respectively. It also exceeds FedSAF by 1.69% and 2.61% on Tiny-ImageNet and ranks first on CIFAR-10, reaching 87.21% and 83.34%. Compared with FedORGP, GeoProtoFL improves CIFAR-100 accuracy by 2.30% and 1.78%, while its gain on Tiny-ImageNet increases from 3.93% to 4.88% under the practical partition.

The comparison with prototype-based methods clarifies the role of the proposed aggregation rule. FedORGP and FedSAF improve prototype targets through orthogonality regularization and relational alignment, respectively, whereas GeoProtoFL changes how server-side prototypes are constructed. In particular, it registers client prototype configurations before forming intrinsic consensus on the shared hypersphere. Compared with FedProto, which directly aggregates prototype coordinates without registration, GeoProtoFL improves by 4.17% on CIFAR-10 and by 8.69% on CIFAR-100 under the practical setting. Section 4.5 examines how this gain divides among the spherical interface, gauge registration, and intrinsic aggregation.

4.3 ROBUSTNESS TO STATISTICAL HETEROGENEITY AND CLIENT SCALABILITY

Statistical heterogeneity. We vary the number of classes per client under pathological partitioning and the concentration parameter α under Dirichlet partitioning on CIFAR-100. Table 2 shows that GeoProtoFL achieves the highest mean accuracy in all four settings. With 5 and 20 classes per client, it reaches 77.27% and 45.65%, compared with 75.88% and 44.93% for FedORGP. Under Dirichlet partitioning, it exceeds FedORGP by 1.97% at $\alpha = 0.3$ and 1.44% at $\alpha = 0.01$. The margin is larger relative to FedSAF and FedProto. With 20 classes per client, GeoProtoFL exceeds them by 3.65% and 10.26%, respectively. GeoProtoFL therefore remains the strongest method under both label-skew protocols and across different levels of class coverage per client.

Table 1: Comparison with HtFL baselines. Test accuracy (%) across three datasets in pathological and practical (Dirichlet, α = 0.1) non-IID partitions.

Method	Pathological Setting			Practical Setting		
	CIFAR-10	CIFAR-100	Tiny-ImageNet	CIFAR-10	CIFAR-100	Tiny-ImageNet
Local	83.32±0.21	55.83±0.08	32.14±0.22	80.33±0.20	42.48±0.09	24.00±0.10
LG-FedAvg	83.72±0.07	56.06±0.28	31.98±0.15	79.73±0.03	42.01±0.24	24.71±0.06
FedGH	84.28±0.04	55.81±0.15	32.07±0.16	79.28±0.09	40.91±0.28	23.78±0.24
FedProto	80.98±0.43	53.94±0.26	30.73±0.18	79.17±0.20	39.21±0.67	20.78±0.04
FedTGP	83.96±0.02	54.87±0.45	31.20±0.10	80.97±0.20	43.16±0.29	25.05±0.09
FedRE	83.61±0.10	55.56±0.24	32.06±0.10	80.63±0.20	42.30±0.21	24.25±0.27
FedORGP	85.59±0.11	60.55±0.10	31.60±0.24	81.10±0.19	46.12±0.19	23.78±0.28
FedSAF	86.02±0.17	58.69±0.52	33.84±0.18	81.92±0.32	43.75±0.02	26.05±0.22
GeoProtoFL	**87.21±0.14**	**62.85±0.24**	**35.53±0.13**	**83.34±0.25**	**47.90±0.15**	**28.66±0.34**

Table 2: Robustness to statistical heterogeneity and federation scale. Test accuracy (%) on CIFAR-100 with HtFE^{img}8. Scalability settings vary the number of clients N and participation ratio ρ.

Method	**Distribution Heterogeneity**				**Scalability**		
	Pathological Setting		Practical Setting		$N = 40$	$N = 80$	$N = 100$
	5 cls/client	20 cls/client	$\alpha = 0.3$	$\alpha = 0.01$	$\rho = 50\%$	$\rho = 25\%$	$\rho = 20\%$
Local	71.25±0.30	37.38±0.33	28.73±0.23	63.68±0.26	38.36±0.33	35.70±0.26	35.89±0.21
LG-FedAvg	71.92±0.13	36.98±0.10	27.99±0.21	64.05±0.07	37.73±0.20	34.84±0.06	34.20±0.14
FedGH	71.97±0.29	37.10±0.33	27.06±0.40	62.20±0.19	36.46±0.23	32.30±0.13	32.19±0.41
FedProto	70.60±0.04	35.39±0.35	25.77±0.20	60.13±0.11	33.85±0.27	25.16±0.89	21.40±0.63
FedTGP	70.59±0.22	35.93±0.30	28.19±0.25	63.46±0.40	37.95±0.47	34.71±0.26	34.21±0.22
FedRE	71.28±0.25	37.09±0.83	28.37±0.10	63.83±0.16	38.64±0.08	35.82±0.11	36.06±0.21
FedORGP	75.88±0.59	44.93±0.16	30.75±0.07	68.03±0.51	41.97±0.16	37.27±0.30	37.28±0.23
FedSAF	74.47±0.15	42.00±0.27	29.09±0.13	66.52±0.40	38.99±0.34	34.24±0.23	34.23±0.17
GeoProtoFL	**77.27±0.09**	**45.65±0.13**	**32.72±0.03**	**69.47±0.32**	**43.26±0.16**	**37.74±0.14**	**37.35±0.38**

Client scalability. Table 2 reports the results as we increase the federation size from $N = 40$ to $N = 100$ while reducing the participation ratio from 50% to 20%. GeoProtoFL remains first at every scale, reaching 43.26% at $N = 40$ and 37.74% at $N = 80$, compared with 41.97% and 37.27% for FedORGP. At $N = 100$, the two methods remain within one standard deviation, achieving 37.35% and 37.28%, respectively. In contrast, FedProto is the most sensitive to scale, dropping from 33.85% to 21.40% and widening its gap with GeoProtoFL from 9.41% to 15.95%.

4.4 Robustness to Model Heterogeneity and Feature Dimension

Model heterogeneity. We replace the HtFE^{img}8 pool with three alternatives. HtFE^{img}4 uses fewer and smaller backbones, HtM^{img}10 adds two vision transformers, and HtFE^{img}8-HtC^{img}4 additionally varies the classifier head across clients. GeoProtoFL achieves the highest mean accuracy on all three pools in Table 3. It reaches 48.39% on HtFE^{img}4 and 48.93% on HtFE^{img}8-HtC^{img}4, outperforming FedORGP by 1.59% and 2.06%, respectively, and achieves 46.09% on HtM^{img}10, compared with 45.50% for FedORGP. LG-FedAvg and FedGH are not applicable to the latter two pools because their partial-parameter aggregation requires compatible tensor shapes across clients. GeoProtoFL has no such requirement, since clients exchange only prototypes in the shared spherical interface.

Feature dimension. We vary the shared post-pooling feature dimension $d_i \in \{128, 256, 1024\}$ on CIFAR-100 under Dirichlet partitioning with $\alpha = 0.1$ and $N = 20$, while fixing GeoProtoFL's interface dimension at $d = 128$. As shown in Table 3, GeoProtoFL ranks first in all settings, obtaining 47.53%, 47.13%, and 47.67%, respectively, while prototype-based baselines vary more across d_i. Its performance remains stable because registration and intrinsic aggregation operate in the fixed d-dimensional spherical interface rather than the backbone feature space. Consequently, both the communicated prototype dimension and the aggregation geometry are independent of the post-pooling backbone dimension.

Table 3: Robustness to model heterogeneity and feature dimension. Test accuracy (%) on CIFAR-100 (Dirichlet, $\alpha = 0.1$, $N = 20$).

Method	Model Heterogeneity			Feature Dimension		
	$\text{HtFE}^{\text{img}}4$	$\text{HtM}^{\text{img}}10$	$\text{HtFE}^{\text{img}}8\text{-HtC}^{\text{img}}4$	$d_i = 128$	$d_i = 256$	$d_i = 1024$
Local	45.53±0.29	41.83±0.23	41.65±0.47	41.19±0.32	41.65±0.19	42.77±0.05
LG-FedAvg	45.41±0.33	–	–	41.19±0.27	41.53±0.33	42.60±0.26
FedGH	44.41±0.16	–	–	40.24±0.23	40.37±0.07	40.89±0.18
FedProto	40.04±0.62	37.49±0.32	36.64±0.76	35.53±0.03	37.46±0.12	39.91±0.18
FedTGP	46.64±0.27	42.19±0.36	43.25±0.01	41.80±0.07	41.86±0.03	43.12±0.03
FedRE	45.59±0.09	41.63±0.12	41.66±0.39	41.65±0.15	41.66±0.14	42.61±0.02
FedORGP	46.80±0.24	45.50±0.22	46.87±0.31	45.28±0.28	45.52±0.13	45.97±0.14
FedSAF	45.98±0.26	42.91±0.22	44.94±0.16	45.50±0.23	44.51±0.23	42.76±0.37
GeoProtoFL	**48.39±0.06**	**46.09±0.27**	**48.93±0.06**	**47.53±0.22**	**47.13±0.07**	**47.67±0.14**

Table 4: Ablation of GeoProtoFL components. Test accuracy (%) with $\text{HtFE}^{\text{img}}8$ (Dirichlet, $\alpha = 0.1$, $N = 20$). Mean denotes the unnormalized mean, and Euclidean and Fréchet the Fréchet barycenter of Eq. 11

	Interface	Registration	Aggregation	CIFAR-10	CIFAR-100	Tiny-ImageNet
FedProto	–	–	Mean	79.17±0.20	39.21±0.67	20.78±0.04
V1	✓	–	Euclidean	82.38±0.17	46.74±0.28	28.23±0.16
V2	✓	✓	Euclidean	82.70±0.23	47.49±0.18	28.14±0.14
V3	✓	–	Fréchet	82.72±0.20	47.49±0.30	28.38±0.12
V4 (GeoProtoFL)	✓	✓	Fréchet	**83.34±0.25**	**47.90±0.15**	**28.66±0.34**

4.5 Ablation Study

Table 4 ablates the spherical prototype interface, gauge registration, and intrinsic aggregation under the practical setting of Table 1, each of which targets one requirement of geometric aggregatability. FedProto serves as the baseline that uses none of the three components, directly averaging unregistered prototypes in the post-pooling feature space. V1 introduces the shared spherical interface by mapping client representations onto a common d-dimensional unit hypersphere, which yields the largest improvement over FedProto. V2 adds gauge registration before Euclidean aggregation and yields dataset-dependent changes, whereas V3 replaces Euclidean averaging with Fréchet aggregation and improves over V1 on every dataset. V4, the full GeoProtoFL, combines registration with Fréchet aggregation and improves over both V2 and V3 on all three datasets, so removing either server-side component lowers accuracy. Registration therefore improves accuracy consistently once consensus is formed by intrinsic aggregation, consistent with geometric aggregatability requiring gauge consistency and intrinsic consensus together. Overall, GeoProtoFL improves over FedProto by 4.17%, 8.69%, and 7.88% on CIFAR-10, CIFAR-100, and Tiny-ImageNet, respectively.

5 Conclusion

GeoProtoFL addresses prototype aggregation in HtFL as a problem of geometric aggregatability. A shared prototype dimension makes client prototypes communicable but not necessarily comparable, since independently trained encoders may express the same class structure in different client-specific gauges. GeoProtoFL constructs a shared spherical prototype interface, registers client prototype configurations to a server reference, and forms global prototypes through intrinsic Fréchet aggregation. Across three image benchmarks, GeoProtoFL achieves the highest accuracy in every setting, improving over the strongest baseline in each setting by up to 2.61%, and retains the highest mean accuracy under varying statistical heterogeneity, federation scale, and model heterogeneity. Gauge registration currently assumes that client configurations differ by an orthogonal transformation, and its advantage narrows under sparse participation. Future work will relax this assumption and extend the evaluation beyond vision.

Reproducibility Statement

Algorithms 1 and 2 in Appendix A.1 specify the bi-projection stage and the federated procedure, including the order of local training, prototype upload, gauge registration, and intrinsic aggregation in each round, the handling of partial participation, and the Karcher update in Eq. 12. The ETF anchors follow Eq. 3, and any choice of $\mathbf{P}$ with orthonormal columns yields the same anchor configuration up to a global orthogonal transformation. Appendix A.3 gives the bi-projection architecture, and Section 4.1 gives the interface dimension for each dataset. Appendix A.2 states the modeling assumption and gives complete proofs of all results, and Appendix A.2.5 lists which results depend on that assumption. Appendix A.4 reports the hardware and software versions, the data split, the architecture pools and their assignment to clients (Table 8), and the hyperparameters of every method (Table 7). All methods are implemented in HtFLlib, methods included in its public release use their default method-specific hyperparameters, and FedORGP and FedSAF use the hyperparameters reported in their original publications. Section 4.1 specifies the partition protocols and the numbers of clients, rounds, and local epochs. Test accuracy is the average of the per-client test accuracies, and for every method we take the highest test accuracy over communication rounds in each of three seeds and report the mean and standard deviation of these values. The sensitivity study in Appendix A.5 instead reports the highest value of the seed-averaged accuracy curve, with the standard deviation across seeds at that round. The communication costs in Table 5 are analytic per-round message sizes rather than measured network traffic. Source code will be released upon publication.

AI Use Statement

We used a large language model (LLM) as a writing and analysis assistant. It assisted with formulating and revising mathematical claims, drafting and polishing manuscript prose, organizing sections, and checking references. It also provided feedback on research methodology and experiments and assisted with result interpretation. We did not use it to generate synthetic datasets, develop theoretical models or conceptual frameworks, propose or refine hypotheses, or implement methods. It was not used for translation, dataset cleaning or reformatting, or qualitative and thematic data analysis. Its use in coding was limited to code review, bug identification, and bug fixing. All AI-assisted work was reviewed by the authors. AI-assisted mathematical proofs were independently checked against the stated assumptions and derivations, and every reported result was verified against the corresponding experiment logs. We take responsibility for the final content of this work, including the text, claims, and artifacts produced with generative AI assistance.

References

Bijan Afsari. Riemannian L^p center of mass: existence, uniqueness, and convexity. *Proceedings of the American Mathematical Society*, 139(2):655–673, 2011.

Mario Casado-Diez, Alejandro Dopico-Castro, Verónica Bolón-Canedo, and Bertha Guijarro-Berdiñas. Closing the alignment-maturity gap in federated prototype learning. *arXiv preprint arXiv:2606.02172*, 2026.

Haowen Guan, Xuan Zhao, Chengjie Zheng, Allen Yang, Tales Imbiriba, and Shichao Pei. Aggregation-free heterogeneous federated learning with data-free knowledge exchange. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=BLQ9JulhQm.

Fucheng Guo, Zeyu Luan, Qing Li, Dan Zhao, and Yong Jiang. Fedorgp: Guiding heterogeneous federated learning with orthogonality regularization on global prototypes. *arXiv preprint arXiv:2502.16119*, 2025.

Ghani Haider, Majid Kundroo, Leo Zhang, Jinchul Choi, and Taehong Kim. Fedcsga: Evolutionary client selection with joint statistical and system heterogeneity in federated learning. *Journal of Systems Architecture*, pp. 103813, 2026.

Zhenwei Huang, Wen Huang, Pratik Jawanpuria, and Bamdev Mishra. Federated learning on riemannian manifolds with differential privacy. *Machine Learning*, 115(4):84, 2026a.

Zhenwei Huang, Wen Huang, Pratik Jawanpuria, and Bamdev Mishra. Riemannian federated learning via averaging gradient streams. In *International Conference on Learning Representations*, volume 2026, pp. 20377–20415, 2026b.

Hongjie Jia, Nie Jiaxin, Di Wu, Huan Wang, Jun Bai, Longxia Huang, and Wei Xiang. FedDOR: Orthogonal initialization and dual regularization for prototype integrity in heterogeneous federated learning, 2026. URL https://openreview.net/forum?id=daVRWlIX0Z.

Hermann Karcher. Riemannian center of mass and mollifier smoothing. *Communications on Pure and Applied Mathematics*, 30(5):509–541, 1977.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMlR, 2019.

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

Majid Kundroo, Tinku Singh, and Taehong Kim. Fedlbw: A loss-based weighting strategy for federated learning on non-iid data in wireless networks. *Expert Systems with Applications*, pp. 130487, 2025.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS231N*, 7(7):3, 2015.

Jiaxiang Li and Shiqian Ma. Federated learning on Riemannian manifolds. *Applied Set-Valued Analysis and Optimization*, 5(2):213–232, 2023.

Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B Allen, Randy P Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020.

Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in neural information processing systems*, 33:2351–2363, 2020.

Zili Lu, Heng Pan, Yueyue Dai, Xueming Si, and Yan Zhang. Federated learning with non-iid data: A survey. *IEEE Internet of Things Journal*, 11(11):19188–19209, 2024.

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

Chuiyang Meng, Ming Tang, and Vincent WS Wong. Florg: Federated fine-tuning with low-rank gram matrices and procrustes alignment. *arXiv preprint arXiv:2602.17095*, 2026.

Peter H Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966.

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, pp. 8432–8440, 2022.

Xinghao Wu, Jianwei Niu, Guogang Zhu, Xuefeng Liu, Shaojie Tang, and Jiayuan Zhang. From coordinate matching to structural alignment: Rethinking prototype alignment in heterogeneous federated learning. *arXiv preprint arXiv:2605.05959*, 2026.

He Xiao, Tao Yan, and Shimin Zhao. Riemannian svrg with barzilai-borwein scheme for federated learning. *Journal of Industrial and Management Optimization*, 21(2):1546–1567, 2025.

Yuan Yao, Lixu Wang, Jiaqi Wu, Jin Song, Simin Chen, Zehua Wang, Zijian Tian, Wei Chen, Huixia Li, and Xiaoxiao Li. Fedre: A representation entanglement framework for model-heterogeneous federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39466–39475, 2026.

Mang Ye, Xiuwen Fang, Bo Du, Pong C Yuen, and Dacheng Tao. Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Computing Surveys*, 56(3):1–44, 2023.

Liping Yi, Gang Wang, Xiaoguang Liu, Zhuan Shi, and Han Yu. Fedgh: Heterogeneous federated learning with generalized global header. In *Proceedings of the 31st ACM international conference on multimedia*, pp. 8686–8696, 2023.

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, pp. 16768–16776, 2024.

Jianqing Zhang, Xinghao Wu, Yanbing Zhou, Xiaoting Sun, Qiqi Cai, Yang Liu, Yang Hua, Zhenzhe Zheng, Jian Cao, and Qiang Yang. Htfllib: A comprehensive heterogeneous federated learning library and benchmark. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 5900–5911, 2025a.

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 Computer Vision and Pattern Recognition Conference*, pp. 20632–20642, 2025b.

Yong Zhang, Feng Liang, Guanghu Yuan, Min Yang, Chengming Li, and Xiping Hu. Fedpall: prototype-based adversarial and collaborative learning for federated learning with feature drift. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3111–3120. IEEE, 2025c.

Guogang Zhu, Xuefeng Liu, Shaojie Tang, and Jianwei Niu. Aligning before aggregating: Enabling communication efficient cross-domain federated learning via consistent feature extraction. *IEEE Transactions on Mobile Computing*, 23(5):5880–5896, 2023.

A Appendix

A.1 Algorithm and Complexity

GeoProtoFL has two stages. Algorithm 1 learns the spherical bi-projection on each client before federated training, and Algorithm 2 gives the federated procedure. The first stage establishes a common prototype interface with a coarse shared class configuration. It does not remove client gauges, which the server estimates from the prototypes uploaded in every round.

Algorithm 1 Local Spherical Bi-Projection Learning

Require: Local dataset $\{\mathcal{D}_i\}$, initialized backbones $\{\theta_i^g\}$ (kept fixed), interface dimension d, epochs E_{proj}
Ensure: Bi-projections $\{\Phi_i\}$ with fixed encoders $\{\Phi_i^{\mathrm{enc}}\}$

```
Compute ETF anchors {μ^c}_{c=1}^C from (C, d, P) via Eq. 3
for each client i in parallel do
    for e = 1, ..., E_proj do
        for each mini-batch from D_i do
            Compute L_i^proj via Eq. 4 and update Φ_i
        end for
    end for
end for
return {Φ_i}
```

In the first round of FL (Algorithm 2), no global prototypes have been aggregated yet, so local training uses only the task loss. From the second round on, each participating client trains toward global prototypes expressed in its own frame and then uploads prototypes computed with its updated backbone, which the server registers and aggregates. Under full participation, client i therefore trains in round $t+1$ toward $(\mathbf{R}_i^t)^\top \mathbf{p}^{c,t}$, which is the client-frame target $\bar{\mathbf{p}}_i^{c,t}$ of Section 3.2.3. Under partial participation, $\mathbf{R}_i$ is the gauge estimated at the most recent round in which client i participated, and it is the identity before the first participation.

Algorithm 2 GeoProtoFL Federated Training

Require: Fixed encoders $\{\Phi_i^{\mathrm{enc}}\}$ from Algorithm 1, local models $\{\theta_i\}$, rounds T, local epochs E, ETF anchors $\{\boldsymbol{\mu}^c\}$, Karcher iterations K_{bar}
Ensure: Local models $\{\theta_i\}$ and global prototypes $\{\mathbf{p}^{c,T}\}_{c=1}^{C}$
Server initializes $\mathbf{p}^{c,0} \leftarrow \boldsymbol{\mu}^c$ for all c and $\mathbf{R}_i \leftarrow \mathbf{I}_d$ for all i
for $t = 1, \dots, T$ **do**
  Server selects participating clients $\mathcal{S}^t \subseteq \{1, \dots, N\}$
  if $t > 1$ **then**
    Server sends $\bar{\mathbf{p}}_i^{c,t-1} = \mathbf{R}_i^\top \mathbf{p}^{c,t-1}$ for $c \in \mathcal{C}_i$ to each client $i \in \mathcal{S}^t$ ▷ Client-frame targets
  end if
  for each client $i \in \mathcal{S}^t$ **in parallel do** ▷ Prototype-guided local update (clients)
    Update θ_i for E epochs by minimizing $\mathcal{L}_i^{\mathrm{local}}$ in Eq. 7 (task loss only at $t = 1$)
    Compute $\{\mathbf{p}_i^{c,t}\}_{c \in \mathcal{C}_i}$ via Eq. 8 and upload them with the class counts $\{n_i^c\}_{c \in \mathcal{C}_i}$
  end for
  for each client $i \in \mathcal{S}^t$ **do** ▷ Gauge registration (server)
    Compute $\mathbf{R}_i^t$ via Eq. 10, set $\mathbf{R}_i \leftarrow \mathbf{R}_i^t$, and set $\hat{\mathbf{p}}_i^{c,t} \leftarrow \mathrm{Norm}(\mathbf{R}_i^t \mathbf{p}_i^{c,t})$ for $c \in \mathcal{C}_i$
  end for
  for each class c **do** ▷ Intrinsic aggregation (server)
    if $\mathcal{S}^{c,t} \neq \varnothing$ **then**
      Compute $\mathbf{p}^{c,t}$ via Eq. 11 with K_{bar} iterations of Eq. 12 from $\mathbf{p}^{c,t-1}$
    else
      $\mathbf{p}^{c,t} \leftarrow \mathbf{p}^{c,t-1}$
    end if
  end for
end for

Karcher iteration. For class c, let $\mathbf{q}_0 = \mathbf{p}^{c,t-1}$. Each iteration moves $\mathbf{q}_k$ along the weighted mean of the logarithmic maps of the registered prototypes,

$$\mathbf{v}_k = \sum_{i \in \mathcal{S}^{c,t}} w_i^{c,t} \operatorname{Log}_{\mathbf{q}_k}\big(\hat{\mathbf{p}}_i^{c,t}\big), \qquad \mathbf{q}_{k+1} = \operatorname{Exp}_{\mathbf{q}_k}(\mathbf{v}_k), \tag{12}$$

where $\operatorname{Log}_{\mathbf{q}}(\mathbf{p}) = \frac{\vartheta}{\sin\vartheta}\big(\mathbf{p} - \cos\vartheta\,\mathbf{q}\big)$ with $\vartheta = d_{\mathbb{S}}(\mathbf{q}, \mathbf{p})$, and $\operatorname{Exp}_{\mathbf{q}}(\mathbf{v}) = \cos\|\mathbf{v}\|_2\,\mathbf{q} + \sin\|\mathbf{v}\|_2\,\mathbf{v}/\|\mathbf{v}\|_2$. We use a unit step and set $\mathbf{p}^{c,t} = \mathbf{q}_{K_{\mathrm{bar}}}$, an approximate minimizer of Eq. 11 whose quality is certified by Corollary 2 when the registered prototypes are concentrated. We set $\operatorname{Log}_{\mathbf{q}}(\mathbf{p}) = \mathbf{0}$ when $\vartheta = 0$, and in the guidance loss the inner product is clipped to $[-1 + 10^{-7}, 1 - 10^{-7}]$ before $\arccos$ is applied.

Communication cost. The bi-projection is learned locally and incurs no communication. During each federated round, client i uploads one d-dimensional prototype and one class count per local class and receives one d-dimensional client-frame target per local class, so both uplink and downlink cost $\mathcal{O}(C_i d)$ with $C_i = |\mathcal{C}_i|$. Prototype-based baselines exchange prototypes in the backbone feature space, whose dimension is $d_i = 512$ for all clients under the HtFLlib protocol, and their server returns prototypes for all C classes. GeoProtoFL exchanges only local-class prototypes in the interface with $d = 128$ ($d = 256$ on Tiny-ImageNet), so its communication depends on d and not on d_i or the backbone size. Table 5 compares the per-round cost with the baselines.

Computation cost. Learning the bi-projection costs $\mathcal{O}\big(E_{\mathrm{proj}}\, n_i\, d_{\mathrm{hid}}(d_i + d)\big)$ on client i, excluding backbone forward passes, where d_{hid} is the hidden width of the two-layer MLPs. This cost is paid once before federated training. In each round, a participating client additionally runs one forward pass over $\mathcal{D}_i$ to compute its prototypes. On the server, building $\mathbf{H}_i^t$ costs $\mathcal{O}(C_i d^2)$, its $d \times d$ SVD costs $\mathcal{O}(d^3)$, and applying $\mathbf{R}_i^t$ to the prototypes and $(\mathbf{R}_i^t)^\top$ to the returned targets costs $\mathcal{O}(C_i d^2)$ for every participating client. The Karcher iterations cost $\mathcal{O}\big(K_{\mathrm{bar}}\, d \sum_{i \in \mathcal{S}^t} C_i\big)$. The total server cost per round is therefore

$$\mathcal{O}\Big(\sum_{i \in \mathcal{S}^t} \big(C_i d^2 + d^3\big) + K_{\mathrm{bar}}\, d \sum_{i \in \mathcal{S}^t} C_i\Big), \tag{13}$$

which depends on d and not on d_i or the backbone size.

Table 5: Per-round communication complexity for a participating client i. $C_i = |\mathcal{C}_i|$ is the number of classes at client i, C is the total number of classes, d_i is the backbone feature dimension ($d_i = 512$ in our experiments), and d is the dimension of the GeoProtoFL prototype interface ($d = 128$ on CIFAR-10 and CIFAR-100, $d = 256$ on Tiny-ImageNet).

Method	Shared information	Uplink	Downlink
Local	–	–	–
LG-FedAvg (Liang et al., 2020)	Partial model	$\mathcal{O}(\lvert\theta^h\rvert)$	$\mathcal{O}(\lvert\theta^h\rvert)$
FedGH (Yi et al., 2023)	Partial model	$\mathcal{O}(S_i r)$	$\mathcal{O}(\lvert\theta^h\rvert)$
FedRE (Yao et al., 2026)	Representation	$\mathcal{O}(d_{\mathrm{RE}})$	$\mathcal{O}(\lvert\theta^h\rvert)$
FedProto (Tan et al., 2022)	Prototype	$\mathcal{O}(C_i d_i)$	$\mathcal{O}(C d_i)$
FedTGP (Zhang et al., 2024)	Prototype	$\mathcal{O}(C_i d_i)$	$\mathcal{O}(C d_i)$
FedORGP (Guo et al., 2025)	Prototype	$\mathcal{O}(C_i d_i)$	$\mathcal{O}(C d_i)$
FedSAF (Wu et al., 2026)	Prototype	$\mathcal{O}(C_i d_i)$	$\mathcal{O}(C d_i)$
GeoProtoFL	Prototype	$\mathcal{O}(C_i d)$	$\mathcal{O}(C_i d)$

Table 6: Parameter overhead of the bi-projection Φ_i relative to the feature extractor of each backbone for the CIFAR configuration ($d_i = 512$, $d = 128$, $d_{\mathrm{hid}} = 256$, $|\Phi_i| \approx 329$K for every client).

Backbone	$\lvert\theta_i^g\rvert$ (M)	Overhead (%)
4-layer CNN	0.86	38.24
MobileNetV2 / GoogLeNet	3.50–6.60	4.98–9.40
ResNet-18 to ResNet-152	11.7–60.2	0.55–2.81
ViT-B/16 / ViT-B/32	86.6–88.2	0.37–0.38

Parameter overhead. With $d_i = 512$, $d = 128$, and $d_{\mathrm{hid}} = 256$, the bi-projection has approximately 329K parameters, independently of the backbone, and approximately 394K parameters with $d = 256$ on Tiny-ImageNet. Table 6 reports this overhead relative to each backbone. It is below 3% for ResNet-18 and larger backbones and below 1% for ResNet-101 and larger, and it is largest for the 4-layer CNN at 38.24%. Only the encoder, about half of these parameters, is used during federated training, and the bi-projection adds no per-round communication.

A.2 Theoretical Analysis

We analyze the server operation of one communication round of GeoProtoFL. Propositions 1 and 2 concern gauge registration and show why a class-wise rule cannot be gauge consistent while registered aggregation is. Theorem 2 and Corollary 2 concern intrinsic aggregation. Theorems 1 and 3 and Corollaries 1 and 3 bound the error of the registered and global prototypes, and Propositions 3 and 4 concern the targets returned to clients. None of the statements concerns convergence of federated training.

A.2.1 Setup

Fix a round t and a participating client i, and order the classes of $\mathcal{C}_i$ in a fixed way. We stack the uploaded prototypes and the corresponding reference prototypes as

$$\mathbf{P}_i^t = \left[\mathbf{p}_i^{c,t}\right]_{c\in\mathcal{C}_i}, \qquad \mathbf{G}_i^{t-1} = \left[\mathbf{p}^{c,t-1}\right]_{c\in\mathcal{C}_i} \in \mathbb{R}^{d\times|\mathcal{C}_i|}, \tag{14}$$

so that $\mathbf{P}_i^t$ is the matrix form of the prototype set $\mathbf{p}_i^t$ in Figure 2. Each uploaded prototype is a mean of unit vectors (Section 3.2.2), so $\kappa_i^{c,t} = \|\mathbf{p}_i^{c,t}\|_2 \le 1$, and we assume $\kappa_i^{c,t} > 0$. We write $\mathbf{\Lambda}_i^t = \mathrm{diag}(\kappa_i^{c,t})_{c\in\mathcal{C}_i}$, apply Norm column by column so that $\mathrm{Norm}(\mathbf{P}_i^t) = \mathbf{P}_i^t(\mathbf{\Lambda}_i^t)^{-1}$, and use $\|\mathbf{X}\|_\Lambda = \|\mathbf{X}(\mathbf{\Lambda}_i^t)^{1/2}\|_F$ and $d_{\mathbb{S}}(\mathbf{u},\mathbf{v}) = \arccos(\mathbf{u}^\top\mathbf{v})$. The columns of $\mathbf{G}_i^{t-1}$ are ETF anchors or Fréchet barycenters and therefore have unit norm. Registration in Eq. 9 is

$$\mathbf{R}_i^t \in \operatorname*{arg\,min}_{\mathbf{R}\in O(d)} \left\|\mathbf{R}\mathbf{P}_i^t - \mathbf{G}_i^{t-1}\right\|_F^2, \qquad \mathbf{H}_i^t = \mathbf{G}_i^{t-1}(\mathbf{P}_i^t)^\top. \tag{15}$$

Lemma 1 (Registering prototypes weights their directions by concentration)**.** *The minimizers of Eq. 15 are exactly the minimizers of* $\sum_{c\in\mathcal{C}_i}\kappa_i^{c,t}\|\mathbf{R}\,\mathrm{Norm}(\mathbf{p}_i^{c,t})-\mathbf{p}^{c,t-1}\|_2^2=\|\mathbf{R}\,\mathrm{Norm}(\mathbf{P}_i^t)-\mathbf{G}_i^{t-1}\|_\Lambda^2$ *over* $O(d)$*, and the maximizers of* $\mathrm{tr}(\mathbf{R}^\top\mathbf{H}_i^t)$.

Proof. Write $\mathbf{g}^c=\mathbf{p}^{c,t-1}$. For orthogonal $\mathbf{R}$, $\|\mathbf{R}\mathbf{P}_i^t-\mathbf{G}_i^{t-1}\|_F^2=\|\mathbf{P}_i^t\|_F^2+\|\mathbf{G}_i^{t-1}\|_F^2-2\sum_c(\mathbf{g}^c)^\top\mathbf{R}\mathbf{p}_i^{c,t}$. Since $\|\,\mathrm{Norm}(\mathbf{p}_i^{c,t})\|_2=\|\mathbf{g}^c\|_2=1$ and $\kappa_i^{c,t}\,\mathrm{Norm}(\mathbf{p}_i^{c,t})=\mathbf{p}_i^{c,t}$, the weighted objective equals $2\,\mathrm{tr}\,\mathbf{\Lambda}_i^t-2\sum_c(\mathbf{g}^c)^\top\mathbf{R}\mathbf{p}_i^{c,t}$. Both are constants minus $2\,\mathrm{tr}(\mathbf{R}^\top\mathbf{H}_i^t)$. □

Registering prototypes before normalization is thus registration of their directions with reliability weights $\kappa_i^{c,t}$, so a class with diffuse projected features influences the estimated gauge less. The error bounds require a model of the residual mismatch.

Assumption 1 (Approximate orthogonal gauge)**.** *There exist unit vectors* $\mathbf{p}_\star^{c,t}$*, shared by all clients that hold class* c*, and* $\mathbf{Q}_i^t\in O(d)$ *such that, with* $\mathbf{P}_{i,\star}^t=[\mathbf{p}_\star^{c,t}]_{c\in\mathcal{C}_i}$*,*

$$\mathrm{Norm}(\mathbf{P}_i^t)=\mathbf{Q}_i^t\mathbf{P}_{i,\star}^t+\mathbf{E}_i^t,\qquad \mathbf{G}_i^{t-1}=\mathbf{P}_{i,\star}^t+\mathbf{F}_i^t, \tag{16}$$

where $\mathbf{E}_i^t$ *collects prototype estimation error and any departure from orthogonal equivalence, and* $\mathbf{F}_i^t$ *collects the error and staleness of the server reference.*

Assumption 1 is a modeling choice, not a consequence of projecting to a common dimension. Its fit can be monitored through the Procrustes residual $\|\mathbf{R}_i^t\,\mathrm{Norm}(\mathbf{P}_i^t)-\mathbf{G}_i^{t-1}\|_\Lambda$.

A.2.2 Gauge Consistency

A rule is *class-wise* if it computes the global prototype of class c only from the class-c uploads and their weights and returns it unchanged to every client. Coordinate-wise averaging and the Fréchet barycenter of unregistered prototypes are class-wise.

Proposition 1 (Class-wise rules are not gauge consistent)**.** *Let* $d\geq 2$*. (i) For any* $\mathbf{p},\mathbf{v}_1,\dots,\mathbf{v}_n\in\mathbb{S}^{d-1}$*, there exist* $\mathbf{Q}_1,\dots,\mathbf{Q}_n\in O(d)$ *with* $\mathbf{Q}_j\mathbf{p}=\mathbf{v}_j$*, so the uploads of one class carry no information about the client gauges. (ii) Consequently, for every class-wise rule there exist error-free uploads (*$\mathbf{E}_i^t=\mathbf{0}$*) from two clients for which the returned prototype* $\mathbf{a}$ *is undefined or satisfies* $\max_j d_{\mathbb{S}}(\mathbf{a},\mathbf{Q}_j\mathbf{p}_\star^{c,t})\geq\pi/2$*, so it lies at geodesic distance at least* $\pi/2$ *from the latent prototype expressed in the frame of some client. For the same uploads and an exact reference (*$\mathbf{F}_i^t=\mathbf{0}$*), registration followed by intrinsic aggregation returns* $\mathbf{Q}_j\mathbf{p}_\star^{c,t}$ *to every client* j*.*

Proof. (i) If $\mathbf{v}_j=\mathbf{p}$, take $\mathbf{Q}_j=\mathbf{I}_d$. Otherwise the Householder reflection $\mathbf{Q}_j=\mathbf{I}_d-2\mathbf{u}\mathbf{u}^\top/\|\mathbf{u}\|_2^2$ with $\mathbf{u}=\mathbf{p}-\mathbf{v}_j$ maps $\mathbf{p}$ to $\mathbf{v}_j$. (ii) Fix a unit vector $\mathbf{u}$ and, by (i), take gauges with $\mathbf{Q}_1\mathbf{p}_\star^{c,t}=\mathbf{u}$ and $\mathbf{Q}_2\mathbf{p}_\star^{c,t}=-\mathbf{u}$, defining the uploads of the other classes of each client through the same gauges. These uploads are error-free, and any returned $\mathbf{a}$ satisfies $d_{\mathbb{S}}(\mathbf{a},\mathbf{u})+d_{\mathbb{S}}(\mathbf{a},-\mathbf{u})=\pi$, so one of the two distances is at least $\pi/2$. With $\mathbf{E}_j^t=\mathbf{F}_j^t=\mathbf{0}$, Theorem 1 gives $\delta_j^t=0$, hence $\mathbf{R}_j^t\mathbf{Q}_j\mathbf{P}_{j,\star}^t=\mathbf{P}_{j,\star}^t$ because $\kappa_j^{c,t}>0$. Every registered prototype of class c then equals $\mathbf{p}_\star^{c,t}$, the Karcher iterates start at $\mathbf{p}^{c,t-1}=\mathbf{p}_\star^{c,t}$ and stay there, and $(\mathbf{R}_j^t)^\top\mathbf{p}_\star^{c,t}=\mathbf{Q}_j\mathbf{p}_\star^{c,t}$. □

Proposition 1 is a worst case and does not claim that typical data reach it. It shows that the uploads of one class carry no information about client gauges. Registration obtains this information from the joint configuration of each client's classes, compared with the server reference.

Proposition 2 (Registered aggregation is gauge consistent)**.** *Keep the server reference fixed. Let* $r=\mathrm{rank}(\mathbf{H}_i^t)$ *with singular value decomposition* $\mathbf{H}_i^t=\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$*, split as* $\mathbf{U}=[\mathbf{U}_r,\mathbf{U}_\perp]$ *and* $\mathbf{V}=[\mathbf{V}_r,\mathbf{V}_\perp]$ *after the first* r *columns.*

(i) The solutions of Eq. 15 are exactly $\mathbf{R}=\mathbf{U}_r\mathbf{V}_r^\top+\mathbf{U}_\perp\mathbf{W}\mathbf{V}_\perp^\top$ *with* $\mathbf{W}\in O(d-r)$*. If the columns of* $\mathbf{G}_i^{t-1}$ *are linearly independent, all solutions give the same registered configuration* $\mathbf{R}\mathbf{P}_i^t$*.*

(ii) For $\mathbf{Q} \in O(d)$, $\mathbf{R}$ solves Eq. 15 for the uploads $\mathbf{Q}\mathbf{P}_i^t$ if and only if $\mathbf{R}\mathbf{Q}$ solves it for $\mathbf{P}_i^t$. Both yield the same registered configurations, and corresponding solutions return client-frame targets related by $(\mathbf{R}\mathbf{Q}^\top)^\top \mathbf{q} = \mathbf{Q}(\mathbf{R}^\top \mathbf{q})$ for every $\mathbf{q} \in \mathbb{R}^d$.

Hence, when the registered configuration is unique, the aggregation output is unchanged if client i expresses its prototypes in any other orthogonal frame.

Proof. (i) By Lemma 1, $\mathbf{R}$ is a solution if and only if it maximizes $\mathrm{tr}(\mathbf{R}^\top \mathbf{H}_i^t) = \sum_{j \le r} \sigma_j \mathbf{u}_j^\top \mathbf{R} \mathbf{v}_j$. Each term satisfies $\mathbf{u}_j^\top \mathbf{R} \mathbf{v}_j \le 1$, with equality if and only if $\mathbf{R}\mathbf{v}_j = \mathbf{u}_j$, so the solutions map $\mathbf{V}_r$ to $\mathbf{U}_r$ and act on the orthogonal complements through an arbitrary $\mathbf{W} \in O(d-r)$ (Schönemann, 1966). The columns of $\mathbf{V}_\perp$ are orthogonal to $\mathrm{col}((\mathbf{H}_i^t)^\top) = \mathrm{col}(\mathbf{P}_i^t(\mathbf{G}_i^{t-1})^\top)$, which equals $\mathrm{col}(\mathbf{P}_i^t)$ when $\mathbf{G}_i^{t-1}$ has linearly independent columns. Then $\mathbf{V}_\perp^\top \mathbf{P}_i^t = \mathbf{0}$ and $\mathbf{R}\mathbf{P}_i^t = \mathbf{U}_r \mathbf{V}_r^\top \mathbf{P}_i^t$. (ii) The map $\mathbf{R} \mapsto \mathbf{R}\mathbf{Q}$ is a bijection of $O(d)$ with $\|\mathbf{R}(\mathbf{Q}\mathbf{P}_i^t) - \mathbf{G}_i^{t-1}\|_F = \|(\mathbf{R}\mathbf{Q})\mathbf{P}_i^t - \mathbf{G}_i^{t-1}\|_F$, so the minimizers correspond and $\mathbf{R}(\mathbf{Q}\mathbf{P}_i^t) = (\mathbf{R}\mathbf{Q})\mathbf{P}_i^t$. A solution $\mathbf{R}$ for $\mathbf{P}_i^t$ corresponds to $\mathbf{R}\mathbf{Q}^\top$ for $\mathbf{Q}\mathbf{P}_i^t$, whose transpose is $\mathbf{Q}\mathbf{R}^\top$. □

The uniqueness condition of Proposition 2(i) can be checked at the server from $\mathbf{G}_i^{t-1}$. At $t = 1$ the reference consists of ETF anchors, which are linearly independent for every client that observes fewer than C classes and linearly dependent for a client that observes all C classes, since the anchors sum to zero.

Remark 1 (Anchoring does not replace registration). Anchoring aligns the normalized class means of a client with the shared anchors only approximately, because it is optimized on finite, label-skewed data during the bi-projection stage (Section 3.2.1). Afterwards the encoder is fixed while the backbone continues to train, so the class configuration of a client can move away from the anchors. Registration estimates the resulting mismatch in every round. When the condition of Proposition 2(i) holds, directions orthogonal to the class directions of a client do not affect its registered prototypes and enter its returned targets only through the term bounded in Proposition 3.

A.2.3 Accuracy of Registered and Global Prototypes

Theorem 1 (Registration stability without a rank condition)**.** *Under Assumption 1, every solution of Eq. 15 satisfies*

$$\left\|\mathbf{R}_i^t \,\mathrm{Norm}(\mathbf{P}_i^t) - \mathbf{P}_{i,\star}^t\right\|_\Lambda \le \delta_i^t := \|\mathbf{E}_i^t\|_\Lambda + 2\|\mathbf{F}_i^t\|_\Lambda \le \|\mathbf{E}_i^t\|_F + 2\|\mathbf{F}_i^t\|_F. \tag{17}$$

Proof. By Lemma 1, $\mathbf{R}_i^t$ minimizes $\|\mathbf{R}\,\mathrm{Norm}(\mathbf{P}_i^t) - \mathbf{G}_i^{t-1}\|_\Lambda$, and $(\mathbf{Q}_i^t)^\top$ is feasible, so this residual is at most $\|(\mathbf{Q}_i^t)^\top \mathbf{E}_i^t - \mathbf{F}_i^t\|_\Lambda \le \|\mathbf{E}_i^t\|_\Lambda + \|\mathbf{F}_i^t\|_\Lambda$, using $\|\mathbf{O}\mathbf{X}\|_\Lambda = \|\mathbf{X}\|_\Lambda$ for orthogonal $\mathbf{O}$. The triangle inequality with $\mathbf{G}_i^{t-1} - \mathbf{P}_{i,\star}^t = \mathbf{F}_i^t$ gives the first bound, and $\kappa_i^{c,t} \le 1$ gives the second. □

Theorem 1 concerns the registered prototypes and requires no rank condition on $\mathbf{P}_i^t$. It does not recover $\mathbf{Q}_i^t$, which is determined only on the span of the client's class directions, a proper subspace whenever $|\mathcal{C}_i| < d$ as in all our settings. Aggregation does not require more.

Corollary 1 (Per-class error of a registered prototype)**.** *Under Assumption 1, every $c \in \mathcal{C}_i$ satisfies*

$$d_{\mathbb{S}}\big(\hat{\mathbf{p}}_i^{c,t}, \mathbf{p}_\star^{c,t}\big) \le \beta_i^{c,t} := 2\arcsin\Big(\min\Big\{1, \frac{\delta_i^t}{2\sqrt{\kappa_i^{c,t}}}\Big\}\Big). \tag{18}$$

Proof. Since $\hat{\mathbf{p}}_i^{c,t} = \mathrm{Norm}(\mathbf{R}_i^t \mathbf{p}_i^{c,t}) = \mathbf{R}_i^t \,\mathrm{Norm}(\mathbf{p}_i^{c,t})$, column c of $(\mathbf{R}_i^t \,\mathrm{Norm}(\mathbf{P}_i^t) - \mathbf{P}_{i,\star}^t)(\mathbf{\Lambda}_i^t)^{1/2}$ equals $\sqrt{\kappa_i^{c,t}}(\hat{\mathbf{p}}_i^{c,t} - \mathbf{p}_\star^{c,t})$, so $\|\hat{\mathbf{p}}_i^{c,t} - \mathbf{p}_\star^{c,t}\|_2 \le \delta_i^t / \sqrt{\kappa_i^{c,t}}$ by Theorem 1. For unit vectors, $\|\mathbf{u} - \mathbf{v}\|_2 = 2\sin(d_{\mathbb{S}}(\mathbf{u}, \mathbf{v})/2) \le 2$. □

For class c, write $F^{c,t}(\mathbf{q}) = \sum_{i \in \mathcal{S}^{c,t}} w_i^{c,t} d_{\mathbb{S}}(\mathbf{q}, \hat{\mathbf{p}}_i^{c,t})^2$ for the objective of Eq. 11, whose weights sum to one, and $\mathrm{grad}\, F^{c,t}(\mathbf{q}) = -2\sum_i w_i^{c,t} \,\mathrm{Log}_{\mathbf{q}}(\hat{\mathbf{p}}_i^{c,t})$ for its Riemannian gradient. A minimizer exists by compactness but need not be unique when registered prototypes are dispersed.

Theorem 2 (Well-posedness of the Fréchet barycenter). *Let $\mathbf{a} \in \mathbb{S}^{d-1}$ and $0 < \rho < \pi/4$ be such that all registered prototypes of class c lie in the open geodesic ball $B(\mathbf{a}, \rho)$, and let $m_\rho = 2\rho \cot(2\rho) \in (0, 1)$. Then $F^{c,t}$ has a unique minimizer $\mathbf{p}_\circ^{c,t}$ on $\mathbb{S}^{d-1}$, which lies in $B(\mathbf{a}, \rho)$, and every $\mathbf{q} \in B(\mathbf{a}, \rho)$ satisfies*

$$d_{\mathbb{S}}\big(\mathbf{q}, \mathbf{p}_\circ^{c,t}\big) \le \frac{\|\operatorname{grad} F^{c,t}(\mathbf{q})\|_2}{2m_\rho}, \qquad F^{c,t}(\mathbf{q}) - F^{c,t}(\mathbf{p}_\circ^{c,t}) \le \frac{\|\operatorname{grad} F^{c,t}(\mathbf{q})\|_2^2}{4m_\rho}. \tag{19}$$

Proof. For data in a ball of radius $\rho < \pi/4$ on the unit sphere, the minimizer exists, is unique, and lies in the closure of the geodesic convex hull of the data (Karcher, 1977; Afsari, 2011). This hull lies in the closed ball of radius $\max_i d_{\mathbb{S}}(\mathbf{a}, \hat{\mathbf{p}}_i^{c,t}) < \rho$, so $\mathbf{p}_\circ^{c,t} \in B(\mathbf{a}, \rho)$. For $\mathbf{q} \in B(\mathbf{a}, \rho)$, every distance $r_i = d_{\mathbb{S}}(\mathbf{q}, \hat{\mathbf{p}}_i^{c,t})$ is below $2\rho < \pi/2$. The Riemannian Hessian of $\frac{1}{2} d_{\mathbb{S}}(\cdot, \hat{\mathbf{p}}_i^{c,t})^2$ has eigenvalue 1 in the radial direction and $r_i \cot r_i \ge m_\rho$ in the transverse directions (Karcher, 1977), so $\operatorname{Hess} F^{c,t} \succeq 2m_\rho \mathbf{I}$ on the ball, which is geodesically convex because $\rho < \pi/2$. Let γ be the unit-speed geodesic from $\mathbf{q}$ to $\mathbf{p}_\circ^{c,t}$, of length ℓ, which stays in the ball, and $h(s) = \langle \operatorname{grad} F^{c,t}(\gamma(s)), \dot\gamma(s) \rangle$. Then $h' \ge 2m_\rho$ and $h(\ell) = 0$, so $2m_\rho \ell \le -h(0) \le \|\operatorname{grad} F^{c,t}(\mathbf{q})\|_2$. Strong convexity along γ also gives $F^{c,t}(\mathbf{p}_\circ^{c,t}) \ge F^{c,t}(\mathbf{q}) - \|\operatorname{grad} F^{c,t}(\mathbf{q})\|_2 \, \ell + m_\rho \ell^2$, and minimizing the right-hand side over ℓ yields the second bound. □

Corollary 2 (Accuracy and a computable certificate). *(a) If all registered prototypes of class c lie in $B(\mathbf{p}_\star^{c,t}, \rho)$ with $\rho < \pi/4$, then $d_{\mathbb{S}}(\mathbf{p}_\circ^{c,t}, \mathbf{p}_\star^{c,t}) \le m_\rho^{-1} \| \sum_i w_i^{c,t} \operatorname{Log}_{\mathbf{p}_\star^{c,t}}(\hat{\mathbf{p}}_i^{c,t}) \|_2 \le m_\rho^{-1} \sum_i w_i^{c,t} d_{\mathbb{S}}(\hat{\mathbf{p}}_i^{c,t}, \mathbf{p}_\star^{c,t})$. (b) For the returned prototype $\mathbf{p}^{c,t}$, let $\hat\rho^{c,t} = \max_i d_{\mathbb{S}}(\mathbf{p}^{c,t}, \hat{\mathbf{p}}_i^{c,t})$ and $g^{c,t} = \|\operatorname{grad} F^{c,t}(\mathbf{p}^{c,t})\|_2$. If $\hat\rho^{c,t} < \pi/4$, then for every $\rho \in (\hat\rho^{c,t}, \pi/4)$, $d_{\mathbb{S}}(\mathbf{p}^{c,t}, \mathbf{p}_\circ^{c,t}) \le g^{c,t}/(2m_\rho)$ and $F^{c,t}(\mathbf{p}^{c,t}) - F^{c,t}(\mathbf{p}_\circ^{c,t}) \le (g^{c,t})^2/(4m_\rho)$. Both $\hat\rho^{c,t}$ and $g^{c,t}$ are computed at the server.*

Proof. Apply Theorem 2 with $\mathbf{a} = \mathbf{q} = \mathbf{p}_\star^{c,t}$ for (a), using $\|\operatorname{Log}_{\mathbf{a}}(\mathbf{x})\|_2 = d_{\mathbb{S}}(\mathbf{a}, \mathbf{x})$, and with $\mathbf{a} = \mathbf{q} = \mathbf{p}^{c,t}$ for (b). □

Corollary 2(b) certifies the K_{bar} Karcher iterations of Eq. 12 directly, without relying on convergence conditions for the iteration (Afsari, 2011). Without a concentration condition, let $\varepsilon_{\text{opt}}^{c,t} \ge 0$ satisfy $F^{c,t}(\mathbf{p}^{c,t}) \le \inf_{\mathbf{q}} F^{c,t}(\mathbf{q}) + \varepsilon_{\text{opt}}^{c,t}$.

Theorem 3 (Accuracy of the returned global prototype). *For any $\mathbf{p}_\star^{c,t} \in \mathbb{S}^{d-1}$,*

$$d_{\mathbb{S}}\big(\mathbf{p}^{c,t}, \mathbf{p}_\star^{c,t}\big) \le 2\Big(\sum_{i \in \mathcal{S}^{c,t}} w_i^{c,t} d_{\mathbb{S}}\big(\hat{\mathbf{p}}_i^{c,t}, \mathbf{p}_\star^{c,t}\big)^2 \Big)^{1/2} + \sqrt{\varepsilon_{\text{opt}}^{c,t}}. \tag{20}$$

Proof. By the triangle inequality, $d_{\mathbb{S}}(\mathbf{p}^{c,t}, \mathbf{p}_\star^{c,t}) \le d_{\mathbb{S}}(\mathbf{p}^{c,t}, \hat{\mathbf{p}}_i^{c,t}) + d_{\mathbb{S}}(\hat{\mathbf{p}}_i^{c,t}, \mathbf{p}_\star^{c,t})$ for each i. Averaging with the weights $w_i^{c,t}$ and applying the Cauchy–Schwarz inequality gives $d_{\mathbb{S}}(\mathbf{p}^{c,t}, \mathbf{p}_\star^{c,t}) \le \sqrt{F^{c,t}(\mathbf{p}^{c,t})} + \sqrt{F^{c,t}(\mathbf{p}_\star^{c,t})}$. The claim follows from $F^{c,t}(\mathbf{p}^{c,t}) \le F^{c,t}(\mathbf{p}_\star^{c,t}) + \varepsilon_{\text{opt}}^{c,t}$ and $\sqrt{a+b} \le \sqrt{a} + \sqrt{b}$. □

Corollary 3 (Error of one server round). *Under Assumption 1, $d_{\mathbb{S}}(\mathbf{p}^{c,t}, \mathbf{p}_\star^{c,t}) \le 2(\sum_i w_i^{c,t} (\beta_i^{c,t})^2)^{1/2} + \sqrt{\varepsilon_{\text{opt}}^{c,t}}$. If, in addition, $\max_i \beta_i^{c,t} < \rho < \pi/4$, then $\mathbf{p}_\circ^{c,t}$ is unique and $d_{\mathbb{S}}(\mathbf{p}_\circ^{c,t}, \mathbf{p}_\star^{c,t}) \le m_\rho^{-1} \sum_i w_i^{c,t} \beta_i^{c,t}$.*

Proof. Insert Corollary 1 into Theorem 3. Under the additional condition, Corollary 1 places every registered prototype in $B(\mathbf{p}_\star^{c,t}, \rho)$, and Corollary 2(a) applies. □

A.2.4 CLIENT-FRAME TARGETS

The returned target $\bar{\mathbf{p}}_i^{c,t} = (\mathbf{R}_i^t)^\top \mathbf{p}^{c,t}$ applies the full matrix $\mathbf{R}_i^t$, which is not unique when $r < d$ by Proposition 2(i).

Proposition 3 (Stability of client-frame targets)**.** *Let $\mathbf{\Pi}_i^t$ be the orthogonal projector onto* $\mathrm{col}(\mathbf{H}_i^t)$*. Any two solutions $\mathbf{R}, \mathbf{R}'$ of Eq. 15 satisfy $\|\mathbf{R}^\top \mathbf{q} - \mathbf{R}'^\top \mathbf{q}\|_2 \le 2\|(\mathbf{I}_d - \mathbf{\Pi}_i^t)\mathbf{q}\|_2$ for every $\mathbf{q} \in \mathbb{R}^d$. If $\mathbf{P}_i^t$ has linearly independent columns, then for every $c \in \mathcal{C}_i$ and unit $\mathbf{q}$,*

$$\|\mathbf{R}^\top \mathbf{q} - \mathbf{R}'^\top \mathbf{q}\|_2 \le 2\, d_{\mathbb{S}}\big(\mathbf{q}, \mathbf{p}^{c,t-1}\big), \qquad \big|d_{\mathbb{S}}(\mathbf{z}, \mathbf{R}^\top \mathbf{q}) - d_{\mathbb{S}}(\mathbf{z}, \mathbf{R}'^\top \mathbf{q})\big| \le \pi\, d_{\mathbb{S}}\big(\mathbf{q}, \mathbf{p}^{c,t-1}\big) \quad (21)$$

for every $\mathbf{z} \in \mathbb{S}^{d-1}$. For $\mathbf{q} = \mathbf{p}^{c,t}$, the non-identifiable part of the target is therefore controlled by the drift of the global prototype since the reference used for registration.

Proof. By Proposition 2(i), $\mathbf{R}^\top \mathbf{q} - \mathbf{R}'^\top \mathbf{q} = \mathbf{V}_\perp (\mathbf{W} - \mathbf{W}')^\top \mathbf{U}_\perp^\top \mathbf{q}$ with $\|\mathbf{W} - \mathbf{W}'\|_2 \le 2$ and $\|\mathbf{U}_\perp^\top \mathbf{q}\|_2 = \|(\mathbf{I}_d - \mathbf{\Pi}_i^t)\mathbf{q}\|_2$. If $\mathbf{P}_i^t$ has linearly independent columns, $\mathrm{col}(\mathbf{H}_i^t) = \mathrm{col}(\mathbf{G}_i^{t-1}) \ni \mathbf{p}^{c,t-1}$, so $\|(\mathbf{I}_d - \mathbf{\Pi}_i^t)\mathbf{q}\|_2 \le \|\mathbf{q} - \mathbf{p}^{c,t-1}\|_2 \le d_{\mathbb{S}}(\mathbf{q}, \mathbf{p}^{c,t-1})$. The second bound follows from the triangle inequality and $d_{\mathbb{S}}(\mathbf{u}, \mathbf{v}) \le \frac{\pi}{2}\|\mathbf{u} - \mathbf{v}\|_2$ for unit vectors. □

Proposition 4 (Guidance and aggregation use the same discrepancy)**.** *For every $\mathbf{z} \in \mathbb{S}^{d-1}$ and $\mathbf{R} \in O(d)$, $d_{\mathbb{S}}(\mathbf{z}, \mathbf{R}^\top \mathbf{p}^{c,t}) = d_{\mathbb{S}}(\mathbf{R}\mathbf{z}, \mathbf{p}^{c,t})$. Hence, in round $t+1$, the guidance term of Eq. 7 for a sample $\mathbf{x}$ of class c is the squared geodesic distance between the registered feature $\mathbf{R}_i \Phi_i^{\mathrm{enc}}(g(\mathbf{x}; \theta_i^g))$ and $\mathbf{p}^{c,t}$, the point that approximately minimizes the weighted sum of the same squared distances to the registered prototypes.*

Proof. $(\mathbf{R}^\top \mathbf{p}^{c,t})^\top \mathbf{z} = (\mathbf{p}^{c,t})^\top \mathbf{R}\mathbf{z}$, and $d_{\mathbb{S}}$ depends only on the inner product. □

Proposition 4 formalizes intrinsic consensus, and Proposition 3 bounds how much another valid registration would change the target.

A.2.5 SCOPE

The analysis covers one server round and does not establish convergence of federated training or bound classification risk. Theorem 1, Corollaries 1 and 3, and the last statement of Proposition 1(ii) are stated under Assumption 1, and the remaining results hold without it. Under partial participation, a stale gauge enters through $\mathbf{F}_i^t$ and through the drift term of Proposition 3. The quantities in the bounds, namely the Procrustes residual, the angular spread $\hat{\rho}^{c,t}$, the Karcher gradient norm $g^{c,t}$, and the round-to-round drift of global prototypes, can be logged at the server.

A.3 BI-PROJECTION DESIGN

Section 3.2.1 introduces the client-specific bi-projection $\Phi_i = \{\Phi_i^{\mathrm{enc}}, \Phi_i^{\mathrm{dec}}\}$, which each client learns before federated training with the reconstruction and anchoring objectives in Eqs. 5 and 6. This section motivates the encoder-decoder design and describes its implementation.

The anchoring objective constrains only the normalized class means of projected features and therefore does not, by itself, prevent distinct features within the same class from mapping to nearly identical points. The reconstruction objective penalizes such information loss by requiring the decoder to recover the input feature from its projection. It thus discourages feature collapse and encourages the encoder to retain information about its input. When $d < d_i$, the encoder cannot be invertible, so reconstruction acts as a retention regularizer rather than providing an invertibility guarantee.

Both maps are two-layer MLPs with a ReLU activation, and the encoder additionally normalizes its output onto the unit sphere,

$$\Phi_i^{\mathrm{enc}}(\mathbf{z}_i) = \mathrm{Norm}\Big(\mathbf{W}_i^{2,\mathrm{enc}}\, \mathrm{ReLU}\Big(\mathbf{W}_i^{1,\mathrm{enc}} \mathbf{z}_i + \mathbf{b}_i^{1,\mathrm{enc}}\Big) + \mathbf{b}_i^{2,\mathrm{enc}}\Big), \quad (22)$$

where $\mathbf{W}_i^{1,\mathrm{enc}} \in \mathbb{R}^{d_{\mathrm{hid}} \times d_i}$ and $\mathbf{W}_i^{2,\mathrm{enc}} \in \mathbb{R}^{d \times d_{\mathrm{hid}}}$. The decoder has the same two-layer architecture without the final normalization, and the hidden width d_{hid} is shared across clients. Because both maps act on backbone features rather than raw inputs, their parameter count and computational cost depend only on $(d_i, d_{\mathrm{hid}}, d)$ and not on the backbone architecture. After E_{proj} epochs, the encoder is

Table 7: Method-specific hyperparameters. Unless stated otherwise, all methods use SGD with local learning rate 0.01, batch size 32, and five local epochs per round. E_{proj} denotes GeoProtoFL bi-projection epochs, and E_s denotes server-side training epochs.

Method	Hyperparameters
Local	–
LG-FedAvg (Liang et al., 2020)	–
FedGH (Yi et al., 2023)	Server learning rate $= 0.01$
FedProto (Tan et al., 2022)	$\lambda_{\text{proto}} = 1.0$
FedTGP (Zhang et al., 2024)	Margin threshold $= 100.0$; $E_s = 100$
FedRE (Yao et al., 2026)	–
FedORGP (Guo et al., 2025)	$\lambda_c = 100.0$; $\lambda_s = 1.0$; $\gamma = 10.0$; $E_s = 1$
FedSAF (Wu et al., 2026)	Structural method: GCSA; $\lambda_{\text{proto}} = 1.0$; $\gamma_{\text{inst}} = 1.0$
GeoProtoFL	$E_{\text{proj}} = 5$ (learning rate 0.01); $\lambda_{\text{rec}} = 1.0$; $\lambda_{\text{anc}} = 0.2$; $\lambda_{\text{proto}} = 1.0$; Karcher iterations $K_{\text{bar}} = 5$

fixed, while the backbone continues to be trained during FL. The fixed encoder therefore defines a stable interface, and the prototypes computed through it in each round reflect the current backbone. Appendix A.1 reports the resulting parameter, computation, and communication costs.

A.4 Detailed Experimental Setup

Hardware and software. All experiments are run on a single NVIDIA GeForce RTX 4090 GPU with 126GB of system RAM. We implement GeoProtoFL and all baselines in PyTorch 2.0.1 under Python 3.11, built on top of HtFLlib (Zhang et al., 2025a), and use its data partitioning, training loop, and evaluation protocol unmodified to ensure a fair comparison across methods.

Dataset splits. We follow HtFLlib's default protocol throughout, splitting each client's local data into 75% train and 25% test, with non-IID partitioning applied independently to each split so that no client's train and test distributions diverge from one another beyond the partitioning scheme itself (Section 4.1).

Hyperparameters. All methods use 5 local epochs per round (Section 4.1); Table 7 reports the remaining method-specific hyperparameters. For methods included in HtFLlib, we use its default configuration unmodified. For FedORGP (Guo et al., 2025) and FedSAF (Wu et al., 2026), which are not part of the public HtFLlib release, we use the hyperparameters reported in their respective publications.

Model architecture pools. Table 8 lists the backbones used in each heterogeneity configuration. Following HtFLlib's deterministic assignment rule, the $(i \bmod X)$-th architecture in a group of size X is assigned to client i (Zhang et al., 2025a). With $N = 20$, this yields five clients per architecture under HtFE^{img}4, three clients for each of the first four architectures and two for each of the remaining four under HtFE^{img}8, two clients per architecture under HtM^{img}10, and five clients per classifier head under HtFE^{img}8-HtC^{img}4.

Following the standard HtFLlib protocol, an average-pooling interface maps each backbone output to the common post-pooling dimension $d_i = 512$, for every client in the reported experiments. The backbones nevertheless remain heterogeneous in architecture and parameterization.

A.5 Hyperparameter Sensitivity

To assess whether GeoProtoFL depends on precise tuning, we vary each of its five hyperparameters around the default configuration on CIFAR-100 (HtFE^{img}8, $\alpha = 0.1$, $N = 20$) while keeping the others at their defaults. These are the reconstruction and anchoring weights λ_{rec} and λ_{anc} of the

Table 8: Model architecture pools across heterogeneity configurations. All backbones output a shared feature dimension of 512 after average pooling.

Configuration	Architectures
$\text{HtFE}^{\text{img}}4$	4-layer CNN, GoogLeNet, MobileNetV2, ResNet-18
$\text{HtFE}^{\text{img}}8$	4-layer CNN, GoogLeNet, MobileNetV2, ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152
$\text{HtM}^{\text{img}}10$	$\text{HtFE}^{\text{img}}8$ + ViT-B/16, ViT-B/32
$\text{HtFE}^{\text{img}}8\text{-HtC}^{\text{img}}4$	$\text{HtFE}^{\text{img}}8$ backbones with four classifier heads: Head([512]), Head([512,512]), Head([512,256]), Head([512,128])

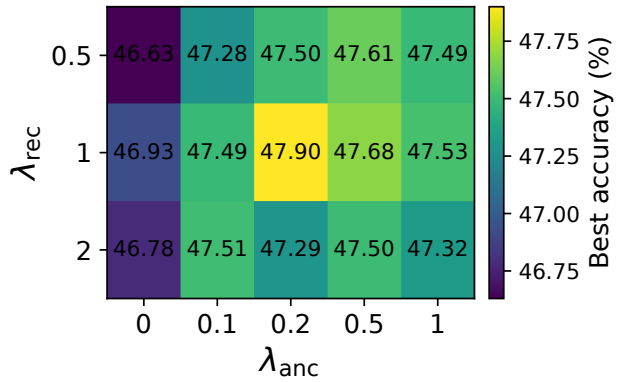

Figure 3: Sensitivity to the bi-projection weights λ_{rec} and λ_{anc} in Eq. 4. Test accuracy (%) on CIFAR-100 with $\text{HtFE}^{\text{img}}8$ (Dirichlet, $\alpha = 0.1$, $N = 20$)

bi-projection, the guidance weight λ_{proto}, the interface dimension d, and the number of bi-projection epochs E_{proj}.

Bi-projection weights. Figure 3 shows the joint effect of the reconstruction and anchoring weights λ_{rec} and λ_{anc} in Eq. 4. Setting $\lambda_{\text{anc}} = 0$ gives the lowest accuracy in every row, ranging from 46.63 to 46.93%, whereas positive anchoring weights yield 47.28%–47.90%. This indicates that the ETF anchors are important for establishing the shared spherical interface, while the exact balance between reconstruction and anchoring has a smaller effect once anchoring is active. We therefore use $\lambda_{\text{rec}} = 1$ and $\lambda_{\text{anc}} = 0.2$, which achieves the highest mean accuracy.

Guidance weight. To evaluate how strongly clients should follow the global prototypes, Table 9 varies the weight λ_{proto} in Eq. 7, which balances client-frame prototype guidance against the task loss. Accuracy is highest at $\lambda_{\text{proto}} = 1$, similar at $\lambda_{\text{proto}} = 0.5$, and lower at $\lambda_{\text{proto}} = 2$ and 3, suggesting that overly strong prototype guidance can begin to compete with the task loss.

Interface dimension. Table 9 also reports the effect of the interface dimension d, which sets the capacity of the shared sphere and the per-round communication cost. The setting $d = 128$ gives the highest accuracy with 47.90%, whereas $d = 256$ and $d = 512$ give 47.50% and 47.55%, so larger interfaces bring no gain. Because the ETF anchors in Eq. 3 require $d \geq C$, $d = 128$ is the smallest power of two admissible on CIFAR-100, and Tiny-ImageNet with $C = 200$ uses $d = 256$.

Bi-projection epochs. To evaluate how long the bi-projection stage must run before the encoder is fixed, the last group of Table 9 varies E_{proj}. A single epoch gives the lowest accuracy in this group, whereas 5 and 10 epochs perform similarly. Together with the anchoring result above, this suggests that the encoder mainly needs to establish a class-indexed layout on the sphere rather than undergo prolonged fitting, consistent with its role as a fixed interface during federated training.

Overall, removing ETF anchoring causes the largest drop, yielding the lowest accuracy in every row of Figure 3, whereas all other evaluated configurations remain between 47.18% and 47.90%.

Table 9: Sensitivity to the guidance weight λ_{proto}, the interface dimension d, and the bi-projection epochs E_{proj}. Test accuracy (%) on CIFAR-100 with HtFE^{img}8 (Dirichlet, $\alpha = 0.1$, $N = 20$).

Hyperparameter	Value	Accuracy (%)
λ_{proto}	0.5	47.65±0.11
	1.0	**47.90±0.15**
	2.0	47.38±0.20
	3.0	47.35±0.12
d	**128**	**47.90±0.15**
	256	47.50±0.21
	512	47.55±0.25
E_{proj}	1	47.18±0.24
	5	**47.90±0.15**
	10	47.55±0.10

The default configuration, $\lambda_{\text{rec}} = 1$, $\lambda_{\text{anc}} = 0.2$, $\lambda_{\text{proto}} = 1$, $d = 128$, and $E_{\text{proj}} = 5$, gives the highest mean accuracy in every sweep, and we use it in all experiments, except that Tiny-ImageNet uses $d = 256$ as required by the anchor construction. Substantially different datasets, architectures, degrees of heterogeneity, or training setups may require re-tuning.