# SHEAF-BASED FEDERATED REPRESENTATION LEARNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Heterogeneous federated systems require agents to learn and exchange informative representations despite differences in data distributions, model architectures, and latent dimensionalities. We propose Sheaf-based Federated Representation Learning (SFRL), a framework that jointly learns and aligns heterogeneous latent representations without sharing parameters, labels, or a single latent space. SFRL relates agent-specific latent spaces through learnable orthogonal transformations and isometric embeddings, promoting consistency via a sheaf-Laplacian gluing regularizer evaluated on a shared set of pilot samples. We develop Sheaf-FRL, a decentralized alternating algorithm combining local gradient updates with closed-form Procrustes updates of the alignment maps, and establish convergence to first-order stationary points in deterministic and stochastic settings. Applied to collaborative classification in semantic communication under model and data heterogeneity, Sheaf-FRL improves private and communication accuracy over federated baselines and is more robust to latent-space compression.

## 1 INTRODUCTION

Modern machine learning systems increasingly operate in decentralized and heterogeneous environments, where data, models, sensing modalities, and computational resources are distributed across multiple agents, devices, or institutions. In this context, we adopt the term *agent* to denote any physical or artificial entity performing a task in an environment. Privacy constraints, regulatory requirements, and communication limitations often prevent centralized data aggregation, motivating *federated learning* (FL), where agents optimize local models on private data while exchanging limited information rather than raw samples (Kairouz et al., 2021).

A central challenge in such systems is that agents may learn representations that are useful locally but incompatible across the network. Existing FL methods commonly rely on parameter averaging, global regularization, or shared encoders, and therefore implicitly assume that features learned by different agents are directly comparable (McMahan et al., 2017; Scardapane & Di Lorenzo, 2017; Li et al., 2020b; Karimireddy et al., 2020; Collins et al., 2021; Li et al., 2021; Dong & Voiculescu, 2021). This assumption can fail when agents differ in their data distributions, sensing modalities, architectures, latent dimensions, or local objectives. Independently trained encoders may produce representations that encode equivalent task-relevant information while differing by rotations, reflections, or more general isometric embeddings. Such ambiguity reflects the inherent *gauge freedom* of representation learning objectives, whereby equivalent solutions may differ by transformations that preserve geometric structure (Wang & Isola, 2020). Consequently, forcing all agents into a single shared latent space can be unnecessarily restrictive and may degrade local representation quality or hinder knowledge transfer.

These observations motivate *semantic alignment*: rather than requiring agents to adopt identical latent coordinates, one should learn how their local representation spaces are related. This perspective is supported by recent work on relative representations, which shows that independently learned latent spaces can support zero-shot communication when their relational structure is preserved (Moschella et al., 2023). Semantic alignment can support heterogeneous federated representation learning across supervised, semi-supervised, and self-supervised settings, as well as cooperative tasks in which agents exploit representations received from their neighbors. For example, it is relevant to multimodal sensing, collaborative robotics, distributed channel charting, transfer across



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.

heterogeneous models, and task-aware semantic communication, where latent information generated by one agent must remain useful to another despite differences in their internal representations.

A natural mathematical framework for these settings is a *cellular sheaf* on graph, hereafter referred to as a *network sheaf* (Curry, 2014). A network sheaf associates a local vector space with each node and linear restriction maps with graph edges, thereby encoding how information in neighboring spaces should be compared or transported. Recent work in graph signal processing and graph neural networks has shown that sheaf-based models can represent vector-valued data coupled through local consistency constraints (Hansen & Ghrist, 2019; Di Nino et al., 2024; 2026; Bodnar et al., 2022; Barbero et al., 2022). This provides a natural abstraction for federated representation learning: each agent maintains its own latent space, while learnable edge-wise maps encode the geometric relations required to align representations across neighboring agents. Global consistency then emerges from local compatibility relations, rather than from a shared latent space.

**Contributions.** We propose a sheaf-based framework for federated representation learning that enables semantic alignment across heterogeneous agents without requiring a shared latent space. Specifically, *(i)* we formulate federated representation learning over a learnable network sheaf, where agent-specific latent spaces are related through edge-wise geometric transport maps; *(ii)* we introduce a quadratic gluing penalty induced by a learnable sheaf Laplacian, evaluated on a shared set of reference representations, hereinafter *pilots*, to promote alignment while limiting communication; *(iii)* we employ orthogonal and Stiefel restriction maps to capture rotations, reflections, and isometric embeddings across latent spaces of possibly different dimensions; *(iv)* we develop a fully decentralized alternating algorithm, termed Sheaf-FRL, that combines local gradient updates with closed-form Procrustes updates of the transport maps, and establish convergence to first-order stationary points in deterministic and stochastic settings; and *(v)* we demonstrate that Sheaf-FRL improves over baseline approaches in learning and representation transfer for collaborative supervised classification in semantic communication, under model and data heterogeneity. The code implementing the proposed method and reproducing all the experiments is publicly available. [1]

## 2 Related Work

**Federated learning under heterogeneity.** *Federated learning* (FL) trains a shared model through repeated local stochastic gradient updates followed by either server-side aggregation (McMahan et al., 2017) or neighbor-based combination over a communication graph (Scardapane & Di Lorenzo, 2017), with stabilizing mechanisms such as proximal regularization (Li et al., 2020b) and control variates (Karimireddy et al., 2020) addressing the client drift induced by statistical and system heterogeneity. *Federated multi-task learning* (FMTL) instead trains client-specific models while exploiting relatedness across tasks or clients (Smith et al., 2017), most prominently through graph-based regularization, where Laplacian penalties enforce smoothness of *model parameters* across neighboring clients (Dinh et al., 2022; Issaid et al., 2025). *Multimodal FL* (MFL) (Che et al., 2023) targets clients holding distinct modalities; recent graph-based methods such as CoMFed (Badi et al., 2026) handle heterogeneity by aligning compressed latent prototypes, yet still require a predefined common embedding space. Finally, *federated representation learning* (FRL) seeks transferable representations via auxiliary consistency losses on latent embeddings (Li et al., 2021), split architectures with a common encoder and client-specific heads (Collins et al., 2021; McLaughlin & Su, 2024), or prototype alignment (Tan et al., 2022; Tran et al., 2024); analogous ideas extend to unsupervised and self-supervised regimes through contrastive objectives (Zhuang et al., 2021; Han et al., 2022; Zhang et al., 2023; Miao & Koyuncu, 2024; Ghalkha et al., 2026); a further line instead distils or correlates model outputs on shared public data (Li & Wang, 2019; Huang et al., 2024). These paradigms differ mainly in *what* is assumed to be shared among agents (a single global model, a coupled set of parameters, a common multimodal embedding, a shared latent space, or a public dataset) rather than in the underlying goal of exploiting relatedness. We instead learn edge-wise geometric transport maps relating agent-specific latent spaces and promote their semantic compatibility. We recover each of these paradigms as a special case; App. B expands on this positioning.

**Representation alignment, synchronization, and manifold methods.** Alignment up to orthogonal transformations arises naturally when representations are identifiable only up to rotations or reflec-

[1] https://anonymous.4open.science/r/sheaf-based-federated-representation-learning-203C/README.md

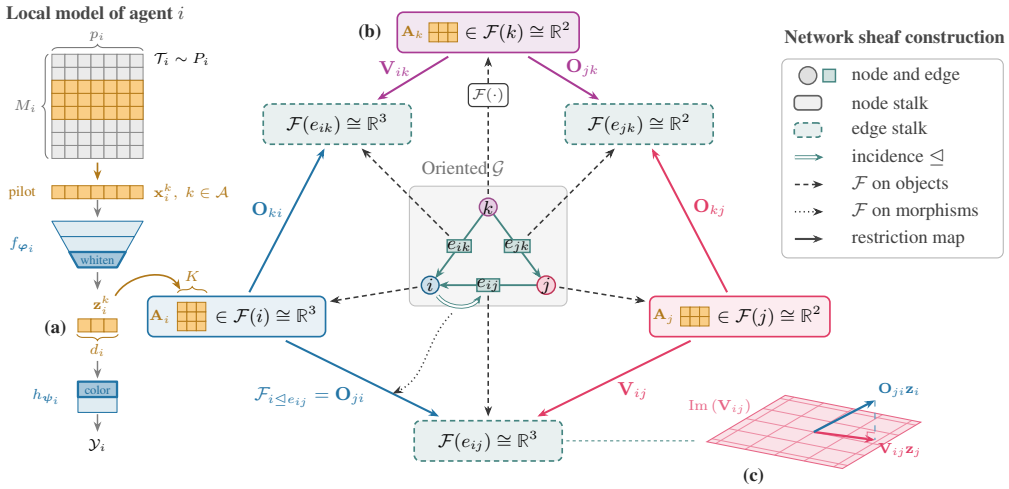


Figure 1: *Sheaf-based federated representation learning* with three heterogeneous agents ($d_i = 3$ and $d_j = d_k = 2$). *Left:* the local model of agent $i$, from the local dataset $\mathcal{T}_i \sim P_i$ to the output space $\mathcal{Y}_i$, through the learnable encoder $f_{\boldsymbol{\varphi}_i}$ and head $h_{\boldsymbol{\psi}_i}$, which handle heterogeneity in the local geometry. Only the $K$ pilots take part in the alignment, their latent representations forming the pilot matrix $\mathbf{A}_i \in \mathbb{R}^{d_i \times K}$. In our experiments the pilot pool is a held-out dataset, disjoint from every local pool, so that no sample is seen by more than one agent during training and the heterogeneity of the latent spaces is not diluted by shared data. *Right:* the network sheaf $\mathcal{F}$ assigns a stalk to every node and edge of the oriented graph $\mathcal{G}$ (with $d_{ij} = \max(d_i, d_j)$), and a restriction map to every incidence relation, e.g. $\mathcal{F}_{i \trianglelefteq e_{ij}} = \mathbf{O}_{ji}$ (orthogonal map) and $\mathcal{F}_{j \trianglelefteq e_{ij}} = \mathbf{V}_{ij}$ (isometric embedding). Each edge is oriented from the lower- to the higher-dimensional stalk; on $e_{jk}$, the two stalks have equal dimension, so both maps are orthogonal and the orientation is arbitrary. Three features are highlighted. **(a)** Agents exchange latent representations, not model parameters or raw data. **(b)** No single global latent space is assumed: global consistency may only emerge from the alignment of the agents' node stalks over the sheaf. **(c)** The gluing penalty measures the residual $\|\mathbf{O}_{ji}\mathbf{z}_i - \mathbf{V}_{ij}\mathbf{z}_j\|_2^2$ on the edge stalk, i.e. the component of the transported representation orthogonal to $\mathrm{Im}\,(\mathbf{V}_{ij})$.

tions, with the orthogonal Procrustes problem providing a classical closed-form solution (Schönemann, 1966). We leverage this solution in a federated and decentralized setting, both under homogeneous latent spaces (cf. App. C) and under stringent privacy constraints (cf. App. D). Related ideas arise in group synchronization and connection-Laplacian methods (Singer, 2011; Singer & Wu, 2012; Bandeira et al., 2013; Thunberg et al., 2017), which enforce consistency across pairwise transformations; unlike these approaches, we jointly learn latent representations at the nodes and alignment maps on the edges. To accommodate heterogeneous latent dimensions, we further consider semi-orthogonal maps on the Stiefel manifold, enabling isometric embeddings across representation spaces. This connects our formulation to optimization on matrix manifolds (Absil et al., 2008; Edelman et al., 1998) and federated optimization under manifold constraints (Li & Ma, 2022); however, these works focus on optimizing model parameters subject to geometric constraints, rather than learning alignment maps as a component of representation learning.

## 3 Framework and Problem Formulation

### 3.1 Local Agents and Network Model

Let $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ denote a finite undirected graph with $|\mathcal{N}| = N$ nodes, illustrated in Fig. 1; the notation used throughout is summarized in App. A. Each node is an agent, while an edge $(i, j) \in \mathcal{E}$ encodes a structural relationship (similarity in tasks, data distributions, modalities, or architectures) realized through a physical communication channel.

Each agent $i \in \mathcal{N}$ observes a local dataset $\mathcal{T}_i$ comprising $M_i$ samples $\mathbf{x}_i^n \in \mathbb{R}^{p_i}$, $n \in [M_i]$, drawn from an underlying distribution $P_i$, and learns a local representation model comprising a neural encoder

$$f_{\boldsymbol{\varphi}_i} : \mathbb{R}^{p_i} \to \mathcal{F}(i)\,, \tag{1}$$

where $\boldsymbol{\varphi}_i$ is the set of learnable parameters, and $\mathcal{F}(i) \cong \mathbb{R}^{d_i}$ is the latent representation space of agent $i$. When required by the local task, the encoder is followed by a personalized head parametrized by $\boldsymbol{\psi}_i$, i.e.,

$$h_{\boldsymbol{\psi}_i} : \mathcal{F}(i) \to \mathcal{Y}_i\,, \tag{2}$$

where $\mathcal{Y}_i$ is an agent-dependent output space. We let $\boldsymbol{\theta}_i$ denote the collection of local trainable parameters, including $\boldsymbol{\varphi}_i$ and, when present, $\boldsymbol{\psi}_i$. Each agent minimizes an arbitrary local learning objective $\mathcal{L}_i(\boldsymbol{\theta}_i)$ based on $\mathcal{T}_i$, e.g., the empirical risk $\mathcal{L}_i(\boldsymbol{\theta}_i) = M_i^{-1} \sum_{n=1}^{M_i} \ell_i(h_{\boldsymbol{\psi}_i} \circ f_{\boldsymbol{\varphi}_i}(\mathbf{x}_i^n))$ in the supervised setting.[2]

### 3.2 The learnable latent network sheaf

To formalize the relationships, and thus the communication, among the latent representation spaces $\mathcal{F}(i)$, we adopt the framework of network sheaves (Bredon, 1997; Curry, 2014). A *network sheaf* $\mathcal{F}$ on $\mathcal{G}$ consists of the following assignments. To each node $i \in \mathcal{N}$ it associates a $d_i$-dimensional real vector space $\mathcal{F}(i)$, the learnable latent space of agent $i$, referred to as *node stalk*, whose valuation is the embedding $\mathbf{z}_i = f_{\boldsymbol{\varphi}_i}(\mathbf{x}_i)$. We consider the general case in which the node stalks have different dimensionalities $d_i$. To each edge $e_{ij} = (i,j) \in \mathcal{E}$ it assigns a $d_{ij}$-dimensional real vector space $\mathcal{F}(e_{ij})$, called *edge stalk*, encoding pairwise compatibility between agents $i$ and $j$. Stacking node and edge valuations yields the cochain spaces $C^0(\mathcal{G};\mathcal{F}) = \bigoplus_{i\in\mathcal{N}} \mathcal{F}(i)$ and $C^1(\mathcal{G};\mathcal{F}) = \bigoplus_{e_{ij}\in\mathcal{E}} \mathcal{F}(e_{ij})$. Finally, to each incidence relation $i \trianglelefteq e_{ij}$ it specifies a linear *restriction map* $\mathcal{F}_{i \trianglelefteq e_{ij}} : \mathcal{F}(i) \to \mathcal{F}(e_{ij})$, which relates the local latent representation at node $i$ to the corresponding edge space and thereby enables communication among latent spaces.

We focus on learnable restriction maps that preserve distances and angles, so as to handle the well-known geometric invariance of latent representations (Moschella et al., 2023) without amplifying latent perturbations. These are orthogonal transformations and Stiefel matrices, where the orthogonal and Stiefel manifolds are

$$\mathrm{O}(d) \coloneqq \{\mathbf{O} \in \mathbb{R}^{d\times d} \mid \mathbf{O}^\top = \mathbf{O}^{-1}\}\,, \quad \text{and} \quad \mathrm{St}(d,k) \coloneqq \{\mathbf{V} \in \mathbb{R}^{d\times k} \mid \mathbf{V}^\top\mathbf{V} = \mathbf{I}_k,\ k < d\}\,. \tag{3}$$

Accordingly, for each edge $e_{ij} = (i,j) \in \mathcal{E}$ with $d_i > d_j$ we set $\mathcal{F}(e_{ij}) = \mathcal{F}(i) \cong \mathbb{R}^{d_i}$, which implies $\mathcal{F}_{i \trianglelefteq e_{ij}} = \mathbf{O}_{ji} \in \mathrm{O}(d_i)$ and $\mathcal{F}_{j \trianglelefteq e_{ij}} = \mathbf{V}_{ij} \in \mathrm{St}(d_i, d_j)$, as depicted in Fig. 1. App. E discusses the benefits of the choice $d_{ij} = \max(d_i, d_j)$ as opposed to compressing.

**Heterogeneous latent space geometry.** Encoders are trained independently, on different data and possibly with different architectures, so the induced latent spaces exhibit heterogeneous geometry (different scales and anisotropies) even when the dimensionalities $d_i$ coincide (Pandolfo et al., 2026; Svendsen et al., 2026). Since isometric restriction maps measure inconsistency reliably only when the representations they compare adhere to a common geometric reference on the edge stalk, each encoder terminates in a whitening layer (Zhang et al., 2021) that is not part of $\boldsymbol{\theta}_i$. It is instead estimated in the forward pass by a decoupled decorrelation criterion and held fixed in the backward pass, whereas $\boldsymbol{\theta}_i$ collects the backpropagated parameters. Each head begins with its left inverse, the coloring layer. Whitening and coloring layers compose to the identity along the local path, so $\mathcal{L}_i$ is unaffected by the choice of reference; the coloring is what allows a representation transported from a neighbor to be read by a head trained in local coordinates (Fig. 1). App. E details this construction.

**Sheaf Laplacian and global sections.** The sheaf induces a coboundary operator $\boldsymbol{\Delta} : C^0(\mathcal{G};\mathcal{F}) \to C^1(\mathcal{G};\mathcal{F})$ defined edgewise as

$$(\boldsymbol{\Delta}\mathbf{z})_{e_{ij}} = \mathbf{O}_{ji}\mathbf{z}_i - \mathbf{V}_{ij}\mathbf{z}_j\,, \qquad e_{ij} = (i,j) \in \mathcal{E}\,. \tag{4}$$

[2] Depending on the local paradigm, $\mathcal{T}_i$ may additionally include labels, side information, paired observations, or locally generated views of the samples. Our framework is agnostic to the choice of $\mathcal{L}_i$, and thus accommodates semi-supervised, unsupervised, and self-supervised learning.

Equipping $C^0$ and $C^1$ with the canonical Euclidean inner products makes the adjoint $\boldsymbol{\Delta}^\top$ well defined, and yields the symmetric positive semidefinite network sheaf Laplacian

$$\mathbf{L}_{\mathcal{F}} = \boldsymbol{\Delta}^\top \boldsymbol{\Delta}\,. \tag{5}$$

A valuation $\mathbf{z}^\star \in \ker(\mathbf{L}_{\mathcal{F}})$ is said *global section*: it assigns latent representations to the node stalks without breaking any local consistency rule, i.e., $\mathbf{O}_{ji}\mathbf{z}_i^\star = \mathbf{V}_{ij}\mathbf{z}_j^\star$ for each $e_{ij} \in \mathcal{E}$. Requiring the learned representations to be a global section of the learned sheaf would be the idealized notion of network-wide alignment. Such a constraint, however, is unrealistic across heterogeneous agents, and couples all node representations simultaneously, thus precluding a decentralized solution; we state and discuss this centralized formulation in App. E. This motivates the relaxed, distributed formulation developed next.

### 3.3 Scalable gluing penalty and decentralized problem

We replace the global-section requirement with a soft penalty based on the sheaf total variation, which promotes geometric consistency across agents without enforcing exact agreement:

$$\mathcal{TV}(\mathbf{z}) \coloneqq \|\boldsymbol{\Delta}\mathbf{z}\|_2^2 = \mathbf{z}^\top \mathbf{L}_{\mathcal{F}}\mathbf{z} = \sum_{e_{ij}\in\mathcal{E}} \|\mathbf{O}_{ji}\mathbf{z}_i - \mathbf{V}_{ij}\mathbf{z}_j\|_2^2\,. \tag{6}$$

Constraining $\mathbf{z}$ to be a global section is equivalent to zeroing Eq. (6). Additional discussion on the geometric consistency promoted by the action of the restriction maps, as well as on the use of the total variation penalty, is provided in App. E.

**Semantic embedding and orientation.** Exploiting the transitive action of the orthogonal group on the Stiefel manifold, we reparameterize $\widetilde{\mathbf{V}}_{ij} \coloneqq \mathbf{O}_{ji}^\top \mathbf{V}_{ij}$, so that each edge term in Eq. (6) becomes

$$\|\mathbf{O}_{ji}\mathbf{z}_i - \mathbf{V}_{ij}\mathbf{z}_j\|_2^2 = \left\|\mathbf{z}_i - \widetilde{\mathbf{V}}_{ij}\mathbf{z}_j\right\|_2^2; \tag{7}$$

hereafter we drop the tilde and write $\mathbf{V}_{ij}$ for the reparameterized map. Eq. (7) highlights that null total variation corresponds to the higher-dimensional latent space being an isometric embedding of the lower-dimensional one. While this condition does not always hold, our restriction maps adhere to the *semantic embedding principle* (D'Acunto et al., 2025; 2026): informally, the semantics of the lower-dimensional node stalk $\mathcal{F}(j)$ is preserved when embedded into the higher-dimensional $\mathcal{F}(i)$. The reparameterization also induces a natural *embedding orientation* on $\mathcal{G}$, from nodes with lower-dimensional latent spaces to those with higher-dimensional ones, i.e., along the direction in which semantics is preserved (Fig. 1); we exploit it in the sequel to reduce the communication cost. For $e_{ij}$ we let node $i$ be the head and $j$ the tail, consistently with $d_i \geq d_j$. Ties $d_i = d_j = d$ are broken arbitrarily, since $\mathbf{V}_{ij} = \mathbf{V}_{ji}^\top \in \mathrm{O}(d)$ makes the edge contribution orientation-invariant (cf. $e_{jk}$ in Fig. 1). On such edges the Stiefel constraints in Prob. 1 are understood as orthogonality constraints.

**Incoming and outgoing embedding maps.** For each $i \in \mathcal{N}$, let $\mathcal{N}(i)^-$ collect the neighbors $j$ for which $i$ is the head, and $\mathcal{N}(i)^+$ those for which $i$ is the tail, so that $d_j \leq d_i$ on $\mathcal{N}(i)^-$, $d_j \geq d_i$ on $\mathcal{N}(i)^+$, with equality only on tied edges, and $\mathcal{N}(i) = \mathcal{N}(i)^- \cup \mathcal{N}(i)^+$. Symmetrizing Eq. (6) over the two sets, the total variation decouples across nodes as $\mathcal{TV}(\mathbf{z}) = \sum_{i\in\mathcal{N}} \mathcal{TV}(\mathbf{z})|_i$, where the contribution of node $i$ is made of two terms with different geometric meanings:

$$\mathcal{TV}(\mathbf{z})|_i = \underbrace{\frac{1}{2}\sum_{j\in\mathcal{N}(i)^-} \|\mathbf{z}_i - \mathbf{V}_{ij}\mathbf{z}_j\|_2^2}_{\text{Incoming Embedding}} + \underbrace{\frac{1}{2}\sum_{j\in\mathcal{N}(i)^+} \|\mathbf{z}_j - \mathbf{V}_{ji}\mathbf{z}_i\|_2^2}_{\text{Outgoing Embedding}}\,. \tag{8}$$

Accordingly, the matrices $\mathbf{V}_{ij}$ in the first term are referred to as the *incoming* embedding maps of node $i$, and the matrices $\mathbf{V}_{ji}$ in the second term as the *outgoing* ones. Eq. (8) is what enables local updates and reduced communication cost in our decentralized algorithm.

**Pilots and problem formulation.** For the sake of exposition, we assume that a common index set $\{1,\dots,M\}$ of pilot inputs has been agreed upon, and we let $\mathcal{A} \subset \{1,\dots,M\}$, with $|\mathcal{A}| = K$, denote fixed pilot indices, so that agent $i$ holds the pilot feature matrix $\mathbf{A}_i \coloneqq \mathbf{A}_i(\boldsymbol{\varphi}_i) = [\mathbf{z}_i^k]_{k\in\mathcal{A}} \in \mathbb{R}^{d_i\times K}$. The pilot pool need not be part of the local training sets: it may consist of public data,

calibration signals, or an overlapping portion of the local datasets (App. F). Identifiability of the transport maps requires $K \geq \max_i d_i$ (App. G), which lower-bounds the pilot budget. Restricting Eq. (8) to $\mathcal{A}$, we associate a positive regularization coefficient $\lambda_{ij} = \lambda_{ji} > 0$ to each undirected edge $(i, j) \in \mathcal{E}$, treated as an edge-wise hyperparameter and kept fixed throughout training. The gluing penalty decouples as $\mathcal{R}_{\mathcal{A}} = \sum_{i \in \mathcal{N}} \mathcal{R}_{\mathcal{A}}|_i$, where

$$\mathcal{R}_{\mathcal{A}}|_i(\boldsymbol{\varphi}_i, \{\mathbf{V}_{ij}\}, \{\mathbf{V}_{ji}\}) = \frac{1}{2K}\left[\sum_{j \in \mathcal{N}(i)^-} \lambda_{ij} \|\mathbf{A}_i - \mathbf{V}_{ij}\mathbf{A}_j\|_{\mathrm{F}}^2 + \sum_{j \in \mathcal{N}(i)^+} \lambda_{ij} \|\mathbf{A}_j - \mathbf{V}_{ji}\mathbf{A}_i\|_{\mathrm{F}}^2\right]. \quad (9)$$

Thus, the same edge coefficient $\lambda_{ij}$ weights the alignment error at both endpoints of an edge, while allowing different edges to have different regularization strengths. Enforcing alignment on $\mathcal{A}$ only is a deliberately weak requirement: agreement on the pilots does not imply that the remaining local representations are perfectly alignable through the learned restriction maps, consistently with the fact that we neither assume nor aim for a single global latent representation.

**Problem 1** (Sheaf-based Federated Representation Learning)**.** *Given the graph $\mathcal{G}$ endowed with the embedding orientation, the local datasets $\{\mathcal{T}_i\}_{i \in \mathcal{N}}$, and the pilot indices $\mathcal{A}$, the goal of sheaf-based federated representation learning (SFRL) is to learn, for each node $i$, a local model $\boldsymbol{\theta}_i = (\boldsymbol{\varphi}_i, \boldsymbol{\psi}_i)$ together with incoming and outgoing embeddings $\{\mathbf{V}_{ij} \in \mathrm{St}(d_i, d_j)\}$ and $\{\mathbf{V}_{ji} \in \mathrm{St}(d_j, d_i)\}$ aligning $\mathcal{F}(i)$ with the latent spaces of its neighbors, by solving in a decentralized manner*

$$\min_{\substack{\{\boldsymbol{\theta}_i = (\boldsymbol{\varphi}_i, \boldsymbol{\psi}_i)\} \\ \{\mathbf{V}_{ij} \in \mathrm{St}(d_i, d_j)\} \\ \{\mathbf{V}_{ji} \in \mathrm{St}(d_j, d_i)\}}} \quad \sum_{i \in \mathcal{N}} \mathcal{L}_i(\boldsymbol{\theta}_i) + \sum_{i \in \mathcal{N}} \mathcal{R}_{\mathcal{A}}|_i(\boldsymbol{\varphi}_i, \{\mathbf{V}_{ij}\}, \{\mathbf{V}_{ji}\}); \quad \text{(SFRL)}$$

*while preserving agents' privacy and reducing communication costs over the network.*

Prob. (SFRL) is a regularized nonconvex optimization problem, which couples smooth local neural parameters optimization with manifold-constrained edge transports. It can be interpreted as a scalable decentralized relaxation of the exact global-section formulation reported in App. E.

## 4 The Sheaf-FRL Algorithm

We solve Prob. (SFRL) in a decentralized manner via alternating minimization, consisting of the following two steps.

### 4.1 Step 1: Isometric Embedding Update

Given the neural parameters $\{\boldsymbol{\theta}_i^t = (\boldsymbol{\varphi}_i^t, \boldsymbol{\psi}_i^t)\}$ at communication round $t$, let $\mathbf{A}_i^t \coloneqq \mathbf{A}_i(\boldsymbol{\varphi}_i^t)$ denote the pilot feature matrix of agent $i$. Since the gluing penalty in Eq. (9) separates over the oriented edges, each node $i$ updates its embedding maps by solving two symmetric alignment problems, involving only its neighbors:

$$\mathbf{V}_{ij}^t = \underset{\mathbf{V}_{ij} \in \mathrm{St}(d_i, d_j)}{\arg\min} \quad \left\|\mathbf{A}_i^t - \mathbf{V}_{ij}\mathbf{A}_j^t\right\|_{\mathrm{F}}^2, \quad \forall\, j \in \mathcal{N}(i)^-, \quad \text{(P1a)}$$

$$\mathbf{V}_{ji}^t = \underset{\mathbf{V}_{ji} \in \mathrm{St}(d_j, d_i)}{\arg\min} \quad \left\|\mathbf{A}_j^t - \mathbf{V}_{ji}\mathbf{A}_i^t\right\|_{\mathrm{F}}^2, \quad \forall\, j \in \mathcal{N}(i)^+. \quad \text{(P1b)}$$

Expanding the Frobenius norm in (P1a), the terms $\mathrm{tr}\{\mathbf{A}_i^t(\mathbf{A}_i^t)^\top\}$ and $\mathrm{tr}\{\mathbf{A}_j^t(\mathbf{A}_j^t)^\top\}$ do not depend on $\mathbf{V}_{ij}$, so the problem reduces to maximizing $\mathrm{tr}\{\mathbf{A}_i^t(\mathbf{A}_j^t)^\top\mathbf{V}_{ij}^\top\}$. Denoting by $\mathbf{U}\boldsymbol{\Sigma}\mathbf{W}^\top$ the thin SVD of the cross-covariance $\mathbf{A}_i^t(\mathbf{A}_j^t)^\top$, the solution to (P1a) is

$$\mathbf{V}_{ij}^t = \mathbf{U}\mathbf{W}^\top, \quad \text{where} \quad \mathbf{U} \in \mathrm{St}(d_i, d_j) \text{ and } \mathbf{W} \in \mathrm{O}(d_j); \quad (10)$$

and the solution to (P1b) follows analogously from the thin SVD of $\mathbf{A}_j^t(\mathbf{A}_i^t)^\top$. Each node computes Eq. (10) locally, once its neighbors have shared their pilot features $\{\mathbf{A}_j^t\}_{j \in \mathcal{N}(i)}$, using the orientation to distinguish the edges on which it acts as the head from those on which it acts as the tail. In the homogeneous case, as well as on tied edges of the heterogeneous setting, $\mathrm{St}(d_i, d_j)$ coincides with $\mathrm{O}(d)$, and (P1a)–(P1b) reduce to canonical orthogonal Procrustes problems whose solution follows from the full SVD by setting $\mathbf{O}_{ij}^t = \mathbf{U}\mathbf{W}^\top$ and $\mathbf{O}_{ji}^t = \mathbf{O}_{ij}^{t\top}$, so that tied edges require no separate treatment. Conditions ensuring identifiability of the transport maps, in both the homogeneous and heterogeneous settings, are discussed in App. G.

### 4.2 Step 2: Gradient Update of Neural Parameters

Each agent $i$ updates its local parameters $\boldsymbol{\theta}_i$ through a gradient step on the objective of Prob. (SFRL), while keeping the neighboring parameters $\{\boldsymbol{\theta}_j^t\}_{j\in\mathcal{N}(i)}$ fixed at the current iteration. The regularization term depends exclusively on the encoder parameters $\boldsymbol{\varphi}_i$. Differentiating Eq. (9), and adopting for $j \in \mathcal{N}(i)^+$ the convention $\mathbf{V}_{ij} \coloneqq \mathbf{V}_{ji}^\top$, which satisfies $\mathbf{V}_{ij}\mathbf{V}_{ji} = \mathbf{I}_{d_i}$ by orthonormality of the columns of $\mathbf{V}_{ji}$, the incoming and outgoing contributions admit the unified expression

$$\nabla_{\boldsymbol{\varphi}_i}\mathcal{R}_{\mathcal{A}}|_i = \nabla_{\boldsymbol{\varphi}_i}\mathcal{R}_{\mathcal{A}} = \frac{1}{K}\sum_{j\in\mathcal{N}(i)}\lambda_{ij}\sum_{k\in\mathcal{A}}\left(\nabla_{\boldsymbol{\varphi}_i}f_{\boldsymbol{\varphi}_i}(\mathbf{x}_i^k)\right)^\top\left(f_{\boldsymbol{\varphi}_i}(\mathbf{x}_i^k) - \mathbf{V}_{ij}f_{\boldsymbol{\varphi}_j}(\mathbf{x}_j^k)\right), \tag{11}$$

where $\nabla_{\boldsymbol{\varphi}_i}f_{\boldsymbol{\varphi}_i}(\mathbf{x}_i^k) \in \mathbb{R}^{d_i\times|\boldsymbol{\varphi}_i|}$ denotes the Jacobian of the feature map with respect to $\boldsymbol{\varphi}_i$, evaluated at $\mathbf{x}_i^k$. The full derivation is reported in App. H. Letting $\mathbf{r}_i^t \in \mathbb{R}^{|\boldsymbol{\theta}_i|}$ denote Eq. (11) evaluated at round $t$ and zero-padded on the head coordinates $\boldsymbol{\psi}_i$, the neural parameters are updated locally via

$$\boldsymbol{\theta}_i^{t+1} = \boldsymbol{\theta}_i^t - \eta_\theta\left(\nabla_{\boldsymbol{\theta}_i}\mathcal{L}_i(\boldsymbol{\theta}_i^t) + \mathbf{r}_i^t\right), \tag{12}$$

where $\eta_\theta > 0$ is the stepsize. These computations are fully decentralized: agent $i$ evaluates Eq. (12) using only locally available information together with the pilot representations $\{\mathbf{A}_j^t\}_{j\in\mathcal{N}(i)}$ already received in Step 1, so that no additional communication is required. The resulting algorithm, termed *Sheaf-FRL*, is summarized in App. C for both the heterogeneous and homogeneous settings.

In App. H we establish that Sheaf-FRL converges to block-wise first-order stationary points of the objective $J$ of Prob. (SFRL) under standard-privacy setting. In the deterministic case the objective decreases monotonically and $\min_{0\le t\le T}\|\nabla_{\boldsymbol{\theta}}J(\boldsymbol{\theta}^t,\mathbf{V}^t)\|^2 = \mathcal{O}(1/T)$, whereas with unbiased stochastic gradients of bounded conditional variance a constant stepsize yields $\mathcal{O}(1/\sqrt{T})$ in expectation, and Robbins–Monro stepsizes yield almost-sure convergence. Beyond smoothness and boundedness from below of $J$, the analysis requires exact minimization over $\mathbf{V}$ and compactness of the feasible set, both of which hold by construction in Sheaf-FRL, since (P1a)–(P1b) are solved in closed form.

**Discussion.** *Communication)* At each iteration node $i$ broadcasts $\mathbf{A}_i^t \in \mathbb{R}^{d_i\times K}$ to its neighbors, so that the *outgoing* volume per agent scales as $\mathcal{O}(|\mathcal{N}(i)|d_iK)$. The *incoming* volume received by agent $i$ instead scales as $\mathcal{O}(K\sum_{j\in\mathcal{N}(i)}d_j)$, which can exceed the outgoing payload whenever neighbors hold higher-dimensional latent spaces ($d_j \gg d_i$); only pilot representations are exchanged in either direction, while model parameters and labels remain local (cf. Tab. 1 for the full breakdown). *Privacy)* Sharing latent representations may expose agents to reconstruction and inference attacks (Fredrikson et al., 2015; Shokri et al., 2017; Melis et al., 2019); this is mitigated by exchanging only transported representations, at the price of additional local computation and communication (App. D adapts Sheaf-FRL for strict-privacy settings). *Local computation)* The number of alignment updates could in principle be halved, since the isometric embedding is only required at the lower-dimensional endpoint; we instead exchange raw pilot features and let both endpoints compute their maps locally, shifting the burden from communication to computation, and the trade-off improves further in the homogeneous case, where the number of SVDs can be halved (Apps. C and D). *Accelerated methods)* Momentum-based and adaptive schemes, such as Nesterov acceleration or Adam, can be incorporated into Eq. (12) without affecting the decentralized structure of the algorithm.

## 5 Numerical Results

### 5.1 Collaborative supervised classification with semantic communication

We evaluate Sheaf-FRL on a collaborative and decentralized classification task in the setting of semantic communication, where agents exchange compressed latent representations of data (Gündüz et al., 2022; Barbarossa et al., 2023; Strinati et al., 2024; Pandolfo et al., 2025; Grimaldi et al., 2025). Each agent minimizes a local cross-entropy loss $\mathcal{L}_i(\boldsymbol{\theta}_i) \coloneqq \mathrm{CE}(\boldsymbol{\theta}_i)$, while representation alignment across agents is promoted via the sheaf regularization term. Accordingly, we monitor two metrics, both averaged across agents: the *private accuracy*, i.e., the accuracy of agent $i$ on its own test set through its local latent representations, and the *communication accuracy*, i.e., the accuracy of the head $h_{\boldsymbol{\psi}_i}$ on the latent representations produced by the neighbors $j \in \mathcal{N}(i)$ and transported through the corresponding restriction maps. Formal definitions are reported in App. I.

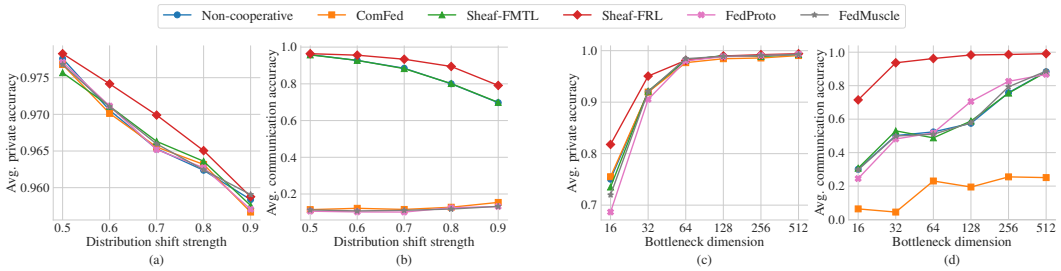


Figure 2: Private and post-communication accuracy on MNIST. **(a)–(b)** Network of $N = 15$ heterogeneous agents. (a) Average private accuracy and (b) average post-communication accuracy versus the distribution-shift strength $s$ in Eq. (13). Private accuracy is a plain average across agents. Post-communication accuracy weights each agent by its share of the total node degree to account for the communication graph's unbalanced degree distribution. **(c)–(d)** Robustness to semantic compression: two heterogeneous CNNs sharing a latent bottleneck of dimension $d$, trained at distribution-shift strength $s = 0.7$. (c) Average private accuracy and (d) average post-communication accuracy versus $d$ (log scale), averaged over the two agents. In (a)–(b), FedProto and FedMuscle are extended to heterogeneous latent dimensions by a learned projection onto a shared space (App. I); in (c)–(d) they apply natively and, like the non-cooperative baseline and Sheaf-FMTL, are aligned post hoc.

**Setup.** We consider $N = 15$ agents, each with a CNN encoder and an MLP head; heterogeneity is enforced by varying encoder and decoder widths and the dropout rate. We use MNIST as the global dataset, partitioned disjointly across agents so that each agent $i$ has a target class set $\mathcal{C}_i \subset \mathcal{Y}$ and local label distribution

$$P_i(y) := s\,\mathrm{Unif}\{\mathcal{C}_i\} + (1-s)\,\mathrm{Unif}\{\mathcal{Y}\}\,, \qquad \forall i \in [N]\,; \tag{13}$$

where $s \in [0, 1)$ is the distribution-shift strength. The shared pilot set $\mathcal{A}$ amounts to $10\%$ of MNIST and is held out from every agent's local pool, so that no sample is seen by more than one agent during training (cf. App. F). Agents communicate over a connected, undirected graph with $40\%$ density, whose edges are selected so as to connect agents with larger target-class overlap. This construction, the data splits, and the agent architectures are detailed in App. I.

**Baselines.** Model and submodel averaging (Scardapane & Di Lorenzo, 2017; Diao et al., 2021; Li et al., 2020a), as well as shared-representation methods (Collins et al., 2021; Liang et al., 2020), assume homogeneous architectures or a shared latent space, and are therefore not applicable here. We thus compare against four methods designed for heterogeneous settings, *ComFed* (Badi et al., 2026), *Sheaf-FMTL* (Issaid et al., 2025), *FedProto* (Tan et al., 2022) and *FedMuscle* (Setayesh et al., 2026), the latter two extended in this benchmark to heterogeneous latent dimensions by a learned projection onto a shared space (App. I). We further add a *non-cooperative* baseline in which models are trained independently, providing a natural lower bound. All baselines use the same communication topology as Sheaf-FRL. Non-cooperative and Sheaf-FMTL require a post-hoc alignment step to enable semantic communication, obtained in closed form from Eq. (10) on the same pilot set (App. I).

**Results.** Fig. 2(a)–(b) show the two metrics against the distribution-shift parameter $s$, which controls the level of data heterogeneity according to Eq. (13). Overall, Sheaf-FRL consistently outperforms all baselines on both metrics. Looking at the communication accuracy, the performance gap with respect to the most competitive baselines widens as $s$ increases, suggesting that our framework is more effective at learning latent spaces that are *(i)* well aligned across agents, thus enhancing transferability, and *(ii)* more relevant to the downstream task, thus being semantically meaningful. Point *(ii)* is further supported by the private accuracy, where Sheaf-FRL still compares favorably, although the margin is less pronounced. Among the baselines, ComFed, FedProto and FedMuscle perform poorly in communication accuracy (below the non-cooperative baseline) while remaining comparable in private accuracy, despite all three learning their alignment maps during training. This suggests that the joint learning of general, unstructured alignment maps and compressed representations of a single shared latent space is not well suited for transferability. Sheaf-FMTL, in turn, shows no consistent improvement in communication accuracy over the non-cooperative baseline. Since Sheaf-FMTL connects parameter spaces rather than latent representation spaces, it inherits their much

higher dimensionality. The resulting memory overhead requires applying a high compression factor to the edge stalks, which negatively affects the diffusion of information across the network (App. I).

#### 5.1.1 Robustness to semantic compression

An important feature in semantic communication is robustness to latent-space compression (Gündüz et al., 2022; Barbarossa et al., 2023). This can be understood as the ability of methods to extract semantically meaningful information to be communicated. We restrict our focus to two agents, say $i$ and $j$, connected by an edge $(i, j)$, set the level of distribution shift to $s = 0.7$, and monitor the two metrics across different bottleneck dimensions. Their architectures differ, but they share the same latent space dimensionality $d = d_i = d_j$. Thus FedProto and FedMuscle apply natively here, with no projection added, and are aligned post hoc like the non-cooperative baseline and Sheaf-FMTL; ComFed instead retains its learned projection, whose dimension is set to $d$.

**Results.** Fig. 2(c)–(d) show the results. Sheaf-FRL achieves a better trade-off between accuracy and bottleneck dimensionality than the baselines, with the gap increasing in the high-compression regime. As expected, all methods suffer as the compression increases; however, Sheaf-FRL degrades much more gracefully. It preserves high accuracy even at a bottleneck dimension of 16, where we approach the minimum number of dimensions required by the network to discriminate the 10 MNIST classes. The post-hoc aligned baselines cluster together but do not close the gap to Sheaf-FRL, which widens as the bottleneck narrows: an agent's latent space is not readily reusable by another, even when an orthogonal post-training alignment is applied. ComFed, in turn, stays below every other method at all bottleneck dimensions, even though here its shared space introduces no compression, so compression alone does not account for the behaviour of the shared-space baselines (App. I).

## 6 Limitations

The pilot budget is bounded below by the identifiability requirement $K \geq \max_i d_i$ (App. G), while the per-agent pools shrink with the number of agents: in our fifteen-agent benchmark the two are comparable, and the regime of sparse pilots, together with the trade-off it induces between alignment quality and communication cost, is left unexplored. Our formulation is agnostic to the local objective, but we validate it only on supervised classification over a single dataset, so its behaviour under unsupervised, self-supervised, and multi-modal objectives remains a conjecture. The communication graph and the edge coefficients $\lambda_{ij}$ are fixed throughout training, whereas the sheaf formulation would naturally accommodate adapting them to the inconsistency actually observed on each edge. We presume a known correspondence between pilot inputs across agents, which is realistic for calibration signals or public data but excludes settings in which the correspondence must itself be inferred. Finally, we constrain the restriction maps to be isometries, which yields the closed-form Procrustes step and the compact feasible set underlying our convergence analysis. How much of the cross-agent relation an isometry captures is a quantitative question we plan to address by measuring the residual alignment error on held-out data, alongside an ablation replacing the Stiefel constraint with an unconstrained linear map.

## 7 Conclusions and Future Work

We introduced *Sheaf-based Federated Representation Learning*, which aligns heterogeneous agents through learnable edge-wise transport maps rather than a shared global latent space, and *Sheaf-FRL*, a fully decentralized alternating algorithm that solves it by coupling local gradient updates with closed-form Procrustes updates of the restriction maps. For the standard-privacy case, we establish convergence to first-order stationary points in both the deterministic and the stochastic setting. On a cooperative classification task in semantic communication, Sheaf-FRL outperformed baselines designed for heterogeneous settings in both local and post-communication accuracy, and degraded more gracefully under latent-space compression, while exchanging only pilot representations, never parameters or data. That a weak, pilot-restricted notion of alignment suffices suggests that global consistency need not be imposed on heterogeneous agents, but can be left to emerge from local compatibility relations. The limitations above set the agenda for future work. The formulation also points to a natural extension: the sheaf total variation decomposes edge by edge, so the disagreement it measures could itself drive the adaptation of the edge coefficients, or of the graph.

### AI USE STATEMENT

In this work, we used generative AI tools to aid in and polish writing, to aid in code development, and to improve proofs readability. Additionally, we used generative AI tools to aid in the development of diagrams and tables, and to improve the layout and readability of figures.

We have not used generative AI tools in research ideation, in the design and development of the overall framework, in the formulation of mathematical claims, in the design of the proof strategies, in the design of the experimental settings, or in the interpretation of results. The rest of the required disclosure tasks are not applicable to this work.

We have reviewed all AI-assisted work. The final paper content has been double-checked and refined by more than two authors. The generated code was verified and tested throughout the experimental development and execution. We take responsibility for the final content of this work.

### ETHICS STATEMENT

Sheaf-FRL exchanges latent representations rather than raw data or model parameters, which limits direct exposure of local data but does not eliminate it: latent features remain susceptible to model inversion, membership inference, and attribute inference (Fredrikson et al., 2015; Shokri et al., 2017; Melis et al., 2019). App. D addresses this by a variant in which agents exchange only transported representations, and Prop. D.1 shows that under unknown restriction maps the underlying pilots are identifiable only up to an orthogonal transformation. A consideration specific to our setting is that the pilot pool is shared by construction: every agent observes the same inputs, so the pilot set should be drawn from public data, calibration signals, or standardized inputs, and not from records that any single agent is obliged to keep private. Our experiments use MNIST and involve no human subjects or personally identifiable information.

### REPRODUCIBILITY STATEMENT

The code implementing Sheaf-FRL and reproducing every experiment is available at the anonymized repository linked in Sec. 1. App. C reports the pseudocode for all four protocol variants, and App. I specifies the agent architectures, the construction of the communication graph, the data partitioning and splits, the evaluation metrics, and the hyperparameter selection procedure. The assumptions and full proofs of the convergence results are in App. H.

## REFERENCES


P.-A. Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

Mohamed Badi, Chaouki Ben Issaid, and Mehdi Bennis. Communication-efficient and robust multimodal federated learning via latent-space consensus. *IEEE Wireless Communications Letters*, pp. 1–1, 2026. doi: 10.1109/LWC.2026.3676885.

Afonso S. Bandeira, Amit Singer, and Daniel A. Spielman. A Cheeger inequality for the graph connection Laplacian. *SIAM Journal on Matrix Analysis and Applications*, 34(4):1611–1630, 2013. doi: 10.1137/120875338.

Sergio Barbarossa, Danilo Comminiello, Eleonora Grassucci, Francesco Pezone, Stefania Sardellitti, and Paolo Di Lorenzo. Semantic communications based on adaptive generative models and information bottleneck. *IEEE Communications Magazine*, 61(11):36–41, 2023.

Federico Barbero, Cristian Bodnar, Haitz Sáez de Ocáriz Borde, Michael Bronstein, Petar Veličković, and Pietro Liò. Sheaf neural networks with connection Laplacians. In *Topological, Algebraic and Geometric Learning Workshops 2022*, pp. 28–36. PMLR, 2022.

Amir Beck. *First-order methods in optimization*. SIAM, 2017.

Cristian Bodnar, Francesco Di Giovanni, Benjamin Paul Chamberlain, Pietro Liò, and Michael M. Bronstein. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing

in GNNs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://dblp.org/rec/conf/nips/BodnarGCLB22.

Glen E Bredon. *Sheaf theory*, volume 170. Springer Science & Business Media, 1997.

Liwei Che, Jiaqi Wang, Yao Zhou, and Fenglong Ma. Multimodal federated learning: A survey. *Sensors*, 23(15):6986, 2023.

Fan Chung, Wenbo Zhao, and Mark Kempton. Ranking and sparsifying a connection graph. *Internet Mathematics*, 10(1-2):87–115, 2014.

Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *PMLR*, 2021.

Justin Michael Curry. *Sheaves, cosheaves and applications*. PhD thesis, University of Pennsylvania, 2014.

Gabriele D'Acunto, Fabio Massimo Zennaro, Yorgos Felekis, and Paolo Di Lorenzo. Causal abstraction learning based on the semantic embedding principle. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=J16AIOkjjY.

Gabriele D'Acunto, Paolo Di Lorenzo, and Sergio Barbarossa. Networks of causal abstractions: A sheaf-theoretic framework. *arXiv preprint arXiv:2509.25236*, 2026.

Leonardo Di Nino, Sergio Barbarossa, and Paolo Di Lorenzo. Learning sheaf Laplacian optimizing restriction maps. In *2024 58th Asilomar Conference on Signals, Systems, and Computers*, pp. 59–63. IEEE, 2024.

Leonardo Di Nino, Gabriele D'Acunto, Sergio Barbarossa, and Paolo Di Lorenzo. Learning the structure of connection graphs. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 76–80. IEEE, 2026.

Enmao Diao, Jie Ding, and Vahid Tarokh. HeteroFL: Computation and communication efficient federated learning for heterogeneous clients. In *International Conference on Learning Representations (ICLR)*, 2021.

Canh T Dinh, Tung T Vu, Nguyen H Tran, Minh N Dao, and Hongyu Zhang. A new look and convergence rate of federated multitask learning with Laplacian regularization. *IEEE Transactions on Neural Networks and Learning Systems*, 35(6):8075–8085, 2022.

Nanqing Dong and Irina Voiculescu. Federated contrastive learning for decentralized unlabeled medical images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 378–387. Springer, 2021.

Alan Edelman, Tomás A. Arias, and Steven T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998. doi: 10.1137/S0895479895290954.

Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 109–117, 2004.

Theodoros Evgeniou, Charles A Micchelli, Massimiliano Pontil, and John Shawe-Taylor. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6(4), 2005.

Simone Fiorellino, Claudio Battiloro, Emilio Calvanese Strinati, and Paolo Di Lorenzo. Dynamic relative representations for goal-oriented semantic communications. In *Proc. IEEE EUSIPCO*, pp. 2107–2111, 2024.

Simone Fiorellino, Claudio Battiloro, Emilio Calvanese Strinati, and Paolo Di Lorenzo. Frame-based zero-shot semantic channel equalization for AI-native communications. *IEEE Transactions on Cognitive Communications and Networking*, 2026.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 1322–1333, 2015.

Abdulmomen Ghalkha, Zhuojun Tian, Chaouki Ben Issaid, and Mehdi Bennis. SheafAlign: A sheaf-theoretic framework for decentralized multimodal alignment. *IEEE Communications Letters*, 2026.

Enrico Grimaldi, Mario Edoardo Pandolfo, Gabriele D'Acunto, Sergio Barbarossa, and Paolo Di Lorenzo. Learning network sheaves for AI-native semantic communication. In *2025 59th Asilomar Conference on Signals, Systems, and Computers*, pp. 1692–1696. IEEE, 2025.

Deniz Gündüz, Zhijin Qin, Inaki Estella Aguerri, Harpreet S Dhillon, Zhaohui Yang, Aylin Yener, Kai Kit Wong, and Chan-Byoung Chae. Beyond transmitting bits: Context, semantics, and task-oriented communications. *IEEE Journal on Selected Areas in Communications*, 41(1):5–41, 2022.

Sungwon Han, Sungwon Park, Fangzhao Wu, Sundong Kim, Chuhan Wu, Xing Xie, and Meeyoung Cha. Fedx: Unsupervised federated learning with cross knowledge distillation. In *European Conference on Computer Vision*, pp. 691–707. Springer, 2022.

Jakob Hansen and Robert Ghrist. Toward a spectral theory of cellular sheaves. *Journal of Applied and Computational Topology*, 3(4):315–358, 2019.

Wenke Huang, Mang Ye, and Bo Du. Learn from others and be yourself in heterogeneous federated learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10133–10143. IEEE, 2022.

Wenke Huang, Mang Ye, Zekun Shi, and Bo Du. Generalizable heterogeneous federated cross-correlation and instance similarity learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(2):712–728, 2023.

Wenke Huang, Mang Ye, Zekun Shi, Guancheng Wan, He Li, Bo Du, and Qiang Yang. Federated learning for generalization, robustness, fairness: A survey and benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9387–9406, 2024.

Tomás Hüttebräucker, Simone Fiorellino, Mohamed Sana, Paolo Di Lorenzo, and Emilio Calvanese Strinati. Relative representations of latent spaces enable efficient semantic channel equalization. In *Proc. IEEE GLOBECOM*, 2024.

Chaouki Ben Issaid, Praneeth Vepakomma, and Mehdi Bennis. Tackling feature and sample heterogeneity in decentralized multi-task learning: A sheaf-theoretic approach. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=JlPq0LmApB.

Laurent Jacob, Jean-Philippe Vert, and Francis Bach. Clustered multi-task learning: A convex formulation. *Advances in Neural Information Processing Systems*, 21, 2008.

Martin Jaggi, Virginia Smith, Martin Takáč, Jonathan Terhorst, Sanjay Krishnan, Thomas Hofmann, and Michael I Jordan. Communication-efficient distributed dual coordinate ascent. *Advances in Neural Information Processing Systems*, 27, 2014.

Peter Kairouz, Brendan H. McMahan, et al. *Advances and Open Problems in Federated Learning*, volume 14 of *Foundations and Trends® in Machine Learning*. 2021. doi: 10.1561/2200000083. URL https://inria.hal.science/hal-02406503.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning*, PMLR, pp. 5132–5143, 2020.

Agnan Kessy, Alex Lewin, and Korbinian Strimmer. Optimal whitening and decorrelation. *The American Statistician*, 72(4):309–314, 2018.

Ang Li, Jingwei Sun, Binghui Wang, Lin Duan, Sicheng Li, Yiran Chen, and Hai Li. LotteryFL: Personalized and communication-efficient federated learning with lottery ticket hypothesis on non-iid datasets. *arXiv preprint arXiv:2008.03371*, 2020a.

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

Jiaxiang Li and Shiqian Ma. Federated learning on Riemannian manifolds. *arXiv preprint arXiv:2206.05668*, 2022.

Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of the Third Conference on Machine Learning and Systems (MLSys)*, 2020b.

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.

Chenxin Ma, Virginia Smith, Martin Jaggi, Michael Jordan, Peter Richtárik, and Martin Takác. Adding vs. averaging in distributed primal-dual optimization. In *International Conference on Machine Learning*, pp. 1973–1982. PMLR, 2015.

Connor J McLaughlin and Lili Su. Personalized federated learning via feature distribution adaptation. *Advances in Neural Information Processing Systems*, 37:77038–77059, 2024.

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 (AISTATS)*, volume 54, pp. 1273–1282. PMLR, 2017.

Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *IEEE Symposium on Security and Privacy (S&P)*, 2019.

Runxuan Miao and Erdem Koyuncu. Contrastive and non-contrastive strategies for federated self-supervised representation learning and deep clustering. *IEEE Journal of Selected Topics in Signal Processing*, 18(6):1070–1084, 2024.

Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. Relative representations enable zero-shot latent space communication. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=SrC-nwieGJ`.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87 of *Applied Optimization*. Springer, 2004.

Mario Edoardo Pandolfo, Simone Fiorellino, Emilio Calvanese Strinati, and Paolo Di Lorenzo. Latent space alignment for AI-native mimo semantic communications. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2025.

Mario Edoardo Pandolfo, Enrico Grimaldi, Lorenzo Marinucci, Leonardo Di Nino, Simone Fiorellino, Sergio Barbarossa, and Paolo Di Lorenzo. SEMASIA: A large-scale dataset of semantically structured latent representations. *arXiv preprint arXiv:2605.09485*, 2026.

Herbert Robbins and David Siegmund. A convergence theorem for non negative almost supermartingales and some applications. pp. 233–257. Academic Press, 1971.

Simone Scardapane and Paolo Di Lorenzo. A framework for parallel and distributed training of neural networks. *Neural Networks*, 91:42–54, 2017.

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

Mehdi Setayesh, Mahdi Beitollahi, Yasser H. Khalil, and Hongliang Li. Toward enhancing representation learning in federated multi-task settings. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=nIOIfHHYzk.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy (S&P)*, pp. 3–18. IEEE, 2017.

Amit Singer. Angular synchronization by eigenvectors and semidefinite programming. *Applied and Computational Harmonic Analysis*, 30(1):20–36, 2011.

Amit Singer and H-T Wu. Vector diffusion maps and the connection Laplacian. *Communications on Pure and Applied Mathematics*, 65(8):1067–1144, 2012.

Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. *Advances in Neural Information Processing Systems*, 30, 2017.

Emilio Calvanese Strinati, Paolo Di Lorenzo, et al. Goal-oriented and semantic communication in 6G AI-native networks: The 6G-GOALS approach. In *Proceedings of EuCNC/6G Summit*, pp. 1–6, 2024. doi: 10.1109/EuCNC/6GSummit60053.2024.10597087.

Oscar Thorsted Svendsen, Nikolaj Holst Jakobsen, Fabian Mager, and Hiba Nassar. Improving relative representations with learned anchors and whitened inner products. *arXiv preprint arXiv:2605.30596*, 2026.

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*, 2022.

Johan Thunberg, Florian Bernard, and Jorge Gonçalves. Distributed methods for synchronization of orthogonal matrices over graphs. *Automatica*, 80:243–252, 2017.

Trung-Kien Tran, Ha-Phuong Tran, Thi-Lan Le, and Thanh-Hai Tran. Fedntproto: A prototype-based approach for personalized federated learning. In *2024 International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*, pp. 1–6. IEEE, 2024.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. PMLR, 2020.

Fengda Zhang, Kun Kuang, Long Chen, Zhaoyang You, Tao Shen, Jun Xiao, Yin Zhang, Chao Wu, Fei Wu, Yueting Zhuang, and Xiaolin Li. Federated unsupervised representation learning. *Frontiers of Information Technology & Electronic Engineering*, 24(8):1181–1193, 2023.

Shengdong Zhang, Ehsan Nezhadarya, Homa Fashandi, Jiayi Liu, Darin Graham, and Mohak Shah. Stochastic whitening batch normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10978–10987, 2021.

Yu Zhang and Qiang Yang. Learning sparse task relations in multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Yu Zhang and Dit-Yan Yeung. A convex formulation for learning task relationships in multi-task learning. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, pp. 733–742, 2010.

Weiming Zhuang, Xin Gan, Yonggang Wen, Shuai Zhang, and Shuai Yi. Collaborative unsupervised visual representation learning from decentralized data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4912–4921, 2021.

# A NOTATION

Scalars are denoted by lowercase italic letters, vectors by lowercase boldface, matrices by uppercase boldface, and sets, spaces, and operators by calligraphic letters. Vectors of learnable parameters are denoted by boldface Greek letters.

| | |
|---|---|
| *Sets and indices* | |
| $\mathbb{R}$, $[N]$ | real numbers; the index set $\{1, \dots, N\}$ |
| $\lvert\mathcal{S}\rvert$ | cardinality of the set $\mathcal{S}$ |
| $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ | communication graph, with $\lvert\mathcal{N}\rvert = N$ agents |
| $e_{ij} = (i, j) \in \mathcal{E}$ | edge oriented from head $i$ to tail $j$ |
| $i \trianglelefteq e_{ij}$ | incidence relation between node $i$ and edge $e_{ij}$ |
| $\mathcal{N}(i)^-$, $\mathcal{N}(i)^+$ | neighbors for which $i$ is the head, resp. the tail |
| $\mathcal{N}(i)$ | $\mathcal{N}(i)^- \cup \mathcal{N}(i)^+$ |
| *Data and local models* | |
| $\mathcal{T}_i$, $M_i$ | local dataset of agent $i$ and its size |
| $\mathbf{x}_i^n \in \mathbb{R}^{p_i}$, $P_i$ | $n$-th local sample and its distribution |
| $\mathcal{Y}_i$, $\mathcal{C}_i$ | local output space and target class set |
| $\mathcal{A}$, $K$ | pilot index set and its cardinality $\lvert\mathcal{A}\rvert = K$ |
| $f_{\boldsymbol{\varphi}_i}$, $h_{\boldsymbol{\psi}_i}$ | local encoder and head, with parameters $\boldsymbol{\varphi}_i$, $\boldsymbol{\psi}_i$ |
| $\boldsymbol{\theta}_i = (\boldsymbol{\varphi}_i, \boldsymbol{\psi}_i)$ | backpropagated parameters of agent $i$ |
| $\mathcal{L}_i$, $\mathcal{R}_{\mathcal{A}}$ | local learning objective and gluing penalty |
| $\lambda_{ij}$, $\eta_\theta$ | edge regularization coefficient and stepsize |
| *Sheaf structure* | |
| $\mathcal{F}$ | network sheaf on $\mathcal{G}$ |
| $\mathcal{F}(i) \cong \mathbb{R}^{d_i}$ | node stalk of agent $i$, of dimension $d_i$ |
| $\mathcal{F}(e_{ij}) \cong \mathbb{R}^{d_{ij}}$ | edge stalk, with $d_{ij} = \max(d_i, d_j)$ |
| $\mathcal{F}_{i \trianglelefteq e_{ij}}$ | restriction map from $\mathcal{F}(i)$ to $\mathcal{F}(e_{ij})$ |
| $C^0(\mathcal{G}; \mathcal{F})$, $C^1(\mathcal{G}; \mathcal{F})$ | spaces of 0- and 1-cochains |
| $\boldsymbol{\Delta}$, $\mathbf{L}_{\mathcal{F}} = \boldsymbol{\Delta}^\top \boldsymbol{\Delta}$ | coboundary operator and sheaf Laplacian |
| $\mathcal{TV}(\mathbf{z})$ | sheaf total variation |
| *Latent quantities and transport maps* | |
| $\mathbf{z}_i = f_{\boldsymbol{\varphi}_i}(\mathbf{x}_i)$ | latent representation of agent $i$ |
| $\mathbf{A}_i \in \mathbb{R}^{d_i \times K}$ | pilot feature matrix of agent $i$ |
| $\mathbf{O}_{ji} \in \mathrm{O}(d_i)$ | orthogonal restriction map |
| $\mathbf{V}_{ij} \in \mathrm{St}(d_i, d_j)$ | isometric embedding (semi-orthogonal map) |
| $\mathrm{O}(d)$, $\mathrm{St}(d, k)$ | orthogonal group and Stiefel manifold, Eq. (3) |
| *Operators* | |
| $\lVert\cdot\rVert_2$, $\lVert\cdot\rVert_{\mathrm{F}}$ | Euclidean and Frobenius norms |
| $(\cdot)^\top$, $\mathrm{tr}\{\cdot\}$ | transpose and trace |
| $\ker(\cdot)$, $\mathrm{Im}\,(\cdot)$ | kernel and image of a linear map |
| $\oplus$ | direct sum of vector spaces |
| $\mathbf{I}_k$ | $k \times k$ identity matrix |
| $\mathbf{1}[\cdot]$ | indicator function |
| $\nabla_{\boldsymbol{\varphi}_i}$ | gradient with respect to $\boldsymbol{\varphi}_i$ |

# B EXTENDED RELATED WORK AND POSITIONING

Here we expand the discussion of Sec. 2, detailing how our framework relates to each of the paradigms mentioned there.

**Federated learning under heterogeneity.** A large body of work studies the impact of statistical and system heterogeneity, which can lead to client drift and unstable convergence, and proposes stabilizing mechanisms such as proximal regularization (FedProx) (Li et al., 2020b) and control variates (SCAFFOLD) (Karimireddy et al., 2020). More broadly, these approaches aim to improve

optimization stability while maintaining a *single global model*. In contrast, we focus on settings where enforcing a shared latent representation is inherently restrictive.

**Federated multi-task learning.** Early formulations build upon regularized multi-task learning objectives, where task relationships are captured through shared covariance or relational structures (Evgeniou & Pontil, 2004; Evgeniou et al., 2005; Jacob et al., 2008; Zhang & Yeung, 2010; Zhang & Yang, 2017). In the federated setting, this leads to approaches that jointly optimize local models and their coupling structure via alternating or primal-dual procedures (Smith et al., 2017; Jaggi et al., 2014; Ma et al., 2015). A prominent class of methods leverages graph-based regularization, where a predefined or learned graph encodes similarities among clients and enforces smoothness of model parameters across the network. In particular, Laplacian-based approaches penalize discrepancies between neighboring models through total variation terms (Dinh et al., 2022; Issaid et al., 2025). While these methods capture task relatedness, they operate in the *parameter space* and typically assume homogeneous representations. In contrast, our approach operates in the *representation space* and models relationships between clients up to learnable transformations, enabling alignment across heterogeneous latent spaces and accommodating different architectures or dimensions.

**Multimodal Federated Learning.** Existing MFL approaches are typically categorized based on data heterogeneity into congruent (shared modalities and feature spaces) and incongruent settings, the latter including vertical, transfer, and hybrid scenarios with partially overlapping or distinct modalities. In general, MFL optimizes a global objective by aggregating client-specific multimodal losses, often defined as weighted combinations of modality-wise contributions. However, most methods assume a shared latent space and rely on centralized or weakly decentralized protocols. Recent graph-based approaches such as CoMFed (Badi et al., 2026) address heterogeneity by aligning compressed latent prototypes across clients, but still require a predefined common embedding space and enforce similarity via projections. In contrast, our sheaf-based framework does not assume a global latent space; instead, it enables decentralized learning in which each client maintains its own representation, and cross-client consistency is enforced through edge-dependent geometric alignments without information loss, naturally accommodating heterogeneous representation spaces, including settings where different modalities admit a set of semantically corresponding pilot inputs.

**Federated Representation Learning.** In the supervised setting, existing approaches either enforce consistency via auxiliary losses on latent embeddings (Li et al., 2021), typically assuming a shared global model, or adopt a split architecture with a common encoder and client-specific heads (Collins et al., 2021; McLaughlin & Su, 2024), still requiring a shared latent dimensionality. More recent methods based on prototype alignment (Tan et al., 2022; Tran et al., 2024) partially address heterogeneity but continue to rely on a common embedding space and often centralized training. In contrast, our framework allows each client to learn its own latent space and enforces consistency in a decentralized manner through structured transport maps, so that classical FRL methods can be interpreted as a special case in which all latent spaces share the same dimensionality and are constrained to coincide via identity alignments. Beyond supervised settings, unsupervised and self-supervised approaches operating on unlabeled data aim to improve representation transferability across clients, often leveraging contrastive learning frameworks (Zhuang et al., 2021; Han et al., 2022; Zhang et al., 2023; Miao & Koyuncu, 2024; Ghalkha et al., 2026). Our proposed framework is not limited to any specific local loss or model architecture; thus, it is also amenable to the latter settings.

**Distillation and correlation-based alignment on public data.** A further line of heterogeneous FL achieves interoperability by evaluating all participants on a shared set of inputs rather than by sharing parameters. FedMD (Li & Wang, 2019) assumes a labeled public dataset and has a central server average the participants' class scores into a consensus that each model then distils towards, while FCCL+ (Huang et al., 2022; 2023) removes the labeling requirement and aligns, again through the server, both the cross-correlation matrix of the logits and the instance-similarity matrix of a public batch towards their network-wide averages; broader coverage of this literature is given in Huang et al. (2024). Our pilot set plays the same role of a common reference, but these methods are centralized and drive every participant towards a single global target, whereas we impose pairwise compatibility on the edges of a communication graph and never posit a global latent space. They further rely on the labels of the public data, or on a common label space for the logits, while our gluing penalty uses the pilot inputs alone and compares latent representations of possibly different dimensions. Most importantly, FCCL+ accommodates differing feature dimensions precisely by comparing the instance-similarity matrix, which is invariant to any orthogonal transformation of the feature space:

**Algorithm 1** Sheaf-FRL

**Require:** Oriented $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, pilots $\mathcal{A}$, edge weights $\{\lambda_{ij} > 0\}$, $\eta_\theta > 0$, iterations $T$

1: Initialize $\{\boldsymbol{\theta}_i^0 = (\boldsymbol{\varphi}_i^0, \boldsymbol{\psi}_i^0)\}_{i \in \mathcal{N}}$
2: **for** $t = 0, \dots, T-1$ **do**
3:  **for all** $i \in \mathcal{N}$ **in parallel do**
4:  $\mathbf{A}_i^t \leftarrow [f_{\boldsymbol{\varphi}_i^t}(\mathbf{x}_i^k)]_{k \in \mathcal{A}}$
5:  **end for**
6:  Each node $i$ broadcasts $\mathbf{A}_i^t$ to those $j \in \mathcal{N}(i)$
7:  **for all** $i \in \mathcal{N}$ **in parallel do**
8:  **for** $j \in \mathcal{N}(i)$ **do**
9:  **if** $j \in \mathcal{N}(i)^-$ **then**
10:  $[\mathbf{U}, \boldsymbol{\Sigma}, \mathbf{W}^\top] \leftarrow \mathrm{thinSVD}(\mathbf{A}_i^t \mathbf{A}_j^{t^\top})$
11:  $\mathbf{V}_{ij}^t \leftarrow \mathbf{U}\mathbf{W}^\top$
12:  **else**
13:  $[\mathbf{U}, \boldsymbol{\Sigma}, \mathbf{W}^\top] \leftarrow \mathrm{thinSVD}(\mathbf{A}_j^t \mathbf{A}_i^{t^\top})$
14:  $\mathbf{V}_{ji}^t \leftarrow \mathbf{U}\mathbf{W}^\top$
15:  **end if**
16:  **end for**
17:  **end for**
18:  **for all** $i \in \mathcal{N}$ **in parallel do**
19:  $\boldsymbol{\theta}_i^{t+1} \leftarrow \boldsymbol{\theta}_i^t - \eta_\theta \left( \nabla_{\boldsymbol{\theta}_i} \mathcal{L}_i(\boldsymbol{\theta}_i^t) + \mathbf{r}_i^t \right)$ ; with $\mathbf{r}_i^t$ as in Eq. (12)
20:  **end for**
21: **end for**

it constrains representations up to exactly the gauge freedom we exploit, without ever constructing the transformation, so that a participant still cannot consume a neighbor's latent vector after training, whereas our restriction maps are the object being learned and are what makes transport possible.

**Positioning of this paper.** Overall, our framework generalizes across all paradigms discussed above, which differ mainly in *what* is assumed to be shared among agents (a single global model, a coupled set of parameters, a common multimodal embedding, or a shared latent space) rather than in the underlying goal of exploiting relatedness across agents. Indeed, methods introduced within one paradigm, such as prototype alignment in FRL or graph-based regularization in FMTL, are often applicable to the others as well. Rather than enforcing any of these forms of sharing, we learn edge-wise geometric transport maps that relate neighboring latent spaces and promote their semantic compatibility. In the homogeneous case, our formulation becomes compatible with classical FL and recovers its shared-representation structure as a limiting case, together with orthogonal alignment, synchronization, and connection-Laplacian models. More generally, it extends graph-based FMTL and Laplacian regularization through a *learnable sheaf Laplacian* acting on latent representations rather than model parameters. Additionally, it relaxes the shared, predefined embedding space assumed by MFL approaches such as CoMFed, and recovers FRL methods as the special case in which all latent spaces coincide via identity alignments. Finally, pilot-restricted gluing enables communication-efficient alignment, while manifold-constrained transport maps admit decentralized optimization through local model updates and closed-form edge-wise Procrustes steps.

## C Pseudocode of the Sheaf-FRL Algorithm

In this section, we provide additional details on the proposed decentralized algorithm, reported in Alg. 1. The method follows an alternating minimization strategy, where each communication round consists of local feature extraction, decentralized alignment updates, and gradient-based optimization of the neural parameters.

**Heterogeneous case.** Alg. 1 implements a fully decentralized procedure in which agents iteratively update both their local representations and the alignment maps across the network. At each iteration $t$, every agent $i$ first computes its pilot feature matrix $\mathbf{A}_i^t$ using the current encoder (lines 3–5). These pilot representations provide a compact summary of the local latent space and constitute

**Algorithm 2** Sheaf-FRL (homogeneous case)

**Require:** Oriented $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, pilots $\mathcal{A}$, edge weights $\{\lambda_{ij} > 0\}$, $\eta_\theta > 0$, iterations $T$

1: Initialize $\{\boldsymbol{\theta}_i^0 = (\boldsymbol{\varphi}_i^0, \boldsymbol{\psi}_i^0)\}_{i\in\mathcal{N}}$ and $\{\mathbf{O}_{ji}^0 = \mathbf{I}_d\}_{j\in\mathcal{N}(i), i\in\mathcal{N}}$
2: **for** $t = 0, \dots, T-1$ **do**
3:  **for all** $i \in \mathcal{N}$ **in parallel do**
4:  $\mathbf{A}_i^t \leftarrow [f_{\boldsymbol{\varphi}_i^t}(\mathbf{x}_i^k)]_{k\in\mathcal{A}}$
5:  **end for**
6:  Each node $i$ exchanges $\mathbf{O}_{ji}^t \mathbf{A}_i^t$ with all $j \in \mathcal{N}(i)$
7:  **for all** $i \in \mathcal{N}$ **in parallel do**
8:  **for all** $j \in \mathcal{N}(i)^+$ **do**
9:  $[\mathbf{U}, \boldsymbol{\Sigma}, \mathbf{W}^\top] \leftarrow \mathrm{SVD}\big(\mathbf{O}_{ij}^t \mathbf{A}_j^t \mathbf{A}_i^{t^\top}\big)$
10:  $\mathbf{O}_{ji}^t \leftarrow \mathbf{U}\mathbf{W}^\top$
11:  **end for**
12:  **end for**
13:  **for all** $i \in \mathcal{N}$ **in parallel do**
14:  $\boldsymbol{\theta}_i^{t+1} \leftarrow \boldsymbol{\theta}_i^t - \eta_\theta \left(\nabla_{\boldsymbol{\theta}_i} \mathcal{L}_i(\boldsymbol{\theta}_i^t) + \mathbf{r}_i^t\right)$
15:  **end for**
16: **end for**

the only information exchanged across agents. The pilot features are then broadcast to neighboring nodes (line 6). Upon receiving $\{\mathbf{A}_j^t\}_{j\in\mathcal{N}(i)}$, each agent locally updates the alignment maps associated with its incident edges (lines 7–17). In particular, for each neighbor $j$, agent $i$ solves a small Procrustes problem based on the cross-covariance of the corresponding pilot features. The orientation of the graph determines whether the update corresponds to an incoming map $\mathbf{V}_{ij}^t$ or an outgoing map $\mathbf{V}_{ji}^t$, ensuring consistency with the underlying sheaf structure. Finally, each agent performs a gradient-based update of its local parameters $\boldsymbol{\theta}_i$ (lines 18–20), combining the gradient of the local loss with the contribution of the sheaf-based regularization. This step enforces alignment of neighboring representations while preserving local task objectives. Overall, the algorithm alternates between closed-form alignment updates and gradient-based parameter updates. All computations are carried out locally, and communication is limited to low-dimensional pilot features, resulting in an efficient and scalable decentralized learning procedure.

**Homogeneous case.** When the latent dimensions are homogeneous, i.e., $d_i = d$ for all $i \in \mathcal{N}$, the alignment maps reduce to orthogonal transformations $\mathbf{O}_{ij} \in \mathrm{O}(d)$, for each $(i, j) \in \mathcal{E}$. In this setting, as discussed in Sec. 4 and further detailed in App. D, the local computational cost of the alignment step can be reduced without increasing communication. Alg. 2 presents a simplified Sheaf-FRL protocol tailored to this scenario. For each agent $i$, we initialize to the identity the maps used to transport its pilot representations toward its neighbors, i.e., $\mathbf{O}_{ji}^0 = \mathbf{I}_d$ for all $j \in \mathcal{N}(i)$. At each iteration $t$, each agent first computes its pilot representations $\mathbf{A}_i^t$ and sends to each neighbor $j \in \mathcal{N}(i)$ the transported features $\mathbf{O}_{ji}^t \mathbf{A}_i^t$. Notably, since we exploit the reparameterization in Eq. (20), when $j \in \mathcal{N}^-(i)$, we have $\mathbf{O}_{ji}^t = \mathbf{O}_{ji}^0 = \mathbf{I}_d$ for all $t = 0, \dots, T-1$, so that the transmitted features coincide with the original ones and no additional communication cost is incurred. From a local computation perspective, since transported pilot representations are communicated instead of raw ones, agent $i$ does not need to solve the orthogonal Procrustes problem arising in (P1a) for evaluating the alignment and the gradient required for the local parameter update. As a consequence, only the SVDs for $j \in \mathcal{N}(i)^+$ are required, effectively halving the computational cost of the alignment step, as specified in lines 8-10 of Alg. 2. Overall, this variant preserves the communication efficiency of the general algorithm while reducing the computational burden (cf. Tab. 1).

## D Trade-off between Privacy, Communication, and Local Computation

Our proposed algorithmic solution in Sec. 4 leverages sharing of raw latent representations to lower the communication cost, according to the objective in Prob. 1. This choice is deliberate. In principle, the number of alignment updates could be halved, since for each edge $e_{ij}$ the isometric embed-

ding is only required at the lower-dimensional node. However, this would require each node $i$ to transmit its transported representations $\mathbf{V}_{ji}\mathbf{A}_i^t$, whose dimension is at least $d_i$, in order to evaluate Eq. (11), thereby increasing the communication cost. We therefore exchange the raw pilot features $\mathbf{A}_i^t$ and let both endpoints compute the corresponding restriction maps locally, shifting the burden from communication to computation, which is well aligned with the increasing computational capabilities of modern edge devices. This is compatible with standard regulatory constraints on privacy, for instance when latent representations do not allow direct identification of individuals or when appropriate safeguards such as aggregation, anonymization, or secure communication protocols are in place, as commonly assumed in federated learning systems (Kairouz et al., 2021; McMahan et al., 2017). However, as highlighted in Sec. 4, in some circumstances more stringent regulatory constraints on privacy impair the sharing of raw latent representations. For example, this occurs when latent representations may still encode sensitive or identifiable information and are therefore susceptible to reconstruction or inference attacks, such as model inversion (Fredrikson et al., 2015), membership inference (Shokri et al., 2017), or attribute inference (Melis et al., 2019), making their direct exchange incompatible with strict privacy requirements. Our proposed approach naturally extends to this more stringent setting. Instead of sharing raw latent representations, agents exchange their transported counterparts, i.e., latent features mapped through the restriction maps associated with the edges. This limits the direct exposure of local features. However, this enhanced privacy comes at a cost: it requires additional local computation and increases the communication overhead, as detailed below.

Concerning the local computational cost, each agent $i \in \mathcal{N}$ needs to encode its latent representations toward both sets of neighbors $\mathcal{N}(i)^-$ and $\mathcal{N}(i)^+$. To this end, we adopt the formulation in Eq. (7) *before reparameterization*, where both orthogonal and Stiefel maps explicitly appear, allowing consistent encoding across heterogeneous neighbors. Specifically, $i$ must communicate $\mathbf{V}_{ji}^t\mathbf{A}_i^t \in \mathbb{R}^{d_j \times K}$ to neighbors $j \in \mathcal{N}(i)^+$, where $\mathbf{V}_{ji}^t \in \mathrm{St}(d_j, d_i)$ is the outgoing embedding matrix, and $\mathbf{O}_{ji}^t\mathbf{A}_i^t \in \mathbb{R}^{d_i \times K}$ to neighbors $j \in \mathcal{N}(i)^-$, where $\mathbf{O}_{ji}^t \in \mathrm{O}(d_i)$ is an orthogonal transformation. Hence, in this setting we cannot exploit the reparameterized form in Eq. (7), which avoids the explicit use of orthogonal maps. In detail, after receiving the encoded latent representations from its neighbors, namely

$$\mathbf{V}_{ij}^t\mathbf{A}_j^t \text{ from } j \in \mathcal{N}(i)^- \quad \text{and} \quad \mathbf{O}_{ij}^t\mathbf{A}_j^t \text{ from } j \in \mathcal{N}(i)^+ ; \tag{14}$$

under stringent privacy constraints, agent $i$ must solve the following problems to update its restriction maps:

$$\mathbf{O}_{ji}^t = \underset{\mathbf{O}_{ji}\in\mathrm{O}(d_i)}{\arg\min} \left\|\mathbf{O}_{ji}\mathbf{A}_i^t - \mathbf{V}_{ij}^t\mathbf{A}_j^t\right\|_{\mathrm{F}}^2, \quad \forall\, j \in \mathcal{N}(i)^- , \tag{P1c}$$

$$\mathbf{V}_{ji}^t = \underset{\mathbf{V}_{ji}\in\mathrm{St}(d_j,d_i)}{\arg\min} \left\|\mathbf{O}_{ij}^t\mathbf{A}_j^t - \mathbf{V}_{ji}\mathbf{A}_i^t\right\|_{\mathrm{F}}^2, \quad \forall\, j \in \mathcal{N}(i)^+ . \tag{P1d}$$

Unlike the case of standard privacy constraints, where for $j \in \mathcal{N}(i)^-$ we can only update the incoming restriction map $\mathbf{V}_{ij}$ and solve Prob. (P1a) via a thin SVD at a computational cost of $\mathcal{O}(d_i d_j^2)$ with $d_j \leq d_i$, Prob. (P1c) is a canonical orthogonal Procrustes problem. Its solution is obtained via a classical SVD of $\mathbf{V}_{ij}^t\mathbf{A}_j^t\mathbf{A}_i^{t^\top}$ at a cost of $\mathcal{O}(d_i^3)$, thus leading to an increase in the local computational burden. Conversely, similarly to Problem (P1b), for $j \in \mathcal{N}(i)^+$ the update of the outgoing restriction map $\mathbf{V}_{ji}$ can be computed via a thin SVD of $\mathbf{O}_{ij}^t\mathbf{A}_j^t\mathbf{A}_i^{t^\top}$ at a cost of $\mathcal{O}(d_j d_i^2)$, where in this case $d_j \geq d_i$.

We remark that the costs above account only for the SVD factorization step. Constructing the cross-covariance matrix $\mathbf{A}_j^t\mathbf{A}_i^{t^\top} \in \mathbb{R}^{d_j \times d_i}$ (or its transported counterpart under strict privacy), requires an additional $\mathcal{O}(d_i d_j K)$ operations per neighbor, for both privacy regimes. Since the number of pilots $K$ is typically much larger than the latent dimensions in practice (cf. App. I.1), this construction step dominates the overall local computational burden and is reported explicitly in Tab. 1.

Please notice that more stringent privacy constraints affect only the update of the restriction maps. Indeed, the local gradient computation in Eq. (11) remains unchanged:

$$\begin{aligned}\nabla_{\boldsymbol{\varphi}_i}\mathcal{R}_{\mathcal{A}}|_i =& \frac{1}{K}\Bigg[\sum_{j\in\mathcal{N}(i)^-}\lambda_{ij}\sum_{k\in\mathcal{A}}\left(\nabla_{\boldsymbol{\varphi}_i}f_{\boldsymbol{\varphi}_i}(\mathbf{x}_i^k)\right)^\top\mathbf{O}_{ji}^\top\left(\mathbf{O}_{ji}f_{\boldsymbol{\varphi}_i}\left(\mathbf{x}_i^k\right)-\mathbf{V}_{ij}f_{\boldsymbol{\varphi}_j}\left(\mathbf{x}_j^k\right)\right)\\ &-\sum_{j\in\mathcal{N}(i)^+}\lambda_{ij}\sum_{k\in\mathcal{A}}\left(\nabla_{\boldsymbol{\varphi}_i}f_{\boldsymbol{\varphi}_i}(\mathbf{x}_i^k)\right)^\top\mathbf{V}_{ji}^\top\left(\mathbf{O}_{ij}f_{\boldsymbol{\varphi}_j}\left(\mathbf{x}_j^k\right)-\mathbf{V}_{ji}f_{\boldsymbol{\varphi}_i}\left(\mathbf{x}_i^k\right)\right)\Bigg]\\ =&\frac{1}{K}\Bigg[\sum_{j\in\mathcal{N}(i)^-}\lambda_{ij}\sum_{k\in\mathcal{A}}\left(\nabla_{\boldsymbol{\varphi}_i}f_{\boldsymbol{\varphi}_i}(\mathbf{x}_i^k)\right)^\top\left(f_{\boldsymbol{\varphi}_i}\left(\mathbf{x}_i^k\right)-\mathbf{O}_{ji}^\top\mathbf{V}_{ij}f_{\boldsymbol{\varphi}_j}\left(\mathbf{x}_j^k\right)\right)\\ &-\sum_{j\in\mathcal{N}(i)^+}\lambda_{ij}\sum_{k\in\mathcal{A}}\left(\nabla_{\boldsymbol{\varphi}_i}f_{\boldsymbol{\varphi}_i}(\mathbf{x}_i^k)\right)^\top\left(\mathbf{V}_{ji}^\top\mathbf{O}_{ij}f_{\boldsymbol{\varphi}_j}\left(\mathbf{x}_j^k\right)-f_{\boldsymbol{\varphi}_i}\left(\mathbf{x}_i^k\right)\right)\Bigg]\\ \overset{(a)}{=}&\frac{1}{K}\sum_{j\in\mathcal{N}(i)}\lambda_{ij}\sum_{k\in\mathcal{A}}\left(\nabla_{\boldsymbol{\varphi}_i}f_{\boldsymbol{\varphi}_i}(\mathbf{x}_i^k)\right)^\top\left(f_{\boldsymbol{\varphi}_i}\left(\mathbf{x}_i^k\right)-\mathbf{O}_{ji}^\top\mathbf{V}_{ij}f_{\boldsymbol{\varphi}_j}\left(\mathbf{x}_j^k\right)\right);\end{aligned} \tag{15}$$

where in $(a)$, for those $j\in\mathcal{N}(i)^+$ we used that, under the reparameterization exploited in Eq. (7), we have $\mathbf{V}_{ji}^\top\mathbf{O}_{ij}=\mathbf{O}_{ji}^\top\mathbf{V}_{ij}$. This also shows the equivalence between Eq. (11) and Eq. (15).

Regarding the communication cost, sharing transported latent representations increases the communication overhead toward neighbors $j\in\mathcal{N}(i)^+$. Under standard privacy constraints, agent $i$ transmits $\mathbf{A}_i^t\in\mathbb{R}^{d_i\times K}$, whereas in the more stringent setting it must send $\mathbf{V}_{ji}\mathbf{A}_i^t\in\mathbb{R}^{d_j\times K}$, with $d_j\geq d_i$. Thus, the outgoing payload scales with $d_j$, and strictly increases whenever $d_j>d_i$.

The *incoming* volume received by agent $i$ also changes under strict privacy, but asymmetrically across the two neighbor sets. From $j\in\mathcal{N}(i)^+$, agent $i$ receives the orthogonally transformed representation $\mathbf{O}_{ij}^t\mathbf{A}_j^t\in\mathbb{R}^{d_j\times K}$, whose dimension is unchanged relative to standard privacy. From $j\in\mathcal{N}(i)^-$, however, agent $i$ now receives the already-embedded representation $\mathbf{V}_{ij}^t\mathbf{A}_j^t\in\mathbb{R}^{d_i\times K}$ rather than the raw $\mathbf{A}_j^t\in\mathbb{R}^{d_j\times K}$, so the incoming payload from this neighbor set grows from $d_j$ to $d_i$. The resulting incoming communication cost is

$$\mathcal{O}(|\mathcal{N}(i)^-|d_iK)\quad\text{(standard: }\mathcal{O}(K\textstyle\sum_{j\in\mathcal{N}(i)^-}d_j))\qquad+\qquad\mathcal{O}\Big(K\textstyle\sum_{j\in\mathcal{N}(i)^+}d_j\Big)\quad\text{(unchanged).}$$

Our modified Sheaf-FRL protocols under stringent privacy regulatory constraints for both the heterogeneous and homogeneous cases are given in Algs. 3 and 4, respectively. Additionally, Tab. 1 details the privacy vs. communication and computational costs trade-off.

**Non-identifiability and privacy guarantees.** Tab. 1 highlights a clear trade-off. Under stricter privacy requirements, the proposed protocol incurs higher communication and local computational costs, since agents exchange transported latent representations and must explicitly update the corresponding restriction maps. The benefit, however, is a stronger form of privacy: the communicated messages no longer reveal the latent pilots in their original coordinates. More precisely, when only transported pilots are shared, an observer that does not know the underlying restriction maps cannot uniquely reconstruct the original latent representations, but can only recover geometric information that is invariant under isometries, as formalized in the following proposition.

**Proposition D.1** (Non-identifiability of latent pilots under unknown maps)**.** *Let $\mathbf{A}\in\mathbb{R}^{d\times K}$ be a matrix of latent pilot representations, and let $\mathbf{Y}=\mathbf{T}\mathbf{A}$, where $\mathbf{T}$ is an unknown restriction map satisfying $\mathbf{T}^\top\mathbf{T}=\mathbf{I}_d$. This includes both the homogeneous case $\mathbf{T}\in\mathrm{O}(d)$ and the heterogeneous case $\mathbf{T}\in\mathrm{St}(D,d)$ with $D\geq d$. Then the latent pilots $\mathbf{A}$ are not uniquely identifiable from $\mathbf{Y}$ without knowledge of $\mathbf{T}$. In particular, $\mathbf{A}$ is determined only up to an orthogonal transformation.*

*Proof.* Let $\mathbf{Q}\in\mathrm{O}(d)$ and $\mathbf{A}'=\mathbf{Q}\mathbf{A}$. Then $\mathbf{Y}=\mathbf{T}\mathbf{A}=\mathbf{T}\mathbf{Q}^\top\mathbf{A}'$, and $(\mathbf{T}\mathbf{Q}^\top)^\top(\mathbf{T}\mathbf{Q}^\top)=\mathbf{I}_d$. Thus $\mathbf{T}\mathbf{Q}^\top$ is a valid map, implying that $\mathbf{A}$ is identifiable only up to orthogonal transformations. □

Table 1: Trade-off between privacy and communication and local computation cost per agent. The local computation refers to the update of the restriction maps, i.e., the part influenced by more stringent privacy constraints. To aid comparison, for the heterogeneous case we highlight in blue the larger of $d_i$ and $d_j$ (either one when $d_i = d_j$) when considering either $j \in \mathcal{N}(i)^-$ or $j \in \mathcal{N}(i)^+$. In red, we instead highlight the (potential) increase in cost implied by more stringent privacy constraints.

| | | Standard Privacy | Strict Privacy |
|---|---|---|---|
| Heterogeneous | *Communication* | | |
| | Outgoing | $\mathcal{O}(\lvert\mathcal{N}(i)\rvert d_i K)$ | $\mathcal{O}\big(\lvert\mathcal{N}(i)^-\rvert d_i K + K\sum_{j\in\mathcal{N}(i)^+} d_j\big)$ |
| | Incoming | $\mathcal{O}(K\sum_{j\in\mathcal{N}(i)} d_j)$ | $\mathcal{O}\big(\lvert\mathcal{N}(i)^-\rvert d_i K + K\sum_{j\in\mathcal{N}(i)^+} d_j\big)$ |
| | *Local computation* | | |
| | Cross-covariance | $\mathcal{O}\big(K\big(\sum_{j\in\mathcal{N}(i)^-} d_i d_j + \sum_{j\in\mathcal{N}(i)^+} d_i d_j\big)\big)$ | $\mathcal{O}\big(K\big(\lvert\mathcal{N}(i)^-\rvert d_i^2 + \sum_{j\in\mathcal{N}(i)^+} d_i d_j\big)\big)$ |
| | SVD | $\mathcal{O}\big(d_i\sum_{j\in\mathcal{N}(i)^-} d_j^2 + d_i^2\sum_{j\in\mathcal{N}(i)^+} d_j\big)$ | $\mathcal{O}\big(\lvert\mathcal{N}(i)^-\rvert d_i^3 + d_i^2\sum_{j\in\mathcal{N}(i)^+} d_j\big)$ |
| Homogeneous | *Communication* | | |
| | Outgoing | $\mathcal{O}(\lvert\mathcal{N}(i)\rvert dK)$ | $\mathcal{O}(\lvert\mathcal{N}(i)\rvert dK)$ |
| | Incoming | $\mathcal{O}(\lvert\mathcal{N}(i)\rvert dK)$ | $\mathcal{O}(\lvert\mathcal{N}(i)\rvert dK)$ |
| | *Local computation* | | |
| | Cross-covariance | $\mathcal{O}(K\lvert\mathcal{N}(i)^+\rvert d^2)$ | $\mathcal{O}(K\lvert\mathcal{N}(i)^- \cup \mathcal{N}(i)^+\rvert d^2)$ |
| | SVD | $\mathcal{O}(\lvert\mathcal{N}(i)^+\rvert d^3)$ | $\mathcal{O}(\lvert\mathcal{N}(i)^- \cup \mathcal{N}(i)^+\rvert d^3)$ |

**Algorithm 3** Sheaf-FRL under stringent privacy regulatory constraints

**Require:** Oriented $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, pilots $\mathcal{A}$, edge weights $\{\lambda_{ij} > 0\}$, $\eta_\theta > 0$, iterations $T$

Initialize $\{\boldsymbol{\theta}_i^0 = (\boldsymbol{\varphi}_i^0, \boldsymbol{\psi}_i^0)\}_{i\in\mathcal{N}}$, $\{\mathbf{O}_{ji}^0 \in \mathrm{O}(d_i)\}_{j\in\mathcal{N}(i)^-,\, i\in\mathcal{N}}$, $\{\mathbf{V}_{ji}^0 \in \mathrm{St}(d_j, d_i)\}_{j\in\mathcal{N}(i)^+,\, i\in\mathcal{N}}$
**for** $t = 0, \dots, T-1$ **do**
  **for all** $i \in \mathcal{N}$ **in parallel do**
    $\mathbf{A}_i^t \leftarrow [f_{\boldsymbol{\varphi}_i^t}(\mathbf{x}_i^k)]_{k\in\mathcal{A}}$
  **end for**
  Each node $i$ broadcasts $\mathbf{O}_{ji}^t\mathbf{A}_i^t$ to those $j \in \mathcal{N}(i)^-$ and $\mathbf{V}_{ji}^t\mathbf{A}_i^t$ to $j \in \mathcal{N}(i)^+$
  **for all** $i \in \mathcal{N}$ **in parallel do**
    **for** $j \in \mathcal{N}(i)$ **do**
      **if** $j \in \mathcal{N}(i)^-$ **then**
        $[\mathbf{U}, \boldsymbol{\Sigma}, \mathbf{W}^\top] \leftarrow \mathrm{SVD}(\mathbf{V}_{ij}^t\mathbf{A}_j^t{\mathbf{A}_i^t}^\top)$
        $\mathbf{O}_{ji}^{t+1} \leftarrow \mathbf{U}\mathbf{W}^\top$
      **else**
        $[\mathbf{U}, \boldsymbol{\Sigma}, \mathbf{W}^\top] \leftarrow \mathrm{thinSVD}(\mathbf{O}_{ij}^t\mathbf{A}_j^t{\mathbf{A}_i^t}^\top)$
        $\mathbf{V}_{ji}^{t+1} \leftarrow \mathbf{U}\mathbf{W}^\top$
      **end if**
    **end for**
  **end for**
  **for all** $i \in \mathcal{N}$ **in parallel do**
    $\boldsymbol{\theta}_i^{t+1} \leftarrow \boldsymbol{\theta}_i^t - \eta_\theta\left(\nabla_{\boldsymbol{\theta}_i}\mathcal{L}_i(\boldsymbol{\theta}_i^t) + \mathbf{r}_i^t\right)$
  **end for**
**end for**

## E Additional discussion of our framework

This section collects additional material on the framework introduced in Sec. 3. We discuss, in order, the choice of the edge stalk dimensionality, the heterogeneous geometry of local latent spaces, the exact centralized formulation under global alignment, the geometric consistency enforced by the restriction maps, and the homogeneous setting in which all node stalks share the same dimensionality.

**Embedding vs. compression.** Choosing the edge stalk dimensionality $d_{ij} = \max(d_i, d_j)$ avoids introducing an information bottleneck when aligning heterogeneous latent representations. In contrast,

**Algorithm 4** Sheaf-FRL under stringent privacy regulatory constraints (homogeneous case)

**Require:** Oriented $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, pilots $\mathcal{A}$, edge weights $\{\lambda_{ij} > 0\}$, $\eta_\theta > 0$, iterations $T$

Initialize $\{\boldsymbol{\theta}_i^0 = (\boldsymbol{\varphi}_i^0, \boldsymbol{\psi}_i^0)\}_{i\in\mathcal{N}}$, $\{\mathbf{O}_{ji}^0 \in \mathrm{O}(d)\}_{j\in\mathcal{N}(i), i\in\mathcal{N}}$
**for** $t = 0, \dots, T-1$ **do**
  **for all** $i \in \mathcal{N}$ **in parallel do**
    $\mathbf{A}_i^t \leftarrow [f_{\boldsymbol{\varphi}_i^t}(\mathbf{x}_i^k)]_{k\in\mathcal{A}}$
  **end for**
  Each node $i$ exchanges $\mathbf{O}_{ji}^t \mathbf{A}_i^t$ with all $j \in \mathcal{N}(i)$
  **for all** $i \in \mathcal{N}$ **in parallel do**
    **for all** $j \in \mathcal{N}(i)$ **do**
      $[\mathbf{U}, \boldsymbol{\Sigma}, \mathbf{W}^\top] \leftarrow \mathrm{SVD}\big(\mathbf{O}_{ij}^t \mathbf{A}_j^t \mathbf{A}_i^{t^\top}\big)$
      $\mathbf{O}_{ji}^{t+1} \leftarrow \mathbf{U}\mathbf{W}^\top$
    **end for**
  **end for**
  **for all** $i \in \mathcal{N}$ **in parallel do**
    $\boldsymbol{\theta}_i^{t+1} \leftarrow \boldsymbol{\theta}_i^t - \eta_\theta \left(\nabla_{\boldsymbol{\theta}_i} \mathcal{L}_i(\boldsymbol{\theta}_i^t) + \mathbf{r}_i^t\right)$
  **end for**
**end for**

as done for instance in (Badi et al., 2026), enforcing $d_{ij} \leq \min(d_i, d_j)$ would require projecting both representations onto a shared lower-dimensional subspace, implicitly assuming the existence of a common latent structure and potentially discarding node-specific information. Moreover, such a projection only enforces alignment on the retained subspace, leaving the remaining components unconstrained. As a result, the learned restriction maps capture consistency only up to an information loss, potentially leading to inconsistencies when transferring latent representations across models. Our choice instead pursues a geometric notion of consistency across representations, and preserves the semantics of local representations. Importantly, this also makes our framework well-suited for downstream transfer of latent representations among pre-trained models, where alignment based on compression may result in performance degradation on downstream local tasks.

**Heterogeneous latent space geometry.** We expand here on the whitening and coloring layers introduced in Sec. 3.2. In our setting, the encoders $f_{\boldsymbol{\varphi}_i}$ are trained independently, on different data, and possibly with different architectures. Then, it is reasonable to expect the induced latent spaces $\mathcal{F}(i)$ to exhibit heterogeneous geometry, in the sense of different scales and anisotropies (Pandolfo et al., 2026; Svendsen et al., 2026). This may occur even when dimensionalities $d_i$ coincide. However, the geometric restriction maps considered in this work are effective for measuring inconsistency only when the latent representations $\mathbf{z}_i$ and $\mathbf{z}_j$ adhere to a common geometric reference on the edge stalk. Thus, to make representations from distinct node stalks comparable once transported, each encoder includes a final whitening layer (Zhang et al., 2021) that approximately normalizes the latent representations to zero mean and identity covariance. This normalization reduces the node-specific geometry to a common Euclidean reference. We equip each head with an initial coloring layer, defined as the left inverse of the whitening map applied by the encoder. Along the local path the two compose to the identity, so the local objective $\mathcal{L}_i(\boldsymbol{\theta}_i)$ is invariant to the choice of geometric reference. The coloring is instead essential on the communication path: a representation transported from a neighbor arrives in the common whitened reference, and must be returned to the local coordinates in which $h_{\boldsymbol{\psi}_i}$ was trained.

Concretely, write $f_{\boldsymbol{\varphi}_i} = w_i \circ g_{\boldsymbol{\varphi}_i}$, where $g_{\boldsymbol{\varphi}_i}$ is the trainable backbone and $w_i$ is the whitening layer of Zhang et al. (2021), mapping $\mathbf{u} \in \mathbb{R}^{d_i}$ to

$$w_i(\mathbf{u}) = \mathbf{W}_i \boldsymbol{\Lambda}_i^{-1/2} (\mathbf{u} - \boldsymbol{\mu}_i)\,, \tag{16}$$

with $\boldsymbol{\mu}_i$ and $\boldsymbol{\Lambda}_i$ the mean and $\epsilon$-regularized diagonal variance used for standardization (batch statistics during training, their accumulated estimates at inference) and $\mathbf{W}_i \in \mathbb{R}^{d_i \times d_i}$ the whitening matrix. The affine transformation of the original layer is disabled here, so that $w_i$ is a pure normalization, and the coloring layer is its left inverse, $c_i = w_i^{-1}$, so that $h_{\boldsymbol{\psi}_i} = \tilde{h}_{\boldsymbol{\psi}_i} \circ c_i$.

The whitening matrix is not optimized by the training objective. At each forward pass it is updated by a stochastic step on a whitening criterion; for the Kullback–Leibler criterion of Zhang et al. (2021)

this reads

$$\mathbf{W}_i \leftarrow \mathbf{W}_i - \alpha\big(\mathbf{W}_i\widehat{\mathbf{\Sigma}}_i\mathbf{W}_i^\top - \mathbf{I}_{d_i}\big)\mathbf{W}_i\,, \qquad \mathbf{W}_i \leftarrow \tfrac{1}{2}\big(\mathbf{W}_i + \mathbf{W}_i^\top\big)\,, \tag{17}$$

where $\widehat{\mathbf{\Sigma}}_i$ is the sample covariance of the standardized batch and $\alpha$ is a fixed step size; the symmetrization selects the ZCA transform, the unique symmetric whitening matrix. In the backward pass the whitening matrix is held fixed and the derivative with respect to $\mathbf{W}_i$ is never formed (Zhang et al., 2021), while the standardization statistics are differentiated through as in batch normalization. The whitening transform is therefore a state variable, like the running statistics of batch normalization, and does not belong to $\boldsymbol{\theta}_i$.

Two consequences follow. First, with the affine disabled the construction introduces no trainable parameters at all, and $c_i \circ w_i$ being the identity yields $h_{\boldsymbol{\psi}_i} \circ f_{\boldsymbol{\varphi}_i} = \tilde{h}_{\boldsymbol{\psi}_i} \circ g_{\boldsymbol{\varphi}_i}$: the local objective $\mathcal{L}_i$ and its gradient coincide with those of the model without whitening and coloring, whose optimization is therefore left unchanged. Second, the gluing penalty does see the whitened geometry, since $\mathbf{A}_i = [\,w_i(g_{\boldsymbol{\varphi}_i}(\mathbf{x}_i^k))\,]_{k\in\mathcal{A}}$; as $\mathbf{W}_i$ is constant in the backward pass, $\nabla_{\boldsymbol{\varphi}_i}\mathcal{R}_{\mathcal{A}}$ never differentiates the inverse square root of the full covariance, and the only inverse it traverses is the $\epsilon$-regularized diagonal $\mathbf{\Lambda}_i^{-1/2}$, whose norm is bounded by $\epsilon^{-1/2}$. The smoothness required by Assumption 2 is therefore inherited from $g_{\boldsymbol{\varphi}_i}$, with no condition on the conditioning of $\mathbf{\Sigma}_i$.

**Exact global alignment: the centralized formulation.** In Sec. 3.2 we mention that requiring the learned representations to be a global section of the learned sheaf is the idealized notion of network-wide alignment, and that such a requirement is impractical. We make both statements precise here. Recall that a valuation $\mathbf{z}^\star \in C^0(\mathcal{G};\mathcal{F})$ belonging to $\ker(\mathbf{L}_\mathcal{F})$ is said *global section*, and can be thought of as an assignment of latent representations to the node stalks that does not break local rules. Indeed, from Eq. (5) it is clear that $\mathbf{\Delta}\mathbf{z}^\star = \mathbf{0}$, which implies

$$\mathbf{O}_{ji}\mathbf{z}_i^\star = \mathbf{V}_{ij}\mathbf{z}_j^\star \quad \text{for each } e_{ij} \in \mathcal{E}\,; \tag{18}$$

where $\mathbf{z}_i^\star \in \mathbb{R}^{d_i}$ and $\mathbf{z}_j^\star \in \mathbb{R}^{d_j}$ are the components of $\mathbf{z}^\star$ at node $i$ and $j$, respectively. A plausible strategy would then be to constrain the learned latent representations across the network to be globally consistent with respect to the learned network sheaf $\mathcal{F}$. For simplicity, assume the existence of a set of shared sample indices across agents, and let $M \leq \min_{i\in\mathcal{N}} M_i$ be the number of such comparable samples. Then, according to the previous rationale, we can pose the following centralized problem formulation

$$\begin{aligned} \min_{\substack{\{\boldsymbol{\theta}_i=(\boldsymbol{\varphi}_i,\boldsymbol{\psi}_i)\}_{i\in\mathcal{N}} \\ \{\mathbf{O}_{ji}\in\mathrm{O}(d_i),\,\mathbf{V}_{ij}\in\mathrm{St}(d_i,d_j)\}_{e_{ij}\in\mathcal{E}}}} & \sum_{i\in\mathcal{N}} \mathcal{L}_i(\boldsymbol{\theta}_i) \\ \text{s.t.} \quad & \mathbf{z}^n = \{f_{\boldsymbol{\varphi}_i}(\mathbf{x}_i^n)\}_{i\in\mathcal{N}} \in \ker(\mathbf{L}_\mathcal{F}), \qquad \forall n \in [M]\,; \end{aligned} \tag{19}$$

where the network sheaf $\mathcal{F}$ is learned through its restriction maps under a known communication topology. However, the constraints in Prob. (19) might be too restrictive in practice. Enforcing exact compatibility across heterogeneous agents is generally unrealistic. Moreover, satisfying the constraint for all samples would in principle require a centralized solution, as it couples all node representations simultaneously. This limitation motivates the relaxed, decentralized formulation of Sec. 3.3, of which Prob. (SFRL) is the scalable counterpart.

**On the geometric consistency.** To better understand the action of the restriction maps as well as the role of the total variation penalty, let us consider $\mathcal{F}(i) \rightarrow \mathcal{F}(e_{ij}) \leftarrow \mathcal{F}(j)$ in Fig. 1(c). When transported to the edge stalk $\mathcal{F}(e_{ij}) \cong \mathbb{R}^3$, the latent representation $\mathbf{z}_i \in \mathcal{F}(i)$ is rotated and reflected by the orthogonal map $\mathbf{O}_{ji}$. Instead, the latent representation $\mathbf{z}_j$ is embedded via the Stiefel map $\mathbf{V}_{ij}$ into the higher-dimensional edge stalk. Interestingly, the image $\mathrm{Im}\,(\mathbf{V}_{ij})$ of $\mathbf{V}_{ij}$ defines a two-dimensional subspace embedded in $\mathcal{F}(e_{ij})$. Thus, a perfect alignment between the latent representations $\mathbf{z}_i$ and $\mathbf{z}_j$ implies that $\mathbf{z}_i$ is a rotated and reflected version of a vector $\mathbf{y}_{j\rightarrow e_{ij}} = \mathbf{V}_{ij}\mathbf{z}_j$ belonging to $\mathrm{Im}\,(\mathbf{V}_{ij})$. When this occurs for all the latent representations in $\mathcal{F}(i)$, it means that the latter representations lie on a linear two-dimensional subspace of $\mathcal{F}(i)$.

In Fig. 1(c), we notice that there is only a partial alignment between $\mathbf{z}_i$ and $\mathbf{z}_j$ when they are transported onto $\mathcal{F}(e_{ij})$. Indeed, even though the projection of $\mathbf{O}_{ji}(\mathbf{z}_i)$ onto $\mathrm{Im}\,(\mathbf{V}_{ij})$ exactly matches $\mathbf{V}_{ij}\mathbf{z}_j$, there is still an orthogonal component that makes non-null the Euclidean distance between the two transported vectors. Equivalently, in $\mathcal{F}(i)$ this means that $\mathbf{z}_i$ does not lie on the subspace

$\mathbf{O}_{ji}^\top(\mathrm{Im}\,(\mathbf{V}_{ij}))$, onto which the transported representation of agent $j$ is mapped. Thus, the total variation term acts as a regularizer that, for all $e_{ij} \in \mathcal{E}$, enforces the zeroing of components orthogonal to $\mathrm{Im}\,(\mathbf{V}_{ij})$.

Notably, the latter discussion highlights that within our proposed framework we do not assume or leverage the existence of a *single* global latent representation, as instead pursued in recent works (Setayesh et al., 2026; Badi et al., 2026). Indeed, in our framework a global representation may eventually emerge from the alignment of local representations over the network sheaf.

**Homogeneous setting.** When $\mathcal{F}(i) = \mathcal{F}(e) \cong \mathbb{R}^d$ for all $i \in \mathcal{N}$ and $e \in \mathcal{E}$, the local total variation term in Eq. (7) reduces to

$$\|\mathbf{z}_i - \mathbf{O}_{ij}\mathbf{z}_j\|_2^2\,, \tag{20}$$

where $\mathbf{O}_{ij} \in \mathrm{O}(d)$. In this case, alignment between neighboring representations is achieved up to an orthogonal transformation, i.e., rotations and reflections of the latent space. This setting connects Eq. (6) to the Dirichlet energy of the connection Laplacian (Chung et al., 2014), and more generally to synchronization problems over the orthogonal group (Singer & Wu, 2012). In particular, the transport maps satisfy $\mathbf{O}_{ji} = \mathbf{O}_{ij}^\top$, and the two variation terms in Eq. (8) coincide. In contrast, in the heterogeneous setting we allow semi-orthogonal maps on the Stiefel manifold, which act as isometric embeddings and are only left-invertible. Finally, since all node stalks share the same dimension, the orientation of $\mathcal{G}$ does not affect the model and can be chosen arbitrarily.

## F Pilot Selection Strategies

The alignment of latent spaces relies on a set of *semantic pilots*: inputs whose correspondence across agents is known a priori, so that $f_{\boldsymbol{\varphi}_i}(\mathbf{x}_i^k)$ and $f_{\boldsymbol{\varphi}_j}(\mathbf{x}_j^k)$ can be treated as two latent representations of the same underlying sample. In practice, pilots arise from public datasets, calibration signals, standardized inputs, or an overlapping portion of the local datasets, and we assume that a common index set $\{1, \dots, M\}$ has been agreed upon.

The choice of $\mathcal{A} \subset \{1, \dots, M\}$, with $|\mathcal{A}| = K \ll M$, governs the trade-off between communication cost, computational efficiency, geometric coverage, and alignment quality. In the sequel, we outline the most widely used practical selection criteria. Additionally, a normalization step, applied to the resulting pilot matrices for balancing directions in their corresponding latent spaces, is discussed at the end of the section.

**Original pilots.** Each agent transmits the embeddings $f_{\boldsymbol{\varphi}_i}(\mathbf{x}_i^k)$ of the shared inputs themselves, where $k \in \mathcal{A}$. Taking $\mathcal{A} = \{1, \dots, M\}$ evaluates the penalty exactly and is the most reliable option, but scales linearly in $M$ in both communication and Procrustes cost; subsampling to a budget $K \ll M$ instead yields an unbiased Monte Carlo estimator of the full alignment energy.

*Unsupervised selection:* Pilots are drawn uniformly, which requires no side information but may under-represent low-density regions of the representation space and lead to ill-conditioned $\mathbf{A}_i$. Clustering the representation space and drawing a fixed number of pilots per cluster recovers the same stratification that labels provide in the supervised case, at the cost of one clustering pass, and guarantees that sparsely populated regions are represented.

*Supervised selection:* Labels are exploited since they induce a natural stratification, and a fixed number of pilots is drawn per class, guaranteeing balanced coverage even for small $K$.

**Prototype pilots.** The shared set is partitioned into $K$ groups and each pilot is defined as the group average (centroid),

$$\bar{\mathbf{f}}_{i,c} = \frac{1}{|\mathcal{S}_c|} \sum_{k \in \mathcal{S}_c} f_{\boldsymbol{\varphi}_i}(\mathbf{x}_i^k), \qquad c = 1, \dots, K\,;$$

which preserves semantic structure while averaging out per-sample noise. When groups are large, the centroid need not be computed over all their members: averaging over a subset $\mathcal{S}_c' \subset \mathcal{S}_c$ gives an unbiased estimate at a fraction of the local forward passes, trading estimation variance for computational cost.

*Unsupervised partitioning:* The partition is obtained by clustering the representation space (e.g., via $k$-means), with the resulting index assignment broadcast once and reused thereafter (Hüttebräucker et al., 2024; Fiorellino et al., 2026).

*Supervised partitioning:* Class labels provide the partition directly, recovering prototype-based federated schemes such as FedProto (Tan et al., 2022).

**Ensuring correspondence and adaptive selection.** To ensure that the transmitted quantities are in correspondence, both original and prototype pilots require the selection to be replicated across agents. Specifically: *(i)* the sampled indices in the first case, and *(ii)* the partition and the subsets used to estimate the centroids in the second. Furthermore, the selection can be made adaptive by prioritizing the samples that currently exhibit the largest cross-client mismatch, biasing which inputs are transmitted or which groups are refined so as to concentrate the alignment effort on geometrically inconsistent regions.

**Dynamic selection.** Any of the above criteria can be applied per iteration, letting $\mathcal{A}_t$ vary over time and turning the penalty into a stochastic approximation of the full alignment energy. A practical recipe combines a small persistent subset (for stability of the transport maps) with a periodically refreshed remainder (for coverage), which empirically improves generalization of the learned maps beyond the sampled pilots (Fiorellino et al., 2024).

## G Identifiability of the Transport Maps

The edge-wise transports are computed via (semi-)orthogonal Procrustes problems

$$\mathbf{V}_{ij} \in \arg\min_{\mathbf{V}\in\mathcal{C}_{ij}} \|\mathbf{A}_i - \mathbf{V}\mathbf{A}_j\|_F^2 , \tag{21}$$

where $\mathcal{C}_{ij} = \mathrm{O}(d)$ in the homogeneous case and $\mathcal{C}_{ij} = \mathrm{St}(d_i, d_j)$ for $d_i > d_j$. We summarize conditions under which the minimizer is unique.

**Homogeneous case ($d_i = d_j = d$).** Let $\mathbf{M}_{ij} \coloneqq \mathbf{A}_i\mathbf{A}_j^\top \in \mathbb{R}^{d\times d}$. An optimal solution is $\mathbf{V}_{ij}^\star = \mathbf{U}\mathbf{W}^\top$, where $\mathbf{M}_{ij} = \mathbf{U}\mathbf{\Sigma}\mathbf{W}^\top$ is an SVD (Schönemann, 1966). The solution is unique if $\mathbf{M}_{ij}$ is full rank, i.e., $\mathrm{rank}(\mathbf{M}) = d$. Repeated singular values may make the individual SVD factors $\mathbf{U}$ and $\mathbf{W}$ non-unique, but the product $\mathbf{U}\mathbf{W}^\top$, hence the optimal polar factor, remains unique. In particular, if $\mathrm{rank}(\mathbf{A}_i) = \mathrm{rank}(\mathbf{A}_j) = d$ (e.g., $K \geq d$ with non-degenerate pilots), then $\mathbf{M}_{ij}$ is generically full rank and $\mathbf{V}_{ij}^\star$ is uniquely defined up to measure-zero degeneracies.

**Stiefel case ($d_i > d_j$).** Let $\mathbf{M}_{ij} = \mathbf{A}_i\mathbf{A}_j^\top \in \mathbb{R}^{d_i\times d_j}$ and $\mathbf{M}_{ij} = \mathbf{U}\mathbf{\Sigma}\mathbf{W}^\top$ a thin SVD, with $\mathbf{U} \in \mathbb{R}^{d_i\times d_j}$, $\mathbf{W} \in \mathbb{R}^{d_j\times d_j}$. A minimizer is

$$\mathbf{V}_{ij}^\star = \mathbf{U}\mathbf{W}^\top \in \mathrm{St}(d_i, d_j) . \tag{22}$$

Uniqueness holds if $\mathrm{rank}(\mathbf{M}_{ij}) = d_j$. Crucially, $\mathbf{V}_{ij}$ is identifiable only on the subspace spanned by $\mathbf{A}_j$: if $\mathrm{rank}(\mathbf{A}_j) = r < d_j$, the solution is underdetermined on the orthogonal complement. Hence, identifiability requires $\mathrm{rank}(\mathbf{A}_j) = d_j$ (typically ensured when $K \geq d_j$ and pilots provide sufficient geometric coverage), in which case $\mathbf{V}_{ij}^\star$ is generically unique up to measure-zero degeneracies.

## H Convergence Analysis

In this section we establish convergence guarantees for the proposed Sheaf-FRL algorithm in both deterministic and stochastic settings. The convergence analysis in this appendix applies to the standard-privacy Sheaf-FRL iteration, for which the restriction-map block is updated by the exact closed-form minimizers in (P1a)–(P1b). Alg. 3 uses a strict-privacy parallel update of the coupled restriction maps (P1c)–(P1d). Although each individual Procrustes/SVD subproblem is solved exactly conditional on the exchanged representations, the simultaneous updates do not constitute an exact joint minimization of the full restriction-map block. Therefore, the descent and convergence results below are not claimed for Alg. 3. We first recall the classical descent result for smooth functions, and provide auxiliary derivations.

**Lemma H.1** (Descent Lemma, Nesterov, 2004, Prop. 1.2.3,Beck, 2017, Sec. 2.1)**.** *Let $f$ be differentiable with $L$-Lipschitz gradient. Then for any $\mathbf{x}$ and any $\eta > 0$,*

$$f(\mathbf{x} - \eta \nabla f(\mathbf{x})) \leq f(\mathbf{x}) - \eta\Big(1 - \frac{L\eta}{2}\Big)\|\nabla f(\mathbf{x})\|^2.$$

### H.1 Auxiliary derivations

**Reduction of the alignment subproblems to Procrustes form.** The objective of Prob. (P1a), and analogously of (P1b), can be recast as

$$\left\|\mathbf{A}_i^t - \mathbf{V}_{ij}\mathbf{A}_j^t\right\|_{\mathrm{F}}^2 = \mathrm{tr}\big\{\mathbf{A}_i^t(\mathbf{A}_i^t)^\top\big\} + \mathrm{tr}\big\{\mathbf{A}_j^t(\mathbf{A}_j^t)^\top\big\} - 2\,\mathrm{tr}\big\{\mathbf{A}_i^t(\mathbf{A}_j^t)^\top\mathbf{V}_{ij}^\top\big\}. \tag{23}$$

The first two terms do not depend on $\mathbf{V}_{ij}$. Minimizing Eq. (23) thus corresponds to maximizing the third term, which yields the closed-form solution reported in Eq. (10).

**Edge-wise regularization weights.** For each undirected edge $(i,j) \in \mathcal{E}$, we use a single symmetric coefficient $\lambda_{ij} = \lambda_{ji} > 0$, treated as a fixed edge-wise hyperparameter. Hence, the same coefficient multiplies the two endpoint contributions in Eq. (9). For an edge $e_{ij}$, the two local terms sum to $\frac{\lambda_{ij}}{K}\|\mathbf{A}_i - \mathbf{V}_{ij}\mathbf{A}_j\|_{\mathrm{F}}^2$. Therefore, differentiation with respect to $\boldsymbol{\varphi}_i$ yields the coefficient $\lambda_{ij}/K$, exactly as in Eqs. (11) and (24). Thus, the local neural update uses the exact partial gradient $\nabla_{\boldsymbol{\varphi}_i} J(\boldsymbol{\theta}, \mathbf{V})$ of the global objective, and no information about a neighbor-specific node weight is required.

**Gradient of the gluing penalty.** We report here the full derivation of Eq. (11). Starting from Eq. (9), and recalling that the regularization term depends exclusively on the encoder parameters $\boldsymbol{\varphi}_i$, for each agent $i \in \mathcal{N}$ we obtain

$$\begin{aligned}
\nabla_{\boldsymbol{\varphi}_i}\mathcal{R}_{\mathcal{A}}|_i = \nabla_{\boldsymbol{\varphi}_i}\mathcal{R}_{\mathcal{A}} = &\frac{1}{K}\Bigg[\sum_{j\in\mathcal{N}(i)^-}\lambda_{ij}\sum_{k\in\mathcal{A}}\left(\nabla_{\boldsymbol{\varphi}_i} f_{\boldsymbol{\varphi}_i}(\mathbf{x}_i^k)\right)^\top\left(f_{\boldsymbol{\varphi}_i}\left(\mathbf{x}_i^k\right) - \mathbf{V}_{ij} f_{\boldsymbol{\varphi}_j}\left(\mathbf{x}_j^k\right)\right) \\
&- \sum_{j\in\mathcal{N}(i)^+}\lambda_{ij}\sum_{k\in\mathcal{A}}\left(\nabla_{\boldsymbol{\varphi}_i} f_{\boldsymbol{\varphi}_i}(\mathbf{x}_i^k)\right)^\top\mathbf{V}_{ji}^\top\left(f_{\boldsymbol{\varphi}_j}\left(\mathbf{x}_j^k\right) - \mathbf{V}_{ji} f_{\boldsymbol{\varphi}_i}\left(\mathbf{x}_i^k\right)\right)\Bigg] \\
&\overset{(a)}{=} \frac{1}{K}\sum_{j\in\mathcal{N}(i)}\lambda_{ij}\sum_{k\in\mathcal{A}}\left(\nabla_{\boldsymbol{\varphi}_i} f_{\boldsymbol{\varphi}_i}(\mathbf{x}_i^k)\right)^\top\left(f_{\boldsymbol{\varphi}_i}\left(\mathbf{x}_i^k\right) - \mathbf{V}_{ij} f_{\boldsymbol{\varphi}_j}\left(\mathbf{x}_j^k\right)\right);
\end{aligned} \tag{24}$$

where $\nabla_{\boldsymbol{\varphi}_i} f_{\boldsymbol{\varphi}_i}(\mathbf{x}_i^k) \in \mathbb{R}^{d_i \times |\boldsymbol{\varphi}_i|}$ denotes the Jacobian of the feature map $f_{\boldsymbol{\varphi}_i}$ with respect to the parameter vector $\boldsymbol{\varphi}_i$, evaluated at $\mathbf{x}_i^k$. In $(a)$ we adopt, for $j \in \mathcal{N}(i)^+$, the convention $\mathbf{V}_{ij} \coloneqq \mathbf{V}_{ji}^\top$, which satisfies $\mathbf{V}_{ij}\mathbf{V}_{ji} = \mathbf{I}_{d_i}$ by orthonormality of the columns of $\mathbf{V}_{ji} \in \mathrm{St}(d_j, d_i)$. Note that, under this convention, $\mathbf{V}_{ij}$ has orthonormal rows rather than columns on outgoing edges. This shows that both incoming and outgoing contributions admit a unified expression, which is the one reported in Eq. (11) of the main text.

### H.2 Deterministic Setting

Let

$$J(\boldsymbol{\theta}, \mathbf{V}) = \sum_{i\in\mathcal{N}}\mathcal{L}_i(\boldsymbol{\theta}_i) + \mathcal{R}_{\mathcal{A}}\big(\{\boldsymbol{\varphi}_i\}_{i\in\mathcal{N}}, \{\mathbf{V}_{ij}\}_{(i,j)\in\mathcal{E}}, \{\mathbf{V}_{ji}\}_{(i,j)\in\mathcal{E}}\big) \tag{25}$$

in (SFRL) where, with slight abuse of notation, $\boldsymbol{\theta} = \{\boldsymbol{\theta}_i\}_{i\in\mathcal{N}} = \{(\boldsymbol{\varphi}_i, \boldsymbol{\psi}_i)\}_{i\in\mathcal{N}}$, and $\mathbf{V} = \{\mathbf{V}_{ij}\}_{(i,j)\in\mathcal{E}} \cup \{\mathbf{V}_{ji}\}_{(i,j)\in\mathcal{E}}$. For each edge $(i,j) \in \mathcal{E}$, the matrices $\mathbf{V}_{ij} \in \mathrm{St}(d_i, d_j)$ and $\mathbf{V}_{ji} \in \mathrm{St}(d_j, d_i)$, where $\mathrm{St}(d_i, d_j)$ denotes a Stiefel manifold as defined in Eq. (3). The feasible set is therefore $\mathcal{V} \coloneqq \Big(\prod_{(i,j)\in\mathcal{E}}\mathrm{St}(d_i, d_j)\Big) \times \Big(\prod_{(i,j)\in\mathcal{E}}\mathrm{St}(d_j, d_i)\Big)$. The proposed method alternates between an exact minimization with respect to $\mathbf{V}$ and a gradient step with respect to $\boldsymbol{\theta}$, and can be compactly cast as:

$$\mathbf{V}^t \in \arg\min_{\mathbf{V}\in\mathcal{V}} J(\boldsymbol{\theta}^t, \mathbf{V}), \tag{26}$$

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^t, \mathbf{V}^t) \tag{27}$$

We introduce the following assumptions.

**Assumption 1.** *The objective $J(\boldsymbol{\theta}, \mathbf{V})$ is bounded from below, with lower bound*

$$J_{\inf} \coloneqq \inf_{\boldsymbol{\theta}, \mathbf{V}} J(\boldsymbol{\theta}, \mathbf{V}) ;$$

*and its sublevel sets are bounded.*

The second condition holds whenever the local objectives include a weight-decay term, as in our experiments (App. I).

**Assumption 2.** *For every fixed $\mathbf{V} \in \mathcal{V}$, $J(\boldsymbol{\theta}, \mathbf{V})$ is differentiable and its gradient is $L$-Lipschitz:*

$$\|\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_1, \mathbf{V}) - \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_2, \mathbf{V})\| \leq L\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|.$$

**Assumption 3.** *At each iteration,*

$$\mathbf{V}^t \in \arg\min_{\mathbf{V} \in \mathcal{V}} J(\boldsymbol{\theta}^t, \mathbf{V}),$$

*i.e., $\mathbf{V}^t$ is an exact minimizer of $J$ given $\boldsymbol{\theta}^t$.*

**Assumption 4.** *The feasible set $\mathcal{V}$ is compact.*

Under 1-4, the following convergence result holds.

**Theorem H.2.** *Suppose Assumptions 1–4 hold and choose $0 < \eta \leq 1/L$. Then:*

1. ***Descent, Convergence, and Stationarity.*** *The sequence $\{J(\boldsymbol{\theta}^t, \mathbf{V}^t)\}$ is monotonically decreasing and therefore convergent. Moreover, $\lim_{t\to\infty} \|\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^t, \mathbf{V}^t)\| = 0$, and every accumulation point $(\boldsymbol{\theta}^\star, \mathbf{V}^\star)$ is a block-wise first-order stationary point.*
2. ***Sublinear stationarity rate.*** *For all $T \geq 0$,*

$$\min_{0 \leq t \leq T} \|\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^t, \mathbf{V}^t)\|^2 \leq \frac{2\big(J(\boldsymbol{\theta}^0, \mathbf{V}^0) - J_{\inf}\big)}{\eta(T+1)} . \tag{28}$$

*Proof.* **(i) Descent, Convergence, and Stationarity**. Fix $\mathbf{V}^t$ and consider $J(\boldsymbol{\theta}, \mathbf{V}^t)$. By Assumption 2, $J(\boldsymbol{\theta}, \mathbf{V}^t)$ has $L$-Lipschitz gradient. Applying Lemma H.1 with $\eta \leq 1/L$ yields

$$J(\boldsymbol{\theta}^{t+1}, \mathbf{V}^t) \leq J(\boldsymbol{\theta}^t, \mathbf{V}^t) - \frac{\eta}{2}\|\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^t, \mathbf{V}^t)\|^2. \tag{29}$$

By exact minimization of $\mathbf{V}$ at iteration $t+1$ (Assumption 3), $J(\boldsymbol{\theta}^{t+1}, \mathbf{V}^{t+1}) \leq J(\boldsymbol{\theta}^{t+1}, \mathbf{V}^t)$. Combining this inequality with the previous bound yields $J(\boldsymbol{\theta}^{t+1}, \mathbf{V}^{t+1}) \leq J(\boldsymbol{\theta}^t, \mathbf{V}^t)$, which shows that the sequence $\{J(\boldsymbol{\theta}^t, \mathbf{V}^t)\}$ is monotonically decreasing. Since $J$ is bounded below (Assumption 1), the sequence therefore converges.

Summing (29) over $t$ yields $\sum_{t=0}^{\infty} \|\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^t, \mathbf{V}^t)\|^2 < \infty$, which implies

$$\lim_{t\to\infty} \|\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^t, \mathbf{V}^t)\| = 0.$$

Since $\{J(\boldsymbol{\theta}^t, \mathbf{V}^t)\}$ is non-increasing, the iterates remain in a bounded sublevel set (Assumption 1), so the sequence admits a convergent subsequence. Let $(\boldsymbol{\theta}^{t_k}, \mathbf{V}^{t_k})$ be a convergent subsequence. By continuity of $\nabla_{\boldsymbol{\theta}} J$, it holds $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^\star, \mathbf{V}^\star) = 0$. Moreover, since $\mathbf{V}^t$ minimizes $J(\boldsymbol{\theta}^t, \cdot)$ over the compact set $\mathcal{V}$, passing to the limit gives

$$\mathbf{V}^\star \in \arg\min_{\mathbf{V} \in \mathcal{V}} J(\boldsymbol{\theta}^\star, \mathbf{V}).$$

Thus every accumulation point is block-wise first-order stationary.

**(ii) Sublinear Stationarity Rate.** From the descent inequality (29), summing from $t = 0$ to $T$ and telescoping yields

$$\frac{\eta}{2} \sum_{t=0}^{T} \|\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^t, \mathbf{V}^t)\|^2 \leq J(\boldsymbol{\theta}^0, \mathbf{V}^0) - J(\boldsymbol{\theta}^{T+1}, \mathbf{V}^T).$$

Since $J(\boldsymbol{\theta}^{T+1}, \mathbf{V}^T) \geq J_{\inf}$ (Assumption 1),

$$\sum_{t=0}^{T} \|\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^t, \mathbf{V}^t)\|^2 \leq \frac{2\big(J(\boldsymbol{\theta}^0, \mathbf{V}^0) - J_{\inf}\big)}{\eta}.$$

Finally, using $\min_{0\leq t\leq T} a_t \leq \frac{1}{T+1}\sum_{t=0}^{T} a_t$ with $a_t = \|\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^t, \mathbf{V}^t)\|^2$, we obtain

$$\min_{0\leq t\leq T} \|\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^t, \mathbf{V}^t)\|^2 \leq \frac{2\big(J(\boldsymbol{\theta}^0, \mathbf{V}^0) - J_{\inf}\big)}{\eta(T+1)},$$

which proves (28). □

### H.3 Stochastic Setting

We consider the same deterministic objective $J(\boldsymbol{\theta}, \mathbf{V})$ in (25), and extend the $\boldsymbol{\theta}$-update to the stochastic (mini-batch) setting while keeping the $\mathbf{V}$-block *exactly* minimized with respect to the full objective. Let $\{\Phi_t\}_{t\geq 0}$ be the natural filtration generated by the iterates and the sampling randomness up to time $t$. At iteration $t$, we first update the alignment variables by exact minimization,

$$\mathbf{V}^t \in \arg\min_{\mathbf{V}\in\mathcal{V}} J(\boldsymbol{\theta}^t, \mathbf{V}), \tag{30}$$

and then form a mini-batch stochastic gradient estimator $\mathbf{g}^t \equiv \mathbf{g}(\boldsymbol{\theta}^t, \mathbf{V}^t; \xi_t)$, where $\xi_t$ denotes the mini-batch sampling randomness. The stochastic update of the neural parameters is

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta_t\, \mathbf{g}^t. \tag{31}$$

To analyze the convergence behavior of the stochastic scheme, we impose the following assumptions on the stochastic gradient estimator.

**Assumption 5** (Unbiased stochastic gradient)**.** *For all $t \geq 0$,*

$$\mathbb{E}[\mathbf{g}^t \mid \Phi_t] = \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^t, \mathbf{V}^t).$$

**Assumption 6** (Bounded conditional variance)**.** *There exists $\sigma^2 < \infty$ such that for all $t \geq 0$,*

$$\mathbb{E}\Big[\big\|\mathbf{g}^t - \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^t, \mathbf{V}^t)\big\|^2 \;\Big|\; \Phi_t\Big] \leq \sigma^2.$$

Under these assumptions, we establish convergence guarantees for the stochastic variant of the proposed method in the following sections.

#### H.3.1 Convergence Guarantee with Optimized Step Size

**Theorem H.3** (Stochastic convergence rate)**.** *Assume Assumptions 1 to 4, and the stochastic gradient conditions in Assumptions 5 and 6. Let $\{(\boldsymbol{\theta}^t, \mathbf{V}^t)\}_{t\geq 0}$ be generated by* (30)–(31)*. Define $\Delta_0 := J(\boldsymbol{\theta}^0, \mathbf{V}^0) - J_{\inf}$, where $J_{\inf} := \inf_{\boldsymbol{\theta},\mathbf{V}} J(\boldsymbol{\theta}, \mathbf{V})$. Then, for any horizon $T \geq 0$, choosing the constant step size*

$$\eta_t \equiv \eta := \min\left\{\frac{1}{L}, \sqrt{\frac{2\Delta_0}{L\sigma^2(T+1)}}\right\}$$

*yields the bound*

$$\min_{0\leq t\leq T} \mathbb{E}\Big[\big\|\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^t, \mathbf{V}^t)\big\|^2\Big] \leq 2\sqrt{\frac{2L\sigma^2\,\Delta_0}{T+1}} \;+\; \frac{2L\Delta_0}{T+1}.$$

*In particular, the method attains the stochastic convergence rate*

$$\min_{0\leq t\leq T} \mathbb{E}\Big[\big\|\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^t, \mathbf{V}^t)\big\|^2\Big] = \mathcal{O}\left(\frac{1}{\sqrt{T}}\right).$$

*Proof.* For brevity, define

$$J_t := J(\boldsymbol{\theta}^t, \mathbf{V}^t), \qquad \nabla_t := \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^t, \mathbf{V}^t), \qquad \mathbf{g}^t := \mathbf{g}(\boldsymbol{\theta}^t, \mathbf{V}^t; \xi_t).$$

Fix $t \geq 0$. By Assumption 2, for the $L$-smooth map $\boldsymbol{\theta} \mapsto J(\boldsymbol{\theta}, \mathbf{V}^t)$, we have for any vector $\mathbf{u}$,

$$J(\boldsymbol{\theta}^t + \mathbf{u}, \mathbf{V}^t) \leq J(\boldsymbol{\theta}^t, \mathbf{V}^t) + \nabla_t^\top \mathbf{u} + \frac{L}{2}\|\mathbf{u}\|^2.$$

Applying this with $\mathbf{u} = -\eta\, \mathbf{g}^t$ yields

$$J(\boldsymbol{\theta}^{t+1}, \mathbf{V}^t) \leq J(\boldsymbol{\theta}^t, \mathbf{V}^t) - \eta\, \nabla_t^\top \mathbf{g}^t + \frac{L\eta^2}{2}\|\mathbf{g}^t\|^2. \tag{32}$$

By exact minimization in $\mathbf{V}$ (Assumption 3),

$$J(\boldsymbol{\theta}^{t+1}, \mathbf{V}^{t+1}) \leq J(\boldsymbol{\theta}^{t+1}, \mathbf{V}^t). \tag{33}$$

Combining (32)–(33) gives

$$J_{t+1} \leq J_t - \eta\, \nabla_t^\top \mathbf{g}^t + \frac{L\eta^2}{2}\|\mathbf{g}^t\|^2. \tag{34}$$

Taking conditional expectation w.r.t. $\Phi_t$ and using unbiasedness (Assumption 5) gives

$$\mathbb{E}\big[\nabla_t^\top \mathbf{g}^t \mid \Phi_t\big] = \nabla_t^\top \mathbb{E}[\mathbf{g}^t \mid \Phi_t] = \|\nabla_t\|^2.$$

Moreover, expanding the second moment and using the bounded conditional variance (Assumption 6) yields

$$\begin{aligned}
\mathbb{E}\big[\|\mathbf{g}^t\|^2 \mid \Phi_t\big] &= \mathbb{E}\big[\|\mathbf{g}^t - \nabla_t + \nabla_t\|^2 \mid \Phi_t\big] \\
&= \|\nabla_t\|^2 + \mathbb{E}\big[\|\mathbf{g}^t - \nabla_t\|^2 \mid \Phi_t\big] + 2\,\nabla_t^\top \mathbb{E}[\mathbf{g}^t - \nabla_t \mid \Phi_t] \\
&\leq \|\nabla_t\|^2 + \sigma^2,
\end{aligned}$$

since $\mathbb{E}[\mathbf{g}^t - \nabla_t \mid \Phi_t] = \mathbf{0}$ by Assumption 5. Substituting these two identities into the conditional expectation of (34) yields

$$\mathbb{E}[J_{t+1} \mid \Phi_t] \leq J_t - \eta\Big(1 - \frac{L\eta}{2}\Big)\|\nabla_t\|^2 + \frac{L\eta^2}{2}\sigma^2. \tag{35}$$

Assuming $0 < \eta \leq 1/L$ implies $1 - \frac{L\eta}{2} \geq \frac{1}{2}$, hence

$$\mathbb{E}[J_{t+1} \mid \Phi_t] \leq J_t - \frac{\eta}{2}\|\nabla_t\|^2 + \frac{L\eta^2}{2}\sigma^2. \tag{36}$$

Taking total expectation in (36) and summing from $t = 0$ to $T$ gives

$$\mathbb{E}[J_{T+1}] \leq J_0 - \frac{\eta}{2}\sum_{t=0}^{T}\mathbb{E}\|\nabla_t\|^2 + \frac{L\eta^2}{2}\sigma^2(T+1).$$

Rearranging and using the lower bound $\mathbb{E}[J_{T+1}] \geq J_{\inf}$ (Assumption 1) yields

$$\frac{\eta}{2}\sum_{t=0}^{T}\mathbb{E}\|\nabla_t\|^2 \leq \Delta_0 + \frac{L\eta^2}{2}\sigma^2(T+1), \qquad \Delta_0 := J_0 - J_{\inf}. \tag{37}$$

Dividing by $\eta(T+1)/2$ and using $\min_{0\leq t\leq T} a_t \leq \frac{1}{T+1}\sum_{t=0}^{T} a_t$ gives

$$\min_{0\leq t\leq T}\mathbb{E}\|\nabla_t\|^2 \leq \frac{2\Delta_0}{\eta(T+1)} + L\eta\,\sigma^2. \tag{38}$$

We now optimize the right-hand side in $\eta$ under the constraint $\eta \leq 1/L$. Let

$$\eta^\star := \sqrt{\frac{2\Delta_0}{L\sigma^2(T+1)}}.$$

If $\eta^\star \le 1/L$, choosing $\eta = \eta^\star$ in (38) yields

$$\min_{0\le t\le T} \mathbb{E}\|\nabla_t\|^2 \le 2\sqrt{\frac{2L\sigma^2\Delta_0}{T+1}}.$$

If instead $\eta^\star > 1/L$, we choose $\eta = 1/L$ in (38) to obtain

$$\min_{0\le t\le T} \mathbb{E}\|\nabla_t\|^2 \le \frac{2L\Delta_0}{T+1} + \sigma^2.$$

Both cases are covered by the choice

$$\eta = \min\left\{\frac{1}{L},\ \sqrt{\frac{2\Delta_0}{L\sigma^2(T+1)}}\right\}.$$

Moreover, with this choice we have the unified bound

$$\min_{0\le t\le T} \mathbb{E}\|\nabla_t\|^2 \le 2\sqrt{\frac{2L\sigma^2\Delta_0}{T+1}} + \frac{2L\Delta_0}{T+1},$$

where the second term is redundant when $\eta = \eta^\star$ but remains valid. This proves Theorem H.3. □

### H.3.2 Almost-sure Convergence with Diminishing Step Sizes

We establish an almost-sure convergence guarantee under a Robbins–Monro step-size schedule. Throughout, let $\{\Phi_t\}_{t\ge0}$ denote the natural filtration and let the iterates be generated by (30)–(31).

**Assumption 7** (Robbins–Monro step sizes)**.** *The step sizes satisfy $0 < \eta_t \le 1/L$ for all $t \ge 0$, and*

$$\sum_{t=0}^{\infty} \eta_t = \infty, \qquad \sum_{t=0}^{\infty} \eta_t^2 < \infty.$$

**Theorem H.4** (Almost-sure convergence)**.** *Assume Assumptions 1 to 4, and Assumptions 5 and 6 and Assumption 7. Let $\{(\boldsymbol{\theta}^t, \mathbf{V}^t)\}_{t\ge0}$ be generated by* (30)–(31)*. Then:*

1. *The sequence $\{J(\boldsymbol{\theta}^t, \mathbf{V}^t)\}$ converges almost surely to a finite random variable.*
2. *Moreover,*
$$\sum_{t=0}^{\infty} \eta_t \left\|\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^t, \mathbf{V}^t)\right\|^2 < \infty \qquad \textit{almost surely}.$$
3. *Consequently,*
$$\liminf_{t\to\infty} \left\|\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^t, \mathbf{V}^t)\right\| = 0 \qquad \textit{almost surely}.$$

*Proof.* We begin by recalling the Robbins–Siegmund lemma, which provides a standard convergence result for quasi-supermartingale sequences.

**Lemma H.5** (Robbins–Siegmund, Robbins & Siegmund, 1971)**.** *Let $\{X_t\}_{t\ge0}$ be a nonnegative adapted sequence with respect to a filtration $\{\Phi_t\}_{t\ge0}$. Assume there exist nonnegative $\Phi_t$-measurable sequences $\{a_t\}_{t\ge0}$ and $\{b_t\}_{t\ge0}$ such that*

$$\mathbb{E}[X_{t+1} \mid \Phi_t] \le X_t - a_t + b_t \qquad \textit{a.s. for all } t,$$

*and $\sum_{t=0}^{\infty} b_t < \infty$ almost surely. Then $X_t$ converges almost surely to a finite random variable and $\sum_{t=0}^{\infty} a_t < \infty$ almost surely.*

We now show that the sequence generated by the algorithm satisfies the conditions of this lemma. Define

$$J_t := J(\boldsymbol{\theta}^t, \mathbf{V}^t), \qquad \nabla_t := \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^t, \mathbf{V}^t), \qquad X_t := J_t - J_{\inf} \ge 0.$$

From the one-step bound (cf. (36)), for all $t \ge 0$,

$$\mathbb{E}[J_{t+1} \mid \Phi_t] \le J_t - \frac{\eta_t}{2}\|\nabla_t\|^2 + \frac{L\eta_t^2}{2}\sigma^2.$$

Subtracting $J_{\inf}$ from both sides yields

$$\mathbb{E}[X_{t+1} \mid \Phi_t] \le X_t - a_t + b_t, \qquad a_t := \frac{\eta_t}{2}\|\nabla_t\|^2, \quad b_t := \frac{L\eta_t^2}{2}\sigma^2.$$

By Assumption 7 we have $\sum_{t=0}^{\infty} \eta_t^2 < \infty$, hence $\sum_{t=0}^{\infty} b_t < \infty$. Therefore all the conditions of Lemma H.5 are satisfied. Applying the lemma implies that $X_t$ converges almost surely to a finite random variable, and

$$\sum_{t=0}^{\infty} \eta_t \|\nabla_t\|^2 < \infty \qquad \text{a.s.}$$

Finally, since $\sum_{t=0}^{\infty} \eta_t = \infty$ (Assumption 7), the above summability implies

$$\liminf_{t\to\infty} \|\nabla_t\| = 0 \qquad \text{a.s.}$$

This proves Theorem H.4. □

# I Experimental details

## I.1 Supervised classification

All experiments run on MNIST with the label-shift-induced data heterogeneity of Eq. (13). Every agent trains with stochastic gradient descent (learning rate $10^{-2}$, momentum $0.9$, weight decay $5 \cdot 10^{-4}$) and batch size $64$; the fifteen-agent runs additionally clip gradient norms at $1.0$. A shared pilot set, amounting to $10\%$ of MNIST and held out from every agent's local pool, provides the pilots exchanged during sheaf-based training and the data on which the post-hoc alignment maps of the baselines are fitted. Holding it out means that no sample is seen by more than one agent during training, so the encoders remain fully independent and the heterogeneity of the latent spaces is not diluted by shared data. Since the shift strength satisfies $s \in [0, 1)$, every class retains nonzero probability at every agent, so the pilot set is a uniform subsample and no encoder is asked to embed samples from a class it has not seen. Reported accuracies are computed on the private test sets, disjoint from both training and pilot data. Samples of each class are split across agents in proportion to $\{P_i(y)\}_{i=1}^{N}$, and each agent's local pool is then split $80\%/10\%/10\%$ into train/validation/test, preserving its label skew across all three splits. The whitening statistics are estimated on the pilot representations alone, and the coloring layer is exercised only at communication time; along the local path the two layers cancel and are therefore not computed.

**Evaluation metrics.** Denote by $\mathcal{T}_i^{\text{test}}$ the test set for agent $i$ consisting of $M_i^{\text{test}}$ pairs $\{(\mathbf{x}_i^{\text{test},n}, y_i^{\text{test},n})\}_{n\in M_i^{\text{test}}}$. Then,

$$\begin{aligned} A_i^{\text{priv}} &= \frac{1}{M_i^{\text{test}}} \sum_{n=1}^{M_i^{\text{test}}} \mathbf{1}[\arg\max_{c\in\mathcal{Y}} [h_{\boldsymbol{\psi}_i} \circ f_{\boldsymbol{\varphi}_i}(\mathbf{x}_i^{\text{test},n})]_c = y_i^{\text{test},n}] \\ A_i^{\text{comm}} &= \frac{1}{|\mathcal{N}(i)|} \Bigg( \sum_{j\in\mathcal{N}(i)^-} \frac{1}{M_j^{\text{test}}} \sum_{n=1}^{M_j^{\text{test}}} \mathbf{1}[\arg\max_{c\in\mathcal{Y}} [h_{\boldsymbol{\psi}_i} \circ \mathbf{V}_{ij} \circ f_{\boldsymbol{\varphi}_j}(\mathbf{x}_j^{\text{test},n})]_c = y_j^{\text{test},n}] + \\ &\qquad \sum_{j\in\mathcal{N}(i)^+} \frac{1}{M_j^{\text{test}}} \sum_{n=1}^{M_j^{\text{test}}} \mathbf{1}[\arg\max_{c\in\mathcal{Y}} [h_{\boldsymbol{\psi}_i} \circ \mathbf{V}_{ji}^{\top} \circ f_{\boldsymbol{\varphi}_j}(\mathbf{x}_j^{\text{test},n})]_c = y_j^{\text{test},n}] \Bigg). \end{aligned} \tag{39}$$

**Two-agent heterogeneous pair (Fig. 2(c)–(d)).** The two agents are drawn from a parametric CNN family indexed by the bottleneck dimension $d \in \{16, 32, 64, 128, 256, 512\}$, matched so that the last block of both encoders outputs the same $d$ (Tab. 3). Each convolutional block consists of a convolution, batch normalization, ReLU, and max-pooling; the final feature map is globally pooled, so the latent dimension equals the width of the last block. Agent $0$ uses a dropout of $0.3$ while agent $1$ uses a dropout of $0.1$. Agent $0$'s target classes are $\{4, \dots, 9\}$ and agent $1$'s are $\{0, \dots, 5\}$,

Table 2: The fifteen heterogeneous CNN agents of the multi-agent benchmark (Sec. 5). Encoder: output channels of each convolutional block; head: hidden widths of the MLP classifier; $d_i$: latent (stalk) dimension. Total: 3.41M parameters.

| Agent | Encoder widths | Head widths | Dropout | $d_i$ | Target classes $\mathcal{C}_i$ |
|---|---|---|---|---|---|
| 0 | $(32, 64, 128)$ | $(256, 128, 64)$ | 0.30 | 128 | $\{4, 5, 6, 7, 8\}$ |
| 1 | $(32, 64, 128, 224)$ | $(120)$ | 0.10 | 224 | $\{0, 1, 2, 3, 4\}$ |
| 2 | $(16, 32, 64)$ | $(128)$ | 0.20 | 64 | $\{0, 1, 2, 3, 4, 5\}$ |
| 3 | $(64, 128, 128)$ | $(256, 128)$ | 0.25 | 128 | $\{4, 5, 6, 7, 8, 9\}$ |
| 4 | $(32, 48, 96, 192)$ | $(192, 96)$ | 0.30 | 192 | $\{0, 1, 2, 7, 8, 9\}$ |
| 5 | $(24, 48, 96)$ | $(96)$ | 0.15 | 96 | $\{2, 3, 4, 5, 6\}$ |
| 6 | $(32, 64, 128, 256)$ | $(256)$ | 0.40 | 256 | $\{1, 2, 3, 4, 5, 6\}$ |
| 7 | $(24, 48, 96, 144)$ | $(64, 32)$ | 0.10 | 144 | $\{3, 4, 5, 6, 7\}$ |
| 8 | $(48, 96, 160)$ | $(160, 80)$ | 0.20 | 160 | $\{0, 2, 4, 6, 8\}$ |
| 9 | $(32, 80)$ | $(100)$ | 0.05 | 80 | $\{5, 6, 7, 8, 9\}$ |
| 10 | $(40, 80, 120)$ | $(120, 60)$ | 0.20 | 120 | $\{1, 2, 3, 4, 5\}$ |
| 11 | $(48, 96, 192, 240)$ | $(256, 128)$ | 0.20 | 240 | $\{0, 1, 2, 8, 9\}$ |
| 12 | $(20, 40, 60, 80)$ | $(80)$ | 0.20 | 80 | $\{3, 4, 5, 6, 7, 8\}$ |
| 13 | $(32, 64, 96)$ | $(192, 96)$ | 0.25 | 96 | $\{1, 4, 6, 8, 9\}$ |
| 14 | $(56, 112, 160)$ | $(128)$ | 0.10 | 160 | $\{0, 1, 2, 3, 7\}$ |

at a fixed shift strength $s = 0.7$; the graph has two nodes corresponding to the agents and a single edge connecting them. Training lasts 20 epochs with the common optimizer above.

**Fifteen-agent network (Fig. 2(a)–(b)).** Tab. 2 lists the fifteen architectures and the target classes assigned to each agent. All encoders use batch normalization; the latent (stalk) dimension $d_i$ equals the width of the last convolutional block. The proposed methods run for a total of 100 epochs.

**Network topology.** Each agent $i \in [N]$ is assigned to a distinct node, so that $|\mathcal{N}| = N$. The edge set $\mathcal{E}$ is constructed from the target-class sets $\{\mathcal{C}_i\}_{i\in[N]}$ as follows. For every pair $(i, j)$, let $w_{ij} \coloneqq |\mathcal{C}_i \cap \mathcal{C}_j|$ denote the number of target classes shared by agents $i$ and $j$, collected into the weight vector $\mathbf{w}$. These weights define the fully-connected, weighted graph $\widetilde{\mathcal{G}} \coloneqq (\mathcal{N}, \widetilde{\mathcal{E}}, \mathbf{w})$, with $\widetilde{\mathcal{E}} \coloneqq \{(i, j) \in [N] \times [N] \mid i \neq j\}$. We then extract the maximum-weight spanning tree $\hat{\mathcal{G}} \coloneqq (\mathcal{N}, \hat{\mathcal{E}}, \hat{\mathbf{w}})$ from $\widetilde{\mathcal{G}}$, and initialize $\mathcal{E}$ with the (unweighted) edges of $\hat{\mathcal{E}}$, which guarantees connectivity. We next add edges from $\widetilde{\mathcal{E}} \setminus \hat{\mathcal{E}}$, in decreasing order of weight, until the target $40\%$ density is reached. Finally, each edge is assigned an orientation, consistent with the orientation rule in Sec. 3.3.

**Post-hoc alignment of the baselines.** In the fifteen-agent benchmark, Sheaf-FMTL and the non-cooperative baseline require post-hoc alignment to enable semantic communication; in the two-agent sweep FedProto and FedMuscle require it as well. Specifically, for each edge $(i, j)$ we solve (P1a) and (P1b) in closed form via Eq. (10), fitting isometric maps between whitened latent spaces using the same pilot sets as in Sheaf-FRL. In our experiments, the whitening for these baselines is pursued through classical zero-phase component analysis (Kessy et al., 2018) fitted on the latent representations from the private training set, independently for each agent.

**Selection of the regularization coefficient.** For Sheaf-FRL, as well as Sheaf-FMTL, ComFed, FedProto, and FedMuscle baselines, we need to find the best regularization coefficient. The applied strategy in each architecture configuration under consideration (either the bottleneck dimension in the two-agent sweep, or the distribution-shift strength in the fifteen-agent benchmark) is a grid search for the best value in the $[10^{-5}, 1]$ interval.

**Method-specific hyperparameters.** Beyond the regularization coefficient, every baseline also carries its own additional, fixed hyperparameters that are untouched by the grid-search above. Sheaf-FMTL introduces two further constants, the local-regularization weight $\gamma = 10^{-3}$ and the personalization rate $\eta = 10^{-2}$, both held fixed across every shift strength and every bottleneck dimension. For further information to support the selection process of the Sheaf-FMTL hyperparameter $\gamma$ refer to the paragraph "Memory footprint and choice of $\gamma$ for Sheaf-FMTL" below. ComFed instead learns, for every agent $i$, an unconstrained linear projection matrix $\mathbf{P}_i \in \mathbb{R}^{r\times d_i}$ mapping that agent's local $d_i$-dimensional latent space (the right, input-side dimension of $\mathbf{P}_i$) into a shared space of di-

Table 3: The shared two-agent architecture family used at every bottleneck dimension $d$ in Fig. 2(c)–(d); both agents' encoders are matched to end at width $d$. Encoder/head: output channels of each convolutional block / hidden widths of the MLP classifier. Dropout is $0.3$ for agent 0 and $0.1$ for agent 1.

| $d$ | Agent | Encoder widths | Head widths |
|---|---|---|---|
| 16 | 0 | $(4, 8, 16)$ | $(8, 4)$ |
| | 1 | $(8, 16)$ | $(8)$ |
| 32 | 0 | $(8, 16, 32)$ | $(16, 8)$ |
| | 1 | $(16, 32)$ | $(16)$ |
| 64 | 0 | $(16, 32, 64)$ | $(32, 16)$ |
| | 1 | $(32, 64)$ | $(32)$ |
| 128 | 0 | $(32, 64, 128)$ | $(64, 32)$ |
| | 1 | $(64, 128)$ | $(64)$ |
| 256 | 0 | $(64, 128, 256)$ | $(128, 64)$ |
| | 1 | $(128, 256)$ | $(128)$ |
| 512 | 0 | $(64, 128, 256, 512)$ | $(128, 64)$ |
| | 1 | $(64, 256, 512)$ | $(128, 64)$ |

mension $r$, the left, output-side dimension common to every agent, in which class prototypes from all agents are aligned. We fix $r = 64$ in the fifteen-agent benchmark. In the two-agent bottleneck sweep, where every configuration already forces both agents to share a common latent dimension $d$, we instead set the projection dimension equal to $d$ at each bottleneck value, so that ComFed's shared space coincides with the bottleneck dimension already under study, rather than further compressing or expanding it. FedProto and FedMuscle assume a common latent dimensionality, which does not hold under model heterogeneity. In the fifteen-agent experiment, we therefore equip each agent with the same construction used for ComFed: a learnable linear projection $\mathbf{P}_i \in \mathbb{R}^{r \times d_i}$ onto a shared space of dimension $r$, applied before the respective regularizer, with $r = 64$. In that experiment they therefore learn their alignment maps during training and require no post-hoc step, whereas in the two-agent sweep, where both agents already share the latent dimension $d$, no projection is added and they are aligned post hoc like Sheaf-FMTL and the non-cooperative baseline. FedProto maintains a single global prototype per class, updated as an exponential moving average of the agents' local per-class embeddings with momentum $0.9$. FedMuscle organizes training into repeating communication rounds, each consisting of four local epochs of ordinary task-loss training followed by one additional epoch in which a contrastive alignment loss, with temperature $0.1$, pulls every agent's representations toward the network average.

**What distinguishes the shared-space baselines.** Across the two experiments, the methods that learn an unconstrained linear projection onto a shared space are exactly the ones that lose communication accuracy, and this does not reduce to compression. In the fifteen-agent experiment ComFed, FedProto and FedMuscle all carry such a projection and all collapse, while the non-cooperative baseline and Sheaf-FMTL, aligned post hoc by an orthogonal Procrustes map, do not. In the two-agent sweep FedProto and FedMuscle apply natively and are aligned post hoc, and they recover; ComFed retains its projection with $r = d$, so that no compression is introduced, and still fails at every bottleneck dimension. These observations indicate that compression alone does not account for the behaviour of the shared-space baselines, and are consistent with the structure of the map being the relevant factor: an unconstrained map optimized jointly with the task need not preserve the latent geometry a receiving head relies on, whereas an isometry does so by construction. Sheaf-FRL learns its maps during training, as the former group does, but constrains them to be isometries, as the latter group's post-hoc Procrustes step implicitly does. The methods compared here differ in several respects besides their alignment map, so we regard this as suggestive rather than conclusive; a direct ablation replacing the Stiefel constraint with an unconstrained linear map is left to future work.

**Memory footprint and choice of $\gamma$ for Sheaf-FMTL.** Let the number of parameters of agents $i$ and $j$ be respectively $d_i$ and $d_j$, then the restriction maps matrix of Sheaf-FMTL $\mathbf{P}_{ij}$ acts directly in parameter space, and its size is $d_{ij} \times d_i$ with $d_{ij} = \max(1, \lfloor \gamma \cdot \min(d_i, d_j) \rfloor)$. The aggregate storage across all edges grows as $O(\gamma \sum_{(i,j)\in\mathcal{E}} \min(d_i, d_j)\, d_i)$, with the hyperparameter $\gamma$ controlling how fast the memory usage scales with the architecture sizes. With the $15$-agent heterogeneous ensemble of models of Tab. 2 and the class-overlap graph of Sec. 5, this places a hard ceiling on the values of

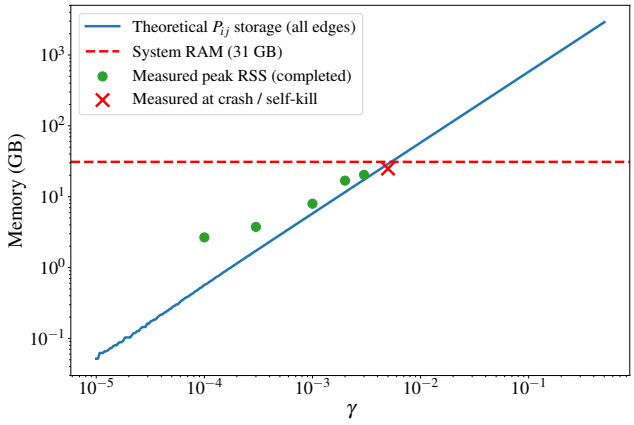


Figure 3: Aggregate memory footprint of the Sheaf-FMTL restriction maps $\{\mathbf{P}_{ij}\}$ as a function of $\gamma$, for the 15-agent architecture and communication graph of Sec. 5 (log-log axes). Solid blue curve: exact storage in the half-precision format used at rest. Dashed line: available system memory. Markers: measured peak resident memory of an isolated, single-epoch Sheaf-FMTL run at each $\gamma$ (crosses mark values at which the run was terminated for exceeding available memory). The operating point used throughout the paper ($\gamma = 10^{-3}$) sits below the single-run ceiling, leaving headroom for the repeated model instantiations of the full evaluation protocol (hyperparameter search preceding each reported run).

$\gamma$ that fit in the machine memory (GPU RAM), well before any accuracy trade-off becomes relevant. Fig. 3 reports a solid blue curve providing the theoretical storage occupied by the restriction maps $\mathbf{P}_{ij}$ and model parameters in the real 15-agent setup, and the green markers then represent the empirical peak resident-memory measurements from isolated single-epoch instantiations of Sheaf-FMTL at escalating $\gamma$. We use $\gamma = 10^{-3}$ throughout the 15-agent benchmark (Tab. 2); a value that sits a margin below the maximum available memory on the considered machine that accounts for the full evaluation protocol rather than a single run.