LATENT DOMAIN DISCOVERY FOR DOMAIN-GENERALISED PROTOTYPICAL FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated Learning (FL) enables collaborative model training without sharing raw data, but heterogeneous client distributions can substantially hinder the learning of transferable representations. This challenge is particularly difficult under domain shift, where clients may contain diverse or mixed domain characteristics, while explicit domain labels are often unavailable. SOTA domain-aware FL methods require predefined domain identities, limiting their applicability to such settings. We propose **FedLDD**, a Latent Domain Discovery framework that jointly discovers latent domain structure from client prototypes and exploits it for domain-aware federated representation learning. Our key insight is that variations in class-wise client prototypes can be factorised into a compact set of latent domain prototypes, while each client can be represented as a soft composition of these latent domains. Specifically, FedLDD factorises the client prototype matrix into a client-to-latent-domain assignment matrix and a latent-domain prototype matrix. The learnt assignments are not treated as domain identities; instead, they provide soft domain supervision for local representation learning. Each client constructs class-specific soft prototypes by combining latent-domain prototypes according to its learnt assignment and uses them to perform domain-consistent prototype alignment and cross-domain prototype contrastive learning. This creates a closed-loop process in which client representations reveal latent domain structure, the discovered structure guides local learning, and the resulting prototypes refine subsequent domain discovery. Extensive experiments on diverse domain-shift benchmarks demonstrate that FedLDD improves federated generalisation while discovering compact latent domain structures. Further analyses show that soft latent-domain supervision effectively captures mixed client characteristics and provides a more flexible representation of cross-client distribution shifts than hard domain assignments.

1 INTRODUCTION

Federated learning (FL) enables multiple clients to collaboratively train machine learning models without directly sharing their private data, making it attractive for applications where data are naturally distributed across devices or organisations Konevcny et al. (2016); McMahan et al. (2017). However, statistical heterogeneity across clients poses fundamental challenges to FL, including unstable local optimisation, slower convergence, and limited generalisation across clients McMahan et al. (2017); Karimireddy et al. (2020). Such statistical and systemic heterogeneity can arise from differences in label distributions, sample sizes, data quality, and input distributions across participating clients Li et al. (2020); Zhang et al. (2021). Among these sources, *domain shift* is particularly crucial, as clients may share the same label space while observing data from different environments, acquisition conditions, styles, or other domain-specific factors Huang et al. (2023). Recent studies have, therefore, increasingly investigated federated domain generalisation and domain-aware learning to improve the robustness of federated models across heterogeneous domains Huang et al. (2023); Le et al. (2026).

Prototype-based FL addresses client heterogeneity by communicating compact class-level representations rather than full model parameters or gradients. FedProto Tan et al. (2022) enables knowledge sharing through class-wise prototypes, but its single prototype per class may collapse domain-specific variations under domain shift. FPL Huang et al. (2023) mitigates this issue by preserving multiple cluster prototypes and unbiased prototypes, while FedDAP Le et al. (2026) further intro-

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



(a) Dual prototype alignment in FedDAP by Le et al.. (b) Proposed soft latent-domain prototype supervision.

Figure 1: Visualisation of domain-aware learning during local training with the existing prototype alignment in FedDAP (left) and our soft latent-domain prototype supervision (right). Marker shapes denote domains, colours denote classes, and stars indicate prototypes. Single- and double-arrow lines indicate pulling representations closer and pushing representations apart, respectively. Unlike FedDAP, our method uses soft assignment weights to adaptively control the pull and push strengths.

duces domain-specific prototypes and domain-aware alignment. Despite its effectiveness, FedDAP requires domain identities to organise client prototypes into domain-specific representations, which may be unavailable or inconsistently defined across clients in practical federated environments. This limitation motivates moving beyond predefined domain identities toward discovering the underlying domain structure directly from distributed prototype representations.

More fundamentally, even when domain identities are available, treating every observed domain as an independent entity may introduce unnecessary redundancy, as different domains can share common visual or semantic characteristics. Instead of assuming a predefined domain taxonomy, we examine a central question: *what latent domain factors explain the prototype variations across clients?* By discovering these factors directly within the prototype space, the server can capture shared structures across correlated domains without relying on explicit or globally consistent domain identities. This data-driven perspective offers a compact and flexible paradigm for domain-aware prototype learning in federated environments with unknown domain structures.

In this work, we introduce FedLDD, a *Federated Latent Domain Discovery* framework that discovers a compact latent domain space directly from client prototypes, without requiring explicit domain information from clients. Our central hypothesis is that cross-client prototype variations can be explained by a small number of latent domain factors, while each client can be represented as a soft composition of these factors. Specifically, FedLDD factorises the collection of class-wise client prototypes into a client-to-latent-domain assignment structure and a set of latent domain prototypes. The assignment associated with each client, therefore, does not represent a predefined domain identity; instead, it describes the client's position within the learnt latent domain space, allowing domain knowledge to emerge naturally from federated representations.

Importantly, rather than treating the latent domain space as an auxiliary representation output, FedLDD leverages the learnt assignments as *soft domain supervision* to actively shape local representations (Fig. 1). Instead of forcing a client to align with a single explicitly specified domain prototype, the local model draws on multiple latent domain prototypes according to their learnt contributions, thereby preserving mixed domain characteristics that may be lost under hard domain assignments. By establishing a closed-loop framework where client prototypes reveal latent domain structure, soft supervision guides local training, and refined representations produce richer subsequent prototypes, FedLDD shifts domain-aware FL from *predefined domain specification* to *data-driven latent domain discovery* without requiring globally consistent domain identities or independent modelling of every observed domain. The key contributions are summarised as follows:

- **Latent domain discovery:** We introduce a domain-label-free framework for prototypical FL that discovers latent domain structures directly from client prototypes. A compact set of latent prototypes represents diverse client distributions through soft compositions, avoiding the need for predefined domain labels or redundant domain-specific prototypes.
- **Soft latent-domain prototype supervision:** We leverage the learnt client-to-latent-domain assignments as soft supervision for local representation learning. This enables clients to exploit

latent domain prototypes for domain-aware feature alignment while preserving mixed-domain characteristics that rigid domain assignments may overlook.

- **Extensive empirical evaluation and analysis:** We extensively evaluate FedLDD under several domain-shift datasets and provide comprehensive analyses of its latent domain structures, soft assignment behaviour, and sensitivity to the number of latent domains, demonstrating how the learnt representation adapts to varying levels of domain complexity.

The remainder of this paper is organised as follows. Section 2 reviews related work on FL under domain shift and prototype-based learning. Section 3 presents the proposed FedLDD framework, including latent domain discovery and soft latent-domain contrastive learning. Section 4 describes the experimental setup and reports comprehensive comparisons and analyses. Finally, Section 5 concludes the paper and discusses potential directions for future work.

2 Related work

We briefly review FL under data heterogeneity and prototypical FL, emphasising how existing approaches represent and exploit domain information.

FL under data heterogeneity. FL was originally introduced to enable collaborative model training without centralising clients' private data McMahan et al. (2017). Subsequent studies have addressed statistical heterogeneity through improved optimisation and personalisation strategies, including FedProx Li et al. (2020) and SCAFFOLD Karimireddy et al. (2020). COPA Wu & Gong (2021) performs collaborative optimisation and aggregation for decentralised domain generalisation and adaptation. FedDG Liu et al. (2021) employs data or feature augmentation to improve generalisation to unseen domains, while FedADG Zhang et al. (2023a) uses adversarial domain alignment to reduce domain-specific discrepancies. FedGA Zhang et al. (2023b) incorporates generalisation into client aggregation, and FedRDN Yan et al. (2025) mitigates feature distribution shifts through federated random data normalisation. These methods explicitly target domain generalisation, but primarily operate on representations, distributions, augmentation, or aggregation rather than explicitly modelling class-level semantic knowledge across domains.

Prototypical FL Prototype-based FL communicates compact class-level representations to capture transferable semantic knowledge while avoiding direct exchange of model parameters or private samples. Prototype-based FL communicates compact semantic representations instead of relying solely on model parameters. FedProto Tan et al. (2022) introduces class prototypes for knowledge sharing under heterogeneous FL, providing a foundation for prototype-based semantic modelling, although it does not explicitly target domain shifts. Recent methods extend this to domain-heterogeneous settings. FPL Huang et al. (2023) uses clustered and unbiased prototypes to capture diverse domain knowledge, while FedPLVM Wang et al. (2024) models cross-domain representation variance through prototype clustering and variance-aware learning. FedDAP Le et al. (2026) further incorporates explicit domain labels to construct domain-aware prototypes and guide local representation learning. While domain-aware supervision can improve the robustness of local representations, requiring explicit domain information introduces two limitations. First, the server relies on domain identities that must be consistently defined and interpreted across clients; furthermore, semantically identical or closely related domains may be represented separately. Second, assigning an independent prototype to each observed domain can be redundant when multiple domains share common underlying factors or when variations in one domain can be explained by combinations of others. These limitations motivate replacing explicit domain-specific prototypes with a compact latent representation that can be discovered directly from client prototypes.

3 Proposed method

3.1 Problem definition and proposed framework

Problem definition. We consider a standard heterogeneous FL setting comprising a central server and K clients, with C classes. Client data distributions are potentially non-IID and may exhibit substantial cross-client domain shifts. Client $k \in \{1, ..., K\}$ holds a private local dataset $\mathcal{D}_k = \{(I_i^k, y_i^k)\}$, where I_i^k denotes the i-th image, and $y_i^k \in \{1, ..., C\}$ denotes its corresponding label,

with $i \in \{1, ..., N_k\}$ and $N_k = |\mathcal{D}_k|$. We assume that each client collects data from a single domain, resulting in heterogeneous local data distributions across clients.

Prototypical learning framework. All clients employ the same model architecture, comprising a feature extractor $f_{\boldsymbol{\theta}}$ and a classifier $g_{\boldsymbol{\phi}}$, parameterised by $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, respectively. Given an input image x, the feature extractor maps x to an embedding representation $f_{\boldsymbol{\theta}}(x)$, which is subsequently fed to the classifier $g_{\boldsymbol{\phi}}$ to predict its class label. At communication round t, each client k performs local training on its private dataset $\mathcal{D}_k$, and subsequently computes class-wise prototypes from the resulting local representations. The updated local model parameters and class-wise prototypes are transmitted to the central server. The server aggregates the local model parameters to obtain the global model and constructs global class-wise prototypes by aggregating the corresponding local prototypes across clients. The resulting global parameters and prototypes are then broadcast to all clients. The global model parameters initialise local training in the subsequent communication round, while the global prototypes provide class-level regularisation targets that encourage clients to learn representations consistent with the global feature space.

Following local training, a client k computes class-wise prototype c by averaging the learnt feature representations of its local samples within class c (denoted by $\mathcal{D}_{k,c}$). Specifically, the class-c prototype of client k, denoted by $\boldsymbol{p}_{k,c}$, is defined as:

$$\boldsymbol{p}_{k,c} = \frac{1}{|\mathcal{D}_{k,c}|} \sum_{I_i \in \mathcal{D}_{k,c}} f_{\boldsymbol{\theta}_k}(I_i). \tag{1}$$

Latent domain discovery framework. FedLDD aims to discover a compact latent domain space directly from client-level prototypes, without requiring explicit domain annotations. The key idea is to model variations among client prototypes as a composition of a number of latent domain factors, while jointly learning the latent-domain prototypes and their client-specific contributions.

Specifically, at each communication round, the server collects local prototypes of participating clients, vectorises each client's prototypes, and stacks them into a matrix $\boldsymbol{X}$. Matrix $\boldsymbol{X}$ is then factorised into a *learnt latent global domain* prototype matrix, $\boldsymbol{G}$, and *client-latent-domain assignment* matrix, $\boldsymbol{A}$. Unlike hard domain assignments, each row of $\boldsymbol{A}$ provides a soft representation of a client's domain composition, allowing a client to be associated with either a single latent domain or a mixture of multiple latent domains. The resulting latent-domain knowledge is subsequently communicated to each client, which constructs domain-aware prototype supervision to promote consistency between its local class prototypes and the corresponding domain-aware global prototypes.

Overall, FedLDD performs prototype-level matrix factorisation on the server to discover a compact latent domain space and characterise each client through soft latent-domain assignments. The discovered domain knowledge is then incorporated into local training through domain-aware prototype regularisation, establishing a closed-loop learning process that jointly captures latent domain structure and improves robust representation learning. An overview of FedLDD is illustrated in Fig. 2, and the communication procedure is summarised in Algorithm 1. The server-side latent domain discovery and client-side domain-aware training procedures are detailed in the following subsections.

3.2 Server-side latent domain discovery

Client-prototype matrix construction. Each client computes a prototype for each class and concatenates them into a single prototype vector. For client k, the resulting prototype vector is defined:

$$\boldsymbol{x}_k = \left[\boldsymbol{p}_{k,1}; \boldsymbol{p}_{k,2}; \ldots; \boldsymbol{p}_{k,C}\right] \in \mathbb{R}^b, \qquad b = C * |\boldsymbol{p}_{k,1}|. \tag{2}$$

At each communication round, suppose that a subset of m clients participates in the federated training process. Let $\mathcal{S} = \{k_1, \ldots, k_m\} \subseteq \{1, \ldots, K\}$ denote the set of participating clients. The server stacks m received prototype vectors to construct the round-specific prototype matrix:

$$\boldsymbol{X} = \left[\boldsymbol{x}_{k_1}, \boldsymbol{x}_{k_2}, \ldots, \boldsymbol{x}_{k_m}\right]^\top \in \mathbb{R}^{m \times b}. \tag{3}$$

Each row of $\boldsymbol{X}$ represents the class-level prototype structure of one participating client.

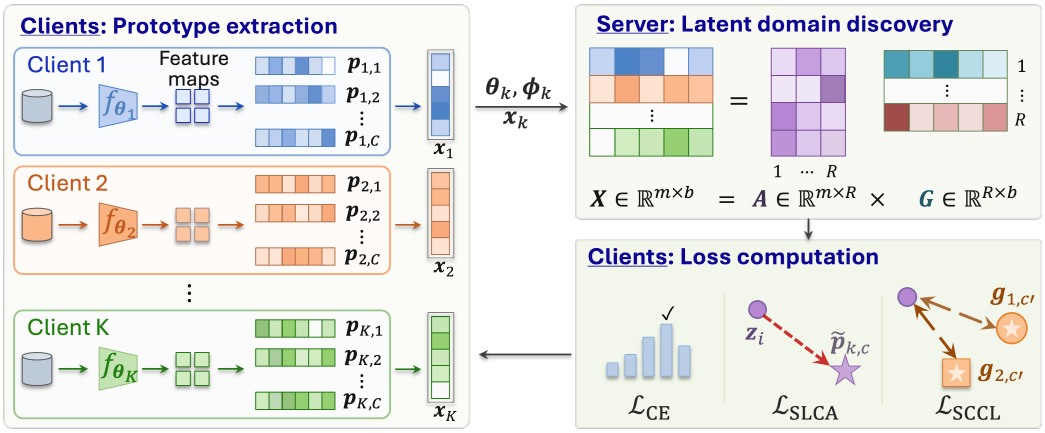


Figure 2: Visualisation of the main steps in the latent domain discovery, FedLDD, framework.

Simplex-structured matrix factorisation. We decompose the matrix $\boldsymbol{X}$ into a *latent global domain prototype matrix*, $\boldsymbol{G}$, and a *client–latent-domain assignment matrix*, $\boldsymbol{A}$:

$$\boldsymbol{X} \approx \boldsymbol{A}\boldsymbol{G}, \qquad \boldsymbol{A} \in \mathbb{R}^{m \times R}, \qquad \boldsymbol{G} \in \mathbb{R}^{R \times b}. \tag{4}$$

Here, R denotes the number of latent domains, which is typically smaller than the number of participating clients m. The factorisation adopts a *simplex structure* Lee & Seung (1999); Liu et al. (2025), by representing all client prototypes through a shared set of R latent domain prototypes in $\boldsymbol{G}$, with $\boldsymbol{A}$ specifying the contribution of each latent domain to each client. Specifically, the assignment coefficients are constrained to form a probability-like distribution over the latent domains:

$$\sum_{r=1}^{R} A_{k,r} = 1, \qquad A_{k,r} \geq 0, \qquad k = 1, \ldots, m. \tag{5}$$

Under this simplex constraint, the prototype representation of client k can be expressed as a convex combination of the latent domain prototypes:

$$\boldsymbol{x}_k \approx \sum_{r=1}^{R} A_{k,r} \boldsymbol{g}_r, \tag{6}$$

where $\boldsymbol{g}_r \in \mathbb{R}^b$ denotes the r-th row of $\boldsymbol{G}$, representing a latent domain prototype. This formulation naturally accommodates both single-domain and mixed-domain clients. For a client k predominantly associated with a single domain, its assignment vector may become nearly one-hot:

$$\boldsymbol{A}_k \approx [0, \ldots, 1, \ldots, 0]. \tag{7}$$

In contrast, a client k whose data distribution exhibits characteristics of multiple domains can be represented by a soft assignment:

$$\boldsymbol{A}_k = [A_{k,1}, , \ldots, , A_{k,R}], \qquad A_{k,r} > 0, \tag{8}$$

where the assignment weights quantify the client's association with each latent domain.

Consequently, rather than learning an independent prototype representation for each predefined domain, as in FedDAP Le et al. (2026), FedLDD represents each client through a soft combination of shared latent-domain components. This formulation captures the underlying domain structure directly from the observed client prototypes, eliminating the need for predefined domain identities while naturally accommodating clients with mixed domain characteristics.

Latent domain discovery algorithm. The latent domain discovery process aims to reconstruct the observed client prototypes using a compact set of shared latent-domain prototype representations. Accordingly, we formulate the server-side factorisation objective as:

$$\mathcal{L}_{\text{server}} = \mathcal{L}_{\text{recon}} + \lambda_{\text{sp}} \mathcal{L}_{\text{sp}} = \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{G}\|_F^2 + \lambda_{\text{sp}} \sum_{r=1}^{R} \|\boldsymbol{A}_{:,r}\|_2 , \tag{9}$$

subject to the domain-assignment constraints in Eq. (5). The reconstruction loss $\mathcal{L}_{\text{recon}}$ encourages the latent domain prototypes to preserve the prototype structures observed across participating clients. The sparsity regularisation term $\mathcal{L}_{\text{sp}}$) promotes group-wise sparsity across the columns of $\boldsymbol{A}$, encouraging the factorisation to activate only a small number of latent domain prototypes.

We adopt an iterative joint optimisation strategy to solve the constrained factorisation problem in Eq. (9), as detailed in Algorithm 2 in Appendix A.1. Two key challenges arise: (i) initialising $\boldsymbol{G}$ and (ii) determining the number of latent domains R. For the initialisation of $\boldsymbol{G}$, we employ a clustering algorithm, such as FINCH Sarfraz et al. (2019) or K-means McQueen (1967), and use the resulting cluster centroids as the initial latent-domain prototypes. This provides a natural initialisation, as each latent-domain prototype represents the centre of a group of similar client-level representations.

Determining the effective value of R is more challenging because the number of underlying domains is generally unavailable a priori. Rather than requiring R to be specified, we initialise the factorisation with an over-complete set of latent components and subsequently determine the effective latent-domain capacity through a prune-and-merge procedure. Specifically, we *merge* latent-domain prototypes that exhibit sufficiently high similarity, since maintaining multiple near-duplicate prototypes provides limited additional representational capacity. This merging strategy is inspired by the prototype-merging mechanism of ISODATA Ball & Hall (1965). In parallel, we *prune* latent components whose corresponding columns in $\boldsymbol{A}$ have a contribution below a predefined threshold, indicating that the associated latent domains are weakly represented across participating clients. The remaining components constitute the active latent domains. This *prune-and-merge* refinement enables the factorisation to begin with an over-complete set of latent components while progressively eliminating weak or redundant ones. Consequently, the effective number of latent domains is determined adaptively by the structure of the learnt client representations, rather than being explicitly prescribed or constrained by the number of observed domains.

3.3 Client-side domain-aware training

At each communication round, the client performs local training by jointly leveraging its local data and latent domain prototypes received from the server. Local data provide direct task supervision, while the latent prototypes provide domain-level guidance for learning representations that are consistent with the underlying domain structure. Specifically, we employ the standard cross-entropy loss for local task supervision and introduce latent-domain prototype supervision through two complementary objectives: (i) soft latent-domain-aware prototype alignment and (ii) soft cross-latent-domain prototype contrastive learning.

Local data supervision. The client optimises its model using the standard cross-entropy loss:

$$\mathcal{L}_{\text{CE}}^{(k)} = -\frac{1}{N_k} \sum_{(I_i, y_i) \in \mathcal{D}_k} y_i \log \Big(\text{softmax} \big(g_{\boldsymbol{\phi}}(\boldsymbol{z}_i) \big) \Big), \quad \boldsymbol{z}_i = f_{\boldsymbol{\theta}}(I_i) \tag{10}$$

This objective provides direct task supervision, preserving the discriminative capability of the local model while adapting its representations to the client's observations.

Latent-domain prototype supervision. In addition to local task supervision, the client exploits the latent domain structure discovered by the server to guide local representation learning. Similar to FedDAP, we encourage each client's class representations to align with the corresponding domain-specific representations while promoting discrimination across different domains. However, whereas FedDAP relies on predefined domain identities and hard domain assignments Le et al. (2026), FedLDD derives domain supervision from the learnt client-to-latent-domain assignments. Consequently, rather than selecting a single domain-specific representation according to a hard domain label, FedLDD performs prototype supervision in a *soft manner*, allowing each latent domain to contribute to the client representation according to its learnt assignment weight.

The soft latent-domain consistent prototype alignment (SLCA) loss accounts for the possibility that a client may exhibit characteristics of multiple latent domains. Rather than aligning local representations with a single domain-specific prototype, FedLDD constructs a personalised class prototype by taking a weighted combination of the corresponding latent-domain prototypes, with each component weighted according to the client's learnt latent-domain assignment. The SLCA loss then

minimises the cosine distance between each local representation and its corresponding personalised class prototype:

$$\mathcal{L}_{\text{SLCA}}^{(k)} = \frac{1}{N_k} \sum_{c=1\ldots C} \sum_{I_i \in \mathcal{D}_{k,c}} \Big(1 - \cos\left(\boldsymbol{z}_i,\, \widetilde{\boldsymbol{p}}_{k,c}\right)\Big), \tag{11}$$

where $\widetilde{\boldsymbol{p}}_{k,c}$ is the personalised prototype for client k and class c, obtained by combining the latent-domain prototypes using the client's learnt assignment weights in the approximation in Eq. 6.

Similarly, we extend cross-domain prototype contrastive learning to a soft setting, termed soft cross-latent-domain contrastive learning (SCCL). SCCL exposes each local representation to the latent-domain prototypes, treating prototypes with the same class label as positives and those with different class labels as negatives. For each local sample, SCCL computes the ratio between the assignment-weighted similarities to the positive latent-domain prototypes and the corresponding similarities to all latent-domain prototypes across all classes, and then averages the resulting contrastive objective over the local dataset:

$$\mathcal{L}_{\text{SCCL}}^{(k)} = \frac{1}{N_k} \sum_{I_i \in \mathcal{D}_k} \frac{\sum_{r=1}^{R} A_{k,r} \exp\Big(\cos\left(\boldsymbol{z}_i,\, \boldsymbol{g}_{r,y_i}\right)\Big)}{\sum_{r=1}^{R}\sum_{c'=1}^{C} \exp\Big(\cos\left(\boldsymbol{z}_i,\, \boldsymbol{g}_{r,c'}\right)\Big)}. \tag{12}$$

The overall local objective is then given by:

$$\mathcal{L}_{\text{local}}^{(k)} = \mathcal{L}_{\text{CE}}^{(k)} + \lambda_{\text{SLCA}}\mathcal{L}_{\text{SLCA}}^{(k)} + \lambda_{\text{SCCL}}\mathcal{L}_{\text{SCCL}}^{(k)}, \tag{13}$$

where λ_{SLCA} and λ_{SCCL} control the contributions of the latent-domain prototype alignment and cross-domain contrastive objectives, respectively. Through this joint optimisation, the client retains supervision from its local data while exploiting the latent domain structure discovered collaboratively across clients.

4 Experiment and analysis

4.1 Experimental setup

Implementation details. For a fair comparison with existing methods, we adopt the same feature extractors, classifiers, and federated parameter settings as in Le et al. (2026). Detailed dataset descriptions and experimental configurations are provided in Appendix A.2.

4.2 Performance comparison with SOTA methods

Quantitative comparison. Table 1 compares FedLDD with existing methods on DomainNet, Office-10, and PACS. FedLDD achieves average accuracies of 64.98%, 70.22%, and 85.25%, respectively, ranking second on DomainNet and Office-10 and first on PACS. Compared with existing prototypical FL methods, FedLDD achieves accuracy improvements of at least 2.76, 1.45, and 3.19 percentage points on DomainNet, Office-10, and PACS, respectively. Compared with FedDAP, which requires domain labels, FedLDD achieves comparable performance without relying on domain annotations. These results highlight the potential of latent domain discovery as a domain-label-agnostic approach for leveraging domain-aware prototypes to guide robust local representation learning in FL under domain shifts. Fig. 3 further demonstrates that the learnt latent-domain prototypes supervise local training, promoting class-wise representation convergence.

4.3 Ablation study and analysis

Impact of loss components. Table 2 evaluates the three learning objectives in FedLDD: SLCA, SCCL, and SR. SLCA and SCCL provide the main performance gains, improving the base configuration by up to 6.91 and 2.85 percentage points, respectively. The combination further improves accuracy to 64.24%, 65.28%, and 84.34% on DomainNet, Office-10, and PACS. SR alone has a

Table 1: Comparison on DomainNet, Office-10, and PACS in terms of top-1 accuracy. "Avg" denotes the mean accuracy across all domains, and Δ denotes the improvement over the FedAvg baseline. The best and second-best results are highlighted in **red** and **blue**, respectively.

Category	Method	Year	DomainNet		Office-10		PACS	
			Avg	Δ	Avg	Δ	Avg	Δ
Non prototypical FL	FedAvg	2017	59.59	–	57.47	–	77.07	–
	FedProx	2020	60.57	+0.98	59.96	+2.49	77.60	+0.53
	MOON	2021	60.15	+0.56	59.89	+2.42	80.59	+3.52
	COPA	2021	61.36	+1.77	64.50	+7.03	78.03	+0.96
	FedGA	2023b	61.00	+1.41	62.49	+5.02	79.13	+2.06
	FedRDN	2025	61.01	+1.42	65.54	+8.07	83.17	+6.10
Prototypical FL **w/o** domain labels	FedProto	2022	60.51	+0.92	64.69	+7.22	80.95	+3.88
	FPL	2023	62.17	+2.58	67.52	+10.05	81.11	+4.04
	FedPLVM	2024	62.22	+2.63	68.77	+11.30	82.06	+4.99
	FedLDD	Proposed	**64.98**	**+5.39**	**70.22**	**+12.75**	**85.25**	**+8.18**
Prototypical FL **with** domain labels	FedDAP	2026	**65.20**	**+5.61**	**72.53**	**+15.06**	**84.63**	**+7.56**

Table 2: Ablation study of the learning objectives with fixed R = 8 in FedLDD on DomainNet, Office-10, and PACS. SLCA, SCCL, and SR denote the three proposed learning objectives. Δ denotes the gain in average accuracy relative to the base configuration without these objectives.

SLCA	SCCL	SR	DomainNet		Office-10		PACS	
			Acc.	Δ	Acc.	Δ	Acc.	Δ
×	×	×	61.03	–	59.71	–	78.90	–
×	×	✓	61.03	0.00	59.85	0.14	80.39	1.49
×	✓	×	61.80	0.77	60.60	0.89	81.75	2.85
×	✓	✓	61.47	0.44	62.75	3.04	82.80	3.90
✓	×	×	61.98	0.95	66.62	6.91	81.36	2.46
✓	×	✓	62.52	1.49	**67.25**	**7.54**	81.82	2.92
✓	✓	×	64.24	3.21	65.28	5.57	**84.34**	**5.44**
✓	✓	✓	**64.68**	**3.65**	64.47	4.76	83.46	4.56

limited effect, while adding it to SLCA provides additional gains of 0.54, 0.63, and 0.46 points. When combined with both SLCA and SCCL, SR yields 64.68%, 64.47%, and 83.46%. Since these experiments fix R at 8, which is not the optimal capacity identified in Table 3, these results do not necessarily indicate that SR degrades the full objective. Overall, SR provides an additional but configuration-dependent contribution, while SLCA and SCCL account for the main performance improvements.

Sensitivity to latent-domain capacity. Table 3 examines the sensitivity of FedLDD to the latent-domain capacity R. Overall, performance remains relatively stable across the tested values, although the effect of increasing R is dataset-dependent and non-monotonic. The best results are obtained at $R = 6$ on DomainNet (64.98%) and Office-10 (70.22%), and at $R = 4$ on PACS (85.25%). Notably, increasing R does not consistently improve performance, while even the smallest capacity, $R = 2$, achieves competitive results across all three datasets. These results suggest that FedLDD does not require a large latent-domain capacity and is reasonably robust to the choice of R.

Impact of sparse regularisation and merge-and-prune refinement. Table 4 evaluates the contributions of sparse regularisation (SR), merge-and-prune refinement, and SSML Liu et al. (2025) across different latent-domain capacities R. Removing SR has a particularly pronounced effect on Office-10, where accuracy drops from 70.22% to 63.60% at $R = 6$, while its effect is smaller and less consistent on DomainNet and PACS. Replacing the proposed factoriser with SSMF leads to substantially larger degradation, especially on Office-10, reducing accuracy from 70.22% to 54.14% at $R = 6$ and from 64.47% to 50.70% at $R = 8$. Removing merge-and-prune refinement has a

smaller and dataset-dependent effect; for example, it reduces accuracy by 2.49 and 0.80 points on Office-10 and by 2.17 and 1.01 points on PACS at $R = 4$ and $R = 6$, respectively. Overall, the results highlight the importance of the proposed factoriser, while SR and merge-and-prune refinement provide additional, configuration-dependent benefits.

Impact of latent-domain refinement. Table 5 evaluates whether additional revitalisation of pruned latent domains is beneficial. The Rev. variant, which reactivates latent domains whose contribution becomes too small, produces mixed effects across datasets and latent capacities. While it improves DomainNet at $R = 2$ and PACS slightly at $R = 2$, it consistently degrades performance on Office-10 for $R \geq 3$ and also reduces accuracy on several DomainNet and PACS settings. These results suggest that repeatedly revitalizing low-contribution factors does not provide a consistent benefit. Instead, the default pruning mechanism appears sufficient to control redundant or weak latent factors without introducing additional refinement steps.

Impact of latent-domain initialisation and clustering. Table 6 examines whether FedLDD depends on the initialisation of the latent-domain prototypes or on a specific clustering algorithm. Reinitialising $\boldsymbol{G}$ at every communication round (Init) remains competitive at small R, but its performance becomes less stable as the latent-domain capacity increases. In particular, at $R = 6$, Init reduces accuracy from 64.98% to 63.95% on DomainNet, from 70.22% to 64.21% on Office-10, and from 85.01% to 83.40% on PACS. Retaining $\boldsymbol{G}$ across rounds therefore provides a persistent representation of the latent structure and avoids repeatedly estimating the factorisation from scratch. Replacing Finch with K-means produces comparable results in several settings, although its performance varies more across datasets and values of R. This indicates that the discovered representation is not tied to a particular clustering algorithm, while the clustering strategy can still affect the resulting latent factorisation.

Impact of client participation and domain composition. Table 7 evaluates the robustness of FedLDD under different client participation and client-domain composition settings. The online-ratio setting reduces participation to 25% per round while proportionally increasing the number of communication rounds. Its performance remains comparable to the default setting, with similar results on DomainNet and PACS, indicating that latent-domain structure can still be learnt under reduced per-round participation. In contrast, assigning all domains to every client causes a substantial degradation across all datasets. At $R = 4$, for example, accuracy decreases from 64.49% to 56.07% on DomainNet, from 70.10% to 61.50% on Office-10, and from 85.25% to 74.40% on PACS. Unlike the default heterogeneous partition, where clients exhibit different domain compositions, the multi-domain setting substantially reduces cross-client distributional variation. The resulting degradation therefore highlights the importance of heterogeneous client-domain composition for exposing the cross-client structure that FedLDD factorises into latent-domain representations.

5 Conclusion

In this work, we introduced FedLDD, a FL framework that discovers and exploits latent domain structure without requiring explicit domain labels or predefined domain identities. FedLDD characterises cross-client heterogeneity through a compact set of latent-domain prototypes and soft client-to-latent-domain assignments, with an adaptive prune-and-merge mechanism that eliminates weak or redundant latent factors. The discovered structure is further incorporated into local representation learning through soft latent-domain prototype supervision, forming a closed-loop process between latent-domain discovery and representation learning. Experiments across diverse domain-shift benchmarks demonstrate that FedLDD effectively discovers latent-domain prototypes and leverages them to improve federated generalisation. The results further show that soft client-to-latent-domain assignments can capture mixed client characteristics, while the discovered latent factors need not correspond one-to-one with observed domains. This flexibility is particularly valuable in practical federated deployments, where clients may exhibit heterogeneous or overlapping distributional characteristics and reliable domain metadata may be unavailable or costly to obtain. FedLDD is extensively evaluated under the FL setting under domain shifts; however, broader forms of federated heterogeneity, such as label-distribution shift and quantity skew, remain unexplored. Extending the evaluation to these heterogeneous settings is an important direction for future work to further assess the robustness and generalisability of FedLDD.

AI USE STATEMENT

AI-assisted tools, including ChatGPT, Claude, and Gemini, were used as supplementary tools during the research and manuscript preparation process, code assistance and revision, manuscript editing, and assistance with result analysis and visualisation. The authors independently reviewed and verified all AI-assisted outputs and made the final decisions regarding the research methodology, experiments, analysis, and presentation. The authors take full responsibility for the content and conclusions of this paper.

REFERENCES

Geoffrey H. Ball and David J. Hall. ISODATA, a novel method of data analysis and pattern classification. Technical Report Technical Report AD-699616, Stanford Research Institute, Menlo Park, CA, USA, 1965.

Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Proc. CVPR*, pp. 2066–2073, 2012.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, pp. 770–778, 2016.

Wenke Huang, Mang Ye, Zekun Shi, He Li, and Bo Du. Rethinking federated learning with domain shift: A prototype view. In *Proc. CVPR*, pp. 16312–16322, 2023.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *Proc. ICML*, pp. 5132–5143. PMLR, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. ICLR*, 2015.

Jakub Konevcny, H. Brendan McMahan, Felix X. Yu, Peter Richtarik, Ananda Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. In *Proc. NIPS Workshop on Private Multi-Party Machine Learning*, 2016.

Huy Q. Le, Loc X. Nguyen, Yu Qiao, Seong Tae Kim, Eui-Nam Huh, and Choong Seon Hong. Feddap: Domain-aware prototype learning for federated learning under domain shift. In *Proc. CVPR*, pp. 3390–3399, 2026.

Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proc. ICCV*, pp. 5543–5551, 2017.

Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proc. CVPR*, pp. 10708–10717, 2021.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proc. MLSys*, pp. 429–450, 2020.

Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. FedDG: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In *Proc. CVPR*, pp. 1013–1023, 2021.

Wenxuan Liu, Thomas Brendan Murphy, and Lorraine Brennan. Simplex-structured matrix factorisation: application of soft clustering to metabolomic data. *Scientific Reports*, 15(1):17817, 2025.

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

James B McQueen. Some methods of classification and analysis of multivariate observations. In *Proc. Berkeley Symp. on Math. Statist. and Prob.*, pp. 281–297, 1967.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proc. ICCV*, pp. 1406–1415, 2019.

Saquib Sarfraz, Vivek Sharma, and Rainer Stiefelhagen. Efficient parameter-free clustering using first neighbor relations. In *Proc. CVPR*, pp. 8926–8935, 2019.

Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Proc. AAAI*, pp. 8432–8440, 2022.

Lei Wang, Jieming Bian, Letian Zhang, Chen Chen, and Jie Xu. Taming cross-domain representation variance in federated prototype learning with heterogeneous data domains. In *Proc. NeurIPS*, pp. 88348–88372, 2024.

Guile Wu and Shaogang Gong. Collaborative optimization and aggregation for decentralized domain generalization and adaptation. In *Proc. ICCV*, pp. 6464–6473, 2021.

Yunlu Yan, Huazhu Fu, Yuexiang Li, Jinheng Xie, Jun Ma, Guang Yang, and Lei Zhu. A simple data augmentation for feature distribution skewed federated learning. In *Proc. CVPR*, pp. 25749–25758, 2025.

Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. A survey on federated learning. *Knowledge-Based Systems*, 216, 2021.

Liling Zhang, Xinyu Lei, Yichun Shi, Hongyu Huang, and Chao Chen. Federated learning for IoT devices with domain generalization. *IEEE Internet of Things Journal*, 10(11):9622–9633, 2023a.

Ruipeng Zhang, Qinwei Xu, Jiangchao Yao, Ya Zhang, Qi Tian, and Yanfeng Wang. Federated domain generalization with generalization adjustment. In *Proc. CVPR*, pp. 3954–3963, 2023b.

Tailin Zhou, Jun Zhang, and Danny HK Tsang. FedFA: Federated learning with feature anchors to align features and classifiers for heterogeneous data. *IEEE Transactions on Mobile Computing*, pp. 6731–6742, 2024.

A Appendix

A.1 Algorithm details

This section presents the algorithms for latent domain discovery procedure and joint optimisation of simplex-structured matrix factorisation in Algorithms 1 and 2, respectively.

A.2 Implementation details

Similar to settings in FedDAP Le et al. (2026), we use ResNet-10 He et al. (2016) for DomainNet and Office-10, and ResNet-18 for PACS. All experiments are conducted for 100 communication rounds, with each participating client performing 10 local epochs per round. The local batch size is set to 32 for DomainNet and Office-10 and 16 for PACS. Model performance is evaluated using top-1 classification accuracy. Each experiment is independently repeated 3 times, and we report the mean accuracy averaged over the final 5 communication rounds. All experiments are conducted on a single NVIDIA TITAN RTX GPU with 24 GB of memory.

Hyperparameter settings. Unless otherwise specified, we use the same training settings as the corresponding federated learning baselines. Client-side model training is performed using the SGD optimiser, while the server-side latent-domain factorisation is optimised using Adam. We perform 50 alternating/joint optimisation steps per communication round with a learning rate of 0.05. The factorisation objective uses $\lambda_{\text{sparse}} = 0.01$ for group sparsity on the columns of $\boldsymbol{A}$. After factorisation, latent domains with assignment mass below $\tau_p = 0.05$ of the maximum column mass are

Algorithm 1 Latent domain discovery (FedLDD) procedure

Require: Communication rounds T, local epochs E, number of clients K, local datasets $\{\mathcal{D}_k\}_{k=1}^{K}$
Ensure: Global model parameters $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, latent-domain prototypes $\boldsymbol{G}$, latent-domain assignment matrix $\boldsymbol{A}$

Initialise global model parameters $\boldsymbol{G}$ and $\boldsymbol{A}$, $\boldsymbol{\theta}, \boldsymbol{\phi}$
for $t = 0, \ldots, T-1$ **do**
  for each client $k \in \mathcal{S}_t \subseteq \{1, \ldots, K\}$ **do**
    $\boldsymbol{\theta}_t^k, \boldsymbol{\phi}_t^k \leftarrow$ LocalUpdate($\boldsymbol{\theta}_t, \boldsymbol{\phi}_t, \boldsymbol{A}_t, \boldsymbol{G}_t$)
    Compute local class-wise prototypes $\boldsymbol{p}_k^c$ according to Eq. 1
  end for
  Stack the local prototypes from all participating clients in $\mathcal{S}_t$ into $\boldsymbol{X}_t$ according to Eq. 3
  Factorise $\boldsymbol{X}_t$ into $\boldsymbol{A}_t, \boldsymbol{G_t}$ using Algorithm 2
  Global model aggregation:

$$\boldsymbol{\theta}_{t+1} \leftarrow \sum_{k \in \mathcal{S}_t} \frac{|\mathcal{D}_k|}{\sum_{j \in \mathcal{S}_t} |\mathcal{D}_j|} \boldsymbol{\theta}_t^k, \qquad \boldsymbol{\phi}_{t+1} \leftarrow \sum_{k \in \mathcal{S}_t} \frac{|\mathcal{D}_k|}{\sum_{j \in \mathcal{S}_t} |\mathcal{D}_j|} \boldsymbol{\phi}_t^k$$

end for

LocalUpdate ($\boldsymbol{\theta}_t, \boldsymbol{\phi}_t, \boldsymbol{A}_t, \boldsymbol{G}_t$):
for $e = 1, \ldots, E$ **do**
  for each mini-batch $\mathcal{B} \subset \mathcal{D}_k$ **do**
    Update $\boldsymbol{\theta}_t, \boldsymbol{\phi}_t$ using $\mathcal{B}$, $\boldsymbol{A}_t$ and $\boldsymbol{G}_t$ according to Eq. 13
  end for
end for
return $\boldsymbol{\theta}_t^k, \boldsymbol{\phi}_t^k$

Algorithm 2 Joint optimisation for simplex-structured matrix factorisation

Require: Client prototype matrix $\boldsymbol{X} \in \mathbb{R}^{m \times b}$, maximum number of latent domains R, sparsity coefficient λ_{sp}, latent prototype matrix initialisation $\boldsymbol{G}$, and number of optimisation steps T.
Ensure: Client–domain assignment matrix $\boldsymbol{A}$ and latent-domain prototype matrix $\boldsymbol{G}$.

Initialise $\boldsymbol{A} \in \mathbb{R}^{m \times R}$ with $A_{kr} = 1/R$
for $t = 1$ to T **do**
  Compute the reconstruction loss:

$$\mathcal{L}_{\text{recon}} = \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{G}\|_F^2$$

  Compute the group sparsity loss:

$$\mathcal{L}_{\text{sp}} = \sum_{r=1}^{R} \|\boldsymbol{A}_{:,r}\|_2$$

  Compute the total objective:

$$\mathcal{L}_{\text{server}} = \mathcal{L}_{\text{recon}} + \lambda_{\text{sp}} \mathcal{L}_{\text{sp}}$$

  Jointly update $\boldsymbol{A}$ and $\boldsymbol{G}$ using Adam Kingma & Ba (2015):

$$(\boldsymbol{A}, \boldsymbol{G}) \leftarrow \text{Adam}\left(\nabla_{\boldsymbol{A},\boldsymbol{G}} \mathcal{L}_{\text{server}}\right)$$

  Project each row of $\boldsymbol{A}$ onto the probability simplex:

$$\boldsymbol{A}_{k,:} \leftarrow \Pi_{\Delta}(\boldsymbol{A}_{k,:}), \qquad \Delta = \left\{ \boldsymbol{a} \geq 0 : \sum_{r=1}^{R} a_r = 1 \right\}$$

end for
return $\boldsymbol{A}, \boldsymbol{G}$

pruned, while domains with cosine similarity above $\tau_m = 0.95$ are merged. For the prototype learning objectives, we set $\lambda_{SLCA} = 10$ and $\lambda_{SCCL} = 1$ for the intra- and inter-level terms, respectively. We set the number of optimisation steps in Algorithm 2 to 50 and the number of communication

rounds in Algorithm 1 to 100. These hyperparameters are fixed across all datasets and communication rounds unless explicitly stated otherwise.

Datasets. We evaluate our proposed method on three established multi-domain benchmarks with varying degrees of visual domain heterogeneity: DomainNet Peng et al. (2019), Office-10 Gong et al. (2012), and PACS Li et al. (2017). **DomainNet** spans six diverse domains: Clipart, Infograph, Painting, Quickdraw, Real, and Sketch; and we follow Zhou et al. (2024) by selecting 10 classes from its original 345 categories. **Office-10** contains four domains: Caltech, Amazon, Webcam, and DSLR, with 10 shared object categories. **PACS** covers seven object categories across four distinct visual styles: Photo, Art Painting, Cartoon, and Sketch. Together, these benchmarks provide diverse domain shifts and varying domain characteristics for evaluating latent domain discovery and domain-generalised federated learning.

Client settings. We follow the federated domain-shift setting of Le et al. (2026), where clients are distributed across multiple domains and may observe data from different domain sources. We consider 20 clients for DomainNet and 10 clients for both Office-10 and PACS. The clients are allocated across domains as follows: *DomainNet*: Clipart (3), Infograph (1), Painting (4), Quickdraw (6), Real (4), and Sketch (2); *Office-10*: Caltech (3), Amazon (2), Webcam (1), and DSLR (4); and *PACS*: Photo (3), Art Painting (2), Cartoon (1), and Sketch (4). To simulate heterogeneous local data distributions, we randomly sample 10%, 20%, and 30% of the available data for each client in DomainNet, Office-10, and PACS, respectively.

A.3 Tables and figures

Table 3: Sensitivity of FedLDD to the latent-domain capacity R on DomainNet, Office-10, and PACS. The best result for each dataset is highlighted in **bold**.

R	**DomainNet**	**Office-10**	**PACS**
2	64.49	68.81	85.02
3	64.72	67.72	84.20
4	64.49	70.10	**85.25**
5	64.16	66.66	84.26
6	**64.98**	**70.22**	85.01
8	64.68	64.47	83.46

Table 4: Ablation of factorisation in FedLDD. **Full** denotes the complete joint optimisation described in Algorithm 2; −**M&P** removes the merging and pruning steps; −**SR** removes sparse regularisation; and **SSMF** replaces the joint optimisation with the alternative optimisation Liu et al..

	DomainNet				**Office-10**				**PACS**			
R	Full	−M&P	−SR	SSMF	Full	−M&P	−SR	SSMF	Full	−M&P	−SR	SSMF
2	64.49	64.29	65.02	64.62	68.81	69.06	69.97	66.79	85.02	84.79	84.60	84.89
3	64.72	64.46	63.62	64.25	67.72	69.28	64.25	66.04	84.20	83.23	84.43	85.53
4	64.49	64.61	63.57	64.06	70.10	67.61	63.43	58.10	85.25	83.08	82.84	85.08
5	64.16	63.90	63.40	63.31	66.66	68.21	64.26	56.21	84.26	84.94	83.19	84.56
6	64.98	64.10	64.43	61.58	70.22	69.42	63.60	54.14	85.01	84.00	83.98	84.33
8	64.68	64.20	64.24	62.34	64.47	63.48	65.28	50.70	83.46	84.30	84.34	81.90

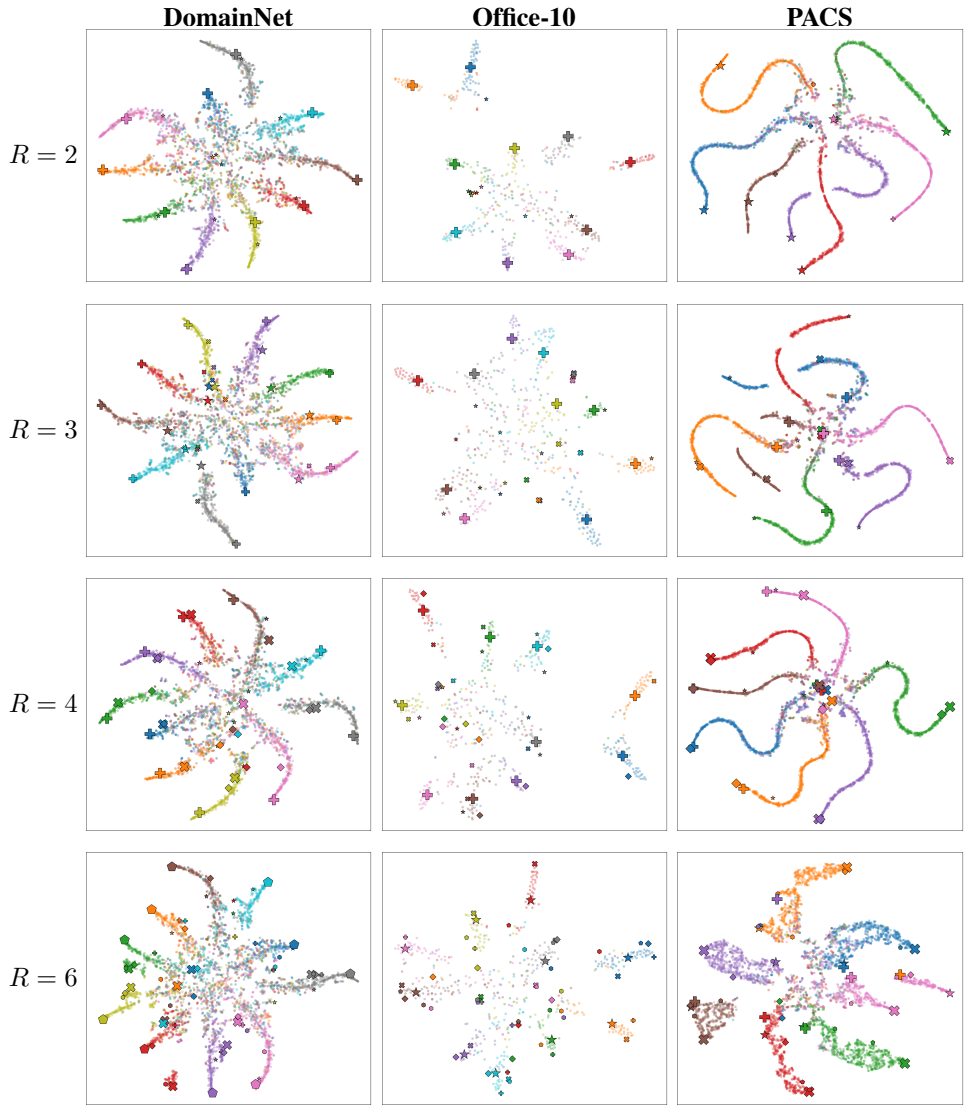


Figure 3: t-SNE visualisation of the latent-domain representations obtained with K-means and latent-domain pruning across DomainNet, Office-10, and PACS. Each column corresponds to a benchmark dataset, while each row shows a different latent-domain capacity R. The visualisations illustrate how the learnt latent-domain representations vary as the number of latent domains increases.

Table 5: Ablation of latent-domain refinement in FedLDD. "Rev." additionally revitalises pruned latent domains when their contribution becomes too small, while the default model only prunes low-contribution latent domains.

	DomainNet		**Office-10**		**PACS**	
R	**Ours**	**Rev.**	**Ours**	**Rev.**	**Ours**	**Rev.**
2	64.49	65.02	68.81	69.97	85.02	85.18
3	64.72	63.62	67.72	64.25	84.20	83.19
4	64.49	63.57	70.10	63.43	85.25	82.84
6	64.98	64.43	70.22	63.60	85.01	83.98

Table 6: Ablation of latent-domain initialisation and clustering in FedLDD. "Init" reinitialises the latent-domain prototypes G at the beginning of each communication round, while "Kmean" replaces Finch clustering with K-means.

	DomainNet			**Office-10**			**PACS**		
R	**Finch**	**Init**	**Kmean**	**Finch**	**Init**	**Kmean**	**Finch**	**Init**	**Kmean**
2	64.49	64.60	64.57	68.81	70.47	70.51	85.02	84.65	85.10
3	64.72	64.49	64.20	67.72	69.39	65.62	84.20	83.31	84.85
4	64.49	64.22	64.89	70.10	68.69	68.23	85.25	83.06	84.86
6	64.98	63.95	63.26	70.22	64.21	67.78	85.01	83.40	84.08

Table 7: Ablation of client participation and client-domain composition in FedLDD. "1D-All" uses all clients with one domain per client, "1D-25%" uses 25% client participation with one domain per client, and "All-D" assigns all domains to every client.

	DomainNet			**Office-10**			**PACS**		
R	**1D-All**	**1D-25%**	**All-D**	**1D-All**	**1D-25%**	**All-D**	**1D-All**	**1D-25%**	**All-D**
2	64.49	64.57	55.43	68.81	71.46	61.01	85.02	84.32	74.58
3	64.72	63.59	55.98	67.72	71.50	61.05	84.20	84.12	74.62
4	64.49	63.34	56.07	70.10	67.83	61.50	85.25	82.94	74.40
6	64.98	63.88	55.64	70.22	69.03	59.03	85.01	84.79	74.26