Discovering Latent Knowledge Prototypes for Heterogeneous Federated Learning

Qilei Li [a,b,*] **and Ahmed M. Abdelmoniem** [a]

[a]School of Electronic Engineering and Computer Science, Queen Mary University of London, London, E1 4NS, United Kingdom.
[b]Faculty of Artificial Intelligence in Education, Central China Normal University, Wuhan, 430079, China.

Abstract. Federated learning (FL) is crucial for ensuring data privacy, a major concern in many applications. However, FL faces significant challenges due to data and model heterogeneity arising from diverse learning environments and the varying capabilities of participating entities. Most existing methods primarily concentrate on aggregating knowledge that is represented by models, logits, or features. which rely on specific assumptions that may not hold in real-world scenarios and thus fail to address both data and model heterogeneity simultaneously. In this work, we aim to address these challenges by tackling heterogeneity from both model and data perspectives while maintaining efficiency. To this end, we leverage locally encoded latent prototypes produced from the local knowledge memory bank to represent per-client knowledge updates, which are then aggregated on the server and transferred back to the clients for knowledge decoding and integration as global constraints for further local training. Considering the heterogeneity in model architectures, we design the knowledge encoder and decoder to be compatible with different model architectures and ensure robust prototype aggregation by aligning latent spaces to a common prior distribution, to enhance compatibility under diverse data distributions. We evaluate our method on multiple benchmarks and demonstrate its superior performance in terms of accuracy and effectiveness under various heterogeneous settings.

1 Introduction

Federated learning (FL) [23] has emerged as an important learning paradigm for collaborative model training while preserving data privacy. By enabling decentralized learning across clients, FL avoids sharing raw data, thus addressing critical privacy concerns. However, real-world FL scenarios are often characterized by substantial heterogeneity, such as in both client data distributions and model architectures. Such entangled heterogeneity presents significant challenges for effective aggregation and robust model performance across diverse clients, especially in settings where clients have varying computational capabilities and domain-specific requirements.

Numerous FL methods, such as FedAvg [10], rely on aggregating client models under the assumption of homogeneous architectures and Independent and Identically Distributed (IID) data. While this approach is simple and scalable, it breaks down when applied in scenarios where clients deploy heterogeneous model tailored to their specific needs. In such scenarios, the differences in model structures across clients cause FedAvg and other parameter-averaging methods to be ineffective [3]. Furthermore, the reliance on the IID assumption fails to address the natural diversity in the real-world, which leads to suboptimal generalization and performance degradation [22].

Table 1: Comparison of different types of aggregated knowledge in FL. MH - Model Heterogeneity, DH - Data Heterogeneity, PP - Privacy Preservation, CE - Communication Efficiency.

Knowledge Type	MH	DH	PP	CE
Model Parameters	✗	✗	✗	✗
Logits Predictions	✓	✗	✓	✓
Feature Prototype	✗	✓	✗	✓
Latent Prototype	✓	✓	✓	✓

In addition to model parameters, knowledge can also be presented as logits or features for aggregation. We compared different knowledge aggregation methods in Tabel 1. Specifically, Logits-based methods, such as FedDistill [5], address model heterogeneity by aggregating predicted logits instead of raw parameters. These methods bypass the need for architectural alignment and rely on soft-label information to transfer knowledge across clients. Although effective in alleviating architectural mismatches, logits carry limited information due to their low dimensionality, which can only capture class-level probabilities without semantic richness [11]. Consequently, these methods often struggle to generalize well in complex tasks, where high-dimensional feature representations are essential for robust learning. Feature-based approaches, such as FedProto [14], represent another line of work where class-level feature prototypes are aggregated instead of parameters or logits. By aligning feature spaces across clients, FedProto offers better generalization under both model and data heterogeneity. However, this approach is not without drawbacks. The transmission of raw feature embeddings exposes clients to significant privacy risks, as adversaries can exploit feature inversion techniques [21] to reconstruct sensitive data. This trade-off between feature alignment and privacy preservation underlines the limitations of existing feature-based methods. In contrast, our latent prototypes are compact and sampled with Gaussian noise, making them inherently more resistant to inversion and compliant with privacy-preserving principles.

The intertwined challenges of both data and model heterogeneity exacerbate these limitations. Clients with non-IID data distributions introduce further complexity into aggregation, amplifying the difficulty of aligning feature spaces across diverse models. Existing methods typically address either model or data heterogeneity but rarely provide a unified solution capable of handling both challenges simultaneously.

* Corresponding Author. Completed the work while at Queen Mary University of London, UK. Email: *q.li@qmul.ac.uk, ahmed.sayed@qmul.ac.uk*

Several recent strategies have been proposed to address these intertwined challenges in federated learning (FL). For example, FedPAC [17] adopts a cautious collaboration mechanism to address personalization and heterogeneity, but it faces difficulties when collaboration decisions are overly conservative or when adapting to highly diverse client scenarios. FedCAC [20] combines feature alignment with classifier collaboration, providing strong personalization capabilities, but it introduces significant computational overhead when applied to large-scale federated systems. In addition, these approaches, while addressing certain aspects of heterogeneity, often rely on assumptions that may not hold in real-world scenarios, such as precise knowledge of client data distributions or the ability to perform effective knowledge distillation across diverse datasets, which can be impractical in privacy-sensitive and heterogeneous environments [26]. Other works introduce significant computational or communication overheads, limiting their applicability in real-world FL deployments [12]. The intertwined challenges of model and data heterogeneity remain insufficiently addressed, whether considered separately or together. It highlights the urgent need for a unified and efficient solution that can operate under practical constraints.

In this work, we propose a novel federated learning framework termed as Latent Knowledge Prototypes (FedLKP), to address the intertwined challenges of model and data heterogeneity while ensuring efficiency, robustness, and privacy preservation. Our approach centres on the discovery and utilization of compact knowledge representation encoded from local training samples, encoded from distributed local clients, to achieve seamless integration of client-specific insights with global learning objectives.

To this end, we introduce, to each client, a knowledge vault as its secure local repository to store client-specific knowledge and extract from the value globally aligned latent prototypes that serve as per-client knowledge updates. These prototypes are aggregated on the server in a global knowledge consensus manner and then decoded via a local knowledge decoder to guide further local training. This enables effective knowledge sharing across heterogeneous clients by serving as individual client constraints.

1.1 Contributions

By tailoring the knowledge encoder and decoder to meet the dimensional requirements respect to the model architectures in each learning entity, our framework can efficiently address model heterogeneity in FL systems. By leveraging transformed latent prototypes, our framework incorporates global knowledge to prevent overfitting to local datasets while enhancing privacy by mitigating the risk of feature inversion. In short, FedLKP provides a robust and scalable solution that balances the need for personalized local training with the benefits of globally aligned guidance, achieving superior performance in diverse FL scenarios. Our contributions are summerised as follows:

- We propose Latent Knowledge Prototypes (FedLKP), to address both model and data heterogeneity in federated learning by leveraging locally aligned latent prototypes, that can be aggregated globally and decoded locally as a constraint for global knowledge distillation.
- We realize FedLKP with a knowledge encoder that generates latent prototypes from local knowledge vaults and a knowledge decoder that decodes the global knowledge distributed from the global server, enabling a privacy-aware local training while also benefiting from globally aligned prototype guidance.
- We demonstrate that FedLKP provides a robust and efficient solution that balances the need for personalized local training with the benefits of globally aligned guidance, showing superior performance in terms of accuracy, efficiency, and privacy preservation under various heterogeneous settings.

2 Related Work

Heterogeneous Federated Learning. Federated Learning (FL) [23] enables collaborative model training across decentralized data sources while preserving data privacy. However, real-world FL systems must contend with heterogeneous client environments characterized by differences in model architectures, computational capacities, and non-IID data distributions. These challenges significantly hinder the effectiveness of FL methods. Classical approaches like FedAvg [10] aggregate client-side model parameters and assume IID data and homogeneous architectures. While effective in controlled environments, these assumptions lead to suboptimal performance in heterogeneous settings. To address this, recent methods explore client-specific optimizations. For example, FedADKD [26] employs adaptive knowledge distillation to address data heterogeneity without requiring identical model architectures. Similarly, HFedCWA [2] uses contribution-weighted aggregation, accounting for data distribution differences and client resource constraints. However, both approaches rely on idealized assumptions, such as accurately estimating client contributions, which may not hold in practice. Despite these advancements, FL systems often fail to balance the trade-offs between efficiency and robustness in highly heterogeneous environments. This gap highlights the need for unified solutions that simultaneously address model and data heterogeneity while maintaining privacy.

Knowledge Aggregation in FL Aggregation methods in FL can be broadly categorized into model-based, feature-based, and logit-based approaches, each addressing specific aspects of heterogeneity. *Model-Based Aggregation*: Classical methods like FedAvg and its variants aggregate model parameters to achieve consensus. However, when clients use heterogeneous architectures, such aggregation becomes infeasible due to incompatible parameter spaces [10].Recent methods like FedPAC [17] and FedCAC [20] attempt to address heterogeneity by leveraging cautious collaboration and classifier alignment, respectively. However, these approaches may struggle with computational overhead or conservative updates that hinder effective aggregation in dynamic environments. *Feature-Based Aggregation*: FedProto [14] aligns feature spaces by aggregating class-level feature prototypes, offering better generalization under model and data heterogeneity. However, transmitting raw feature embeddings exposes clients to potential privacy risks, as attackers can reconstruct sensitive data using inversion techniques [4]. Additionally, the resources required to align high-dimensional feature spaces often introduce computational overhead. *Logit-Based Aggregation*: FedDistill [5] combining logit aggregation with augmentation techniques to improve performance under non-IID conditions but still lacking the semantic richness necessary for diverse applications. While logits provide a lightweight representation, their low dimensionality limits their ability to capture rich semantic information, which is essential for robust generalization in complex tasks. In addition, logit-based FL methods are prone to privacy leakage once being attacked [19]. Recent advancements, such as FedCAC [20], focus on leveraging feature alignment and guiding local training in heterogeneous environments, which overlooks the privacy issue of features

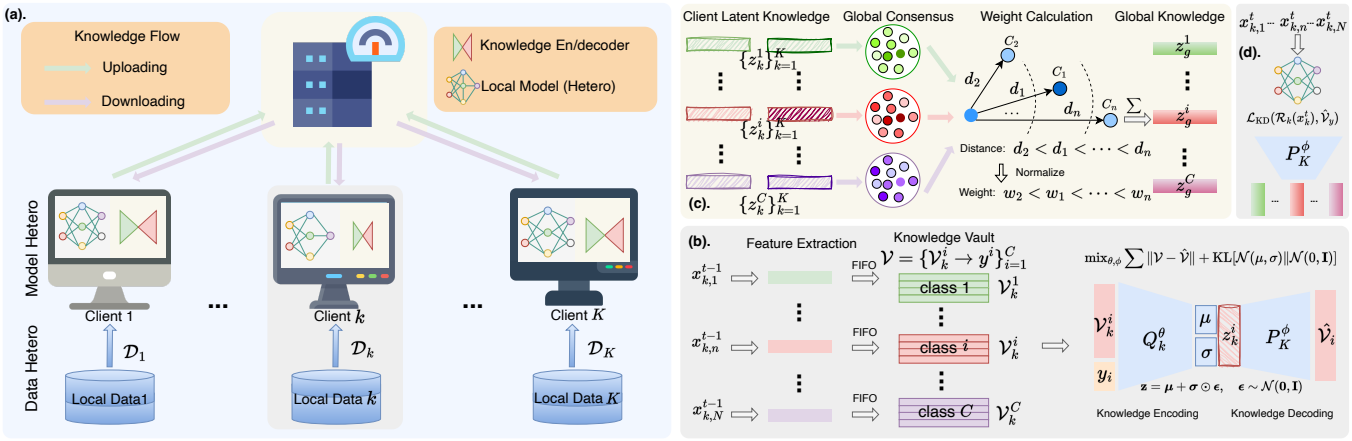

Figure 1: Illustration of the proposed Latent Knowledge Prototypes (FedLKP) framework. (a) The overall workflow encompasses local knowledge extraction and uploading, global knowledge aggregation, and downloading. (b) For each client k, the local private data D_k is utilized for feature extraction to update the knowledge vault $\mathcal{V}_k$, which is subsequently encoded into latent prototypes $\mathbf{z}_k$ and uploaded to the server. (c) The server aggregates the latent prototypes from all clients and produces global prototypes $\mathbf{z}_g$ for each label in a globally consensus-aware manner, assigning weights based on distance. (d) The global prototypes are then distributed to each client, decoded into local knowledge $\hat{\mathcal{V}}_k$, and used as constraints during local training via knowledge distillation.

that are prone to inversion attacks [9]. While these methods demonstrate improved personalization and generalization, they often rely on additional assumptions, such as accurate distribution estimation or computationally expensive alignment mechanisms, which can hinder their practicality in large-scale FL systems.

Latent Representation in Deep Learning. Latent representations are fundamental to deep learning models, capturing compressed and meaningful feature representations. Variational Autoencoders (VAEs) [7] are widely used for learning such representations. It can align latent vectors with a normal distribution, which facilitates model regularization, and improves generalization while reducing overfitting. Introducing noise into latent vectors has further enhanced privacy by obfuscating sensitive information, making it an effective mechanism for privacy-preserving federated learning. In this paper, we demonstrate that the advantage of latent representations can be extended to federated learning, in which the latent prototypes can be used as compact and privacy-preserving knowledge representations that enable efficient aggregation and knowledge distribution. By aligning latent spaces across clients, global knowledge can be integrated while maintaining local customization, mitigating overfitting, and preventing privacy leakage.

3 Method

3.1 Problem Formulation

Federated learning (FL) involves a set of K distributed clients, denoted as $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$. Each client optimizes its local model $\mathcal{M}_k$ by using knowledge derived from other clients' data while not directly accessing the raw data. Each client C_k owns a local dataset $\mathcal{D}_k = \{(x_{k,j}, y_{k,j})\}_{j=1}^{N_k}$, where $x_{k,j}$ and $y_{k,j}$ represent the j^{th} input and label pair. Due to the decentralized nature of FL, the data distributions $\mathcal{P}_i(x, y)$ across clients are often non-IID and imbalanced, which leads to *data heterogeneity*. In addition to data heterogeneity, clients may deploy different local models $\mathcal{M}_k$ to accommodate their computational or application-specific requirements. These variations in architecture, size, and parameter space Θ_k result in *model heterogeneity*, which poses challenges to traditional FL methods that rely on parameter aggregation.

3.2 Framework Overview

In this work, we propose Latent Knowledge Prototypes (FedLKP) framework to address both data and model heterogeneity by leveraging latent prototypes as compact and generalizable representations of client knowledge. Each client maintains a local model $\mathcal{M}_k$ that extracts task-relevant knowledge from its local data. The extracted knowledge is stored in a locally private memory bank, referred to as the knowledge vault $\mathcal{V}_k$. An encoder Q_k^θ, parameterized by θ, processes the knowledge vault to produce latent prototypes. These prototypes are aligned to a joint distribution, such as Gaussian, to ensure compatibility for global aggregation. The prototypes are then transmitted to a central server, where a consensus-aware mechanism aggregates them into a global prototype. This global prototype, resembling a generalized knowledge of the label space, is distributed back to the participating clients. Upon receiving the global prototype, each client uses the decoder P_k^ϕ, parameterized by ϕ, to convert it into task-specific representations compatible with its local model for knowledge distillation.

3.3 Local Latent Prototype Generation

Consider a local client C_k with a local model $\mathcal{M}_k$ consisting of a *feature extractor* $\mathcal{R}_k$ and a *task head* $\mathcal{T}_k$, the feature extractor processes the local training data $\mathcal{D}_k = \{(x_i, y_i)\}_{i=1}^{N_k}$ to extract feature representation $\mathbf{h}_i \in \mathbb{R}^{d_{\text{feat}}}$:

$$\mathbf{h}_i = \mathcal{R}_k(x_i), \quad \forall (x_i, y_i) \in \mathcal{D}_k. \tag{1}$$

These feature representations are organized into label-specific knowledge slots to formulate a knowledge vault $\mathcal{V}_k$, which is a codebook about the local data for each class. Each knowledge slot $\mathcal{V}_k^i$ stores the representations corresponding to class i. It is defined as:

$$\mathcal{V}_k^i = \{\mathbf{h}_j \mid y_j = i\}, \quad i \in \mathcal{Y}_k, \tag{2}$$

where $\mathcal{Y}_k$ denotes the set of labels present in the local dataset $\mathcal{D}_k$. The knowledge vault $\mathcal{V}_k$ has a fixed capacity N for each slot. When the size of $\mathcal{V}_k^i$ exceeds N, earlier appended entries are replaced following the First-In-First-Out (FIFO) strategy to ensure the vault remains consistent. This design accounts for earlier members derived from earlier training epochs, during which the features are less optimized compared to those obtained in later epochs.

$$\mathcal{V}_k^i \leftarrow \text{FIFO}(\mathcal{V}_k^i, \mathbf{h}_j). \tag{3}$$

To formulate a compact latent prototype $\mathbf{z}_k^i$ from the knowledge vault, we design an *encoder* Q_k^θ, parameterized by θ, which takes the label y and the corresponding knowledge slice $\mathcal{V}_k^i$ as input. The encoder is constrained to generate latent representations from the extracted representation in the knowledge vault and also ensure the privacy of the latent prototypes, via means of differential privacy, e.g., by adding Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$. The latent prototype is thus updated as follows:

$$\boldsymbol{\mu}_k^i, \boldsymbol{\sigma}_k^i = Q_k^\theta(i, \mathcal{V}_k^i), \quad \mathbf{z}_k^i = \boldsymbol{\mu}_k^i + \boldsymbol{\sigma}_k^i \odot \boldsymbol{\epsilon}. \tag{4}$$

To reconstruct the latent prototype back into the knowledge space, a decoder P_k^ϕ, parameterized by ϕ, takes $\mathbf{z}_k^i$ as input and outputs the reconstructed knowledge:

$$\hat{\mathcal{V}}_k^i = P_k^\phi(\mathbf{z}_k^i). \tag{5}$$

The training of this encoder-decoder pair is optimized by combining reconstruction and distribution alignment losses:

$$\mathcal{L}_{\text{Latent}} = \mathbb{E}_{q(\mathbf{z}_k^i|i,\mathcal{V}_k^i)}\left[\|\hat{\mathcal{V}}_k^i - \mathcal{V}_k^i\|^2\right] + \text{KL}(q(\mathbf{z}_k^i|i, \mathcal{V}_k^i)\|p(\mathbf{z}_k^i)). \tag{6}$$

Remark: Note that the generated latent representation $\mathbf{z}_c$ is derived directly from local samples and captures the local data characteristics. The alignment of the latent prototype with the normal distribution ensures global compatibility and removes local domain-specific biases. Adding noise during the sampling process enhances privacy by obfuscating sensitive details of local data and reduces the risk of feature inversion attacks. This approach effectively combines privacy preservation with compatibility for global aggregation.

3.4 Consensus-Driven Global Prototype Construction

Once local latent prototypes $\{\mathbf{z}_k^i\}_{k=1}^K$ are generated, all the clients transmit their latent knowledge prototypes as well as their class labels to a central server for aggregation. The server organizes the received latent prototypes into groups based on their corresponding class label i as $\{\mathbf{z}_k^i\}_{k=1}^K$. The server's objective is to construct a *global prototype* $\mathbf{z}_g^i$, which aggregates knowledge in each class group using Mahalanobis-based consensus weighting, proven robust under highly skewed distributions (e.g., Dirichlet-$\alpha = 0.1$). To achieve this consensus, we apply the Mahalanobis distance-based aggregation mechanism [1], which weights client contributions according to their alignment with the central distribution of the latent space. Then, the server computes the center $\boldsymbol{\mu}^i$ and covariance matrix $\boldsymbol{\Sigma}^i$. The Mahalanobis distance measures the deviation of each prototype from the centre:

$$dist_M(\mathbf{z}_k^i, \boldsymbol{\mu}^i) = \sqrt{(\mathbf{z}_k^i - \boldsymbol{\mu}^i)^\top (\boldsymbol{\Sigma}^i)^{-1} (\mathbf{z}_k^i - \boldsymbol{\mu}^i)}. \tag{7}$$

By doing so, we have $dist_M(\mathbf{z}_k^i)$ to indicate how well each prototype aligns with the group consensus. To ensure that the aggregation reflects the contributions of consistent prototypes while mitigating the influence of outliers, the prototypes are re-weighted following the principle that higher weights are assigned to prototypes with smaller distances:

$$w_k^i = \frac{\exp(-dist_M(\mathbf{z}_k^i, \boldsymbol{\mu}^i))}{\sum_{k'=1}^K \exp(-dist_M(\mathbf{z}_{k'}^i, \boldsymbol{\mu}^i))}. \tag{8}$$

The global prototype for label i is then computed as a weighted combination of the local prototypes:

$$\mathbf{z}_g^i = \sum_{k=1}^K w_k^i \cdot \mathbf{z}_k^i. \tag{9}$$

Remark: The global prototypes $\{\mathbf{z}_g^i\}$ represent consensus-aligned and generalizable knowledge of the shared label space, which are contributed by all participating clients. These prototypes are distributed back to the clients for local adaptation. Since they are class-wise aligned and decoder-compatible, they serve as a bridge between diverse model architectures and to ensure cross-client knowledge transfer without requiring parameter alignment.

3.5 Infusing Global Knowledge for Local Adaptation

Once the server constructs the global prototypes $\{\mathbf{z}_g^i\}$, these global prototypes are distributed back to the participating clients. Each client enhances its local model by incorporating global knowledge as a constraint for model optimization. To utilize such global prototypes, each client employs a *latent prototype decoder* P_k^ϕ, parameterized by ϕ. The decoder maps the received global prototypes $\mathbf{z}_g^i$ back into the knowledge space:

$$\hat{\mathcal{V}}_k^i = P_k^\phi(\mathbf{z}_g^i), \tag{10}$$

where $\hat{\mathcal{V}}_k^i$ represents the reconstructed knowledge for class i. The decoder is designed to ensure compatibility with the local model's architecture, so that makes the reconstructed knowledge meets the specific dimensionality requirements of the client. The reconstructed global knowledge $\hat{\mathcal{V}}_k^i$ serves as a prior constraint during the client's local training process. It helps the local model $\mathcal{M}_k$ align its learning with the global knowledge while retaining the flexibility to adapt to the client's unique data characteristics. This is achieved through a knowledge distillation objective where the global prototype acts as a teacher, in addition to the primary task loss. Combining the encoder-decoder loss in Eq. (6) the local model parameters are updated by optimizing the following objectives:

$$\mathcal{L}_{\text{local}} = \mathcal{L}_{\text{task}}(\mathcal{M}_k(x_i), y_i) + \lambda \mathcal{L}_{\text{distill}}(\mathbf{h}_i, \hat{\mathcal{V}}_k^i) + \beta \mathcal{L}_{\text{Latent}}, \tag{11}$$

where $\mathcal{L}_{\text{task}}$ is the primary task loss, such as cross-entropy for classification tasks, and $\mathcal{L}_{\text{distill}}$ is the knowledge distillation loss ensuring alignment between the local model and the global prototype. The term λ balances the importance of the two objectives. This process enables clients to leverage the globally aggregated knowledge to improve generalization and reduce overfitting to local domain-specific data. The integration of global prototypes ensures that clients benefit from a shared understanding of the label space while retaining the flexibility for local adaptations.

Remark: Global prototypes are distributed as encoded representations, inherently preserving privacy by preventing the exposure of feature representations or model parameters. The aggregation process further considers individual client contributions.

3.6 Discussion

Flexibility: The proposed FedLKP framework provides significant flexibility by allowing clients to design their own encoder and decoder architectures to meet their specific computational and task requirements. Despite this flexibility, the framework enforces constraints to ensure interoperability: all encoders must produce latent prototypes $\mathbf{z}_c$ of a fixed dimensionality d_{hid} and align them to a shared distribution (e.g., Gaussian $\mathcal{N}(0, \mathbf{I})$). Similarly, decoders must reconstruct global prototypes $\mathbf{z}_g$ into representations compatible with the client's local model architecture. This balance between flexibility and standardization ensures scalability and generalizability across diverse federated learning settings.

Privacy and Security: The framework incorporates robust privacy and security mechanisms to protect client data throughout the learning process. Latent prototypes are generated as latent representations, so as to prevent direct reconstruction of raw data and ensure privacy by design. During prototype generation, Gaussian noise is injected into the latent representations to enhance differential privacy and obscure sensitive information. Adding Gaussian noise during the sampling process enhances privacy by obfuscating sensitive details in the latent space, following VAE-style variational sampling [7], and discouraging inversion even under strong non-IID conditions. These prototypes are transmitted to the server using encryption, safeguarding them against interception or unauthorized access. Additionally, by aligning latent prototypes to a shared distribution, the framework mitigates risks of feature inversion and ensures compatibility for global aggregation. Unlike certain logits-based methods [18] that require transferring sample-level labels to the server, potentially revealing the data distribution, FedLKP framework utilizes only class-level labels as keys for the prototypes. This design ensures that the data distribution remains protected and avoids unnecessary disclosure of sensitive information.

Communication Cost: The framework significantly reduces communication overhead compared to traditional parameter-based FL methods by exchanging compact latent prototypes rather than full model parameters. For each client, the communication cost of transmitting latent prototypes is proportional to the prototype dimension d and the number of labels $|\mathcal{Y}_c|$, i.e., $O(d_{\text{hid}} \cdot |\mathcal{Y}_c|)$. This is considerably smaller than transmitting high-dimensional features or model weights, making the framework suitable for bandwidth-constrained environments. The reduction in communication costs ensures efficiency, even in scenarios involving a large number of clients.

Computational Overhead: The extra computational complexity of the framework primarily arises from the operations of the encoder and decoder. Encoding involves extracting label-wise latent representation $\mathcal{V}_c$ from the knowledge vault $\mathbf{z}_c$, while decoding involves reconstructing global knowledge representation from the global prototypes. These operations are computationally efficient as it involves only a few Fully Connected layers. Furthermore, the modularity of the framework allows clients to choose encoder and decoder architectures that align with their computational resources, enabling adaptability without compromising efficiency.

4 Experiments

We evaluate the effectiveness of FedLKP through a series of experiments designed to answer the following key questions: (1) How does FedLKP compare with existing personalized FL baselines in terms of model performance across clients? (2) How robust is FedLKP to variations in client data heterogeneity and participation rates? We implement all methods in a common FL simulation framework and conduct experiments on three widely used benchmark datasets under realistic non-IID settings. The results consistently demonstrate that FedLKP achieves superior personalization performance, validating both the theoretical and practical merits of our approach.

4.1 Datasets and Evaluation Metrics

We evaluated FedLKP on three tasks across different domains, including: (1) CIFAR-10 for Computer Vision (CV) classification tasks; (2) COVIDx [16] for medical image classification; and and (3) AG News [25] for Natural Language Processing (NLP) task, *e.g.* news classification. Each dataset was partitioned into N clients, where each client had evenly partitioned but non-IID subsets, in which 75% of the data was used for training and 25% for testing. To reflect the framework's performance on local datasets, we employed the *Averaged Classification Accuracy* (ACA) [13] as the objective metric to evaluate the quantitative performance, which calculates the average accuracy across all clients at the final round.

4.2 Implementation Details

For the CV tasks involving the CIFAR-10 and COVIDx datasets, we utilized a three-layer CNN as the base model. For the natural language processing task involving the AG News dataset, we employed FastText [6] as the base model. All the models are trained for 2000 rounds with a batch size of 10, and the learning rate was set to $5e^{-3}$, optimized by the Adam optimizer. All the clients were selected in each round. We set the number of clients to 10 and 20 in the following experiments, and all the clients were selected in each round. For the latent prototype generation, both the encoder and decoder were implemented as three-layer multilayer perceptrons (MLPs). The balancing parameters α and β in Eq. 11 were both set to 0.1. The encoder transforms representations with dimensions $\{d_{\text{feat}}+C, 128, 64, d_{\text{hid}}\}$, where C is the number of classes, d_{feat} is the feature dimensionality (set to $\mathbb{R}^{512}$), and d_{hid} is the dimensionality of the latent prototypes. Typically, d_{hid} is smaller than d_{feat} to minimize communication overhead, with a default dimensionality of $\mathbb{R}^{32}$ to ensure efficient communication [1]. The decoder reverses this transformation with mirrored architecture. The number of slots in each knowledge vault was empirically set to 500 to balance the storage-vs-performance trade-off, which means it can memorize the 500 most recent features for each class. All the models, including the task model and encoder/decoder were trained for one epoch in each round. The experiments were implemented using the PyTorch-based Personalized Federated Learning Library (PFLib) [24]. The experiments' FL training jobs were simulated and ran on a single NVIDIA A100 GPU.

4.3 Comparative Analysis

We compared FedLKP under different FL heterogeneous settings: one under data heterogeneity and the other under Full (*i.e.* data and model) heterogeneity.

Data-Heterogeneity FL: In this scenario, we assume that all clients share identical model architectures. To simulate highly skewed data distributions, datasets were partitioned into Dirichlet distributions

[1] We experimentally evaluated the influence of d_{feat} across the range {8, 16, 32, 64, 128} and selected 32 as the default value to achieve a balance between communication cost and model performance

Table 2: Averaged classification accuracy (%) comparison across different datasets and methods under **data-heterogeneity setting**. Best results are highlighted in **bold**.

Knowledge	Method	CIFAR-10		COVIDx		AGNews	
		C=10	C=20	C=10	C=20	C=10	C=20
Model	FedAvg [10]	63.78	63.75	78.78	76.01	69.20	76.13
	FedCAC [17]	87.48	88.37	91.89	97.96	96.69	93.91
Classifier	FedPAC [20]	86.46	86.44	79.20	86.70	96.27	92.64
Label Logits	FedDistill [5]	88.22	88.94	91.55	98.02	96.40	94.18
Prototype Feature	FedProto [14]	88.37	88.69	91.60	98.01	95.52	89.99
Latent Prototype	FedLKP (Ours)	**88.52**	**89.60**	**92.28**	**98.02**	**96.96**	**94.22**

Table 3: Averaged classification accuracy (%) comparison across different datasets and methods under **full-heterogeneity setting**. Best results are highlighted in **bold**.

Protocol	Method	COVIDx		Cifar10	
		C=10	C=20	C=10	C=20
Partial	FedGH	91.58	82.68	88.05	88.30
	FedDistill	91.92	97.99	88.06	88.88
	FedLKP (Ours)	**92.20**	**98.00**	**88.32**	**89.45**
Holistic	FedDistill	91.11	97.92	88.17	88.76
	FedLKP (Ours)	**91.78**	**98.09**	**88.46**	**89.48**

with the concentration parameter α set to 0.1, which represents a highly non-IID distribution. Under these conditions, we evaluated the proposed **FedLKP** framework against several existing methods, including: **FedAvg** [10], the traditional model-averaging approach; **FedProto** [14], which addresses data heterogeneity by learning class-level prototypes; **FedDistill** [5], a logits-based method employing federated distillation to mitigate non-IID challenges; **FedCAC** [17], which uses cautious collaboration to balance generalization and personalization; and **FedPAC** [20], a personalized federated learning method that aligns feature spaces and enables classifier collaboration. The performance of these approaches was measured using ACA across various datasets and client configurations. The results, summarized in Table 2, highlight the superior performance of the proposed approach FedLKP in enhancing local model accuracy and robustness under conditions of extreme non-IID data.

Full-Heterogeneity FL: On top of the data-heterogeneous setting, we further introduce varying model architectures across clients in the experiments to highlight challenges posed by data and model heterogeneity [8]. For this setting, we designed two scenarios that simulate two types of model heterogeneity: (1) **Partial**: In this scenario, models across clients have a few layer differences while majority of the layers are the same. (2) **Holistic**: In this scenario, models across clients have different numbers of hidden units and cause the feature dimensionality to be different. For both settings, we evenly grouped the clients into five groups, each with a distinct model architecture among different groups, and within each group, the models were identical. In this setting, only FedGH, FedDistill, and FedLKP are compared, as most of the compared methods in the data-heterogeneous setting could not handle model heterogeneity. The results, summarized in Table 3, demonstrate the robustness of the proposed approach FedLKP to the fully heterogeneous setting. These results further validate our framework's ability to operate under full heterogeneity, where neither model weights nor logits can be easily aligned.

4.4 *Component Analysis*

We analyzed the contributions of individual components in the proposed framework by incrementally integrating its key elements. Specifically, the baseline method corresponds to the FedProto model. The LatentGen variant extends it by introducing latent prototype generation for compact client knowledge. Finally, the full FedLKP model enhances the framework by integrating global consensus-based aggregation to align prototypes across clients.

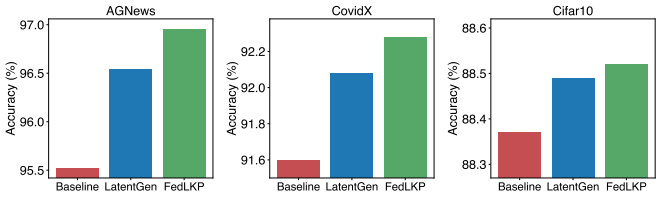

Figure 2: Component Analysis on FedLKP. The baseline refers to the FedProto method, LatentGen involves the latent prototype generation, FedLKP further incorporates consensus-based aggregation.

It can be observed from Fig. 2 that the LatentGen variant consistently outperforms the baseline by leveraging the knowledge vault to store label-specific representations and introducing latent prototypes to encode compact client knowledge. This improvement demonstrates the importance of using latent prototypes to capture local data characteristics while mitigating domain-specific biases. Moreover, when global consensus-based aggregation is introduced in the full FedLKP model, the performance improves further, as the aggregated global prototypes provide a generalized understanding of the label space, and therefore enhancing the robustness of local models to data heterogeneity.

4.5 *Visualizations of Feature Space Clustering*

To further evaluate the effectiveness of the proposed FedLKP framework in aligning and generalizing feature representations, we visualized the feature distribution of the final layer (before the classification layer) with t-SNE [15]. As shown in Fig. 3, the feature distribution produced by the proposed FedLKP framework is significantly more compact and well-clustered compared to the other methods.

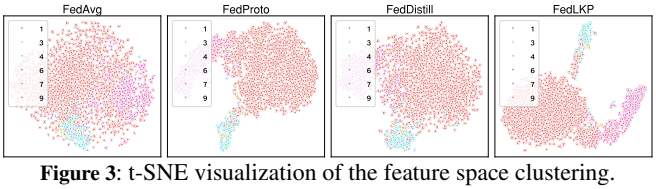

Figure 3: t-SNE visualization of the feature space clustering.

The dense clustering of features within the same class indicates improved intra-class consistency, while the clear separation between clusters highlights better inter-class discrimination. This further validates the framework's design in facilitating effective knowledge sharing while preserving the individuality of local client models.

4.6 Limitations

Despite its effectiveness, the proposed framework has certain limitations that warrant further investigation. First, the global consensus mechanism relies on label information to align latent prototypes across clients. This may restrict the framework's utility in settings where label information is incomplete, inconsistent, or noisy. Furthermore, our current method relies on labeled prototypes, which may limit its applicability in semi-supervised or unsupervised FL. Extending the framework to label-free scenarios via clustering or contrastive learning is a promising future direction.

5 Conclusion and Future Work

In this paper, we proposed the FedLKP framework to effectively address the intertwined challenges of model and data heterogeneity in Federated Learning, while ensuring privacy and efficiency. By introducing latent prototypes generated from the local knowledge vaults as compact representations of client knowledge, FedLKP enables robust knowledge sharing without requiring access to raw data or full model parameters. FedLKP allows clients to employ their own encoder and decoder architectures to fit the extracted features while enabling adaptation to diverse computational and task-specific requirements. Meanwhile, latent prototypes ensure privacy through an additional layer of encoding and differential protection, allowing global aggregation while maintaining local data privacy. FedLKP is designed to significantly reduce communication overhead by transmitting customized compact latent prototypes rather than full model parameters while maintaining better generalization and performance. Extensive experiments on diverse benchmarks demonstrate, under various heterogeneity settings, the superior performance of our framework in terms of accuracy and effectiveness.

One future direction is to develop label-agnostic aggregation methods that rely on latent feature similarities rather than explicit labels. Second, the framework assumes static client participation and relatively stable data distributions during training. However, clients often join or leave dynamically in real-world federated learning environments, and their local data distributions may evolve over time. Such dynamics can disrupt the consistency of global prototype aggregation and the effectiveness of global-to-local knowledge transfer. Future work could focus on designing adaptive mechanisms that account for client heterogeneity with dynamic data shifts.

Acknowledgments

This work is supported by UKRI-EPSRC GRANT Reference EP/X035085/1. The research presented in this paper was conducted while Qilei Li was with Queen Mary University of London, UK.

References

[1] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart. The mahalanobis distance. *Chemometrics and intelligent laboratory systems*, 50 (1):1–18, 2000.

[2] J. Du, H. Wang, J. Li, K. Wang, and R. Fei. Hfedcwa: heterogeneous federated learning algorithm based on contribution-weighted aggregation. *Applied Intelligence*, 55(1):186, 2024.

[3] B. Fan, S. Jiang, X. Su, S. Tarkoma, and P. Hui. A survey on model-heterogeneous federated learning: Problems, methods, and prospects. In *2024 IEEE International Conference on Big Data (BigData)*, pages 7725–7734. IEEE, 2024.

[4] B. Hitaj, G. Ateniese, and F. Perez-Cruz. Deep models under the gan: Information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 603–618, 2017.

[5] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. In *Advances in Neural Information Processing Systems*, 2018.

[6] A. Joulin, É. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, 2017.

[7] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[8] T. Lin, L. Kong, S. Stich, and M. Jaggi. Federated distillation and augmentation for communication-efficient federated learning. In *Advances in Neural Information Processing Systems*, pages 7592–7603, 2020.

[9] Z. Lin, R. Guo, K. Zhang, M. Li, F. Yang, S. Xu, D. Liu, and A. Abubakar. Feature-based inversion using variational autoencoder for electrical impedance tomography. *IEEE Transactions on Instrumentation and Measurement*, 71:1–12, 2022.

[10] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[11] Q. Pan, S. Sun, Z. Wu, Y. Wang, M. Liu, B. Gao, and J. Wang. Fedcache 2.0: Federated edge learning with knowledge caching and dataset distillation. *Authorea Preprints*, 2024.

[12] G. Saman Nariman and H. K. Hamarashid. Communication overhead reduction in federated learning: a review. *International Journal of Data Science and Analytics*, 2024.

[13] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar. Personalized federated learning: An optimization perspective. In *Advances in Neural Information Processing Systems*, volume 2017, pages 66–78, 2017.

[14] Y. Tan, G. Long, L. Liu, T. Zhou, Q. Lu, J. Jiang, and C. Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[15] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[16] L. Wang and A. Wong. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *Scientific Reports*, 10:19549, 2020.

[17] X. Wu, X. Liu, J. Niu, G. Zhu, and S. Tang. Bold but cautious: Unlocking the potential of personalized federated learning through cautiously aggressive collaboration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19375–19384, 2023.

[18] Z. Wu, S. Sun, Y. Wang, M. Liu, K. Xu, W. Wang, X. Jiang, B. Gao, and J. Lu. Fedcache: A knowledge cache-driven federated learning architecture for personalized edge intelligence. *IEEE Transactions on Mobile Computing*, 2024.

[19] D. Xiao, D. Yang, J. Li, X. Chen, and W. Wu. Privacy leakage from logits attack and its defense in federated distillation. In *2024 54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 169–182. IEEE, 2024.

[20] J. Xu, X. Tong, and S.-L. Huang. Personalized federated learning with feature alignment and classifier collaboration. In *The Eleventh International Conference on Learning Representations*, 2023.

[21] J. Yang, T. Baker, S. S. Gill, X. Yang, W. Han, and Y. Li. A federated learning attack method based on edge collaboration via cloud. *Software: Practice and Experience*, 54(7):1257–1274, 2024.

[22] M. Ye, X. Fang, B. Du, P. C. Yuen, and D. Tao. Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Computing Surveys*, 56(3):1–44, 2023.

[23] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, 2021.

[24] J. Zhang, Y. Liu, Y. Hua, H. Wang, T. Song, Z. Xue, R. Ma, and J. Cao. Pfllib: Personalized federated learning algorithm library. *arXiv preprint arXiv:2312.04992*, 2023.

[25] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems*, pages 649–657, 2015.

[26] X. Zhang, W. Yin, M. Hong, and T. Chen. Fedadkd: heterogeneous federated learning via adaptive knowledge distillation. *Pattern Analysis and Applications*, 27(1):1350, 2024.