Rethinking Client Knowledge for Federated Learning: An Entangled Representation Perspective

Anonymous Author(s)
Affiliation
Address
email

Abstract

Federated learning (FL) enables collaborative learning across clients without compromising privacy. Most existing FL studies focus on model-homogeneous scenarios. However, ensuring uniform model architecture across clients is challenging, leading to a *model-heterogeneous FL* problem. To address this problem, we rethink client knowledge and design a novel form termed the *entangled representation*, which entangles all representations from distinct categories on each client into a single representation. Building on this concept, we propose a Federated Representation Entanglement (FedRE), which first synthesizes **a single entangled representation on each client** and then uploads it to the server to train the global classifier. Each entangled representation integrates information from distinct categories and is categorized into multiple categories according to their respective contributions, enabling the global classifier to learn a decision boundary that captures inter-category relationships. As a result, the entangled representations offer effective, privacy-preserving, and lightweight client knowledge. Theoretically, we analyze the convergence of FedRE, and empirically, we demonstrate its superiority in balancing model performance, privacy protection, and communication overhead. The codes are available at `https://anonymous.4open.science/r/FedRE-x`.

1 Introduction

Federated learning (FL) [26, 43] is a collaborative learning paradigm that aggregates privacy-preserving client knowledge (*e.g.*, model parameters) from multiple clients. Numerous FL methods have been developed and applied in various fields, such as healthcare [1, 48] and the Internet of Things [27, 7]. Most existing FL studies [26, 52, 24, 20, 4, 3, 49] assume that the architectures of local models across clients are homogeneous. In practice, however, assuming the same model architecture for all clients is unrealistic due to differences in sample distribution, hardware, and computational capabilities. Also, the model architecture adopted by each client is private and may not be shared with the server or other clients. Those issues motivate a practical but challenging problem known as *model-heterogeneous FL* [44], where the representation extractors adopt heterogeneous architectures across clients, while the classifiers share a homogeneous architecture.

To address this problem, various model-heterogeneous FL methods[18, 12, 25, 36, 22, 45, 11, 39, 46, 41] have been proposed. Most of them focus on aggregating the representations [25], logits [12], small-models [46, 41], classifiers [22], or prototypes (*i.e.*, category means) [36, 45, 11, 39], from clients. On the one hand, although representations, logits, and small-models encode high-level client knowledge, uploading them to the server incurs significant communication overhead and exposes a serious privacy risk, as they can be exploited to infer the original samples by launching representation and model inversion attacks [37, 47]. On the other hand, uploading classifiers and prototypes that encapsulate category-specific knowledge to the server reduces both communication overhead and

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

the risk of inferring original samples. However, they may fail to model inter-category relationships, leading to suboptimal model performance. This raises a question: "*For model-heterogeneous FL, is there more effective, privacy-preserving, and lightweight client knowledge available?*"

To find a potential solution, we observe that prototypes effectively reduce communication overhead and privacy risks. However, prototypes aggregate same-category representations into a single representation, failing to capture inter-category relationships. *This inspires us to entangle representations from distinct categories on each client into a single representation to alleviate this issue*. Unlike prototypes, the *entangled representation* integrates knowledge from multiple categories with varying contributions. Building on this concept, we propose a Federated Representation Entanglement (FedRE). In FedRE, each client first entangles all representations from distinct categories into a single representation and then uploads it to the server for training the global classifier. During training, each entangled representation is simultaneously classified into multiple categories based on the corresponding category weights, enabling the global classifier to learn a decision boundary that effectively captures the relationships of different categories. Furthermore, the entangled representations can effectively resist representation inversion attacks, as they do not correspond to any single sample, and significantly reduce communication overhead by uploading only a single representation per client. As a result, those entangled representations provide privacy-preserving, lightweight, and effective client knowledge.

The main contributions of this paper are three-fold. **(1)** To the best of our knowledge, we are the first to utilize the entangled representation as client knowledge to train the global classifier. **(2)** We propose FedRE based on the entangled representation and analyze its theoretical convergence. **(3)** Extensive experimental results on three benchmark datasets verify the superiority of FedRE in balancing model performance, privacy protection, and communication overhead compared to state-of-the-art methods.

2 Related Work

Existing FL methods can be roughly categorized into model-heterogeneous and model-homogeneous approaches based on their ability to handle model heterogeneity.

Model-heterogeneous FL approach handles both heterogeneous local models and sample distributions. Due to the heterogeneity of local models, it is not feasible to aggregate all their parameters. Thus, most of the studies turns to aggregate client knowledge (*e.g.*, representations [25], logits [12], small-models [46, 41], classifiers [22], or prototypes [36, 45, 11, 39]). For example, DS-FL [12] designs a logit aggregation strategy that integrates the local logits from different clients into the global logit on the server. FedHeNN [25] aligns the representations extracted by distinct local models using a common representation alignment dataset. LG-FedAvg [22] aggregates the classifiers from distinct clients on the server. FedProto [36], FPL [11], FedGH [45], and FedTGP [53] treat prototypes as a form of client knowledge. Specifically, FedProto [36] averages the local prototypes of each category to aid in learning local representation extractors. FPL employs a clustering strategy to derive unbiased global prototypes, and FedTGP optimizes trainable global prototypes dynamically. Moreover, FedGH [45] utilizes the prototypes from multiple clients to train the global classifier on the server. Additionally, several studies [55, 42] focus on knowledge distillation. For instance, FedGen [55] learns a global generator to augment the training samples for local models, while FedKD [42] distills a global student model to assist the learning of local models. Furthermore, another line of research [41, 46] uses homogeneous small-models as client knowledge. A recent example is FedMRL [46], which facilitates inter-client knowledge aggregation via a shared small-model.

Model-homogeneous FL approach deals with homogeneous local models but heterogeneous sample distributions. Most of the studies [26, 52, 24, 20, 4, 3, 40] focus on aggregating all parameters of local models. For instance, FedAvg [26] aggregates all parameters of local models from distinct clients on the server. Based on FedAvg, FedAvgDBE [51], FedFN [14], and FedDecorr [33] alleviates the representation bias issue in local models. Another example is FedALA [52], which adaptively integrates the global and local models to align with the local objective. In addition, several studies [34, 2, 28, 29] aim to aggregate partial parameters of local models. For example, FedRep [2] aggregates the representation extractors of local models to enhance representation capability. Moreover, FedBABU [28], SphereFed [6], FedETF [21], and FedDr+ [13] only updates the representation extractors during local training and then aggregates them on the server.

3 Problem Formulations

Model-heterogeneous FL. Following [36, 45], in the model-heterogeneous FL problem, we are given K clients and a server. Let $\mathcal{D}_k = \{(\mathbf{x}_i^k, \mathbf{y}_i^k)\}_{i=1}^{n_k}$ be a private local dataset in the k-th client, where $\mathbf{x}_i^k$ represents the i-th sample in $\mathcal{D}_k$, and $\mathbf{y}_i^k$ denotes its associated one-hot label over C categories. Moreover, let $h_k(\boldsymbol{\theta}_k; \cdot) = \boldsymbol{g}_k(\boldsymbol{\phi}_k; \cdot) \circ f_k(\boldsymbol{\omega}_k; \cdot)$ denote the local model with parameters $\boldsymbol{\theta}_k = \{\boldsymbol{\phi}_k, \boldsymbol{\omega}_k\}$ in the k-th client, where $\boldsymbol{g}_k(\boldsymbol{\phi}_k; \cdot)$ represents the representation extractor with parameters $\boldsymbol{\phi}_k$, and $f_k(\boldsymbol{\omega}_k; \cdot)$ denotes the local classifier with parameters $\boldsymbol{\omega}_k$. There exist distinct clients i and j such that $\boldsymbol{g}_i(\boldsymbol{\phi}_i; \cdot)$ and $\boldsymbol{g}_j(\boldsymbol{\phi}_j; \cdot)$ have heterogeneous architectures, while $f_i(\boldsymbol{\omega}_i; \cdot)$ and $f_j(\boldsymbol{\omega}_j; \cdot)$ maintain a homogeneous architecture across all clients. The goal is to learn a model using $\{\mathcal{D}_k\}_{k=1}^K$ to achieve optimal average classification accuracy across all clients.

Threat Model. We assume the server is *semi-honest*: It follows the protocols *honestly but is curious* to infer original samples from the representations by launching a representation inversion attack [37]. Furthermore, we assume the server can *illegally* access the representation extractors of clients.

4 Methodology

4.1 Motivation

In the model-heterogeneous FL problem, the global classifier often struggles to be effectively trained due to the heterogeneity of both local models and sample distributions across clients. To address this challenge, a promising solution is to aggregate client knowledge to train a high-quality global classifier on the server, which is then deployed to clients to replace their local classifiers. A vanilla approach, FedAllRep, uploads all ***sample representations*** to the server as client knowledge for training the global classifier (as illustrated in the left of Figure 1). While FedAllRep achieves superior model performance, it poses significant risks of leaking original samples through representation inversion attacks [37] and incurs substantial communication overhead. To alleviate this issue, FedGH [45] uses ***prototypes*** as client knowledge to train the global classifier, enhancing privacy protection and reducing communication overhead (as illustrated in the middle of Figure 1). However, those prototypes only encode the representative knowledge of their respective categories, failing to model inter-category relationships, resulting in suboptimal model performance.

To mitigate this drawback, we observe that prototypes are constructed solely from same-category representations, which limits their ability to capture inter-category relationships. This observation motivates the design of ***entangled representation***—a novel form of client knowledge that entangles representations from different categories (as illustrated in the right of Figure 1)—thereby incorporating multi-category information and capturing inter-category dependencies. Specifically, we first entangle all representations from distinct categories on each client into a single representation, where each

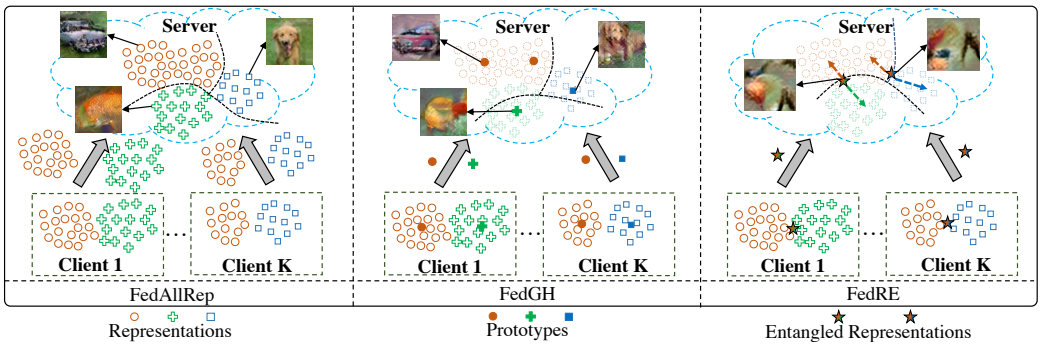


Figure 1: Illustrations of FedAllRep, FedGH, and FedRE. Here, different shapes represent distinct categories, dotted shapes indicate the absence of uploads, solid arrows (→) point to reconstructed images obtained via a representation inversion attack, and dashed arrows (⇢) indicate the direction in which representations are pulled toward different categories. In FedAllRep, vanilla representations are uploaded, ensuring model performance but increasing communication overhead and privacy risks. In FedGH, prototypes are uploaded, reducing communication overhead and privacy risks, while lowering model performance. In FedRE, entangled representations are uploaded, balancing model performance, privacy protection, and communication overhead.

category contributes in a random proportion. Then, each entangled representation is uploaded to the server to train the global classifier by *simultaneously categorizing it into multiple categories* (see dash arrows in the right of Figure 1). As a result, the global classifier learns a decision boundary that captures the relationships of multiple categories, improving generalizability. Thus, classifying only a few entangled representations can enable effective generalization to vanilla representations. Besides, the entangled representation provides enhanced privacy protection since it cannot be mapped to any individual sample. Furthermore, uploading only an entangled representation per client significantly reduces communication overhead.

In summary, the entangled representations provide effective, privacy-preserving, and lightweight client knowledge, forming the FedRE's foundation. Next, we detail the FedRE.

4.2 FedRE

We begin by outlining the workflow of FedRE, as depicted in Figure 2. Each client has a local model comprising a representation extractor and a classifier. The process initiates with the calculation of an entangled representation per client through a simple *Representation Entanglement* (RE) mechanism. Subsequently, those entangled representations, generated from multiple clients, are uploaded to the server for training the global classifier. Next, we detail the update process of FedRE, which involves three main steps: (i) local model update; (ii) representation entanglement and upload; and (iii) global classifier update and broadcast.

Local Model Update. Similar to vanilla federated learning approaches such as FedAvg [26], FedRE requires each client to update its local model to effectively learn from local samples. To accomplish this, the optimization objective for the k-th client is formulated by

$$\min_{\boldsymbol{\theta}_k} \frac{1}{n_k} \sum_{(\mathbf{x}_i^k, \mathbf{y}_i^k) \in \mathcal{D}_k} \mathcal{L}_{ce}\big[h_k(\boldsymbol{\theta}_k; \mathbf{x}_i^k), \mathbf{y}_i^k\big], \quad (1)$$

where $\mathcal{L}_{ce}(\cdot, \cdot)$ denotes the cross-entropy loss.

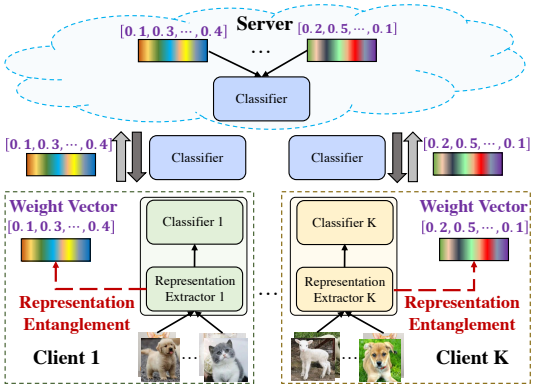


Figure 2: FedRE workflow. Each client has a local model including a representation extractor and a classifier. All representations are entangled into a single representation using a random normalized weight vector on each client, and both are then uploaded to the server to train the global classifier.

Representation Entanglement and Upload. We now introduce how to calculate a single entangled representation on each client. First, to address the heterogeneity of representations across clients with varying model architectures, we apply a simple average pooling $\mathrm{AP}(\cdot)$ [8] operation to map all client representations into a unified space, ensuring consistency and enabling the use of a shared global classifier. This approach allows for effective representation aggregation despite architectural differences, facilitating the collaborative learning of heterogeneous models (see Section 5.3 for an analysis of other representation mapping operations). Then, we calculate the entangled representation as

$$\widetilde{\mathbf{r}}_k = \sum_{c \in \mathcal{C}_k} \frac{w_k^c}{|\mathcal{D}_k^c|} \sum_{(\mathbf{x}_i^k, \mathbf{y}_i^k) \in \mathcal{D}_k^c} \mathrm{AP}\big[\boldsymbol{g}_k(\boldsymbol{\phi}_k; \mathbf{x}_i^k)\big], \quad (2)$$

where $\mathcal{D}_k^c$ be the set of samples belonging to category c in the k-th client, $\mathcal{C}_k$ be the label set of the k-th client, and $\mathbf{w}_k = [w_k^1, \cdots, w_k^{|\mathcal{C}_k|}]$ is a normalized weight vector, where the elements are randomly drawn from a uniform distribution $\mathcal{U}(0, 1)$, and then normalized by dividing each element by the sum of all elements to ensure their sum equals one. Applying random entanglement weights for each category ensures diverse entangled representations, helping the global classifier learn more generalizable decision boundaries. *Besides this RE method, we also detail other methods in Section 5.3.* The weight vector indicates the probability distribution of each entangled representation across different categories, which can be interpreted as its label encoding. The specific usage is detailed below.

Global Classifier Update and Broadcast. Upon receiving the entangled representations and their corresponding weight vectors $\widetilde{\mathcal{R}} = \{(\widetilde{\mathbf{r}}_k, \mathbf{w}_k)\}_{k=1}^K$, the server utilizes those entangled representations and their associated weight vectors to update the global classifier. Accordingly, the server's optimization objective is formulated as

$$\min_{\boldsymbol{\omega}} \sum_{k=1}^{K} \sum_{c \in \mathcal{C}_k} w_k^c \mathcal{L}_{ce}\big[f(\boldsymbol{\omega}; \widetilde{\mathbf{r}}_k), \mathbf{y}_c\big], \tag{3}$$

where $f(\boldsymbol{\omega}; \cdot)$ represents the global classifier with parameters of $\boldsymbol{\omega}$, $\mathbf{y}_c$ stands for the one-hot label associated with category c. By minimizing Eq. (3), we learn a global classifier based on the entangled representations, which can effectively categorize all training samples from various clients. Finally, the server broadcasts the updated global classifier to all clients for the next iteration.

With the above update process, the FedRE method can be summarized in Algorithm 1.

Algorithm 1 FedRE

Input: K clients with their respective datasets $\{\mathcal{D}_k\}_{k=1}^K$.
Output: Local models for all clients, *i.e.*, $\{h_k(\boldsymbol{\theta}_k; \cdot)\}_{k=1}^K$.

```
Randomly initialize the global classifier f(ω; ·) and the local models {h_k(θ_k; ·)}_{k=1}^K.
for t = 0 to T − 1 do
    for each client k in parallel do                                   ▷ Client Side
        Receive ω to update ω_k and update θ_k according to Eq. (1).
        Randomly generate and normalize w_k.
        Calculate r̃_k according to Eq. (2) and upload (r̃_k, w_k) to the server.
    end for
    Update ω according to Eq. (3) and broadcast ω to all clients.      ▷ Server Side
end for
```

4.3 RE vs. Mixup

We now compare RE and mixup [50]. Mixup is a popular data augmentation technique that aims to linearly interpolate two representations and their corresponding labels to synthesize a new representation. Let $\mathcal{R} = \{(\mathbf{r}_i, \mathbf{y}_i)\}_{i=1}^n$ denote the set of representations on a single client, where $\mathbf{r}_i$ represents the representation of the i-th sample, and $\mathbf{y}_i$ denotes its associated one-hot label over C categories. Recall that mixup is formulated as follows:

$$\widetilde{\mathbf{r}}_{\text{mixup}} = \lambda \mathbf{r}_i + (1-\lambda)\mathbf{r}_j, \widetilde{\mathbf{y}}_{\text{mixup}} = \lambda \mathbf{y}_i + (1-\lambda)\mathbf{y}_j, \tag{4}$$

where $\lambda \sim \text{Beta}(\alpha, \alpha)$, for $\alpha \in (0, \infty)$. In contrast, RE, specifically designed for FL, aims to ensure model performance while reducing communication costs and privacy risks. Its general form is formulated as follows:

$$\widetilde{\mathbf{r}}_{\text{RE}} = \sum_{i=1}^{n} w_i \mathbf{r}_i, \widetilde{\mathbf{y}}_{\text{RE}} = \sum_{i=1}^{n} w_i \mathbf{y}_i, \tag{5}$$

where $w_i \in [0, 1]$ is the weight of $\mathbf{r}_i$ and can be determined by various RE methods (see details in Appendix B). As indicated by Eqs. (4)–(5), mixup performs linear interpolation between pairs of representations, while RE entangles the entire set of representations from each client using a weighted sum. Furthermore, RE trains the classifier by calculating a weighted sum of losses across multiple categories, with the weights determined by their contributions.

4.4 Convergence Analysis

We present the convergence analysis of FedRE in the model-heterogeneous setting, focusing on the convergence conditions and rate for the local objective of an arbitrary client. Let $t \in \{0, \ldots, T-1\}$ be the global communication round and $e \in \{0, 1, \ldots, E\}$ be the local iteration. At the start of the $(t+1)$-th round, denoted as $(tE+0)$, each client replaces the local classifier with the global one trained in the t-th round. The first iteration of the $(t+1)$-th round is $(tE+1)$, and $(tE+E)$ marks the final iteration of that round. Our convergence analysis is based on the following assumptions, which are similar to those used in existing FL studies [45, 36].

Assumption 1. *Each local objective function is L_1-Lipschitz smooth, i.e., there exists a constant $L_1 > 0$ such that $\|\nabla \mathcal{L}\left(\boldsymbol{\theta}_k^{t_1}; \mathcal{D}_k\right) - \nabla \mathcal{L}\left(\boldsymbol{\theta}_k^{t_2}; \mathcal{D}_k\right)\|_2 \leq L_1 \left\|\boldsymbol{\theta}_k^{t_1} - \boldsymbol{\theta}_k^{t_2}\right\|_2$, $\forall t_1, t_2 > 0$, $k \in \{1, \cdots, K\}$.*

Assumption 2. *The stochastic gradient (i.e., mini-batch gradient) on each client is unbiased, i.e.,* $\mathbb{E}_{\mathcal{B}_k \subseteq \mathcal{D}_k}\left[\nabla \mathcal{L}\left(\boldsymbol{\theta}_k; \mathcal{B}_k\right)\right] = \nabla \mathcal{L}\left(\boldsymbol{\theta}_k; \mathcal{D}_k\right)$*, and its variance is bounded by a constant* σ^2*, i.e.,* $\mathbb{E}_{\mathcal{B}_k \subseteq \mathcal{D}_k}\left[\left\|\nabla \mathcal{L}\left(\boldsymbol{\theta}_k; \mathcal{B}_k\right) - \nabla \mathcal{L}\left(\boldsymbol{\theta}_k; \mathcal{D}_k\right)\right\|_2^2\right] \leq \sigma^2$*,* $k \in \{1, \cdots, K\}$.

Assumption 3. *The divergence between the local and global classifiers is bounded by two constants* ϵ^2 *and* δ^2*, i.e.,* $\|\boldsymbol{\omega}_k^{t+1} - \boldsymbol{\omega}^t\|_2^2 \leq \epsilon^2$*,* $\|\boldsymbol{\omega}_k^t - \boldsymbol{\omega}^t\|_2^2 \leq \delta^2$*,* $\forall t > 0$*,* $k \in \{1, \cdots, K\}$.

Assumption 1 ensures that the true gradient (*i.e.*, full-batch gradient) of the local objective function remains stable. Assumption 2 ensures the expectation of the stochastic gradient matches the true gradient and limits excessive variations. Assumption 3 ensures that the divergence between the local and global classifiers in consecutive rounds remains small. Based on them, we first establish the convergence conditions for FedRE in Theorem 1 and then derive its convergence rate in Theorem 2.

Theorem 1. *Under Assumptions 1 to 3 and the local learning rate satisfying* $\eta_l^{e'} < \frac{2\sum_{e=0}^{e'}\|\nabla L_{tE+e}\|_2^2}{L_1 E\sigma^2 + L_1\sum_{e=0}^{e'}\|\nabla L_{tE+e}\|_2^2}$*,* $e' \in \{0, 1, \ldots E-1\}$*, the local objective functions convergence.*

Theorem 2. *Under Assumptions 1 to 3, let* $\Delta = \mathcal{L}_0 - \mathcal{L}^*$ *where* $\mathcal{L}_0$ *and* $\mathcal{L}^*$ *denote the initial and optimal values of the local objective function, respectively. For an arbitrary client, given any* $\xi > 0$*, when* $T > \frac{2\Delta}{\xi E(2\eta_l - L_1\eta_l^2) - L_1 E\eta_l^2\sigma^2 - L_1\delta^2}$*, and the local learning rate satisfying* $\eta_l < \frac{2\xi}{\xi L_1 + L_1\sigma^2}$*, we have* $\frac{1}{TE}\sum_{t=0}^{T-1}\sum_{e=0}^{E-1}\mathbb{E}\left\|\nabla L_{tE+e}\right\|_2^2 \leq \xi$.

The proofs of Theorems 1 and 2 are provided in Appendix C.

5 Experiments

5.1 Experimental Setup

Datasets and Baselines. We adopt three benchmark datasets: CIFAR-10 [15], CIFAR-100 [15], and TinyImageNet [16]. Moreover, we compare FedRE with seven state-of-the-art approaches: LG-FedAvg [22], FedGH [45], FedKD [42], FedGen [55], FedProto [36], FPL [11], FedTGP [53], as well as a Local method that trains local models independently on each client without communication.

Evaluation Metrics. We evaluate model performance by calculating the classification accuracy on the test set across all clients. To ensure a fair comparison, we report the average classification accuracy of the final round after 100 rounds, calculated across three random experiments, along with its standard deviation. Communication overhead is measured by the total number of parameters uploaded and broadcast per round, respectively. To evaluate privacy protection, we adopt Peak Signal-to-Noise Ratio (PSNR) [31] and Mean Squared Error (MSE), which measure the fidelity of reconstructed images. Low PSNR and high MSE indicate substantial reconstruction errors, suggesting that the original image is difficult to recover and thus privacy is well preserved. For privacy protection, we adopt Peak Signal-to-Noise Ratio (PSNR) [31] and Mean Squared Error (MSE) as evaluation metrics. Those metrics reflect the quality of reconstructed images, with lower PSNR and higher MSE indicating greater distortion and, thus, stronger privacy protection.

Model-Heterogeneous settings. We configure 10 clients with 10 distinct model architectures: a 4-layer CNN from [51], MobileNetV2 [30], GoogleNet [35], five ResNet models (ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152) [10], and two Vision Transformer (ViT) models (ViT-B/16 and ViT-B/32) [9].

Statistic-Heterogeneous settings. We follow [51] to adopt both practical (PRA) [23, 19] and pathological (PAT) [32] settings. In the PRA setting, samples are distributed across clients using a Dirichlet distribution [23] with a parameter α, which is set to 0.1 by default across all datasets. In the PAT setting, each client is assigned samples from 2, 10, and 20 categories, drawn from a total of 10, 100, and 200 categories in CIFAR-10, CIFAR-100, and Tiny-ImageNet, respectively, with varying sample sizes. More experimental details can be found in Appendix D.

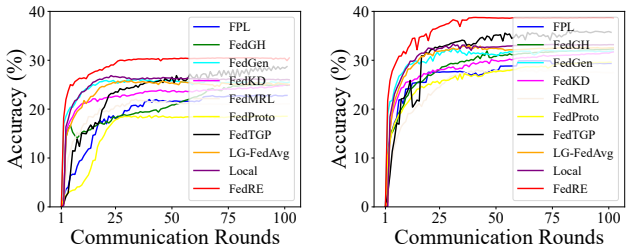
5.2 Main Experiments

Q1: How does FedRE perform in the model-heterogeneous setting across different statistical heterogeneity scenarios? The results under the model-heterogeneous FL setting are listed in

Table 1: Accuracy (%) comparison on three datasets under the model-heterogeneous setting. In each column, the best results are **bolded**, and the second-best results are underlined.

Method	PRA			PAT			Average
	CIFAR-10	CIFAR-100	TinyImageNet	CIFAR-10	CIFAR-100	TinyImageNet	
LG-FedAvg [22]	80.90 ± 0.17	41.96 ± 0.03	25.16 ± 0.42	85.35 ± 0.25	58.24 ± 0.33	32.26 ± 0.28	53.98
FedGH [45]	78.66 ± 0.34	40.91 ± 0.26	25.04 ± 0.11	85.43 ± 0.03	58.07 ± 0.33	31.98 ± 0.29	53.35
FedKD [42]	80.79 ± 0.38	41.33 ± 0.25	25.39 ± 0.36	84.03 ± 0.17	55.61 ± 0.10	31.73 ± 0.20	53.15
FedGen [55]	81.16 ± 0.12	41.46 ± 0.10	25.45 ± 0.19	84.88 ± 0.18	57.87 ± 0.67	31.96 ± 0.21	53.80
FedProto [36]	78.36 ± 0.52	35.00 ± 0.34	18.16 ± 0.08	83.81 ± 0.18	56.72 ± 0.11	29.61 ± 0.02	50.28
FPL [11]	77.40 ± 0.07	36.66 ± 0.30	22.64 ± 0.34	83.89 ± 0.20	53.21 ± 0.09	29.16 ± 0.05	50.49
FedMRL [46]	81.28 ± 0.02	34.41 ± 0.03	20.92 ± 0.05	83.30 ± 0.01	54.25 ± 0.14	27.37 ± 0.01	50.26
FedTGP [53]	81.32 ± 0.33	35.89 ± 0.07	28.70 ± 0.01	84.68 ± 0.11	54.67 ± 0.24	35.64 ± 0.06	53.48
Local	81.20 ± 0.05	41.57 ± 0.10	25.81 ± 0.15	84.68 ± 0.07	57.96 ± 0.12	33.02 ± 0.14	54.04
FedRE	$\mathbf{82.60 \pm 0.01}$	$\mathbf{46.36 \pm 0.09}$	$\mathbf{30.48 \pm 0.13}$	$\mathbf{86.20 \pm 0.14}$	$\mathbf{62.56 \pm 0.32}$	$\mathbf{38.52 \pm 0.08}$	**57.12**

Table 1. We have several insightful observations. (1) FedRE substantially outperforms all the baselines across various scenarios. In particular, FedRE achieves an accuracy on TinyImageNet that surpasses those of LG-FedAvg, FedGH, and FedKD by 6.26%, 6.54%, and 6.79%, respectively, under the PAT setting. Also, several methods do not exceed the performance of Local, indicating that the scenarios are challenging. (2) FedRE outperforms FedGH, suggesting that using entangled representations to train the global classifier is more effective than using prototypes. One reason is that entangled representations incorporate information from multiple categories, whereas prototypes do not. (3) LG-FedAvg is worse than FedRE, which indicates that directly aggregating local classifiers from clients to form the global classifier is less effective than carefully optimizing the global classifier using entangled representations. Also, Figures 3(a)-(b) show that the accuracy of FedRE increases initially and then stabilizes on the TinyImageNet dataset, consistently outperforming other baselines during training, suggesting stable performance convergence. Moreover, we evaluate FedRE under various statistical-heterogeneous settings in **Q8** of Section 5.3, including large-scale participation (*i.e.*, 100 clients).



(a) PRA (b) PAT

Figure 3: Accuracy (%) comparison between distinct communication rounds on the TinyImageNet dataset in the model-heterogeneous setting.

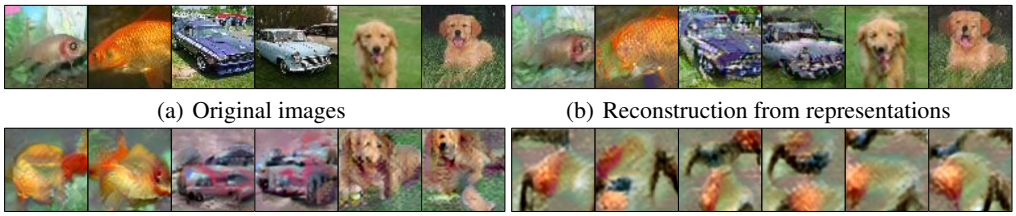
Table 2: Communication overhead (# params $\times 10^3$) comparison on the CIFAR-100 dataset. In each row, the best results are **bolded**, and the second-best results are underlined.

Metric	LG-FedAvg	FedGH	FedKD	FedGen	FedProto	FedMRL	FedTGP	FPL	FedRE
Upload	513.00	257.02	4234.28	9247.08	257.02	8863.08	257.02	257.02	**5.12**
Broadcast	513.00	**512.00**	4234.28	513.00	**512.00**	8863.08	**512.00**	916.48	513.00

Q2: What is the communication overhead of FedRE? We conduct communication overhead experiments on the CIFAR-100 dataset in the model-heterogeneous settings. As shown in Table 2, FedRE achieves the lowest communication overhead during the upload phase, as it uploads only a single entangled representation from each client to the server. During the broadcast phase, its overhead is comparable to that of classifier-based methods (e.g., LG-FedAvg) and prototype-based methods (e.g., FedProto). More results are offered in Appendix F.1

Q3: Can entangled representations effectively resist privacy attacks? We adhere to the setting stated in the Threat Model section (Section 3) to reconstruct the original samples from entangled representations. Also, we compare the resilience of both *representations* and *prototypes*. Figure 4 depicts the reconstruction results on the TinyImageNet datasets (More results are offered in Appendix F.2). We can make several key observations: (1) Most image contours are reconstructed from the representations, indicating their vulnerability to privacy attacks. (2) Some category information, such as the presence of a *fish*, is leaked through reconstructed prototypes, as prototypes encapsulate representative category information. (3) The reconstructed images from entangled representations reveal no identifiable information, demonstrating their effectiveness against privacy attacks. In addition, the PSNR values for images reconstructed from representations, prototypes, and entangled

representations are 12.89, 10.25, and 9.66, with corresponding MSE values of 4514.91, 6992.04, and 7781.87. These results indicate that entangled representations yield the lowest PSNR and highest MSE, suggesting stronger privacy protection. Overall, all results demonstrate the superior privacy protection provided by entangled representations.



(a) Original images (b) Reconstruction from representations

(c) Reconstruction from prototypes (d) Reconstruction from entangled representations

Figure 4: Privacy protection comparison between representations, prototypes, and entangled representations on the TinyImageNet dataset.

5.3 Analysis

Q4: What are the advantages of uploading a single entangled representation per client compared to uploading all representations?

To explore the benefits of entangled representations, we compare FedRE with FedAllRep, which uses all clients' representations to train the global classifier. Table 3 lists the results on the TinyImageNet dataset in both the PRA and PAT settings. As can be seen, FedRE achieves performance comparable to FedAllRep, suggesting that uploading a single entangled representation per client can effectively support global classifier training. Also, FedRE significantly reduces communication overhead, providing a clear advantage over uploading all representations.

Table 3: Accuracy (%) and communication overhead comparison on the TinyImageNet dataset. In each column, the best results are **bolded**, and the second-best results are underlined.

Method	PRA		PAT	
	Accuracy (%)	# Params ($\times 10^3$)	Accuracy (%)	# Params ($\times 10^3$)
FedAllRep	**31.20**	42160.39	**38.62**	42258.88
FedRE	30.48	**4118.48**	38.52	**4118.48**

Q5: How effective are various RE methods? We provide a comprehensive analysis of various RE methods (see mathematical details in Appendix B): (1) *Random Select Representation* (**RSR**) randomly selects one representation from each client; (2) *Vanilla Average Representation* (**VAR**) averages all representations per client into a single representation, with equal weight assigned to each; (3) *Random Average Representation* (**RAR**) entangles all representations per client into a single representation using a normalized weight vector, with elements randomly drawn from $\mathcal{U}(0, 1)$ and normalized to sum to one; (4) *Random Select Prototype* (**RSP**) calculates prototypes for each client and randomly selects one prototype per client; (5) *Vanilla Average Prototype* (**VAP**) calculates prototypes for each client and averages them into a single representation, with equal weight assigned to each; (6) *Random Average Prototype* (**RAP**) calculates prototypes for each client and entangles them into a single representation using a normalized weight vector, with elements randomly drawn from $\mathcal{U}(0, 1)$ and normalized to sum to one. Table 4 lists the results on the CIFAR-10 and CIFAR-100 datasets in the PRA setting. We have the following observations. (1) RSR performs the worst, as each client uploads only a randomly selected representation, which is insufficient to train the global classifier. (2) RSP outperforms RSR, as the prototype encodes the knowledge of all representations within a category, it is more representative than a single representation. (3) VAP and RAP outperform VAR and RAR, respectively, indicating that prototype-based entanglement yields better model performance. (4) RAP surpasses VAP, demonstrating that random weights for entanglement are more effective than equal weights. Thus, we empirically choose RAP in the implementation of FedRE.

Q6: How effective are various representation mapping operations? We analyze various representation mapping operations to address the heterogeneity of representations with different model architectures: (1) Average Pooling (**AP**) maps the representations to a unified space by averaging the values across regions. (2) Max Pooling (**MP**) maps the representations to a unified space by selecting the maximum values across regions. (3) Full Connection (**FC**) maps the representations to a unified space by using a fully connected layer transformation. Table 5 presents the results on the CIFAR-100

Table 4: Accuracy (%) comparison between distinct representation entanglement mechanisms on the CIFAR-10 and CIFAR-100 datasets in the PRA setting. In each row, the best results are **bolded**, and the second-best results are underlined.

Dataset	RSR	VAR	RAR	RSP	VAP	RAP
CIFAR-10	79.10	81.32	80.20	80.45	81.42	**82.60**
CIFAR-100	40.41	44.88	43.19	43.25	46.12	**46.36**

Table 5: Accuracy (%) comparison between distinct representation mapping operation on the CIFAR-100 dataset in PRA and PAT settings. In each row, the best results are **bolded**, and the second-best results are underlined.

Setting	AP	MP	FC
PRA	**46.36**	45.97	44.53
PAT	**62.56**	61.93	60.19

dataset in both PRA and PAT settings. We observe that AP achieves the best performance. One possible reason is that AP captures more comprehensive information by averaging all values across regions. Thus, we empirically select AP in the implementation of FedRE.

Q7: How does FedRE perform under different participation ratios with varying statistical heterogeneity? We conduct experiments on the CIFAR-10 dataset under the *partial participation setting* with varying levels of statistical heterogeneity. Specifically, we adopt 100 clients and set participation rates to 10/100 and 20/100, while adjusting the Dirichlet distribution parameter α to 0.07 and 0.1, respectively, in the PRA setting. Furthermore, we follow [17] and simulate the long-tail settings by modifying imbalance factors (IF) to 100 and 50, then set α to 0.07. The results in Table 6 show that FedRE outperforms other methods across most scenarios, confirming its effectiveness under partial participation with highly heterogeneous distributions. In addition, we evaluate other statistical-heterogeneous settings in Appendix F.3. All the results indicate that FedRE is effective across diverse scenarios.

Table 6: Accuracy (%) comparison for partial participation scenarios with varying statistical heterogeneity in the PRA setting on the CIFAR-10 dataset. Here, α is a Dirichlet distribution parameter, and IF denotes imbalance factors of the long-tail setting. In each column, the best results are **bolded**, and the second-best results are underlined.

Method	Participation rate	$\alpha = 0.07$	$\alpha = 0.1$	IF = 100, $\alpha = 0.07$	IF = 50, $\alpha = 0.07$	Participation rate	$\alpha = 0.07$	$\alpha = 0.1$	IF = 100, $\alpha = 0.07$	IF = 50, $\alpha = 0.07$
FedProto [36]		54.00	51.18	45.21	43.92		56.90	55.47	47.08	44.68
FedGH [45]	10 / 100	78.23	76.87	**67.30**	63.73	20 / 100	80.57	77.84	65.56	64.90
FedRE		**81.17**	**79.56**	67.12	**66.37**		**82.80**	**81.99**	**69.33**	**68.81**

Q8: Does FedRE remain effective in the model-homogeneous setting? Since model-homogeneous FL can be regarded as a special case of model-heterogeneous FL, we evaluate FedRE under the model-homogeneous setting in both PRA and PAT settings. In addition to model-heterogeneous methods, we include additional model-homogeneous baseline methods, *i.e.*, FedAvg [26], FedAvgDBE [51], and FedALA [52], for comparison. The results are presented in Table 10 in Appendix F.4. As shown, FedRE achieves the best performance across all datasets. Also, Figures 10(c)-(d) in Appendix F.4 show that FedRE maintains superior accuracy during the training process on the TinyImageNet dataset, indicating stable performance convergence. Those results further confirm that FedRE is still effective in the model-homogeneous FL scenarios.

We investigate two additional analysis experiments in Appendix E: (1) a combined analysis of representation entanglement and mapping, and (2) a feature visualization analysis. Those analyses further verify the effectiveness of FedRE.

6 Conclusion

In this paper, we rethink client knowledge in FL and introduce the entangled representation, which serves as an effective, privacy-preserving, and lightweight form of client knowledge. Building on this concept, we propose FedRE to address the model-heterogeneous FL problem, where each client entangles all local representations from distinct categories into a single representation and uploads it to the server to collaboratively train the global classifier. We provide a theoretical analysis of FedRE's convergence properties. Experiments on three datasets confirm that FedRE achieves a well-balanced trade-off between model performance, privacy protection, and communication overhead. Accordingly, we believe that the concept of entangled representations offers a novel perspective to balancing those critical factors in FL. One promising direction for future work is to extend this concept to more challenging FL scenarios, for instance, federated class-incremental learning [5].

References

[1] Rodolfo Stoffel Antunes, Cristiano André da Costa, Arne Küderle, Imrana Abdullahi Yari, and Björn Eskofier. Federated learning for healthcare: Systematic review and architecture proposal. *ACM TIST*, 13(4):1–23, 2022.

[2] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *ICML*, pages 2089–2099, 2021.

[3] Yongheng Deng, Feng Lyu, Ju Ren, Yi-Chao Chen, Peng Yang, Yuezhi Zhou, and Yaoxue Zhang. Fair: Quality-aware federated learning with precise user incentive and model aggregation. In *INFORCOM*, pages 1–10, 2021.

[4] Yongheng Deng, Feng Lyu, Ju Ren, Yi-Chao Chen, Peng Yang, Yuezhi Zhou, and Yaoxue Zhang. Improving federated learning with quality-aware user incentive and auto-weighted model aggregation. *TPDS*, 33(12):4515–4529, 2022.

[5] Jiahua Dong, Lixu Wang, Zhen Fang, Gan Sun, Shichao Xu, Xiao Wang, and Qi Zhu. Federated class-incremental learning. In *CVPR*, pages 10164–10173, 2022.

[6] Xin Dong, Sai Qian Zhang, Ang Li, and HT Kung. Spherefed: Hyperspherical federated learning. In *ECCV*, pages 165–184, 2022.

[7] Jiamin Fan, Kui Wu, Guoming Tang, Yang Zhou, and Shengqiang Huang. Taking advantage of the mistakes: Rethinking clustered federated learning for iot anomaly detection. *TPDS*, 2024.

[8] Hossein Gholamalinezhad and Hossein Khosravi. Pooling methods in deep neural networks, a review. *arXiv preprint arXiv:2009.07485*, 2020.

[9] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *TPAMI*, 45(1):87–110, 2022.

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

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

[12] Sohei Itahara, Takayuki Nishio, Yusuke Koda, Masahiro Morikura, and Koji Yamamoto. Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data. *TMC*, 22(1):191–205, 2021.

[13] S. Kim, M. Jeong, S. Kim, S. Cho, S. Ahn, and S. Y. Yun. Feddr+: Stabilizing dot-regression with global feature distillation for federated learning. *TMLR*, 2025.

[14] S. Kim, G. Lee, J. Oh, and S. Y. Yun. Fedfn: Feature normalization for alleviating data heterogeneity problem in federated learning. In *NeurIPS 2023 Workshop: Federated Learning in the Age of Foundation Models*, 2023.

[15] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical report, University of Toronto*, 2009.

[16] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[17] Kangwook Lee, Hoon Kim, Kyungmin Lee, Changho Suh, and Kannan Ramchandran. Synthesizing differentially private datasets using random mixing. In *ISIT*, pages 542–546, 2019.

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

[19] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *CVPR*, pages 10713–10722, 2021.

[20] Zexi Li, Tao Lin, Xinyi Shang, and Chao Wu. Revisiting weighted aggregation in federated learning with neural networks. In *ICML*, pages 19767–19788, 2023.

[21] Zexi Li, Xinyi Shang, Rui He, Tao Lin, and Chao Wu. No fear of classifier biases: Neural collapse inspired federated learning with synthetic and fixed classifier. In *ICCV*, pages 5319–5329, 2023.

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

[23] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. In *NeurIPS*, volume 33, pages 2351–2363, 2020.

[24] Xiaosong Ma, Jie Zhang, Song Guo, and Wenchao Xu. Layer-wised model aggregation for personalized federated learning. In *CVPR*, pages 10092–10101, 2022.

[25] Disha Makhija, Xing Han, Nhat Ho, and Joydeep Ghosh. Architecture agnostic federated learning for neural networks. In *ICML*, pages 14860–14870, 2022.

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

[27] Dinh C Nguyen, Ming Ding, Pubudu N Pathirana, Aruna Seneviratne, Jun Li, and H Vincent Poor. Federated learning for internet of things: A comprehensive survey. *IEEE Commun Surv Tutorials*, 23(3):1622–1658, 2021.

[28] Jaehoon Oh, Sangmook Kim, and Se-Young Yun. Fedbabu: Toward enhanced representation for federated image classification. In *ICLR*, 2022.

[29] Krishna Pillutla, Kshitiz Malik, Abdel-Rahman Mohamed, Mike Rabbat, Maziar Sanjabi, and Lin Xiao. Federated learning with partial model personalization. In *ICML*, pages 17716–17758, 2022.

[30] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018.

[31] Torsten Schlett, Christian Rathgeb, Olaf Henniger, Javier Galbally, Julian Fierrez, and Christoph Busch. Face image quality assessment: A literature survey. *ACM Comput Surv*, 54(10s):1–49, 2022.

[32] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hypernetworks. In *ICML*, pages 9489–9502, 2021.

[33] Yujun Shi, Jian Liang, Wenqing Zhang, Chuhui Xue, Vincent YF Tan, and Song Bai. Understanding and mitigating dimensional collapse in federated learning. *TPAMI*, 46(5):2936–2949, 2023.

[34] Guangyu Sun, Matias Mendieta, Jun Luo, Shandong Wu, and Chen Chen. Fedperfix: Towards partial model personalization of vision transformers in federated learning. In *ICCV*, pages 4988–4998, 2023.

[35] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

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

[37] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *CVPR*, pages 9446–9454, 2018.

[38] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(11), 2008.

[39] Lei Wang, Jieming Bian, Letian Zhang, Chen Chen, and Jie Xu. Taming cross-domain representation variance in federated prototype learning with heterogeneous data domains. *arXiv preprint arXiv:2403.09048*, 2024.

[40] Lixu Wang, Shichao Xu, Xiao Wang, and Qi Zhu. Addressing class imbalance in federated learning. In *AAAI*, volume 35, pages 10165–10173, 2021.

[41] Feijie Wu, Xingchen Wang, Yaqing Wang, Tianci Liu, Lu Su, and Jing Gao. Fiarse: Model-heterogeneous federated learning via importance-aware submodel extraction. 2024.

[42] Wu F. Lyu L. et al Wu, C. Communication-efficient federated learning via knowledge distillation. *Nat Commun*, 2022.

[43] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM TIST*, 10(2):1–19, 2019.

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

[45] Liping Yi, Gang Wang, Xiaoguang Liu, Zhuan Shi, and Han Yu. Fedgh: Heterogeneous federated learning with generalized global header. In *ACM MM*, pages 8686–8696, 2023.

[46] Liping Yi, Han Yu, Chao Ren, Gang Wang, Xiaoxiao Li, et al. Federated model heterogeneous matryoshka representation learning. *NeurIPS*, 37:66431–66454, 2024.

[47] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *CVPR*, pages 8715–8724, 2020.

[48] Feilong Zhang, Deming Zhai, Guo Bai, Junjun Jiang, Qixiang Ye, Xiangyang Ji, and Xianming Liu. Towards fairness-aware and privacy-preserving enhanced collaborative learning for healthcare. *Nat Commun*, 16(1):2852, 2025.

[49] Hao Zhang, Chenglin Li, Nuowen Kan, Ziyang Zheng, Wenrui Dai, Junni Zou, and Hongkai Xiong. Improving generalization in federated learning with model-data mutual information regularization: A posterior inference approach. *NeurIPS*, 37:136646–136678, 2024.

[50] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

[51] Jianqing Zhang, Yang Hua, Jian Cao, Hao Wang, Tao Song, Zhengui XUE, Ruhui Ma, and Haibing Guan. Eliminating domain bias for federated learning in representation space. In *NeurIPS*, 2023.

[52] Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Fedala: Adaptive local aggregation for personalized federated learning. In *AAAI*, volume 37, pages 11237–11244, 2023.

[53] Jianqing Zhang, Yang Liu, Yang Hua, and Jian Cao. Fedtgp: Trainable global prototypes with adaptive-margin-enhanced contrastive learning for data and model heterogeneity in federated learning. In *AAAI*, number 15, pages 16768–16776, 2024.

[54] Jianqing Zhang, Yang Liu, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Jian Cao. Pfllib: Personalized federated learning algorithm library. *arXiv preprint arXiv:2312.04992*, 2023.

[55] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *ICML*, pages 12878–12889, 2021.

Additional details and results are provided in the appendices, covering the following contents.

- Appendix A: Limitations and broader impacts of FedRE.
- Appendix B: Implementation details for various representation entanglement methods.
- Appendix C: Convergence proof for FedRE.
- Appendix D: Detailed experimental setup.
- Appendix E: Additional experiments for analysis.
- Appendix F: Supplementary experimental results (not included in the main text).

A Limitations and Broader Impacts

Limitations. While entangled representations provide effective, privacy-preserving, and lightweight client knowledge for FL, they are primarily designed for federated discriminative tasks such as image classification. Consequently, they cannot be directly applied to federated generative tasks, including text generation. Extending entangled representations to federated generative modeling remains a promising direction for future research.

Broader Impacts. This work tackles the dual challenges of model and statistical heterogeneity in FL. The proposed method requires each client to upload only an entangled representation to the server, which achieves a balanced trade-off among model performance, communication overhead, and privacy protection. Such a design is especially well-suited for collaborative learning scenarios involving diverse, resource-limited devices like smartphones, improving practical usability and supporting scalable deployment.

B Mathematical Details of Various Representation Entanglement Methods

We now introduce the mathematical details of different RE mechanisms. The general form of RE, calculated from a single client's representation set $\mathcal{R} = \{\mathbf{r}_i, \mathbf{y}_i\}_{i=1}^n$, is formulated as follows:

$$\widetilde{\mathbf{r}}_{\text{RE}} = \sum_{i=1}^{n} w_i \mathbf{r}_i, \widetilde{\mathbf{y}}_{\text{RE}} = \sum_{i=1}^{n} w_i \mathbf{y}_i, \tag{6}$$

where $w_i \in (0, 1)$ is the weight of $\mathbf{r}_i$, which can be determined by different RE methods as follows:

- **Random Select Representation** (RSR) randomly selects one representation from each client per global communication round. Thus, w_i can be considered as a binary value:

$$w_i = \begin{cases} 1, & \text{if } \mathbf{r}_i \text{ is selected} \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

- **Vanilla Average Representation** (VAR) averages all representations per client into a single representation, with equal weight assigned to each. Thus, w_i is simply:

$$w_i = \frac{1}{n}, \forall i \in \{1, 2, \cdots, n\}. \tag{8}$$

- **Random Average Representation** (RAR) entangles representations per client into a single representation using a normalized weight vector, with elements randomly drawn from $\mathcal{U}(0, 1)$ and normalized to sum to one. Therefore, the specific form of w_i is:

$$w_i = \frac{u_i}{\sum_{i=1}^{n} u_i}, \text{where } u_i \sim \mathcal{U}(0, 1). \tag{9}$$

- **Random Select Prototype** (RSP) calculates prototypes for each client and randomly selects one prototype per client in each global communication round. Hence, w_i can be formulated as follows:

$$w_i = \begin{cases} \frac{1}{n_c}, & \text{if } c\text{-th prototype is selected} \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

- **Vanilla Average Prototype** (VAP) calculates prototypes for each client and averages them into a single representation, with equal weight assigned to each. Thus, w_i is:

$$w_i = \frac{1}{Cn_c}, \text{if } \mathbf{r}_i \text{belongs to category } c. \tag{11}$$

- **Random Average Prototype** (RAP) calculates prototypes for each client and entangles them into a single representation using a normalized weight vector, with elements randomly drawn from $\mathcal{U}(0,1)$ and normalized to sum to one. Thus, w_i can be calculated as follows:

$$w_i = \frac{u_i}{n_c \sum_{i=1}^{n} u_i}, \text{if } \mathbf{r}_i \text{ belongs to category } c, \text{where } u_i \sim \mathcal{U}(0,1). \tag{12}$$

C Convergence Proof for FedRE

To prove Theorem 1 and Theorem 2, we first introduce and prove the Lemma 1 and Lemma 2.

Lemma 1. *Under Assumptions 1 and 2, in the $t+1$-th communication round, the local objective function is bounded from the initial to the final local iteration, i.e.,* $\mathbb{E}\left[\mathcal{L}_{(t+1)E}\right] \leq \mathcal{L}_{tE+0} - (\eta_l - \frac{L_1\eta_l^2}{2}) \sum_{e=0}^{E-1} \|\nabla \mathcal{L}_{tE+e}\|_2^2 + \frac{L_1 E \eta_l^2}{2}\sigma^2$.

Lemma 1, originally derived by [36], is accompanied by a detailed proof in their work.

Lemma 2. *Under Assumptions 1 to 3, the local objective function is bounded after the local classifier is replaced by global one, i.e.,* $\mathbb{E}[\mathcal{L}_{(t+1)E+0}] \leq \mathbb{E}[\mathcal{L}_{(t+1)E}] + \frac{L_1}{2}\delta^2$.

Proof.

$$\mathcal{L}_{(t+1)E+0} = \mathcal{L}_{(t+1)E} + \mathcal{L}_{(t+1)E+0} - \mathcal{L}_{(t+1)E} \tag{13}$$

$$= \mathcal{L}_{(t+1)E} + \mathcal{L}(\boldsymbol{\phi}_k^{(t+1)E}, \boldsymbol{\omega}^{(t+1)E}; \mathcal{D}_k) - \mathcal{L}(\boldsymbol{\phi}_k^{(t+1)E}, \boldsymbol{\omega}_k^{(t+1)E}; \mathcal{D}_k) \tag{14}$$

$$\leq \mathcal{L}_{(t+1)E} + \left\langle \nabla \mathcal{L}(\boldsymbol{\phi}_k^{(t+1)E}, \boldsymbol{\omega}_k^{(t+1)E}), (\boldsymbol{\phi}_k^{(t+1)E}, \boldsymbol{\omega}^{(t+1)E}) - (\boldsymbol{\phi}_k^{(t+1)E}, \boldsymbol{\omega}_k^{(t+1)E}) \right\rangle \tag{15}$$

$$+ \frac{L_1}{2} \left\| (\boldsymbol{\phi}_k^{(t+1)E}, \boldsymbol{\omega}^{(t+1)E}) - (\boldsymbol{\phi}_k^{(t+1)E}, \boldsymbol{\omega}_k^{(t+1)E}) \right\|_2^2$$

$$\leq \mathcal{L}_{(t+1)E} + \frac{L_1}{2} \left\| (\boldsymbol{\phi}_k^{(t+1)E}, \boldsymbol{\omega}^{(t+1)E}) - (\boldsymbol{\phi}_k^{(t+1)E}, \boldsymbol{\omega}_k^{(t+1)E}) \right\|_2^2 \tag{16}$$

$$= \mathcal{L}_{(t+1)E} + \frac{L_1}{2} \left\| \boldsymbol{\omega}^{(t+1)E} - \boldsymbol{\omega}_k^{(t+1)E} \right\|_2^2. \tag{17}$$

Eq. (15) follows from the quadratic bound in Assumption 1, *i.e.*, $\mathcal{L}_{t_1} - \mathcal{L}_{t_2} \leq \langle \nabla \mathcal{L}_{t_2}, (\boldsymbol{\theta}^{t_1} - \boldsymbol{\theta}^{t_2}) \rangle + \frac{L_1}{2}\|\boldsymbol{\theta}^{t_1} - \boldsymbol{\theta}^{t_2}\|_2^2$. Eq. (16) is derived based on Assumption 3. Furthermore, taking expectations on both sides of Eq. (17), we obtain

$$\mathbb{E}[\mathcal{L}_{(t+1)E+0}] \leq \mathbb{E}[\mathcal{L}_{(t+1)E}] + \frac{L_1}{2}\mathbb{E}\left[\|\boldsymbol{\omega}^{(t+1)E} - \boldsymbol{\omega}_k^{(t+1)E}\|_2^2\right]. \tag{18}$$

According to Assumption 3, Eq. (18) can be derived as

$$\mathbb{E}[\mathcal{L}_{(t+1)E+0}] \leq \mathbb{E}[\mathcal{L}_{(t+1)E}] + \frac{L_1}{2}\delta^2. \tag{19}$$

Next, we provide detailed proofs for Theorem 1 and Theorem 2.

C.1 Proof for Theorem 1

Based on Lemma 1 and Eq. (19), we have

$$\mathbb{E}[\mathcal{L}_{(t+1)E+0}] \leq \mathcal{L}_{tE+0} - (\eta_l - \frac{L_1\eta_l^2}{2}) \sum_{e=0}^{E-1} \|\nabla \mathcal{L}_{tE+e}\|_2^2 + \frac{L_1 E \eta_l^2}{2}\sigma^2 + \frac{L_1}{2}\delta^2. \tag{20}$$

If the local objective function is to converge, then the following inequality holds:

$$-(\eta_l - \frac{L_1\eta_l^2}{2})\sum_{e=0}^{E-1}\|\nabla L_{tE+e}\|_2^2 + \frac{L_1 E\eta_l^2}{2}\sigma^2 + \frac{L_1}{2}\delta^2 < 0. \tag{21}$$

Accordingly, we obtain

$$\eta_l < \frac{2\sum_{e=0}^{E-1}\|\nabla L_{tE+e}\|_2^2}{L_1 E\sigma^2 + L_1\sum_{e=0}^{E-1}\|\nabla L_{tE+e}\|_2^2}. \tag{22}$$

Thus, based on Eq. (22), if the local learning rate satisfies:

$$\eta_l^{e'} < \frac{2\sum_{e=0}^{e'}\|\nabla L_{tE+e}\|_2^2}{L_1 E\sigma^2 + L_1\sum_{e=0}^{e'}\|\nabla L_{tE+e}\|_2^2}, e' = 0, 1, \ldots, E-1, \tag{23}$$

then the convergence of the local objective function is guaranteed.

C.2 Proof for Theorem 2

Taking expectations on both sides of Eq. (20), we have

$$\begin{aligned}
&\mathbb{E}[L_{(t+1)E+0}] \leq L_{tE+0} - (\eta_l - \frac{L_1\eta_l^2}{2})\sum_{e=0}^{E-1}\mathbb{E}\left[\|\nabla L_{tE+e}\|_2^2\right] + \frac{L_1 E\eta_l^2}{2}\sigma^2 + \frac{L_1}{2}\delta^2 \\
\Rightarrow &\sum_{e=0}^{E-1}\mathbb{E}\left[\|\nabla L_{tE+e}\|_2^2\right] \leq \frac{L_{tE+0} - \mathbb{E}[L_{(t+1)E+0}] + \frac{L_1 E\eta_l^2}{2}\sigma^2 + \frac{L_1}{2}\delta^2}{\eta_l - \frac{L_1\eta_l^2}{2}}.
\end{aligned} \tag{24}$$

Accordingly, we have

$$\frac{1}{TE}\sum_{t=0}^{T-1}\sum_{e=0}^{E-1}\mathbb{E}\left[\|\nabla L_{tE+e}\|_2^2\right] \leq \frac{\frac{1}{TE}\sum_{t=0}^{T-1}\left(L_{tE+0} - \mathbb{E}[L_{(t+1)E+0}]\right) + \frac{L_1\eta_l^2}{2}\sigma^2 + \frac{L_1}{2E}\delta^2}{\eta_l - \frac{L_1\eta_l^2}{2}}. \tag{25}$$

If the local objective function is to converge, then the right side of Eq. (25) satisfies

$$\frac{\frac{2}{TE}\sum_{t=0}^{T-1}\left(L_{tE+0} - \mathbb{E}[L_{(t+1)E+0}]\right) + L_1\eta_l^2\sigma^2 + \frac{L_1}{E}\delta^2}{2\eta_l - L_1\eta_l^2} < \xi, \tag{26}$$

where ξ is an arbitrarily small positive value. Let $\Delta = \mathcal{L}_0 - \mathcal{L}^*$ where $\mathcal{L}_0$ and $\mathcal{L}^*$ denote the initial and optimal values of the local objective function, respectively. Since $\sum_{t=0}^{T-1}\left(\mathcal{L}_{tE+0} - \mathbb{E}[\mathcal{L}_{(t+1)E+0}]\right) \leq \Delta$, Eq. (26) holds when

$$\frac{\frac{2\Delta}{TE} + L_1\eta_l^2\sigma^2 + \frac{L_1}{E}\delta^2}{2\eta_l - L_1\eta_l^2} < \xi \Rightarrow T > \frac{2\Delta}{\xi E(2\eta_l - L_1\eta_l^2) - L_1 E\eta_l^2\sigma^2 - L_1\delta^2}. \tag{27}$$

Since $T > 0$ and $\Delta \geq 0$, based on Eq. (27), we obtain

$$\xi E(2\eta_l - L_1\eta_l^2) - L_1 E\eta_l^2\sigma^2 - L_1\delta^2 > 0 \Rightarrow \eta_l < \frac{2\xi}{\xi L_1 + L_1\sigma^2}. \tag{28}$$

D Detailed Experimental Setup

We implement the FedRE based on the PFLlib framework [54], engaging ten clients with a default participation rate of 100%. On each client, training involves one local epoch utilizing mini-batch stochastic gradient descent (SGD) with a learning rate of 0.05. The dimensionality of the unified space across distinct clients is set to 512. The local samples are divided in a 3:1 ratio for training and testing. For the CIFAR-10 and CIFAR-100 datasets, we set the batch size to 32, and for the TinyImageNet dataset, it is set to 64. On the server, the optimizer is SGD with a learning rate of 0.01 and a batch size of 10. For clarity, the detailed setup is summarized in Table 7. In addition, the experiments are conducted on NVIDIA GPUs, primarily including the GeForce RTX A800.

Table 7: *Detailed experimental setup utilized in this paper.*

Devices
CPU: Intel(R) Xeon(R) Gold 6348
SSD:100GB
GPU: A800-80GB
Software Tools
CUDA 12.1
Pytorch 2.1.0
Python 3.10
Statistic-heterogeneous Setting
Practical setting (PRA) & Pathological setting (PAT)
Model Training
Local batch size: 64 (TinyImageNet), 32 (CIFAR-10&100)
Local Optimizer: SGD
Local learning rate η_l:
0.06 (Model-heterogeneity with PRA & PAT, CIFAR-10&CIFAR-100&TinyImageNet);
0.01 (Model-homogeneity with PRA & PAT, CIFAR-100&TinyImageNet);
0.007 (Model-homogeneity with PRA, CIFAR-10);
0.008 (Model-homogeneity with PAT, CIFAR-10)
Server batch size: 10
Server Optimizer : SGD
Server learning rate: 0.01
Model Setting
Local model in model-heterogeneity: CNN model/ GoogleNet/ MobileNetv2
/ ResNet family/ Vision Transformer models
Local model in model-homogeneity: CNN model (CIFAR-10&100)
ResNet18 (TinyImageNet)

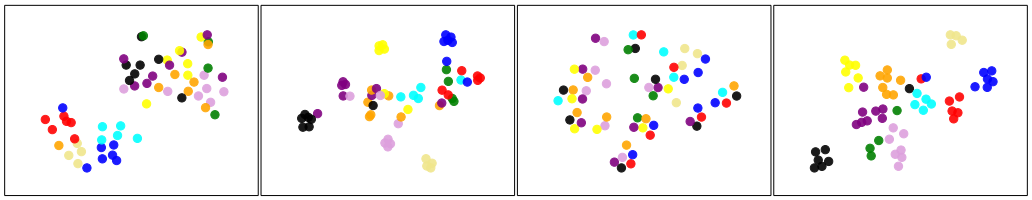
E Additional Analysis

Q9: How effective are different combinations of representation entanglement and mapping strategies? We further investigate the combination of distinct representation entanglement methods and representation mapping operations. Table 8 lists the results on CIFAR-10 in the model-heterogeneous setting under the PRA setting. Among the tested combinations, RAP + AP yields the highest accuracy compared to other combinations, highlighting the effectiveness of this combination.

Table 8: Accuracy (%) comparison with distinct combinations of representation entanglement and mapping strategies on the CIFAR-10 datasets in the PRA setting. The best results are **bolded**, and the second-best results are underlined.

	RSR	**VAR**	**RAR**	**RSP**	**VAP**	**RAP**
AP	79.10	81.32	80.20	80.45	81.42	**82.60**
MP	78.37	81.17	80.31	80.07	80.64	<u>81.93</u>
FC	78.02	80.60	79.69	79.92	80.29	81.28

Q10: What does the representation distribution learned by FedRE look like? We employ the t-SNE technique [38] to visualize the learned prototypes of FedAvg, FedProto, FedGH, and FedRE on the CIFAR-10 dataset. Figure 5 reveals that in FedRE, prototypes from different clients belonging to the same category cluster more tightly, with clearer boundaries across distinct categories. This indicates that FedRE effectively integrates knowledge from different clients to facilitate their learning.



(a) FedAvg (b) FedProto (c) FedGH (d) FedRE

Figure 5: t-SNE visualization of prototypes learned by distinct approaches, where distinct colors represent different categories.

F Supplementary Experimental Results

F.1 Communication Overhead Evaluation

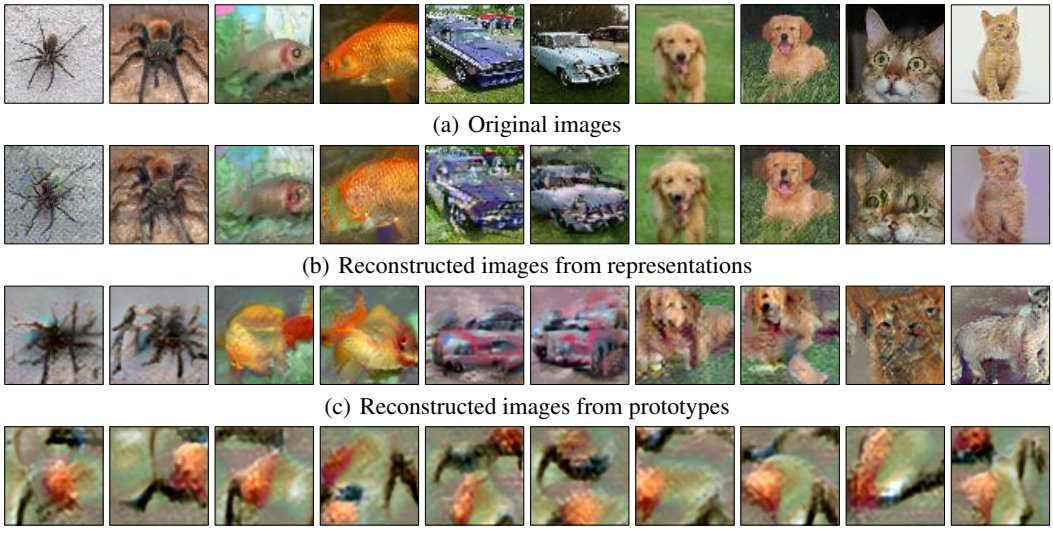
Table 9: Communication overhead (# params $\times 10^3$) comparison on three datasets. In each column, the best results are **bolded**, and the second-best results are underlined.

Method	CIFAR-10				CIFAR-100				TinyImageNet			
	Model-homo		Model-hete		Model-homo		Model-hete		Model-homo		Model-hete	
	Upload	Broadcast	Upload	Broadcast	Upload	Broadcast	Upload	Broadcast	Upload	Broadcast	Upload	Broadcast
LG-FedAvg [22]	51.30	51.30	51.30	51.30	513.00	513.00	513.00	513.00	4098.00	4098.00	4098.00	4098.00
FedGH [45]	31.23	**51.20**	31.23	**51.20**	257.02	**512.00**	257.02	**512.00**	1918.98	**4096.00**	1918.98	**4096.00**
FedKD [42]	3374.28	3374.28	3353.68	3353.68	4234.28	4234.28	3524.67	3524.67	90503.00	90503.00	57544.97	57544.97
FedGen [55]	8785.38	8785.38	51.30	51.30	9247.08	9247.08	513.00	513.00	239178.32	239178.32	4098.00	4098.00
FedProto [36]	31.23	**51.20**	31.23	**51.20**	257.02	**512.00**	257.02	**512.00**	1918.98	**4096.00**	1918.98	**4096.00**
FPL [11]	31.23	87.04	31.23	112.64	257.02	916.48	257.02	1182.72	1918.98	9768.96	1918.98	10567.68
FedMRL [46]	8746.98	8746.98	8746.98	8746.98	8863.08	8863.08	8863.08	8863.08	56178.00	56178.00	56178.00	56178.00
FedTGP [53]	31.23	**51.20**	31.23	**51.20**	257.02	**512.00**	257.02	**512.00**	1918.98	**4096.00**	1918.98	**4096.00**
FedAvg [26]	8785.38	8785.38	-	-	9247.08	9247.08	-	-	239178.32	239178.32	-	-
FedALA [52]	8785.38	8785.38	-	-	9247.08	9247.08	-	-	239178.32	239178.32	-	-
FedAvgDBE [51]	8785.38	8785.38	-	-	9247.08	9247.08	-	-	239178.32	239178.32	-	-
FedRE	**5.12**	51.30	**5.12**	51.30	**5.12**	513.00	**5.12**	513.00	**20.48**	4098.00	**20.48**	4098.00

Table 9 lists all the results of communication overhead on the CIFAR-10, CIFAR-100, and TinyImageNet datasets under both model-heterogeneous (Model-hete) and model-homogeneous (Model-homo) scenarios. we can see that FedRE consistently exhibits the lowest communication overhead across all scenarios. Furthermore, as the dataset size and number of categories increase, the communication overhead of FedRE decreases more significantly.

F.2 Privacy Protection Evaluation

Figure 6 provides more image reconstruction results. As can be observed, the content and categories of images reconstructed from the entangled representations are indistinguishable, further demonstrating that FedRE provides superior privacy protection.



(a) Original images

(b) Reconstructed images from representations

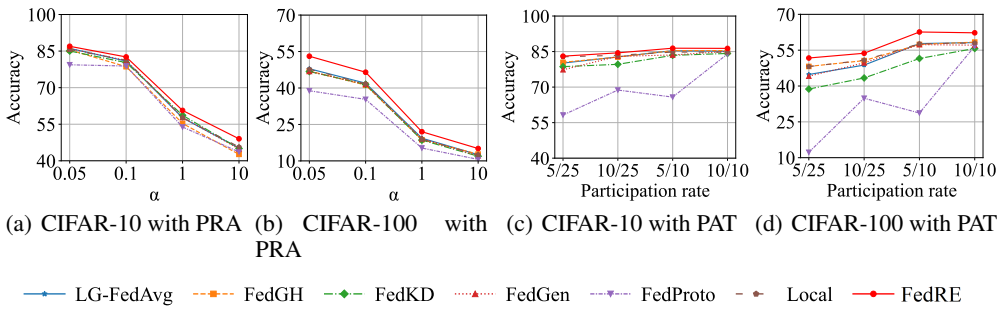
(c) Reconstructed images from prototypes

(d) Reconstructed images from entangled representations

Figure 6: Privacy protection comparison on the TinyImageNet dataset.

F.3 Statistical Heterogeneity Analysis

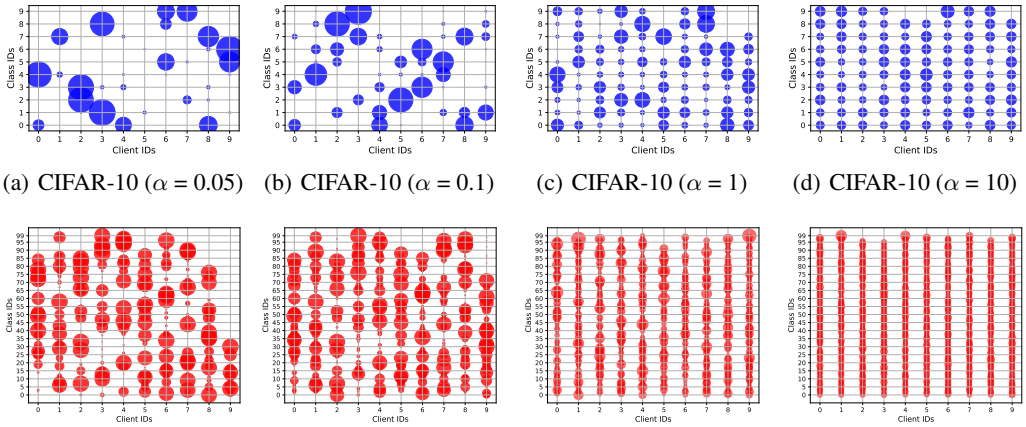
To further assess the effectiveness of FedRE under varying degrees of statistical heterogeneity, we adjust the Dirichlet distribution parameter α (*i.e.*, 0.05, 0.1, 1, 10) in the PRA setting and the



(a) CIFAR-10 with PRA (b) CIFAR-100 with PRA (c) CIFAR-10 with PAT (d) CIFAR-100 with PAT

Figure 7: Accuracy (%) comparison between distinct statistic-heterogeneous scenarios on the CIFAR-10 and CIFAR-100 datasets.

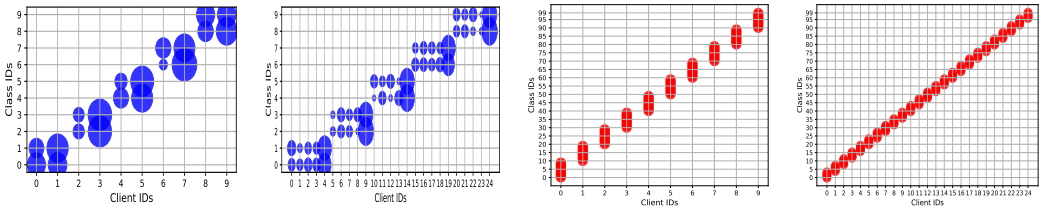
client participation rate (*i.e.*, 5/25, 10/25 for 25 clients, 5/10, 10/10 for 10 clients) in the PAT setting, respectively, to control sample skewness. The resulting sample distributions are visualized in Figures 8-9. The results on the CIFAR-10 and CIFAR-100 datasets, under the model-heterogeneous setting, are shown in Figure 7. As can be seen, FedRE consistently achieves the highest accuracy across different levels of statistical heterogeneity, demonstrating its adaptability to various statistic-heterogeneous scenarios.



(a) CIFAR-10 ($\alpha = 0.05$) (b) CIFAR-10 ($\alpha = 0.1$) (c) CIFAR-10 ($\alpha = 1$) (d) CIFAR-10 ($\alpha = 10$)

(e) CIFAR-100 ($\alpha = 0.05$) (f) CIFAR-100 ($\alpha = 0.1$) (g) CIFAR-100 ($\alpha = 1$) (h) CIFAR-100 ($\alpha = 10$)

Figure 8: The sample distributions for all clients on the CIFAR-10 and CIFAR-100 datasets under the PRA settings with varying parameters α. The size of each circle indicates the number of samples.



(a) CIFAR-10 (10 Clients) (b) CIFAR-10 (25 Clients) (c) CIFAR-100 (10 Clients) (d) CIFAR-100 (25 Clients)

Figure 9: The sample distributions for all clients on the CIFAR-10 and CIFAR-100 datasets under the PAT settings with varying client numbers. The size of each circle indicates the number of samples.

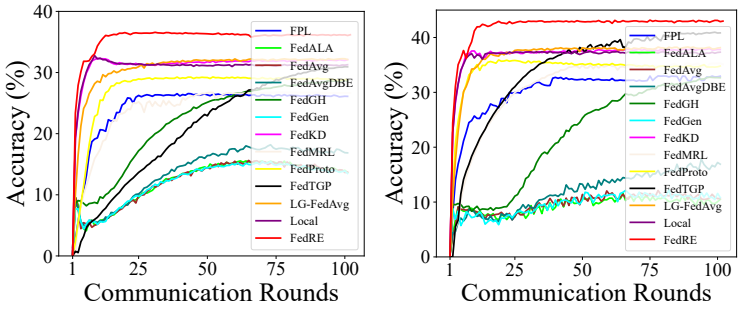
F.4 Model-homogeneous FL Evaluation

Model-homogeneous FL can be regarded as a special case of model-heterogeneous FL, where all clients utilize the same local model. In our experiments, we adopt a four-layer CNN for the CIFAR-10 and CIFAR-100 datasets and use ResNet-18 for the TinyImageNet dataset.

Table 10 presents the results under the model-homogeneous setting, where FedRE achieves the best performance across all datasets. Specifically, FedRE's average accuracy is 63.88%, outperforming the second-best method, *i.e.*, LG-FedAvg, by 3.25%. Those results further confirm the effectiveness of FedRE. Additionally, Figures 10(c)-(d) provide performance convergence comparisons on the TinyImageNet dataset, where FedRE consistently maintains superior accuracy during the training process, indicating stable performance convergence.

Table 10: Accuracy (%) comparison on three datasets under the model-homogeneous setting. In each column, the best results are **bolded**, and the second-best results are underlined.

Algorithm	PRA			PAT			Average
	CIFAR-10	CIFAR-100	TinyImageNet	CIFAR-10	CIFAR-100	TinyImageNet	
LG-FedAvg [22]	86.92 ± 0.25	49.82 ± 0.39	32.00 ± 0.13	90.59 ± 0.17	66.00 ± 0.27	38.43 ± 0.23	60.63
FedAvg [26]	55.21 ± 0.12	30.37 ± 0.02	13.66 ± 0.41	52.70 ± 0.11	24.89 ± 0.20	9.98 ± 0.48	31.14
FedALA [52]	55.02 ± 0.14	29.89 ± 0.22	13.63 ± 0.10	52.83 ± 0.19	24.91 ± 0.15	10.65 ± 0.15	31.16
FedGH [45]	86.02 ± 0.17	48.59 ± 0.60	28.64 ± 0.26	90.46 ± 0.22	65.14 ± 0.26	32.40 ± 0.19	58.54
FedKD [42]	86.23 ± 0.12	51.91 ± 0.28	29.47 ± 0.31	90.01 ± 0.09	67.23 ± 0.38	35.34 ± 0.33	60.03
FedAvgDBE [51]	78.10 ± 0.20	35.23 ± 0.24	16.92 ± 0.52	82.27 ± 0.45	35.21 ± 0.27	16.80 ± 0.23	44.76
FedGen [55]	55.21 ± 0.14	29.90 ± 0.17	13.76 ± 0.23	52.37 ± 0.22	24.82 ± 0.38	10.67 ± 0.54	31.12
FedProto [36]	85.63 ± 0.22	50.52 ± 0.19	28.67 ± 0.17	91.04 ± 0.16	69.28 ± 0.07	34.75 ± 0.49	59.65
FPL [11]	83.60 ± 0.01	49.10 ± 0.06	26.87 ± 0.01	90.59 ± 0.01	67.31 ± 0.01	32.95 ± 0.07	58.40
FedMRL [46]	82.55 ± 0.01	48.41 ± 0.01	26.78 ± 0.01	89.02 ± 0.00	65.97 ± 0.11	35.22 ± 0.01	57.99
FedTGP [53]	85.59 ± 0.02	47.05 ± 0.04	30.89 ± 0.00	90.49 ± 0.01	67.47 ± 0.01	40.88 ± 0.00	60.40
Local	86.33 ± 0.11	49.88 ± 0.40	31.44 ± 0.15	90.54 ± 0.12	66.57 ± 0.17	37.46 ± 0.27	60.37
FedRE	**86.99 ± 0.01**	**52.12 ± 0.04**	**36.12 ± 0.21**	**91.06 ± 0.01**	**70.52 ± 0.17**	**42.45 ± 0.17**	**63.88**



(a) PRA (b) PAT

Figure 10: Accuracy (%) comparison between distinct communication rounds on the TinyImageNet dataset in the model-homogeneous FL setting in both the PRA and PAT settings.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We highlight our contributions in the Introduction (as detailed in Section 1).

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of this work are discussed in Appendix A.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: We present the main theoretical conclusion regarding convergence in Section 4.4 (Convergence Analysis). The corresponding detailed proof is provided in Appendix C.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: We describe the experimental setup in Section 5.1, with further details provided in Appendix D.

Guidelines:

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

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

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

Answer: [Yes]

Justification: We utilize three benchmark datasets, *i.e.*, CIFAR-10 [15], CIFAR-100 [15], and TinyImageNet [16], all of which are publicly available. Our codes are available at `https://anonymous.4open.science/r/FedREx`.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: We describe the experimental setup in Section 5.1, with further details provided in Appendix D.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: Table 1 and Table 10 (see Appendix F.4) report the results as the mean and standard deviation over three random trials.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The computational resources used in this work are described in Appendix D and Table 7.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: This work has been conducted in accordance with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The broader impacts of this work are discussed in Appendix A.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

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

11. **Safeguards**

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

    Answer: [NA]

    Justification: This work does not pose such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The open-source resources utilized in this work are properly cited in Section 5.1.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We have released our codes and a detailed README file in `https://anonymous.4open.science/r/FedREx`.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: This work does not involve crowdsourcing and research with human subjects.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: This work does not involve crowdsourcing and research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [NA]

    Justification: The LLM is used only for writing, editing, or formatting purposes.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.