GRAPE: Gossip-Based Representation Alignment Using Prototypical Embeddings for Heterogeneous IoT Devices

Saira Bano,* and Gianluca Rizzo†‡
saira.bano@unifg.it, gianluca.rizzo@hevs.ch
* Università di Foggia, Italy † HES SO Valais, Switzerland ‡ Università di Torino, Italy

***Abstract*—Gossip learning (GL) introduces a fully decentralized framework for training machine learning models across large-scale networks of heterogeneous IoT devices, leveraging peer-to-peer communication to enable collaborative learning without centralized coordination or raw data exchange. However, in real-world IoT environments, devices commonly operate under non-IID data distributions and often contains different model architectures, computational power, and storage capacity which leads to model heterogeneity. In such settings, conventional GL fails to converge reliably, as weight averaging is incompatible with heterogeneous architectures, while existing alternatives still rely on centralized coordination. To address these challenges, we introduce GRAPE, a fully decentralized, prototype-based GL framework that tackles both data and model heterogeneity. Instead of exchanging full model parameters, participating devices communicate compact class prototypes that serve as semantic summaries of local knowledge. We propose a contrastive prototype alignment objective that enables devices to refine their learned representations by attracting semantically consistent prototypes while repelling inconsistent ones during peer-to-peer exchanges. This approach enforces consistent representations across heterogeneous models within a unified embedding space. Extensive experiments show that GRAPE outperforms standard GL baselines, achieving higher accuracy, improved robustness to heterogeneity, and lower communication costs.**

***Index Terms*—Gossip Learning, Model heterogeneity, Data heterogeneity, IoT.**

I. Introduction

Deep learning's remarkable generalization capabilities are driven by access to large, diverse datasets. In the rapidly expanding Internet of Things (IoT) ecosystem, however, data are fundamentally distributed across vast, heterogeneous device populations, which cannot be centralized due to stringent privacy requirements and bandwidth constraints [1]. Federated Learning (FL) has emerged as a technique for enabling distributed model training without raw data exchange [2]. Nonetheless, central aggregation in FL poses major vulnerabilities, including a single point of failure, security risks, and limited scalability in large and dynamic IoT networks. [3]. Decentralized learning approaches, by removing the need for any central coordinator, offer a promising solution. Among them, Gossip Learning (GL) [4] stands out for its ability to harness peer-to-peer, serverless collaboration. In GL, each node retains its local dataset and periodically communicates with neighboring nodes to share and combine model information, enabling distributed knowledge sharing in a fully decentralized manner [4]. This makes GL particularly suited to the decentralized, resource-constrained, and intermittently connected nature of real-world IoT deployments.

However, practical deployment of GL faces two significant challenges: (i) data across devices is typically non-i.i.d., causing local models to drift apart and hindering global convergence, and (ii) devices vary significantly in computational capabilities, memory, and energy constraints leading to substantial model heterogeneity. These factors fundamentally complicate collaborative learning and exacerbate difficulties in aggregating knowledge across the network. Most existing solutions in server-based FL tackles data heterogeneity by improving aggregation schemes [5]–[7] or stabilizing local updates through regularization [8], [9]. Model heterogeneity is often tackled using knowledge distillation techniques that require a coordinating server and, sometimes, access to a representative global dataset [9], [10]. However, these strategies are predicated on server-dependent orchestration and global data availability, assumptions that fundamentally do not hold in GL contexts. The inherently asynchronous, peer-driven, and uncoordinated nature of GL thus magnifies both data and model heterogeneity, for which prior methods are insufficient.

To address these challenges, we propose GRAPE (*Gossip-based Representation Alignment using Prototypical Embeddings*), a server-less prototype-based decentralized learning framework in which devices exchange compact class prototypes instead of full model parameters. Each IoT device constructs these prototypes by computing the mean feature representation of its local samples belonging to the same class. During gossip interactions, prototypes are shared and incorporated into local training to enable effective collaborative learning. The training objective combines supervised classification for local optimization with contrastive prototype alignment for global consistency, attracting same-class prototypes across nodes while repelling different classes to ensure discriminability and coherence in heterogeneous settings.

Prototype sharing significantly reduces communication costs, enhances non-IID generalization, and enables decentralized representation alignment without central coordination. By exchanging compact class-level representations and leveraging a contrastive prototype loss, our method learns robust and discriminative embeddings under both data and model heterogeneity. To the best of our knowledge, this is the

first work to integrate prototype aggregation and contrastive learning within a fully decentralized gossip framework. The main contributions of this work are as follows:

- We propose GRAPE, a fully decentralized framework that addresses both data and model heterogeneity while drastically reducing communication overhead. By exchanging compact class-level prototypes instead of full model parameters, GRAPE aligns local representations in a lightweight and model-agnostic manner.
- We conduct extensive experiments on MNIST and CIFAR-10 under non-IID data and heterogeneous architectures, demonstrating that GRAPE achieves higher accuracy than baseline methods while significantly reducing communication cost.
- We further validate GRAPE through ablation studies, and the results show that it works even when connectivity is dynamic (not just static graphs), demonstrating that GRAPE remains effective under time-varying communication patterns and different hyperparameter settings.

II. Related Works

A. Gossip Learning

In recent years GL has received growing attention as a scalable framework for decentralized model training in heterogeneous environments. Ghosh et al. [11] propose a data-driven neighbor weighting methodology that leverages homophily to improve the convergence of gossip algorithms in the presence of both data and model heterogeneity. Their technique accelerates diffusion and fairness by exploiting network clustering, but it is based on full model exchanges and aggregation. The DFPL framework [12] introduces decentralized federated prototype learning for handling heterogeneous data distributions and model architectures across clients. While DFPL also focuses on prototype communication, it typically operates within a FL structure involving central aggregation or coordination, rather than pure peer-to-peer gossip protocols. GLow [13] offers a modular simulation toolkit for heterogeneous GL environments. Although GLow enables the modeling and experimental analysis of agents with heterogeneous behaviors and architectures, it does not enforce or specifically address communication constraints to prototype-level updates. Unlike prior methods requiring full-parameter exchange or central coordination, our approach uses only class-level representations, enabling greater communication efficiency in decentralized and heterogeneous environments.

B. Prototype Contrastive Learning

The concept of prototypes, typically defined as the mean of feature vectors, has been widely adopted across various learning tasks. In image classification, prototypes represent class proxies computed as averages of class-specific embeddings [14]. Similarly, in action recognition, features across video timestamps are aggregated to form a unified representation. Prototypes also serve as compact descriptors in image retrieval [15]. In the distributed learning context, prototypes have been used to address data scarcity and heterogeneity. Hoang et al. [16] represent task-agnostic knowledge using prototypes and propose fusion strategies to integrate them for novel tasks. More recently, contrastive learning has emerged as a powerful paradigm in self-supervised representation learning [17]. The core idea is to pull together an anchor and a "positive" sample in the embedding space while pushing apart the anchor from "negative" samples, typically constructed via data augmentation or random minibatch sampling. The prototype contrastive loss facilitates learning robust and discriminative class embeddings across heterogeneous clients, reducing communication and enhancing privacy.

III. Problem Formulation

Consider a decentralized learning scenario represented by a connected gossip graph $G = (V, E)$, where $|V| = C$ denotes the number of clients. Each client $i \in V$ holds a private dataset $\mathcal{D}_i$. Each local model h_i, consists of an encoder f_i (acting as a projection head with parameters θ_i) and a classifier g_i.

We assume that clients communicate via direct D2D wireless links whenever they are within mutual transmission range. In a decentralized learning setting, GL enables a population of clients to collaboratively train models without relying on a central server. During each communication round, clients exchange model parameters only with their neighbors and update their local weights through averaging. The learning objective of GL is to minimize the global optimization objective.

$$\min_{\theta} \sum_{i=1}^{|V|} \mathbb{E}_{(x,y)\sim d_i} \left[\mathcal{L}_{\text{sup}}(h_\theta(x), y) \right], \tag{1}$$

where $|V|$ denotes the total number of clients, $d_i(x, y)$ is the local data distribution of client i, and $h_\theta(x)$ is the local model parameterized by θ. The term $\mathcal{L}_{\text{sup}}$ represents a supervised loss function (e.g., cross-entropy). Repeated peer-to-peer exchanges with weighted averaging enable convergence to a global model. At each client i the parameters of client i and its neighborhood's clients parameters are aggregated as follows:

$$\theta_i^t = \sum_{j \in \mathcal{N}_i} \mu_j \, \theta_j^{t-1}, \tag{2}$$

where t is the communication round index, $\mathcal{N}_i$ denotes the set consisting of client i and its neighboring clients, and μ_j represents the contribution weight of client j, computed as $\mu_j = \frac{|D_j|}{\sum_{k \in \mathcal{N}_i} |D_k|}$.

While GL eliminates the single point of failure as in FL, it fundamentally relies on parameter aggregation. This makes it suitable only when all clients share the same architecture. However, in real world IoT, mobile, and edge environments, clients experience both data heterogeneity and model heterogeneity. To support heterogeneous decentralized environments, a learning framework must avoid full-model averaging and instead align knowledge at a semantic level. To tackle this challenge in GL, we propose GRAPE as shown in Figure 1, which replaces model-weight exchange with prototype exchange and contrastive representation alignment,

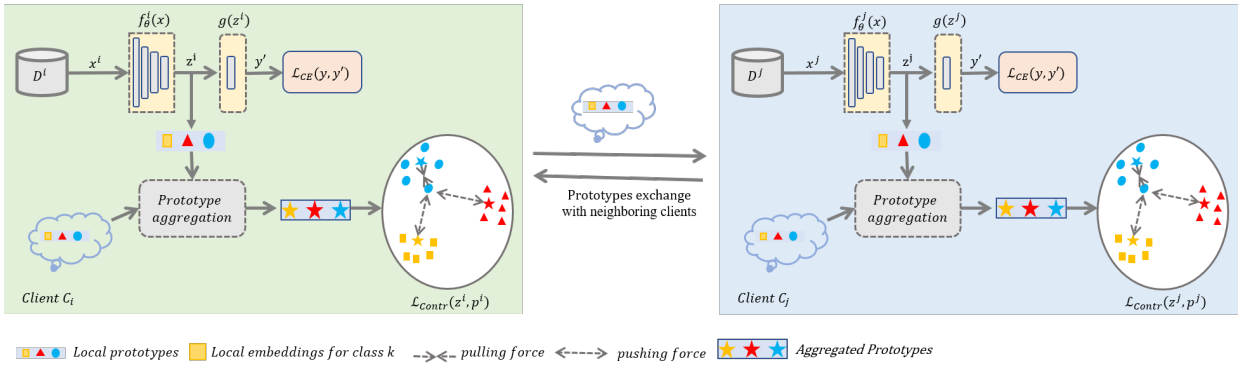


Fig. 1: In the GRAPE framework, each client trains a local encoder $f_\theta(\cdot)$ to extract embeddings. Local class prototypes are computed and aggregated with those received from neighboring peers to form a shared prototype set. A contrastive prototype loss then aligns representations by minimizing the *intra-class distance*, pulling embeddings toward their corresponding class prototype, while maximizing the *inter-class separation*, pushing them away from other class prototypes to promote well-formed and discriminative clusters.

Algorithm 1 GRAPE: Prototype-based Gossip Learning

Input: Local datasets $\{\mathcal{D}_i\}$, models $\{h_i\}$, neighbors $\mathcal{N}(i)$, rounds T, epochs E
Output: Trained models $\{h_i\}$
for each round $t = 1$ to T **do**
  for each client i in parallel **do**
    Train h_i on $\mathcal{D}_i$ for E epochs using $\mathcal{L} = \mathcal{L}_{CE} + \lambda \mathcal{L}_{InfoNCE}$ (Equation 3)
    Compute class prototypes $p_{i,k}$ from local embeddings
    Send $\{p_{i,k}\}$ to neighbors $\mathcal{N}(i)$ and receive their prototypes
    Aggregate prototypes by weighted averaging to obtain updated p_k
  end for
end for
Each client refines h_i using the final prototypes for prediction or personalization
Return: $\{h_i\}$ as the final decentralized models

enabling effective collaboration without requiring identical models while reducing the communication cost.

A. The GRAPE algorithm

To form a shared semantic space across heterogeneous clients, we eliminate costly weight synchronization and instead exchange compact class prototypes. These prototypes summarize the latent representations of local data, reducing communication while still enabling effective decentralized learning. For a class k, the prototype of client i is defined as

$$p_{i,k} = \frac{1}{|\mathcal{D}_{i,k}|} \sum_{x \in \mathcal{D}_{i,k}} \phi_i(x), \tag{3}$$

where $\mathcal{D}_{i,k}$ is the subset of samples belonging to class k and $\phi_i(\cdot)$ denotes the embedding function of client i to produce embedding z. These prototypes serve as lightweight and model agnostic knowledge units that can be communicated efficiently over the gossip network, enabling clients with different architectures to align their internal feature spaces.

To enforce representation consistency across clients, we integrate contrastive learning into the decentralized optimization process. For a sample embedding z with ground-truth label y, the corresponding class prototype acts as the positive reference in the contrastive objective. To compute these class prototypes, each client first generates embeddings for its local samples using the encoder $f_i(\cdot)$. For a given class k, the prototype is obtained by taking the mean of all embeddings belonging to that class as given in Equation 3. After local optimization, clients exchange prototypes with their neighbors and aggregate them through weighted averaging

$$p_{i,k}^{(t+1)} = \frac{\sum_{j \in \mathcal{N}_i} n_{j,k} p_{j,k}^{(t)}}{\sum_{j \in \mathcal{N}_i} n_{j,k}}, \tag{4}$$

where $n_{j,k}$ is the number of local samples of class k at client $j \in \mathcal{N}_i$ and $\mathcal{N}_i$ denotes the neighborhood of client i. To optimize this objective, we use an InfoNCE-style contrastive loss [17]. In classical contrastive learning, augmented views of the same sample (or same class) are treated as positive pairs, while augmented views from different samples or classes act as negative pairs, encouraging the model to learn representations that are invariant within a class and discriminative across classes. Following this principle, in our setting each sample embedding is paired with its corresponding class prototype as the positive sample, while prototypes of all other classes serve as negative samples thereby enhancing intra-class compactness and inter-class separation as shown in Figure 1 . The InfoNCE contrastive loss is given in the Equation 5.

$$\mathcal{L}_{InfoNCE} = -\log \frac{\exp(\text{sim}(z, p_y)/\tau)}{\sum_{k=1}^{K} \exp(\text{sim}(z, p_k)/\tau)}, \tag{5}$$

with $\text{sim}(\cdot)$ denoting cosine similarity and τ a temperature parameter. Each client trains its model by minimizing a joint objective consisting of a cross-entropy loss, which learns discriminative features from its local dataset, and a

contrastive alignment loss, which enforces consistent representations across clients despite data heterogeneity. The detailed algorithm can be seen in 1. The resulting training objective is:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda_{contr}\mathcal{L}_{InfoNCE}, \tag{6}$$

λ_{contr} is a weighting factor that controls the contribution of the contrastive loss relative to the supervised loss in the training objective.

IV. Numerical Assessment

A. Setup

We evaluate GRAPE on MNIST (70,000 grayscale 28×28 images, 10 classes) and CIFAR-10 (60,000 RGB 32×32 images, 10 classes). For CIFAR-10, we use a lightweight CNN with three convolutional blocks (ReLU and 2×2 max-pooling), followed by a projection MLP that outputs a 128-dimensional L2-normalized embedding and a linear classifier. For MNIST, we adopt a simple MLP with two ReLU-activated fully connected layers that produce a 128-dimensional L2-normalized embedding, followed by a linear classifier. To support heterogeneous models, we construct multiple encoder variants by varying the hidden layer widths while keeping the embedding dimension fixed, ensuring a shared latent space for prototype alignment. Statistical heterogeneity is simulated by partitioning each dataset across 10 clients using a Dirichlet distribution, where smaller α yields skewed label splits and larger α yields more uniform splits. We assume a fully connected topology in which every client can communicate with all others, and we train for 50 rounds with E=5 local epochs per round. We use a learning rate of 0.01, a batch size of 32, and a contrastive temperature of τ=0.1. A complete summary of hyperparameters is provided in Table I. .

B. Accuracy Assessment

We compare GRAPE with two decentralized baselines: GL, which exchanges full model parameters, and Dec-FedProto, which shares only class prototypes. Since FedProto [18] is a prototype-based method in centralized FL, we adapt it to a peer-to-peer setting for fairness, resulting in Dec-FedProto, whose communication pattern closely matches GRAPE.

Tables II and III shows the test accuracy of all methods on CIFAR-10 and MNIST datasets for different values of α, where smaller α values indicate stronger data heterogeneity across clients. Each experiment was repeated three times with different random seeds, and the mean and standard deviation is reported. The results show that when ($\alpha = 0.05, 0.1, 0.5$), the data distribution across clients becomes highly unbalanced. Consequently, GL performs poorly under such heterogeneous conditions, particularly on CIFAR-10, because exchanging full model parameters causes local updates on different clients to diverge. This leads to client drift and leading to a significant drop in overall accuracy. Dec-FedProto, which communicates class prototypes instead of full models, performs better than GL in these settings but remains limited by its MSE-based objective that minimizes intra-class distances without explicitly enforcing inter-class separation. However, the proposed GRAPE frameworks achieves the best performance because prototypes also called mean embeddings in representation space act as semantic anchors. Even if a client hardly ever sees some classes, exchanging prototypes gives it global class context without needing to average full and diverging weights. This reduces client drift and improves discrimination.

However, when α is large ($\alpha = 1$), client drift is reduced because local data distributions become more aligned. In this case, full parameter sharing enables GL to exploit complete model information, resulting in stronger performance, though at the expense of significantly higher communication overhead due to frequent full-model exchanges. Moreover, as α increases, the accuracy of Dec-FedProto drops sharply from 87.31% to 62.02%. This is because the shared prototypes aggregate increasingly diverse client information, which reduces their discriminability. Since the MSE loss only pulls samples toward their own class prototype and does not push different classes apart, the learned features gradually lose clear boundaries. In contrast, GRAPE not only outperforms Dec-FedProto across all settings, but also achieves accuracy close to GL with lower communication cost. However, the results also show that as the data becomes more IID, clients naturally learn similar feature distributions even without strong coordination; therefore, enforcing additional alignment through prototype sharing can introduce unnecessary constraints that hinder optimization and slightly reduce accuracy. For MNIST, as shown in Table III, all methods perform consistently well across different α values due to the dataset's low intra-class variance and well-separated digit clusters. Even under highly imbalanced splits, local models still learn discriminative representations.

Tables II and III also show that the performance of Het-GRAPE remains close to GRAPE, demonstrating that our approach continues to provide strong alignment even under model heterogeneity. As α increases, the variance in performance is partly due to differences in local architectures and optimizers, which naturally make coordination more difficult. Nevertheless, Het-GRAPE still preserves meaningful class separation and global consistency, indicating that our method remains effective even when clients do not share the same model structure. Overall, the results show that prototype-based contrastive learning provides a more stable, and communication-efficient framework under highly non-IID conditions, while full-model sharing offers benefits mainly in IID settings but at the cost of significantly higher communication cost.

C. Communication efficiency

In GRAPE, each client communicates only a compact set of class prototypes, requiring $C \times dim$ floating-point values per round, where C is the number of classes and dim is the prototype dimension. In our setup with $C = 10$ and $dim = 128$, this corresponds to 1280 values (approximately 5 KB) per client per round, resulting in a very low communication cost. In contrast, GL must transmit the full set of model parameters during each round. The MNIST model contains $468{,}874$

TABLE I: Hyperparameters used in all experiments.

Parameter	**Default value**
Learning rate (η)	0.01
Batch size (B)	32
Number of clients (m)	10
Number of communication rounds (T)	50
Number of local epochs (E)	5
Contrastive Loss weight (λ_{contr})	1
Temperature (τ)	0.1

TABLE II: Comparison of GRAPE with baseline methods on CIFAR 10 under different non IID data splits (different α). The best results are highlighted in bold.

Method	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 1.00$
GL	63.22 ± 0.07	68.40 ± 0.03	71.54 ± 0.01	**72.04 ± 0.01**
Dec-FedProto	87.31 ± 0.02	81.26 ± 0.02	66.88 ± 0.01	62.02 ± 0.02
GRAPE	**89.74 ± 0.01**	**84.66 ± 0.01**	**72.58 ± 0.05**	65.98 ± 0.02
Het-GRAPE	89.61 ± 0.04	78.57 ± 1.15	67.36 ± 0.15	63.47 ± 0.20

parameters and the CIFAR-10 model contains 1,307,850 parameters, which leads to significantly higher communication cost. Figure 2 also compares the accuracy over communication rounds for GL, Dec-FedProto, and GRAPE under non-IID partitions with $\alpha = 0.1$ and $\alpha = 0.05$. As shown, GL shows gradual convergence, requiring many communication rounds to reach its final performance. In contrast, Dec-FedProto and GRAPE achieve a rapid increase in accuracy within the first few rounds, reaching higher accuracy much earlier in training. This accelerated convergence is attributed to the decentralized prototype-sharing mechanism in GRAPE.

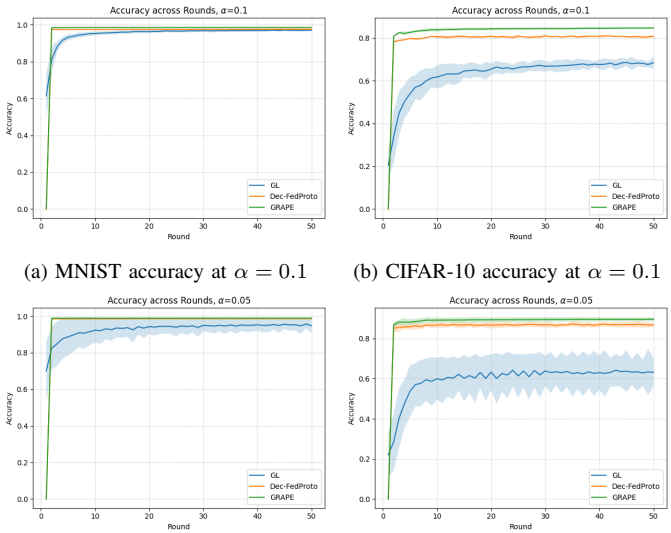
D. Ablation Studies

For ablation studies, we fix $\alpha = 0.5$ on CIFAR-10 and vary the number of local epochs, the contrastive weight λ_{contr}, and the communication probability p to evaluate GRAPE under a highly non-IID setting.

Effect of the Contrastive Weight: In GRAPE, we evaluate different values of the contrastive weight λ_{contr} to analyze its impact on accuracy and determine the optimal setting. Since GRAPE jointly optimizes a supervised cross-entropy loss and a contrastive prototype loss, λ_{contr} controls the extent to which the model enforces intra-class compactness and inter-class separation during decentralized representation learning. As shown in Figure 3a, the test accuracy increases steadily as λ_{contr} grows from 0.1 to 1.0, achieving its best performance at $\lambda_{\text{contr}} = 1.0$. In this scenario, the contrastive term is strong enough to pull embeddings toward their class prototypes while pushing them away from other classes, leading to more discriminative local representations and better global prototype alignment across clients. However, when λ_{contr} becomes too large (e.g., 3.0 or 5.0), accuracy drops sharply. In such cases, the contrastive objective dominates the training signal and suppresses the supervised cross-entropy, causing the model to over-separate classes while ignoring label structure. This imbalance weakens generalization and ultimately reduces overall performance.

TABLE III: Accuracy on MNIST for different values of α.The best results are highlighted in bold.

Method	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 1$
GL	94.84 ± 0.04	97.09 ± 0.01	**98.21 ± 0.02**	**98.31 ± 0.01**
Dec-FedProto	97.91 ± 0.01	96.65 ± 0.01	92.47 ± 0.01	91.71 ± 0.00
GRAPE	**98.95 ± 0.02**	**98.60 ± 0.01**	96.98 ± 0.03	96.41 ± 0.01
Het-GRAPE	98.70 ± 0.02	98.56 ± 0.02	96.86 ± 0.02	96.43 ± 0.09



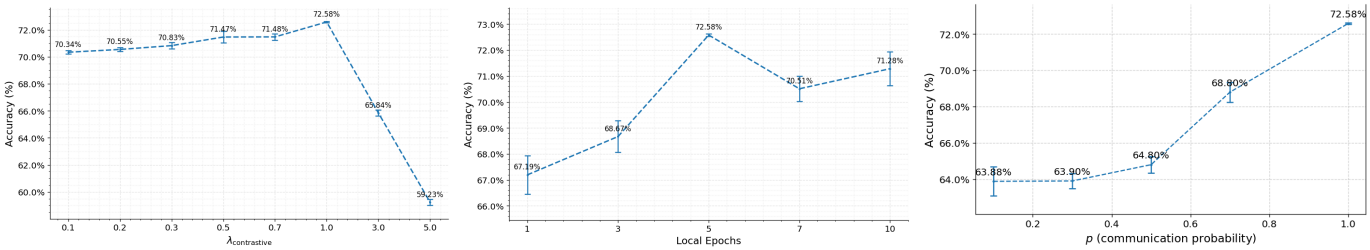
(a) MNIST accuracy at $\alpha = 0.1$ (b) CIFAR-10 accuracy at $\alpha = 0.1$

(c) MNIST accuracy at $\alpha = 0.05$ (d) CIFAR-10 accuracy at $\alpha = 0.05$

Fig. 2: Accuracy over communication rounds for MNIST and CIFAR-10 under non-IID partitions with $\alpha = 0.1$ and $\alpha = 0.05$.

Local Epochs: The results in Figure 3b shows that when each client performs only one local epoch, the accuracy remains relatively low (about 67.04%) due to immature and weakly aligned prototypes. As the number of local epochs increases to 3 and 5, the accuracy improves 72.58%, since clients perform more computation per round to refine their local embeddings and better align with their class prototypes. This results in more stable contrastive signals across the network and a more discriminative global representation. However, with 7 or 10 epochs, accuracy begins to fluctuate and slightly decline to 70.53% and 71.11%. This is because excessive local training introduces local bias, causing clients to overfit their own data distributions before exchanging prototypes, which weakens cross-client alignment.

Impact of Graph Dynamicity: We conduct all the above experiments under a fixed mesh topology in which every client can communicate with all others. To further evaluate the robustness of GRAPE, we also consider a dynamic graph setting where each client connects to other clients with probability p, and the communication graph is randomly connected at every round. The results in Figure 3c that accuracy increases as p grows. When p is small (e.g., 0.1–0.3), each device communicates with only a few neighbors, resulting in weaker and less stable contrastive signals. With fewer



(a) Accuracy vs. λ_{contr} (b) Accuracy vs. local epochs (c) Accuracy vs. communication probability p

Fig. 3: Ablation results on CIFAR-10 with $\alpha = 0.5$, illustrating the effect of λ_{contr}, local epochs, and communication probability p on GRAPE accuracy. Mean $\pm$ std over three runs.

negative prototypes, the InfoNCE loss cannot enforce strong inter-class separation, leading to reduced prototype alignment and lower accuracy. As p increases toward 0.5 and 0.7, the communication graph becomes better connected, allowing nodes to exchange more diverse and informative prototypes. When $p = 1$, every device communicates with all peers at each round, providing the maximum number of negative prototypes and the strongest InfoNCE signal. In this fully connected case, GRAPE achieves its highest accuracy because each client best aligns its representations with the global semantic structure.

V. Conclusion

In this work, we introduced GRAPE, a communication-efficient and architecture-agnostic decentralized learning framework that aligns local representations through prototype sharing and contrastive optimization. By exchanging only compact class prototypes instead of full model parameters, GRAPE achieves a $10^2 - 10^3\times$ reduction in communication cost while maintaining high accuracy and supporting heterogeneous client models. Through extensive experiments on MNIST and CIFAR-10 under varying levels of data heterogeneity, we demonstrated that GRAPE consistently outperforms GL and Dec-FedProto, particularly in challenging non-IID scenarios. Our results show that combining supervised learning with contrastive prototype alignment yields more discriminative and stable representations than regression-based prototype matching. These properties make GRAPE a practical solution for large-scale decentralized and cross-device learning, especially in bandwidth-constrained IoT environments. In future work, we will extend GRAPE to dynamic, time-varying networks and investigate adaptive and explainable prototype mechanisms.

Acknowledgments

This work is supported by the FAIR GANDEEP project. This work is partially supported by UNITY-6G project, co-funded from European Union's Horizon Europe Smart Networks and Services Joint Undertaking (SNS JU) research and innovation programme under the Grant Agreement No 101192650. This work has received funding from the Swiss State Secretariat for Education, Research and Innovation (SERI).

References

[1] P. Voigt and A. Von dem Bussche, "The eu general data protection regulation (gdpr)," *A practical guide, 1st ed., Cham: Springer International Publishing*, 2017.

[2] A. Ahmad, W. Luo, and A. Robles-Kelly, "Robust federated learning under statistical heterogeneity via hessian-weighted aggregation," *Machine Learning*, pp. 633–654, 2023.

[3] C.-Y. Huang, K. Srinivas, X. Zhang, and X. Li, "Overcoming data and model heterogeneities in decentralized federated learning via synthetic anchors," *arXiv preprint arXiv:2405.11525*, 2024.

[4] M. A. Dinani, A. Holzer, H. Nguyen, M. A. Marsan, and G. Rizzo, "A gossip learning approach to urban trajectory nowcasting for anticipatory ran management," *IEEE Transactions on Mobile Computing*, 2024.

[5] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," *arXiv preprint arXiv:1907.02189*, 2019.

[6] A. Ahmad, W. Luo, and A. Robles-Kelly, "Federated learning under statistical heterogeneity on riemannian manifolds," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2023, pp. 380–392.

[7] ——, "Robust federated learning under statistical heterogeneity via hessian spectral decomposition," *Pattern Recognition*, vol. 141, p. 109635, 2023.

[8] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *International conference on machine learning*. PMLR, 2020, pp. 5132–5143.

[9] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," in *International conference on machine learning*. PMLR, 2021, pp. 12 878–12 889.

[10] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," *Advances in neural information processing systems*, 2020.

[11] A. Ghosh and C. Mascolo, "Modeling with homophily driven heterogeneous data in gossip learning." in *IJCAI*, 2023, pp. 3741–3749.

[12] H. Zhang, F. Xu, Z. Yu, S. Pang, C. Hu, and J. Yu, "Dfpl: Decentralized federated prototype learning across heterogeneous data distributions," *arXiv preprint arXiv:2505.04947*, 2025.

[13] A. Belenguer, J. A. Pascual, and J. Navaridas, "Glow–a novel, flower-based simulated gossip learning strategy," *arXiv preprint arXiv:2501.10463*, 2025.

[14] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[15] A. Babenko and V. Lempitsky, "Aggregating local deep features for image retrieval," in *Proceedings of the IEEE international conference on computer vision*, 2015.

[16] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE journal on selected areas in communications*, 2019.

[17] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in neural information processing systems*, vol. 33, pp. 18 661–18 673, 2020.

[18] Y. Tan, G. Long, L. Liu, T. Zhou, Q. Lu, J. Jiang, and C. Zhang, "Fedproto: Federated prototype learning across heterogeneous clients," in *Proceedings of the AAAI conference on artificial intelligence*, 2022.