Multi-order Orchestrated Curriculum Distillation for Model-Heterogeneous Federated Graph Learning

Guancheng Wan[1†], **Xu Cheng**[1†], **Run Liu**[1], **Wenke Huang**[1], **Zitong Shi**[1], **Pinyi Jin**[1], **Guibin Zhang**[2], **Bo Du**[1*], **Mang Ye**[1*]
[1]Wuhan University [2]NUS
{guanchengwan, yemang}@whu.edu.cn

Abstract

Federated Graph Learning (FGL) has been shown to be particularly effective in enabling collaborative training of Graph Neural Networks (GNNs) in decentralized settings. Model-heterogeneous FGL further enhances practical applicability by accommodating client preferences for diverse model architectures. However, existing model-heterogeneous approaches primarily target Euclidean data and fail to account for a crucial aspect of graph-structured data: topological relationships. To address this limitation, we propose `TRUST`, a novel knowledge distillation-based model-heterogeneous FGL framework. Specifically, we propose Progressive Curriculum Node Scheduler to progressively introduce challenging nodes based on learning difficulty. In Adaptive Curriculum Distillation Modulator, we propose an adaptive temperature modulator that dynamically adjusts knowledge distillation temperature to accommodate varying client capabilities and graph complexity. Moreover, we leverage Wasserstein-Driven Affinity Distillation to enable models to capture cross-class structural relationships through optimal transport. Extensive experiments on multiple graph benchmarks and model-heterogeneous settings show that `TRUST` outperforms existing methods, achieving an average 3.6% $\uparrow$ performance gain, particularly under moderate heterogeneity conditions. The code is available for anonymous access at `https://github.com/GuanchengWan/TRUST`.

1 Introduction

Federated Learning (FL) [20, 19, 18, 43] has emerged as a distributed machine learning paradigm that enables multiple clients to collaboratively train a global model without sharing their privacy-sensitive data, thereby preserving data confidentiality. Traditional FL methods operate by aggregating locally computed model updates (*e.g.*, gradients or weights) from participating clients under the coordination of a central server, eliminating the need for direct data exchange. A prominent branch of FL is Federated Graph Learning (FGL) [53, 6, 41, 3, 5, 44, 42], which specializes in handling graph-structured data. In addition to inheriting privacy-preserving benefits of FL, FGL usually leverages Graph Neural Networks (GNNs) [21, 11, 45, 30] to capture topology information in graph data, offering a flexible and expressive framework for modeling graph-structured information.

Although many existing FGL works have made significant progress in improving the performance of the global model, these studies are often based on the assumption that client models follow the same architecture. This assumption rarely holds in real-world scenarios, where computational resources and task requirements vary across participants and they prefer to design private models independently rather than agreeing on a unified model architecture [54], ultimately restricting their real-world

[†] Equal Contribution.
[*] Corresponding Author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

applicability. This challenge is formally termed **model-heterogeneous federated learning**. To address this challenge, recent works have proposed several solutions. For instance, pFedHR [48] generates a personalized model for each client through model reassembly to transfer knowledge between clients. DESA [15] leverages synthetic global data to distill knowledge from other client models. FedTGP [56] employs server-side maintained global prototypes to bridge heterogeneous models. However, these methods are primarily tailored for traditional data types like images and do not generalize to non-Euclidean graph-structured data. This gap motivates our core research question:

How can we design a model-heterogeneous FL framework specifically tailored for graph data?

Some previous model-heterogeneous FL methods leverage knowledge distillation (KD) [13, 9, 50, 45] to transfer knowledge between clients [16, 24, 54]. For example, FedType [49] introduces small identical proxy models for clients to bridge the global architecture discrepancy, and then they leverage KD to transfer knowledge between large private and small proxy models. However, in the model-heterogeneous FGL scenario, there are significant differences in model architectures and computational capabilities between clients, and the graph data itself has high-order information such as complex non-Euclidean topology, multi-hop path pattern, and community structure. Traditional knowledge distillation methods maintain a constant "distillation difficulty" (fixed task complexity) [10, 58] throughout the training process. This *one-size-fits-all* approach creates dual dilemmas: For lightweight models, the early introduction of higher-order topology information causes learning bottlenecks and slow convergence. Conversely, for large-scale models with strong expressive ability, it may be too easy for them to exploit their full capacity, resulting in inadequate capture of global structure information. This *fixed-difficulty paradigm* cannot take into account the requirements brought about by model heterogeneity, preventing progressive learning and ultimately leading to degraded FGL Performance. This naturally raises the following question: **I)** ***How can we design a KD strategy that dynamically adjusts the task difficulty to accommodate heterogeneous architectures and complex graph topologies?***

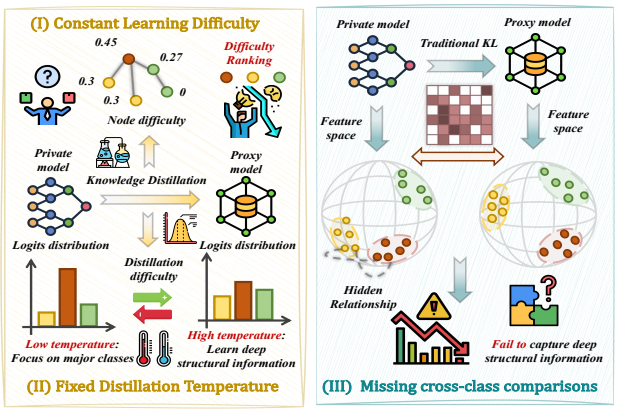


Figure 1: **Problem Illustration**. We describe three challenges **model-heterogeneous FGL** encounters: constant learning difficulty and fixed distillation temperature limit client-specific adaptation and topology preservation. And the lack of cross-class comparison impedes the capture of deep structural graph relationships.

In addition to the KD task itself, current KL-divergence based KD approaches maintain static temperature parameters (controlling label distribution smoothness), which govern transfer intensity. This rigidity prevents adaptation to varying graph complexity, label sparsity, and noise patterns. To be specific, KD at low temperatures will neglect small probability events, blurring subtle but important structural information like multi-hop paths and community features. In contrast, high temperatures lose fine-grained node-level information. This inflexibility in distillation signaling hinders client differentiation and deep topological knowledge transfer, ultimately limiting the capacity of the proxy model. This leads us to think that: **II)** ***How can we devise an adaptive KD scheme that dynamically calibrates temperature to harmonize global topology transfer with fine-grained node details?*** Furthermore, the mentioned KD methods rely on KL-divergence loss, which only compares probability distributions within the same class. This approach lacks cross-class comparison mechanisms, making proxy models fail to capture topological relationships across different categories. This limitation leads to the third problem: **III)** ***How can we enable cross-class comparison to fully leverage deep structural information during the KD phase?***

To address these challenges, inspired by curriculum learning [2, 51], we propose Mul**T**i-o**R**der Orchestrated C**U**rriculum Di**ST**illation (`TRUST`), a novel model-heterogeneous FGL framework. For Problem **I**), we introduce the **Progressive Curriculum Node Scheduler (PCNS)**, which progressively schedules node samples for each client from easy (nodes with typical class representations) to difficult (nodes situated near class boundaries that may confuse proxy models), thereby enabling heterogeneous

models to first absorb low-order semantic cues and then incrementally acquire high-order topological knowledge. For Issue **II)**, we propose the **Adaptive Curriculum Distillation Modulator (ACDM)**, a module that dynamically calibrates the distillation temperature during training—this mechanism allows the framework to fluidly shift emphasis between capturing global structures and preserving fine-grained node details. For Problem **III)**, in addition to the KL-Divergence loss, we introduce **Wasserstein-Driven Affinity Distillation (WDAD)**, which leverages class prototypes from private models and computes cross-class relational distances via the Wasserstein metric, thereby enabling comprehensive cross-class topological knowledge transfer. The contributions are as follows.

❶ ***Problem Identification.*** We are the first to systematically study **model-heterogeneous FGL** and formally characterize three core challenges in KD-based methods: the need for dynamic task difficulty, adaptive distillation signaling, and cross-class relational transfer.

❷ ***Practical Solution.*** We develop a curriculum-guided distillation framework that progressively schedules node difficulty, dynamically adjusts distillation strength, and enables knowledge transfer across classes, effectively reconciling heterogeneous architectures with complex graph topologies.

❸ ***Experimental Validation.*** We conduct extensive experiments on multiple graph benchmarks under diverse architecture heterogeneity settings. Empirical results demonstrate that `TRUST` consistently outperforms state-of-the-art baselines.

2 Preliminaries

2.1 Notations

Graph Neural Networks. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of nodes and $\mathcal{E}$ represents the set of edges. For every node $v_i \in \mathcal{V}$, v_i is associated with a k-dimensional feature vector x_i. The feature vectors of all nodes are represented collectively as the feature matrix $\mathbf{X} \in \mathbb{R}^{N\times k}$. The topology of the graph $\mathcal{G}$ is encoded in the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ where $\mathbf{A}(v,u) = 1$ if $(v,u) \in \mathcal{E}$ and $\mathbf{A}(v,u) = 0$ otherwise. GNN models operate through iteratively updating node representations using a message-passing mechanism, where each node aggregates information from its local neighborhood and updates its own state. Let $\mathcal{N}_i$ denote the neighborhood nodes of v_i, and for the l-th layer of a GNN model, the representation of h_i^l of node v_i can be computed as:

$$h_i^l = \text{Update}\Big(h_i^{l-1}, \text{ Aggregate}\big(\{h_j^{l-1} | v_j \in \mathcal{N}_i\}\big)\Big), \tag{1}$$

where h_i^l denotes the representation of node v_i at layer l.

Knowledge Distillation Traditional knowledge distillation methods employ the Kullback-Leibler (KL) Divergence loss to align the output distributions of student and teacher models:

$$L_{KL}(p^T, p^S) = \sum_i p^T(i) \log \frac{p^T(i)}{p^S(i)}, \tag{2}$$

where p^T and p^S denote the class probability distributions predicted by the teacher and student model respectively. These are computed via softmax function σ and distillation temperature τ:

$$p^T = \sigma(\frac{h^T}{\tau}), \quad p^S = \sigma(\frac{h^S}{\tau}). \tag{3}$$

Model Heterogeneous Federated Graph Learning framework. Let S denotes the central server and C^k denotes the k-th client with K clients in total. Each client k holds its own graph $\mathcal{G}^k = (\mathcal{V}^k, \mathcal{E}^k)$. In a model heterogeneous setting, each client trains a model w^k parameterized by θ^k on its own training data and then uploads it to the server, and the client models $\{w^1, w^2, ..., w^K\}$ do not share identical architectures. Mathematically, the learning objective can be formulated as:

$$\min_\theta \sum_{k=1}^{K} \frac{|\mathcal{V}^k|}{|\mathcal{V}|} L^k(y, w^k(\mathcal{G}^k)), \tag{4}$$

where $|\mathcal{V}^k|$ and $|\mathcal{V}|$ represents the number of samples of the client k and the total number of samples across all clients respectively, y denotes ground truth labels, and $\mathrm{L}(\cdot,\cdot)$ is the empirical loss.

2.2 Model Heterogeneous FGL with proxy model

Inspired by FedType [49], we leverage proxy models to bridge heterogeneous models. In our framework, each client maintains both its private model and a proxy model. The proxy model is a small model with an identical architecture across all clients. The training process consists of three key phases: (1) Forward Distillation: First, each client distills knowledge from its private model to its

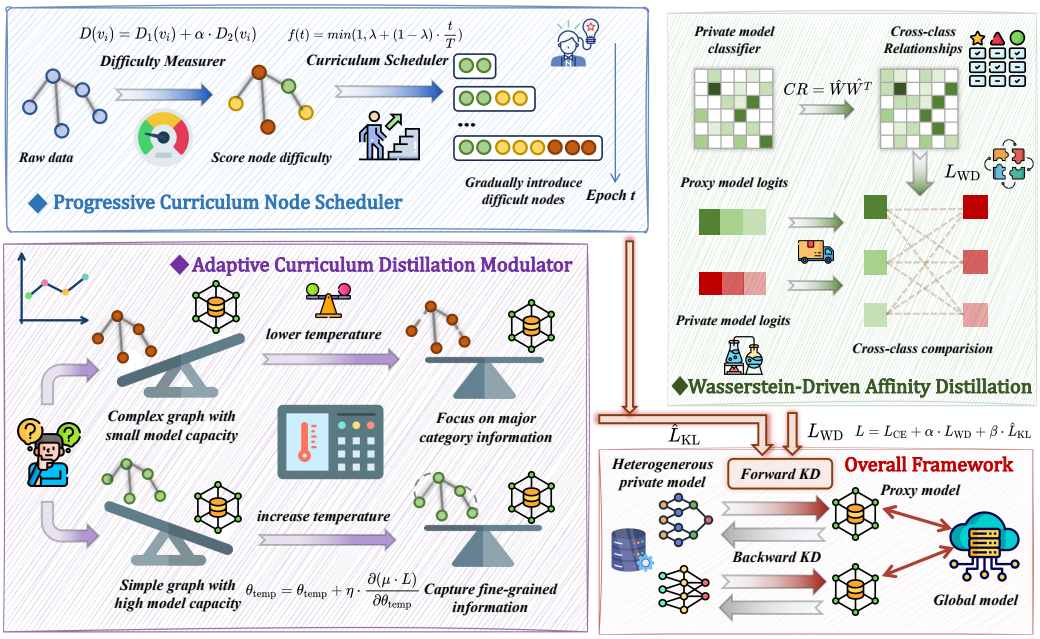


Figure 2: Architecture overview of TRUST, which includes three core components: (1) **PCNS** is a difficulty-progressive curriculum learning module. (2) **ACDM** provides capability-aware dynamic temperature adjustment. (3) **WDAD** employs Wasserstein-based cross-class distillation.

proxy model. (2) Global Aggregation: Then the weights of all proxy models are transmitted to the server where they are aggregated through weighted averaging to form a global model. (3) Backward Distillation: Next, the updated global model is distributed back to proxy models, which conducts knowledge distillation to transfer global knowledge to private models using a conformal model. The details of backward distillation are given in Section E in Appendix. This cyclic process continues until the global model converges. However, FedType is originally tailored for Euclidean data types as it does not fully exploit unique topological information of graph data. In order to generalize to graph data, our approach specifically addresses the three key challenges identified in Section **1**.

3 Methodology

3.1 Framework Overview

In this section, we present an overview of TRUST. TRUST adds three key components to the knowledge distillation process: (1) At the client side, during knowledge transfer from private to proxy models, we propose a PCNS to gradually introduce challenging nodes to proxy models. (2) At the same time, distillation temperature automatically calibrates throughout the process to adjust to both client model capabilities and local graph complexity. (3) After forward propagation, we introduce a WDAD loss, which incorporates cross-class comparison through optimal transport theory, preserving topological information that conventional distillation methods typically ignore. These components work synergistically to address the unique challenges of model-heterogeneous federated learning on graph data, while maintaining the privacy-preserving benefits of the federated paradigm. The framework is illustrated in Figure 2.

3.2 Progressive Curriculum Node Scheduler

Motivation. As established in Sec. 1, graph-structured data contains complex topological relationships. Therefore, premature exposure to this intricate high-order information during initial training phases results in client models struggling to learn generalizable patterns from highly complex nodes, and nodes situated near class boundaries with ambiguous class representations may even lead to misleading learning signals. Therefore, we need to design a progressive learning framework to introduce complex graph knowledge step by step.

Curriculum Learning. Drawing inspiration from human cognitive development, Curriculum Learning (CL) enables models to learn from data in a structured manner, transitioning from simpler to more complex samples, rather than processing all data uniformly in each epoch [2]. Prior works have proved that CL's effectiveness in improving model convergence, generalization, and final performance [23]. To implement this approach, we need to (1) make a formal definition of sample "difficulty" and (2) schedule data samples based on the proposed definition.

Difficulty Measurer. In GNN, each layer aggregates neighboring nodes to update node representations, as formalized in Equation 1. Intuitively, GNN models are more proficient at learning nodes whose neighbors belong to the same class because they share certain features. By contrast, for cross-class nodes connected to neighbors with divergent labels, GNN models receive conflicting gradient updates during aggregation, resulting in impaired learning. Therefore, we can quantify node learning difficulty through the neighborhood label distribution, where for a node v_i, the node difficulty is formally defined as the entropy of the distribution:

$$\begin{aligned} P_c(v_i) &= \frac{|\{y_n = c \mid n \in \mathcal{N}_i \cup \{v_i\}\}|}{|\mathcal{N}_i \cup \{v_i\}|}, \\ D_1(v_i) &= -\sum_{c \in C} P_c(v_i) \log(P_c(v_i)), \end{aligned} \tag{5}$$

where $P_c(v_i)$ computes proportion of class c in neighborhood $\mathcal{N}_i \cup \{v_i\}$, y_n denotes the label of node n, and C represents the set of labels.

Notably, the neighbors of node v_i may include samples from both the training and test datasets, and theoretically, the labels of test dataset neighbors are unavailable during the training phase, which means we need a locally pretrained GNN to generate pseudo-labels for unlabeled nodes. However, this approach introduces a fundamental challenge in decentralized FGL settings: locally pretrained GNN models may be unreliable due to the lack of global knowledge, potentially resulting in pseudo-label inaccuracies. Therefore, our difficulty measurer must exhibit robustness to potential label noise. Intuitively, we can implement this by evaluating the alignment between node representations and class prototypes. For a node v_i with pseudo-label y_i, if its node representation h_i demonstrates high similarity with prototype of class y_i, it is less likely to be assigned a wrong label because its features exhibit strong class-typical characteristics. But if it's the opposite, we can identify it as a "difficult" node because it has ambiguous node representation and the pretrained model has low confidence in the predicted pseudo-label. We can summarize this difficulty measurer as:

$$\begin{aligned} p_{y_i} &= \frac{1}{|V_{y_i}|} \sum_{v \in V_{y_i}} h_v, \\ D_2(v_i) &= 1 - \frac{\exp(h_i \cdot p_{y_i})}{\max_{c \in C} \exp(h_i \cdot p_c)}, \end{aligned} \tag{6}$$

where V_{y_i} is a subset of V with all nodes belonging to class y_i in it, and p_{y_i} denotes the prototypes of class y_i (mean embedding of nodes in V_{y_i}). Combining these two difficulty measurer, the overall node difficulty score can be formalized as follows:

$$D(v_i) = D_1(v_i) + \alpha \cdot D_2(v_i), \tag{7}$$

where α balances the weight of $D_2(v_i)$.

Curriculum scheduler. Once data samples are sorted by node difficulty in ascending order, each client implements a curriculum scheduler to gradually expose proxy model to more complex samples. The scheduler regulates the proportion of sorted training data used at each epoch t through a pacing function. For simplicity, we adopt a linear function:

$$f(t) = \min(1, \lambda + (1 - \lambda) \cdot \frac{t}{T}), \tag{8}$$

where λ denotes the available proportion of data samples at epoch 0, and T is the epoch when full training data is first utilized. $f(t)$ monotonically increases from λ to 1 over T epochs. Notably, after $f(t)$ reaches 1, the model should still continue training for several additional epochs to ensure complete assimilation of challenging knowledge patterns.

Incorporating the proposed curriculum scheduler, we can reformulate the KL divergence loss as:

$$\hat{L}_{\mathrm{KL}}(p^T, p^S) = \frac{1}{B} \sum_{j=1}^{B} \sum_i p_j^T(i) \log \frac{p_j^T(i)}{p_j^S(i)}, \tag{9}$$

where, for a given client k, the nodes are pre-sorted as $v_1, v_2, \ldots, v_{|\mathcal{V}^k|}$ in ascending order of their difficulty $D(v_i)$. At epoch t, the number of nodes selected from the beginning of this sorted list is $B_t = \lfloor f(t) \cdot |\mathcal{V}^k| \rfloor$. In Equation 9, B corresponds to this B_t at epoch t.

3.3 Adaptive Curriculum Distillation Modulator

Motivation. In model-heterogeneous FGL, each client adjusts local model architecture based on its available computational resources, and each client operates on a subgraph with varying complexity. Fixed temperature fails to account for differences in client model capabilities and topological knowledge complexity, limiting the ability of proxy models to learn intricate graph structures. To address this issue, we need to propose an ACDM that dynamically calibrates temperature.

Adaptive Temperature Modulator. To dynamically adjust distillation signals during training, intuitively if the current task difficulty is too low for a client model, increasing the difficulty can fully exploit model capability. Conversely, if the task exceeds the model's current capacity, reducing the difficulty prevents ineffective training, which resembles an adversarial process. Inspired by Generative Adversarial Networks (GANs) [25, 8], we implement this mechanism by converting the constant temperature value into a learnable parameter θ_{temp}, which is optimized in the opposite direction of the client model parameters. In this way, θ_{temp} controls the difficulty of the loss minimization process, thereby enabling indirect dynamic adjustment of the distillation difficulty. Mathematically, the learning objective can be formulated as:

$$\theta_{\text{model}} = \theta_{\text{model}} - \eta \cdot \frac{\partial L}{\partial \theta_{\text{model}}}, \quad \theta_{\text{temp}} = \theta_{\text{temp}} + \eta \cdot \frac{\partial L}{\partial \theta_{\text{temp}}}, \tag{10}$$

where θ_{model} denotes the client model parameters (the proxy model more precisely in this framework), θ_{temp} is the temperature parameter, and η denotes the learning rate.

Curriculum Distillation Modulator. While the Adaptive Temperature Modulator provides dynamic adjustment, we decide to prevent excessive interference at the early training stage since the model is still initializing and has limited learning capacity. Therefore, we incorporate curriculum learning to progressively increase the influence of the modulator. Specifically, we scale the loss L by a factor μ:

$$\theta_{\text{temp}} = \theta_{\text{temp}} + \eta \cdot \frac{\partial(\mu \cdot L)}{\partial \theta_{\text{temp}}}. \tag{11}$$

Here, μ is determined by the pacing function, which smoothly increases from 0 to 1 over the training epochs. We implement the pacing function as a cosine scheduler to progress more smoothly:

$$\mu = \frac{1 - cos(\frac{min(t,T)}{T} \cdot \pi)}{2}, \tag{12}$$

where t denotes the current epoch.

Module Pipeline. Having established the optimization objective of ACDM, we now formalize its overall pipeline. The module can be conceptualized as a network layer denoted as l_{temp} parameterized by θ_{temp}. Each training epoch executes the following steps: (1) we first compute the scaling factor μ using Equation 12. (2) During forward propagation, l_{temp} takes μ as input and outputs θ_{temp} to compute the distillation temperature τ for current epoch. (3) In backward propagation, l_{temp} updates θ_{temp} via gradient descent using Equation 11. This cyclic process continues throughout the entire training phase. Notably, rather than directly use θ_{temp} as the distillation temperature τ, we constrain it to a reasonable range via a sigmoid function. Therefore, KL-divergence loss is reformulated as:

$$\begin{aligned} \hat{L}_{\text{KL}}(p^T, p^S) &= \frac{1}{B} \sum_{j=1}^{B} \sum_{i} \hat{p}_j^T(i) \log \frac{\hat{p}_j^T(i)}{\hat{p}_j^S(i)}, \\ \hat{p}_j^T &= \sigma(\frac{h_i^T}{\tau}), \quad \hat{p}_j^S = \sigma(\frac{h_i^S}{\tau}), \\ \tau &= \tau_{min} + \tau_{max} \cdot sigmoid(\theta_{temp}), \end{aligned} \tag{13}$$

where τ_{min} and τ_{max} are the upper bound and lower bound for distillation temperature.

3.4 Wasserstein-Driven Affinity Distillation

Motivation. As shown in Equation 2, KL-divergence only compares intra-class probability distributions. For graph data with complex topological relationships, this approach may fail to capture

Table 1: **Comparison with the state-of-the-art methods** on five real-world datasets under moderate heterogeneity. For each dataset, we report local and global accuracy(%) (with red/green markers indicating regression/improvement over FedAvg). The best and second-best results are marked with **bold** and underline, respectively. Additional results under more settings are in Appendix D.

Category	Methods	Cora		CiteSeer		PubMed		CS		Photo	
	acc Type	local	global	local	global	local	global	local	global	local	global
FL	FedAvg [ASTAT17]	81.36	64.52	82.61	65.48	88.10	82.09	90.10	83.35	90.14	84.10
	FedNOVA [NeurIPS20]	81.54$_{\uparrow 0.18}$	64.97$_{\uparrow 0.45}$	82.76$_{\uparrow 0.15}$	66.22$_{\uparrow 0.74}$	88.20$_{\uparrow 0.10}$	82.87$_{\uparrow 0.78}$	90.13$_{\uparrow 0.03}$	82.37$_{\downarrow 0.98}$	90.34$_{\uparrow 0.20}$	86.70$_{\uparrow 2.60}$
	FedProto [AAAI22]	79.17$_{\downarrow 2.19}$	64.79$_{\uparrow 0.27}$	82.61$_{\uparrow 0.00}$	67.24$_{\uparrow 1.76}$	88.10$_{\uparrow 0.00}$	83.46$_{\uparrow 1.37}$	91.97$_{\uparrow 1.87}$	80.81$_{\downarrow 2.54}$	86.38$_{\downarrow 3.76}$	84.04$_{\downarrow 0.06}$
	MOON [CVPR21]	81.52$_{\uparrow 0.16}$	65.70$_{\uparrow 1.18}$	81.58$_{\downarrow 1.03}$	63.84$_{\downarrow 1.64}$	88.15$_{\uparrow 0.05}$	82.34$_{\uparrow 0.25}$	91.78$_{\uparrow 1.68}$	83.81$_{\uparrow 0.46}$	90.40$_{\uparrow 0.26}$	86.18$_{\uparrow 2.08}$
	FedType [ICML24]	82.25$_{\uparrow 0.89}$	72.96$_{\uparrow 8.44}$	83.20$_{\uparrow 0.59}$	64.24$_{\downarrow 1.24}$	87.39$_{\downarrow 0.71}$	84.34$_{\uparrow 2.25}$	91.64$_{\uparrow 1.54}$	86.36$_{\uparrow 3.01}$	88.91$_{\downarrow 1.23}$	90.17$_{\uparrow 6.07}$
FGL	AdaFGL [ICDE24]	81.93$_{\uparrow 0.57}$	64.61$_{\uparrow 0.09}$	80.40$_{\downarrow 2.21}$	65.48$_{\uparrow 0.00}$	85.35$_{\downarrow 2.75}$	82.87$_{\uparrow 0.78}$	91.11$_{\uparrow 1.01}$	84.24$_{\uparrow 0.89}$	90.64$_{\uparrow 0.50}$	82.54$_{\downarrow 1.56}$
	FedGTA [VLDB24]	77.73$_{\downarrow 3.63}$	64.43$_{\downarrow 0.09}$	79.96$_{\downarrow 2.65}$	66.18$_{\uparrow 0.70}$	85.62$_{\downarrow 2.48}$	82.57$_{\uparrow 0.48}$	91.80$_{\uparrow 1.70}$	83.89$_{\uparrow 0.54}$	90.87$_{\uparrow 0.73}$	84.56$_{\uparrow 0.46}$
Model-Heterogeneous FGL	TRUST	**83.90**$_{\uparrow 2.54}$	**75.32**$_{\uparrow 10.80}$	**85.84**$_{\uparrow 3.23}$	**67.43**$_{\uparrow 1.95}$	**89.06**$_{\uparrow 0.96}$	**84.57**$_{\uparrow 2.48}$	**92.19**$_{\uparrow 2.09}$	**87.07**$_{\uparrow 3.72}$	**91.42**$_{\uparrow 1.28}$	**91.50**$_{\uparrow 7.40}$

subtle but important cross-class structural dependencies. To address this limitation, we propose a cross-class comparison mechanism.

Cross-class Relationships. We quantify cross-class relationships (CR) by computing cosine similarity between prototypes of different categories. Following Equation 6, prototypes can be derived from the mean node embeddings for each category. Drawing inspiration from [37], for implementation efficiency we decide to instead adopt the classifier weight vectors of the private model to compute class prototypes. For categories with certain connections, their classifier weight vectors also share certain similarity. Let $W \in \mathbb{R}^{c \times n}$ denotes the classifier weight matrix, where c is the number of classes and n is the feature dimensions. The CR metric is computed as:

$$\hat{w_i} = \frac{w_i}{||w_i||_2}, \quad CR = \hat{W}\hat{W}^T, \tag{14}$$

where w_i represents the weight vector for class i in W , $\hat{w_i}$ denotes the L_2-normalized weight vector of w_i, and CR captures the pairwise similarity between all classes.

Wasserstein-Driven Affinity Distillation Loss. To quantify the probability distribution difference between p^T and p^S, based on Wasserstein distance [4], we define WDAD loss as:

$$L_{\text{WD}}(p^T, p^S) = \min \sum_{i,j} c_{ij} q_{ij} + \eta \cdot q_{ij} \log q_{ij}, \tag{15}$$

where q_{ij} and c_{ij} represents the transmitted probability mass and transmission cost from teacher (private model in forward distillation) category i to student (proxy model) category j respectively, and η is a regularization hyper-parameter. q_{ij} is constrained by:

$$\sum_j q_{ij} = p_i^T, \quad \sum_i q_{ij} = p_j^S, \quad q_{ij} \geq 0. \tag{16}$$

Intuitively, if two categories are similar in the feature space, the transmission cost c_{ij} should be lower. Therefore, we can leverage CR to compute c_{ij} using a Gaussian kernel:

$$c_{ij} = 1 - \exp(-\kappa(1 - CR_{ij})), \tag{17}$$

where κ is a hyper-parameter adjusting the sensitivity to CR. Overall, WDAD loss explicitly introduces cross-class relationships through transmission cost c_{ij}. By minimizing this loss, we enforce consistency between private and proxy models in their probability allocation, particularly for categories that are similar in the feature space.

Moreover, rather than completely replace $\hat{L}_{\text{KL}}$ with L_{WD}, we decide to employ a weighted combination of L_{WD} and $\hat{L}_{\text{KL}}$ to ensure smooth transition between the two objectives. Therefore, our **final optimization objective** can be formulated as:

$$L = L_{\text{CE}} + \alpha \cdot L_{\text{WD}} + \beta \cdot \hat{L}_{\text{KL}}, \tag{18}$$

where L_{CE} denotes the cross-entropy function for classification tasks, and α and β are two hyper-parameters regulating the weight of L_{WD} and $\hat{L}_{\text{KL}}$ respectively.

4 Experiment

In this section, we comprehensively evaluate TRUST through four axes: **Q1** (Superiority), **Q2** (Resilience). **Q3** (Effectiveness), **Q4** (Sensitivity),

4.1 Experimental Setup

Datasets. To effectively evaluate the performance of our approach, we employed five benchmark graph datasets of various scales and distributions, including Cora [31], CiteSeer [7], PubMed [38], CS, and Photo. Detailed descriptions and splits for these datasets can be found in Appendix C.1. Moreover, the implementation details and parameter settings can be found in Appendix C.3.

Counterparts. We compare `TRUST` against several traditional FL methods: (1) **FedAvg** [ASTAT17] [32], (2) **FedNOVA** [NeurIPS20] [47], (3) **FedProto**[AAAI22] [39], (4) **MOON** [CVPR 21][26],(5) **FedType** [ICML24] [49]; two popular FGL approaches: (6) **AdaFGL** [ICDE24] [28]; (7) **FedGTA** [VLDB24] [29].Detailed descriptions can be found in Appendix C.2.

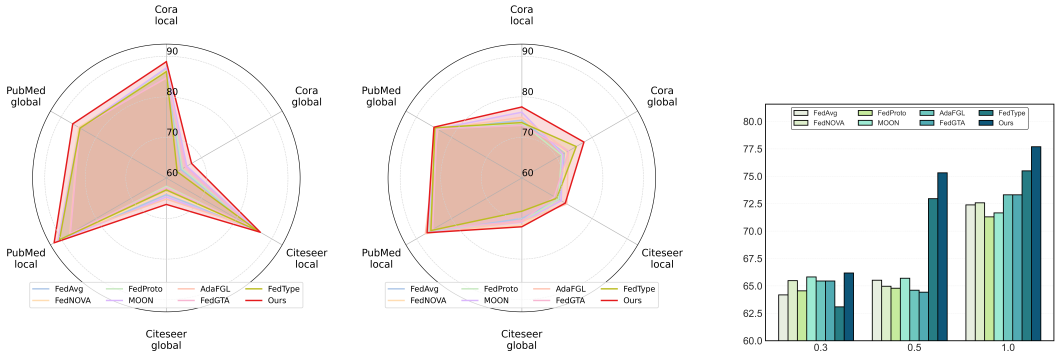


Figure 3: We report the performance of different methods under varying data heterogeneity levels on Cora, CiteSeer, and PubMed, with both local and global accuracy. The red color denotes the performance of `TRUST`. (*First*): severe heterogeneity(α=0.3). (*Second*): mild heterogeneity(α=1.0). (*Third*): Results on Cora under different heterogeneity levels with α set to 0.3, 0.5, and 1.0.

4.2 Superiority

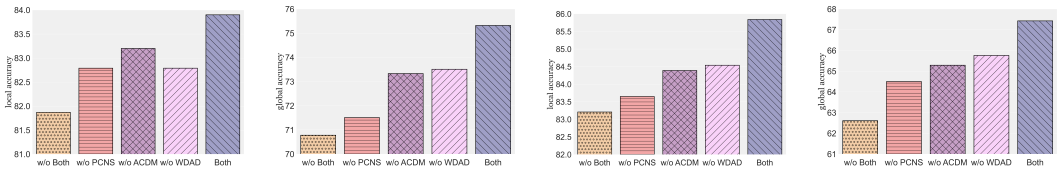
To answer **Q1**, we conducted systematic experimental evaluations in a variety of subgraph data heterogeneity environments, which we control by partitioning the graph using a Dirichlet distribution with parameter α. Smaller values of α lead to more extreme subgraph distributions. We compared the performance of `TRUST` with existing approaches based on two metrics: local accuracy and global accuracy. Local accuracy is measured as the average classification accuracy across all clients, reflecting the model's effectiveness on decentralized subsets. Global accuracy is evaluated on the aggregated model at the server. For most existing methods, the global model is updated via knowledge distillation. In contrast, both `TRUST` and FedType update the global model by simply averaging the parameters of the proxy models. The results are summarized in Tab. 1 and Figure 3.

From the table, several key observations can be made (**Obs.**): **Obs. ❶Existing approaches show suboptimal performance in heterogeneous FGL scenarios.** For instance, when α is set to 0.3, most existing methods achieve accuracy on the Cora dataset that is either lower than or comparable to FedAvg. Notably, when α is reduced to 0.1, the performance of these methods drops significantly, with the average global accuracy consistently falling below 35%.

Obs. ❷ `TRUST` demonstrates remarkable robustness across various graph data heterogeneity scales. Under moderate heterogeneity conditions ($\alpha = 0.5$), `TRUST` exhibits clear advantages. As shown in Tab. 1, `TRUST` consistently outperforms both FL and FGL baseline methods across different datasets in terms of both local and global accuracy. For example, `TRUST` achieves a global accuracy of 75.32% on the Cora dataset, surpassing the best baseline method, FedType (72.96%) by 2.36%, and outperforming FedAvg (64.52%) by a significant 10.80% margin. Furthermore, as shown in Figure 3, `TRUST` consistently outperforms all baselines across various heterogeneity levels. In highly heterogeneous environments,`TRUST` achieves varying degrees of performance improvement over all baselines .In mildly heterogeneous settings, it demonstrates an average accuracy gain of 2.25%.

4.3 Resilience

To address **Q2**, we evaluate the performance of each method on the Cora dataset across varying levels of data heterogeneity, where α is set to 0.3, 0.5, and 1.0.Figure 3(Third) illustrates that `TRUST`



(a) Cora Local (b) Cora Global (c) Citeseer Local (d) Citeseer Global

Figure 4: **Ablation Study** of the key components PCNS,ACDM and WDAD on Cora and Citeseer datasets. For an in-depth analysis, please refer to Sec. 4.4.

achieves robust performance gains across varying levels of data heterogeneity, outperforming other algorithms by an average of 5.78%, and at $\alpha = 0.5$ on the Cora dataset, it even surpasses FedAvg by 10.80%.This demonstrates that `TRUST` can effectively identify heterogeneous graphs and maintain superior performance, even under challenging conditions with extreme data heterogeneity.

4.4 Effectiveness

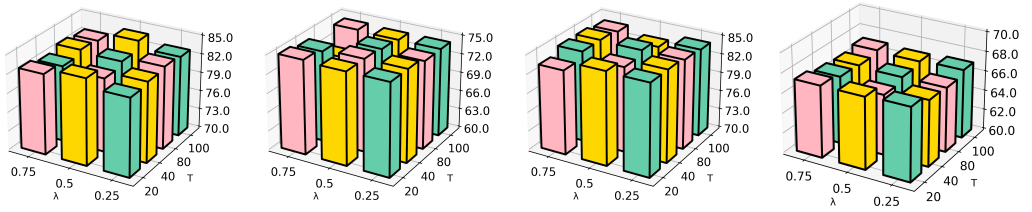
To address **Q3**, we conduct an ablation study on the key components of the framework: PCNS, ACDM, and WDAD. Experimental results on Cora and CiteSeer are presented in Figure 4.

The ablation study demonstrates that all three components contribute substantially to performance improvement. PCNS has the most pronounced individual effect; its removal results in a 2.9% drop in accuracy (Citeseer Global: 67.43% $\rightarrow$ 64.51%), confirming its effectiveness in progressive node scheduling. ACDM provides dataset-dependent benefits, yielding a 1.93% gain on Citeseer Global through dynamic temperature modulation. WDAD consistently contributes 0.8–1.6% improvements by enabling cross-class knowledge transfer.

When PCNS, ACDM, and WDAD are combined, the model achieves the best performance, effectively distilling both structural and semantic knowledge into a well-generalizable student model.

4.5 Sensitivity

To address **Q4**, we conduct analyses on hyperparameters of `TRUST`. Specifically, we analyze the model's performance under different values of λ and T, as defined in Equation 8.We evaluate all combinations of $\lambda \in \{0.25, 0.5, 0.75\}$ and $T \in \{20, 40, 80, 100\}$. Results shown in Figure 5 demonstrate that the choice of hyperparameters λ and T has a minimal impact on the performance of `TRUST`, proving the robustness of `TRUST`.



(a) Cora local (b) Cora global (c) CiteSeer local (d) CiteSeer global

Figure 5: **Analysis on hyper-parameter** in `TRUST`.Node classification results are evaluated on Cora and CiteSeer datasets under various hyperparameter combinations, testing both global and local accuracy. All experiments are conducted under the setting of $\alpha = 0.5$.

5 Conclusion

In this paper, we propose `TRUST` to address model heterogeneity in Federated Graph Learning. Based on knowledge distillation to bridge heterogenous client models, we integrates three key strategies to effectively handle complex topological information in graph-structured data. We first employ PCNS to progressively introduce complex samples based on learning difficulty. Then we propose ACDM for dynamic temperature adjustment. We further propose WDAD that captures cross-class structural relationships. Comprehensive experiments across five benchmark datasets demonstrate `TRUST`'s

superior capability in resolving model heterogeneity challenges while preserving graph topological properties. The framework establishes a state-of-the-art for heterogeneous FGL systems.

Acknowledgement

This work is supported by National Natural Science Foundation of China under Grant (62361166629, 62225113, 623B2080), the Major Project of Science and Technology Innovation of Hubei Province (2024BCA003, 2025BEA002), and the Innovative Research Group Project of Hubei Province under Grants 2024AFA017. The supercomputing system at the Supercomputing Center of Wuhan University supported the numerical calculations in this paper.

References

[1] A. Angelopoulos, S. Bates, J. Malik, and M. I. Jordan. Uncertainty sets for image classifiers using conformal prediction, 2022.

[2] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA, 2009. Association for Computing Machinery.

[3] J. Cai, Y. Zhang, J. Fan, and S.-K. Ng. Lg-fgad: An effective federated graph anomaly detection framework. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3760–3769, 2024.

[4] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transportation distances, 2013.

[5] X. Fu, Z. Chen, Y. He, S. Wang, B. Zhang, C. Chen, and J. Li. Virtual nodes can help: Tackling distribution shifts in federated graph learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

[6] X. Fu, B. Zhang, Y. Dong, C. Chen, and J. Li. Federated graph machine learning: A survey of concepts, techniques, and applications. *arXiv preprint arXiv:2207.11812*, 2022.

[7] C. L. Giles, K. D. Bollacker, and S. Lawrence. Citeseer: An automatic citation indexing system. In *Proceedings of the third ACM conference on Digital libraries*, pages 89–98, 1998.

[8] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks, 2014.

[9] J. Gou, B. Yu, S. J. Maybank, and D. Tao. Knowledge distillation: A survey. *IJCV*, pages 1789–1819, 2021.

[10] Q. Guo, X. Wang, Y. Wu, Z. Yu, D. Liang, X. Hu, and P. Luo. Online knowledge distillation via collaborative learning. In *CVPR*, pages 11020–11029, 2020.

[11] W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017.

[12] C. He, K. Balasubramanian, E. Ceyani, C. Yang, H. Xie, L. Sun, L. He, L. Yang, P. S. Yu, Y. Rong, P. Zhao, J. Huang, M. Annavaram, and S. Avestimehr. Fedgraphnn: A federated learning system and benchmark for graph neural networks, 2021.

[13] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[14] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network, 2015.

[15] C.-Y. Huang, K. Srinivas, X. Zhang, and X. Li. Overcoming data and model heterogeneities in decentralized federated learning via synthetic anchors, 2025.

[16] W. Huang, M. Ye, and B. Du. Learn from others and be yourself in heterogeneous federated learning. In *CVPR*, 2022.

[17] W. Huang, M. Ye, and B. Du. Learn from others and be yourself in heterogeneous federated learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10133–10143, 2022.

[18] W. Huang, M. Ye, Z. Shi, and B. Du. Generalizable heterogeneous federated cross-correlation and instance similarity learning. *TPAMI*, 2023.

[19] W. Huang, M. Ye, Z. Shi, H. Li, and B. Du. Rethinking federated learning with domain shift: A prototype view. In *CVPR*, 2023.

[20] W. Huang, M. Ye, Z. Shi, G. Wan, H. Li, B. Du, and Q. Yang. A federated learning for generalization, robustness, fairness: A survey and benchmark. *arXiv*, 2023.

[21] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[22] S. Kotz, N. Balakrishnan, and N. L. Johnson. *Continuous multivariate distributions, Volume 1: Models and applications*, volume 1. John Wiley & Sons, 2004.

[23] H. Li, X. Wang, and W. Zhu. Curriculum graph machine learning: A survey, 2024.

[24] M. Li, X. Zhang, Q. Wang, T. LIU, R. Wu, W. Wang, F. Zhuang, H. Xiong, and D. Yu. Resource-aware federated self-supervised learning with global class representations. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[25] Q. Li, B. He, and D. Song. Adversarial collaborative learning on non-iid features. *arXiv*, 2021.

[26] Q. Li, B. He, and D. Song. Model-contrastive federated learning. In *CVPR*, pages 10713–10722, 2021.

[27] 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.

[28] X. Li, Z. Wu, W. Zhang, H. Sun, R.-H. Li, and G. Wang. Adafgl: A new paradigm for federated node classification with topology heterogeneity. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 2379–2392. IEEE, 2024.

[29] X. Li, Z. Wu, W. Zhang, Y. Zhu, R.-H. Li, and G. Wang. Fedgta: Topology-aware averaging for federated graph learning. *Proceedings of the VLDB Endowment*, 17(1):41–50, 2023.

[30] Z. Liu, G. Wan, B. A. Prakash, M. S. Lau, and W. Jin. A review of graph neural networks in epidemic modeling. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6577–6587, 2024.

[31] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.

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

[33] S. Mehta and A. Aneja. Securing data privacy in machine learning: The fedavg of federated learning approach. In *2024 4th Asian Conference on Innovation in Technology (ASIANCON)*, pages 1–5. IEEE, 2024.

[34] M. Morafah, V. Kungurtsev, H. Chang, C. Chen, and B. Lin. Towards diverse device heterogeneous federated learning via task arithmetic knowledge integration, 2024.

[35] C. Pan, J. Xu, Y. Yu, Z. Yang, Q. Wu, C. Wang, L. Chen, and Y. Yang. Towards fair graph federated learning via incentive mechanisms, 2023.

[36] H. Papadopoulos, K. Proedrou, V. Vovk, and A. Gammerman. Inductive confidence machines for regression. In *Machine Learning: ECML 2002: 13th European Conference on Machine Learning Helsinki, Finland, August 19–23, 2002 Proceedings 13*, pages 345–356. Springer, 2002.

[37] V. Papyan, X. Y. Han, and D. L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, Sept. 2020.

[38] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

[39] Y. Tan, G. Long, L. Liu, T. Zhou, Q. Lu, J. Jiang, and C. Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *AAAI*, 2022.

[40] Z. Tan, G. Wan, W. Huang, and M. Ye. Fedssp: Federated graph learning with spectral knowledge and personalized preference. In *Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[41] G. Wan, W. Huang, and M. Ye. Federated graph learning under domain shift with generalizable prototypes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15429–15437, 2024.

[42] G. Wan, Z. Huang, W. Zhao, X. Luo, Y. Sun, and W. Wang. Rethink graphode generalization within coupled dynamical system. In *Forty-second International Conference on Machine Learning*, 2025.

[43] G. Wan, Z. Liu, X. Shan, M. S. Lau, B. A. Prakash, and W. Jin. Epidemiology-aware neural ode with continuous disease transmission graph. In *Forty-second International Conference on Machine Learning*, 2025.

[44] G. Wan, Z. Shi, W. Huang, G. Zhang, D. Tao, and M. Ye. Energy-based backdoor defense against federated graph learning. In *International Conference on Learning Representations*, 2025.

[45] G. Wan, Y. Tian, W. Huang, N. V. Chawla, and M. Ye. S3gcl: Spectral, swift, spatial graph contrastive learning. In *Forty-first International Conference on Machine Learning*, 2024.

[46] J. Wang, Q. Li, L. Lyu, and F. Ma. pfedclub: Controllable heterogeneous model aggregation for personalized federated learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[47] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *NeurIPS*, pages 7611–7623, 2020.

[48] J. Wang, X. Yang, S. Cui, L. Che, L. Lyu, D. Xu, and F. Ma. Towards personalized federated learning via heterogeneous model reassembly, 2023.

[49] J. Wang, C. Zhao, L. Lyu, Q. You, M. Huai, and F. Ma. Bridging model heterogeneity in federated learning via uncertainty-based asymmetrical reciprocity learning, 2024.

[50] L. Wang and K.-J. Yoon. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *IEEE TPAMI*, 2021.

[51] X. Wang, Y. Chen, and W. Zhu. A survey on curriculum learning, 2021.

[52] C. Wu, F. Wu, L. Lyu, T. Qi, Y. Huang, and X. Xie. A federated graph neural network framework for privacy-preserving personalization. *Nature Communications*, 13(1), June 2022.

[53] H. Xie, J. Ma, L. Xiong, and C. Yang. Federated graph classification over non-iid graphs. *NeurIPS*, 34:18839–18852, 2021.

[54] M. Ye, X. Fang, B. Du, P. C. Yuen, and D. Tao. Heterogeneous federated learning: State-of-the-art and research challenges, 2023.

[55] L. Yi, H. Yu, C. Ren, G. Wang, X. Liu, and X. Li. Federated model heterogeneous matryoshka representation learning, 2024.

[56] J. Zhang, Y. Liu, Y. Hua, and J. Cao. Fedtgp: Trainable global prototypes with adaptive-margin-enhanced contrastive learning for data and model heterogeneity in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

[57] K. Zhang, C. Yang, X. Li, L. Sun, and S. M. Yiu. Subgraph federated learning with missing neighbor generation, 2021.

[58] B. Zhao, Q. Cui, R. Song, Y. Qiu, and J. Liang. Decoupled knowledge distillation. In *CVPR*, 2022.

A Notations

We present a comprehensive review of the commonly used notations and their definitions in Tab. 2.

Table 2: Notation and Definitions

Notation	Definition
K	The number of clients.
$\mathcal{D}^k$	The dataset of k-th client.
w^k	The model of k-th client.
$\mathcal{G}$	Graph data.
$\mathcal{V}$	The node set of $\mathcal{G}$.
$\mathcal{E}$	The edge set of $\mathcal{G}$.
C	The label set of $\mathcal{G}$.
$\mathbf{X}$	The feature matrix of $\mathcal{G}$.
$\mathbf{A}$	The adjacency matrix of $\mathcal{G}$.
v_i	Node i in $\mathcal{V}$.
x_i	Feature vector of node i.
y_i	Label of node i.
$\mathcal{N}_i$	The neighborhood nodes of v_i.
h_i^l	The representation of v_i at the l-th layer of GNN.
$D(v_i)$	The node difficulty of v_i.
V_c	The node subset consisting of all nodes labeled as c.
p_c	The prototype of class c.
λ	The available proportion of data samples at epoch 0.
T	The epoch when full training data is first utilized.
η	The learning rate.
μ	The scaling factor of Adaptive Temperature Modulator.
p^T	The class probability distribution predicted by the teacher model.
p^S	The class probability distribution predicted by the student model.
τ	The distillation temperature.
CR	The pairwise similarity matrix between all classes.
c_{ij}	The probability mass transferred from teacher category i to student category j.
q_{ij}	The transmission cost from teacher category i to student category j.
κ	The hyper-parameter adjusting the sensitivity to cross-class similarity in Gaussian kernel.
L_{CE}	The cross-entropy function for classification tasks.
L_{WD}	The proposed Wasserstein-Driven Affinity Distillation loss.
L_{KL}	The Kullback-Leibler (KL) Divergence loss.

B Related Work.

Federated Graph Learning. With recent advances in Federated Learning (FL) for vision and language tasks [20, 18], Federated Graph Learning (FGL) further extends FL to graph-structured data [41, 42]. Existing FGL-related researches are mainly focused on processing graph-structured data. Based on how graphs are distributed across clients, these methods can fall into three categories: graph-level, subgraph-level and node-level [12]. Graph-level FGL methods consider settings where clients possess completely disjoint graphs while in subgraph-level FGL settings, each client holds a subgraph that is part of a larger global graph [57, 52]. In node-level FGL, each agent possesses the ego-networks of one or multiple nodes [35, 12]. However, these methods often assume homogeneous model architectures cross clients [17, 54, 40], which is an impractical constraint that degrades performance in real-world scenarios. To tackle this problem, we propose a model-heterogeneous method that transfers knowledge between clients in a model-agnostic manner through a novel knowledge distillation framework, overcoming limitations of existing FGL approaches.

Model-heterogeneous Federated Learning. Model-heterogeneous federated learning entails learning from others without disclosing information about local model architectures. Recent works on model-heterogeneous federated learning can be categorized into three approaches: data-level,

model-level and server-level [54]. Data-level methods, such as TAKFL [34] and DESA [15], distill knowledge via external public data distributions. Model-level methods, such as pFedHR [48] and pFedClub [46], share partial model structures or reassembled components to other clients. Server-level methods, such as FedMRL [55] and FedType [49], deploy proxy models as intermediaries to bridge heterogeneous models. While effective for Euclidean data, these methods fail to preserve graph topological features during knowledge transfer. Our work breaks this limitation by uniquely integrating curriculum learning and Wasserstein distance to capture complex structural information, improving graph representation in model-heterogeneous FGL.

Knowledge Distillation. Knowledge Distillation (KD) is a machine learning method designed for model compression and knowledge transfer [13, 14]. Traditional KD transfers knowledge from a large teacher model to a compact student model through softened outputs or intermediate representations [13]. Recently, researches have demonstrated its effectiveness in facilitating model collaboration in federated learning (FL), particularly in scenarios involving heterogeneous models or data distributions [54]. For instance, TAKFL [34] introduces KD to FL framework to distill knowledge from heterogeneous clients and then integrate the separately distilled knowledge with task arithmetic. Similarly, FedMKD [24] combines KD and attention mechanisms to work with model heterogeneity in FL. However, existing KD methods typically rely on KL-Divergence minimization, which only performs intra-category comparisons between teacher and student models. By contrast, our work proposes a noval Wasserstein knowledge distillation framework which introduces Wasserstein-distance to enable cross-class comparison, explicitly modeling graph-structured relationships during knowledge transfer and maintaining topological properties of the original graph data.

C Experimental Details.

C.1 Dataset Details.

To assess the effectiveness of `TRUST`, we conduct experiments on five real-world graph datasets: Cora, CiteSeer, PubMed, Amazon-Photo and CoAuthor-CS. Each dataset is split into training, validation, and test sets in a fixed 20%/40%/40% ratio. The key statistics of these datasets are summarized in Tab. 3. A detailed description is provided below:

- **Cora, CiteSeer, and PubMed.** These three citation network datasets are standard benchmarks in graph-based machine learning, especially for tasks like node classification and link prediction. In these datasets, nodes correspond to academic papers, while edges represent citation links. Each node is assigned a class label, and its feature vector is constructed from textual information such as words in the title or abstract. These datasets exhibit sparsity and high dimensionality, making them well-suited for evaluating the effectiveness and scalability of graph neural networks (GNNs).
- **CoAuthor-CS.** This dataset represents a co-authorship network in the field of computer science, where nodes correspond to research papers, and edges denote co-authorship relations. Each paper is associated with a topic category, and features are extracted from the paper's title and abstract. This dataset is commonly used to evaluate node classification and community detection algorithms.
- **Amazon-Photo.** This dataset is built from the Amazon product catalog, where nodes represent product images and edges indicate co-purchase relationships. Each photo is categorized into a specific class, and node features are derived from image metadata. Amazon-Photo serves as a benchmark for testing graph-based learning models in visual domains.

Table 3: **Statistics** of datasets used in experiments.

Dataset	#Nodes	#Edges	#Classes	#Features
Cora	2,708	5,278	7	1,433
Citeseer	3,327	4,552	6	3,703
Pubmed	19,717	44,324	3	500
Coauthor-CS	18,333	327,576	15	6,805
Amz-Photo	7,650	287,326	8	745

C.2 Counterpart Details.

This section provides a comprehensive overview of the baseline approaches employed in our study.

- **FedAvg** [ASTAT17]. A foundational algorithm in Federated Learning, FedAvg operates by allowing clients to independently train models on their local datasets and subsequently transmit their model updates to a central server. The server performs a weighted aggregation of these updates to refine the global model, which is then redistributed to the clients for further local training. By transmitting only model parameters instead of raw data, FedAvg reduces communication costs and enhances privacy. However, it struggles with performance degradation in scenarios where client data distributions are highly non-IID [27, 33].
- **FedNova** [NeurIPS20]. FedNova refines the FedAvg framework by introducing normalization to local updates before aggregation. Unlike standard averaging methods, FedNova ensures that each client's contribution to the global model is proportional to the amount of data it possesses. This approach addresses the issue of unequal client influence, leading to more balanced and efficient convergence. FedNova is particularly beneficial in federated environments where data distributions are skewed across clients.
- **FedProto** [AAAI22].FedProto introduces prototype-based federated learning to address both data and model heterogeneity across clients. Unlike gradient-based approaches, the framework exchanges class prototypes (mean feature representations) between server and clients, enabling knowledge transfer while accommodating different model architectures and non-IID data distributions. Through prototype aggregation and local regularization, FedProto achieves superior communication efficiency and convergence guarantees while preserving privacy [39]. The method demonstrates strong performance on image datasets while requiring significantly fewer communicated parameters than traditional FL approaches.
- **Moon** [CVPR21].MOON adopts a model-contrastive approach to address data heterogeneity in federated learning. The framework utilizes similarities between model representations to correct local training through model-level contrastive learning, providing an effective solution for collaborative training with deep learning models on image datasets while preserving data privacy.
- **FedType** [ICML24]. FedType[49] introduces a novel uncertainty-based asymmetrical reciprocity learning framework to address model heterogeneity in federated learning. The approach employs small identical proxy models as secure intermediaries for information exchange, eliminating the need for public data while ensuring privacy protection. Through bidirectional knowledge distillation with dynamic conformal prediction, FedType achieves superior performance across diverse model architectures and datasets, demonstrating significant improvements in communication efficiency and model security compared to existing methods.
- **AdaFGL** [ICDE24].AdaFGL introduces a novel paradigm for federated node classification with topology heterogeneity, addressing the critical challenge of structural divergence among clients in federated graph learning. The framework employs a decoupled two-step approach: first obtaining a federated knowledge extractor through collaborative training, then performing personalized propagation optimized by local topology. By incorporating adaptive mechanisms that automatically balance homophilous and heterophilous propagation based on quantified structural characteristics, AdaFGL achieves state-of-the-art performance across 12 benchmark datasets while minimizing communication overhead and privacy risks [28].
- **FedGTA** [VLDB24]. FedGTA is tailored for large-scale graph federated learning, tackling issues of slow convergence and suboptimal scalability. Unlike prior methods that focus on either optimization strategies or complex local models, FedGTA integrates topology-aware local smoothing with mixed neighbor feature aggregation to improve learning efficiency [29]. By leveraging graph structures in aggregation, it enhances scalability and performance in federated graph learning.

C.3 Implementation Details.

The experiments are conducted on NVIDIA GeForce RTX 3090 GPUs, paired with dual Intel(R) Xeon(R) Gold 6240 CPUs @ 2.60GHz (36 cores per socket, Turbo Boost up to 3.90GHz). The deep learning framework used is PyTorch (v2.5.1) with CUDA 12.1.

The experimental setup involves 10 clients. To simulate real-world model heterogeneity, each client maintains a private model whose architecture is randomly selected from GCN, GAT, or GraphSAGE. All private models are configured with three layers, a hidden dimension of 64, and a dropout rate of 0.3. To facilitate collaboration, each client is equipped with an additional small proxy model that serves as a communication bridge. This proxy model employs a standardized GCN architecture with 3 layers to ensure compatibility across clients. On the server side, we implement a global model

Table 4: **Comparison with the state-of-the-art methods** on three selected real-world datasets. The alpha is set to 0.3 and 1.0. The best and second-best results are highlighted with **bold** and underline, respectively.

Category	Methods	0.3 alpha						1.0 alpha					
		Cora		CiteSeer		PubMed		Cora		CiteSeer		PubMed	
		local	global	local	global	local	global	local	global	local	global	local	global
FL	FedAvg [ASTAT17]	86.39	64.18	84.43	64.14	90.82	84.64	74.21	72.39	70.57	69.99	86.82	84.16
	FedNOVA [NeurIPS20]	86.58$_{\uparrow 0.19}$	65.48$_{\uparrow 1.30}$	85.18$_{\uparrow 0.75}$	63.54$_{\downarrow 0.60}$	90.92$_{\uparrow 0.10}$	84.99$_{\uparrow 0.35}$	74.77$_{\uparrow 0.56}$	72.58$_{\uparrow 0.19}$	71.17$_{\uparrow 0.60}$	71.77$_{\uparrow 1.78}$	86.84$_{\uparrow 0.02}$	83.95$_{\downarrow 0.21}$
	FedProto [IJCAI23]	86.93$_{\uparrow 0.54}$	64.55$_{\uparrow 0.37}$	84.88$_{\uparrow 0.45}$	62.05$_{\downarrow 2.09}$	90.75$_{\downarrow 0.07}$	84.59$_{\downarrow 0.05}$	73.50$_{\downarrow 0.71}$	71.30$_{\downarrow 1.09}$	70.42$_{\downarrow 0.15}$	70.73$_{\uparrow 0.74}$	86.64$_{\downarrow 0.18}$	84.08$_{\downarrow 0.08}$
	MOON [CVPR21]	87.12$_{\uparrow 0.73}$	65.82$_{\uparrow 1.64}$	85.73$_{\uparrow 1.30}$	64.69$_{\uparrow 0.55}$	90.85$_{\uparrow 0.03}$	84.69$_{\uparrow 0.05}$	76.04$_{\uparrow 1.83}$	71.66$_{\downarrow 0.73}$	71.21$_{\uparrow 0.64}$	70.88$_{\uparrow 0.89}$	86.97$_{\uparrow 0.15}$	84.16$_{\uparrow 0.00}$
	FedType [ICML24]	86.21$_{\downarrow 0.18}$	63.09$_{\downarrow 1.09}$	85.72$_{\uparrow 1.29}$	62.95$_{\downarrow 1.19}$	90.44$_{\downarrow 0.38}$	84.59$_{\downarrow 0.05}$	73.66$_{\downarrow 0.55}$	75.50$_{\uparrow 3.11}$	69.97$_{\downarrow 0.60}$	68.20$_{\downarrow 1.79}$	85.91$_{\downarrow 0.91}$	84.74$_{\uparrow 0.58}$
FGL	AdaFGL [ICDE24]	84.60$_{\downarrow 1.79}$	65.45$_{\uparrow 1.27}$	84.35$_{\downarrow 0.08}$	65.22$_{\uparrow 1.08}$	87.52$_{\downarrow 3.30}$	84.89$_{\uparrow 0.25}$	74.96$_{\uparrow 0.75}$	73.31$_{\uparrow 0.92}$	**72.82**$_{\uparrow 2.25}$	70.73$_{\uparrow 0.74}$	**87.47**$_{\uparrow 0.65}$	84.03$_{\downarrow 0.13}$
	FedGTA [VLDB24]	84.72$_{\downarrow 1.67}$	65.45$_{\uparrow 1.27}$	83.66$_{\downarrow 0.77}$	62.35$_{\downarrow 1.79}$	87.30$_{\downarrow 3.52}$	84.67$_{\uparrow 0.03}$	72.94$_{\downarrow 1.27}$	73.31$_{\uparrow 0.92}$	69.26$_{\downarrow 1.31}$	70.58$_{\uparrow 0.59}$	85.39$_{\downarrow 1.43}$	84.16$_{\uparrow 0.00}$
Model-Heterogeneous FGL	TRUST	**88.67**$_{\uparrow 2.28}$	**67.18**$_{\uparrow 3.00}$	**86.76**$_{\uparrow 2.33}$	**66.52**$_{\uparrow 2.38}$	**91.97**$_{\uparrow 1.15}$	**86.67**$_{\uparrow 2.03}$	**77.50**$_{\uparrow 3.29}$	**77.70**$_{\uparrow 5.31}$	72.38$_{\uparrow 1.81}$	**72.03**$_{\uparrow 2.04}$	86.99$_{\uparrow 0.17}$	**85.04**$_{\uparrow 0.88}$

that also adopts a GCN backbone and uses a hidden dimension of 32 while sharing the remaining configurations with the client models.

To simulate data heterogeneity, client data is partitioned using a Dirichlet distribution [22], drawing $p_k \sim \text{Dir}(\alpha)$ to allocate a fraction p_k^c of class c to client k. Each client's subgraph is split into training, validation, and test sets with a ratio of 0.6/0.2/0.2, respectively. We use the Adam optimizer with a learning rate of 5×10^{-3} and a weight decay of 4×10^{-4} for training. The number of communication rounds is set to 200.

For PCNS, in the Difficulty Measurer, we set $\alpha = 0.5$. In the Curriculum Scheduler, the hyperparameters λ and T are selected via grid search over $\{0.25, 0.5, 0.75\}$ and $\{20, 40, 80, 100\}$, respectively. For ACDM, the model parameters θ_{model} are optimized using Adam (learning rate: 0.01, weight decay: 5×10^{-4}), while the temperature parameters θ_{temp} are optimized using SGD (momentum: 0.9, weight decay: 4×10^{-4}). The conformal mode parameters in backward distillation follow the configuration in FedType. In WDAD, we set $\eta = 0.05$, $\kappa = 1.0$, with loss weights $\alpha = 0.025$ (Wasserstein) and $\beta = 0.01$ (KL divergence).

Regarding baseline implementations, all selected FL baselines, except for FedAvg, support model heterogeneity and operate similarly to their applications in computer vision tasks. However, for FGL methods, to the best of our knowledge, our work is the first to explore model heterogeneity in FGL settings. Consequently, the chosen FGL baselines are originally designed for model homogeneity only. To include them (along with FedAvg) in heterogenous settings, we remove direct parameter sharing between architecturally different models and maintain all other original components and hyperparameters. For evaluation, we distill a global model from local private models and assess its accuracy.

D Additional Experimental Results.

D.1 Comparison with More FGL Baselines.

To further validate the efficacy of our approach, we compare TRUST against two additional FGL baselines, FedTAD and FedSSL, on the Cora, Citeseer, Pubmed, and CS datasets. The corresponding results are presented in Tab. 5.

Table 5: Local and gobal accuracy results of TRUST and two additional FGL baselines on the Cora, Citeseer, Pubmed and CS datasets under moderate heterogeneity($\alpha = 0.5$).

Methods	Cora		Citeseer		Pubmed		CS	
	local	global	local	global	local	global	local	global
FedTAD	80.27	65.70	83.24	66.37	87.88	83.30	90.72	82.93
FedSSL	79.53	64.79	82.32	**68.15**	88.07	83.58	91.97	80.73
TRUST	**83.90**	**75.32**	**85.8**4	67.43	**89.06**	**84.57**	**92.19**	**87.07**

As shown in Tab. 5, TRUST achieves the best performance on 7 out of the 8 evaluation metrics. The only exception is the global accuracy on the Citeseer dataset, where FedSSL attains the best result. These comprehensive results further confirm the effectiveness of TRUST against state-of-the-art FGL methods.

D.2 Comparison under Extreme Data Heterogeneity.

We provide additional local and global accuracy results under the α settings of 0.3 and 1.0 in Tab. 4, which is previously visualized in Figure 3. We also experiment at more extreme level of data heterogeneity, where $\alpha = 0.1$. Results are presented in Tab. 6.

Table 6: Comparison with state-of-the-art methods on three datasets under $\alpha = 0.1$. The best and second-best results are highlighted with bold and underline, respectively.

Methods	Cora		Citeseer		Pubmed	
	local	**global**	**local**	**global**	**local**	**global**
FedAvg	93.14	35.04	86.33	54.98	97.61	79.99
FedNOVA	93.51	36.50	86.47	56.17	97.66	79.92
FedProto	92.59	30.84	87.51	49.48	<u>97.69</u>	78.45
MOON	93.32	31.75	86.61	51.56	97.56	77.16
FedType	<u>94.06</u>	<u>40.88</u>	87.64	<u>60.03</u>	97.48	79.36
AdaFGL	85.44	36.13	88.22	56.91	92.88	79.69
FedGTA	85.43	35.40	**88.79**	56.17	92.86	<u>80.02</u>
TRUST	**94.60**	**42.52**	<u>88.44</u>	**64.04**	**97.76**	**82.65**

The results in Tab. 6 demonstrate that TRUST maintains superior performance even under extreme data heterogeneity, achieving the best performance on 5 out of the 6 evaluation metrics, with only the local accuracy on the Citeseer dataset showing slightly lower performance. These findings strongly validate the robustness and effectiveness of our approach.

E Backward Distillation in FedType.

Backward distillation leverages knowledge distillation to transfer knowledge from proxy models to private models. However, since proxy models are smaller than private models, conventional distillation approaches may lead to performance degradation. To address this, FedType introduces an Uncertainty-based Behavior Imitation Learning method that selectively transfers high-confidence knowledge rather than complete logits. For every node v_i, we construct a prediction set $\mathcal{S}_i$, which guarantees inclusion of the true label with high probability (e.g., 95% confidence). To compute $\mathcal{S}_i$, we need to train a conformal model denoted as cp using the validation dataset denoted as D' following Split Conformal Prediction[36, 1]. After training the conformal model, FedType proposes a dynamic conformal prediction with Regularized Adaptive Prediction Sets (RAPS) to calculate the prediction set:

$$\begin{aligned}
\mathcal{S}_i &= \{y \mid u \cdot p_i^t(y) + \rho_i^t(y) + g(\Delta^t, \lambda) \cdot (o_i^t(y) - \kappa_{\text{reg}})^+ \leq \tau\}, \\
\rho_i^t(y) &= \sum p_i^t(y') \mathbb{1}_{\{p_i^t(y') > p_i^t(y)\}}, \\
g(\Delta^t, \lambda) &= \begin{cases} \lambda \cdot \Delta^t - \Delta^t + \lambda, & \text{if } \Delta^t < 0, \\ \lambda, & \text{otherwise,} \end{cases} \\
o_i^t(y) &= |\{y' \mid \rho_i^t(y') > \rho_i^t(y)\}|,
\end{aligned} \tag{19}$$

where $p_i^t(y)$ denotes the probability of class y for data sample v_i predicted by conformal model, u is a randomization factor for prediction set construction. Furthermore, $\rho_i(y)$ denotes the total probability mass of labels more probable than y, $g(\Delta^t, \lambda)$ is a piecewise calibration function where $\Delta^t = A(p^t, D') - A(p^{t-1}, D')$ represents the accuracy difference between epoch t and $t-1$. Moreover, $o_i^t(y)$ denotes the label ranking of y based on predicted possiblity ρ_i^t, κ_{reg} is a regularization hyper-parameter, and $(\cdot)^+$ is the positive part operator.

After constructing the prediction set, FedType computes the knowledge transfer weight η_i, which plays a crucial role in determining the amount of information transferred from the proxy model to the

private model. This weight is inversely proportional to the size of the prediction set $\mathcal{S}_i$. When $\mathcal{S}_i$ contains fewer labels, which indicates higher confidence in the prediction, we assign a larger η_i to encourage the proxy model to transfer such confident knowledge to the private model. Conversely, larger prediction sets result in smaller transfer weights. Therefore, η_i can be defined as:

$$\eta_i = \begin{cases} |\mathcal{S}_i \cap \mathcal{L}_i|/|\mathcal{S}_i \cup \mathcal{L}_i|, & \text{if } |\mathcal{S}_i| \geq |\mathcal{L}_i|, \\ |\mathcal{S}_i \cap \mathcal{L}_i|/|\mathcal{S}_i|, & \text{if } |\mathcal{S}_i| < |\mathcal{L}_i|, \end{cases} \tag{20}$$

where $\mathcal{S}_i$ and $\mathcal{L}_i$ are similarly computed using Equation 19 but with conformal model trained by proxy model and private model respectively. This formulation ensures adaptive knowledge transfer based on the proxy model's confidence level.

During backward distillation, we specifically enhance the probability alignment for labels within the prediction set through the following loss function:

$$\begin{aligned} L_{\text{backward}} &= \sum_{i=1} \eta_i \sum_{y \in \mathcal{S}_i} \log \left(\frac{\exp(\mathbf{w}_i^t(v_i)[y])}{\Phi_i^t} \right), \\ \Phi_i^t &= \sum_{y \in \mathcal{S}_i} \exp(\mathbf{w}_i^t(v_i)[y]) + \sum_{y' \in \hat{\mathcal{S}}_i} \exp(\mathbf{w}_i^t(v_i)[y']), \end{aligned} \tag{21}$$

where $\hat{\mathcal{S}}_i = C - \mathcal{S}_i$ and represents labels with low confidence in the prediction.

F Complexity Analysis.

In this section, we present the complexity analysis of our proposed method TRUST. We begin with FedAvg as a baseline. Let T denote the communication rounds. On the client side, each client executes E epochs of local training per round with model parameter size d. When combined with GNN message passing, the computational complexity is $(L \cdot |\mathcal{E}| \cdot d)$ where L and $|\mathcal{E}|$ represent the number of GNN layers and edges respectively. On the server side, model aggregation across K clients yields $O(K \cdot d)$ complexity. Therefore, the total complexity is: $O(T \cdot (E \cdot L \cdot |\mathcal{E}| \cdot d + K \cdot d))$.

For FedType, knowledge distillation introduces an additional $O(N \cdot C)$ cost per epoch, where N and C are the number of nodes and classes respectively. The total complexity therefore becomes: $O(T \cdot (E \cdot (L \cdot |\mathcal{E}| \cdot d + N \cdot C) + K \cdot d))$.

For TRUST, it extends the knowledge distillation framework with three key components:

- **PCNS** first requires calculating node difficulties via neighborhood distribution entropy and prototype alignment. For neighborhood distribution entropy, it iterates through all edges ($O(|\mathcal{E}|)$). For prototype alignment, we pre-compute prototypes for all classes and calculate prototype similarity for every node ($O(N + C)$). Node sorting then adds $O(N \log N)$.
- **ACDM** involves only lightweight operations: temperature scaling $O(1)$ and cosine scheduling $O(1)$ per epoch, as indicated in Equation 13.
- **WDAD** leverages Sinkhorn algorithm to solve the loss function quickly. This reduces optimal transport complexity from $O(C!)$ to $O(k \cdot C^2)$, where k is the iteration count (in our experiments this is set to 10).

Notably, WDAD's $O(k \cdot C^2)$ complexity remains manageable in practice since C is bounded (e.g., C=7 for Cora and C=3 for Pubmed). Therefore, the total complexity is: $O(T \cdot (E \cdot (L \cdot |\mathcal{E}| \cdot d + N \cdot C + k \cdot C^2) + N \log N + K \cdot d))$.

As shown above, this represents only two additive terms compared to FedType: $O(N \log N)$ and $O(T \cdot E \cdot k \cdot C^2)$.

G Discussion on Limitations.

Although `TRUST` achieves significant success in addressing key challenges in model-heterogeneous Federated Graph Learning (FGL)—such as dynamic task difficulty adjustment, adaptive distillation signaling, and cross-class relational knowledge transfer—it still has some limitations. One notable challenge is the computational complexity involved in dynamically adjusting task difficulty and distillation strength, which could become a bottleneck in large-scale 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: The main claims made in the abstract and introduction accurately reflect the contributions and scope of this paper.

   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]

   We discuss the limitations in Appendix G.

   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: [NA]

Justification: The paper does not include theoretical results.

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 fully disclose all the information needed to reproduce the main experimental results in this paper and our code. We are convinced that the obtained results can be reproduced.

   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: Code is accessible in this paper.

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: Details are included in Appendix C.3.

   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: Statistical significance of the experiments is considered and included in Sec. 4.

   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: We provide sufficient information on the computer resources.

   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: The research conducted in the paper strictly adheres to the NeurIPS Code of Ethics, ensuring that all aspects of the work are in compliance with the guidelines provided.

   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: [NA]

    Justification: The research presented in this paper is foundational. It is not directly tied to any specific applications or deployments.

    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 paper poses no 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: [NA]

    Justification: This paper does not use existing assets.

    - 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: [NA]

    Justification: This paper does not release new assets.

    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 paper does not involve crowdsourcing nor 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: The paper does not involve crowdsourcing nor 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.