TASK-GUIDED CROSS-SUBGRAPH TOPOLOGY LEARNING FOR FEDERATED GRAPH CLUSTERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated graph clustering (FGC) aims to group nodes across distributed subgraphs, where missing cross-subgraph edges and privacy constraints hinder global topology learning and reliable cluster assignment. Existing methods typically recover missing cross-subgraph edges using graph-kernel similarity and then perform clustering on the reconstructed graph. However, this recover-then-cluster paradigm is largely decoupled from the clustering objective, as it lacks a task-aware criterion to evaluate the quality of recovered cross-subgraph edges, leading to inferior clustering performance. To address this issue, we propose **T**ask-guided **C**ross-subgraph **T**opology **L**earning (**TCTL**), which learns cross-subgraph topology explicitly aligned with the clustering objective by exploiting clustering knowledge from both clients and the server. Specifically, we design a gradient-based edge importance estimation scheme to quantify the importance of candidate cross-subgraph edges by leveraging clustering-oriented gradients from the current global clustering state. We further develop a knowledge-driven edge-weight updating scheme, which adopts server-derived gradient feedback as supervision to iteratively refine edge weights conditioned on client-derived cluster semantics. This iterative process allows the learned topology and clustering state to mutually enhance each other. Extensive experiments on five benchmark datasets demonstrate the effectiveness and superiority of TCTL against its competitors.

1 INTRODUCTION

Federated graph clustering (FGC) aims to discover node communities from graph data distributed across multiple clients without exposing their raw local graphs. In many practical applications, including disease prediction (Peng et al., 2022), social network analysis (Lei et al., 2023), and financial fraud detection (Tang & Liang, 2024), graph data are often distributed as subgraphs across different clients, while edges between nodes held by different clients are unavailable due to data isolation. This fragments the global graph, prevents node representations from capturing cross-client structure, and can degrade clustering performance. Consequently, recovering missing cross-subgraph edges that benefit clustering while preserving data privacy is a central challenge in FGC.

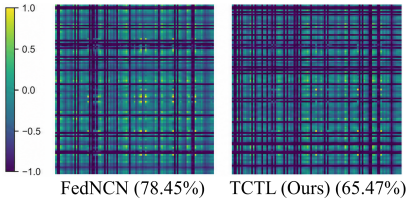
Recent FGC studies have investigated different forms of graph data incompleteness through various techniques (Liu et al., 2025b; 2026b; Han et al., 2026). For example, IFedGC (Liu et al., 2026b) addresses missing node attributes by constructing a global causal structure to guide attribute completion, but does not explicitly consider missing cross-subgraph edges. Moreover, FedNCN (Liu et al., 2025b) leverages clustering prior knowledge to select representative nodes and then recovers potential cross-subgraph edges based on graph-kernel similarity, with an improved N-Cut criterion further removing redundant edges before clustering on the reconstructed graph. Despite its success, FedNCN follows a "recover-then-cluster" learning paradigm, where topology learning is decoupled from the subsequent clustering objective. This separation leads to two key limitations. **First**, from the client perspective, local clustering information is primarily used to select representative nodes, while clustering-friendly signals retained in node representations are not fully exploited to guide cross-subgraph topology learning. **Second**, from the server perspective, candidate edges are typically evaluated based on structural similarity rather than their actual contributions to the clustering objective. Since graph-kernel similarity is agnostic to the evolving global clustering state, structurally plausible edges may not improve and even degrade clustering performance. As shown in Fig. 1, although FedNCN achieves a higher similarity between the recovered topology and the orig-

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

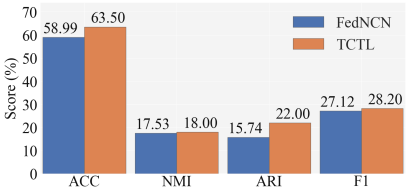
inal topology (78.45%) than TCTL (65.47%), TCTL obtains better clustering performance, achieving higher ACC, NMI, ARI, and F1 scores. This observation indicates that higher similarity to the original topology does not translate into improved clustering performance, revealing an inherent misalignment between structural recovery and the clustering objective.

To overcome these limitations, we move beyond recovering missing cross-subgraph edges and instead learn a cross-subgraph topology explicitly optimized for clustering. This objective is challenging in federated settings, where local cluster semantics are available only at the clients, whereas task-level feedback from global clustering is accessible only at the server. Effectively integrating these complementary signals while preserving privacy raises two key questions: 1) how can clustering-relevant knowledge be extracted from both the client and server sides? and 2) how can these signals be jointly leveraged to learn a clustering-oriented topology? To answer the first question, we exploit representations of client-selected representative nodes to encode local cluster semantics and exploit gradients of the server-side clustering objective to quantify the task-specific utility of candidate edges. To address the second, we adopt the server-derived clustering feedback as task-level supervision to learn shared edge-refinement patterns conditioned on client-derived cluster semantics.

Based on the above insights, we propose **T**ask-guided **C**ross-subgraph **T**opology **L**earning termed **TCTL**, a unified federated graph clustering framework that learns a cross-subgraph topology explicitly aligned with the clustering objective. Unlike existing methods that decouple topology recovery from clustering, TCTL integrates complementary clustering knowledge from clients and the server into an end-to-end task-guided topology learning process. Specifically, TCTL comprises two key stages: candidate-edge evaluation and candidate-edge refinement. In the evaluation stage, TCTL computes the gradient of the clustering objective with respect to each candidate edge weight, so as to directly quantify the contribution of each potential cross-subgraph edge to the clustering task. In the refinement stage, TCTL combines client-side cluster semantics encoded in representative-node representations with server-side edge-importance signals derived from the clustering objective. These complementary signals are integrated through a shared edge-refinement mechanism, enabling TCTL to progressively learn cross-subgraph connectivity patterns that improve clustering across communication rounds. The main contributions of this work are summarized as follows:



(a) Structural similarity between the recovered and original topologies.



(b) Clustering performance comparison.

Figure 1: Motivation illustration.

- We propose TCTL, a novel FGC framework that learns task-guided cross-subgraph topology by leveraging complementary clustering knowledge from clients and the server.
- We develop gradient-based edge importance estimation and knowledge-driven edge-weight updating schemes. The former quantifies the task-specific utility of candidate edges using gradients of the clustering objective, while the latter integrates this server-side feedback with client-derived cluster semantics to iteratively refine cross-subgraph edge weights.
- Extensive experiments on five benchmark graph datasets demonstrate that TCTL consistently outperforms existing federated graph clustering methods.

2 Related Work

2.1 Federated Graph Clustering

Federated graph clustering (FGC) enables multiple clients to collaboratively cluster unlabeled graph data without sharing their raw graphs (Liu et al., 2025a; Wang et al., 2026a; Han et al., 2026; Liu et al., 2026a;b; 2025b). For example, IFedGC addresses incomplete node attributes by constructing a global causal structure that guides attribute completion using clustering-relevant information (Liu et al., 2026b). FedNCN studies missing cross-subgraph edges in FGC (Liu et al., 2025b). It leverages clustering prior knowledge to select representative nodes, based on which cross-subgraph edges

are recovered using graph-kernel similarity and further pruned by an N-Cut criterion. Despite its effectiveness, FedNCN relies primarily on graph-kernel similarity for edge recovery and does not directly assess how recovered edges affect the clustering objective. In contrast, TCTL evaluates the effect of cross-subgraph edges on clustering via a gradient-based criterion.

2.2 Edge Refinement in Deep Graph Clustering

Edge refinement in deep graph clustering (DGC) seeks to mitigate the effects of noisy, redundant, or unreliable edges on representation learning and clustering. For instance, SAMVGC (Guan et al., 2025) refines graph structures based on geometric information in the embedding space. Similarly, FedNCN (Liu et al., 2025b) recovers cross-subgraph edges based on graph-kernel similarity after selecting representative nodes with clustering prior knowledge. More recently, APGC (Wang et al., 2026b) refines graph structures by strengthening reliable edges and suppressing conflicting ones based on attribute similarity. Besides similarity-based refinement strategies, some studies further incorporate information related to clustering to guide edge refinement. Although these methods refine graph structure through various strategies, they do not leverage feedback from the clustering objective to guide edge updates. Moreover, most of these methods are designed for centralized settings and do not consider privacy constraints. To address these limitations, our method exploits gradient-based clustering feedback to refine cross-subgraph edge weights without sharing raw graph data, yielding a topology that progressively improves both clustering and subsequent edge refinement.

3 Methodology

3.1 Notations

Let M and O denote the number of clients and global cluster centers, respectively. The m-th client contains O_m local cluster centers and a local subgraph by $\mathcal{G}_m = \{\mathcal{V}_m, \mathcal{E}_m\}$ comprising N_m nodes. Its node attribute matrix and original adjacency matrix are denoted by $\mathbf{X}_m \in \mathbb{R}^{N_m \times d}$ and $\mathbf{A}_m \in \mathbb{R}^{N_m \times N_m}$, respectively, where d is the attribute dimensionality. For graph neural network (GNN) encoding, we construct the normalized adjacency matrix $\widetilde{\mathbf{A}}_m \in \mathbb{R}^{N_m \times N_m}$ by first adding self-loops to $\mathbf{A}_m$ and then applying symmetric degree normalization. Here, $\mathbf{I}_m \in \mathbb{R}^{N_m \times N_m}$ denotes the identity matrix used to introduce self-loops, and $\bar{\mathbf{D}}_m \in \mathbb{R}^{N_m \times N_m}$ denotes the degree matrix of the resulting adjacency matrix. The remaining notations are summarized in Appendix A.

3.2 Cross-Subgraph Candidate Topology Construction

We first construct an initial candidate topology at the server to provide structural priors for subsequent task-guided cross-subgraph topology learning. Specifically, following FedNCN (Liu et al., 2025b), each client encodes and clusters its local graph to obtain a set of local cluster centers. For each center, the client selects the kN_m nearest nodes as representative nodes, where k denotes the selection ratio. With a privacy-preserving projection mechanism, the server reconstructs the attributes of these representative nodes, resulting in a total of $N = \sum_{m=1}^{M} kN_mO_m$ representative nodes. The representative nodes associated with the same local cluster center are then grouped into a cluster-level subgraph. Within each cluster-level subgraph, edges are established according to the pairwise similarities between node representations, yielding a fixed intra-subgraph topology $\mathbf{A}^I$. We subsequently construct a candidate cross-subgraph topology by measuring the graph-level affinity between different cluster-level subgraphs. Given two different cluster-level subgraphs $\mathcal{G}_p$ and $\mathcal{G}_q$, where $p \neq q$, the candidate cross-subgraph edge weight between $v_i \in \mathcal{G}_p$ and $v_j \in \mathcal{G}_q$ is defined as:

$$[\mathbf{A}^C]_{ij} = \mathrm{Norm}\left(\mathrm{GK}\left(\mathcal{G}_p, \mathcal{G}_q\right)\right) \in [\epsilon, 1-\epsilon], \tag{1}$$

where $\mathrm{GK}(\cdot,\cdot)$ denotes a graph-kernel similarity function, $\mathrm{Norm}(\cdot)$ denotes the normalization operation, and $\epsilon > 0$ is a small constant introduced to ensure numerical stability. Collectively, these candidate cross-subgraph edges constitute $\mathbf{A}^C$. By combining $\mathbf{A}^C$ with the fixed intra-subgraph topology, we obtain the initial candidate topology $\mathbf{A}^\phi = \mathbf{A}^I + \mathbf{A}^C$. During subsequent task-guided topology learning, only $\mathbf{A}^C$ is refined, whereas the intra-subgraph topology $\mathbf{A}^I$ remains fixed.

The representative nodes uploaded by clients not only enable the construction of cluster-level subgraphs but also preserve the cluster semantics captured through local clustering. Based on these rep-

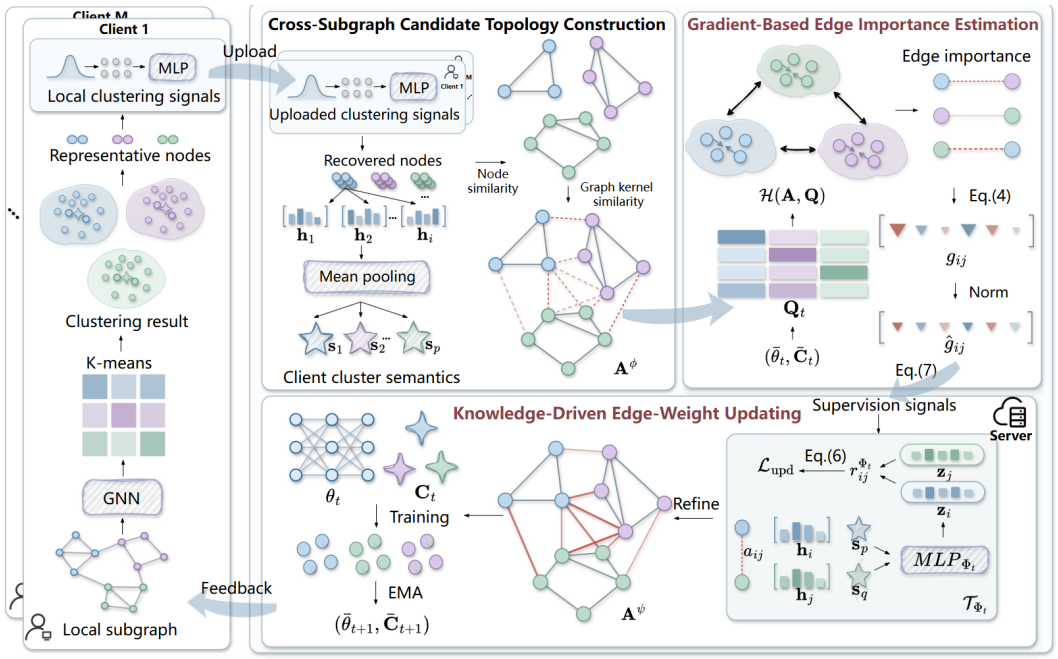


Figure 2: Overall framework of TCTL, comprising three stages: 1) cross-subgraph candidate topology construction, which leverages client-side clustering knowledge to initialize potential cross-subgraph connections; 2) gradient-based edge importance estimation, which quantifies candidate edges according to their contributions to the clustering objective; and 3) knowledge-driven edge-weight updating, which combines client-side cluster semantics with server-side gradient feedback as supervision signals to refine task-relevant cross-subgraph edges. The refined topology is then used to support more reliable clustering, which provides feedback for subsequent topology learning.

resentative nodes, we derive cluster-level semantic representations by aggregating their node representations, which provide client-side clustering information for subsequent cross-subgraph topology learning. For a cluster-level subgraph $\mathcal{G}_p$, we denote the representations of its constituent nodes by $\{\mathbf{h}_i \mid v_i \in \mathcal{V}_p\}$. The corresponding cluster-level semantic representation is obtained through mean pooling: $\mathbf{s}_p = \frac{1}{|\mathcal{V}_p|} \sum_{v_i \in \mathcal{V}_p} \mathbf{h}_i$, where $\mathcal{V}_p$ denotes the node set of $\mathcal{G}_p$, and $\mathbf{s}_p$ represents the cluster-level semantic representation that captures the clustering knowledge contributed by the client.

3.3 Task-Guided Cross-Subgraph Topology Learning

Given the candidate topology $\mathbf{A}^{\phi}$, we further optimize its cross-subgraph connections using task-specific supervision. As illustrated in Fig. 2, TCTL integrates complementary topology-learning signals from clients and the server into a unified edge-weight refinement process. Specifically, client-derived cluster-level semantic representations capture local clustering knowledge, while server-derived gradient feedback provides task-specific guidance for candidate edge refinement. These complementary signals are integrated into a unified edge-weight update function, enabling iterative cross-subgraph topology optimization across communication rounds. The overall training procedure of TCTL is summarized in Algorithm 1.

Gradient-Based Edge Importance Estimation. In addition to the cluster-level semantic information provided by the client side, the server evaluates the importance of candidate cross-subgraph edges using gradient-based feedback. Specifically, at t-th communication round, we denote the exponentially moving averaged (EMA) global clustering state by $(\bar{\theta}_t, \bar{\mathbf{C}}_t)$, where $\bar{\theta}_t$ and $\bar{\mathbf{C}}_t$ are the smoothed global GNN parameters and cluster centers, respectively. The corresponding EMA update rules are provided in Eq. (11). Given the candidate global graph, the current smoothed global model produces the soft cluster-assignment matrix $\mathbf{Q}_t \in \mathbb{R}^{N \times O}$:

$$\mathbf{Q}_t = Q\left(\mathbf{A}^{\phi}; \bar{\theta}_t, \bar{\mathbf{C}}_t\right), \tag{2}$$

Algorithm 1 Task-guided Cross-subgraph Topology Learning (TCTL)

Require: Local subgraphs $\{G_m = (V_m, E_m)\}_{m=1}^{M}$, node attributes $\{X_m\}_{m=1}^{M}$, number of clusters O, communication rounds T
Ensure: Refined topology $\mathbf{A}^{\psi}$ and clustering model Θ
1: Construct candidate topology $\mathbf{A}^{\phi} = \mathbf{A}^{I} + \mathbf{A}^{C}$ using representative nodes.
2: **for** $t = 1$ to T **do**
3: Estimate edge importance $\hat{g}_{ij}$ via clustering-task gradients.
4: Integrate cluster semantics from clients and gradient feedback from the server.
5: Update cross-subgraph topology $\mathbf{A}^{C}$ and obtain refined topology $\mathbf{A}^{\psi} = \mathbf{A}^{I} + \mathbf{A}^{R}$.
6: Optimize clustering objective and update Θ.
7: **end for**
8: **return** $\mathbf{A}^{\psi}, \Theta$

where $Q(\cdot)$ denotes the soft clustering assignment function induced by the current global clustering model. To derive task-specific feedback for candidate-edge refinement, we introduce a task-feedback function that quantifies the compatibility between the current topology and its clustering assignments. Inspired by structural entropy, this function favors topology structures that promote intra-cluster compactness and inter-cluster separation(Zhang et al., 2025). Given an arbitrary adjacency matrix $\mathbf{A}$ and soft clustering assignment matrix $\mathbf{Q}$, the function is defined as:

$$\mathcal{H}(\mathbf{A}, \mathbf{Q}) = -\sum_{o=1}^{O} \frac{\mathbf{d}^{\top}\mathbf{q}_o - \mathbf{q}_o^{\top}\mathbf{A}\mathbf{q}_o}{V} \log \frac{\mathbf{d}^{\top}\mathbf{q}_o}{V}, \tag{3}$$

where $d_i = \sum_j [\mathbf{A}]_{ij}$ is the degree of node v_i, $V = \sum_i d_i$ is the graph volume, and $\mathbf{q}_o$ denotes the assignment vector associated with the o-th cluster. We then quantify the contribution of each cross-subgraph edge using the negative gradient of the task-feedback function with respect to $\mathbf{A}^{C}$:

$$g_{ij} = -\left[\nabla_{\mathbf{A}^C} \mathcal{H}\left(\mathbf{A}^{\phi}, \mathbf{Q}_t\right)\right]_{ij}. \tag{4}$$

The negative sign ensures that a positive g_{ij} indicates that increasing the corresponding edge weight is expected to reduce the task-feedback objective. To prevent excessively large updates caused by variations in gradient magnitude, we normalize the gradient feedback as: $\hat{g}_{ij} = \frac{g_{ij}}{\max_{(i,j)} |g_{ij}|}$, where g_{ij} is the raw gradient-based contribution of candidate cross-subgraph edge (i, j), and $\hat{g}_{ij}$ is its normalized counterpart used for subsequent edge-weight refinement.

Knowledge-Driven Edge-Weight Updating. Building upon the cluster-level semantic representations derived from the clients and the gradient feedback computed by the server, we learn a knowledge-driven edge-weight updating function that integrates these two complementary sources of clustering knowledge for topology optimization. Specifically, the cluster-level representations encode the local cluster semantics associated with individual nodes, whereas the gradient feedback provides task-specific guidance for refining candidate edge weights. Consider two nodes $v_i \in \mathcal{G}_p$ and $v_j \in \mathcal{G}_q$ belong to different cluster-level subgraphs, with their candidate edge weight denoted by $a_{ij} = [\mathbf{A}^{C}]_{ij}$. We jointly project each node representation and its corresponding cluster-level semantic representation into a latent relation space using a shared multilayer perceptron (MLP):

$$\mathbf{z}_i = \mathrm{MLP}_{\Phi_t}([\mathbf{h}_i \| \mathbf{s}_p]), \qquad \mathbf{z}_j = \mathrm{MLP}_{\Phi_t}([\mathbf{h}_j \| \mathbf{s}_q]), \tag{5}$$

where $\|$ denotes vector concatenation and Φ_t represents the shared parameters at communication round t. By continually optimizing Φ_t across communication rounds, the model progressively accumulates task-guided knowledge of node relationships. The relation score $r_{ij}^{\Phi_t}$ between v_i and v_j is computed using cosine similarity: $r_{ij}^{\Phi_t} = \frac{\mathbf{z}_i^{\top}\mathbf{z}_j}{\|\mathbf{z}_i\|_2 \|\mathbf{z}_j\|_2}$. Based on the initial edge weight and the learned relation score, the knowledge-driven edge-weight updating function is defined as:

$$\mathcal{T}_{\Phi_t}(\mathbf{h}_i, \mathbf{h}_j, \mathbf{s}_p, \mathbf{s}_q, a_{ij}) = \sigma\left(\mathrm{logit}(a_{ij}) + \gamma r_{ij}^{\Phi_t}\right), \tag{6}$$

where $\sigma(\cdot)$ denotes the sigmoid function and γ is a learnable scaling factor. To incorporate the gradient-based feedback provided by the server, we transform the normalized edge contribution into a task-guided target edge weight for the current communication round:

$$a_{ij}^{\mathrm{tar}} = \sigma\left(\mathrm{logit}(a_{ij}) + \hat{g}_{ij}\right). \tag{7}$$

The knowledge-driven edge-weight update function is optimized by minimizing the discrepancy between its predicted edge-weight adjustment and the corresponding task-guided target adjustment:

$$\mathcal{L}_{\text{upd}} = \sum_{(i,j)} \left[\mathcal{T}_{\Phi_t} - a_{ij} - \text{sg}\left(a_{ij}^{\text{tar}} - a_{ij}\right)\right]^2, \tag{8}$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operator. Minimizing $\mathcal{L}_{\text{upd}}$ updates the shared parameters from Φ_t to Φ_{t+1}, enabling the update function to progressively accumulate task-guided knowledge for edge refinement across communication rounds. The optimized updating function is subsequently employed to refine the candidate cross-subgraph edge weights: $nijiushuoa_{ij}^{R} = \mathcal{T}_{\Phi_{t+1}}$. The refined cross-subgraph edge weights collectively constitute the adjacency matrix $\mathbf{A}^R$. Combining this matrix with the fixed intra-subgraph topology yields the final learned global topology:

$$\mathbf{A}^{\psi} = \mathbf{A}^{I} + \mathbf{A}^{R}. \tag{9}$$

In summary, TCTL offers several advantages for federated graph clustering. By optimizing cross-subgraph connectivity directly with respect to the clustering objective, it avoids relying solely on structural similarity, whose alignment with clustering performance is not guaranteed. Retaining the reliable intra-subgraph topology while selectively refining cross-subgraph edges constrains the learning space and reduces the influence of noisy candidate connections. Moreover, the integration of client-derived cluster semantics and server-derived gradient feedback enables complementary local and global clustering knowledge to guide edge refinement. EMA-smoothed clustering states provide stable task supervision, whereas the shared update function accumulates transferable refinement patterns across communication rounds. Consequently, TCTL supports a privacy-preserving iterative process in which topology learning and global clustering mutually improve each other.

3.4 Clustering Learning and Model Optimization

Given the refined global graph $\mathcal{G}^{\psi}$, we perform global clustering using the current model state $(\theta_t, \mathbf{C}_t)$. Let $\widehat{\mathbf{X}} \in \mathbb{R}^{N\times d}$ denote the node attributes reconstructed by the GNN decoder. Based on the latent node representations, we compute the soft cluster-assignment matrix $\mathbf{Q} \in \mathbb{R}^{N\times O}$ using Student's t-distribution and derive the target distribution $\mathbf{P} \in \mathbb{R}^{N\times O}$ by sharpening the soft assignments. The global clustering objective is defined as:

$$\mathcal{L} = \frac{1}{N}\left\|\widehat{\mathbf{X}} - \mathbf{X}\right\|_F^2 + \sum_{i=1}^{N}\sum_{o=1}^{O} P_{io}\log\frac{P_{io}}{Q_{io}}, \tag{10}$$

where the first term is the attribute reconstruction loss and the second term minimizes the Kullback-Leibler divergence between the target and soft assignment distributions. Minimizing $\mathcal{L}_{\text{global}}$ updates the global clustering state from $(\theta_t, \mathbf{C}_t)$ to $(\theta_{t+1}, \mathbf{C}_{t+1})$. To provide stable clustering feedback for topology refinement, we maintain an exponential moving average (EMA) of the global clustering state. Let $\mathbf{\Theta}_t = (\theta_t, \mathbf{C}_t)$ denote the global clustering state at round t. Its EMA-smoothed counterpart is updated as (Tarvainen & Valpola, 2017):

$$\bar{\mathbf{\Theta}}_{t+1} = \beta\bar{\mathbf{\Theta}}_t + (1-\beta)\mathbf{\Theta}_{t+1}, \tag{11}$$

where $\beta \in [0, 1)$ is the EMA smoothing coefficient. The resulting smoothed state $\bar{\mathbf{\Theta}}_{t+1} = (\bar{\theta}_{t+1}, \bar{\mathbf{C}}_{t+1})$ is used in the subsequent communication round to provide stable task-specific feedback for topology refinement. The complete training procedure is provided in Appendix B.

4 Experiments

4.1 Experiment Setup

Benchmark Datasets. Following FedNCN (Liu et al., 2025b), we partition each dataset into 5, 10, and 20 client subgraphs to simulate the federated setting, where each client only accesses its local subgraph. Experiments are conducted on five benchmark datasets: CiteSeer (Liu et al., 2023), PubMed (Jiang et al., 2024), Amazon-Computer (abbreviated as Am-Co), Amazon-Photo (abbreviated as Am-Ph) (Lin et al., 2021), and Questions (Platonov et al., 2023). Dataset details are provided in Appendix C.1.

Table 1: Clustering performance comparison of federated node-level clustering methods.

Methods	CiteSeer (5 Clients)				CiteSeer (10 Clients)				CiteSeer (20 Clients)			
	ACC	**NMI**	**ARI**	**F1**	**ACC**	**NMI**	**ARI**	**F1**	**ACC**	**NMI**	**ARI**	**F1**
FedSage+*	16.1±1.7	2.7±0.4	0.2±0.6	11.4±0.8	17.2±2.6	5.4±0.3	−0.1±0.6	11.3±0.6	15.6±2.0	9.5±0.2	0.3±0.5	10.1±0.8
FedPUB*	16.8±3.4	0.0±0.0	0.0±0.0	4.1±0.6	9.1±5.2	0.0±0.0	0.0±0.0	3.0±1.6	11.2±6.5	0.0±0.0	0.0±0.0	3.8±2.0
FedTAD*	17.6±3.3	1.2±1.4	0.6±1.0	7.1±2.5	19.1±1.6	4.0±2.6	1.6±1.3	9.9±2.7	17.8±1.5	3.4±3.6	1.1±1.4	6.8±2.4
FedGTA*	24.1±0.8	5.7±1.1	3.5±0.7	19.9±1.1	26.3±0.6	8.4±1.6	4.7±1.9	19.7±0.7	28.1±1.6	12.3±2.1	4.3±2.1	17.1±2.2
FedIIH*	14.2±2.9	0.0±0.0	0.0±0.0	3.7±0.6	17.6±6.1	0.7±0.8	0.4±0.5	4.8±1.1	18.3±7.5	0.0±0.1	−0.1±0.1	5.3±1.9
FedNCN	55.0±1.8	13.3±2.5	19.7±3.7	24.8±1.4	59.0±3.0	17.5±1.3	15.7±4.0	27.1±2.9	57.5±1.6	19.7±1.9	18.0±1.7	32.8±1.1
TCTL	**56.0±1.2**	**20.9±2.8**	**28.2±4.1**	**30.7±2.4**	**63.5±1.6**	**18.0±2.2**	**22.0±2.0**	**28.2±2.0**	**58.8±1.9**	**26.9±1.0**	**23.8±1.8**	**36.4±1.3**
Methods	**PubMed (5 Clients)**				**PubMed (10 Clients)**				**PubMed (20 Clients)**			
FedSage+*	49.5±6.3	8.7±3.8	4.8±4.7	33.8±4.5	40.7±7.8	2.2±4.5	0.0±0.0	26.2±6.0	43.8±8.3	2.0±1.7	−0.0±3.6	30.8±3.2
FedPUB*	32.0±9.4	0.0±0.0	0.0±0.0	14.1±3.2	32.0±9.0	0.0±0.0	0.0±0.0	14.0±3.2	28.3±9.2	0.0±0.0	0.0±0.0	12.4±3.3
FedTAD*	34.1±4.7	1.2±1.0	0.8±0.4	26.9±3.5	37.3±4.6	0.4±0.5	0.5±0.9	20.2±3.9	36.5±5.2	0.4±0.7	0.7±1.2	18.2±1.6
FedGTA*	46.2±2.5	6.2±1.9	6.3±1.9	39.6±2.8	46.9±1.7	2.1±0.8	1.8±1.2	33.3±1.9	48.8±1.7	2.5±0.7	2.6±0.8	32.2±1.0
FedIIH*	32.3±9.7	0.0±0.1	0.1±0.3	14.6±3.5	33.5±7.8	0.0±0.0	0.0±0.0	14.6±2.8	28.3±9.2	0.0±0.0	0.0±0.0	12.4±3.3
FedNCN	63.4±1.6	10.5±3.5	13.8±3.6	44.0±2.3	65.0±1.5	10.6±1.8	13.6±4.6	44.4±3.5	66.3±1.7	7.5±0.9	11.0±1.2	41.3±1.3
TCTL	**64.2±1.4**	**11.1±3.2**	**15.4±3.5**	**45.3±2.3**	**69.3±1.3**	**14.5±1.5**	**19.4±2.8**	**47.5±1.8**	**67.4±1.6**	**13.1±1.2**	**17.1±0.9**	**45.9±1.7**
Methods	**Am-Co (5 Clients)**				**Am-Co (10 Clients)**				**Am-Co (20 Clients)**			
FedSage+*	21.4±0.9	5.1±2.8	4.0±3.1	14.8±2.1	23.0±0.8	8.2±0.9	6.3±1.0	17.4±0.2	20.6±1.3	7.1±1.4	4.7±2.0	21.2±0.9
FedPUB*	11.7±13.0	0.0±0.0	0.0±0.0	2.3±2.4	22.4±12.3	0.0±0.0	0.0±0.0	5.0±3.2	14.2±12.1	0.0±0.0	0.0±0.0	4.0±3.5
FedTAD*	9.8±4.4	3.2±1.7	3.3±1.6	3.3±1.5	12.8±12.4	3.7±2.0	4.7±2.6	3.4±1.9	17.8±11.3	5.1±1.3	4.0±0.4	3.8±1.4
FedGTA*	21.4±1.8	12.1±4.0	8.4±3.3	9.0±1.0	21.0±1.5	6.5±1.2	3.7±1.0	6.9±0.5	22.1±1.2	7.2±1.0	3.5±0.8	6.3±0.3
FedIIH*	13.5±7.1	0.0±0.0	0.0±0.0	2.6±1.5	7.8±4.2	0.1±0.1	0.1±0.1	1.9±0.9	7.6±4.6	0.0±0.0	0.1±0.1	2.4±1.8
FedNCN	64.2±1.2	23.5±1.0	21.6±2.3	24.5±2.1	71.2±2.0	21.3±3.1	23.8±4.5	24.0±0.8	70.2±2.6	21.6±1.4	22.9±3.2	32.5±0.8
TCTL	**65.1±1.5**	**23.8±2.2**	**22.2±2.8**	**25.7±2.6**	**72.4±1.6**	**25.2±1.9**	**28.1±3.7**	**27.5±1.1**	**73.3±1.5**	**22.5±2.4**	**24.1±3.8**	**32.6±1.9**
Methods	**Am-Ph (5 Clients)**				**Am-Ph (10 Clients)**				**Am-Ph (20 Clients)**			
FedSage+*	32.5±1.9	10.9±2.3	11.4±2.7	18.7±1.1	34.0±1.2	11.1±1.2	9.0±1.8	18.5±0.6	36.8±1.0	11.3±0.2	9.8±0.3	26.5±0.5
FedPUB*	14.9±7.6	0.0±0.0	0.0±0.0	3.4±1.7	8.4±3.2	0.0±0.0	0.0±0.0	1.9±0.5	12.7±6.8	15.2±0.0	15.2±0.0	6.0±3.6
FedTAD*	13.5±5.0	4.1±6.8	4.0±6.8	3.7±2.5	13.1±3.6	5.4±2.2	5.5±2.5	4.0±1.7	11.8±5.0	8.3±1.0	8.6±1.2	4.2±1.4
FedGTA*	35.5±2.0	14.3±4.0	11.9±4.1	12.0±1.1	32.7±1.1	11.8±0.8	9.6±1.1	10.8±0.6	29.4±1.2	10.8±0.8	7.8±0.9	9.6±0.3
FedIIH*	14.3±4.2	0.0±0.0	0.0±0.0	3.1±1.0	11.7±7.8	0.0±0.0	0.0±0.0	2.7±1.9	12.1±7.8	15.2±0.1	15.2±0.0	4.1±3.5
FedNCN	74.0±3.0	37.2±5.3	43.0±7.0	29.1±2.8	73.5±1.8	29.2±2.1	31.1±3.4	27.6±1.1	66.1±2.1	19.9±1.6	16.7±2.0	32.5±0.9
TCTL	**74.8±2.6**	**41.9±4.0**	**49.1±5.0**	**29.7±2.8**	**74.9±2.2**	**30.1±2.2**	**31.5±4.4**	**29.5±2.4**	**67.2±2.2**	**22.2±1.3**	**19.4±2.1**	**33.7±1.0**
Methods	**Questions (5 Clients)**				**Questions (10 Clients)**				**Questions (20 Clients)**			
FedSage+*	78.1±2.1	1.0±1.5	1.3±1.5	39.2±0.8	76.3±1.5	0.8±1.7	0.8±2.2	38.3±1.2	76.0±0.6	0.6±0.7	0.6±1.0	36.2±0.8
FedPUB*	59.4±46.0	0.0±0.0	0.0±0.0	30.7±22.7	59.4±46.1	0.0±0.0	0.0±0.0	30.7±22.7	40.6±46.0	0.0±0.0	−0.0±0.1	21.4±22.7
FedTAD*	56.0±27.8	0.2±0.1	−0.8±0.8	35.8±12.3	66.3±31.3	0.1±0.1	−0.3±0.3	38.6±14.8	69.1±34.4	0.1±0.1	−0.3±0.4	38.7±13.0
FedGTA*	88.5±4.5	0.6±0.3	3.2±1.6	50.5±1.8	88.1±3.0	0.4±0.1	1.3±0.4	49.6±0.5	86.7±2.4	0.7±0.7	0.7±2.1	47.8±1.5
FedIIH*	77.4±6.2	0.0±0.0	0.0±0.0	30.7±22.7	78.3±3.4	0.0±0.0	0.0±0.0	30.7±22.7	80.1±2.8	0.4±0.7	0.6±1.3	41.6±2.9
FedNCN	94.6±2.0	1.9±0.9	6.4±2.9	52.9±1.7	91.3±4.7	1.6±0.8	4.6±2.3	51.4±2.1	93.0±5.1	1.1±0.8	2.9±3.3	50.8±2.3
TCTL	**95.9±2.1**	**2.1±1.1**	**6.8±2.2**	**53.2±1.9**	**93.2±1.3**	**1.8±0.9**	**5.1±1.8**	**52.2±1.5**	**95.5±3.2**	**1.3±0.7**	**3.1±2.2**	**51.0±1.9**

* Note that these supervised node-level FGL methods are adapted to the unsupervised scenario.

Baseline Methods. To evaluate the effectiveness of TCTL, we compare it with three categories of baseline methods. The first category includes advanced federated graph learning (FGL) methods, namely FedSage+ (Zhang et al., 2021), FedPUB (Baek et al., 2023), FedTAD (Zhu et al., 2024), FedGTA (Li et al., 2024), and FedIIH (Yu et al., 2025). The second category consists of FedNCN (Liu et al., 2025b), the most closely related federated graph clustering method. The third category contains classical federated learning aggregation strategies, including FedAvg (McMahan et al., 2017), FedProx (Li et al., 2020), and FedPer (Arivazhagan et al., 2019). For a fair comparison, these aggregation strategies are implemented with our local model while maintaining the same training protocol. Detailed descriptions of all baseline methods are provided in Appendix C.2.

Implementation Details. For a fair evaluation, all methods are tested under the same hardware environment and experimental settings. We use a four-layer GNN on both the client and server sides to obtain node embeddings, with hidden dimensions of 500–500–2000–10. In addition, a one-layer MLP is employed to extract local clustering signals, which are subsequently uploaded to the server for cross-client knowledge aggregation. We optimize all models using Adam (Xiao et al., 2024) with a learning rate of 1×10^{-3}. The client–server interaction is performed for 20 communication rounds, and the local model is trained for 10 epochs in each round. All methods are implemented with PyTorch 2.4.0 and executed on a single NVIDIA GeForce RTX 4090 GPU.

Clustering Metrics. To comprehensively assess clustering performance, we adopt four widely used metrics, including Accuracy (ACC) (Liu et al., 2022b; Wang et al., 2022; Cai et al., 2022), Normalized Mutual Information (NMI) (Liu et al., 2022a; Liang et al., 2023), Adjusted Rand Index

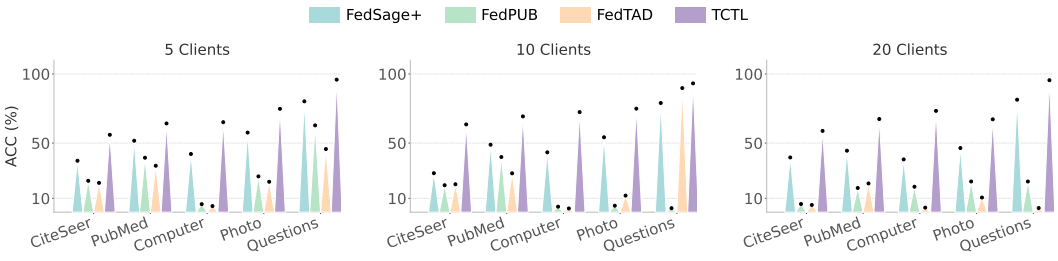


Figure 3: ACC comparison of TCTL and three supervised FGL methods on benchmark datasets.

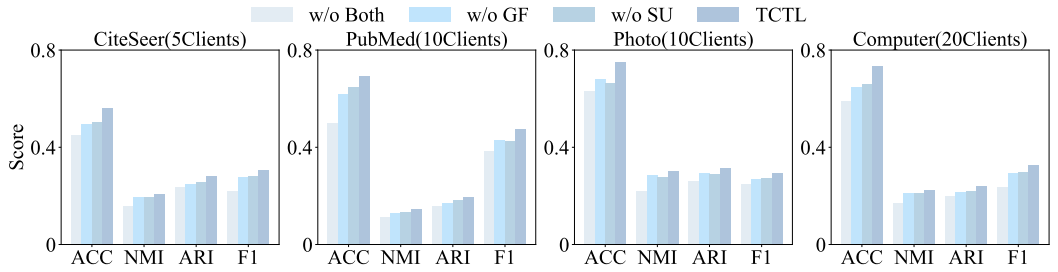


Figure 4: Ablation study of key components in TCTL.

(ARI) (Yang et al., 2023; Tu et al., 2022), and F1 Score (F1) (Tu et al., 2023; 2024; 2025). The detailed metric definitions are provided in Appendix C.3.

4.2 Experimental Results

Comparison in Unsupervised Learning Settings. To evaluate the clustering performance of TCTL under the unsupervised setting, we compare it with FedNCN and five representative federated graph learning methods, including FedSage+, FedPUB, FedTAD, FedGTA, and FedIIH. For a fair comparison under the unsupervised clustering setting, these federated graph learning methods are adapted to perform clustering without label information. To reduce the impact of randomness, each experiment is repeated five times, and the mean values with standard deviations of ACC, NMI, ARI, and F1 are reported in Table 1. The results demonstrate that TCTL consistently achieves the best performance among all compared methods across five benchmark datasets, three client configurations, and four evaluation metrics. Compared with FedNCN, the most closely related federated graph clustering method, TCTL obtains substantial improvements across ACC, NMI, ARI, and F1 under different client partitions, demonstrating the effectiveness of the proposed task-guided cross-subgraph topology learning strategy. Furthermore, the adapted federated graph learning baselines show limited clustering capability without label information, whereas TCTL achieves superior performance by effectively learning informative cross-subgraph topology under the unsupervised setting.

Comparison with Supervised FGL Methods. To further investigate the effectiveness of TCTL in the absence of label supervision, we compare it with five representative supervised FGL methods, including FedSage+, FedPUB, FedTAD, FedGTA, and FedIIH. Following previous settings, these supervised baselines are provided with 5% node labels, while TCTL is trained in a fully unsupervised manner. As shown in Fig. 3, TCTL achieves consistently higher ACC than FedSage+, FedPUB, and FedTAD across all five datasets and different client partitions, demonstrating that the proposed task-guided cross-subgraph topology learning strategy can effectively capture clustering structures without relying on label information. The results of FedGTA and FedIIH, as well as additional comparisons with classical FL aggregation strategies, are reported in Appendix C.4.

Ablation Study. To investigate the contribution of the key components in TCTL, we conduct ablation experiments by removing GF, SU, and both components simultaneously. Specifically, "w/o GF", "w/o SU", and "w/o Both" denote the variants without gradient feedback, shared update mechanism, and both components, respectively. Figure 4 reports the ACC, NMI, ARI, and F1 results

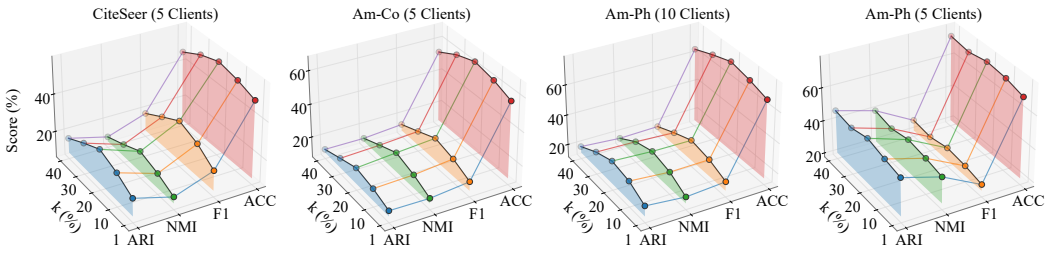


Figure 5: Sensitivity of TCTL to the hyper-parameter k under different datasets and client settings.

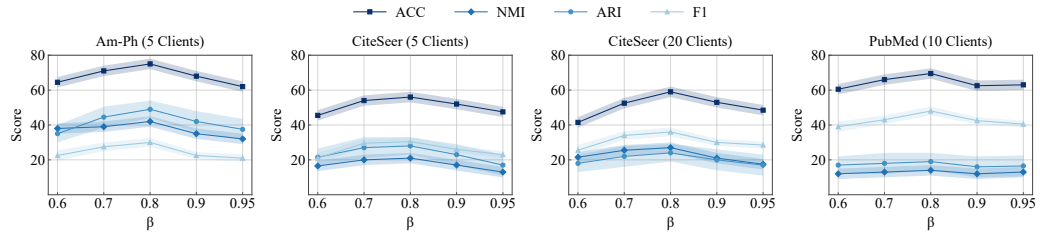


Figure 6: Sensitivity of TCTL to the hyper-parameter β under different datasets and client settings.

on CiteSeer (5 Clients), PubMed (10 Clients), Am-Ph (10 Clients), and Am-Co (20 Clients). The full TCTL model consistently achieves the best overall performance across different datasets and metrics. Removing either GF or SU results in performance degradation, while removing both components leads to a further performance drop. These results demonstrate that gradient feedback and shared update mechanism jointly contribute to federated node-level clustering.

Hyper-Parameter Analysis. We investigate the sensitivity of TCTL to two important hyper-parameters, i.e., the representative-node selection ratio k and the smoothing coefficient β. For the representative-node selection ratio k, which controls the number of representative samples used to preserve cluster-level semantics, we vary k over $\{1\%, 10\%, 20\%, 30\%, 40\%\}$ and evaluate TCTL on CiteSeer, Am-Co, and Am-Ph. The results are shown in Figure 5. TCTL achieves stable clustering performance under a moderate range of k, while extremely small or large values may degrade performance. A small k provides insufficient representative nodes to capture comprehensive cluster semantics, whereas a large k introduces redundant information and potential noise. For the smoothing coefficient β, we vary it over $\{0.6, 0.7, 0.8, 0.9, 0.95\}$. The results in Figure 6 show that an appropriate β achieves better performance by effectively balancing current and historical clustering information, while overly large values may lead to slight performance degradation on some datasets.

5 Conclusion

In this paper, we present **T**ask-guided **C**ross-subgraph **T**opology **L**earning (TCTL), a novel federated graph clustering framework designed to address missing cross-subgraph edges under privacy constraints. By explicitly aligning cross-subgraph topology learning with the clustering objective, TCTL identifies and strengthens sample connections that are beneficial for clustering tasks while mitigating the adverse effects of noisy or task-irrelevant edges. TCTL further leverages complementary clustering knowledge distributed across clients and the server, enabling reliable global topology learning without exposing raw local graph data. Experiments on five benchmark datasets demonstrate that TCTL consistently achieves superior clustering performance over its competitors. Future work will explore more adaptive and scalable topology learning strategies for broader federated graph learning scenarios, including heterogeneous and dynamic graph environments.

AI use statement

No generative AI tools were used in the preparation of this paper.

REFERENCES

Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

Jinheon Baek, Wonyong Jeong, Jiongdao Jin, Jaehong Yoon, and Sung Ju Hwang. Personalized subgraph federated learning. In *Proceedings of the International Conference on Machine Learning*, pp. 1396–1415, 2023.

Jinyu Cai, Jicong Fan, Wenzhong Guo, Shiping Wang, Yunhe Zhang, and Zhao Zhang. Efficient deep embedded subspace clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21–30, 2022.

Renxiang Guan, Wenxuan Tu, Siwei Wang, Jiyuan Liu, Dayu Hu, Chang Tang, Yu Feng, Junhong Li, Baili Xiao, and Xinwang Liu. Structure-adaptive multi-view graph clustering for remote sensing data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 16933–16941, 2025.

Renda Han, Junlong Wu, Wenxuan Tu, Jingxin Liu, Haotian Wang, and Jieren Cheng. Federated graph-level clustering network with attribute inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 21585–21593, 2026.

Bo Jiang, Ziyan Zhang, Sheng Ge, Beibei Wang, Xiao Wang, and Jin Tang. Learning graph attentions via replicator dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):7720–7727, 2024.

Runze Lei, Pinghui Wang, Junzhou Zhao, Lin Lan, Jing Tao, Chao Deng, Junlan Feng, Xidian Wang, and Xiaohong Guan. Federated learning over coupled graphs. *IEEE Transactions on Parallel and Distributed Systems*, 34(4):1159–1172, 2023.

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

Xunkai Li, Zhengyu Wu, Wentao Zhang, Yinlin Zhu, Rong-Hua Li, and Guoren Wang. Fedgta: Topology-aware averaging for federated graph learning. *arXiv preprint arXiv:2401.11755*, 2024.

Ke Liang, Yue Liu, Sihang Zhou, Wenxuan Tu, Yi Wen, Xihong Yang, Xiangjun Dong, and Xinwang Liu. Knowledge graph contrastive learning based on relation-symmetrical structure. *IEEE Transactions on Knowledge and Data Engineering*, 36(1):226–238, 2023.

Zhiping Lin, Zhao Kang, Lizong Zhang, and Ling Tian. Multi-view attributed graph clustering. *IEEE Transactions on Knowledge and Data Engineering*, 35(2):1872–1880, 2021.

Jingxin Liu, Jieren Cheng, Renda Han, Wenxuan Tu, Jiaxin Wang, and Xin Peng. Federated graph-level clustering network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 18870–18878, 2025a.

Jingxin Liu, Renda Han, Wenxuan Tu, Haotian Wang, Junlong Wu, and Jieren Cheng. Federated node-level clustering network with cross-subgraph link mending. In *Proceedings of the International Conference on Machine Learning*, 2025b.

Jingxin Liu, Wenxuan Tu, Renda Han, Junlong Wu, Haotian Wang, Guohui Liu, Xiangyan Tang, and Yue Yang. Personalized federated graph-level clustering network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 23721–23729, 2026a.

Jingxin Liu, Wenxuan Tu, Haotian Wang, Renda Han, Haoyi Li, Junlong Wu, and Xiangyan Tang. Causally-aware attribute completion for incomplete federated graph clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 23730–23738, 2026b.

Suyuan Liu, Siwei Wang, Pei Zhang, Kai Xu, Xinwang Liu, Changwang Zhang, and Feng Gao. Efficient one-pass multi-view subspace clustering with consensus anchors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 7576–7584, 2022a.

Yue Liu, Wenxuan Tu, Sihang Zhou, Xinwang Liu, Linxuan Song, Xihong Yang, and En Zhu. Deep graph clustering via dual correlation reduction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 7603–7611, 2022b.

Yue Liu, Xihong Yang, Sihang Zhou, Xinwang Liu, Siwei Wang, Ke Liang, Wenxuan Tu, and Liang Li. Simple contrastive graph clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 35(10):13789–13800, 2023.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 1273–1282, 2017.

Liang Peng, Nan Wang, Nicha Dvornek, Xiaofeng Zhu, and Xiaoxiao Li. Fedni: Federated graph learning with network inpainting for population-based disease prediction. *IEEE Transactions on Medical Imaging*, 42(7):2032–2043, 2022.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of gnns under heterophily: Are we really making progress? In *Proceedings of the International Conference on Learning Representations*, 2023.

Yuncan Tang and Yongquan Liang. Credit card fraud detection based on federated graph learning. *Expert Systems with Applications*, 256:124979, 2024.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Proceedings of the Conference on Neural Information Processing Systems*, 2017.

Wenxuan Tu, Sihang Zhou, Xinwang Liu, Yue Liu, Zhiping Cai, En Zhu, Changwang Zhang, and Jieren Cheng. Initializing then refining: A simple graph attribute imputation network. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 3494–3500, 2022.

Wenxuan Tu, Qing Liao, Sihang Zhou, Xin Peng, Chuan Ma, Zhe Liu, Xinwang Liu, Zhiping Cai, and Kunlun He. Rare: Robust masked graph autoencoder. *IEEE Transactions on Knowledge and Data Engineering*, 36(10):5340–5353, 2023.

Wenxuan Tu, Bin Xiao, Xinwang Liu, Sihang Zhou, Zhiping Cai, and Jieren Cheng. Revisiting initializing then refining: An incomplete and missing graph imputation network. *IEEE Transactions on Neural Networks and Learning Systems*, 36(2):3244–3257, 2024.

Wenxuan Tu, Sihang Zhou, Xinwang Liu, Zhiping Cai, Yawei Zhao, Yue Liu, and Kunlun He. Wage: Weight-sharing attribute-missing graph autoencoder. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(7):5760–5777, 2025.

Siwei Wang, Xinwang Liu, and En Zhu. Late fusion multi-view clustering via global and local alignment maximization. *arXiv preprint arXiv:2208.01198*, 2022.

Xiaobao Wang, Renda Han, Ronghao Fu, and Di Jin. Federated graph-level clustering network with dual knowledge separation. In *Proceedings of the International Conference on Learning Representations*, 2026a.

Youqing Wang, Tianxiang Zhao, Mengyuan Xin, Ye Su, Jiapu Wang, Tengfei Liu, Junbin Gao, and Jipeng Guo. Deep multi-view graph clustering via attribute-aware bidirectional structural refinement and pseudo-label guided multi-level fusion. In *Proceedings of the International Conference on Machine Learning*, 2026b.

Nachuan Xiao, Xiaoyin Hu, Xin Liu, and Kim-Chuan Toh. Adam-family methods for nonsmooth optimization with convergence guarantees. *Journal of Machine Learning Research*, 25(48):1–53, 2024.

Xihong Yang, Yue Liu, Sihang Zhou, Siwei Wang, Wenxuan Tu, Qun Zheng, Xinwang Liu, Liming Fang, and En Zhu. Cluster-guided contrastive graph clustering network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 10834–10842, 2023.

Wentao Yu, Shuo Chen, Yongxin Tong, Tianlong Gu, and Chen Gong. Modeling inter-intra heterogeneity for graph federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 22236–22244, 2025.

Jingyun Zhang, Hao Peng, Li Sun, Guanlin Wu, Chunyang Liu, and Zhengtao Yu. Unsupervised graph clustering with deep structural entropy. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 3752–3763, 2025.

Ke Zhang, Carl Yang, Xiaoxiao Li, Lichao Sun, and Siu Ming Yiu. Subgraph federated learning with missing neighbor generation. In *Proceedings of the Conference on Neural Information Processing Systems*, pp. 6671–6682, 2021.

Yinlin Zhu, Xunkai Li, Zhengyu Wu, Di Wu, Miao Hu, and Rong-Hua Li. Fedtad: Topology-aware data-free knowledge distillation for subgraph federated learning. *arXiv preprint arXiv:2404.14061*, 2024.

A Notations

All notations used in the proposed TCTL are summarized in Table 2.

B Detailed Training Procedure of TCTL

The overall training procedure of TCTL is summarized in Algorithm 2.

C Additional Experimental Details

C.1 Benchmark Datasets

We evaluate TCTL on five widely used graph benchmark datasets, including CiteSeer, PubMed, Am-Co, Am-Ph, and Questions. Following FedNCN (Liu et al., 2025b), we partition each dataset into 5, 10, and 20 client subgraphs to simulate the federated setting, where each client only accesses its local subgraph. The detailed statistics of these datasets are summarized in Table 3.

CiteSeer (Liu et al., 2023) and PubMed (Jiang et al., 2024) are two citation graph datasets. In these datasets, nodes represent scientific publications, and edges denote citation relationships between papers. The node features are constructed from textual information of publications, while the labels indicate different research topics.

Amazon-Computer and Amazon-Photo (Lin et al., 2021) are two product graph datasets collected from Amazon. In these datasets, nodes correspond to products, and edges represent relationships between products. Amazon-Computer focuses on computer-related products, while Amazon-Photo contains products related to photography. The node features are derived from product information, and the labels correspond to different product categories.

Questions (Platonov et al., 2023) is a question-answering graph dataset, where nodes represent questions and edges indicate relationships among questions. The node features describe question information, and the labels correspond to different question categories.

The statistics of all benchmark datasets, including the numbers of nodes, edges, feature dimensions, and clusters, are reported in Table 3.

C.2 Baselines and Our Method

We compare TCTL with several representative federated graph learning and federated graph clustering methods. The detailed descriptions of these methods are provided as follows.

- **FedSage+** (Zhang et al., 2021): This method proposes a federated graph learning framework based on GraphSAGE, which addresses the missing-neighbor problem caused by graph partitioning among clients. It introduces a neighbor generator to recover missing

Table 2: Basic notations used in TCTL.

Notation	Meaning
Matrices and vectors	
$\mathbf{X}_m \in \mathbb{R}^{N_m \times d}$	Node attribute matrix of client m
$\mathbf{A}_m \in \mathbb{R}^{N_m \times N_m}$	Adjacency matrix of client m
$\tilde{\mathbf{A}}_m \in \mathbb{R}^{N_m \times N_m}$	Normalized adjacency matrix of client m
$\mathbf{A}^I \in \mathbb{R}^{N \times N}$	Intra-subgraph topology matrix
$\mathbf{A}^C \in \mathbb{R}^{N \times N}$	Candidate cross-subgraph topology matrix
$\mathbf{A}^\phi \in \mathbb{R}^{N \times N}$	Initial global topology matrix
$\mathbf{A}^\psi \in \mathbb{R}^{N \times N}$	Refined global topology matrix
$\mathbf{H} \in \mathbb{R}^{N \times d_h}$	Node representation matrix
$\mathbf{Z} \in \mathbb{R}^{N \times d_z}$	Latent relation representation matrix
$\mathbf{Q} \in \mathbb{R}^{N \times O}$	Soft cluster assignment matrix
$\mathbf{P} \in \mathbb{R}^{N \times O}$	Target distribution matrix
$\mathbf{C} \in \mathbb{R}^{O \times d_h}$	Cluster center matrix
Basic parameters	
M	Number of clients
N	Number of nodes in the global graph
N_m	Number of nodes in client m
O	Number of clusters
d	Dimension of node attributes
d_h	Dimension of node representations
d_z	Dimension of latent relation representations
Graph-related symbols	
$G_m = \{V_m, E_m\}$	Local subgraph of client m
v_i, v_j	Nodes connected by an edge
E_m	Edge set of client m
a_{ij}	Edge weight between nodes v_i and v_j
$\hat{a}_{ij}$	Updated edge weight between nodes v_i and v_j
s_p	Cluster-level semantic representation of subgraph p
Learning-related symbols	
$r_{ij}^{\Phi_t}$	Relation score between nodes v_i and v_j
g_{ij}	Gradient-based edge influence
$\hat{g}_{ij}$	Normalized edge influence
a_{ij}^{tar}	Task-guided target edge weight
$T_{\Phi_t}(\cdot)$	Edge-weight update function
$H(\mathbf{A}, \mathbf{Q})$	Task-feedback function
L_{upd}	Edge-weight update loss
L_{global}	Global clustering objective
β	EMA smoothing coefficient
$\mathrm{sg}(\cdot)$	Stop-gradient operator

neighbor information and improves local representation learning under the federated setting.

- **FedPUB** (Baek et al., 2023): This method proposes a personalized subgraph federated learning framework. The server constructs auxiliary graphs and distributes them to clients, where clients learn functional embeddings for identifying communities and improving model aggregation. It further introduces personalized sparse masks to select important parameters for local updates.

Table 3: Dataset summary.

Dataset	Nodes	Edges	Dimensions	Clusters	Type	Domain
CiteSeer	3327	9104	3703	6	Graph	Citation
PubMed	19717	88648	500	3	Graph	Citation
Amazon-Computer	13752	491722	767	10	Graph	Computer Product
Amazon-Photo	7650	238162	754	8	Graph	Photo Product
Questions	48921	153540	301	2	Graph	Q&A System

- **FedTAD** (Zhu et al., 2024): This method is a topology-aware subgraph federated learning framework that aims to alleviate subgraph heterogeneity. It decouples node and structural variations among clients and applies knowledge distillation to transfer reliable knowledge from local models to the global model.
- **FedGTA** (Li et al., 2024): This method introduces a topology-aware optimization strategy for federated graph learning. It employs topology-aware feature aggregation to enhance the capability of local models to capture graph structural information and improve scalability for large-scale graph data.
- **FedIIH** (Yu et al., 2025): This method is a federated graph learning framework designed to address inter-client and intra-client heterogeneity. It models subgraph variations through hierarchical representations and disentangles latent factors to better handle heterogeneous graph distributions.
- **FedNCN** (Liu et al., 2025b): This method is a federated node-level clustering framework that focuses on recovering missing connections across distributed subgraphs. It learns client-side clustering signals and performs cross-subgraph link mending on the server side to construct a global graph. A GNN-based clustering model is then optimized on the recovered graph to improve clustering performance.
- **FedAvg** (McMahan et al., 2017): This method is a classical federated learning aggregation strategy that performs local model optimization on each client and aggregates local parameters by weighted averaging at the server. FedAvg serves as a fundamental baseline for evaluating the effectiveness of federated optimization methods.
- **FedProx** (Li et al., 2020): This method extends FedAvg by introducing a proximal regularization term into the local objective to alleviate the impact of statistical heterogeneity among clients. By constraining the deviation between local and global models, FedProx improves training stability under heterogeneous federated environments.
- **FedPer** (Arivazhagan et al., 2019): This method is a personalized federated learning approach that separates the model into shared and personalized components. The shared parameters are aggregated across clients, while the personalized parameters are kept locally to capture client-specific characteristics.
- **Local**: This method denotes the non-federated setting, where the model is trained locally on each complete subgraph without communication among clients.
- **TCTL**: This is our proposed task-guided cross-subgraph topology learning framework. Different from existing methods that rely on structural similarity or parameter aggregation, TCTL jointly leverages client-side cluster semantic information and server-side task feedback to refine cross-subgraph topology. By optimizing edge weights under clustering guidance, TCTL learns more informative global topology and improves federated graph clustering.

C.3 Details of Evaluation Metrics

To comprehensively evaluate the clustering performance, we employ four widely used unsupervised clustering metrics, including Clustering Accuracy (ACC), Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), and Macro-F1 score. The detailed definitions of these metrics are provided as follows.

Algorithm 2 Complete Training Procedure of TCTL

Require: Local subgraphs $\{G_m = (V_m, E_m)\}_{m=1}^{M}$, node attributes $\{X_m\}_{m=1}^{M}$, number of clusters O, communication rounds T, EMA coefficient β
Ensure: Refined topology $\mathbf{A}^{\psi}$ and global clustering state $\bar{\mathbf{\Theta}}$
1: Construct intra-subgraph topology $\mathbf{A}^{I}$.
2: Select representative nodes and construct candidate cross-subgraph topology: $\mathbf{A}^{\phi} = \mathbf{A}^{I} + \mathbf{A}^{C}$.
3: Initialize global clustering state $\mathbf{\Theta}_0 = (\theta_0, \mathbf{C}_0)$ and EMA state $\bar{\mathbf{\Theta}}_0$.
4: **for** $t = 1$ to T **do**
5: Obtain soft cluster assignments: $\mathbf{Q}_t = Q(\mathbf{A}^{\phi}; \bar{\theta}_t, \bar{\mathbf{C}}_t)$.
6: Compute task-feedback objective: $\mathcal{H}(\mathbf{A}^{\phi}, \mathbf{Q}_t)$.
7: Calculate edge contribution: $g_{ij} = -[\nabla_{\mathbf{A}^C} \mathcal{H}(\mathbf{A}^{\phi}, \mathbf{Q}_t)]_{ij}$.
8: Normalize gradient feedback: $\hat{g}_{ij} = g_{ij} / \max_{(i,j)} |g_{ij}|$.
9: Obtain client-derived cluster-level semantic representations.
10: Fuse cluster semantics and gradient feedback to optimize the edge-weight update function $\mathcal{T}_{\Phi_t}$.
11: Update shared parameters: $\Phi_t \to \Phi_{t+1}$.
12: Refine candidate cross-subgraph edges:

$$a_{ij}^{R} = \mathcal{T}_{\Phi_{t+1}}(\mathbf{h}_i, \mathbf{h}_j, \mathbf{s}_p, \mathbf{s}_q, a_{ij}).$$

13: Construct refined cross-subgraph topology: $\mathbf{A}^{R}$.
14: Obtain final topology: $\mathbf{A}^{\psi} = \mathbf{A}^{I} + \mathbf{A}^{R}$.
15: Optimize global clustering objective on $\mathbf{A}^{\psi}$ and update $\mathbf{\Theta}_t$.
16: Update EMA clustering state: $\bar{\mathbf{\Theta}}_{t+1} = \beta \bar{\mathbf{\Theta}}_t + (1 - \beta)\mathbf{\Theta}_{t+1}$.
17: **end for**
18: **return** $\mathbf{A}^{\psi}, \bar{\mathbf{\Theta}}$

Clustering Accuracy (ACC). ACC measures the consistency between the predicted cluster assignments and the ground-truth labels. Since the cluster indexes generated by an unsupervised model are permutation-invariant, we first find the optimal one-to-one mapping between predicted clusters and real categories using the Hungarian algorithm. The ACC is then calculated as:

$$ACC = \max_{\pi \in \mathcal{P}} \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(y_i = \pi(\hat{y}_i)), \tag{12}$$

where N denotes the number of samples, y_i and $\hat{y}_i$ represent the ground-truth label and predicted cluster assignment of sample i, respectively. π indicates a possible permutation mapping between cluster indexes and class labels, and $\mathbb{I}(\cdot)$ is the indicator function.

Normalized Mutual Information (NMI). NMI evaluates the amount of shared information between the predicted clustering results and the ground-truth distribution. It measures the dependency between two label assignments while normalizing their individual uncertainties. The definition is given by:

$$NMI = \frac{2I(Y; \hat{Y})}{H(Y) + H(\hat{Y})}, \tag{13}$$

where $I(Y; \hat{Y})$ denotes the mutual information between the predicted labels and the ground-truth labels, while $H(\cdot)$ represents the entropy function. The value of NMI ranges from 0 to 1, where a larger value indicates stronger agreement between the two clustering assignments.

Adjusted Rand Index (ARI). ARI measures the similarity between two clustering partitions by considering the pairwise relationships among samples. Compared with the original Rand Index, ARI further adjusts the score by removing the expected agreement caused by random assignments. It is defined as:

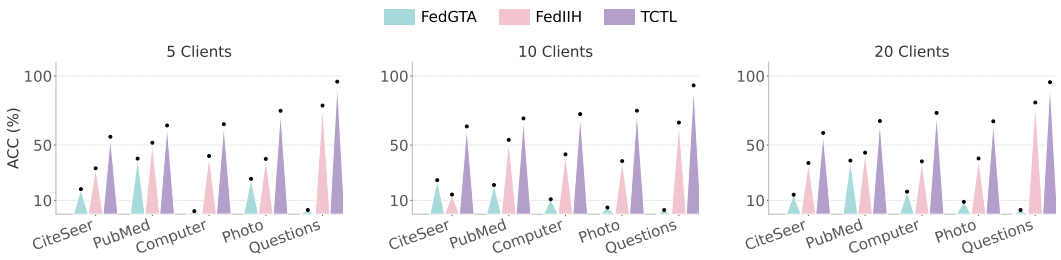


Figure 7: Performance comparison of TCTL with two advanced federated node-level classification methods on five graph datasets.

$$ARI = \frac{RI - \mathbb{E}[RI]}{\max(RI) - \mathbb{E}[RI]}, \tag{14}$$

where RI represents the Rand Index and $\mathbb{E}[RI]$ denotes its expected value under random clustering. A higher ARI value indicates that the obtained clusters are more consistent with the ground-truth partition.

Macro-F1 Score. The Macro-F1 score provides a balanced evaluation by combining precision and recall across all categories. Specifically, we first calculate the F1 score of each class and then average them over all classes:

$$F1 = \frac{1}{C}\sum_{c=1}^{C}\frac{2 \cdot Precision_c \cdot Recall_c}{Precision_c + Recall_c}, \tag{15}$$

where C is the number of classes, and $Precision_c$ and $Recall_c$ denote the precision and recall values of class c, respectively. Macro-F1 gives equal importance to each category and is therefore suitable for evaluating clustering performance under imbalanced class distributions.

C.4 Additional Experimental Results

Additional Comparison with Supervised FGL Methods. To complement the supervised FGL comparisons presented in the main text, we further evaluate TCTL against FedGTA and FedIIH in this section. Following the same experimental protocol, both supervised baselines are trained with 5% of the node labels, while TCTL is trained without label supervision. As illustrated in Figure 7, TCTL achieves higher ACC than both FedGTA and FedIIH on all five benchmark datasets under the 5-, 10-, and 20-client settings. These results provide additional evidence that TCTL can maintain strong clustering performance even when compared with supervised FGL methods that have access to a limited amount of labeled data.

Comparison of FL Aggregation Strategies. To investigate the effectiveness of TCTL under different federated aggregation schemes, we compare it with three classical FL aggregation strategies, including FedAvg, FedProx, and FedPer, as well as the local-only baseline. The experiments are conducted under 5-, 10-, and 20-client settings on five unlabeled benchmark graph datasets, with results reported in Table 4. Overall, TCTL consistently achieves strong clustering performance across different datasets and client configurations. Compared with local-only training, TCTL obtains average improvements of $+14.9$, $+5.9$, $+20.5$, $+30.9$, and $+9.6$ points on CiteSeer, PubMed, Am-Co, Am-Ph, and Questions, respectively, under the 5-client setting. Under the 10-client setting, TCTL achieves average improvements of $+12.9$, $+11.1$, $+21.2$, $+23.5$, and $+7.2$ points, respectively. Under the 20-client setting, the corresponding improvements are $+14.4$, $+10.7$, $+20.7$, $+10.9$, and $+5.9$ points. These results demonstrate that TCTL consistently outperforms conventional FL aggregation strategies and local-only training under different client partitions. This advantage can be attributed to the proposed task-guided cross-subgraph topology learning mechanism, which leverages local clustering knowledge and server-side task feedback to adaptively refine cross-subgraph

Table 4: The performance of different federated learning strategies in federated node-level clustering under different client settings. Here, "Local" denotes using only the local model, while "FedAvg", "FedProx", and "FedPer" apply different server-side aggregation strategies with the same local model.

Dataset	Method	ACC	NMI	ARI	F1	Dataset	Method	ACC	NMI	ARI	F1	Dataset	Method	ACC	NMI	ARI	F1
CiteSeer (5 Clients)	Local	43.5±2.8	6.5±2.2	5.1±4.2	21.1±1.8	CiteSeer (10 Clients)	Local	44.1±1.7	8.6±1.2	4.1±2.1	23.5±0.5	CiteSeer (20 Clients)	Local	45.5±0.6	10.8±0.8	4.0±1.0	28.0±1.0
	FedAvg	53.3±2.2	3.9±1.2	1.9±1.7	16.6±1.6		FedAvg	55.9±2.8	4.2±0.5	0.4±0.8	16.5±0.7		FedAvg	53.9±1.1	8.0±0.8	1.4±0.8	22.3±1.5
	FedProx	41.2±1.9	4.9±1.0	3.2±1.5	21.0±0.7		FedProx	45.0±3.0	8.1±0.9	3.3±1.3	23.7±0.7		FedProx	48.3±3.6	10.1±0.5	2.8±0.3	27.1±1.1
	FedPer	53.1±1.8	7.3±2.2	8.6±2.5	21.6±1.2		FedPer	48.7±1.0	7.8±2.3	4.7±3.0	22.8±1.0		FedPer	47.8±1.3	11.0±0.8	4.3±0.9	28.0±1.5
	TCTL	**56.0±1.2**	**20.9±2.8**	**28.2±4.1**	**30.7±2.4**		**TCTL**	**63.5±1.6**	**18.0±2.2**	**22.0±2.0**	**28.2±2.0**		**TCTL**	**58.8±1.9**	**26.9±1.0**	**23.8±1.8**	**36.4±1.3**
PubMed (5 Clients)	Local	55.5±2.1	9.5±1.3	9.5±1.5	38.1±0.7	PubMed (10 Clients)	Local	54.4±1.5	8.0±1.9	7.1±1.9	37.0±1.0	PubMed (20 Clients)	Local	54.3±0.9	6.3±0.8	4.9±1.2	35.4±1.0
	FedAvg	61.7±0.5	9.6±0.4	9.0±0.9	30.8±1.8		FedAvg	63.7±2.9	5.2±1.7	4.8±0.7	23.9±6.9		FedAvg	64.2±3.2	6.4±2.3	5.1±1.5	20.0±3.8
	FedProx	60.5±1.7	10.3±4.2	11.0±3.6	39.8±2.0		FedProx	56.5±0.9	2.1±0.6	3.2±0.6	35.3±0.7		FedProx	60.8±1.4	6.1±1.3	7.4±1.3	36.7±0.8
	FedPer	56.1±1.7	8.7±0.7	10.6±2.0	38.7±2.3		FedPer	55.5±1.9	7.5±1.0	6.7±1.6	36.8±1.2		FedPer	53.3±0.4	6.2±0.8	4.5±0.8	35.2±0.5
	TCTL	**64.2±1.4**	**11.1±3.2**	**15.4±3.5**	**45.3±2.3**		**TCTL**	**69.3±1.3**	**14.5±1.5**	**19.4±2.8**	**47.5±1.8**		**TCTL**	**67.4±1.6**	**13.1±1.2**	**17.1±1.0**	**45.9±1.7**
Computer (5 Clients)	Local	37.7±1.7	2.9±0.2	1.0±1.3	13.2±0.7	Computer (10 Clients)	Local	45.9±0.3	4.8±0.1	1.9±1.0	15.7±0.3	Computer (20 Clients)	Local	48.1±0.5	3.3±0.1	0.0±0.5	18.2±0.3
	FedAvg	59.6±0.3	0.9±0.2	0.1±0.1	11.4±0.4		FedAvg	67.1±0.0	0.8±0.0	-0.2±0.0	13.5±0.0		FedAvg	64.2±2.0	1.7±0.2	-0.4±0.1	17.9±0.2
	FedProx	39.2±3.9	2.2±0.3	0.5±1.4	13.1±0.7		FedProx	43.1±3.0	2.8±0.3	0.7±0.7	15.9±0.4		FedProx	50.2±1.6	3.1±0.2	1.7±0.4	19.9±0.4
	FedPer	45.7±1.8	2.1±0.2	0.7±0.8	13.3±1.0		FedPer	50.2±1.1	2.5±0.7	1.1±1.0	15.7±0.5		FedPer	52.9±1.5	3.6±0.8	2.4±0.8	20.2±0.6
	TCTL	**65.1±1.5**	**23.8±2.2**	**22.2±2.8**	**25.7±2.6**		**TCTL**	**72.4±1.6**	**25.2±1.9**	**28.1±3.7**	**27.5±1.1**		**TCTL**	**73.3±1.5**	**22.5±1.9**	**24.1±3.8**	**32.6±1.9**
Photo (5 Clients)	Local	47.0±0.6	4.7±0.3	3.8±0.9	16.3±0.5	Photo (10 Clients)	Local	48.2±0.8	5.7±0.5	2.4±0.9	15.8±0.5	Photo (20 Clients)	Local	55.1±0.9	9.7±2.0	8.6±0.3	25.5±0.5
	FedAvg	63.4±0.5	0.9±0.3	0.0±0.3	12.3±0.6		FedAvg	63.5±1.3	3.1±0.8	1.6±0.3	15.0±0.6		FedAvg	65.5±2.2	6.4±2.9	5.0±2.5	23.2±4.9
	FedProx	49.1±1.2	5.2±1.0	7.7±1.4	17.1±0.7		FedProx	49.7±1.8	5.3±0.1	4.2±1.2	16.9±0.3		FedProx	58.0±2.1	10.2±0.4	8.4±0.7	26.7±0.6
	FedPer	55.2±1.6	5.5±1.1	7.1±1.6	16.4±0.9		FedPer	57.0±1.4	5.3±0.8	4.7±1.0	17.7±0.7		FedPer	63.4±0.8	12.2±0.9	11.7±1.3	27.9±0.8
	TCTL	**74.8±2.6**	**41.9±4.0**	**49.1±5.0**	**29.7±2.8**		**TCTL**	**74.9±2.2**	**30.1±2.2**	**31.5±4.4**	**29.5±2.4**		**TCTL**	**67.2±2.2**	**22.2±1.3**	**19.4±2.1**	**33.7±1.0**
Questions (5 Clients)	Local	75.1±2.4	0.4±0.3	-0.7±0.5	44.7±1.0	Questions (10 Clients)	Local	78.1±1.8	0.5±0.1	-0.7±0.7	45.8±0.2	Questions (20 Clients)	Local	80.7±1.6	0.7±0.2	-0.4±0.5	46.5±0.4
	FedAvg	87.4±0.8	0.0±0.0	0.0±0.0	46.2±0.0		FedAvg	87.6±0.8	0.0±0.0	0.0±0.0	46.2±0.0		FedAvg	88.2±1.5	0.0±0.0	-0.1±0.0	46.2±0.0
	FedProx	85.3±4.1	0.3±0.1	1.6±1.1	48.8±1.5		FedProx	84.4±1.3	0.3±0.0	1.9±0.2	49.2±0.3		FedProx	81.7±0.8	0.6±0.2	-0.1±1.2	47.0±0.5
	FedPer	84.1±2.0	0.3±0.3	1.1±0.9	48.4±0.8		FedPer	82.2±2.1	0.3±0.1	1.6±0.7	48.4±0.9		FedPer	81.3±0.5	0.5±0.2	-0.2±0.8	46.9±0.6
	TCTL	**95.9±2.1**	**2.1±1.1**	**6.8±2.2**	**53.2±1.9**		**TCTL**	**93.2±1.3**	**1.8±0.9**	**5.1±1.8**	**52.2±1.5**		**TCTL**	**95.5±3.2**	**1.3±0.7**	**3.1±2.2**	**51.0±1.9**

connections, thereby facilitating more effective cross-client information exchange and clustering-oriented representation learning.