DISTRIBUTION-CALIBRATED DIFFUSION FOR ONE-SHOT FEDERATED GRAPH CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning enables multiple clients to collaboratively train models without sharing their private data. However, conventional approaches typically require repeated client–server communication, leading to substantial communication overhead and repeated exposure of client model parameters. One-shot federated learning has therefore attracted increasing attention by restricting client–server interaction to a single communication round. Recently, this paradigm has been extended from image data to graph data, giving rise to one-shot federated graph learning. However, existing one-shot federated graph learning studies mainly focus on node-level tasks, while one-shot learning for graph-level classification remains largely unexplored. Under strict one-shot communication, the server must reconstruct reliable graph distributions and reconcile heterogeneous client data without any subsequent client feedback. To bridge this gap, we propose a one-shot federated graph classification framework that leverages diffusion models to transfer and reconstruct client graph distributions. Specifically, clients upload diffusion-based generative knowledge and compact graph distribution summaries only once, after which the server generates, validates, and reweights synthetic graphs to reconstruct and fuse client graph distributions. Extensive experiments demonstrate that our framework consistently outperforms state-of-the-art FGL and one-shot FGL baselines.

1 INTRODUCTION

Federated graph learning (FGL) extends federated learning to graph-structured data (McMahan et al., 2017; Chen et al., 2024; Liu et al., 2025), enabling institutions to collaboratively train graph models while retaining raw data locally (Xie et al., 2021; Tan et al., 2023). AI has recently enabled major advances in biomedicine (Abramson et al., 2024; Wong et al., 2024), where small molecules, proteins, and tissue samples can be represented as graphs to support important applications such as molecular property prediction and patient stratification (Wang et al., 2022). With public datasets already widely used for model training, accessing additional, task-relevant data is becoming increasingly important. Companies and hospitals hold valuable experimental and clinical data, yet confidentiality and privacy constraints often prevent these resources from being pooled. Recent cross-company federated protein modeling (Callaway, 2026) and federated graph learning for molecular discovery (Zhang et al., 2026) highlight the value of learning from distributed institutional data, providing renewed motivation for FGL.

Realizing these opportunities calls for graph-level FGL, as many biomedical tasks target entire molecules, proteins, or complexes (Srivastav et al., 2026; Abramson et al., 2024; Team CD et al., 2025; Sosnin, 2025). Graph-level learning differs substantially from node-level prediction, whereas the latter has been a prominent focus of FGL research (Qian et al., 2025; Wan et al., 2025). Graph classification demands cross-graph generalization over heterogeneous structures, a challenge compounded by structural distribution shifts across institutions (Xie et al., 2021; Tan et al., 2023; 2024). Yet multi-round collaboration ties global training to repeated local computation and coordination (Zeng et al., 2024; Kuo et al., 2025). Even communication-efficient CeFGC requires clients to participate again after their initial upload (Wang et al., 2026). For institutions unable to guarantee repeated participation, this dependency can make collaboration infeasible. Strict one-shot learning removes this dependency: each institution contributes once, and the server completes training without further client involvement (Guha et al., 2018; Liu et al., 2025). It thus enables institutions

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.

to contribute without committing to sustained participation throughout global training. Achieving this for graph classification requires a single upload to convey sufficient structural and distributional knowledge for reliable global learning, without subsequent client feedback to correct transfer errors or reconcile heterogeneous contributions.

Recent one-shot FGL methods, including GHOST and OASIS, explore structural knowledge retention and integration primarily for node-level prediction (Qian et al., 2025; Wan et al., 2025). Generative one-shot FL offers a complementary route by constructing synthetic training data for server-side knowledge integration (Zhang et al., 2022; Mendieta et al., 2024). Graph diffusion has also been used to transfer local graph distributions in CeFGC (Wang et al., 2026). For strict one-shot graph classification, illustrated in Figure 1 (left), this suggests learning a global classifier from graph distributions reconstructed entirely at the server. However, generation errors and uneven representation of clients and classes can propagate into global learning without opportunities for client-assisted correction. Reliable learning therefore requires the uploaded knowledge to support graph reconstruction together with server-side assessment and integration of the resulting samples.

To realize this goal, we seek a federated graph classification framework that completes all client–server communication in a single round. However, this strict one-shot setting poses three interdependent challenges (Figure 1, right). First, how can the server reconstruct client graph distributions from a single upload (GENERATE)? After this upload, the server can neither access the original client graphs nor request additional information, so the transferred knowledge must suffice to recover graph-level samples suitable for training. Second, how can the server validate the structural fidelity of generated graphs without querying the clients (VALIDATE)? Even when clients upload graph generators, synthetic samples may deviate from their distributions in size, density, or connectivity, yet clients have already left the training process and cannot verify whether these samples are representative. Third, how can heterogeneous client distributions be fused during global model training (FUSE)? Client generators may introduce different biases and redundant samples, so pooling individually plausible graphs can distort the relative contributions of clients and classes in the global training distribution.

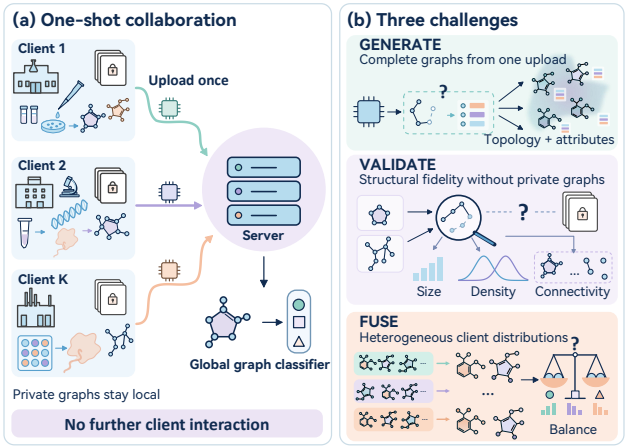


Figure 1: Problem illustration. Left: biotech companies holding private molecular or protein graphs jointly train a global classifier through a single upload; the server then constructs the global model without further client interaction. Right: the three resulting challenges of strict one-shot federated graph classification.

We introduce *OSFGC* (One-Shot Federated Graph Classification), a framework that answers these challenges through three cooperative stages: *distribution-conditioned graph diffusion*, *structure-aware distribution calibration and fusion*, and *sketch-guided global learning and inference*. In the first stage, each client distills its local graph distribution into a class- and size-conditional graph diffusion model together with a class-level *Graph Distribution Sketch* (GDS) of aggregated structural statistics, so that a single upload transfers a samplable surrogate of the local distribution accompanied by a compact signature for verifying it (GENERATE; VALIDATE). In the second stage, the server oversamples candidate graphs from the uploaded generators, screens them against the corresponding sketches for structural consistency and diversity, and fuses the retained candidates through class-conditional reliability-constrained kernel moment matching (CR-KMM), which matches first- and second-order random-feature moments while preserving prescribed client and class masses (FUSE). In the third stage, a global classifier is trained on the weighted synthetic set, with optional prediction calibration from the same sketches. All operations occur at the server after the single upload, and clients never participate again. OSFGC thereby turns one communication round into a verifiable delivery and fusion of client graph distributions. Our contributions are threefold.

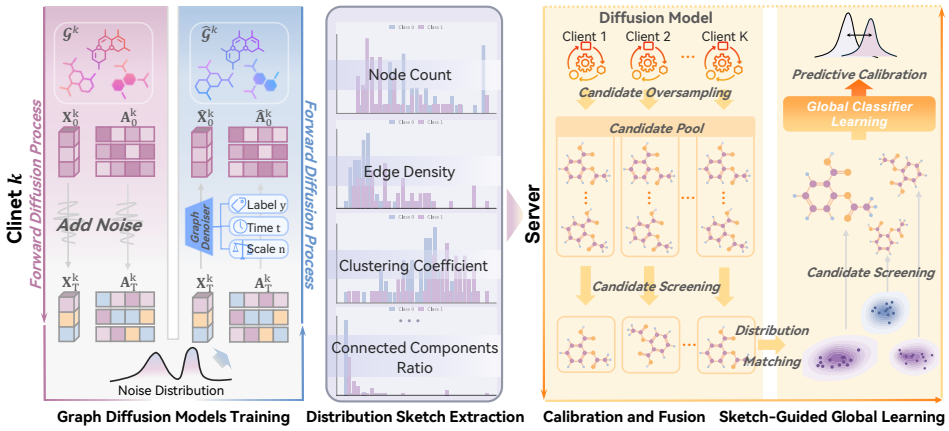


Figure 2: Overview of OSFGC. Each client trains a conditional diffusion model and extracts a Graph Distribution Sketch. After the single upload, the server screens candidates, matches client distributions, and trains a global classifier with sketch-guided calibration.

- We formulate strict one-shot federated graph classification, where each client uploads once and all subsequent generation, validation, fusion, and classifier training complete at the server without further client participation.
- We propose a distribution reconstruction and verification mechanism that couples a class- and size-conditional graph diffusion generator with a class-level Graph Distribution Sketch, and fuses calibrated candidates through reliability-constrained kernel moment matching.
- Experiments across eight federated settings show that OSFGC consistently surpasses one-shot and multi-round baselines in accuracy and Macro-F1.

2 Methodology

2.1 Problem Formulation

Let $G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ denote an attributed graph, where $\mathcal{V}$ and $\mathcal{E}$ denote its node and edge sets. The attributes of its $n = |\mathcal{V}|$ nodes are collected in the feature matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$, whose rows contain the m-dimensional node features. The graph structure is represented by an adjacency matrix $\mathbf{A} \in \{0,1\}^{n \times n}$, where $\mathbf{A}_{uv} = 1$ if nodes u and v are connected and $\mathbf{A}_{uv} = 0$ otherwise. Each graph has a graph-level label $y \in \mathcal{Y} = \{1, \ldots, C\}$, where C is the number of classes. Given a collection of N graphs $\mathcal{G} = \{G_1, \ldots, G_N\}$ and their corresponding labels $y_1, \ldots, y_N$, graph classification aims to learn a classifier F_{ϕ} that predicts labels for unseen graphs from class scores $F_{\phi}(G) \in \mathbb{R}^C$.

In one-shot federated graph classification, a central server collaborates with K clients, where client k holds a private collection $\mathcal{G}_k = \{G_1^k, \ldots, G_{N_k}^k\}$ of N_k labeled graphs. Each observed pair (G_i^k, y_i^k) is a realization of the random graph–label pair $(\mathsf{G}, Y) \sim P_k$, where P_k denotes the local distribution. The objective is to learn a global graph classifier from a single upload of locally derived information per client, while retaining raw graphs locally and requiring no subsequent client interaction, by minimizing the classification loss:

$$\min_{\phi} \sum_{k=1}^{K} \alpha_k \, \mathbb{E}_{(\mathsf{G},Y)\sim P_k} \big[\ell(F_{\phi}(\mathsf{G}), Y) \big], \tag{1}$$

where $\ell : \mathbb{R}^C \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ is the graph-classification loss and $\alpha_k = N_k / \sum_{u=1}^{K} N_u$ weights each client's expected loss by its share of the training graphs. Recent communication-efficient graph classification approaches reduce client–server interaction (Wang et al., 2026), motivating the reconstruction and fusion of heterogeneous graph distributions under a strict single-upload constraint. In this work, we develop a one-shot federated graph classification framework that combines conditional

graph diffusion with distribution sketches to reconstruct local graph distributions and fuse their synthetic samples into a global classifier, to reduce communication overhead and limit opportunities for information exposure through repeated client–server exchanges.

2.2 Distribution-Conditioned Graph Diffusion

To enable the server to reconstruct its local distribution, each client k trains a generator $p_{\boldsymbol{\theta}_k}(G \mid y, n)$ conditioned on class y and node count n. A shared denoiser combines class, time, and normalized-size embeddings; it corrupts categorical node attributes and binary topology through discrete transitions (Austin et al., 2021; Vignac et al., 2023), while applying Gaussian noise to continuous node attributes (Ho et al., 2020).

Forward Diffusion Process. Building on the categorical transition formulation of D3PM (Austin et al., 2021), we corrupt each discrete variable z_t at step $t \in \{1, \dots, T\}$ with a uniform transition kernel. For noise rate $\beta_t \in (0, 1)$, let $\zeta_t = 1-\beta_t$ and $\bar{\zeta}_t = \prod_{s=1}^{t} \zeta_s$. For graphs with continuous node attributes, we adopt the Gaussian corruption $\mathbf{X}_t = \sqrt{\bar{\zeta}_t}\,\mathbf{X}_0 + \sqrt{1-\bar{\zeta}_t}\,\mathbf{E}$ with $\mathbf{E}_{vh} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$. For the discrete components, the one-step transition matrix is

$$\mathbf{Q}_t^{(M_{\text{cat}})} = \zeta_t \mathbf{I}_{M_{\text{cat}}} + \frac{\beta_t}{M_{\text{cat}}} \mathbf{1}_{M_{\text{cat}}} \mathbf{1}_{M_{\text{cat}}}^{\top}, \tag{2}$$

and t-step composition gives the cumulative kernel $\overline{\mathbf{Q}}_t^{(M_{\text{cat}})} = \prod_{s=1}^{t} \mathbf{Q}_s^{(M_{\text{cat}})}$. The resulting marginals for categorical node types $c_{t,v} \in \{1, \dots, M_{\text{cat}}\}$ and upper-triangular edge indicators $\mathbf{A}_{t,uv}$ are

$$\begin{aligned} q(c_{t,v} = j \mid c_{0,v} = i) &= \bar{\zeta}_t \mathbb{I}[i = j] + (1 - \bar{\zeta}_t)/M_{\text{cat}} && \text{(categorical attributes)}, \\ q(\mathbf{A}_{t,uv} = 1 \mid \mathbf{A}_{0,uv}) &= \bar{\zeta}_t \mathbf{A}_{0,uv} + (1 - \bar{\zeta}_t)/2, \qquad u < v && \text{(graph structure)}. \end{aligned} \tag{3}$$

To capture topology, our structural process is the binary specialization of Equation (2): each upper-triangular entry undergoes an independent Bernoulli flip with one-step probability $\beta_t/2$, after which symmetry and a zero diagonal are enforced. As $\bar{\zeta}_T \to 0$, node types, edges, and continuous features approach uniform, Bernoulli$(1/2)$, and standard Gaussian reference distributions, respectively.

Reverse Denoising Process. Given $G_t = (\mathbf{X}_t, \mathbf{A}_t)$ and (y, n, t), the denoiser predicts clean-state probabilities $\widehat{\mathbf{p}}_0 \in [0, 1]^{M_{\text{cat}}}$ for each categorical node or edge variable, together with a clean continuous estimate $\widehat{\mathbf{X}}_0$. For continuous attributes and $t \geq 2$, the Gaussian reverse update is

$$\mathbf{X}_{t-1} = \frac{\sqrt{\bar{\zeta}_{t-1}}\beta_t}{1 - \bar{\zeta}_t} \widehat{\mathbf{X}}_0 + \frac{\sqrt{\zeta_t}(1 - \bar{\zeta}_{t-1})}{1 - \bar{\zeta}_t} \mathbf{X}_t + \sqrt{\widetilde{\beta}_t}\,\mathbf{E}_t, \tag{4}$$

where $\widetilde{\beta}_t = (1 - \bar{\zeta}_{t-1})\beta_t/(1 - \bar{\zeta}_t)$ and $\mathbf{E}_t$ is standard Gaussian noise. For categorical attributes and topology, the reverse step instead mixes the exact forward posterior with the predicted clean-state probabilities:

$$p_{\boldsymbol{\theta}_k}(z_{t-1} = \ell \mid G_t, y, n) = \sum_{i=1}^{M_{\text{cat}}} \widehat{p}_{0,i}\, \overline{\mathbf{Q}}_{t-1,i\ell}^{(M_{\text{cat}})}\, \mathbf{Q}_{t,\ell z_t}^{(M_{\text{cat}})} \,/\, \overline{\mathbf{Q}}_{t,iz_t}^{(M_{\text{cat}})}, \tag{5}$$

where z_t is the current noisy state of the variable being denoised and $\mathbf{Q}_t, \overline{\mathbf{Q}}_t$ are the one-step and cumulative kernels of Equation (2). Iterating these two updates jointly reconstructs attributes and topology, because every prediction is conditioned on the full noisy graph. Appendix A.2 provides the derivations and structural postprocessing.

Diffusion Model Training. Let $\widehat{p}_{0,v,c}^{X}$ and $\widehat{p}_{0,uv}^{A}$ denote the predicted categorical-node and edge probabilities. Client k minimizes the expected reconstruction loss

$$\mathcal{J}_k(\boldsymbol{\theta}_k) = \mathbb{E}_{(G_0,y)\sim\widehat{P}_k,\, t\sim p_t,\, G_t\sim q(\cdot|G_0,t)} \left[\mathcal{L}_{\text{node}} + \mathcal{L}_{\text{edge}}\right], \tag{6}$$

where $\widehat{P}_k$ is the empirical distribution of local labeled graphs and p_t is the public timestep-sampling distribution. Here $\mathcal{L}_{\text{edge}}$ is the binary edge log-loss averaged over node pairs and $\mathcal{L}_{\text{node}}$ is the average node reconstruction error, namely squared error for continuous attributes and negative log-likelihood for categorical ones, with exact forms and edge reweighting in Appendix A.2. The fitted generators are uploaded once and retained as distinct distribution components at the server.

2.3 GRAPH DISTRIBUTION SKETCH EXTRACTION

Beyond the generator, the server requires aggregate evidence of each client's distribution to guide generation and calibrate predictions. Client k therefore extracts a Graph Distribution Sketch (GDS) $\mathcal{S}_k$ from a public permutation-invariant descriptor $\mathbf{s}(G) \in [0,1]^d$ encoding graph size, density, degrees, clustering, connectivity, higher-order structure, and attributes, with public normalization and clipping aligning its coordinates across clients. A shared multi-bandwidth random Fourier feature map $\mathbf{f} : [0,1]^d \to \mathbb{R}^D$ is defined by

$$\mathbf{f}(\mathbf{s}) = \sqrt{2/D}\cos(\mathbf{W}\mathbf{s} + \mathbf{b}), \tag{7}$$

where $\mathbf{W} \in \mathbb{R}^{D\times d}$ has Gaussian rows scaled by inverse bandwidths σ_r and $\mathbf{b} \in \mathbb{R}^D$ is uniform on $[0, 2\pi)$, with σ_r cycling through a public set to approximate a mixture of radial-basis kernels. For each releasable class c containing $N_{k,c}$ local graphs, the client computes

$$\begin{aligned} \widehat{\mathbf{m}}^f_{k,c} &= \frac{1}{N_{k,c}} \sum_{i:y_i^k=c} \mathbf{f}(\mathbf{s}(G_i^k)), \\ \widehat{\mathbf{v}}^f_{k,c} &= \frac{1}{N_{k,c}} \sum_{i:y_i^k=c} \mathbf{f}(\mathbf{s}(G_i^k))^{\odot 2}. \end{aligned} \tag{8}$$

Here $\odot 2$ denotes coordinatewise squaring. The sketch further contains six normalized class-conditional histograms $\mathbf{h}^{(\ell)}_{k,c}$ over size, density, degree, clustering, component ratio, and triangle density, plus node-type and node-degree histograms. Raw descriptor moments $\widetilde{\mathbf{m}}_{k,c}, \widetilde{\mathbf{v}}_{k,c} \in \mathbb{R}^d$ for predictive calibration replace $\mathbf{f}(\mathbf{s}(G))$ with $\mathbf{s}(G)$ in Equation (8). Optional Gaussian perturbation is followed by coordinatewise clipping or histogram renormalization, and tildes denote the resulting released moments. The released generation reliability $\widetilde{q}_{k,c}$ of client k on class c is a clipped, optionally noise-perturbed inverse of the mean local diffusion loss $\overline{\mathcal{L}}^{\text{diff}}_{k,c}$. The released class counts $\widetilde{N}_{k,c}$ and reliability scores $\widetilde{q}_{k,c}$ are set to zero for unreleased classes. Client k uploads these quantities, $\mathcal{S}_k$, and its generator once; raw graphs and classifiers remain local. Appendix A.1 specifies the package and Appendix A.3 details the sketch construction.

2.4 STRUCTURE-AWARE DISTRIBUTION CALIBRATION AND FUSION

Candidate Oversampling and Calibration. The server oversamples separately for each released client–class pair, drawing graph-size conditions from the corresponding GDS histogram and generating candidates with the uploaded conditional diffusion model. Class-conditional node-type histograms adjust categorical probabilities, density histograms determine the final edge count, and degree histograms supply node-level biases on edge probabilities. These aggregate conditions control structural composition without disclosing private per-graph sizes; Appendix A.2 details the sampling and postprocessing. For a candidate G from client k and class c, let $\mathrm{bin}_\ell(\mathbf{s}(G))$ denote its bin in histogram ℓ, and let $\mathcal{R}_{k,c}$ contain candidates already retained from this pool. The structural and diversity scores are

$$\begin{aligned} r_{\text{str}}(G) &= \frac{1}{6}\sum_{\ell=1}^{6} \log\Big(h^{(\ell)}_{k,c,\mathrm{bin}_\ell(\mathbf{s}(G))} + 10^{-8}\Big), \\ r_{\text{div}}(G) &= \min_{G'\in\mathcal{R}_{k,c}} \|\mathbf{s}(G) - \mathbf{s}(G')\|_2. \end{aligned} \tag{9}$$

Greedy selection maximizes $\lambda_s r_{\text{str}} + \lambda_d r_{\text{div}}$, with $\lambda_s, \lambda_d \geq 0$, until a per-pool retention budget, fixed as part of the public configuration, is reached. The selected scores are linearly mapped to $[0,1]$, yielding qualities $\bar{r}_j$. The structural term rewards agreement with released marginals; the distance term discourages repeatedly selecting the same descriptor region. Neither term requires a client teacher or a second communication round.

Reliability-Constrained Distribution Matching. Class-conditional reliability-constrained kernel moment matching (CR-KMM) assigns weights to the calibrated graphs. For each class c, let $\mathcal{K}_c$ be the clients with both a released GDS and a nonempty retained pool; we call c an active class when $\mathcal{K}_c$ is nonempty. Their target masses mix uniform coverage and generation reliability:

$$\beta_{k,c} = \frac{1-\rho}{|\mathcal{K}_c|} + \rho\frac{\widetilde{q}_{k,c}}{\sum_{u\in\mathcal{K}_c}\widetilde{q}_{u,c}}, \quad \overline{\mathbf{m}}^f_c = \sum_{k\in\mathcal{K}_c}\beta_{k,c}\widetilde{\mathbf{m}}^f_{k,c}, \quad \overline{\mathbf{v}}^f_c = \sum_{k\in\mathcal{K}_c}\beta_{k,c}\widetilde{\mathbf{v}}^f_{k,c}, \tag{10}$$

where $\rho \in [0,1]$ and $\widetilde{q}_{k,c} > 0$ is the released reliability from sketch extraction (Section 2.3). Setting $\rho = 0$ gives equal client mass within a class, while increasing ρ shifts mass toward components with larger released reliability.

Index the retained class-c graphs by $j \in \{1, \dots, J_c\}$, write o_j for the source client, and let $J_{k,c} > 0$ count that client's retained graphs. Define $\mathbf{Z}_c \in \mathbb{R}^{D \times J_c}$ by column j equal to $\mathbf{f}(\mathbf{s}(\widehat{G}_j))$, and $\overline{\mathbf{r}}_c = (\bar{r}_1, \dots, \bar{r}_{J_c})^\top$. For nonnegative coefficients $\lambda_2, \lambda_r, \lambda_e$ and $a_{\max} > 0$, CR-KMM solves

$$\begin{aligned} \min_{\mathbf{a}_c \in \mathbb{R}^{J_c}} \quad & \|\mathbf{Z}_c \mathbf{a}_c - \overline{\mathbf{m}}_c^f\|_2^2 + \lambda_2 \|\mathbf{Z}_c^{\odot 2} \mathbf{a}_c - \overline{\mathbf{v}}_c^f\|_2^2 - \lambda_r \overline{\mathbf{r}}_c^\top \mathbf{a}_c + \lambda_e \|\mathbf{a}_c\|_2^2, \\ \text{subject to} \quad & \sum_{j : o_j = k} a_{c,j} = \beta_{k,c} \quad (k \in \mathcal{K}_c), \qquad 0 \le a_{c,j} \le \max\left\{ a_{\max}, \frac{\beta_{o_j,c}}{J_{o_j,c}} \right\}. \end{aligned} \tag{11}$$

The constraints imply $\sum_j a_{c,j} = 1$, and the adaptive cap keeps uniform allocation within every nonempty client group feasible. The first term approximates kernel mean matching in the shared descriptor space, the second additionally matches coordinatewise raw second moments, and the remaining terms reward candidate quality and penalize concentrated weights. Matching these finite statistics does not identify the complete graph distribution. Appendix A.4 details the objective and its feasible set.

2.5 Sketch-Guided Global Learning and Inference

Global Classifier Learning. Let $\mathcal{Y}_{\text{act}} = \{c : \mathcal{K}_c \neq \emptyset\}$ denote the active classes, and let $\widehat{G}_{c,j}$ be the jth retained graph of class c. CR-KMM determines within-class weights $a_{c,j}$; class masses γ_c separately determine how much each class contributes to training. Using released class counts, we set $w_{c,j} = \gamma_c a_{c,j}$ with class masses $\gamma_c \propto \sum_k \widetilde{N}_{k,c}$ normalized over active classes. Absent releases contribute zero to the counts. We restrict the classifier scores $F_{\boldsymbol{\phi}}(G)$ to active-class logits during synthetic training and calibration, retaining the same notation. The server learns the classifier exclusively from the weighted synthetic graphs by minimizing

$$\mathcal{L}_{\text{global}}(\boldsymbol{\phi}) = \sum_{c \in \mathcal{Y}_{\text{act}}} \sum_{j=1}^{J_c} w_{c,j} \, \mathrm{CE}\big(F_{\boldsymbol{\phi}}(\widehat{G}_{c,j}), c\big). \tag{12}$$

The weights sum to one because $\sum_c \gamma_c = 1$ and $\sum_j a_{c,j} = 1$, so Equation (12) is an empirical expectation over the fused synthetic distribution. Its client–class masses $\gamma_c \beta_{k,c}$ encode the chosen coverage and reliability constraints; Appendix A.4 relates this mixture to the population weighting of Equation (1).

Sketch-Guided Predictive Calibration. The released sketches also provide distributional evidence for classifying a new graph. For each released client–class component, the coordinatewise variance estimate from the raw-moment difference $\widetilde{\mathbf{v}}_{k,c} - \widetilde{\mathbf{m}}_{k,c}^{\odot 2}$, optionally pooled across components, is floored at a positive $v_{\min}$ before scoring:

$$\mathbf{V}_{k,c}(v_{\min}) = \mathrm{Diag}\Big(\max\{ \widetilde{\mathbf{v}}_{k,c} - \widetilde{\mathbf{m}}_{k,c}^{\odot 2}, \, v_{\min} \mathbf{1}_d \} \Big) \in \mathbb{R}^{d \times d}, \tag{13}$$

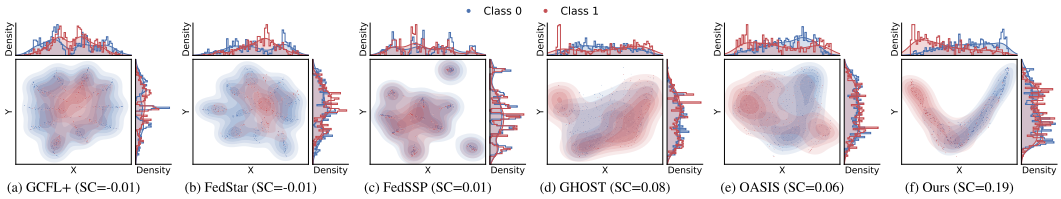
where $\max$ acts coordinatewise and Diag places its vector argument on the diagonal. A fixed scoring operator $\mathcal{H}$ then compares the descriptor $\mathbf{s}(G)$ with the released component summaries $\{\widetilde{\mathbf{m}}_{k,c}, \mathbf{V}_{k,c}(v_{\min}), \widetilde{N}_{k,c}\}_{k,c}$ of the active classes, producing sketch-based class scores $\mathbf{u}(G; v_{\min}) \in \mathbb{R}^{|\mathcal{Y}_{\text{act}}|}$; Appendix A.5 gives a concrete hierarchical construction and variance-pooling options. To combine learned and sketch-based evidence, Std denotes stabilized standardization across active classes (Appendix A.5). The calibrated scores and predicted label are

$$\begin{aligned} \mathbf{z}_{\text{final}}(G) &= \mathrm{Std}\big(F_{\boldsymbol{\phi}}(G)\big) + \lambda_{\text{cal}} \, \mathrm{Std}\big(\mathbf{u}(G; v_{\min})\big), \\ \widehat{y}(G) &= \arg\max_{c \in \mathcal{Y}_{\text{act}}} z_{\text{final},c}(G), \end{aligned} \tag{14}$$

where $\lambda_{\text{cal}} \ge 0$ controls the contribution of the sketch scores and label ties are resolved by a fixed ordering.

Table 1: Main performance (%) on the eight federated graph-classification settings, arranged in two blocks following the source performance table. The best results are shown in **bold**, the second-best results are shaded, and blue ± values denote the reported dispersion. OOM denotes an out-of-memory run.

Datasets	**PTC_MR (3)**		**NCI1 (30)**		**PROTEINS (10)**		**DD (10)**	
Methods	ACC	F1-Score	ACC	F1-Score	ACC	F1-Score	ACC	F1-Score
FedAvg (▷ AISTATS'17)	57.07±0.00	37.70±1.94	55.95±1.09	42.38±1.62	60.26±1.80	49.43±1.05	51.43±2.56	40.38±2.58
FedProx (▷ MLSys'20)	56.06±1.43	35.88±0.64	55.59±1.12	41.74±1.57	57.72±2.24	46.48±2.08	52.31±1.12	41.45±1.93
MOON (▷ CVPR'21)	57.07±0.00	36.33±0.00	56.56±2.43	52.12±4.46	64.39±0.39	46.68±0.61	60.95±0.34	40.07±2.21
FedOpt (▷ ICLR'21)	53.37±5.24	36.04±0.40	55.50±5.34	43.23±13.02	35.47±3.15	26.47±1.10	48.73±9.09	38.24±5.51
GCFL+ (▷ NeurIPS'21)	57.07±0.00	37.70±1.94	55.35±0.39	41.80±1.38	63.14±5.60	55.43±4.89	53.33±1.48	41.01±2.71
FedStar (▷ AAAI'23)	52.44±4.98	45.10±6.14	57.15±0.72	50.50±0.91	62.16±1.45	50.56±0.76	58.01±2.35	39.44±1.60
FedSSP (▷ NeurIPS'24)	53.20±4.84	43.18±4.19	50.83±1.19	36.31±2.18	65.45±0.70	49.37±1.71	OOM	OOM
GHOST (▷ ICML'25)	47.64±6.67	32.13±2.97	51.73±0.00	34.02±0.00	62.30±0.00	38.30±0.00	60.57±0.00	37.67±0.00
OASIS (▷ NeurIPS'25)	47.64±6.67	32.13±2.97	51.73±0.00	34.02±0.00	39.96±3.19	32.56±7.52	39.43±0.00	28.21±0.00
CeFGC (▷ KDD'26)	47.81±2.80	39.15±0.39	59.71±1.66	51.63±4.08	OOM	OOM	OOM	OOM
OSFGC (Ours)	**68.18**±0.32	**64.76**±0.25	**67.37**±0.18	**66.45**±0.15	**76.68**±0.01	**73.36**±0.02	**65.58**±0.23	**56.99**±0.20
Datasets	**CHEM (7)**		**BIO-SN-CV (9)**		**BIO-CHEM (10)**		**BIO-CHEM-SN-CV (16)**	
Methods	ACC	F1-Score	ACC	F1-Score	ACC	F1-Score	ACC	F1-Score
FedAvg (▷ AISTATS'17)	66.79±2.03	44.65±2.66	28.62±0.51	19.32±0.42	55.89±0.91	37.26±1.13	43.23±0.84	27.33±1.55
FedProx (▷ MLSys'20)	66.26±2.83	45.32±5.21	29.45±1.55	20.70±2.34	56.11±0.88	38.26±0.65	43.70±0.35	29.16±0.91
MOON (▷ CVPR'21)	49.87±1.58	35.90±0.28	30.62±2.15	17.75±1.46	52.16±3.94	36.97±3.10	38.28±2.45	24.24±0.55
FedOpt (▷ ICLR'21)	49.64±1.31	32.26±0.62	11.65±12.88	4.92±6.07	32.62±21.89	20.71±14.32	15.55±17.74	8.37±10.48
GCFL+ (▷ NeurIPS'21)	67.67±3.27	45.90±4.45	28.81±1.53	20.19±1.89	56.12±2.19	37.25±3.55	44.20±1.00	29.33±1.80
FedStar (▷ AAAI'23)	71.32±0.60	52.54±0.33	47.68±2.45	42.30±2.16	**66.35**±1.61	52.28±2.21	55.28±0.93	42.92±1.05
FedSSP (▷ NeurIPS'24)	66.17±1.94	50.75±3.26	OOM	OOM	OOM	OOM	OOM	OOM
GHOST (▷ ICML'25)	51.48±0.00	32.79±0.00	16.97±6.94	10.27±5.95	15.71±19.86	9.73±13.09	15.16±13.45	8.80±9.05
OASIS (▷ NeurIPS'25)	49.50±1.40	31.87±0.65	13.99±12.12	6.22±6.17	25.88±17.25	15.40±10.75	23.28±15.72	13.31±9.56
CeFGC (▷ KDD'26)	53.85±1.56	40.58±2.09	OOM	OOM	OOM	OOM	OOM	OOM
OSFGC (Ours)	**74.44**±0.10	**65.24**±0.05	**49.97**±0.35	**42.88**±0.35	66.11±0.11	**59.12**±0.22	**56.24**±2.02	**50.13**±0.77



(a) GCFL+ (SC=-0.01) (b) FedStar (SC=-0.01) (c) FedSSP (SC=0.01) (d) GHOST (SC=0.08) (e) OASIS (SC=0.06) (f) Ours (SC=0.19)

Figure 3: Class-colored t-SNE of graph representations on PROTEINS. Baselines project real training and test graphs; OSFGC additionally projects synthetic training graphs.

3 Experiments

3.1 Experimental Settings

Datasets. We use the same 16 public graph classification datasets as FedStar (Tan et al., 2023), covering four domains: molecular chemistry (MUTAG, BZR, COX2, DHFR, PTC_MR, AIDS, NCI1), bioinformatics (ENZYMES, DD, PROTEINS), social networks (COLLAB, IMDB-BINARY, IMDB-MULTI), and computer vision (Letter-low, Letter-high, Letter-med), all from TU-Dataset (Morris et al., 2020). To simulate federated heterogeneity, we design five non-IID settings: (1) random splits of a single dataset (PTC_MR, NCI1, PROTEINS, DD) over 3, 30, 10, and 10 clients; (2) cross-dataset partition of the seven molecular datasets (CHEM, 7 clients); (3)–(5) cross-domain combinations of datasets from two to four domains (BIO-CHEM, 10 clients; BIO-SN-CV, 9 clients; BIO-CHEM-SN-CV, 16 clients). Each client splits its local data into 80% training, 10% validation, and 10% test.

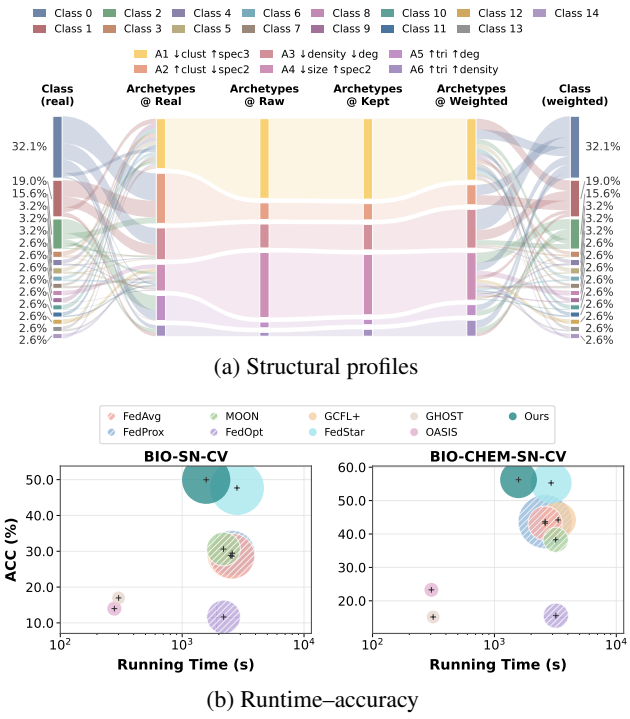
Baselines. We compare against three groups of methods: general federated learning (FedAvg (McMahan et al., 2017), FedProx (Li et al., 2020), MOON (Li et al., 2021), FedOpt (Reddi et al., 2021)), federated graph learning (GCFL+ (Xie et al., 2021), FedStar (Tan et al., 2023), FedSSP (Tan et al., 2024)), and one-shot federated graph learning (GHOST (Qian et al., 2025),

OASIS (Wan et al., 2025), CeFGC (Wang et al., 2026)). All baselines are evaluated in their communication-limited one-shot configurations.

Implementation Details. All methods share a unified training and evaluation protocol, and the complete configuration, including network architectures, diffusion settings, optimization hyperparameters, and hardware, is deferred to Appendix B.

3.2 Main Results

Table 1 shows that OSFGC attains the highest mean ACC in seven of eight settings and the highest Macro-F1 in all eight. Averaged over settings, it outperforms the strongest available baseline by 5.10 percentage points in ACC and 11.85 points in Macro-F1, with the largest margins observed on the four single-dataset settings. These results indicate that the overall advantage is not confined to an isolated benchmark, although its magnitude varies across settings. The consistent lead in Macro-F1 further complements the ACC results by highlighting improvements under a metric that assigns equal importance to each class. Two observations follow. First, the ACC advantage narrows as client distributions become more heterogeneous: BIO-CHEM is the only setting where OSFGC does not lead in ACC, although it still improves Macro-F1 there by a clear margin. This contrast highlights the importance of considering both metrics, since a higher aggregate classification accuracy does not necessarily imply a higher class-averaged F1 score. Second, Macro-F1 gains consistently exceed ACC gains, which is consistent with the intended effect of balancing class-conditional structural contributions across clients.



(a) Structural profiles

(b) Runtime–accuracy

Figure 4: (a) Structural profiles on BIO-SN-CV against six reference archetypes (13,712 real, 504 raw, 252 retained candidates). (b) Runtime–accuracy trade-offs on BIO-SN-CV and BIO-CHEM-SN-CV (bubble: peak GPU memory).

3.3 Representation Analysis

Figure 3 visualizes the final graph representations, with colors indicating class labels. The displayed silhouette coefficient on PROTEINS is 0.19 for OSFGC, compared with -0.01 to 0.08 for the shown baselines; on BIO-SN-CV it is -0.14 versus -0.65 to -0.43, where the negative values indicate remaining class overlap. OSFGC therefore produces the most class-coherent representations among the compared methods on both settings. Because the projected point populations differ across panels, these scores are descriptive rather than a controlled comparison.

3.4 Structural Alignment and Efficiency

Figure 4(a) tracks a fixed six-archetype structural reference, built by clustering real training graphs, through generation, screening, and weighting. Columns are normalized separately, so ribbons are not conserved flows. The distance from the real profile, measured by total variation $\mathrm{TV}(\mathbf{p}, \mathbf{q}) = \frac{1}{2}\sum_{a=1}^{6} |p_a - q_a|$, decreases from 0.348 for raw candidates to 0.328 after screening and 0.226 after weighting. This is consistent with CR-KMM improving structural composition beyond individual plausibility.

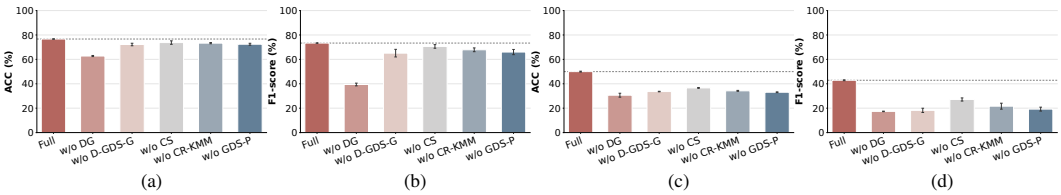


Figure 5: Ablations over three seeds on PROTEINS (a,b) and BIO-SN-CV (c,d). Panels (a,c) report accuracy and panels (b,d) report Macro-F1. Each variant removes one component from the full configuration; bars and error bars show means and standard deviations.

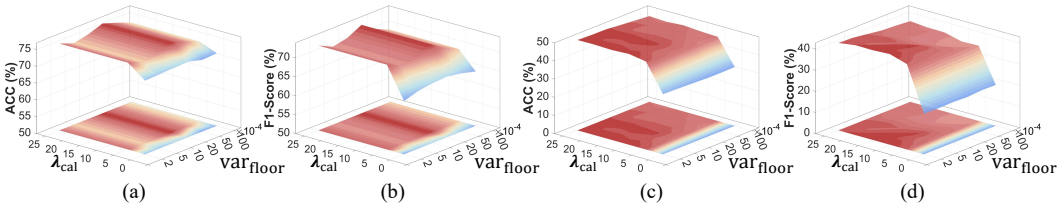


Figure 6: Sensitivity to calibration strength λ_{cal} and variance floor $v_{\min}$ on PROTEINS (a,b) and BIO-SN-CV (c,d), seed 2026; panels (a,c) report accuracy and panels (b,d) report Macro-F1.

Figure 4(b) positions OSFGC above FedStar in accuracy, runtime, and peak GPU memory on both settings, whereas GHOST and OASIS run faster but at substantially lower accuracy. One communication round does not imply a smaller payload, since uploading a generator still incurs a model-size cost.

3.5 Ablation Study and Sensitivity Analysis

Figure 5 assesses component contributions by comparing the complete model (**Full**) with five removals. Without diffusion generator (**w/o DG**) leaves only summary-based sampling of random graphs at the server. Without diffusion–GDS guidance (**w/o D-GDS-G**) disables the density, node-type, and degree histogram guidance during sampling. Without candidate screening (**w/o CS**) retains all generated candidates. Without CR-KMM (**w/o CR-KMM**) uses uniform within-class weights. Without GDS posterior (**w/o GDS-P**) removes the sketch-based correction of the classifier logits. The complete model achieves the highest accuracy and Macro-F1 on both datasets. **w/o DG** produces the largest decrease, indicating that marginal summaries alone do not substitute for learned synthesis; the remaining removals also reduce both metrics, particularly on BIO-SN-CV. These are conditional effects within the tested pipeline, so the advantage of the complete system should not be attributed to the generator alone.

Figure 6 varies calibration strength λ_{cal} and the variance floor $v_{\min}$ of Equation (13) on PROTEINS and BIO-SN-CV. For a fixed floor, accuracy and Macro-F1 change little across positive strengths, whereas the floor itself has a more visible effect. Performance is therefore locally insensitive to the calibration strength within the plotted grid, although this does not establish robustness across all parameter choices or a training-only selection rule.

4 Conclusion

We presented OSFGC, a strict one-shot federated graph classification framework that enables each client to contribute only once while all subsequent learning is completed at the server. OSFGC couples class- and size-conditional graph diffusion with compact Graph Distribution Sketches to transfer, reconstruct, and verify local graph distributions, and further employs structure-aware calibration and reliability-constrained distribution matching to fuse heterogeneous client knowledge for global learning. This design directly addresses the generation, validation, and fusion challenges arising under one-shot communication.

AI USE STATEMENT

Generative AI tools assisted with organizing the supplied method descriptions, drafting and revising the manuscript, and checking consistency between the text and the supplied experimental artifacts.

ETHICS STATEMENT

OSFGC retains raw training graphs at their originating clients, but the uploaded generators and aggregate sketches may still reveal information about the local data. A single upload reduces the number of training interactions; it does not by itself establish differential privacy or resistance to inference attacks. Any privacy guarantee requires a specified threat model, calibrated noise mechanisms, and valid privacy accounting for the complete released package.

REPRODUCIBILITY STATEMENT

We provide the source code needed to reproduce all reported results, and we will release them publicly upon acceptance. All datasets are publicly available benchmarks, and Section 2 together with the appendices specifies the method and experimental protocol in sufficient detail to reimplement the full pipeline independently.

REFERENCES

J. Abramson, J. Adler, J. Dunger, et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630:493–500, 2024.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/958c530554f78bcd8e97125b70e6973d-Abstract.html.

Ewen Callaway. Drug firms' secret data supercharge AI protein models. *Nature*, September 2026. doi: 10.1038/d41586-026-02882-x. URL https://www.nature.com/articles/d41586-026-02882-x. News article, published 14 September 2026.

J. Chen, H. Yan, Z. Liu, et al. When federated learning meets privacy-preserving computation. *ACM Computing Surveys*, 56:1–36, 2024.

Neel Guha, Ameet Talwalkar, and Virginia Smith. One-shot federated learning. In *NeurIPS 2018 Workshop on Machine Learning on the Phone and other Consumer Devices*, 2018. URL https://arxiv.org/abs/1902.11175.

C. He, K. Balasubramanian, E. Ceyani, et al. FedGraphNN: A federated learning system and benchmark for graph neural networks, 2021.

J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems 33*, 2020.

D. Jhunjhunwala, S. Wang, and G. Joshi. FedFisher: Leveraging fisher information for one-shot federated learning. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, 2024.

Tsung-Ting Kuo, Rodney A. Gabriel, Jejo Koola, Robert T. Schooley, and Lucila Ohno-Machado. Distributed cross-learning for equitable federated models—privacy-preserving prediction on data from five california hospitals. *Nature Communications*, 16(1):1371, 2025. doi: 10.1038/s41467-025-56510-9. URL https://doi.org/10.1038/s41467-025-56510-9.

Li, Luo, Cui, Cao, and Xu. OFedED: One-shot federated learning with model ensemble and dataset distillation. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, 2025.

Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. URL https://arxiv.org/abs/2103.16257.

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*, 2020. URL https://arxiv.org/abs/1812.06127.

Liu and Gan. FedDGHD: Diffusion model-based data generation for non-IID challenges in federated learning. In *2025 28th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2025.

Liu et al. FedLPA: One-shot federated learning with layer-wise posterior aggregation. In *Advances in Neural Information Processing Systems 38*, 2024.

X. Liu, Z. Tang, X. Li, Y. Song, S. Ji, Z. Liu, B. Han, L. Jiang, and J. Li. One-shot federated learning methods: A practical guide, 2025.

K. Luo, S. Wang, Y. Fu, R. Shao, X. Li, Y. Lan, M. Gao, and J. Shu. DFDG: Data-free dual-generator adversarial distillation for one-shot federated learning, 2024.

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 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 2017. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy*, 2019.

M. Mendieta, G. Sun, and C. Chen. Navigating heterogeneity and privacy in one-shot federated learning with diffusion models, 2024.

Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. TUDataset: A collection of benchmark datasets for learning with graphs. In *ICML Workshop on Graph Representation Learning and Beyond*, 2020. URL https://arxiv.org/abs/2007.08663.

Jiaru Qian, Guancheng Wan, Wenke Huang, Guibin Zhang, Yuxin Wu, Bo Du, and Mang Ye. GHOST: Generalizable one-shot federated graph learning with proxy-based topology knowledge retention. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 50047–50065. PMLR, 2025. URL https://proceedings.mlr.press/v267/qian25a.html.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H. Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2003.00295.

T. Sang, Z. Chu, J. Xuan, X. Zhang, and X. Li. Personalized federated learning in one-shot: A method for heterogeneous data scenarios. *IEEE Internet of Things Journal*, 12:40415–40425, 2025.

Sergey Sosnin. Chemical space visual navigation in the era of deep learning and big data. *Drug Discovery Today*, 2025. DOI: 10.1016/j.drudis.2025.104392.

Amit Kumar Srivastav, Unnati Modi, Rahul Kumar, Dhiraj Bhatia, and Raghu Solanki. Geometric deep learning-based drug design models for small-molecule drug discovery. *Molecular Informatics*, 2026. DOI: 10.1002/minf.70047.

Yue Tan, Yixin Liu, Guodong Long, Jing Jiang, Qinghua Lu, and Chengqi Zhang. Federated learning on non-IID graphs via structural knowledge sharing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 9953–9961, 2023. doi: 10.1609/aaai.v37i8.26187. URL https://ojs.aaai.org/index.php/AAAI/article/view/26187.

Zihan Tan, Guancheng Wan, Wenke Huang, and Mang Ye. FedSSP: Federated graph learning with spectral knowledge and personalized preference. In *Advances in Neural Information Processing Systems*, volume 37, pp. 34561–34581, 2024. doi: 10.52202/079017-1090. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/3d226fb8fbd6ee6ec70d0427f1319707-Abstract-Conference.html.

Team CD, J. Boitreaud, J. Dent, et al. Zero-shot antibody design in a 24-well plate, 2025. bioRxiv preprint, DOI: 10.1101/2025.07.05.663018.

Clément Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. DiGress: Discrete denoising diffusion for graph generation. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2209.14734.

Frank Wan, Jiaru Qian, Wenke Huang, Qilin Xu, Xianda Guo, Boheng Li, Guibin Zhang, Bo Du, and Mang Ye. OASIS: One-shot federated graph learning via Wasserstein assisted knowledge integration. In *Advances in Neural Information Processing Systems*, volume 38, pp. 80038–80068, 2025. doi: 10.52202/085713-2673. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/7300dd4d625cb42a438c1b822c77569f-Abstract-Conference.html.

Xiuling Wang, Xin Huang, Haibo Hu, and Jianliang Xu. Communication-efficient federated graph classification via generative diffusion modeling. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2026. URL https://arxiv.org/abs/2601.15722.

Yanan Wang, Yu Guang Wang, Changyuan Hu, Ming Li, Yanan Fan, Nina Otter, Ikuan Sam, Hongquan Gou, Yiqun Hu, Terry Kwok, John Zalcberg, Alex Boussioutas, Roger J. Daly, Guido Montúfar, Pietro Liò, Dakang Xu, Geoffrey I. Webb, and Jiangning Song. Cell graph neural networks enable the precise prediction of patient survival in gastric cancer. *npj Precision Oncology*, 6:45, 2022. doi: 10.1038/s41698-022-00285-5. URL https://www.nature.com/articles/s41698-022-00285-5.

F. Wong, E. J. Zheng, J. A. Valeri, et al. Discovery of a structural class of antibiotics with explainable deep learning. *Nature*, 626:177–185, 2024.

C. Wu, F. Wu, Y. Cao, Y. Huang, and X. Xie. FedGNN: Federated graph neural network for privacy-preserving recommendation, 2021.

Han Xie, Jing Ma, Li Xiong, and Carl Yang. Federated graph classification over non-IID graphs. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18839–18852, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/9c6947bd95ae487c81d4e19d3ed8cd6f-Abstract.html.

D. Yao, Y. Shi, T. Liu, and Z. Xu. FedMHO: Heterogeneous one-shot federated learning towards resource-constrained edge devices, 2025.

H. Zeng, M. Xu, T. Zhou, X. Wu, J. Kang, Z. Cai, and D. Niyato. One-shot-but-not-degraded federated learning. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024.

Zhang and Guo. FedGDC-P: Communication-efficient personalized graph federated learning based on dataset condensation. In *2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2024.

Zhang, Liu, and Wang. One-shot federated learning via synthetic distiller-distillate communication. In *Advances in Neural Information Processing Systems 37*, 2024.

Jie Zhang, Chen Chen, Bo Li, Lingjuan Lyu, Shuang Wu, Shouhong Ding, Chunhua Shen, and Chao Wu. DENSE: Data-free one-shot federated learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 21414–21428, 2022. doi: 10.52202/068431-1556. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/868f2266086530b2c71006ea1908b14a-Abstract-Conference.html.

Liang Zhang, Juan Zhang, Rui Huang, Yiwen Wang, Linjing Liu, Yanyong Zhang, Kong Chen, Jun Jiang, and Yuen Wu. A federated graph learning method to realize multi-party collaboration for molecular discovery. *Nature Machine Intelligence*, 8(2):246–256, 2026. doi: 10.1038/s42256-026-01184-1. URL https://doi.org/10.1038/s42256-026-01184-1.

L. Zhu, Z. Liu, and S. Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems 32*, 2019.

A Additional Method Details

A.1 Single-upload communication protocol

OSFGC addresses this problem by transferring local graph generators and class-level distribution summaries in a single upload. After receiving all client packages, the server generates candidate graphs, calibrates their structural composition, assigns training weights, and fits a global classifier. Public configuration fixes the label and feature schemas, graph-size bound, and sketch coordinates before training. The single upload of client k is the knowledge package

$$\mathcal{U}_k = \{\boldsymbol{\theta}_k,\ \mathcal{S}_k,\ \widetilde{\boldsymbol{\pi}}_k,\ \widetilde{\mathbf{q}}_k,\ \mathcal{M}_k^{\mathrm{priv}},\ \mathcal{M}_k^{\mathrm{train}}\}, \tag{15}$$

where $\boldsymbol{\theta}_k$ parameterizes the class- and size-conditional generator $p_{\boldsymbol{\theta}_k}(G \mid y, n)$, $\mathcal{S}_k$ collects the class-level sketches defined in Section 2.3, $\widetilde{\boldsymbol{\pi}}_k \in \mathbb{R}^C_{\geq 0}$ denotes the released class counts, $\widetilde{\mathbf{q}}_k \in \mathbb{R}^C$ collects the class-conditional reliability scores, and $\mathcal{M}_k^{\mathrm{priv}}, \mathcal{M}_k^{\mathrm{train}}$ record optional privacy-accounting and training metadata. The package contains neither raw graphs nor client classifiers, and it excludes per-graph node-count lists; the server infers structural conditions from the released histograms. The generator parameters are retained separately at the server and are never averaged across clients. Under this protocol, the training communication of the complete pipeline consists of exactly one client-to-server upload per client and no download, that is,

$$R_{\mathrm{comm}} = 1, \qquad N_{\mathrm{upload}} = 1, \qquad N_{\mathrm{download}} = 0. \tag{16}$$

Delivering the final global classifier to clients, if desired in deployment, is a post-training action and lies outside this accounting. All server-side training stages use only the uploaded packages and synthetic graphs, without accessing private training or validation graphs.

A.2 Diffusion training and sampling

Graph preprocessing and reconstruction loss. The generator represents simple undirected graphs with at least two and at most $n_{\max}$ nodes. Training graphs exceeding this public bound are reduced to randomly sampled induced subgraphs; continuous attributes are clipped to a fixed public range. The denoiser uses one parameter vector for all local classes, with conditioning embeddings for class, diffusion time, and normalized graph size. Let $\widehat{\mathbf{X}}_0 \in \mathbb{R}^{n\times m}$ be its continuous clean-state estimate, $\mathbf{L}_X \in \mathbb{R}^{n\times M}$ its categorical-node logits when there are M node types, and $\mathbf{L}_A \in \mathbb{R}^{n\times n}$ its clean-edge logits. With node types $c_v \in \{1,\dots,M\}$ and the scalar logistic function $\mathrm{sigm}(x) = (1+\exp(-x))^{-1}$, the node loss is

$$\mathcal{L}_{\mathrm{node}} = \begin{cases} \dfrac{1}{nm}\|\widehat{\mathbf{X}}_0 - \mathbf{X}_0\|_F^2, & \text{continuous attributes,} \\ -\dfrac{1}{n}\displaystyle\sum_{v=1}^{n} \log \mathrm{softmax}(\mathbf{L}_{X,v,:})_{c_v}, & \text{categorical attributes.} \end{cases} \tag{17}$$

The Frobenius norm in Equation (17) averages reconstruction errors across nodes and coordinates. For $N_+ = \sum_{u<v} \mathbf{A}_{0,uv}$, $N_- = \binom{n}{2} - N_+$, and $w_+ = \min\{10, N_-/\max(N_+,1)\}$, the per-graph reconstruction objective is

$$\begin{aligned} \mathcal{L}_{\mathrm{diff}} &= \mathcal{L}_{\mathrm{node}} + \mathcal{L}_{\mathrm{edge}}, \\ \mathcal{L}_{\mathrm{edge}} &= -\frac{1}{\binom{n}{2}} \sum_{u<v} \left[w_+ \mathbf{A}_{0,uv} \log \mathrm{sigm}(\mathbf{L}_{A,uv}) + (1-\mathbf{A}_{0,uv}) \log(1-\mathrm{sigm}(\mathbf{L}_{A,uv})) \right]. \end{aligned} \tag{18}$$

Only the upper triangle contributes to the loss; generated edges are mirrored and diagonal entries are zero. Positive weighting compensates for edge sparsity within a graph, with a cap to limit amplification. For a complete graph, this weighting gives $w_+ = 0$ and hence no positive-edge training signal; a nondegenerate treatment of this boundary case requires an implementation decision.

Categorical posterior transitions. For a variable with M_{cat} categories, define the one-step transition $\mathbf{Q}_t^{(M_{\text{cat}})} = (1-\beta_t)\mathbf{I}_{M_{\text{cat}}} + (\beta_t/M_{\text{cat}})\mathbf{1}_{M_{\text{cat}}}\mathbf{1}_{M_{\text{cat}}}^\top$ in Equation (2), and let $\overline{\mathbf{Q}}_0^{(M_{\text{cat}})} = \mathbf{I}_{M_{\text{cat}}}$. Matrix multiplication gives the cumulative transition $\overline{\mathbf{Q}}_t^{(M_{\text{cat}})} = \prod_{s=1}^{t} \mathbf{Q}_s^{(M_{\text{cat}})}$. For clean category $z_0 = i$, noisy observation $z_t = j$, and predecessor $z_{t-1} = \ell$, Bayes' rule gives, for $t \geq 2$,

$$q(z_{t-1} = \ell \mid z_t = j, z_0 = i) = \frac{\overline{\mathbf{Q}}_{t-1,i\ell}^{(M_{\text{cat}})} \mathbf{Q}_{t,\ell j}^{(M_{\text{cat}})}}{\overline{\mathbf{Q}}_{t,ij}^{(M_{\text{cat}})}}. \tag{19}$$

All denominators are positive under the stated noise rates. Let $\widehat{\mathbf{p}}_0 \in [0,1]^{M_{\text{cat}}}$, with $\mathbf{1}_{M_{\text{cat}}}^\top \widehat{\mathbf{p}}_0 = 1$, be the denoiser's clean-state categorical probabilities, conditioned on the entire noisy graph and the public conditioning variables. A clean-state-prediction parameterization marginalizes the posterior over these probabilities:

$$p_{\boldsymbol{\theta}_k}(z_{t-1} = \ell \mid G_t, y, n) = \sum_{i=1}^{M_{\text{cat}}} \widehat{p}_{0,i} \, \frac{\overline{\mathbf{Q}}_{t-1,i\ell}^{(M_{\text{cat}})} \mathbf{Q}_{t,\ell z_t}^{(M_{\text{cat}})}}{\overline{\mathbf{Q}}_{t,iz_t}^{(M_{\text{cat}})}}. \tag{20}$$

Equation (20) is normalized because each clean-category posterior is normalized. Here $\widehat{\mathbf{p}}_0$ estimates the clean-state conditional given the full noisy graph and the generation conditions. Equation (20) defines a normalized model transition; it equals the exact per-variable reverse conditional only when $\widehat{\mathbf{p}}_0 = q(z_0 \mid G_t, y, n)$. For binary edges, its category indices use $z_t = \mathbf{A}_{t,uv} + 1$, so indices one and two denote absent and present edges, respectively. At $t = 1$, sampling uses the predicted clean-state probabilities directly. The binary adjacency process is the $M_{\text{cat}} = 2$ instance: its cumulative edge probability is $q(\mathbf{A}_{t,uv} = 1 \mid \mathbf{A}_{0,uv}) = \bar{\zeta}_t \mathbf{A}_{0,uv} + (1-\bar{\zeta}_t)/2$. Consequently, independent Bernoulli$(1/2)$ edges and uniform node categories form the terminal reference distributions when $\bar{\zeta}_T$ is close to zero. They are exact terminal marginals only at zero retention.

Continuous posterior transitions and generation conditions. For continuous attributes, write $\zeta_t = 1 - \beta_t$ and set $\bar{\zeta}_0 = 1$. Conditional on a clean matrix $\mathbf{X}_0$, the Gaussian forward chain has posterior mean $\mathbf{M}_t$ and scalar variance $\widetilde{\beta}_t$:

$$\mathbf{M}_t(\mathbf{X}_t, \mathbf{X}_0) = \frac{\sqrt{\bar{\zeta}_{t-1}}\beta_t}{1-\bar{\zeta}_t}\mathbf{X}_0 + \frac{\sqrt{\zeta_t}(1-\bar{\zeta}_{t-1})}{1-\bar{\zeta}_t}\mathbf{X}_t, \qquad \widetilde{\beta}_t = \frac{1-\bar{\zeta}_{t-1}}{1-\bar{\zeta}_t}\beta_t. \tag{21}$$

The clean-state estimate replaces $\mathbf{X}_0$ in the reverse mean, and independent Gaussian noise with this variance yields the corresponding reverse update. Continuous attributes initialize from a standard Gaussian. The diffusion description specifies a cosine schedule with small terminal signal-to-noise ratio for the reported model-upload configuration. For a size histogram with $B \geq 1$ bins and sampled bin $b \in \{0, \ldots, B-1\}$, the server reconstructs the graph-size condition as

$$n = \text{clip}_{[2, n_{\max}]}\left(\text{round}\left[n_{\max}\frac{b + 1/2}{B}\right]\right). \tag{22}$$

Thus size sampling uses aggregated bins and does not require private per-graph sizes. Optional stratification maps low-discrepancy quantiles through the released size and density cumulative distributions. At generation time, node-type histograms adjust categorical probabilities, density determines the final edge count, and degree histograms supply node-level edge-probability biases. If postprocessing produces no edge, the pair with the largest predicted edge logit is connected. The last rule restricts synthetic support to nonempty edge sets even when a released density bin includes zero.

A.3 Sketch construction, perturbation, and interpretation

Descriptor coordinates and kernel approximation. The method description lists normalized node count, edge density, degree mean, degree standard deviation, maximum degree, average local clustering, number of connected components divided by node count, triangle density, the first three

normalized-Laplacian spectral moments, and means and standard deviations of selected attribute coordinates. Each statistic is normalized by public ranges and clipped to $[0, 1]$. Shared feature schemas, histogram bins, random seeds, and bandwidth schedules are necessary for cross-client comparisons. The moments in Equation (8) average feature vectors of entire graphs; they do not identify corresponding nodes across graphs or clients. For deterministic descriptors $\mathbf{s}, \mathbf{s}' \in [0, 1]^d$, independence of phases and frequencies in Equation (7) gives

$$\mathbb{E}_{\mathbf{W},\mathbf{b}}\big[\mathbf{f}(\mathbf{s})^\top \mathbf{f}(\mathbf{s}')\big] = \frac{1}{D}\sum_{r=1}^{D} \exp\left(-\frac{\|\mathbf{s}-\mathbf{s}'\|_2^2}{2\sigma_r^2}\right). \tag{23}$$

The induced kernel is a finite mixture of radial basis kernels. Repeated bandwidths receive weight equal to their fraction of the D coordinates, rather than necessarily equal mixture weights. The shared random draw makes all released moments comparable in the same finite-dimensional feature space. The second-order sketch contains only coordinatewise squares of these features, not their full outer-product matrix.

Optional release noise. For any histogram count vector $\mathbf{n} \in \mathbb{R}^B_{\geq 0}$, let $\mathbf{e}$ be an optional Gaussian perturbation of the same dimension and $[\mathbf{x}]_+$ denote coordinatewise nonnegative clipping. Provided its denominator is positive, the released histogram is

$$\widetilde{\mathbf{h}} = \frac{[\mathbf{n}+\mathbf{e}]_+}{\mathbf{1}_B^\top [\mathbf{n}+\mathbf{e}]_+}. \tag{24}$$

If every perturbed count is nonpositive, Equation (24) is undefined; an implementation must specify a fallback distribution or suppress that release. With $b_f = \sqrt{2/D}$, each random feature lies in $[-b_f, b_f]$, and its square lies in $[0, b_f^2]$. For optional noise vectors $\mathbf{e}_m, \mathbf{e}_v \in \mathbb{R}^D$, the source describes coordinatewise projection:

$$\widetilde{\mathbf{m}}^f_{k,c} = \mathrm{clip}_{[-b_f,b_f]}(\widehat{\mathbf{m}}^f_{k,c} + \mathbf{e}_m), \qquad \widetilde{\mathbf{v}}^f_{k,c} = \mathrm{clip}_{[0,b_f^2]}(\widehat{\mathbf{v}}^f_{k,c} + \mathbf{e}_v). \tag{25}$$

Clipping restores marginal coordinate bounds but need not restore joint realizability: independently perturbed moments may violate $\widetilde{v}^f_{k,c,r} \geq (\widetilde{m}^f_{k,c,r})^2$. CR-KMM therefore treats them as numerical targets, with a potentially nonzero irreducible residual. A released class count is clipped to at least one for an eligible class. Class eligibility itself is determined from the local count and minimum release threshold; a class below this threshold contributes no summary.

Reliability and privacy scope. The released reliability $\widetilde{q}_{k,c} = \mathrm{clip}_{[q_{\min},q_{\max}]}\big((\overline{\mathcal{L}}^{\mathrm{diff}}_{k,c} + 10^{-6})^{-1} + \eta_{k,c}\big)$ is the clipped inverse of the average class-c diffusion loss recorded during local training, with optional scalar Gaussian perturbation. This is a training-loss heuristic rather than an estimate of held-out graph generation fidelity. In particular, heterogeneity can change the scale of a client's reconstruction loss, so the uniform default $\rho = 0$ does not rely on comparisons between clients' loss scales. Optional local gradient clipping and Gaussian perturbation provide additional experimental controls, but no end-to-end differential-privacy guarantee follows from noisy summaries alone. The uploaded generator must also be included in any privacy analysis, together with class eligibility, counts, histograms, descriptor moments, reliability, and their composition. The zero-noise configuration has no differential-privacy guarantee.

A.4 CR-KMM FEASIBILITY, OPTIMIZATION, AND TARGET DISTRIBUTION

Coverage constraints and optimization. All client groups in $\mathcal{K}_c$ are nonempty. Define $u_{k,c} = \max\{a_{\max}, \beta_{k,c}/J_{k,c}\}$ and the group feasible set $\mathcal{A}_{k,c} = \{\mathbf{x} \in \mathbb{R}^{J_{k,c}} : 0 \leq x_j \leq u_{k,c},\ \sum_j x_j = \beta_{k,c}\}$. The vector with all entries $\beta_{k,c}/J_{k,c}$ belongs to this set, establishing that the product of group constraints is nonempty. An empty group cannot satisfy a positive target mass and is excluded before computing Equation (10). Therefore coverage means coverage of eligible components with retained candidates. It does not guarantee recovery of unpublished classes, suppressed clients, or modes absent from the finite generated pool. Since the class masses sum to one, no extra division by J_c appears in the weighted moment $\mathbf{Z}_c\mathbf{a}_c$.

Let $\mathcal{J}_c(\mathbf{a}_c)$ denote the objective in Equation (11). Its gradient and projected update, with step size $\eta^{(r)} > 0$ and iteration $r \geq 0$, are

$$\begin{aligned}\nabla \mathcal{J}_c(\mathbf{a}_c) &= 2\mathbf{Z}_c^\top(\mathbf{Z}_c\mathbf{a}_c - \overline{\mathbf{m}}_c^f) + 2\lambda_2(\mathbf{Z}_c^{\odot 2})^\top(\mathbf{Z}_c^{\odot 2}\mathbf{a}_c - \overline{\mathbf{v}}_c^f) - \lambda_r\overline{\mathbf{r}}_c + 2\lambda_e\mathbf{a}_c, \\ \mathbf{a}_c^{(r+1)} &= \mathrm{Proj}_{\prod_{k\in\mathcal{K}_c}\mathcal{A}_{k,c}}\left(\mathbf{a}_c^{(r)} - \eta^{(r)}\nabla\mathcal{J}_c(\mathbf{a}_c^{(r)})\right).\end{aligned} \tag{26}$$

For any group input $\mathbf{v} \in \mathbb{R}^{J_{k,c}}$, Euclidean projection has coordinates $x_j = \mathrm{clip}_{[0,u_{k,c}]}(v_j - \tau)$, where the scalar τ is chosen so that $\sum_j x_j = \beta_{k,c}$. This is a monotone one-dimensional constraint. The objective is convex for the stated nonnegative coefficients and strictly convex when $\lambda_e > 0$; strict convexity gives a unique minimizing weight vector. The quality reward has a minus sign because a higher quality should reduce the minimized objective. The $\lambda_e\|\mathbf{a}_c\|_2^2$ term penalizes concentration. On the simplex, minimizing this term maximizes order-two Rényi entropy $-\log\sum_j a_{c,j}^2$; Equation (11) does not contain a Shannon-entropy regularizer.

Conditional mixtures versus population weighting. To make the target interpretation explicit, let $P_k(\cdot \mid c)$ be the graph distribution conditional on class c, and let δ_c be the unit point mass at label c. Ignoring estimation and release noise, the intended target and the weighted synthetic empirical distribution are

$$P_\star = \sum_{c\in\mathcal{Y}_{\mathrm{act}}}\gamma_c\sum_{k\in\mathcal{K}_c}\beta_{k,c}\big(P_k(\cdot \mid c)\otimes\delta_c\big), \qquad \widehat{P}_{\mathrm{syn}} = \sum_{c\in\mathcal{Y}_{\mathrm{act}}}\gamma_c\sum_{j=1}^{J_c}a_{c,j}\delta_{(\widehat{G}_j,c)}. \tag{27}$$

Here $\otimes$ attaches the fixed class label to a conditional graph distribution, and each class-local graph index is interpreted within its own pool. For the sample-size weights $\alpha_k = N_k/\sum_{u=1}^K N_u$ in Equation (1), define the population mixture $P_{\mathrm{pop}} = \sum_k \alpha_k P_k$ and set $\pi_{k,c} = P_k(y = c)$. The client–class components of $P_\star$ match this mixture when $\gamma_c\beta_{k,c} = \alpha_k\pi_{k,c}$ for every supported pair, with all positive-mass pairs retained. Uniform within-class client mass and count-based class mass do not generally satisfy this condition. They instead define an explicit alternative weighting of heterogeneous components. This distinction is relevant when comparing pooled accuracy with macro-averaged or worst-client performance; component coverage alone does not imply uniform client influence after summing over classes.

A.5 Posterior calibration and component ablations

Descriptor posterior scores. Hierarchical calibration uses raw descriptor moments, distinct from the random-feature moments matched by CR-KMM. Before optional release noise, the coordinate-wise moments are $\widehat{\mathbf{m}}_{k,c} = N_{k,c}^{-1}\sum_{i:y_i^k=c}\mathbf{s}(G_i^k)$ and $\widehat{\mathbf{v}}_{k,c} = N_{k,c}^{-1}\sum_{i:y_i^k=c}\mathbf{s}(G_i^k)^{\odot 2}$. For positive variance floor $v_{\min}$ (the experimental parameter `var_floor`), the class-specific diagonal variance can be constructed as

$$v_{k,c,h}^{\mathrm{diag}} = \max\{v_{\min}, \widetilde{v}_{k,c,h} - \widetilde{m}_{k,c,h}^2\}, \qquad h \in \{1,\ldots,d\}. \tag{28}$$

With released-count-based priors $p(k)$ and $p(c \mid k)$, the hierarchical instance of $\mathcal{H}$ is

$$u_c(G) = \log\sum_k \exp\left\{-\frac{1}{2}\sum_{h=1}^d\left[\frac{(s_h(G) - \widetilde{m}_{k,c,h})^2}{v_{k,c,h}^{\mathrm{diag}}} + \log v_{k,c,h}^{\mathrm{diag}}\right] + \log p(k) + \log p(c \mid k)\right\}. \tag{29}$$

Only released client–class components enter the sum; $\widetilde{\mathbf{m}}_{k,c}$ and $\widetilde{\mathbf{v}}_{k,c}$ are released raw descriptor moments. This floor also prevents negative estimated variances following independent release perturbations. The implementation description supports sharing diagonal variances across all components, sharing them within each client, or retaining separate client–class variances. Equation (29) then marginalizes Gaussian-shaped descriptor scores over clients. The omitted $d\log(2\pi)$ is common to all classes and does not affect the prediction. The diagonal Gaussian is a scoring model for bounded, dependent graph statistics; it is not a claim that their true conditional distribution is Gaussian.

For a finite score vector $\mathbf{z} \in \mathbb{R}^{C_{\mathrm{act}}}$, where $C_{\mathrm{act}} = |\mathcal{Y}_{\mathrm{act}}| \geq 2$, define its coordinate mean $\overline{z} = C_{\mathrm{act}}^{-1}\sum_c z_c$ and a stabilized classwise standardization with $\epsilon_z > 0$:

$$\mathrm{Std}(\mathbf{z})_c = \frac{z_c - \overline{z}}{\sqrt{C_{\mathrm{act}}^{-1}\sum_d(z_d - \overline{z})^2 + \epsilon_z}}, \qquad \mathbf{z}_{\mathrm{final}}(G) = \mathrm{Std}(F_{\boldsymbol{\phi}}(G)) + \lambda_{\mathrm{cal}}\,\mathrm{Std}(\mathbf{u}(G)). \tag{30}$$

Final class probabilities use the softmax of $\mathbf{z}_{\text{final}}(G)$. Within-graph standardization puts the two score vectors on comparable scales, and λ_{cal} controls their relative contribution. A positive floor stabilizes calibration when few local examples produce small or perturbed variance estimates. At inference time, evaluating descriptors of the graph being classified is permissible; no training on private test graphs or access to their client identity is required. Every evaluated class must have an eligible uploaded component to obtain a finite hierarchical score. An absent class cannot be reconstructed through calibration.

Ablation scope and configuration correspondence. The six-variant study separates the pipeline into five interventions around the full method A0. A1 replaces diffusion with generation from GDS histograms. A2 retains diffusion while removing its GDS structural guidance. A3 removes structural-quality and diversity-based candidate selection, deactivating the quality reward in CR-KMM. A4 removes CR-KMM weighting and uses equal synthetic-example weights. A5 sets the GDS posterior-calibration weight to zero. These interventions test distinct stages: A2 changes candidate construction, A3 changes which candidates remain available, A4 changes their training contribution, and A5 changes the final prediction scores. For A4, equal example weights need not preserve either the class masses or the client-group masses in Equation (11); the experimental definition must determine whether any separate class balancing remains active. The backbone is therefore denoted generically by F_ϕ in the method, with its instantiated architecture specified in the experimental configuration.

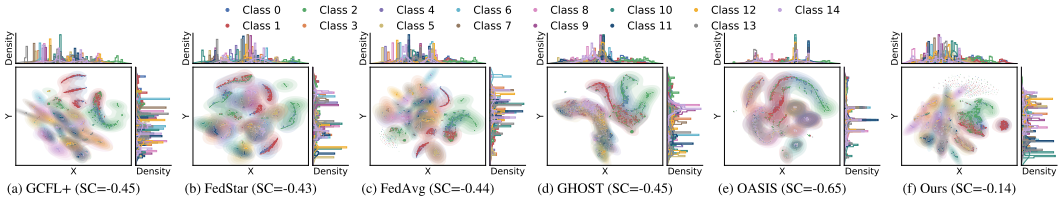
B Experimental Protocol and Complete Results

B.1 Detailed Experimental Setup

Implementation Details. All eight benchmarks share one configuration. Following the baselines' partition protocol, each client's local graphs are split into stratified training, validation, and test sets with a ratio of 8:1:1. Each client trains a three-layer conditional graph denoiser with hidden width 64, T=20 diffusion steps and a cosine noise schedule, using AdamW with learning rate 10^{-3} and weight decay 10^{-5}, and uploads it once together with a bounded graph distribution sketch of 256-dimensional random features and 16-bin histograms. The server keeps 4 of 8 oversampled candidate graphs for each class of every client, fuses them with CR-KMM for 300 iterations, and trains an MLP classifier with hidden width 128 and dropout 0.2 on the weighted synthetic pool for 200 epochs with batch size 128, using AdamW with learning rate 10^{-3} and weight decay 10^{-4}. Graphs are clipped to at most 64 nodes. We run all experiments in PyTorch on a single NVIDIA H20 GPU and report the mean and standard deviation over seeds 2026–2028.

Client data partition. We follow the FedStar data protocol exactly. In the single-dataset setting, the full dataset is divided into disjoint client shards with the official GCFL/FedStar random chunking under split seed 123, and the client number follows GCFL's OneDS scale so that each client holds roughly one hundred training graphs, giving 3 clients for PTC_MR, 30 for NCI1, and 10 for PROTEINS and DD. In the cross-dataset setting, each source dataset of a same-domain group becomes exactly one client, giving 7 clients for CHEM. The cross-domain setting applies the same one-dataset-one-client rule to mixed-domain groups, giving 9 clients for BIO_SN_CV, 10 for BIO_CHEM, and 16 for BIO_CHEM_SN_CV. Within every client, local graphs are then split into stratified training, validation, and test sets with a ratio of 8:1:1. Node attributes keep the dataset-native encoding and are zero-padded to the group-wide maximum dimension, while class indices are kept as-is in one shared label space whose size equals the largest class count in the group, so a source dataset simply occupies its own label subset and the global classifier uses a single shared label space rather than per-task heads. ACC and Macro-F1 are computed on each client's own test graphs and then averaged across clients with equal weight.

Strict one-shot compliance. OSFGC contacts each client exactly once. The client trains its conditional generator locally, builds the bounded sketch, uploads one immutable package, and leaves the protocol; there is no download round, no feedback channel, and no second contact of any kind. The package contains only generator parameters, class-conditional sketch statistics, class priors, and reliability metadata, so every server-side decision, including candidate screening, CR-KMM weighting, posterior calibration, and classifier training, is computed exclusively from the packages and from



(a) GCFL+ (SC=-0.45) (b) FedStar (SC=-0.43) (c) FedAvg (SC=-0.44) (d) GHOST (SC=-0.45) (e) OASIS (SC=-0.65) (f) Ours (SC=-0.14)

Figure 7: Class-colored t-SNE of graph representations on the combined BIO-SN-CV setting. Baselines project real training and test graphs; OSFGC additionally projects synthetic training graphs. Colors indicate class labels.

synthetic graphs sampled by the server itself, and client training or validation graphs never reach the server in any form. Model selection requires no validation signal at all: the server classifier is trained for a fixed number of epochs and the final checkpoint is reported, which removes any dependence on early stopping. Client validation and test partitions are read exactly once by the offline evaluator after training has finished, so no test information can influence any server decision.

One-shot adaptation of baselines. All baselines share the partition protocol above and differ only in their communication pattern. Multi-round methods, namely FedAvg, FedProx, MOON, FedOpt, GCFL+, FedStar, and FedSSP, are restricted to a single communication round: clients train the same three-layer GIN with hidden width 64 using batch size 128, learning rate 10^{-3}, upload once, and the aggregated global model is evaluated as the one-shot result. CeFGC is included through its official generative one-shot baseline, where each client uploads one class-conditional EDP-GNN generator and the server trains a single global GIN on the sampled synthetic graphs without any server-to-client transfer. GHOST and OASIS are natively one-shot but were released with node-classification implementations only, so we keep their official pipelines unchanged, including GHOST's proxy generator with FGW alignment and topology retention and OASIS's synthesizer, topological codebook, and Wasserstein distillation, and replace only the node-level prediction head with graph pooling and graph-level supervision. As simple controls, our ablation further includes a GDS-only variant that replaces the diffusion generator with direct sampling from the released sketch statistics, and a direct-mixture variant that trains on the uniformly weighted union of synthetic graphs.

B.2 Cross-Domain Representation Visualization

Figure 7 complements Figure 3 with the combined BIO-SN-CV setting, in which bioinformatics and social-network graphs are pooled and cross-domain heterogeneity is stronger. Each panel class-colors the final graph representations of one method: the baselines project real graphs, whereas OSFGC additionally projects its synthetic training graphs. The panels reproduce the silhouette ordering reported in the main text, with OSFGC attaining the highest class coherence (-0.14) among the compared methods (-0.65 to -0.43); as before, the projected point populations differ across panels, so these scores remain descriptive rather than a controlled comparison.

C Related Work

Federated graph learning. Federated graph learning adapts federated optimization to graph-structured data and splits along the granularity of the learning task. Intra-graph settings partition a single large graph across clients and target node-level objectives, as in federated recommendation and node-classification benchmarks (He et al., 2021; Wu et al., 2021). Inter-graph settings give each client a collection of independent graphs and target graph-level classification, where structural heterogeneity across clients is the central difficulty. GCFL+ clusters clients by gradient sequences to localize collaboration (Xie et al., 2021); FedStar shares a structural encoder while keeping feature encoders local (Tan et al., 2023); FedSSP distills shared spectral knowledge into personalized preferences (Tan et al., 2024); and CeFGC trains local diffusion generators to synthesize graphs across silos (Wang et al., 2026). These methods establish complementary ways of coping with structural heterogeneity, but their original protocols all rely on repeated client participation over multiple communication rounds.

One-shot federated learning. One-shot federated learning removes this repeated interaction on Euclidean data. Representative strategies include model-space fusion through posterior or Fisher-based aggregation (Liu et al., 2024; Jhunjhunwala et al., 2024; Yao et al., 2025), knowledge distillation from uploaded models (Luo et al., 2024; Sang et al., 2025), dataset condensation or distillation into compact synthetic sets (Zhang & Guo, 2024; Zhang et al., 2024; Li et al., 2025), and generative proxy data synthesized at the server (Zhang et al., 2022; Mendieta et al., 2024; Liu & Gan, 2025). For graph data, GHOST retains topology-sensitive knowledge from client proxy models (Qian et al., 2025), and OASIS synthesizes graphs with a topological codebook and Wasserstein-based distillation (Wan et al., 2025); both formulate node classification, where supervision concerns nodes within client graphs. CeFGC addresses graph classification but redistributes generators to clients and aggregates trained classifiers across multiple communication stages (Wang et al., 2026). Strict one-shot graph classification—one upload per client, no further client contact—remains largely unexplored.

Diffusion models. Diffusion models define a forward process that gradually corrupts data and learn to reverse it; denoising diffusion probabilistic models establish this framework for continuous data with Gaussian noise (Ho et al., 2020), and D3PM generalizes it to discrete state spaces with categorical transition matrices and exact posterior sampling (Austin et al., 2021). DiGress instantiates discrete diffusion over graph nodes and edges, predicting clean node and edge states with a graph network (Vignac et al., 2023). The generative capacity of diffusion models has recently been brought to federated learning: on images, diffusion models synthesize proxy data for one-shot distillation and heterogeneity mitigation (Mendieta et al., 2024; Liu & Gan, 2025); on graphs, local diffusion generators produce synthetic graphs for cross-silo augmentation (Wang et al., 2026). OSFGC builds on this line but moves generation, verification, and fusion entirely to the server after a single upload.

D Privacy Analysis

Threat model. We analyze an honest-but-curious server that follows the protocol of Appendix A.1 but attempts to infer (i) raw training graphs, (ii) per-graph attributes such as the size of a specific graph, or (iii) membership of a target graph in the local collection of client k. The adversary observes exactly the public configuration, the single upload package $\mathcal{U}_k$ of Equation (15) from each client, and any quantity computed from them; it never observes client-local computations, raw graphs, or gradients.

The released package contains no raw record. By construction, $\mathcal{U}_k$ consists of generator parameters, class-level sketches, class counts, reliability scores, and bookkeeping metadata (Equation (15)); it contains no graph object, no per-graph node-count list, and no trained classifier. Every downstream server operation—candidate sampling, structural screening, CR-KMM weighting, and global training—is a deterministic function of $\{\mathcal{U}_k\}_{k=1}^{K}$ with public randomness only, hence a post-processing of the release. Quantifying the privacy of OSFGC therefore reduces to quantifying what the aggregates inside $\mathcal{U}_k$ reveal about a single local graph.

Bounded influence of a single graph. Write $\mathcal{G}_k \simeq \mathcal{G}_k'$ if the two local collections differ by adding or removing one graph of class c. The ℓ_2-sensitivity of a statistic f is $\Delta_2(f) = \max_{\mathcal{G}_k \simeq \mathcal{G}_k'} \|f(\mathcal{G}_k) - f(\mathcal{G}_k')\|_2$. The sketch uses L per-class histograms (graph size, node type, density, and degree). One graph contributes at most unit ℓ_2 mass to each histogram—one count for scalar-descriptor histograms, or a normalized profile for node-type and degree distributions—so the stacked count vector changes by at most

$$\Delta_2(\mathbf{n}) \le \sqrt{L}. \tag{31}$$

Each random feature satisfies $\|\mathbf{f}(\mathbf{s}(G))\|_2 \le b_f\sqrt{D}$ with $b_f = \sqrt{2/D}$, so the class-level moments released through Equation (25) satisfy

$$\Delta_2(\widehat{\mathbf{m}}_{k,c}^{f}) \le \frac{2b_f\sqrt{D}}{N_{k,c}} = \frac{2\sqrt{2}}{N_{k,c}}, \qquad \Delta_2(\widehat{\mathbf{v}}_{k,c}^{f}) \le \frac{2b_f^2\sqrt{D}}{N_{k,c}} = \frac{4}{\sqrt{D}\,N_{k,c}}. \tag{32}$$

The class-count vector has $\Delta_2(\widetilde{\boldsymbol{\pi}}_k) \le 1$, and the released reliability is a scalar clipped to $[q_{\min}, q_{\max}]$ whose unclipped value varies with the class-level training loss; treating the clipped value as the statistic and adding the Gaussian term after clipping, its sensitivity is at most $q_{\max} - q_{\min}$. Crucially, the influence of any single graph on the released moments decays as $1/N_{k,c}$: no released coordinate is a per-graph quantity.

Per-component differential privacy. With the optional Gaussian perturbations of Appendix A.3, each release is a Gaussian mechanism: releasing $\widetilde{\mathbf{x}} = \mathbf{x} + \boldsymbol{\xi}$ with $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ over a statistic of sensitivity Δ_2 is (ε, δ)-differentially private whenever

$$\sigma \geq \frac{\Delta_2\sqrt{2\ln(1.25/\delta)}}{\varepsilon}, \qquad 0 < \varepsilon < 1, \tag{33}$$

and the equivalent per-component budget is $\varepsilon = \Delta_2\sqrt{2\ln(1.25/\delta)}/\sigma$ (**?**). The clipping of Equation (25) and the normalization of Equation (24) are post-processings of the noised vector and preserve the guarantee. Substituting Equations (31)–(32) gives

$$\varepsilon_{\text{hist}} = \frac{\sqrt{2L\ln(1.25/\delta)}}{\sigma_h}, \qquad \varepsilon_{\text{mom}} = \frac{2\sqrt{2}\,\sqrt{2\ln(1.25/\delta)}}{\sigma_m N_{k,c}} = \frac{4\sqrt{\ln(1.25/\delta)}}{\sigma_m N_{k,c}}, \qquad \varepsilon_{\text{count}} = \frac{\sqrt{2\ln(1.25/\delta)}}{\sigma_\pi}, \tag{34}$$

where $\sigma_h, \sigma_m, \sigma_\pi$ are the noise levels of the histogram, moment, and count releases and $N_{k,c}$ is taken as the smallest released class count so that the bound covers the joint release over all classes. Basic composition bounds the complete sketch at $(\sum_j \varepsilon_j,\ \sum_j \delta_j)$, and allocating $\varepsilon/4$ and $\delta/4$ to each of the four component groups (histograms; first moments; second moments; counts and reliability) yields package-level (ε, δ).

Membership inference and reconstruction. A standard consequence of Equation (33) bounds any membership test T that observes only the noised sketch: for every graph G,

$$\Pr\left[T(\widetilde{\mathcal{S}}_k) = 1 \,\middle|\, G \in \mathcal{G}_k\right] \leq e^{\varepsilon}\Pr\left[T(\widetilde{\mathcal{S}}_k) = 1 \,\middle|\, G \notin \mathcal{G}_k\right] + \delta, \tag{35}$$

so the membership advantage obtainable from the sketch is at most $e^{\varepsilon} - 1 + \delta$. Two structural properties further restrict reconstruction: the server never receives a gradient, so record-reconstruction attacks that invert per-round gradient updates (Zhu et al., 2019; Melis et al., 2019) are inapplicable; and the interaction terminates after one upload, so no temporal composition arises—a multi-round baseline exposing (ε, δ) per round degrades to $(R\varepsilon, R\delta)$ after R rounds by basic composition, whereas $R_{\text{comm}} = 1$ here (Equation (16)).

Generator parameters and accounting for training. The generator parameters $\boldsymbol{\theta}_k$ are the only component that depends on raw graphs through more than bounded-influence aggregates. Two remarks qualify the exposure. First, only the final parameter vector is released: the adversary obtains no per-step gradients, and any memorized content is confined to a single static view rather than an interactive oracle. Second, if local diffusion training clips per-sample gradient norms to C and adds noise $\mathcal{N}(\mathbf{0}, \sigma_g^2 C^2\mathbf{I})$ for T sampled steps with sampling rate q, standard DP-SGD accounting certifies $(\varepsilon_g, \delta_g)$ for $\boldsymbol{\theta}_k$ (**?**), and the complete package is $(\varepsilon + \varepsilon_g,\ \delta + \delta_g)$-differentially private by composition.

Scope. Without training-time noise, $\boldsymbol{\theta}_k$ carries no differential-privacy guarantee, and the practical protection is structural: raw graphs never enter $\mathcal{U}_k$ (Equation (15)), every released coordinate has bounded and decaying per-graph influence (Equations (31)–(32)), and the protocol exposes exactly one message per client (Equation (16)). The experiments in this paper instantiate the zero-noise configuration and therefore claim this structural privacy, not differential privacy; enabling the calibrated perturbations above upgrades the sketch and, with DP-SGD training, the generator to an end-to-end (ε, δ) guarantee.