WHEN SHARING HURTS: RISK-REVERSIBLE KNOWLEDGE TRANSFER FOR FEDERATED GRAPH LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Cross-client graph transfer changes both a client's predictor and the representations used to construct later shared evidence. Harmful updates can therefore persist through subsequent rounds even after transfer is downweighted. We introduce Risk-Reversible Graph Transfer (R2GT), a federated graph model that gives each client control over this persistence. R2GT combines semantic and structural sharing with a validation-risk gate, a recoverable checkpoint, and cooldown-based reentry. Clients attempt transfer after warm-up and cooldown, retain candidate states whose validation scores remain within a tolerance of the stored reference, and restore that reference after larger declines. We establish trajectory-wise empirical validation control, extend it to population risk under explicit concentration and validation–test coupling conditions, and quantify the distinction between rollback and loss masking at revocation. Experiments across homophilic and heterophilic graphs with disjoint and overlapping partitions support the predictive value of shared evidence coupled with client-side recovery. Controlled paired analyses show that rollback improves next-round performance in 86.0–91.4% of recovery events and substantially reduces semantic and structural drift relative to mask-only continuation. Code is available at https://anonymous.4open.science/r/R2GT-7FA9.

1 INTRODUCTION

Graph data are often distributed across organizations or devices. Social interactions, product co-purchases, and citation neighborhoods form local subgraphs whose raw data cannot be pooled because of privacy, storage, or governance constraints. Federated learning enables collaborative training while keeping data local (McMahan et al., 2017), and graph neural networks learn representations from local topology and attributes (Kipf & Welling, 2017; Hamilton et al., 2017; Veličković et al., 2018). Their combination introduces a coupled form of heterogeneity: clients differ in feature distributions and label support, as well as in the neighborhoods through which message passing propagates information.

Recent federated graph methods address this heterogeneity by improving the information exchanged between clients. They model inter- and intra-client variation (Yu et al., 2025), separate semantic and structural evidence (Yu et al., 2026), filter noisy structural observations (Tang & Chen, 2026), or refresh compressed graph evidence during training (Chen et al., 2026). These designs determine which clients or statistics are suitable for collaboration. A further decision arises after shared evidence has entered local training: whether the resulting client state should continue to shape subsequent rounds.

This decision matters because transfer has persistent effects. An update changes node representations, which influence neighborhood aggregation and the statistics used for later sharing. A mismatch can therefore feed back into future transfer targets. Lowering the transfer weight changes subsequent gradients, while the client continues from the state already produced by earlier updates. Moreover, the value of shared evidence varies across clients and training stages, so a single transfer schedule cannot express every client's changing need for collaboration. These observations motivate treating *transfer as a reversible intervention* on the local training trajectory.

We introduce Risk-Reversible Graph Transfer (R2GT), which couples a graph predictor with a client-specific transfer state. Each client evaluates its post-update candidate against its best stored

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.

checkpoint. A candidate within the validation tolerance is retained; a larger decline triggers checkpoint restoration and a cooldown period before transfer is retried. Semantic and structural statistics supply shared evidence, while the client's own validation trajectory controls its use. This gives the client a concrete recovery operation alongside the ability to accept or abstain from transfer.

State recovery is central to this design. The restored checkpoint provides a previously observed predictor from which local learning can resume. In graph learning, this interrupts the persistence of representation changes that would otherwise influence both message passing and future shared statistics. R2GT thus connects three decisions over the course of training: admitting evidence, evaluating its effect, and restoring a reference state when the observed degradation exceeds the margin. Client heterogeneity enters these decisions through local validation evidence, allowing clients to follow different transfer schedules within the same model.

Our analysis characterizes the value of this recovery operation. Under uniform validation concentration, rollback bounds population validation regret relative to the stored reference; validation–test coupling extends this result to deployment downside. A separate result quantifies the gap between state restoration and loss masking at a revocation event. Experiments on homophilic and heterophilic graphs, with disjoint and overlapping client partitions, evaluate predictive performance and isolate the roles of transfer admission, rollback, and cooldown. Controlled rollback branches further trace recovery into the next round and the shared semantic and structural statistics. Together, the method and analysis make the persistence of transferred information an explicit object of federated graph learning.

2 Related Work

Federated graph learning. Federated graph learning extends federated optimization (McMahan et al., 2017; Kairouz et al., 2021) to clients that hold local subgraphs. Graph neural networks (Kipf & Welling, 2017; Hamilton et al., 2017; Veličković et al., 2018) couple feature learning with neighborhood aggregation, so differences in topology affect the compatibility of client updates. Graph-specific collaboration enriches local subgraphs with hidden representations (Wu et al., 2021), generates missing neighbors (Zhang et al., 2021a), personalizes aggregation by client similarity (Baek et al., 2023), or distills topology-aware knowledge (Zhu et al., 2024). These mechanisms expand the evidence available to local predictors while keeping raw graph data on clients.

Modeling graph heterogeneity. FedIIH models inter-client and intra-client heterogeneity through subgraph distributions and disentangled latent factors (Yu et al., 2025). FedSSA estimates class-wise node distributions and spectral structural statistics for separate semantic and structural alignment (Yu et al., 2026). Fed-Kalter uses Kalman-style dynamics to filter noisy feature-conditioned structural observations (Tang & Chen, 2026). DANCE studies text-attributed federated graphs through dynamic graph condensation and provenance-preserving evidence packs (Chen et al., 2026). FedHERO shares a structure learner through a dual-channel architecture to handle heterogeneous neighbor distributions (Chen et al., 2025). S2FGL combines a global knowledge repository with frequency alignment to address disrupted label signals and spectral client drift (Tan et al., 2025). These methods model the sources and reliability of shared evidence. R2GT studies the state that this evidence leaves behind after a client update.

Negative transfer and selective sharing. Source knowledge can degrade a target learner when it induces an unsuitable bias, a phenomenon studied as negative transfer (Rosenstein et al., 2005; Pan & Yang, 2010). Selective classification controls prediction risk by allowing abstention (El-Yaniv & Wiener, 2010; Geifman & El-Yaniv, 2017). R2GT applies selective decisions during training, where an accepted update changes the state used by future rounds. Its acceptance test evaluates the updated candidate, and its rollback operator restores a stored state after validation degradation. This stateful setting requires both control over incoming evidence and recovery from its observed effects.

Validation-guided collaboration. Client-side evaluation already guides sharing in general federated learning. FedFomo uses local validation loss to weight other clients' models (Zhang et al., 2021b); APFL learns personalized mixtures of local and global models (Deng et al., 2020). Zeno scores proposed gradient updates using estimated loss descent and update magnitude before server

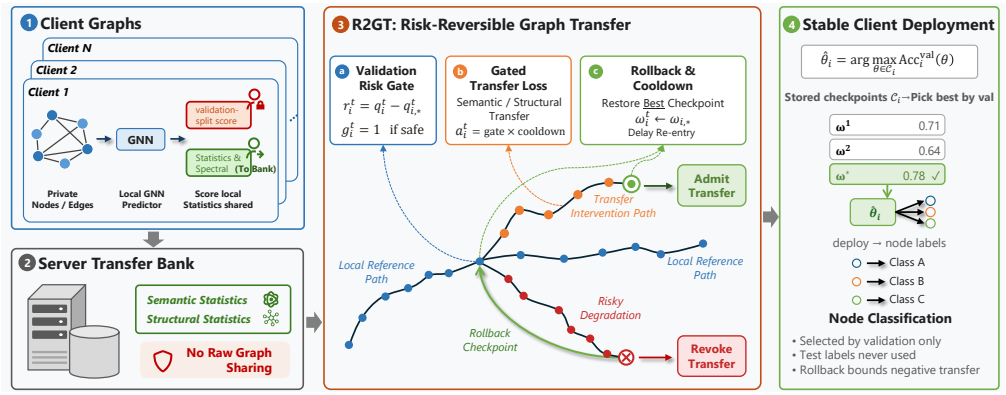


Figure 1: R2GT couples shared graph evidence with client-specific admission, checkpoint recovery, and cooldown-based re-entry.

aggregation (Xie et al., 2019). These approaches establish evaluation-based selection and continuous weighting as relevant alternatives. R2GT focuses on the post-update state retained for subsequent graph computation: its reference is a historical checkpoint, and recovery is followed by a client-specific retry delay. The distinction concerns recovery target and timing, beyond whether a validation signal is used.

Positioning. Client similarity, topology-aware aggregation, and distributional alignment determine how collaboration is constructed (Baek et al., 2023; Li et al., 2023; Yu et al., 2025; 2026). R2GT adds an explicit decision over the persistence of that collaboration in the client state. Shared evidence can be admitted, revoked, and retried, with a validation-selected checkpoint providing the recovery target.

3 Methodology

Figure 1 illustrates Risk-Reversible Graph Transfer (R2GT). We study semi-supervised node classification with M clients. Client i holds a local graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i, \mathbf{X}_i)$, training labels $\mathcal{Y}_i^{\mathrm{tr}}$, and a private validation split $\mathcal{V}_i^{\mathrm{val}}$. Each client trains a graph predictor $f_i(\cdot;\theta_i)$ and keeps its raw graph locally. The server summarizes model-side statistics into shared semantic and structural evidence. Appendix B gives the server-client procedure.

3.1 Transfer as a Reversible Intervention

Let A_0 denote the non-transfer learning dynamics and A_1 denote a cross-client transfer dynamics. At round t, client i has a local state θ_i^t. A conventional transfer method applies a shared update whenever its similarity or alignment rule is active:

$$\theta_i^{t+1} = A_1\left(\theta_i^t; \mathcal{S}_i^t, \mathcal{B}^t\right), \tag{1}$$

where $\mathcal{S}_i^t$ is client-side evidence and $\mathcal{B}^t$ is a server-side bank of transferable information, such as class prototypes, spectral coefficients, or cluster representatives. The resulting state seeds the next round, including the computation of future sharing statistics. R2GT separates the shared *evidence*, the *action* that admits it into training, and the *state* produced by that action. This separation lets a client evaluate an intervention's effect and restore a stored state when the observed outcome deteriorates.

We introduce an intervention variable $a_i^t \in \{0, 1\}$, where $a_i^t = 1$ means that client i admits transfer at round t, and $a_i^t = 0$ means that it follows a local-reference branch:

$$\tilde{\theta}_i^{t+1} = (1 - a_i^t)A_0(\theta_i^t) + a_i^t A_1(\theta_i^t; \mathcal{S}_i^t, \mathcal{B}^t). \tag{2}$$

Equation 2 forms a candidate predictor $\tilde{\theta}_i^{t+1}$ through one of two update branches selected by the binary variable a_i^t. The branches are implemented as local gradient steps with the transfer loss activated or suppressed. The recovery target is the client's best observed validation checkpoint, which may itself have benefited from earlier transfer. It is maintained within the same training run.

3.2 CLIENT-SIDE TRANSFER RISK

The true transfer gain for client i is:

$$\Delta_i = \mathrm{Acc}_i^{\mathrm{test}}(\theta_i^{A_1}) - \mathrm{Acc}_i^{\mathrm{test}}(\theta_i^{A_0}), \tag{3}$$

which motivates evaluating transfer at the client level. At the start of round t, client i holds a retained state ω_i^t and its best validation checkpoint with score $q_{i,\star}^t$. Local optimization produces a candidate $\tilde{\omega}_i^{t+1}$. We evaluate its validation score $\tilde{q}_i^{t+1}$ after the local updates and before checkpoint replacement or recovery, and define

$$r_i^{t+1} = \tilde{q}_i^{t+1} - q_{i,\star}^t. \tag{4}$$

The reference score is held fixed throughout this round's candidate evaluation. The client accepts the candidate when its validation decline is within the tolerance margin ϵ:

$$g_i^{t+1} = \mathbb{I}\left[r_i^{t+1} \geq -\epsilon\right]. \tag{5}$$

Thus g_i^{t+1} is an outcome-dependent acceptance decision, evaluated on the candidate before recovery.

A separate eligibility variable controls whether transfer is attempted during the round. Let $c_i^t \in \{0, \ldots, \kappa\}$ be the remaining cooldown rounds. With the first t_{warm} rounds reserved for local warm-up,

$$a_i^t = \mathbb{I}[t \geq t_{\mathrm{warm}}]\,\mathbb{I}[c_i^t = 0]. \tag{6}$$

The eligibility decision precedes local optimization; the acceptance decision follows it. Re-evaluating an already accepted or restored state on the same deterministic validation split would satisfy the margin by construction. Evaluating the new candidate instead allows the risk test to detect degradation introduced during the current round.

3.3 RISK-REVERSIBLE GRAPH TRANSFER

Each client maintains a recoverable checkpoint $\omega_{i,\star}^t$ that achieves $q_{i,\star}^t$. We use θ for predictor parameters and ω for the recoverable state; a tilde marks the unaccepted candidate and a star marks the stored reference. The checkpoint policy retains one best-validation checkpoint per client. Parameter-only recovery stores the predictor, while full-state recovery additionally includes optimizer buffers and other stateful components. These policies share the same acceptance rule but resume different optimization trajectories. Appendix C.13 specifies the default parameter-only policy and storage costs. The local objective at round t is:

$$\mathcal{L}_i^t = \mathcal{L}_{\mathrm{sup}}(\theta_i; \mathcal{G}_i) + a_i^t \lambda_t \mathcal{L}_{\mathrm{tr}}(\theta_i; \mathcal{B}^t), \tag{7}$$

where $\mathcal{L}_{\mathrm{sup}}$ is the supervised node classification loss on local labels and $\mathcal{L}_{\mathrm{tr}}$ is a graph transfer loss. In our implementation, $\mathcal{L}_{\mathrm{tr}}$ can align semantic class statistics or structural spectral statistics, but it is always multiplied by the transfer eligibility variable. The scalar λ_t follows a warm-up schedule so that the local predictor first establishes a client-specific decision boundary before cross-client pressure is applied.

The transfer loss itself is modular. For semantic sharing, the server bank can contain class-wise distributional summaries $\mathcal{B}_{\mathrm{sem}}^t = \{(\mu_{k,c}^t, \Sigma_{k,c}^t)\}$, where k indexes a client cluster and c indexes a class. A typical semantic loss takes the form:

$$\mathcal{L}_{\mathrm{sem}} = \sum_{c \in \mathcal{Y}_i} w_{i,c}\, d\Big(p_{\theta_i}(h \mid c, \mathcal{G}_i), \mathcal{N}(\mu_{k(i),c}^t, \Sigma_{k(i),c}^t)\Big), \tag{8}$$

where $d(\cdot, \cdot)$ can be a KL divergence or another distributional distance. For structural sharing, the bank can contain spectral or propagation statistics $\mathcal{B}_{\mathrm{str}}^t$, with a loss $\mathcal{L}_{\mathrm{str}}$ aligning local structural responses to a cluster-level representative. The full transfer loss is:

$$\mathcal{L}_{\mathrm{tr}} = \alpha_{\mathrm{sem}} \mathcal{L}_{\mathrm{sem}} + \alpha_{\mathrm{str}} \mathcal{L}_{\mathrm{str}}. \tag{9}$$

Transfer eligibility controls when these semantic and structural losses influence local optimization. Checkpoint recovery controls which resulting states persist into subsequent rounds.

The acceptance decision determines the state retained for the next round:

$$\omega_i^{t+1} = \begin{cases} \tilde{\omega}_i^{t+1}, & g_i^{t+1} = 1, \\ \omega_{i,\star}^t, & g_i^{t+1} = 0. \end{cases} \tag{10}$$

A rejected candidate triggers checkpoint recovery and cooldown. The same validation check applies to local-only candidates during warm-up and cooldown. Restoring the checkpoint replaces the degraded predictor that would otherwise initialize the next round. Loss masking, by comparison, suppresses subsequent transfer gradients while retaining that predictor.

Recovery is triggered by the observed outcome of an update. A candidate can therefore degrade validation performance before revocation occurs. Once the decline exceeds the margin, restoration removes that candidate from the continuing client trajectory, and cooldown provides a period of local adaptation before transfer is retried.

3.4 Transfer-Abstaining Deployment

For reporting and deployment, each client's predictor is selected by validation utility:

$$\hat{\theta}_i = \arg\max_{\theta \in \mathcal{C}_i} \mathrm{Acc}_i^{\mathrm{val}}(\theta), \tag{11}$$

where $\mathcal{C}_i$ denotes the checkpoints encountered during training. Its maximizing checkpoint is maintained online, so deployment requires one stored reference rather than the full history. Checkpoint selection and gate tuning use validation labels exclusively; test labels are reserved for evaluation.

Each round combines local supervised updates, gated transfer, validation evaluation, and checkpoint maintenance. An improved candidate validation score refreshes the checkpoint; a decline beyond ϵ triggers restoration. All other branches retain both the previous checkpoint and its score. Larger margins tolerate more validation fluctuation, while smaller margins trigger recovery sooner.

After a violation, the client restores $\omega_{i,\star}^t$ and sets $c_i^{t+1} = \kappa$. Otherwise $c_i^{t+1} = \max(c_i^t - 1, 0)$. A violation during cooldown restarts the counter. Otherwise, transfer becomes eligible after the counter reaches zero, and the resulting candidate is checked by the same validation-risk test. The cooldown spaces out transfer attempts while preserving the opportunity to use later shared evidence.

3.5 Relation to Existing Sharing Models

Two limiting cases clarify the model. Setting $a_i^t = 1$ after warm-up and disabling rollback gives always-on cross-client transfer. Setting $a_i^t = 0$ removes semantic and structural transfer losses. R2GT operates between these cases: client validation trajectories determine when shared evidence enters optimization and when checkpoint recovery is triggered.

4 Theoretical Analysis

We analyze how checkpoint recovery controls the cost of retaining a transfer-influenced state. The comparison is with the client's best stored reference. We first establish an empirical invariant along the training trajectory, then state the conditions that extend it to population validation utility and test-side downside.

4.1 Observable Regret Under Rollback

For client i, let $Q_i(\theta)$ be the population validation utility and $\widehat{Q}_i(\theta)$ be the observed validation utility. Let $\mathcal{C}_i^t$ contain the states visited or stored before round t together with the current candidate $\tilde{\theta}_i^{t+1}$. The reference $\theta_{i,\star}^t$ is the parameter component of the best recoverable checkpoint selected before evaluating that candidate.

Trajectory-wise empirical control. With a fixed validation metric and exact predictor restoration, define $\rho_i^{t+1} = [q_{i,\star}^{t+1} - \widehat{Q}_i(\theta_i^{t+1})]_+$. For any adaptively generated sequence of candidates,

$$\rho_i^{t+1} \le \epsilon \quad \text{for every } t, \qquad \sum_{t=0}^{T-1} \rho_i^{t+1} \le T\epsilon. \tag{12}$$

An improved candidate becomes the reference; rejection restores the reference; every remaining accepted candidate is within the margin. These three branches prove the bound pathwise, including under repeated validation reuse. Population interpretation requires an additional concentration condition.

Assumption 1 (Uniform validation concentration)**.** *For each client i, there exists $\xi_i \geq 0$ such that, with high probability,*

$$\sup_{\theta \in \mathcal{C}_i^t} \left| \widehat{Q}_i(\theta) - Q_i(\theta) \right| \leq \xi_i.$$

Theorem 1 (Rollback controls observable client regret)**.** *Suppose Assumption 1 holds. Let $\hat{\theta}_i^{t+1}$ be the state deployed by the rollback operator with margin ϵ, i.e., the candidate state is accepted only if its observed validation utility is within ϵ of $\widehat{Q}_i(\theta_{i,\star}^t)$; otherwise the client restores the recoverable checkpoint whose parameter component is $\theta_{i,\star}^t$. Then, with high probability,*

$$Q_i(\theta_{i,\star}^t) - Q_i(\hat{\theta}_i^{t+1}) \leq \epsilon + 2\xi_i.$$

Theorem 1 separates the permitted validation drop ϵ from the uncertainty cost $2\xi_i$. Rejected candidates return to the stored reference; accepted candidates remain within this bound. For adaptively generated checkpoints, the concentration assumption must cover the full validation reuse process. Appendix A.6 gives the audit interpretation and the independent-candidate specialization.

4.2 From Validation Regret to Downside Risk

Let $U_i(\theta)$ denote the population test utility and define the client-level downside of deploying $\hat{\theta}_i^{t+1}$ instead of the best stored checkpoint as

$$D_i^{t+1} = \left(U_i(\theta_{i,\star}^t) - U_i(\hat{\theta}_i^{t+1}) \right)_+ .$$

To connect validation rollback to test-side downside, we use a standard coupling assumption over candidate states.

Assumption 2 (Validation-test coupling over candidate states)**.** *For each client i, there exists $\gamma_i \geq 0$ such that for all $\theta_a, \theta_b \in \mathcal{C}_i^t$,*

$$U_i(\theta_a) - U_i(\theta_b) \geq Q_i(\theta_a) - Q_i(\theta_b) - \gamma_i.$$

Proposition 1 (Downside-risk bound)**.** *Under Assumptions 1 and 2, the average downside risk of the rollback deployment rule satisfies:*

$$\frac{1}{M} \sum_{i=1}^{M} D_i^{t+1} \leq \epsilon + \frac{1}{M} \sum_{i=1}^{M} (2\xi_i + \gamma_i).$$

The margin ϵ controls empirical retention, whereas $2\xi_i$ and γ_i quantify uncertainty in extending it beyond the reused validation split. The deployment bound is conditional on these terms; Eq. 12 remains an exact property of the training rule. Appendix A.6 separates adaptive training from an independent audit of fixed checkpoints.

4.3 Why the Graph Setting Needs Reversibility

Rollback and loss masking differ in the state that continues after revocation. Masking suppresses future transfer gradients while retaining the current parameters; rollback restores the stored checkpoint. Appendix A quantifies their utility gap at the revocation event. In graphs, state persistence also affects how later evidence is constructed. If T_i is the local message-passing operator and δ_i^t is a transfer-induced perturbation, a local layer can be approximated as

$$h_i^{t+1} = \sigma\left(T_i h_i^t W^t + \delta_i^t \right).$$

The perturbed representations enter later semantic or structural statistics, coupling future transfer targets to earlier interventions. Restoring the reference predictor interrupts this dependence on the rejected state. Proofs of Theorem 1, Proposition 1, and the revocation separation result appear in Appendix A.

Repeated recovery. Let $H = (T - t_{\mathrm{warm}})_+$ and let $N_i^{\mathrm{rej,tr}}$ count rejected transfer-active rounds. Cooldown enforces

$$N_i^{\mathrm{rej,tr}} \leq \left\lceil \frac{H}{\kappa + 1} \right\rceil .$$

Each rejected transfer attempt is followed by at least κ local-only rounds; further local violations can extend that interval. This bounds the frequency of failed transfer attempts, while Eq. 12 controls the retained validation state. Both properties hold with changing shared evidence and address repeated recovery without requiring convergence of that evidence.

5 Experiments

5.1 Experiment Settings

Evaluation Tasks. We evaluate R2GT on federated node classification over homophilic and heterophilic graphs under METIS-based disjoint and overlapping client partitions (Karypis & Kumar, 1997; Baek et al., 2023; Yu et al., 2025). Detailed task definitions are provided in Appendix C.1.

Model Architecture and Datasets. R2GT uses the same graph encoder and classifier architecture across datasets. We evaluate eleven public benchmarks spanning citation, co-purchase, web, and heterophilic node-classification graphs (Yang et al., 2016; Shchur et al., 2018; Hu et al., 2020; Platonov et al., 2023); dataset, partition, and architecture details are given in Appendices C.2–C.3.

Baselines. We compare R2GT with local training, standard federated optimization, personalized federated learning, and graph-specific federated learning methods, including FedIIH and FedSSA as recent strong graph-sharing baselines (Yu et al., 2025; 2026). Baseline details are in Appendix C.4.

Implementations. Each setting uses fixed client partitions, fixed communication budgets, and validation-only model selection, with the same selection rule applied to R2GT and all baselines; results are reported as mean $\pm$ standard deviation over five random seeds. We report accuracy for multi-class datasets and AUC for Minesweeper, Tolokers, and Questions (Platonov et al., 2023; Yu et al., 2025); implementation details are in Appendix C.5.

5.2 Main Results

Homophilic Datasets. Table 1 reports results under disjoint client partitions. R2GT achieves the best average performance, increasing the score from 82.24 for FedSSA to 82.86 across citation, co-purchase, and ogbn-arxiv settings. The gain shows that client-side recovery can complement effective graph sharing even when neighborhood labels are relatively aligned. Overlapping results are in Appendix C.6.

Heterophilic Datasets. Table 2 reports results under disjoint client partitions. R2GT increases the average score from 64.45 for FedSSA to 65.56. The larger average gain in this group is consistent with the motivation for client-specific transfer control when neighborhood evidence is less reliably aligned with labels. Section 5.3 isolates the contribution of the control components; overlapping results appear in Appendix C.6.

5.3 Ablation Study

We evaluate three reduced variants: *w/o transfer* removes cross-client transfer, *w/o rollback* masks future transfer without restoring the pre-harm state, and *w/o cooldown* permits transfer in the next round after revocation. Appendix Figure 3 shows that shared graph evidence drives the largest gain. State restoration is the next most important component: removing rollback lowers performance by 1.48 percentage points on average, compared with 0.67 for removing cooldown. Appendix C.7 reports the representative figure and full evaluation grid.

5.4 Rollback Restores the Future Sharing Trajectory

We compare FedSSA with validation-based checkpoint selection (FedSSA+CKPT), risk masking without state restoration (Mask only), count-matched rollback at random rounds (Random rollback),

Table 1: Main results on homophilic graph datasets under non-overlapping subgraph partitioning.

	Cora			CiteSeer			PubMed			
Methods	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients	–
Local	80.28±0.23	78.65±0.22	82.11±0.24	68.72±0.07	66.54±0.14	66.38±0.16	84.12±0.18	80.12±0.28	80.28±0.05	–
FedAvg	72.47±3.52	69.83±1.52	69.53±2.67	69.34±0.55	63.22±2.83	65.02±1.55	79.45±0.12	80.10±0.30	78.64±0.24	–
FedProx	69.63±4.21	65.33±4.53	60.83±5.12	69.29±0.98	63.05±2.74	65.17±1.21	79.52±0.24	79.93±0.26	78.18±0.27	–
FedPer	79.34±0.35	78.01±0.28	79.82±0.34	68.82±0.30	69.88±0.27	66.96±0.28	85.89±0.20	81.52±0.30	82.36±0.30	–
GCFL	79.11±0.42	77.98±0.33	80.12±0.48	68.76±0.45	68.53±0.15	66.71±0.07	85.22±0.31	81.50±0.22	81.58±0.34	–
FedGNN	79.18±0.55	69.85±0.87	74.15±2.86	68.13±0.78	55.12±2.68	52.48±5.23	79.58±0.22	80.62±0.42	79.28±0.52	–
FedSage+	70.58±2.94	67.98±1.24	63.47±8.74	69.28±0.55	65.11±2.54	65.84±0.68	79.65±0.23	80.01±0.28	78.48±0.23	–
FED-PUB	81.32±0.21	80.22±0.17	82.87±0.38	70.48±0.38	71.43±0.47	68.02±0.14	86.85±0.11	83.56±0.20	83.17±0.28	–
FedGTA	80.46±0.45	78.32±0.38	81.18±0.32	68.54±0.12	70.42±0.31	70.28±0.16	87.81±0.02	84.09±0.03	84.72±0.08	–
AdaFGL	80.11±0.36	79.23±0.12	80.95±0.09	69.39±0.25	71.34±0.12	71.24±0.38	86.96±0.26	84.25±0.12	84.21±0.23	–
FedTAD	80.65±0.28	79.54±0.15	81.23±0.18	68.76±0.34	68.95±0.62	68.45±0.57	84.08±0.14	81.93±0.19	81.96±0.20	–
FedIIH	81.84±0.16	80.65±0.12	83.25±0.14	70.48±0.22	75.63±0.08	73.68±0.35	87.86±0.17	84.92±0.20	84.83±0.28	–
FedSSA	82.34±0.05	80.91±0.04	84.17±0.09	69.74±0.08	76.45±0.10	74.11±0.09	87.92±0.07	85.10±0.13	85.01±0.14	–
R2GT (ours)	**82.55**±0.04	**81.64**±0.03	**84.37**±0.07	**70.54**±0.09	**76.77**±0.07	**74.90**±0.07	**88.39**±0.09	**85.97**±0.11	**85.21**±0.16	–

	Amazon-Computer			Amazon-Photo			ogbn-arxiv			Avg.
Methods	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients	All
Local	89.22±0.13	88.59±0.16	89.35±0.19	91.53±0.10	91.58±0.03	90.09±0.14	66.42±0.08	64.58±0.10	64.72±0.07	79.07
FedAvg	84.28±1.52	79.22±0.25	74.55±0.26	89.72±0.72	82.88±3.18	81.00±0.95	65.19±0.09	64.10±0.12	62.90±0.15	73.97
FedProx	84.65±1.12	83.44±0.95	72.82±1.17	90.20±0.42	80.62±3.98	81.97±0.32	64.85±0.18	64.03±0.16	62.70±0.06	73.12
FedPer	89.06±0.30	89.38±0.06	87.65±0.40	91.25±0.35	91.48±0.25	90.25±0.09	66.52±0.07	64.65±0.20	64.32±0.13	79.29
GCFL	88.45±0.75	89.68±0.17	88.88±0.23	91.82±0.27	91.82±0.28	90.45±0.19	66.45±0.14	64.75±0.10	64.74±0.06	79.25
FedGNN	87.48±0.18	87.86±0.35	82.92±0.15	90.08±0.62	86.88±1.78	80.65±3.89	65.12±0.20	63.87±0.28	63.46±0.07	74.82
FedSage+	84.43±0.52	80.16±0.98	70.21±0.78	90.59±0.40	76.55±6.92	80.24±1.02	65.35±0.11	64.18±0.15	62.97±0.22	73.06
FED-PUB	89.80±0.08	90.22±0.14	89.92±0.11	93.11±0.18	92.45±0.19	91.15±0.14	67.42±0.11	66.25±0.10	66.30±0.14	80.81
FedGTA	86.10±0.19	86.32±0.22	84.77±0.72	93.15±0.13	93.28±0.24	91.15±0.18	59.98±0.06	59.88±0.11	58.42±0.16	78.83
AdaFGL	79.62±0.08	83.27±0.28	84.31±0.25	86.52±0.21	89.58±0.71	87.77±0.08	52.38±0.17	51.42±0.32	50.62±0.10	76.29
FedTAD	81.72±0.95	85.17±0.30	83.71±1.28	92.12±0.35	90.32±0.11	88.82±0.72	65.01±0.16	63.72±0.27	64.11±0.15	78.35
FedIIH	89.80±0.14	90.53±0.24	90.26±0.08	93.24±0.04	93.90±0.10	91.15±0.10	69.94±0.08	68.98±0.04	68.32±0.06	82.18
FedSSA	89.67±0.07	90.90±0.13	90.33±0.09	93.56±0.15	93.24±0.14	91.14±0.07	69.41±0.13	68.90±0.08	67.50±0.13	82.24
R2GT (ours)	**89.86**±0.08	**90.93**±0.16	**90.45**±0.10	**93.66**±0.18	**93.95**±0.10	**91.22**±0.06	**71.35**±0.15	**70.09**±0.10	**69.68**±0.10	**82.86**

Table 2: Main results on heterophilic graph datasets under non-overlapping subgraph partitioning. Accuracy is reported for Roman-empire and Amazon-ratings; AUC is reported for Minesweeper, Tolokers, and Questions. Avg. columns give arithmetic means across the five datasets at each client count; All averages all 15 settings.

	Roman-empire			Amazon-ratings			Minesweeper			
Methods	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients	–
Local	27.91±0.14	18.88±0.29	14.73±0.35	42.09±0.34	44.58±0.21	45.75±0.02	70.25±0.19	71.62±0.18	70.99±0.10	–
FedAvg	33.24±0.29	25.83±0.35	23.28±0.43	38.35±0.58	40.31±0.15	41.07±0.23	71.85±0.09	71.02±0.08	72.25±0.18	–
FedProx	22.21±0.65	16.85±1.55	14.35±0.54	34.75±0.02	34.73±0.13	36.61±0.06	71.15±0.30	72.02±0.22	72.40±0.41	–
FedPer	14.95±1.93	6.09±1.24	6.53±3.04	34.44±0.33	30.24±1.11	36.61±0.03	57.86±11.50	66.83±7.72	54.64±12.54	–
GCFL	35.00±0.13	30.79±0.26	29.17±0.28	34.75±0.06	34.73±0.15	36.61±0.02	71.28±0.14	73.25±0.13	70.12±0.20	–
FedGNN	24.51±0.12	19.48±0.12	17.83±0.02	34.63±0.07	34.59±0.00	36.09±0.08	71.39±0.14	72.45±0.08	72.61±0.30	–
FedSage+	51.75±0.00	39.33±0.00	29.69±0.00	34.65±0.00	34.58±0.00	36.67±0.02	76.85±0.00	74.23±0.00	70.61±0.00	–
FED-PUB	35.15±0.29	27.12±0.33	23.94±0.35	42.31±0.45	42.68±0.19	45.10±0.55	71.42±0.02	73.06±0.78	72.32±0.96	–
FedGTA	56.09±0.24	51.09±0.21	51.15±0.31	39.09±0.73	37.28±0.48	38.90±0.13	72.62±1.72	74.11±1.33	70.48±4.99	–
AdaFGL	62.24±0.20	54.66±0.08	53.95±0.29	39.57±0.07	40.15±0.00	42.29±0.15	72.33±1.24	72.26±2.35	72.08±1.44	–
FedTAD	39.63±0.21	34.93±0.42	33.36±0.14	41.48±0.36	41.19±0.32	44.22±0.29	71.57±0.47	73.38±0.12	73.62±0.03	–
FedIIH	62.92±0.06	56.53±0.31	56.15±0.12	42.15±0.26	42.07±0.11	44.91±0.04	73.35±0.02	74.66±0.04	73.68±0.02	–
FedSSA	62.65±0.11	57.47±0.07	56.13±0.12	42.66±0.15	43.36±0.11	45.41±0.08	81.87±0.11	83.11±0.10	83.93±0.11	–
R2GT (ours)	**63.64**±0.09	**57.99**±0.10	**56.41**±0.13	**43.96**±0.20	**45.21**±0.19	**45.84**±0.07	**84.55**±0.09	**84.32**±0.06	**84.55**±0.07	–

	Tolokers			Questions			Avg.			
Methods	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients	All
Local	66.38±0.19	70.80±0.25	60.78±0.74	63.32±0.63	51.89±0.11	55.54±0.23	53.99	51.55	49.56	51.70
FedAvg	60.34±0.34	54.44±0.55	54.87±0.35	63.30±0.25	52.71±0.24	56.05±0.17	53.42	48.86	49.50	50.59
FedProx	42.52±0.26	40.85±0.20	38.90±0.68	44.21±0.42	39.25±0.37	42.55±0.12	42.97	40.74	40.96	41.56
FedPer	46.30±10.87	54.54±14.55	43.41±12.77	56.10±10.33	52.32±10.68	58.93±1.72	41.93	42.00	40.02	41.32
GCFL	64.11±1.29	59.48±0.94	57.41±0.77	58.21±1.30	52.80±0.18	56.88±0.53	52.67	50.21	50.04	50.97
FedGNN	42.72±0.30	41.27±0.06	39.19±0.81	44.40±0.02	39.44±0.13	43.11±0.12	43.53	41.45	41.77	42.25
FedSage+	74.82±0.00	70.88±0.00	68.38±0.00	61.75±0.00	58.96±0.00	55.92±0.00	59.96	55.60	52.25	55.94
FED-PUB	70.61±0.64	72.03±0.75	63.86±0.65	64.45±4.39	53.55±0.57	58.10±3.21	56.79	53.69	52.66	54.38
FedGTA	60.55±0.50	54.76±1.32	56.52±1.77	62.30±2.10	52.20±1.73	57.30±0.39	58.13	53.89	54.87	55.63
AdaFGL	59.02±2.40	54.33±2.33	55.26±3.22	60.95±2.30	52.75±1.25	58.42±0.54	58.82	54.83	56.40	56.68
FedTAD	60.64±0.28	53.95±1.90	54.12±1.74	65.61±1.08	52.36±1.17	57.12±1.40	55.79	51.16	52.49	53.15
FedIIH	70.86±0.29	70.85±0.10	68.97±0.09	65.03±0.03	61.94±0.08	60.97±0.08	62.86	61.21	60.94	61.67
FedSSA	75.06±0.08	74.02±0.11	70.43±0.08	66.74±0.12	62.11±0.14	61.84±0.07	65.80	64.01	63.55	64.45
R2GT (ours)	**75.42**±0.04	**74.27**±0.09	**71.84**±0.08	**67.35**±0.19	**65.74**±0.13	**62.34**±0.09	**66.98**	**65.51**	**64.20**	**65.56**

and R2GT. Each executed R2GT rollback also forks the same pre-decision state into rollback and mask-only branches with shared subsequent randomness. This paired design measures rescue at the event and one round later, together with semantic and structural drift from the stored safe state. Appendix C.8 provides the protocol and diagnostics.

Targeted rollback improves on FedSSA+CKPT by 0.32–2.44 percentage points and exceeds count-matched random rollback by 0.72–1.48 points (Figure 2(a)). Across 100–300 events per setting, the paired branches show 1.85–2.72 points of immediate rescue and 2.12–3.15 points one round

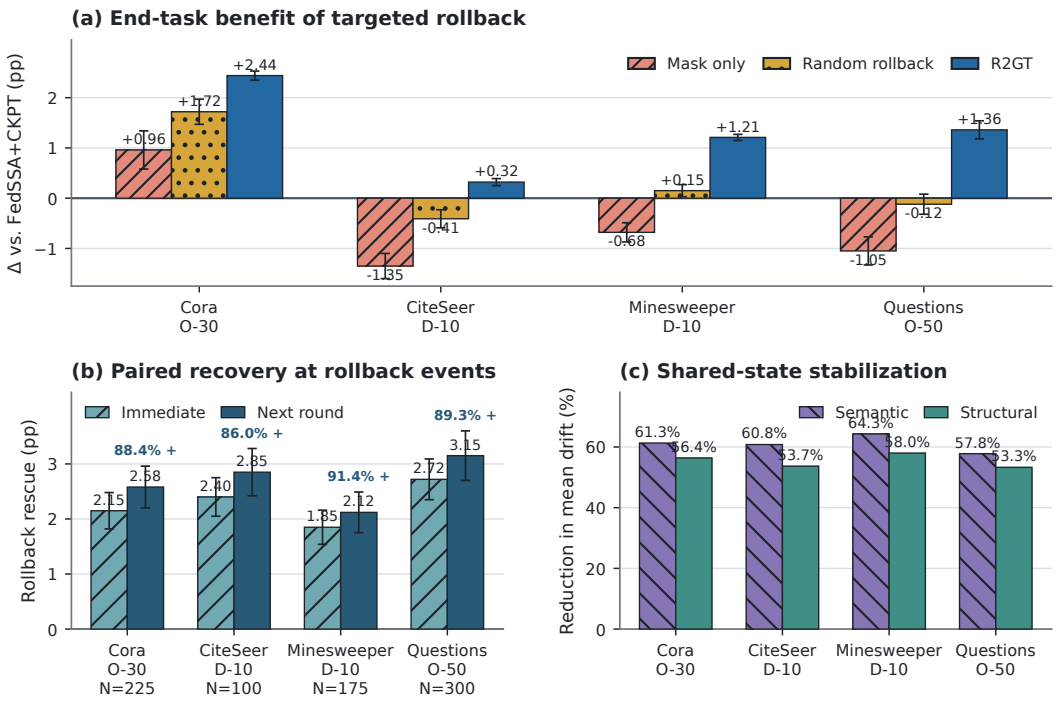


Figure 2: Structural effectiveness of rollback. (a) End-task difference from FedSSA+CKPT; error bars are standard deviations over five paired seeds. Random rollback matches R2GT's rollback count. (b) Paired rescue at the event and one round later with 95% confidence intervals; annotations give the fraction of positive next-round events. (c) Relative reduction in mean normalized drift. "D" and "O" denote disjoint and overlapping partitions.

later (Figure 2(b)). The next-round effect is positive for 86.0–91.4% of events and is larger than the immediate effect in every setting, showing that restoration improves the state from which subsequent training proceeds.

Rollback also reduces mean semantic-statistic drift by 57.8–64.3% and structural-statistic drift by 53.3–58.0% relative to mask-only continuation (Figure 2(c)). Performance recovery and shared-state stabilization thus trace checkpoint restoration into the future transfer trajectory.

5.5 Training Dynamics and Efficiency

R2GT reaches a higher plateau than the strongest competing baseline and remains stable in later rounds (Appendix C.9). Sensitivity results for transfer strength, rollback margin, warm-up, and cooldown appear in Appendix C.10. Across the four dataset groups, mean gains over FedSSA range from 0.62 to 1.15 percentage points, while median per-round cost ratios remain 1.00–1.01 (Appendix Table 7). Recovery stores one local checkpoint without changing the shared payload.

Appendix C.6 reports overlapping-partition results, Appendices C.7–C.10 expand the ablation, convergence, and sensitivity analyses, Appendix C.8 details the rollback-event protocol, Appendix C.11 audits validation-risk dynamics, and Appendices C.12–C.13 report computational and storage costs.

6 Conclusion

We introduced R2GT to make cross-client graph transfer reversible through validation-based retention, checkpoint recovery, and cooldown-based re-entry. The analysis establishes empirical validation control and its population-risk conditions. Across homophilic and heterophilic graphs, ablations and paired rollback branches show that recovery improves subsequent performance and stabilizes the statistics used for later transfer. R2GT thus connects the predictor retained in one round to the graph evidence shared in the next.

AI USE STATEMENT

We use generative AI tools to assist the polishing of the manuscript. Study statements, experiment design decisions, code changes, and any reported results were personally designed by the authors and derived from the project source code, experiment logs, and measurement files.

REPRODUCIBILITY STATEMENT

We use fixed client partitions, fixed communication budgets, and validation-only model selection. The experimental section reports the selection rule, main hyperparameters, and rollback-margin settings. Source code, metric JSON files, and launch commands are released in the anonymous repository so that each reported setting can be reproduced from the same public graph partitions.

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*, 2023.

Zekai Chen, Haodong Lu, Xunkai Li, Henan Sun, Jia Li, Hongchao Qin, Rong-Hua Li, and Guoren Wang. Dance: Dynamic, available, neighbor-gated condensation for federated text-attributed graphs. In *Proceedings of the International Conference on Machine Learning*, 2026.

Zihan Chen, Xingbo Fu, Yushun Dong, Jundong Li, and Cong Shen. FedHERO: A federated learning approach for node classification task on heterophilic graphs. *Transactions on Machine Learning Research*, 2025.

Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(53):1605–1641, 2010.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, 2017.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, 2017.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, 2020.

Keke Huang, Yu Guang Wang, Ming Li, and Pietro Lio. How universal polynomial bases enhance spectral graph neural networks: Heterophily, over-smoothing, and over-squashing. In *Proceedings of the International Conference on Machine Learning*, 2024.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurelien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2): 1–210, 2021.

George Karypis and Vipin Kumar. METIS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices. Technical Report 97-061, Department of Computer Science, University of Minnesota, November 1997. Version 3.0.3.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

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.

Xunkai Li, Zhengyu Wu, Wentao Zhang, Yinlin Zhu, Rong-Hua Li, and Guoren Wang. Fedgta: Topology-aware averaging for federated graph learning. volume 17, pp. 41–50, 2023. doi: 10.14778/3617838.3617842.

Xunkai Li, Zhengyu Wu, Wentao Zhang, Henan Sun, Rong-Hua Li, and Guoren Wang. Adafgl: A new paradigm for federated node classification with topology heterogeneity. In *Proceedings of the IEEE International Conference on Data Engineering*, pp. 2517–2530, 2024. doi: 10.1109/ICDE60146.2024.00198.

Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. Disentangled graph convolutional networks. In *Proceedings of the International Conference on Machine Learning*, 2019.

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*, 2017.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. doi: 10.1109/TKDE.2009.191.

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 *International Conference on Learning Representations*, 2023.

Michael T. Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G. Dietterich. To transfer or not to transfer. In *NIPS 2005 Workshop on Transfer Learning*, 2005.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. In *Relational Representation Learning Workshop*, 2018.

Zihan Tan, Suyuan Huang, Guancheng Wan, Wenke Huang, He Li, and Mang Ye. S2FGL: Spatial spectral federated graph learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 58591–58602, 2025.

Bisheng Tang and Xiaojun Chen. Federated graph learning via structure-aware fusion using a kalman framework with learnable dynamics. In *Proceedings of the International Conference on Machine Learning*, 2026.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Chuhan Wu, Fangzhao Wu, Yang Cao, Yongfeng Huang, and Xing Xie. Fedgnn: Federated graph neural network for privacy-preserving recommendation. *arXiv preprint arXiv:2102.04925*, 2021.

Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 6893–6901, 2019.

Han Xie, Jing Ma, Li Xiong, and Carl Yang. Federated graph classification over non-iid graphs. In *Advances in Neural Information Processing Systems*, 2021.

Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of the International Conference on Machine Learning*, 2016.

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*, 2025.

Wentao Yu, Sheng Wan, Shuo Chen, Bo Han, and Chen Gong. Heterogeneity-aware knowledge sharing for graph federated learning. In *Proceedings of the International Conference on Machine Learning*, 2026.

Ke Zhang, Carl Yang, Xiaoxiao Li, Lichao Sun, and Siu-Ming Yiu. Subgraph federated learning with missing neighbor generation. In *Advances in Neural Information Processing Systems*, 2021a.

Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M. Alvarez. Personalized federated learning with first order model optimization. In *International Conference on Learning Representations*, 2021b.

Wentao Zhang, Ziqi Yin, Zeang Sheng, Yang Li, Wen Ouyang, Xiaosen Li, Yangyu Tao, Zhi Yang, and Bin Cui. Graph attention multi-layer perceptron. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 4560–4570, 2022.

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. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 5716–5724, 2024.

A Proofs and Additional Theory

A.1 Proof of Theorem 1

Let $Q_i(\theta)$ denote the population validation utility and let $\widehat{Q}_i(\theta)$ denote the observed validation utility. Under Assumption 1, for every checkpoint $\theta \in \mathcal{C}_i^t$,

$$\left|\widehat{Q}_i(\theta) - Q_i(\theta)\right| \le \xi_i.$$

Let $\omega_{i,\star}^t$ be the recoverable checkpoint with the largest observed validation utility before the candidate evaluation in round t, and let $\theta_{i,\star}^t$ denote its model-parameter component. Let $\tilde{\omega}_i^{t+1}$ be the candidate recoverable state before applying rollback, with model-parameter component $\tilde{\theta}_i^{t+1}$. Since the validation utility is computed from model predictions, the proof can be written in terms of the parameter component. The rollback operator accepts $\tilde{\omega}_i^{t+1}$ only if

$$\widehat{Q}_i(\tilde{\theta}_i^{t+1}) \ge \widehat{Q}_i(\theta_{i,\star}^t) - \epsilon.$$

If the candidate is rejected, the deployed recoverable state is exactly $\omega_{i,\star}^t$, whose parameter component is $\theta_{i,\star}^t$, so the regret with respect to $\theta_{i,\star}^t$ is zero. If the candidate is accepted, then

$$\begin{aligned} Q_i(\theta_{i,\star}^t) - Q_i(\tilde{\theta}_i^{t+1}) &\le \widehat{Q}_i(\theta_{i,\star}^t) + \xi_i - \left(\widehat{Q}_i(\tilde{\theta}_i^{t+1}) - \xi_i\right) \\ &= \widehat{Q}_i(\theta_{i,\star}^t) - \widehat{Q}_i(\tilde{\theta}_i^{t+1}) + 2\xi_i \\ &\le \epsilon + 2\xi_i. \end{aligned}$$

Thus the deployed post-rollback state has validation regret at most $\epsilon + 2\xi_i$ relative to the client's best observed checkpoint. This proves Theorem 1.

A.2 Finite-Candidate Validation Concentration

Corollary 1 (Finite-candidate validation concentration)**.** *Condition on the information used to construct the finite candidate set $\mathcal{C}_i^t$. Suppose the audit split consists of n_i independent observations $Z_{i,1}, \dots, Z_{i,n_i}$, independent of that construction process. For each fixed candidate, let $u_i(\theta, Z_{i,j}) \in [0,1]$, $\mathbb{E}[u_i(\theta, Z_{i,j})] = Q_i(\theta)$, and $\widehat{Q}_i(\theta) = n_i^{-1} \sum_{j=1}^{n_i} u_i(\theta, Z_{i,j})$. Then, with probability at least $1 - \delta_{\mathrm{conf}}$,*

$$\sup_{\theta \in \mathcal{C}_i^t} \left|\widehat{Q}_i(\theta) - Q_i(\theta)\right| \le \sqrt{\frac{\log(2|\mathcal{C}_i^t|/\delta_{\mathrm{conf}})}{2n_i}}.$$

Consequently, Theorem 1 holds with this value of ξ_i. Independence is required across audit observations for each fixed candidate; utilities of different candidates evaluated on the same audit split can be dependent.

Fix a client i and condition on the candidate construction process. For each $\theta \in \mathcal{C}_i^t$, the empirical utility is an average of independent bounded observations. Hoeffding's inequality gives, for any $s > 0$,

$$\Pr\Big(\Big|\widehat{Q}_i(\theta) - Q_i(\theta)\Big| > s\Big) \le 2\exp(-2n_i s^2).$$

Applying the union bound over the finite set $\mathcal{C}_i^t$ gives

$$\Pr\left(\sup_{\theta\in\mathcal{C}_i^t}\Big|\widehat{Q}_i(\theta) - Q_i(\theta)\Big| > s\right) \le 2|\mathcal{C}_i^t|\exp(-2n_i s^2).$$

Setting the right-hand side to δ_{conf} yields

$$s = \sqrt{\frac{\log(2|\mathcal{C}_i^t|/\delta_{\mathrm{conf}})}{2n_i}}.$$

The conditional bound holds for every candidate set satisfying these conditions, so averaging over its construction gives the stated probability guarantee. Assumption 1 follows with this value of ξ_i.

Graph dependence and evaluation metrics. The independent-observation specialization applies to additive utilities such as accuracy when its sampling conditions hold. Correlated graph observations require a concentration result for the specified dependence structure before an effective sample size can be assigned. AUC is a pairwise statistic whose summands share observations, so its concentration also requires a metric-specific argument. Theorem 1 is stated directly in terms of uniform utility concentration and accommodates either setting once the corresponding ξ_i is established.

A.3 Proof of Proposition 1

Assumption 2 states that population test utility and population validation utility are coupled over the finite candidate set:

$$U_i(\theta_a) - U_i(\theta_b) \ge Q_i(\theta_a) - Q_i(\theta_b) - \gamma_i$$

for all $\theta_a, \theta_b \in \mathcal{C}_i^t$. Apply this inequality with $\theta_a = \hat{\theta}_i^{t+1}$ and $\theta_b = \theta_{i,\star}^t$. By Theorem 1,

$$Q_i(\hat{\theta}_i^{t+1}) - Q_i(\theta_{i,\star}^t) \ge -(\epsilon + 2\xi_i).$$

Therefore,

$$U_i(\hat{\theta}_i^{t+1}) - U_i(\theta_{i,\star}^t) \ge -(\epsilon + 2\xi_i + \gamma_i).$$

Taking the negative part gives

$$\Big(U_i(\theta_{i,\star}^t) - U_i(\hat{\theta}_i^{t+1})\Big)_+ \le \epsilon + 2\xi_i + \gamma_i.$$

Averaging over clients on the intersection of their concentration events yields the stated downside-risk bound. If client i's event has failure probability at most δ_i, this intersection has probability at least $1 - \sum_{i=1}^M \delta_i$ by the union bound. This step permits dependence between clients, including that induced by overlapping partitions.

A.4 Rollback and Loss Masking at Revocation

Theorem 2 (Revocation separates rollback from loss masking)**.** *Suppose Assumption 1 holds and client i triggers revocation at round t:*

$$\widehat{Q}_i(\tilde{\theta}_i^{t+1}) < \widehat{Q}_i(\theta_{i,\star}^t) - \epsilon.$$

Let a loss-masking rule keep $\tilde{\theta}_i^{t+1}$ as the client state after setting the transfer loss to zero, while rollback deploys $\theta_{i,\star}^t$. Then, with high probability,

$$Q_i(\theta_{i,\star}^t) - Q_i(\tilde{\theta}_i^{t+1}) > \epsilon - 2\xi_i.$$

If Assumption 2 also holds, then the corresponding test-side gap satisfies

$$U_i(\theta_{i,\star}^t) - U_i(\tilde{\theta}_i^{t+1}) > \epsilon - 2\xi_i - \gamma_i.$$

At a revocation event,

$$\widehat{Q}_i(\tilde{\theta}_i^{t+1}) < \widehat{Q}_i(\theta_{i,\star}^t) - \epsilon.$$

Equivalently,

$$\widehat{Q}_i(\theta_{i,\star}^t) - \widehat{Q}_i(\tilde{\theta}_i^{t+1}) > \epsilon.$$

Under Assumption 1,

$$\begin{aligned} Q_i(\theta_{i,\star}^t) - Q_i(\tilde{\theta}_i^{t+1}) &\geq \widehat{Q}_i(\theta_{i,\star}^t) - \xi_i - \left(\widehat{Q}_i(\tilde{\theta}_i^{t+1}) + \xi_i\right) \\ &= \widehat{Q}_i(\theta_{i,\star}^t) - \widehat{Q}_i(\tilde{\theta}_i^{t+1}) - 2\xi_i \\ &> \epsilon - 2\xi_i. \end{aligned}$$

This proves the validation-side separation. If Assumption 2 also holds, applying it with $\theta_a = \theta_{i,\star}^t$ and $\theta_b = \tilde{\theta}_i^{t+1}$ gives

$$U_i(\theta_{i,\star}^t) - U_i(\tilde{\theta}_i^{t+1}) \geq Q_i(\theta_{i,\star}^t) - Q_i(\tilde{\theta}_i^{t+1}) - \gamma_i > \epsilon - 2\xi_i - \gamma_i.$$

The loss-masking rule keeps $\tilde{\theta}_i^{t+1}$ as the client state, whereas rollback restores $\theta_{i,\star}^t$. The population validation gap is strictly positive when $\epsilon \geq 2\xi_i$, and the test-side gap is strictly positive when $\epsilon \geq 2\xi_i + \gamma_i$. These conditions identify when the observed revocation margin is large enough to certify a utility advantage for restoration at that event.

A.5 Interpretation of the Recovery Bound

The reference-relative bound describes the cost of retaining a candidate after the validation check. Revocation restores the reference predictor exactly; acceptance permits a validation gap of at most ϵ, with population uncertainty contributing $2\xi_i$. Coupling adds γ_i when the comparison is expressed in test utility. These terms connect a concrete training operation to the quality of the evidence used to authorize it.

A.6 Adaptive Validation Use and Audit Interpretation

The guarantees require uniform concentration over the candidate states used by the client. Corollary 1 instantiates this condition for a candidate set fixed independently of the audit split. During training, checkpoint updates and gate decisions reuse the client validation split, so later candidates depend on earlier validation outcomes. In that regime, ξ_i must account for adaptive selection as well as sampling uncertainty; the independent-candidate formula alone does not establish its value.

The tolerance margin absorbs small observed fluctuations, and cooldown spaces out transfer attempts. Both affect the decision sequence, but neither establishes uniform concentration under adaptive reuse. An independent audit split provides the finite-candidate interpretation when candidates are fixed before auditing. Under the training protocol, Theorem 1 remains conditional on a concentration bound appropriate to the adaptive trajectory.

Independent checkpoint audit. An independent audit separates three roles: training labels optimize the models; routing-validation labels determine checkpoints, thresholds, and cooldown; audit labels are withheld from all of these decisions. Freeze the trained candidate/reference pairs and the analysis plan before accessing the audit labels. Evaluate both states on the same audit examples, report paired utility differences, and stratify results by dataset and client count. If audit feedback changes the method or selects runs, a fresh audit is needed for the revised comparison. This protocol specifies how to assess generalization of the adaptive rule; the reported experiments use the fixed validation split described in Appendix C.5.

For independently sampled audit examples and a fixed finite candidate set, Corollary 1 applies. Graph dependence and AUC require the metric-specific treatment stated after that corollary. A practical audit should report pre-decision candidate gaps, post-recovery gaps, and rejection indicators separately. For validation–test coupling, paired differences across the frozen states provide an empirical discrepancy diagnostic, with uncertainty intervals; those observed discrepancies do not by themselves identify a uniform population γ_i. Test-side quantities are computed only after the candidate pairs and decision rule have been fixed.

Algorithm 1 Risk-Reversible Graph Transfer (R2GT)

Require: clients $\{1, \ldots, M\}$, communication rounds T, local epochs E, warm-up t_{warm}, transfer schedule λ_t, rollback margin ϵ, cooldown length κ
Ensure: deployed client predictors $\{\hat{\theta}_i\}_{i=1}^{M}$

Initialize $\mathcal{B}^0$; each client sets $\omega_{i,\star}^0 \leftarrow \omega_i^0$, $q_{i,\star}^0 \leftarrow \widehat{Q}_i(\omega_i^0)$, and $c_i^0 \leftarrow 0$
for $t = 0, \ldots, T-1$ **do**
  Server broadcasts $\mathcal{B}^t$
  for all clients i in parallel **do**
    Set transfer eligibility $a_i^t \leftarrow \mathbb{I}[t \geq t_{\mathrm{warm}}]\mathbb{I}[c_i^t = 0]$
    Initialize candidate $\tilde{\omega}_i^{t+1} \leftarrow \omega_i^t$
    for $e = 1, \ldots, E$ **do**
      Update candidate using Eq. 7, with a_i^t fixed
    end for
    Evaluate $\tilde{q}_i^{t+1} \leftarrow \widehat{Q}_i(\tilde{\omega}_i^{t+1})$
    Set $g_i^{t+1} \leftarrow \mathbb{I}[\tilde{q}_i^{t+1} \geq q_{i,\star}^t - \epsilon]$
    Log $(\tilde{q}_i^{t+1}, q_{i,\star}^t, g_i^{t+1}, a_i^t)$ before recovery or checkpoint replacement
    Carry forward $(\omega_{i,\star}^{t+1}, q_{i,\star}^{t+1}) \leftarrow (\omega_{i,\star}^t, q_{i,\star}^t)$
    if $g_i^{t+1} = 0$ **then**
      $\omega_i^{t+1} \leftarrow \omega_{i,\star}^t$, $c_i^{t+1} \leftarrow \kappa$
    else
      $\omega_i^{t+1} \leftarrow \tilde{\omega}_i^{t+1}$, $c_i^{t+1} \leftarrow \max(c_i^t - 1, 0)$
      if $\tilde{q}_i^{t+1} > q_{i,\star}^t$ **then**
        $(\omega_{i,\star}^{t+1}, q_{i,\star}^{t+1}) \leftarrow (\tilde{\omega}_i^{t+1}, \tilde{q}_i^{t+1})$
      end if
    end if
    Compute and upload statistics from retained ω_i^{t+1}, with the admissibility summary
  end for
  Server updates $\mathcal{B}^{t+1}$ from uploaded statistics
end for
Each client deploys $\hat{\theta}_i$ from its validation-selected recoverable checkpoint set

B Algorithmic Procedure

Algorithm 1 summarizes server-client execution. The server receives transfer statistics and admissibility summaries. Raw graphs, labels, validation splits, and checkpoints remain local. Transfer eligibility follows Eq. 6; candidate acceptance follows Eq. 5; and restoration follows Eq. 10. The reference checkpoint and score are fixed before local optimization. Candidate validation is logged before either checkpoint replacement or recovery. A parameter checkpoint recovers the predictor; restoring optimizer and other stateful components also recovers the corresponding optimization state.

C Experimental Details

C.1 Evaluation Tasks

We study transductive node classification in federated graph learning. For a dataset graph $G = (V, E, X, Y)$, the nodes and edges are partitioned into M client subgraphs $\{G_m\}_{m=1}^{M}$, where client m observes only its local graph $G_m = (V_m, E_m, X_m, Y_m)$ and its local train, validation, and test masks. The server coordinates communication rounds and aggregates allowed model-side information, but raw node features, labels, and edges are not pooled across clients.

We evaluate four regimes: homophilic disjoint partitions, heterophilic disjoint partitions, homophilic overlapping partitions, and heterophilic overlapping partitions. Disjoint partitions test whether a method can transfer knowledge across structurally separated subgraphs, while overlapping partitions test whether it can exploit partially shared node neighborhoods without over-committing to harmful cross-client trajectories. The disjoint setting uses 5, 10, and 20 clients; the overlapping setting uses 10, 30, and 50 clients.

Table 3: Statistics of the eleven node-classification datasets used in the experiments.

Dataset	#Nodes	#Edges	#Classes	#Features
Homophilic graphs				
Cora	2,708	5,429	7	1,433
CiteSeer	3,327	4,732	6	3,703
PubMed	19,717	44,324	3	500
Amazon-Computer	13,752	491,722	10	767
Amazon-Photo	7,650	238,162	8	745
ogbn-arxiv	169,343	1,166,243	40	128
Heterophilic graphs				
Roman-empire	22,662	32,927	18	300
Amazon-ratings	24,492	93,050	5	300
Minesweeper	10,000	39,402	2	7
Tolokers	11,758	519,000	2	10
Questions	48,921	153,540	2	301

C.2 Datasets, Metrics, and Client Partitions

We use eleven public node-classification benchmarks. The homophilic group contains Cora, CiteSeer, PubMed, Amazon-Computer, Amazon-Photo, and ogbn-arxiv (Yang et al., 2016; Shchur et al., 2018; Hu et al., 2020). The heterophilic group contains Roman-empire, Amazon-ratings, Minesweeper, Tolokers, and Questions (Platonov et al., 2023). These datasets cover small citation graphs, medium-scale co-purchase graphs, a large citation graph, and graphs whose local neighborhoods are not reliably label-homophilic. Dataset statistics are summarized in Table 3.

For all datasets except ogbn-arxiv, we use a fixed node split with 20% of nodes for training, 40% for validation, and 40% for testing. For ogbn-arxiv, we use 5% training nodes, 47.5% validation nodes, and 47.5% test nodes, which is the large-graph split commonly used for subgraph federated learning (Baek et al., 2023; Yu et al., 2025). Accuracy is reported for Cora, CiteSeer, PubMed, Amazon-Computer, Amazon-Photo, ogbn-arxiv, Roman-empire, and Amazon-ratings; AUC is reported for Minesweeper, Tolokers, and Questions because these are binary classification benchmarks with class-imbalance sensitive evaluation (Platonov et al., 2023; Yu et al., 2025).

For disjoint client partitions, we apply graph partitioning to split each full graph into M non-overlapping subgraphs using METIS (Karypis & Kumar, 1997). For overlapping client partitions, we first produce $\lfloor M/5 \rfloor$ base subgraphs, then repeatedly sample half of the nodes and their associated edges from each base subgraph to form five overlapping client groups (Baek et al., 2023; Yu et al., 2025). This creates exactly M clients while allowing controlled overlap in local node neighborhoods. The same partitions and node masks are reused for all compared methods within each setting.

C.3 Model Architecture

R2GT uses a single encoder and classifier architecture across datasets and client counts, with transfer controlled through the gate, checkpoint, and cooldown state. Architecture details for R2GT and the baselines are given in Appendix C.5.

At round t, client m maintains a local recoverable state ω_m^t, including model parameters, transfer-side variables, validation statistics, and the best admissible checkpoint observed so far. Cross-client information enters candidate optimization when transfer is eligible; the validation acceptance test determines whether that candidate is retained. If the candidate violates the admissibility margin, the client rolls back to its best recoverable checkpoint instead of merely masking the current transfer loss.

C.4 Baselines

The baselines cover local training, federated model aggregation, personalization, graph-specific collaboration, and heterogeneity-aware sharing.

Local. Each client trains a model using only its own subgraph, without server aggregation or cross-client information exchange. It measures the contribution of cross-client collaboration relative to local graph evidence.

FedAvg (McMahan et al., 2017). FedAvg is the canonical federated optimization baseline. Clients train local models and periodically upload model parameters or updates to the server, which averages them and broadcasts the global model back to all clients. It tests whether simple parameter sharing is sufficient under graph heterogeneity.

FedProx (Li et al., 2020). FedProx extends FedAvg by adding a proximal penalty that discourages local models from drifting too far from the current global model. It is designed for systems and statistical heterogeneity, and serves as a stronger generic FL baseline when local graph distributions differ.

FedPer (Arivazhagan et al., 2019). FedPer separates shared representation layers from personalized prediction layers. The backbone is federated across clients, while the classifier remains client-specific. This baseline tests whether personalization at the classifier level is enough to handle client-specific graph labels and neighborhoods.

GCFL (Xie et al., 2021). GCFL clusters clients according to gradient similarity and performs aggregation within each cluster. It represents clustered federated learning for graph data: clients with compatible optimization trajectories share a cluster model.

FedGNN (Wu et al., 2021). FedGNN exchanges node embeddings across clients to enrich local subgraphs while preserving raw graph data. It evaluates whether representation-level neighbor sharing can compensate for missing cross-client graph context.

FedSage+ (Zhang et al., 2021a). FedSage+ addresses subgraph incompleteness by learning a missing-neighbor generator. Clients use received node representations and feature-distance signals to reconstruct missing cross-client neighborhood information, providing a graph-structure completion baseline.

FED-PUB (Baek et al., 2023). FED-PUB performs personalized subgraph federated learning by estimating inter-client similarity and using it to weight model aggregation. It further learns sparse personalized masks so that each client updates only the portions of the aggregated model that are relevant to its local subgraph.

FedGTA (Li et al., 2023). FedGTA is a topology-aware aggregation method. Clients compute local smoothing confidence and moments of neighborhood features, upload these topology-aware statistics together with model parameters, and receive client-specific aggregation weights from the server.

AdaFGL (Li et al., 2024). AdaFGL adopts a decoupled personalization strategy for federated node classification with topology heterogeneity. It first obtains a global knowledge extractor through federated training and then performs personalized training on each client's local subgraph.

FedTAD (Zhu et al., 2024). FedTAD introduces topology-aware data-free knowledge distillation. The server uses a generator to synthesize pseudo graphs and distills information from local models to a global model, aiming to transfer knowledge without accessing raw client graph data.

FedIIH (Yu et al., 2025). FedIIH models both inter-client and intra-client heterogeneity. It uses hierarchical variational inference to estimate inter-subgraph similarity from graph data distributions and disentangles each subgraph into multiple latent factors for finer-grained personalization.

FedSSA (Yu et al., 2026). FedSSA shares semantic and structural knowledge through separate alignment mechanisms. It infers class-wise node distributions for semantic clustering and uses spectral energy to characterize structural similarity. Its two alignment mechanisms target node-feature and graph-structure heterogeneity.

For methods with public implementations, we keep their prescribed algorithmic components and tune only standard optimization hyperparameters on validation sets. For methods sharing the same graph learner family, encoder and classifier dimensions are aligned. All reported R2GT comparisons use the same train, validation, and test masks as the corresponding baseline run.

C.5 Implementation and Reporting

All reported experiments were conducted on Linux servers equipped with NVIDIA 6000D GPUs. The implementation uses PyTorch and PyTorch Geometric, and all method comparisons are executed under the same dataset split, client partition, and communication schedule for a given setting.

Every setting is repeated with five random seeds. Hyperparameters are selected using validation performance only; test metrics are used once for the final reporting of the selected run. When a method has multiple admissible candidate checkpoints, selection is driven by validation utility at the client level and then aggregated across clients for the reported setting.

Communication budgets are fixed per dataset setting. The default budget is 100 communication rounds, while larger product-scale and ogbn-arxiv settings use the longer released schedule of 200 communication rounds. The number of local optimization steps, optimizer family, hidden dimension, dropout, and weight decay are held consistent within a setting unless a baseline's original implementation requires a method-specific auxiliary module.

For baseline architectures, methods without a specialized graph module use a two-layer GCN (Kipf & Welling, 2017) followed by a linear classifier. FedSage+ uses a GraphSAGE encoder (Hamilton et al., 2017) together with its missing-neighbor generator; FedGTA uses a graph attention multilayer perceptron backbone (Zhang et al., 2022); and FedIIH uses a DisenGCN-style node-feature projection layer (Ma et al., 2019) followed by an MLP classifier. FedSSA and R2GT use the same spectral graph learner family based on universal polynomial filtering (Huang et al., 2024), followed by the same MLP classifier. R2GT couples this learner with its transfer-control state and rollback rule.

R2GT's transfer-control hyperparameters are the rollback margin, transfer strength, warm-up interval, and cooldown length. Small validation grids select their values for each setting.

Tables report mean $\pm$ standard deviation over the five seeds. Higher values are better in all tables; the best result is shown in bold and the second-best result is underlined. The average column is computed over all dataset-client settings shown in the table, summarizing the evaluation grid displayed in that table.

C.6 Overlapping-Partition Results

Homophilic Datasets. Table 4 reports results under overlapping client partitions. R2GT increases the average score from 77.57 for FedSSA to 78.71. Overlap supplies partially shared neighborhood evidence, yet the gain indicates that client-specific admission and recovery remain useful in this regime. Together with the disjoint results, this extends the empirical support across both forms of client partitioning.

Heterophilic Datasets. Table 5 combines shared neighborhoods with heterophilic labels, so overlap alone gives an incomplete indication of transfer utility. R2GT increases the average score from 60.50 for FedSSA to 61.20. The improvement complements the disjoint heterophilic results and supports applying the same recovery mechanism across different neighborhood and partition structures.

C.7 Additional Ablation Results

Figures 4 and 5 expand the ablation study to the remaining datasets and client partitions not shown in the main text. Together with Figure 3, they cover the full evaluation grid in Tables 1–5. The same pattern as in Section 5.3 remains the organizing trend: removing the transfer term produces the largest drop, removing rollback is more harmful than removing cooldown.

C.8 Rollback-Event Protocol and Diagnostics

The structural-effectiveness study uses Cora O-30, CiteSeer D-10, Minesweeper D-10, and Questions O-50. FedSSA+CKPT applies the same validation-based model selection used throughout the evaluation. Mask only retains the risk gate but continues from the transfer-influenced state after a rejection. Random rollback matches the number of R2GT rollbacks within each setting and seed

Table 4: Results on homophilic graph datasets under overlapping subgraph partitioning.

	Cora			CiteSeer			PubMed			–
Methods	10 Clients	30 Clients	50 Clients	10 Clients	30 Clients	50 Clients	10 Clients	30 Clients	50 Clients	–
Local	67.52±0.25	64.46±0.12	73.57±0.10	59.15±0.08	58.51±0.42	63.28±0.44	79.53±0.07	76.96±0.16	76.93±0.11	–
FedAvg	70.02±0.36	46.80±0.98	50.93±4.53	63.51±0.15	60.12±0.64	63.11±1.00	79.88±0.11	78.29±0.12	76.63±0.35	–
FedProx	71.39±0.50	44.19±1.74	53.21±9.04	63.42±0.35	60.08±0.75	63.13±0.43	79.84±0.17	78.37±0.13	76.89±0.46	–
FedPer	72.27±0.31	66.99±0.24	71.36±0.37	63.84±0.28	59.16±0.81	64.24±0.44	82.52±0.06	80.59±0.38	80.33±0.10	–
GCFL	72.38±0.26	66.22±0.27	73.57±0.16	63.51±0.39	58.89±0.18	62.58±0.30	80.80±0.25	77.01±0.12	77.75±0.11	–
FedGNN	64.17±0.83	54.19±2.33	53.85±0.82	62.75±0.39	53.95±1.52	55.58±0.98	81.46±0.07	78.26±0.22	78.24±0.10	–
FedSage+	71.06±0.46	44.80±0.42	52.42±11.5	62.78±0.48	59.94±0.02	62.53±0.30	79.98±0.08	78.38±0.11	76.70±0.68	–
FED-PUB	73.14±0.12	68.21±0.54	74.78±0.23	64.61±0.20	62.30±0.45	65.81±0.30	82.91±0.08	81.40±0.10	81.23±0.12	–
FedGTA	69.96±0.62	68.44±0.33	74.63±0.14	64.46±0.08	65.68±0.33	65.79±0.32	82.55±0.42	81.23±0.05	80.86±0.06	–
AdaFGL	72.04±0.19	68.61±0.23	71.35±0.00	66.66±0.15	62.15±0.31	59.50±0.75	82.79±0.23	82.09±0.41	80.84±0.07	–
FedTAD	72.83±0.78	53.73±2.17	65.02±0.44	67.50±0.16	61.71±0.57	60.11±0.68	80.19±0.20	78.35±0.15	78.02±0.19	–
FedIIH	74.11±0.23	69.63±0.24	75.52±0.25	67.19±0.18	66.24±0.21	66.16±0.11	83.08±0.03	82.89±0.11	81.55±0.12	–
FedSSA	74.56±0.09	69.70±0.06	75.52±0.15	66.22±0.08	66.35±0.13	66.28±0.10	83.61±0.09	83.07±0.06	81.59±0.04	–
R2GT (ours)	**75.82**±0.05	**72.14**±0.07	**75.69**±0.11	**68.62**±0.07	**66.77**±0.08	**67.16**±0.08	**84.72**±0.07	**83.08**±0.08	**81.66**±0.02	–
	Amazon-Computer			Amazon-Photo			ogbn-arxiv			Avg.
Methods	10 Clients	30 Clients	50 Clients	10 Clients	30 Clients	50 Clients	10 Clients	30 Clients	50 Clients	All
Local	86.28±0.20	83.89±0.00	84.22±0.02	90.96±0.12	88.47±0.12	88.84±0.15	59.18±0.07	58.48±0.04	57.33±0.04	73.20
FedAvg	86.77±0.19	80.60±0.47	73.52±0.12	91.70±0.07	87.61±0.22	72.61±0.57	60.22±0.02	56.88±0.06	58.23±0.24	69.86
FedProx	86.62±0.20	81.07±0.89	73.78±0.47	91.46±0.19	87.48±0.40	70.78±2.06	60.18±0.11	57.02±0.16	58.41±0.04	69.85
FedPer	87.08±0.04	85.22±0.23	85.40±0.27	91.67±0.24	89.54±0.16	89.34±0.38	60.63±0.08	59.45±0.04	58.53±0.11	74.90
GCFL	86.79±0.22	84.47±0.09	84.20±0.22	91.24±0.10	88.89±0.11	88.96±0.08	59.90±0.02	58.82±0.10	57.61±0.01	74.09
FedGNN	85.93±0.09	84.23±0.10	81.14±0.88	90.26±0.11	86.22±1.34	77.32±6.46	59.74±0.19	57.25±0.04	57.80±0.11	70.13
FedSage+	87.02±0.15	78.56±1.20	73.90±0.39	91.55±0.05	87.00±0.99	70.83±1.36	59.90±0.02	57.06±0.12	58.24±0.09	69.59
FED-PUB	87.76±0.08	86.38±0.06	85.94±0.14	92.01±0.07	90.32±0.07	90.13±0.11	60.84±0.04	60.50±0.12	59.84±0.12	76.01
FedGTA	87.88±0.18	86.02±0.27	85.33±0.21	91.92±0.14	90.80±0.06	90.19±0.06	52.64±0.09	53.92±0.07	55.18±0.09	74.86
AdaFGL	78.27±0.00	77.65±0.00	79.30±0.00	88.03±0.00	86.65±0.00	86.10±0.00	53.47±0.06	52.33±0.00	52.11±0.00	72.22
FedTAD	76.87±5.63	76.71±0.85	74.23±0.07	80.73±3.09	84.89±1.75	82.80±1.33	55.11±0.15	54.91±0.54	53.81±0.14	69.86
FedIIH	87.93±0.04	86.79±0.19	86.90±0.00	92.17±0.00	92.48±0.04	91.67±0.16	63.35±0.09	63.26±0.03	62.96±0.06	77.44
FedSSA	88.19±0.07	86.88±0.12	86.52±0.11	92.51±0.15	92.75±0.11	91.78±0.14	64.10±0.12	63.54±0.13	63.02±0.07	77.57
R2GT (ours)	**89.02**±0.05	**87.42**±0.07	**86.97**±0.10	**92.74**±0.19	**93.09**±0.08	**92.80**±0.20	**66.94**±0.16	**66.55**±0.11	**65.67**±0.09	**78.71**

Table 5: Results on heterophilic graph datasets under overlapping subgraph partitioning. Accuracy is reported for Roman-empire and Amazon-ratings; AUC is reported for Minesweeper, Tolokers, and Questions.

	Roman-empire			Amazon-ratings			Minesweeper			–
Methods	10 Clients	30 Clients	50 Clients	10 Clients	30 Clients	50 Clients	10 Clients	30 Clients	50 Clients	–
Local	33.82±0.03	24.22±0.14	23.35±0.18	39.06±0.04	40.39±0.14	41.46±0.12	65.45±0.07	61.75±0.10	60.81±0.23	–
FedAvg	35.24±0.25	28.45±0.08	28.78±0.20	37.49±0.06	39.98±0.02	39.91±0.16	66.53±0.07	65.31±0.04	64.97±0.08	–
FedProx	30.98±0.14	25.10±0.17	25.68±0.59	35.16±0.09	36.01±0.08	36.29±0.07	65.74±0.05	64.11±0.19	64.11±0.16	–
FedPer	18.01±3.27	13.06±3.09	14.30±3.58	29.96±4.23	30.16±0.54	33.37±2.25	59.32±1.02	57.49±1.38	58.14±3.61	–
GCFL	34.32±0.89	28.42±0.49	28.94±0.31	37.17±0.41	40.70±0.11	40.16±0.22	66.63±0.13	65.38±0.10	65.01±1.34	–
FedGNN	31.81±0.12	26.26±0.24	26.99±0.26	34.21±0.16	35.35±0.12	35.84±0.15	66.06±0.21	64.66±0.17	64.49±0.23	–
FedSage+	51.83±0.00	32.34±0.00	29.20±0.00	34.49±0.00	35.29±0.00	35.92±0.00	74.00±0.00	67.92±0.00	68.42±0.00	–
FED-PUB	38.15±0.25	30.25±0.16	29.80±0.09	39.88±0.25	40.88±0.06	41.77±0.34	66.58±0.13	65.12±0.24	65.60±0.14	–
FedGTA	54.21±0.04	48.11±0.09	49.64±0.21	38.44±0.24	38.02±0.06	38.26±0.04	68.11±0.40	65.35±1.60	65.28±1.35	–
AdaFGL	58.79±0.03	51.56±0.02	51.62±0.01	37.02±0.05	39.77±0.15	39.60±0.25	66.54±0.72	65.70±1.82	64.88±1.31	–
FedTAD	38.49±0.13	31.73±0.18	32.89±0.01	37.16±0.17	39.27±0.13	39.47±0.26	66.74±0.33	65.79±0.05	65.31±0.08	–
FedIIH	59.83±0.12	53.11±0.06	54.49±0.10	40.26±0.02	40.98±0.05	41.54±0.21	66.82±0.25	65.45±0.26	65.45±0.14	–
FedSSA	60.01±0.07	53.59±0.09	54.82±0.14	40.46±0.06	41.10±0.10	41.86±0.15	73.12±0.11	69.96±0.06	69.39±0.05	–
R2GT (ours)	**60.09**±0.09	**55.38**±0.08	**56.37**±0.12	**41.09**±0.07	**41.18**±0.14	**42.01**±0.22	**74.05**±0.08	**70.83**±0.05	**70.19**±0.03	–
	Tolokers			Questions			Completed Avg.			
Methods	10 Clients	30 Clients	50 Clients	10 Clients	30 Clients	50 Clients	10 Clients	30 Clients	50 Clients	All
Local	71.59±0.03	68.67±0.31	66.00±0.20	59.56±0.02	53.67±0.08	51.21±0.02	53.90	49.74	48.57	50.73
FedAvg	70.75±0.40	58.17±0.27	54.84±0.42	59.19±0.63	55.38±0.18	55.86±0.27	53.84	49.46	48.87	50.72
FedProx	52.25±1.69	45.25±0.41	40.86±0.45	48.92±0.34	48.04±0.41	45.80±0.40	46.61	43.70	42.55	44.29
FedPer	37.36±0.11	59.10±0.79	41.29±0.06	57.70±0.29	49.91±1.53	45.37±0.10	40.47	41.94	38.49	40.30
GCFL	68.37±0.55	59.38±0.50	57.01±0.71	59.23±0.60	55.96±0.68	55.32±0.41	53.14	49.97	49.29	50.80
FedGNN	53.97±1.20	46.51±0.31	41.55±0.45	49.64±0.15	48.40±0.15	46.30±0.14	47.14	44.24	43.03	44.80
FedSage+	72.20±0.00	70.54±0.00	68.98±0.00	60.61±0.00	61.84±0.00	57.84±0.00	58.63	53.59	52.07	54.76
FED-PUB	71.93±0.29	70.01±0.54	66.17±0.85	61.78±2.44	54.88±1.19	55.81±0.74	55.66	52.23	51.83	53.24
FedGTA	68.10±1.53	59.25±1.10	55.49±1.48	59.59±1.27	54.61±1.06	56.07±0.77	57.69	53.07	52.95	54.57
AdaFGL	67.77±1.91	58.60±1.11	55.62±1.35	58.29±0.80	55.43±2.18	55.76±0.63	57.68	54.21	53.50	55.13
FedTAD	67.10±1.26	61.77±0.27	55.76±0.52	58.35±0.54	55.74±0.36	55.32±0.92	53.57	50.86	49.75	51.39
FedIIH	69.43±0.02	71.35±0.12	69.36±0.03	65.18±0.09	62.90±0.04	59.81±0.35	60.30	58.76	58.13	59.06
FedSSA	72.09±0.09	71.93±0.07	70.40±0.09	65.78±0.05	62.93±0.13	60.02±0.12	62.29	59.90	59.30	60.50
R2GT (ours)	**72.29**±0.13	**72.42**±0.09	**70.84**±0.12	**66.91**±0.03	**62.96**±0.15	**61.38**±0.18	**62.89**	**60.55**	**60.16**	**61.20**

while randomizing their timing. All variants share the data partition, initialization, client schedule, and training budget.

For an executed rollback event $e = (s, i, t)$, we copy the same pre-decision state into rollback and mask-only branches. The two branches reuse subsequent client sampling, data order, and optimization randomness. With M denoting the task metric, immediate and next-round rescue are

$$\Delta_e^{\mathrm{imm}} = M_{i,t}^{\mathrm{rollback}} - M_{i,t}^{\mathrm{mask}}, \qquad \Delta_e^{\mathrm{next}} = M_{i,t+1}^{\mathrm{rollback}} - M_{i,t+1}^{\mathrm{mask}}.$$

The test metric is recorded for this offline paired analysis; the online gate and checkpoint updates continue to use validation utility.

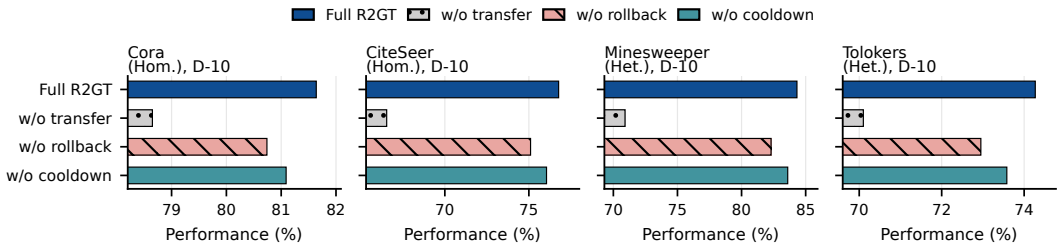


Figure 3: Ablation study of R2GT components on representative homophilic and heterophilic settings.

Table 6: Rollback activity over five seeds. Decision, rollback, and checkpoint-update counts are per-seed means; checkpoint age is measured in communication rounds.

Setting	Decisions	Rollbacks	Rate (%)	Updates	Mean age
Cora O-30	2400	45.0	1.88	1668.4	5.5
CiteSeer D-10	800	20.0	2.50	545.0	6.2
Minesweeper D-10	800	35.0	4.38	510.0	4.8
Questions O-50	4000	60.0	1.50	2780.0	7.1

Let Z denote either the semantic statistic S or structural statistic G, and let $Z_{i,t}^{\text{safe}}$ be its value in the stored safe state. The next-round normalized drift for branch b is

$$D_{e,b}^{Z} = \frac{\|Z_{i,t+1}^{b} - Z_{i,t}^{\text{safe}}\|}{\|Z_{i,t}^{\text{safe}}\| + \varepsilon}, \quad b \in \{\text{rollback}, \text{mask}\},$$

using the Euclidean norm for semantic statistics and the Frobenius norm for structural statistics. Figure 2c reports the reduction in mean drift,

$$100 \times \frac{\overline{D}_{\text{mask}}^{Z} - \overline{D}_{\text{rollback}}^{Z}}{\overline{D}_{\text{mask}}^{Z}}.$$

Table 6 summarizes the activity of the recovery policy. Rollbacks occur in every seed of every setting. Their frequency ranges from 1.50% to 4.38% of eligible transfer decisions, providing 100–300 paired events per setting while leaving most accepted training steps unchanged.

C.9 Additional Convergence Curves

Figure 6 presents representative homophilic and heterophilic settings. Figures 7–10 extend the comparison to the remaining settings. Each panel includes all compared methods and follows the same visual convention, showing how performance differences develop over communication rounds across partition types and graph homophily regimes.

C.10 Additional Hyperparameter Sensitivity

Figures 11 and 12 report sensitivity to transfer strength, rollback margin, warm-up, and cooldown across the evaluated settings. Values are expressed as percentage-point differences from the full-model reference configuration, marked by a black outline.

C.11 Case Study: Validation-Risk Audit

We audit saved transfer-training trajectories on Cora O-30 and Minesweeper D-10. Each client validation score is measured immediately after its local training epochs, before round-end transfer-control decisions or recovery. The lower panels of Figure 13 compare these pre-decision scores with their running historical maxima. With $\tilde{q}_i^s$ denoting the logged score at displayed round s, the plotted fraction is

$$v_s = \frac{1}{M} \sum_{i=1}^{M} \mathbb{I}\left[\tilde{q}_i^s < \max_{1 \le u \le s} \tilde{q}_i^u - \epsilon\right].$$

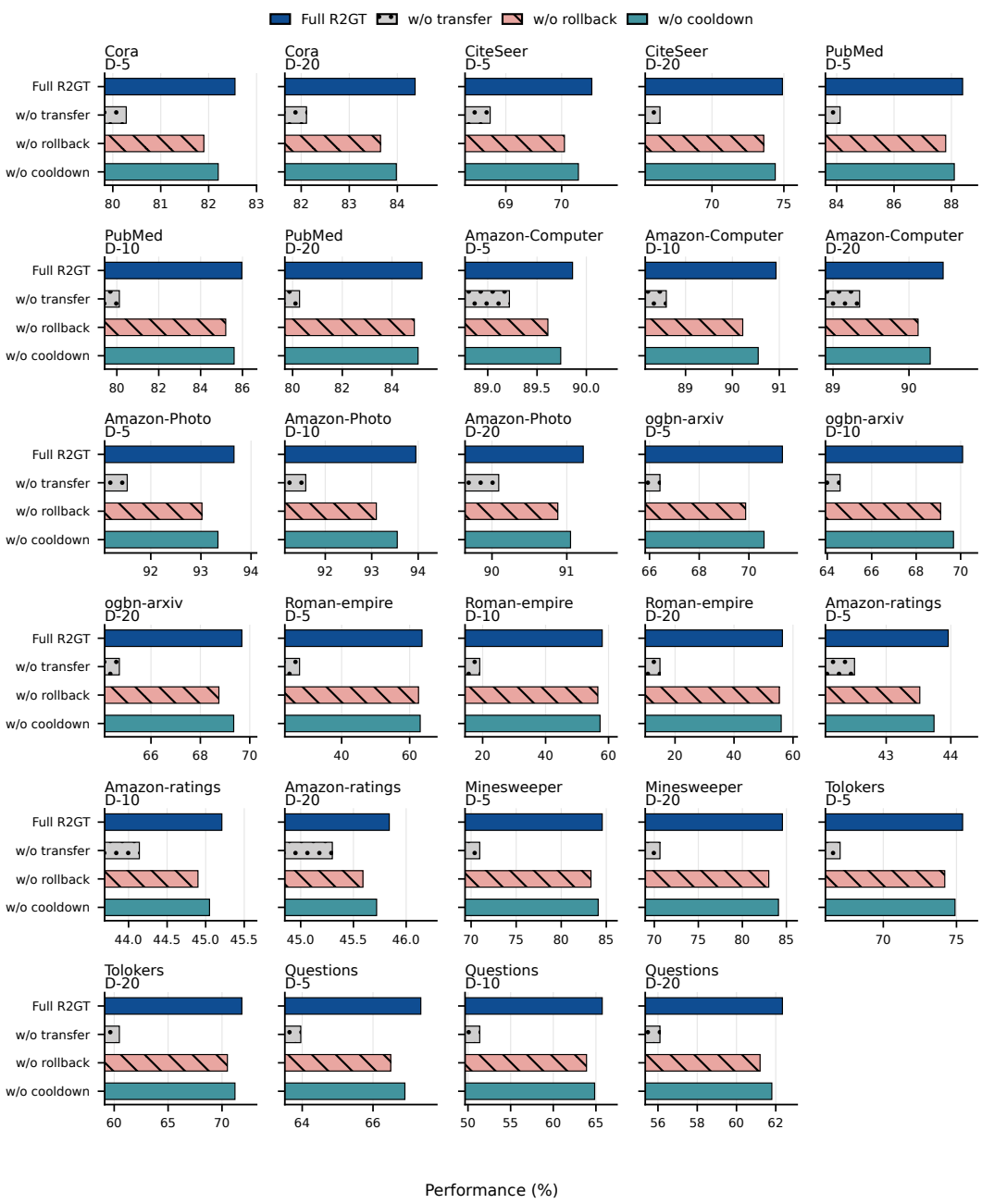


Figure 4: Additional ablation results for non-overlapping client partitions. "D" denotes disjoint client partitioning.

The denominator includes all clients, including those whose transfer is inactive. The audit uses $\epsilon = 0.002$ on the unit-scale validation metric and displays zero for the first 20 warm-up rounds. This view traces exposure to risky candidates across training, while Appendix C.8 measures executed rollbacks and their post-recovery branches.

The upper panels report mean client test performance of the best-validation checkpoints selected up to each round. These scores are recorded when each checkpoint is selected and carried forward until replacement. The resulting trajectories separate the quality of retained states from the pre-decision drawdown of current candidates. Stable checkpoint performance alongside frequent candidate drawdown illustrates the recurring role of admission and recovery during transfer training.

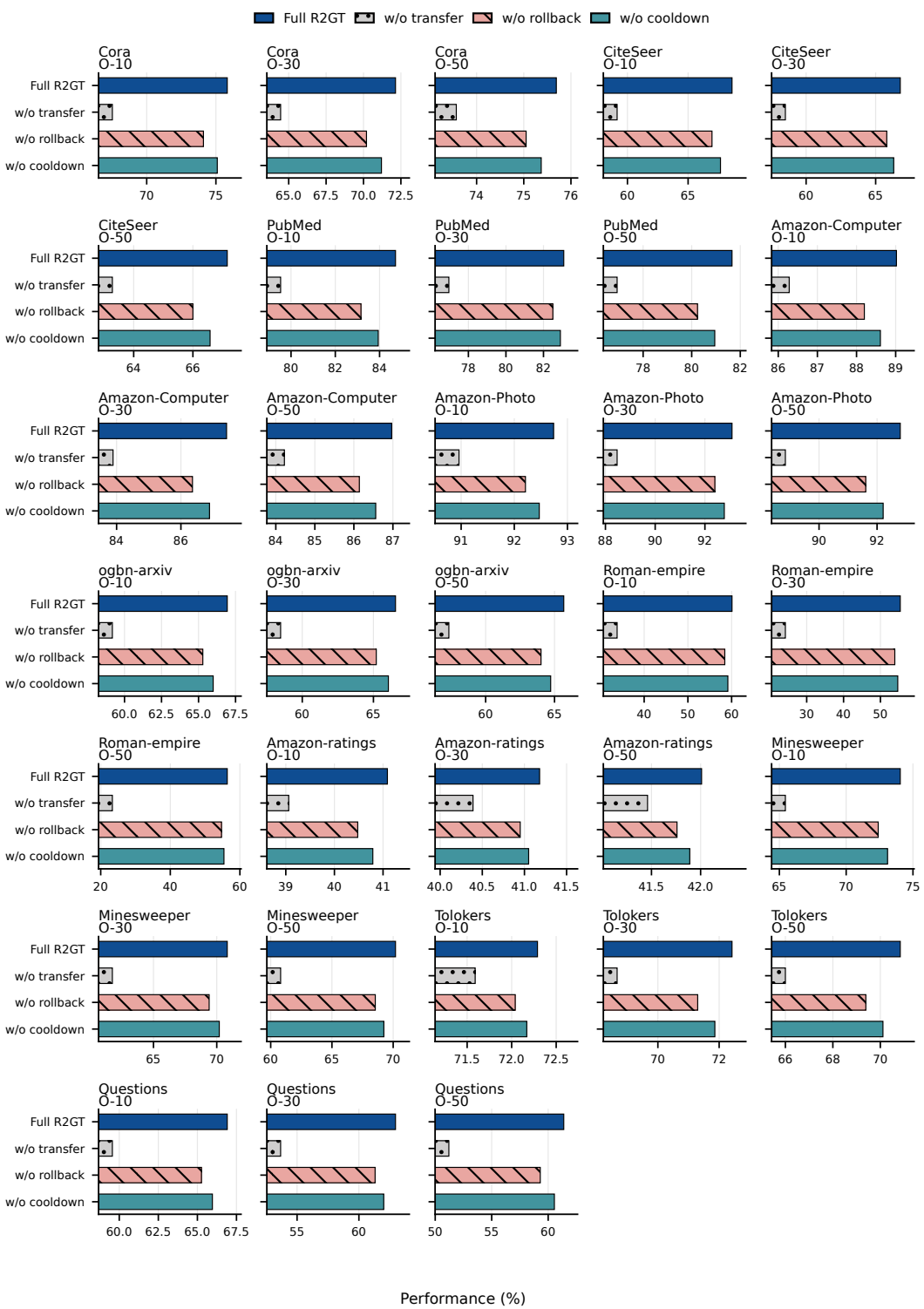


Figure 5: Additional ablation results for overlapping client partitions. "O" denotes overlapping client partitioning.

C.12 Computational Cost

Figures 14–17 report computational cost in seconds per communication round across homophilic and heterophilic datasets under disjoint and overlapping client partitions.

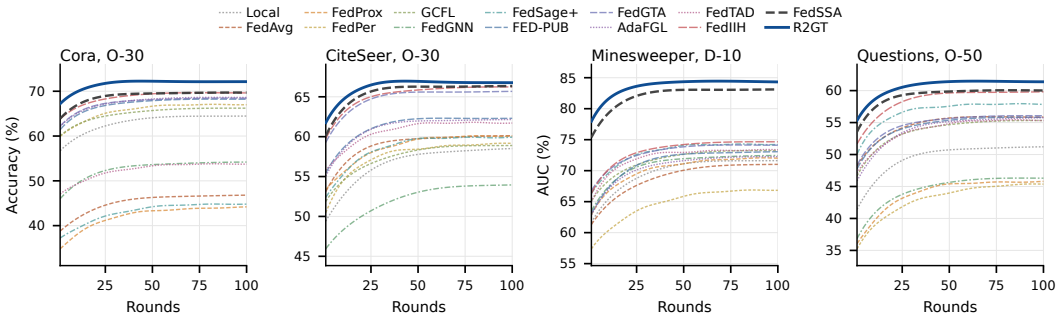


Figure 6: Convergence curves of all compared methods on representative homophilic and heterophilic settings. "D" and "O" denote disjoint and overlapping partitions.

Table 7: Score gain and per-round cost relative to FedSSA. Gains average setting-wise differences in percentage points; costs are median R2GT/FedSSA ratios over the same settings.

Dataset group / partition	Settings	Gain (pp)	Cost ratio
Homophilic / disjoint	18	0.62	1.01
Homophilic / overlapping	18	1.15	1.01
Heterophilic / disjoint	15	1.11	1.01
Heterophilic / overlapping	15	0.70	1.00

C.13 Storage Overhead

Rollback requires each client to keep a recoverable local state in addition to the current training state. Let P denote the number of trainable model parameters, K_{ckpt} the number of stored recoverable checkpoints, and $\mathrm{Mem}(\mathcal{B})$ the memory used by transfer-bank statistics. The client-side storage overhead can be summarized as

$$\mathrm{Mem}_{\mathrm{R2GT}} = \mathrm{Mem}_{\mathrm{current}} + K_{\mathrm{ckpt}}\mathrm{Mem}_{\mathrm{state}} + \mathrm{Mem}(\mathcal{B}_{\mathrm{local}}).$$

In our implementation, $K_{\mathrm{ckpt}} = 1$: each client stores its best recoverable checkpoint rather than a long checkpoint history. The default parameter-only checkpoint has size $O(P)$. If exact training continuation is required after rollback, optimizer moments and normalization or scheduler states can optionally be stored; for Adam-style optimizers, the two moment buffers add approximately $2P$ scalar values. These states remain local to the client and do not include raw node features, edges, labels, or neighbor lists.

The additional storage is determined by the size and contents of one checkpoint and remains independent of the communication horizon. Parameter-only recovery stores the reference predictor; full-state recovery additionally preserves the optimizer and other stateful components needed to resume from the same training state.

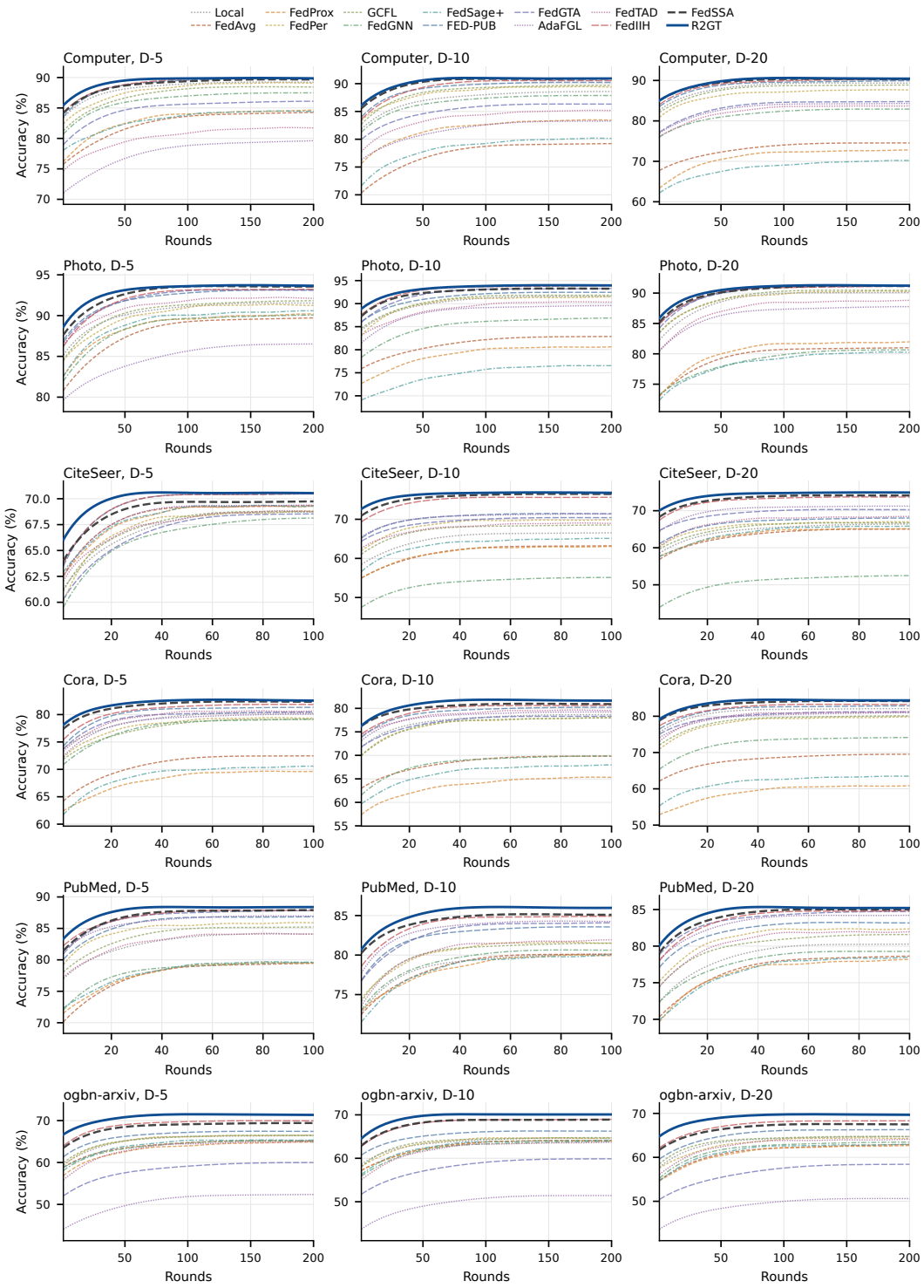


Figure 7: Additional convergence curves on homophilic graphs under disjoint client partitions. "D" denotes disjoint client partitioning.

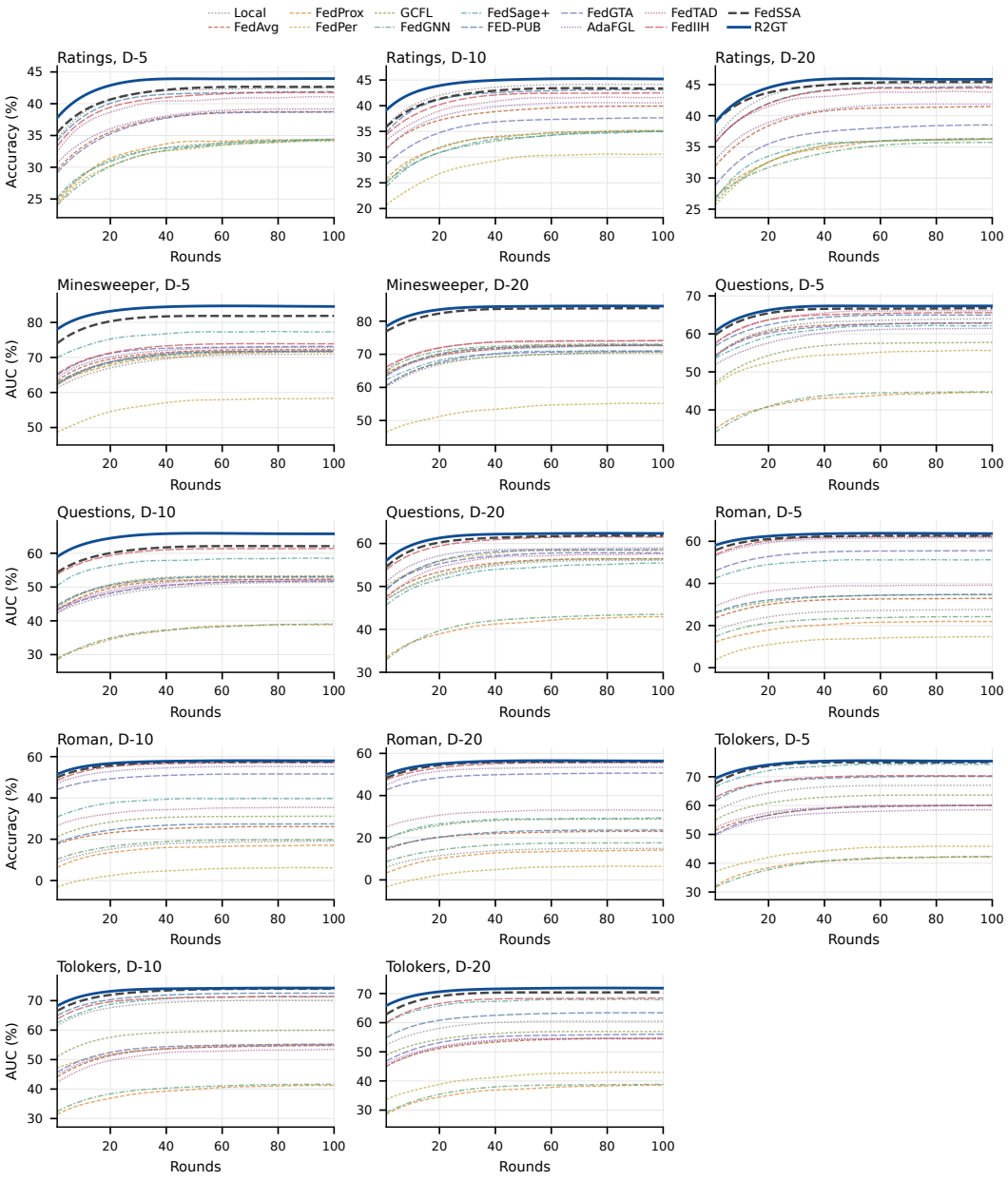


Figure 8: Additional convergence curves on heterophilic graphs under disjoint client partitions. "D" denotes disjoint client partitioning.

Table 8: Client-side storage components introduced by R2GT. P denotes the number of trainable parameters and d denotes the hidden dimension.

Stored component	Required for rollback	Size order
Model parameters	Yes	$O(P)$
Optimizer moments	Optional	$O(P)$ per moment buffer
Normalization or scheduler state	Optional	$O(d)$ or smaller
Local transfer-statistic cache	Optional	Statistic-dependent
Raw graph data	No	Not stored by R2GT checkpoints

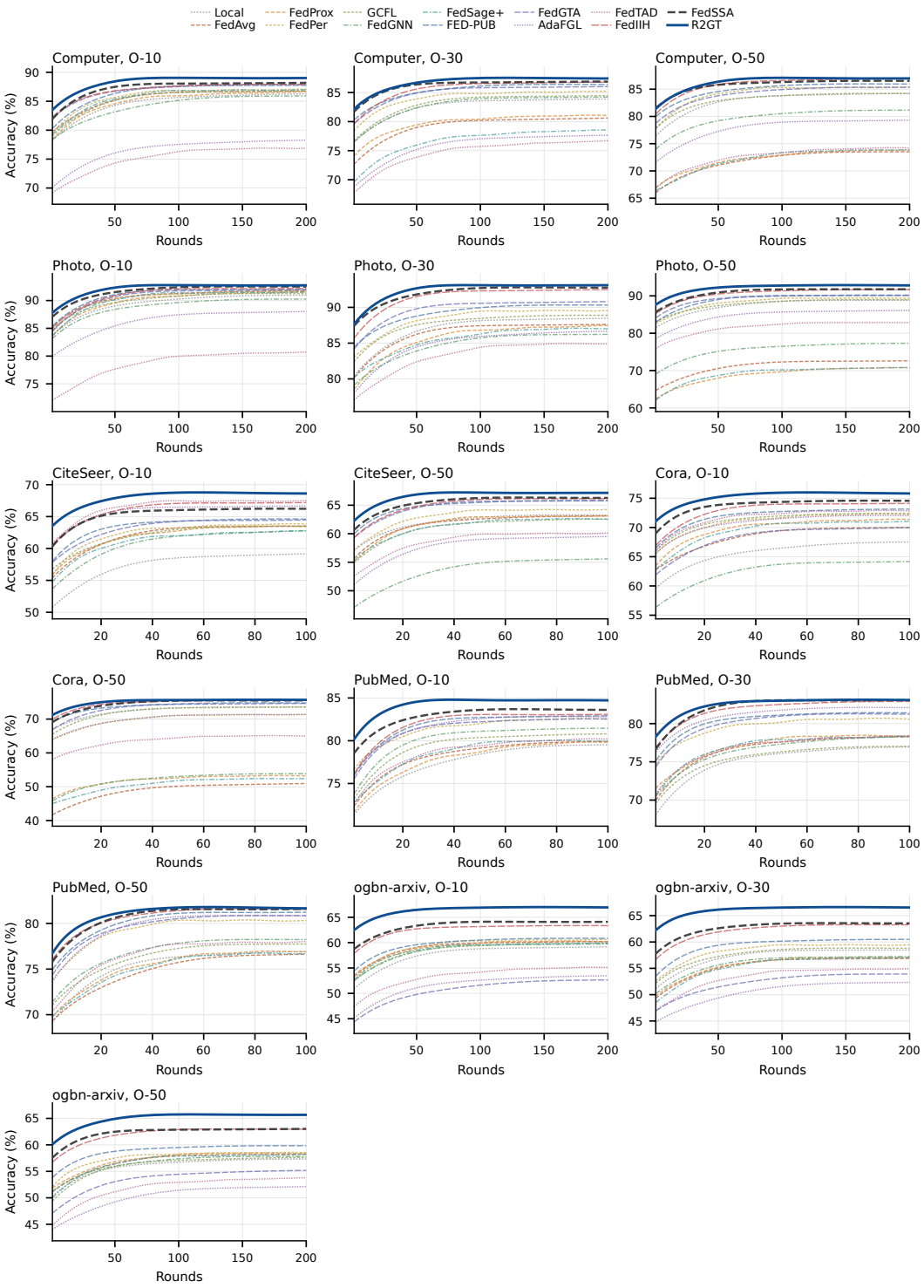


Figure 9: Additional convergence curves on homophilic graphs under overlapping client partitions. "O" denotes overlapping client partitioning.

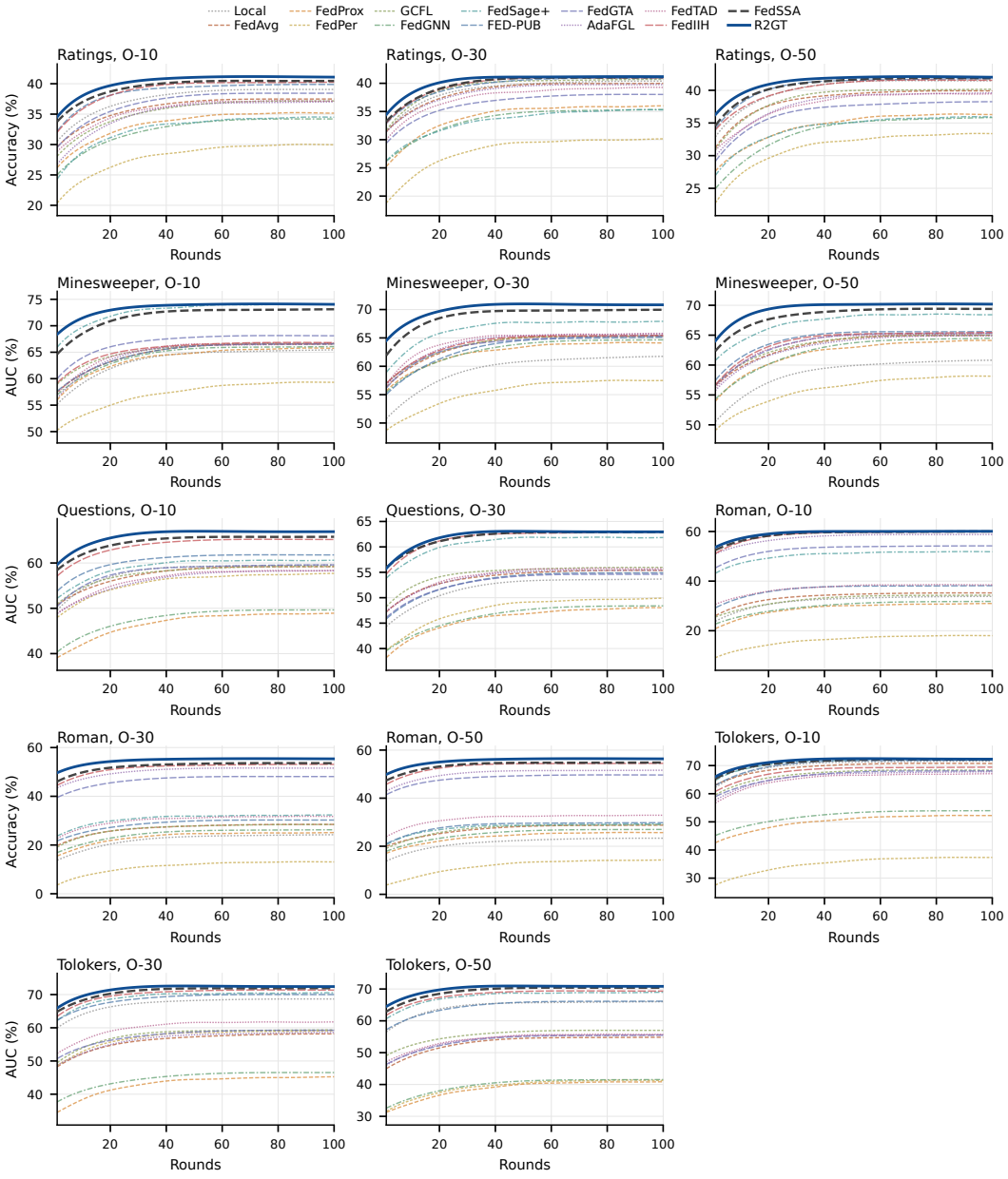


Figure 10: Additional convergence curves on heterophilic graphs under overlapping client partitions. "O" denotes overlapping client partitioning.

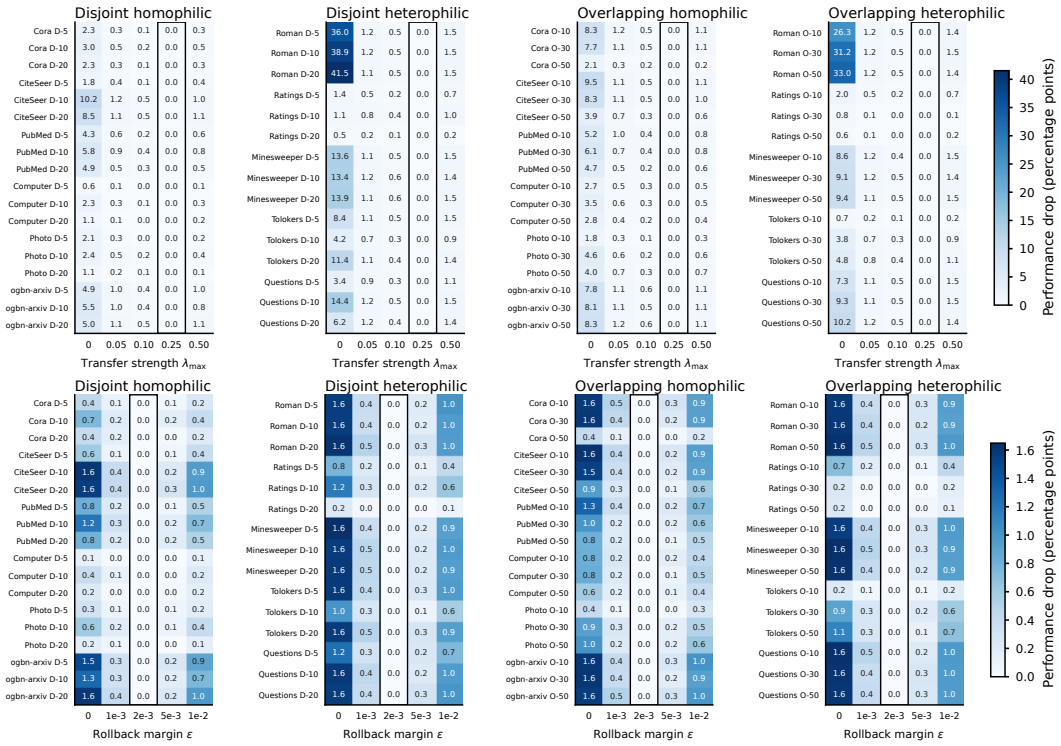


Figure 11: Hyperparameter sensitivity for transfer strength $\lambda_{\max}$ and rollback margin ϵ. Values are percentage-point differences from the reference configuration.

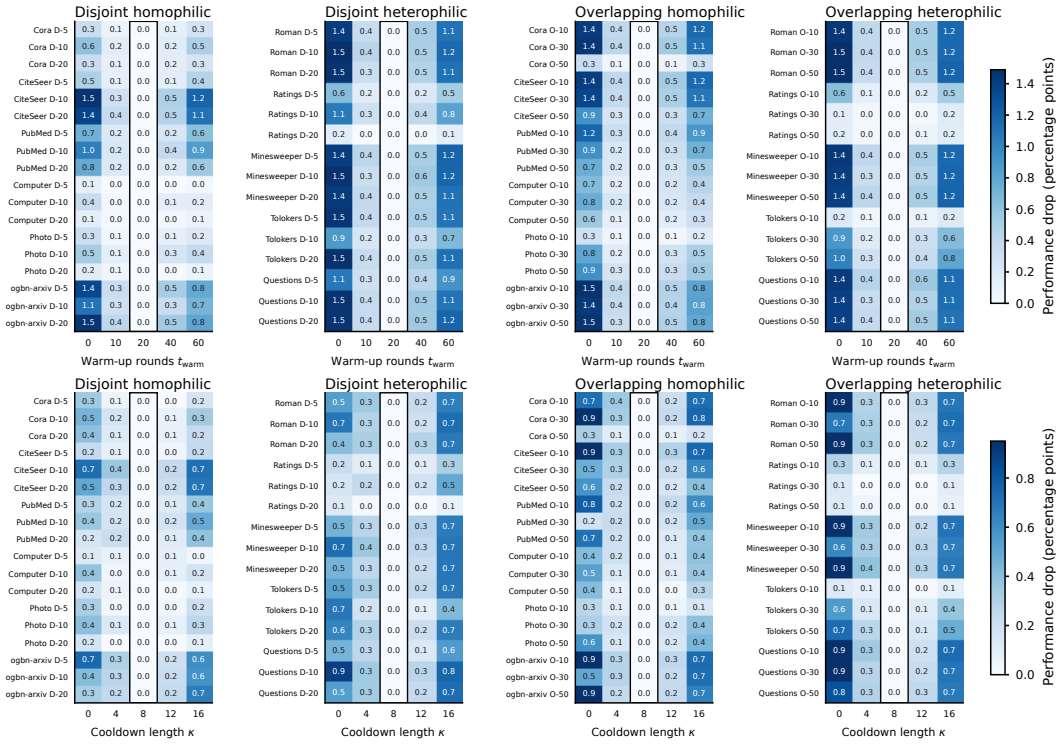


Figure 12: Hyperparameter sensitivity for warm-up t_{warm} and cooldown κ. Values are percentage-point differences from the reference configuration.

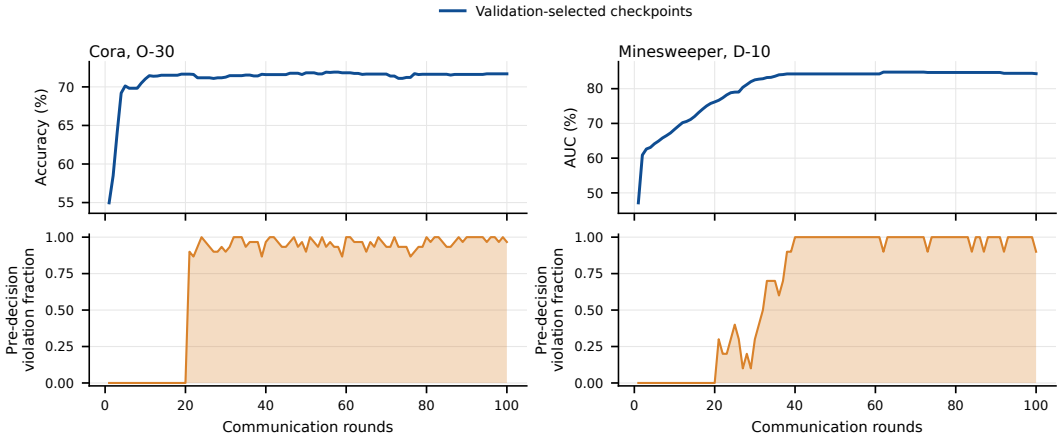


Figure 13: Validation-drawdown audit of saved transfer-training trajectories. Top: mean test performance of client checkpoints selected by validation up to each round. Bottom: fraction of clients whose pre-decision scores fall more than $\epsilon = 0.002$ below their running validation maxima; the first 20 warm-up rounds are displayed as zero.

Homophilic datasets, disjoint partitions

	Cora			CiteSeer			PubMed			Amazon-Computer			Amazon-Photo			ogbn-arxiv		
	5C	10C	20C	5C	10C	20C	5C	10C	20C	5C	10C	20C	5C	10C	20C	5C	10C	20C
FedAvg	4.9	6.8	11.1	5.6	7.9	17.5	30.0	22.0	27.1	42.0	65.2	105.8	21.1	33.6	52.1	130.1	194.3	292.1
FedProx	5.8	8.4	13.4	6.8	9.3	20.2	36.0	25.5	31.9	49.6	78.1	128.0	26.0	40.4	66.7	158.8	241.3	359.0
FedPer	5.4	7.4	12.5	6.2	8.9	18.6	34.3	24.7	30.3	46.1	74.6	119.3	23.3	38.7	58.1	142.5	230.2	330.7
GCFL	10.2	14.2	22.8	12.3	16.4	36.4	65.5	46.3	56.8	85.9	139.5	218.6	45.4	69.7	109.8	275.1	427.9	614.8
FedGNN	5.5	7.8	12.8	6.5	9.0	19.7	33.8	25.3	29.5	47.7	77.2	117.1	23.7	39.2	59.8	149.8	224.2	335.1
FedSage+	8.5	12.0	20.0	10.2	13.9	29.9	54.5	37.5	47.2	73.5	119.5	181.2	37.3	61.7	97.1	232.7	353.1	530.8
FED-PUB	23.4	32.8	54.3	28.3	38.3	84.4	153.5	108.3	129.2	209.5	327.5	513.9	109.5	172.4	275.8	644.1	979.1	1508.5
FedGTA	2.3	3.2	4.9	2.6	3.6	7.9	14.1	9.6	11.8	19.2	29.8	48.1	9.9	15.2	25.0	61.1	93.1	137.6
AdaFGL	2.6	3.4	5.6	3.0	3.9	8.7	15.8	11.3	13.3	21.7	33.3	54.8	11.2	17.6	28.0	68.3	101.8	156.0
FedTAD	5.4	7.9	12.4	6.7	9.1	19.4	35.3	24.4	30.1	46.8	77.1	122.4	24.3	38.9	61.4	150.2	229.4	342.3
FedIIH	19.0	27.3	44.9	23.2	31.0	69.7	120.8	84.2	102.8	165.5	259.5	423.9	87.0	133.0	215.7	512.8	775.4	1191.8
FedSSA	**3.5**	**4.8**	**7.5**	**4.2**	**5.4**	**11.5**	**22.2**	**15.7**	**17.9**	**30.4**	**47.8**	**71.3**	**15.7**	**24.7**	**37.0**	**96.4**	**140.9**	**202.1**
R2GT	**3.6**	**4.8**	**7.9**	**4.5**	**4.9**	**12.1**	**22.4**	**15.8**	**19.2**	**30.6**	**46.3**	**73.0**	**15.8**	**24.4**	**37.0**	**97.4**	**143.6**	**206.9**

Figure 14: Computational cost for homophilic disjoint settings, in seconds per communication round.

Homophilic datasets, overlapping partitions

	Cora			CiteSeer			PubMed			Amazon-Computer			Amazon-Photo			ogbn-arxiv		
	10C	30C	50C	10C	30C	50C	10C	30C	50C	10C	30C	50C	10C	30C	50C	10C	30C	50C
FedAvg	17.8	25.0	26.5	16.5	22.7	43.4	18.0	48.7	172.8	110.2	181.2	214.3	65.2	101.4	128.4	249.6	361.4	463.4
FedProx	21.4	29.7	32.6	21.0	28.3	52.7	21.5	59.2	212.7	133.1	216.3	267.1	79.7	130.2	160.8	305.1	449.4	565.5
FedPer	20.4	27.7	29.7	18.6	26.3	47.2	20.9	54.7	194.8	127.6	199.4	243.2	71.6	118.8	145.8	280.3	404.9	530.1
GCFL	37.9	53.3	56.8	34.6	48.8	91.4	39.3	107.1	369.7	237.8	377.2	464.0	141.4	213.5	284.9	525.2	782.2	976.7
FedGNN	20.0	28.8	29.5	19.2	26.8	48.0	21.0	57.9	202.2	124.3	208.0	246.9	73.5	116.6	151.3	286.3	431.5	530.1
FedSage+	34.2	47.7	50.6	32.7	43.5	79.3	33.3	94.8	339.9	209.0	325.5	403.6	123.5	190.3	255.0	475.5	707.2	879.2
FED-PUB	91.7	132.9	139.0	92.4	121.9	224.0	94.1	260.8	940.4	608.3	969.0	1122.1	352.3	538.2	705.1	1395.4	2029.5	2582.2
FedGTA	8.1	11.3	11.9	7.7	10.5	19.2	8.5	22.4	81.8	50.4	81.1	98.9	30.0	47.5	61.6	118.8	172.6	211.6
AdaFGL	9.3	12.7	13.9	8.9	11.9	22.8	9.4	26.4	89.9	57.5	90.8	110.1	33.4	53.3	69.9	134.1	196.9	238.9
FedTAD	20.8	28.6	30.9	19.8	26.3	48.2	21.4	55.6	198.4	130.0	207.9	246.7	75.2	121.8	149.8	295.5	422.0	536.2
FedIIH	77.1	102.5	109.7	70.0	101.7	185.7	76.6	209.1	749.5	467.0	761.5	881.1	280.1	441.8	553.5	1085.4	1557.7	2031.8
FedSSA	**13.1**	**18.1**	**18.2**	**12.4**	**17.2**	**29.2**	**13.3**	**35.9**	**122.5**	**79.9**	**128.4**	**149.6**	**49.0**	**75.6**	**90.5**	**184.0**	**264.6**	**328.7**
R2GT	**11.4**	**17.7**	**19.4**	**12.9**	**17.6**	**30.2**	**13.6**	**36.4**	**120.7**	**82.6**	**127.2**	**149.2**	**48.8**	**74.7**	**92.0**	**189.2**	**266.1**	**329.8**

Figure 15: Computational cost for homophilic overlapping settings, in seconds per communication round.

Heterophilic datasets, disjoint partitions

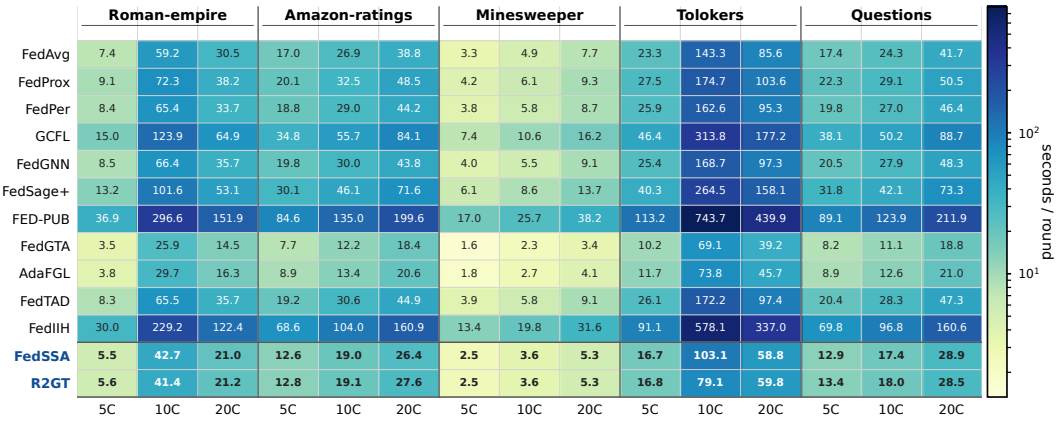


Figure 16: Computational cost for heterophilic disjoint settings, in seconds per communication round.

Heterophilic datasets, overlapping partitions

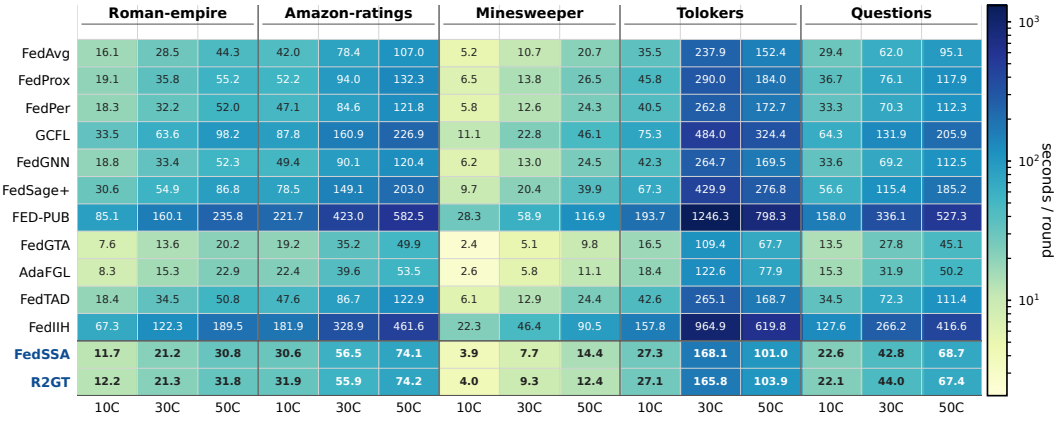


Figure 17: Computational cost for heterophilic overlapping settings, in seconds per communication round.