QUANTIFYING GRAPH STRUCTURAL MEMORIZATION IN LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Memorization in Large Language Models (LLMs) is well documented for sequential content such as text and code. As LLMs are increasingly trained with graph-structured data, however, whether memorization also extends to graph structure remains largely unexplored. To study this question, we introduce an exposure-controlled evaluation framework that examines how graph recoverability changes with training exposure. The framework separates training-specific recovery from general graph prediction by comparing exposed target graphs against graphs never exposed during training. Using this framework, we observe an exposure-dependent form of *graph structural memorization*: graph structure becomes increasingly recoverable as the model is exposed more frequently to the target graph during training. Motivated by this observation, we formalize the underlying recovery procedure as *training-graph stealing*, a black-box attack against graph-trained LLMs that makes this form of memorization concretely exploitable. Given candidate nodes and their textual descriptions, the attack queries each node pair to infer adjacency and assembles the predicted edges into a reconstructed graph. Experiments show that exposure-dependent recovery persists across graph-context settings, graph scales and model families. Together, these findings demonstrate that LLM memorization can extend beyond sequential content to graph structure, creating a new attack surface for extracting graph-structured training data.

1 INTRODUCTION

Memorization in Large Language Models (LLMs) has been extensively studied as a source of training-data leakage, with prior work showing that models can retain and reproduce information encountered during training (Carlini et al., 2021; 2023; Nasr et al., 2023). Meanwhile, LLMs are increasingly trained with graph-structured data, where node information and relational context are incorporated through graph encoders or graph-aware instruction tuning (Tang et al., 2024; Wang et al., 2025; Zhang et al., 2026). This shift makes graph structure itself a potential object of memorization. Such relationships can be costly to construct through experiments or expert curation and may also encode sensitive associations (He et al., 2021; Lin et al., 2025), such as disease–gene, drug–target, or patient-related links in biomedical graphs. If memorization extends from individual sequences to graph structure, these valuable or sensitive relationships may become recoverable from the model, making graph structure a new source of training-data leakage.

Existing studies of memorization and training-data extraction, however, largely evaluate memorization through sequential reproduction. Given a prefix from a training example, an attacker tests whether the model reproduces the corresponding suffix (Carlini et al., 2021; 2023; Nasr et al., 2023). This criterion does not transfer directly to graphs because the same topology may admit many equivalent serializations through different node orders or traversal sequences (You et al., 2018). Conversely, outputs containing the same nodes may encode different relationships. Exact sequence matching therefore cannot reliably determine whether the underlying graph structure has been memorized or recovered. Studying *graph structural memorization* instead requires shifting the object of interest from a token sequence to the graph structure formed by relationships among nodes.

Once memorization is considered at the level of graph structure, a natural next question is whether such structural information can be recovered from model behavior. Prior graph attacks provide the closest evidence that model access can reveal private connectivity. Attacks against graph neural net-

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.

works (GNNs) infer links from model outputs or reconstruct adjacency information from internal parameters (He et al., 2021; Wu et al., 2022; Zhang et al., 2021; Lin et al., 2025), while attacks on knowledge graph APIs and Graph RAG systems recover relationships from external graph stores accessed during inference (Xi, 2025; Liu et al., 2025; Gu et al., 2026). However, these settings do not establish whether graph structure is retained because it was encountered during LLM training. Moreover, successful recovery alone is insufficient to establish memorization: even an unexposed model may infer correct edges from node-text similarity (Guan et al., 2025) or relational patterns learned from other graphs. Graph recovery must therefore be separated from general graph prediction before it can be attributed to target-graph exposure.

To address this gap, we introduce an exposure-controlled evaluation framework to probe graph structural memorization by examining how the recoverability of graph structure changes with training exposure. Within this framework, we compare these changes against graphs that were never exposed during training to determine whether the observed recovery is specific to the target graph rather than a general improvement in graph prediction. An exposure-dependent form of *graph structural memorization* is then observed: graph structure becomes increasingly recoverable as training exposure increases. Motivated by this observation, we further formalize this recovery process as *training-graph stealing*, a black-box attack against graph-trained LLMs, thereby demonstrating its security consequence. Specifically, given candidate nodes and their textual descriptions, the attack queries each node pair for adjacency and assembles the predicted edges into a reconstructed graph, avoiding the need for a canonical graph serialization. Recovery is assessed at both the whole-graph and local-graph levels. Our main contributions are summarized as follows:

- We identify exposure-dependent graph structural memorization in LLMs, showing that graph structure becomes increasingly recoverable with target-graph exposure.
- We introduce an exposure-controlled framework to evaluate and attribute graph structural memorization. By contrasting exposed target graphs with graphs never seen during training, the framework separates training-specific recovery from general graph prediction.
- We formalize the recovery of memorized graph structure as training-graph stealing, a black-box attack that reconstructs graph structure and demonstrates its security consequence across graph-context settings, graph scales and model families.

2 Related Work

LLM Memorization. Existing work shows that LLMs can memorize and reproduce content from their training data (Carlini et al., 2021; Nasr et al., 2023). Memorization and extractability are commonly studied through sequential reproduction, for example by conditioning on a prefix and comparing the generated continuation with the corresponding training suffix (Carlini et al., 2021; 2023). Counterfactual memorization further attributes memorization to training exposure by measuring how model predictions change when a training example is omitted (Zhang et al., 2023). Prior studies also show that extractability varies with factors such as training exposure and model scale, and can persist in production language models (Carlini et al., 2023; Nasr et al., 2023). Meanwhile, LLMs are increasingly trained with graph-structured data. GraphGPT aligns graph-encoder representations with an LLM through graph instruction tuning (Tang et al., 2024), UniGTE jointly encodes graph structure and textual instructions for cross-task generalization (Wang et al., 2025), and RGLM introduces graph reconstruction objectives to strengthen graph–text alignment (Zhang et al., 2026). These methods primarily study graph understanding and generalization, while whether graph structure encountered during training is memorized and recoverable remains largely unexplored. Our work extends LLM memorization from sequential content to graph structure by examining graph recovery under controlled training exposure.

Graph Privacy and Stealing Attacks. Graph privacy attacks show that trained graph models can reveal private connectivity under different access assumptions. Under black-box access, links can be inferred from GNN outputs (He et al., 2021), while LinkTeller (Wu et al., 2022) recovers private edges by measuring the influence of adversarial input perturbations on model predictions. Under stronger access, GraphMI (Zhang et al., 2021) formulates graph reconstruction as model inversion, while candidate graphs can also be generated with a graph diffusion model and identified using

victim GNN parameters (Lin et al., 2025). An LLM has also been used as the attack model to combine node textual features with posterior predictions from a victim GNN for link inference (Guan et al., 2025), but the victim remains a GNN rather than an LLM trained on graph structure. Related attacks instead target graph information stored outside the model: KGX queries knowledge-graph reasoning APIs to reconstruct private subgraphs (Xi, 2025), while Graph RAG attacks extract entities, relationships, or multi-hop structure exposed through retrieval-time access (Liu et al., 2025; Gu et al., 2026). In contrast, our work targets graph structure incorporated into LLM training and studies whether training exposure makes such structure recoverable through black-box queries.

3 METHODOLOGY

3.1 PROBLEM FORMULATION

Let $G = (V, E)$ be the target graph whose structure we seek to recover, where V and E are its node and edge sets, with X denoting the textual descriptions associated with the nodes in V. We denote by M_e an LLM trained at target-graph exposure level e, where e is a non-negative integer specifying the exposure frequency of training records derived from G. The case $e = 0$ corresponds to no target-graph exposure. We study whether increasing e makes the graph structure of G increasingly recoverable from the resulting model. Because correct structural predictions may also arise without target-graph exposure, we further ask whether improvements in recoverability are specific to the exposed target graph rather than explained by a general improvement in graph prediction.

Graph Recoverability. Given the candidate nodes V and their textual descriptions X, let $\mathcal{A}$ denote a fixed recovery procedure that interacts with the LLM M_e and reconstructs the target graph, with $\widehat{E}_e$ denoting the recovered edge set. The reconstructed graph $\widehat{G}_e$ is

$$\widehat{G}_e = \mathcal{A}(M_e; V, X) = (V, \widehat{E}_e). \tag{1}$$

We use a recovery score $R(\widehat{G}_e, G)$, which takes $\widehat{G}_e$ and G as inputs and measures how closely the reconstruction matches the target, with larger values indicating better recovery. The specific recovery criteria for whole-graph and local-graph reconstruction are introduced in Section 3.4.

Definition 1 (Graph Structural Memorization). Given a fixed recovery procedure $\mathcal{A}$ and recovery score R, a target graph G is said to exhibit *graph structural memorization* if increasing its training exposure from the unexposed baseline $e = 0$ produces an increase in $R(\widehat{G}_e, G)$ that exceeds the corresponding recovery change observed on graphs not exposed during training.

This definition differs from conventional sequence-based notions of LLM memorization, which typically determine whether a model reproduces a specific training suffix when conditioned on its prefix (Carlini et al., 2021; 2023). Such criteria rely on exact reproduction under a particular token ordering and therefore do not transfer directly to graphs, which may admit multiple equivalent serializations. We instead define memorization through the recoverability of graph structure. High recoverability alone, however, is not sufficient evidence of memorization, since a model may predict graph relationships even without direct exposure to the target graph. Accordingly, the target recovery gain must exceed the corresponding recovery change on unexposed graphs to distinguish graph structural memorization from general graph prediction.

3.2 EXPOSURE-CONTROLLED EVALUATION

To attribute changes in graph recoverability to training exposure, we systematically vary how frequently the target graph is presented during training while keeping the remaining training conditions fixed. Let $\mathcal{Q}(G)$ denote the set of training records derived from the target graph. At exposure level e, each record in $\mathcal{Q}(G)$ is presented e times during training. To isolate the effect of target-graph exposure, these records replace an equal number of filler records, keeping both the total number of training examples and the number of optimization steps unchanged across exposure levels. Varying e therefore changes the model's exposure to G without increasing the overall amount of training.

Changes in target-graph recovery alone, however, are insufficient to attribute the effect to graph structural memorization. Training under different exposure levels may also alter the model's general ability to predict graph relationships, leading to recovery changes even for graphs that were

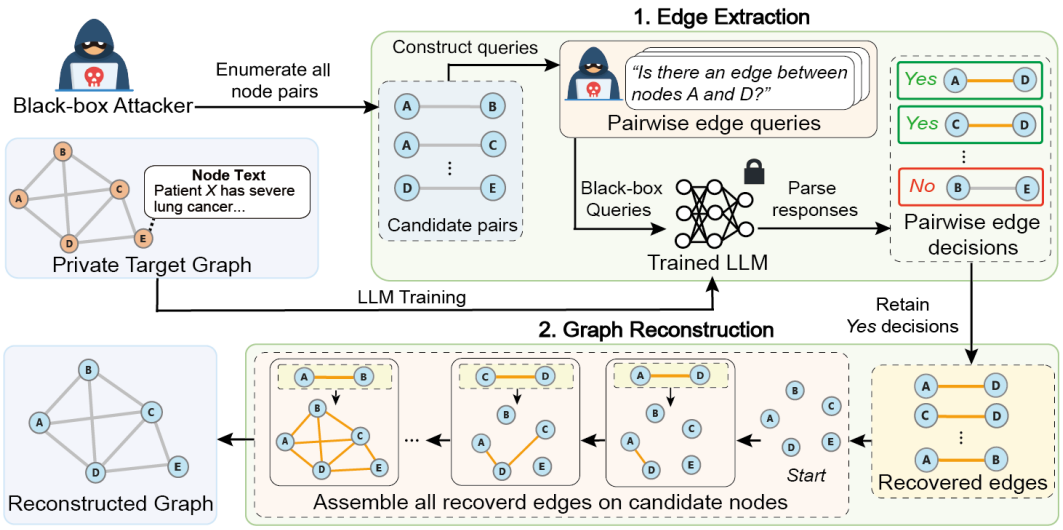


Figure 1: Overview of the training-graph stealing attack. A private target graph is incorporated into LLM training. After training, a black-box attacker queries candidate node pairs for direct adjacency and assembles the positive decisions to reconstruct graph structure.

never exposed during training. Motivated by counterfactual memorization (Zhang et al., 2023), for each target graph, we evaluate its corresponding model M_e on both the exposed target graph and one size-matched unexposed control graph with non-overlapping node identities, using the same recovery procedure $\mathcal{A}$. We first aggregate the recovery scores within each graph-size setting and then across graph sizes. We denote the resulting target and control recovery scores by $R_{\text{tar}}(e)$ and $R_{\text{ctrl}}(e)$, respectively. The controls capture recovery changes that arise without direct exposure to the evaluated graph. To isolate the component of recovery attributable to target-graph exposure, we compare the recovery gain relative to the unexposed baseline $e = 0$ on the target graphs against the corresponding change on the controls:

$$\Gamma_R(e) = [R_{\text{tar}}(e) - R_{\text{tar}}(0)] - [R_{\text{ctrl}}(e) - R_{\text{ctrl}}(0)]\,. \tag{2}$$

The first term measures the target recovery gain, while the second accounts for the corresponding shift on unexposed controls. When target recovery also improves relative to $e = 0$, $\Gamma_R(e) > 0$ indicates that this improvement exceeds the corresponding change on unexposed controls. Tracking these exposure-dependent recovery changes implements Definition 1.

3.3 Training-Graph Stealing

The exposure-controlled evaluation above examines how graph recoverability changes with exposure, determining when recovery gains can be attributed to target-graph exposure. Such attribution, however, establishes memorization only at the level of recovery behavior. It does not reveal what an adversary could recover from the model in practice. If the memorized graph structure can be inferred through black-box interaction, then relationships contained in the training graph may become accessible to an external attacker. To examine this security consequence, we formalize the underlying recovery procedure $\mathcal{A}$ as *training-graph stealing*, a black-box attack against graph-trained LLMs that infers pairwise relationships from model responses and assembles them into a graph.

Attack Setting. The attacker is assumed to know the candidate node set V and the associated textual descriptions X, but not the complete target edge set E. Depending on the access setting, no graph structure is provided, or partial surrounding structure is available while the queried relationship remains unknown. The attacker has only black-box generation access to M_e: queries can be submitted and generated responses observed, while the model parameters and training data remain inaccessible. The objective is to infer the hidden relationships among the known candidate nodes and recover an edge set $\widehat{E}_e$ such that the resulting graph $\widehat{G}_e = (V, \widehat{E}_e)$ approximates the target graph G. As shown in Figure 1, the attack does not assume a canonical serialization of G. Instead, it recovers the graph through individual adjacency decisions.

Edge Extraction. To recover graph structure without relying on a particular graph serialization, we decompose reconstruction into pairwise adjacency decisions over the known candidate nodes. Let $\mathcal{C} = \binom{V}{2}$ denote the set of all unordered pairs of distinct nodes in V. For each candidate pair $\{u, v\} \in \mathcal{C}$, we construct a direct-adjacency query q_{uv} that identifies the two nodes, provides their textual descriptions, and asks *"Is there a direct edge between graph node u and graph node v?"* The same query format is used across all candidate pairs, with direct adjacency distinguishing an edge from a multi-hop relationship. Under our black-box setting, the attacker observes the model's generated response rather than its internal logits, so adjacency must be inferred from the returned text. We therefore apply a deterministic parser ψ to convert each response into a binary edge decision:

$$\widehat{y}_{uv}^{(e)} = \psi(M_e(q_{uv})) \in \{0, 1\}, \{u, v\} \in \mathcal{C}, \tag{3}$$

where $\widehat{y}_{uv}^{(e)} = 1$ indicates that the parsed response supports a direct edge between u and v, while $\widehat{y}_{uv}^{(e)} = 0$ indicates that the response is interpreted as a predicted non-edge.

Graph Reconstruction. Although the pairwise decisions recover individual relationships, the security risk arises from their composition into a graph structure. Once all candidate pairs have been queried, we collect the pairs predicted as adjacent to form the recovered edge set:

$$\widehat{E}_e = \left\{ \{u, v\} \in \mathcal{C} \;\middle|\; \widehat{y}_{uv}^{(e)} = 1 \right\}, \widehat{G}_e = \left(V, \widehat{E}_e\right). \tag{4}$$

The reconstruction retains the candidate node set V, including nodes for which no incident edge is predicted, and adds an undirected edge for every positive adjacency decision. In this way, isolated pairwise predictions are composed into a single graph-level estimate of the target structure. Although the attacker queries only direct adjacency, the assembled graph $\widehat{G}_e$ reveals more than any single query: multi-hop paths, neighborhoods, and reachable regions can all be derived by traversing the recovered edges, without any additional black-box queries. A sequence of local, pairwise queries can therefore expose graph structure well beyond what any individual query was designed to reveal. Section 3.4 evaluates this leakage at both the whole-graph and local-graph levels.

3.4 Recovery Criteria

Whole-Graph Recovery. Whole-graph recovery measures how faithfully $\widehat{G}_e$ reproduces the complete structure of G. We evaluate both exact and partial reconstruction. Since $\widehat{G}_e$ and G share the candidate node set V, a target graph is reconstructed exactly when $\widehat{E}_e = E$, that is, when all target edges are recovered and no spurious edge is introduced. Because exact recovery is binary per graph and target counts vary by graph size, we average indicators within each size and equally across sizes. Let $\mathcal{M}$ be the set of evaluated graph sizes, with N_k target graphs of size $k \in \mathcal{M}$, indexed by i:

$$\text{ExactGraph Rate}(e) = \frac{1}{|\mathcal{M}|} \sum_{k \in \mathcal{M}} \frac{1}{N_k} \sum_{i=1}^{N_k} \mathbb{I}\left[\widehat{E}_e^{(i)} = E^{(i)}\right]. \tag{5}$$

This size-balanced score summarizes how exact recovery changes with exposure. To distinguish among incomplete reconstructions, we further use fused Gromov–Wasserstein (FGW) distance to quantify partial recovery (Titouan et al., 2019). Unlike ExactGraph, which assigns zero to every incomplete reconstruction regardless of whether it differs from the target graph by one edge or many, FGW compares multi-hop structure together with the semantic information associated with the nodes, allowing finer-grained comparison among incomplete reconstructions:

$$\text{FGW}_\alpha\left(\widehat{G}_e, G\right) = \min_T \left[(1-\alpha)\mathcal{L}_{\text{sem}}(T) + \alpha\mathcal{L}_{\text{str}}(T)\right], \tag{6}$$

where T is a transport coupling, $\mathcal{L}_{\text{sem}}$ measures cosine distances between node textual-description embeddings, and $\mathcal{L}_{\text{str}}$ measures discrepancies in all-pairs shortest-path distances. The weight $\alpha \in [0, 1]$ balances the two components. To report partial recovery on a bounded scale, we transform FGW distance into similarity:

$$s_{\text{FGW}}\left(\widehat{G}_e, G\right) = \frac{1}{1 + \text{FGW}_\alpha\left(\widehat{G}_e, G\right)}, \tag{7}$$

with higher values indicating stronger recovery. As with ExactGraph Rate, we average FGW similarity within each graph-size stratum and then across strata, giving each size equal weight. The full FGW formulation is provided in Appendix A.4.

Local-Graph Recovery. Although whole-graph recovery summarizes overall reconstruction quality, it does not show how much multi-hop structure remains accessible from a particular node. This matters because a missing edge near a start node may disconnect an entire downstream region even when most edges are recovered correctly. Moreover, an attacker may focus on the relationships reachable from a particular node rather than on reconstructing the entire graph. We therefore evaluate local recovery through the multi-hop reachable neighborhood, expanding outward from a selected start node. For a graph G' over the candidate node set V, start node $s \in V$, and hop depth $t \geq 1$, let $d_{G'}(s, v)$ denote the graph distance between s and v. The t-hop local structure around s is

$$\mathcal{H}_t(G', s) = G'[\{v \in V : d_{G'}(s, v) \leq t\}], \tag{8}$$

where $G'[U]$ denotes the subgraph of G' induced by node set U. The size of $\mathcal{H}_t(G', s)$ therefore varies with t, s, and graph connectivity. Let $\widehat{N}_s^{(t)}$ denote the nodes other than s in $\mathcal{H}_t(\widehat{G}_e, s)$, and $N_s^{(t)}$ the nodes other than s in $\mathcal{H}_t(G, s)$. To quantify agreement between the recovered and target reachable neighborhoods, we define

$$\mathrm{Reach}_t(s) = \frac{\left|\widehat{N}_s^{(t)} \cap N_s^{(t)}\right|}{\left|\widehat{N}_s^{(t)} \cup N_s^{(t)}\right|}. \tag{9}$$

A higher $\mathrm{Reach}_t(s)$ indicates agreement in the multi-hop reachable neighborhoods of the recovered and target graphs, capturing local reachability while penalizing both missing and spurious nodes. Since local recovery may vary across start nodes, we average $\mathrm{Reach}_t(s)$ over all start nodes per graph to obtain a graph-level measure, and then aggregate across target graphs for each hop depth.

4 Experiments

4.1 Experimental Setup

Datasets and Evaluation Criteria. We evaluate on four text-attributed graph datasets: Cora (Wen & Fang, 2023), PubMed (He et al., 2024), CiteSeer (Kipf & Welling, 2017), and ogbn-products (He et al., 2024). The first three are citation networks, while ogbn-products is a product co-purchase network for cross-domain evaluation. For each dataset, we construct 24 connected induced target graphs with $n \in \{5, 10, 20, 40\}$ nodes, distributed as $9/9/3/3$ across sizes, and one matched unexposed control graph per size for exposure attribution. Additional details of datasets are provided in Appendix A.1. We evaluate whole-graph recovery using ExactGraph Rate and FGW similarity, with local-graph recovery using Reach_t, as defined in Section 3.4. Graph structural memorization is quantified by the target-specific exposure gain $\Gamma_R(e)$ defined in Section 3.2.

Implementation Details. We evaluate GraphGPT-7B (Tang et al., 2024), Qwen2.5-7B-Instruct (Yang et al., 2025), and Llama-3.1-8B-Instruct (Grattafiori et al., 2024), spanning graph-oriented and general-purpose LLMs. Each target graph is trained independently at exposure levels $e \in \{0, 20, 60\}$, with target-derived records replacing matched filler records to keep the training budget fixed. Across all models, we freeze the graph encoder and train a model-specific graph projector together with LoRA adapters on the LLM backbone. Pairwise queries use deterministic greedy decoding. At inference, we consider *no graph context*, where only node information is provided, and *masked graph context*, where local graph context is available but the queried edge is removed. Training hyperparameters and implementation details are provided in Appendix A.2.

4.2 Quantification of Graph Structural Memorization

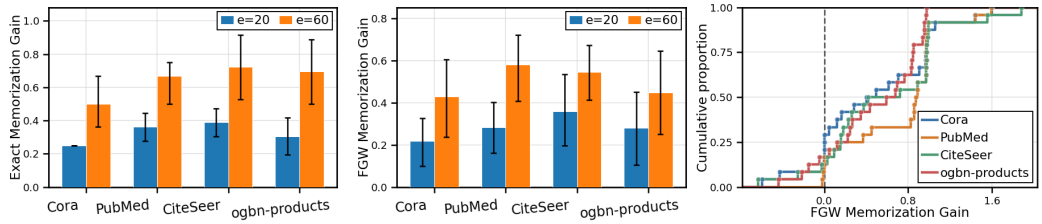
Figures 2a and 2b quantify how *graph structural memorization* changes with repeated target exposure for GraphGPT with masked graph context. On Cora, the ExactGraph memorization gain doubles from 0.250 at $e = 20$ to 0.500 at $e = 60$, while CiteSeer reaches 0.722 at $e = 60$. The target-specific FGW similarity gain reaches 0.580 on PubMed and 0.546 on CiteSeer at $e = 60$. Together with the ExactGraph gains, these results indicate that target-graph exposure strengthens memorization of both complete and partial graph structure beyond recovery changes on unexposed controls. The strength of this effect varies across datasets.

Table 1: Whole-graph recovery across four datasets with no graph context. Higher ExactGraph Rate and FGW similarity s_{FGW} indicate better graph recovery. Best results are in **bold** and second-best results are underlined. Gray highlights the recovery at exposure level $e = 0$.

Model	Exposure	Cora		PubMed		CiteSeer		ogbn-products	
		ExactGraph Rate ↑	s_{FGW} ↑	ExactGraph Rate ↑	s_{FGW} ↑	ExactGraph Rate ↑	s_{FGW} ↑	ExactGraph Rate ↑	s_{FGW} ↑
GraphGPT-7B	$e = 0$	0.000	0.308	0.000	0.176	0.000	0.215	0.000	0.126
	$e = 20$	0.167	0.386	0.250	0.360	0.139	0.548	0.000	0.289
	$e = 60$	0.194	0.436	0.139	0.391	0.250	0.561	0.028	**0.387**
Qwen2.5-7B-Instruct	$e = 0$	0.000	0.210	0.000	0.158	0.000	0.200	0.000	0.182
	$e = 20$	0.139	0.325	0.083	0.279	0.139	0.293	0.083	0.253
	$e = 60$	**0.528**	**0.713**	**0.417**	**0.593**	0.306	0.533	**0.250**	0.344
Llama-3.1-8B-Instruct	$e = 0$	0.000	0.188	0.000	0.197	0.000	0.171	0.000	0.185
	$e = 20$	0.056	0.300	0.139	0.271	**0.389**	**0.572**	0.000	0.195
	$e = 60$	0.194	0.459	0.083	0.353	0.167	0.446	0.028	0.271

Table 2: Whole-graph recovery with masked graph context. We use the same notation as Table 1.

Model	Exposure	Cora		PubMed		CiteSeer		ogbn-products	
		ExactGraph Rate ↑	s_{FGW} ↑	ExactGraph Rate ↑	s_{FGW} ↑	ExactGraph Rate ↑	s_{FGW} ↑	ExactGraph Rate ↑	s_{FGW} ↑
GraphGPT-7B	$e = 0$	0.000	0.148	0.000	0.150	0.000	0.165	0.000	0.125
	$e = 20$	0.250	0.341	0.361	0.390	0.389	0.501	0.306	0.476
	$e = 60$	0.500	0.612	0.667	0.668	**0.722**	0.807	0.694	0.730
Qwen2.5-7B-Instruct	$e = 0$	0.000	0.220	0.000	0.107	0.000	0.203	0.000	0.158
	$e = 20$	0.333	0.370	0.306	0.341	0.306	0.510	0.194	0.331
	$e = 60$	**0.722**	**0.824**	**0.917**	**0.917**	0.667	**0.834**	**0.861**	**0.870**
Llama-3.1-8B-Instruct	$e = 0$	0.000	0.210	0.000	0.189	0.000	0.188	0.000	0.189
	$e = 20$	0.333	0.390	0.389	0.520	0.500	0.584	0.250	0.448
	$e = 60$	0.583	0.589	0.556	0.650	0.583	0.678	0.694	0.728



(a) (b) (c)

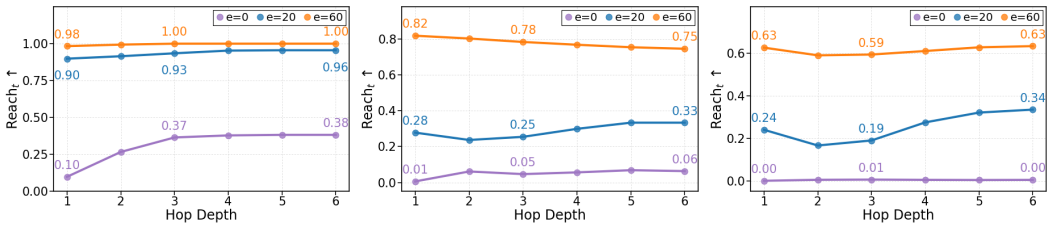
Figure 2: Quantification of graph structural memorization with GraphGPT-7B under *masked graph context*. (a) ExactGraph and (b) FGW memorization gains across datasets at $e = 20$ and $e = 60$. (c) Cumulative distributions of FGW memorization gains at $e = 60$.

Figure 2c further examines how memorization is distributed across individual target graphs. At $e = 60$, positive per-graph FGW memorization gains occur for at least 75.0% of targets in every dataset and peak at 87.5% on CiteSeer. Dataset-level median gains range from 0.447 on Cora to 0.880 on PubMed. Although memorization strength varies across datasets and individual targets, the high positive-gain coverage and positive medians show that it extends beyond a small set of unusually recoverable graphs. Additional results for Qwen2.5 and Llama-3.1, together with further analyses of the memorization mechanism, are provided in Appendices B.1 and B.3.

4.3 Whole-Graph and Local-Graph Recovery

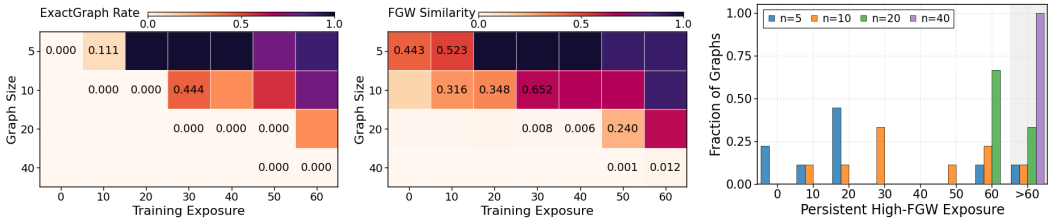
Recovery with No Graph Context. The gray $e = 0$ rows in Table 1 mark recovery before any target-specific exposure. At this unexposed baseline, models show partial recovery with positive FGW similarity, but ExactGraph Rate remains zero throughout. Exposure changes this picture substantially. At $e = 60$, recovery becomes positive for most models and datasets, and Qwen2.5 achieves the strongest recovery overall. From $e = 0$ to $e = 60$, Qwen2.5's ExactGraph Rate and FGW similarity rise from 0 to 0.528 and 0.210 to 0.713 on Cora, and from 0 to 0.417 and 0.158 to 0.593 on PubMed, respectively. GraphGPT peaks at an ExactGraph Rate of 0.250 on PubMed at $e = 20$ and FGW similarity of 0.561 on CiteSeer at $e = 60$. Llama-3.1 reaches its highest ExactGraph Rate of 0.389 and FGW similarity of 0.572 on CiteSeer at $e = 20$. These results show that training exposure alone allows an attacker to recover complete target graphs from node information, without any graph context at inference time.

Recovery with Masked Graph Context. The $e = 0$ rows in Table 2 report the corresponding unexposed baseline under masked graph context. No target graph is reconstructed exactly at this



(a) (b) (c)

Figure 3: Local-graph recovery with GraphGPT-7B on ogbn-products under masked graph context. We report reachable-node recovery Reach_t as the hop depth increases for different target graph sizes: (a) $n = 10$, (b) $n = 20$, and (c) $n = 40$.



(a) (b) (c)

Figure 4: Effects of exposure and graph scale on GraphGPT-7B whole-graph recovery on Cora with masked graph context: (a) ExactGraph Rate, (b) FGW similarity and (c) the distribution of persistent high-FGW exposure across graph sizes.

baseline, though FGW similarity is already positive. Exposure further improves whole-graph recovery when surrounding topology is available. At $e = 60$, ExactGraph Rate reaches at least 0.500 for every model and dataset, peaking at 0.917. The benefit of masked graph context is especially clear on ogbn-products, where it raises ExactGraph Rate from 0 to 0.694 for GraphGPT, from 0.250 to 0.861 for Qwen2.5, and from 0 to 0.694 for Llama-3.1, relative to no graph context. Across nonzero exposure levels, masked graph context yields higher ExactGraph Rate for every model and dataset and generally higher FGW similarity than no graph context. Since the queried edge remains hidden, these gains indicate that an attacker can use surrounding topology to elicit structural information retained through target-graph exposure.

Composition into Multi-Hop Structure. Beyond individual edges, local-graph recovery asks how much target structure an attacker can reach by starting from a particular node and composing the recovered pairwise relations across multiple hops. Figure 3 shows that these relation-level decisions form extended reachable neighborhoods. For GraphGPT on ogbn-products, increasing exposure from $e = 0$ to $e = 60$ raises Reach_6 from 0.383 to 1.000 at $n = 10$, from 0.063 to 0.746 at $n = 20$, and from 0.004 to 0.634 at $n = 40$. Even when little reachable structure is recovered at $e = 0$, exposure enables recovery of extended multi-hop connectivity at $n = 20$ and $n = 40$, though this recovery remains lower on the larger graphs, showing that multi-hop reconstruction grows more difficult as the target structure grows. Additional results are provided in Appendix C.1.

4.4 Effects of Training Exposure and Graph Scale

Figures 4a and 4b show a clear scale-dependent recovery pattern over $e \in \{0, 10, 20, 30, 40, 50, 60\}$. Smaller graphs reach high FGW similarity and exact recovery at lower exposure, while larger graphs require more exposure and may remain only partially recovered by $e = 60$. These results indicate that increasing exposure generally advances recovery from partial similarity toward exact reconstruction. From the attacker's perspective, graph scale determines how much prior exposure a target needs before pairwise extractions compose into a complete reconstruction and whether this point is reached within the tested range. Figure 4c examines this transition at the individual target-graph level. We define persistent high-FGW exposure as the smallest tested exposure at which a target graph achieves and maintains FGW similarity ≥ 0.8, distinguishing near-complete structural agreement from weaker partial matches. Graphs not satisfying this condition by $e = 60$ are grouped under > 60. By $e = 20$, 7 of the 9 targets with $n = 5$ satisfy this criterion, while 8 of the 9 targets

Table 3: Mean edge-recovery rates by shortest-path usage for GraphGPT-7B on Cora and PubMed with masked graph context.

Dataset	Exposure	Shortest-path usage				Range ↓
		Lowest	*Low*	*High*	*Highest*	
Cora	$e = 0$	0.236	0.175	0.250	0.240	0.075
	$e = 20$	0.611	0.625	0.594	0.573	0.052
	$e = 60$	0.753	0.757	0.735	0.762	**0.027**
PubMed	$e = 0$	0.125	0.100	0.219	0.141	0.119
	$e = 20$	0.653	0.708	0.632	0.660	0.076
	$e = 60$	0.833	0.833	0.833	0.833	**0.000**

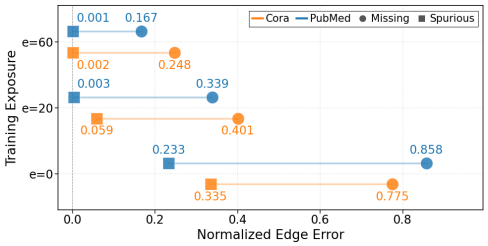


Figure 5: Normalized edge errors across exposures for GraphGPT-7B on Cora and PubMed.

with $n = 10$ meet it by $e = 60$. None of the three targets with $n = 40$ reaches the criterion by the final exposure. These thresholds separate early gains from recovery that remains reliable at later exposures. The complete fine-grained exposure sweep is provided in Appendix C.2.

4.5 Analysis of Training-Graph Stealing

Recovery Across Structural Positions. Edges that appear on many shortest paths help connect graph regions and often form part of the structural backbone (Girvan & Newman, 2002). If *training-graph stealing* preferentially recovers these edges, attack success may depend on structural position instead of extending evenly across the target topology. Table 3 tests this by ranking true edges within each graph by their frequency on shortest paths between node pairs and dividing them into four equal-sized groups: *Lowest*, *Low*, *High*, and *Highest*. We report Range as the difference between the highest and lowest group recovery rates, with smaller values indicating more even recovery. On Cora, group recovery rates span 0.175 to 0.250 at $e = 0$ and 0.735 to 0.762 at $e = 60$, with Range on Cora decreasing from 0.075 to 0.027. On PubMed, all four groups reach 0.833 at $e = 60$, reducing Range on PubMed to zero. Exposure reduces dependence on structural position and expands recovery from the backbone to edges with lower shortest-path usage. As peripheral relations become comparably recoverable, an attacker need not prioritize backbone edges at high exposure.

Decomposition of Graph Reconstruction Errors. We decompose reconstruction errors at the edge level to identify which failure mode limits an attacker under incomplete recovery. Missing-edge errors are unrecovered true relations, and spurious-edge errors are incorrect relations the attack introduces, distinguishing limited coverage from unreliable structure. Both are error metrics, so lower values indicate better reconstruction. Figure 5 shows that from $e = 0$ to $e = 60$, Cora's normalized missing-edge error falls from 0.775 to 0.248, and its spurious-edge error from 0.335 to 0.002. On PubMed, the corresponding errors fall from 0.858 to 0.167 and from 0.233 to 0.001. At high exposure, the graphs reconstructed by the attacker contain almost no false relations and instead resemble incomplete subgraphs of the targets. ExactGraph failures can therefore still expose a large portion of the true topology with few spurious relations: limited coverage, rather than unreliable edges, remains the main barrier to complete graph stealing. Additional attack analyses and qualitative visualizations are provided in Appendices D.2 and G.1.

5 Conclusion

In this work, we investigate whether LLM memorization extends from sequential content to graph structure. We introduce an exposure-controlled evaluation framework to quantify *graph structural memorization*, comparing exposed target graphs against unexposed controls to isolate target-specific recovery from general graph prediction. Using this framework, we find that target graphs become increasingly recoverable with greater training exposure, and that this recovery advantage is target-specific rather than explained by a general improvement in graph prediction. Motivated by this finding, we further formalize the underlying recovery procedure as *training-graph stealing*, a black-box attack showing that graph structural memorization can translate into concrete extraction of hidden training graphs. The attack remains effective under both *no graph context* and *masked graph context*, and recovered edges further compose into larger local multi-hop structure without additional queries. These findings establish graph structure as a distinct form of memorized training information and a new privacy attack surface for graph-trained LLMs.

LLM USE STATEMENT

Generative AI tools were used to assist with drafting and polishing the manuscript, primarily to improve wording, clarity, and organization. All AI-assisted text was reviewed and revised by the authors to ensure that it accurately reflects the methods, results, and conclusions, and the authors take full responsibility for the final content.

ETHICS STATEMENT

The research presented in this paper studies graph structural memorization in Large Language Models and its exploitation through black-box extraction. We acknowledge that this work is dual-use: while our primary motivation is to characterize and quantify graph structural memorization, we recognize that malicious users could potentially adapt the underlying querying procedure to infer sensitive relationships, such as disease–gene or patient-related links, from deployed graph-trained models. However, precisely characterizing and quantifying this phenomenon is important for understanding when recovered graph structure reflects training-specific leakage rather than incidental graph prediction. Such measurement can help reveal the conditions under which graph structural memorization emerges and becomes recoverable, providing useful evidence for auditing and mitigating structural leakage in graph-trained models. All experiments in this study were conducted using publicly available academic benchmark datasets and open-weights models. We did not target any private deployed systems or utilize undisclosed sensitive graphs, and we do not condone the use of our methods to extract relational information from systems without authorization.

REFERENCES

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *USENIX Security Symposium*, 2021.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *ICLR*, 2023.

M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 2002.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Jinze Gu, Qinghua Mao, Xi Lin, and Jun Wu. Graphsteal: Structural knowledge stealing from graph rag via traversal reconstruction. *arXiv preprint arXiv:2605.28645*, 2026.

Faqian Guan, Tianqing Zhu, Wenhan Chang, Wei Ren, and Wanlei Zhou. Large language models merging for enhancing the link stealing attack on graph neural networks. *IEEE Transactions on Dependable and Secure Computing*, 2025.

Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning. In *ICLR*, 2024.

Xinlei He, Jinyuan Jia, Michael Backes, Neil Zhenqiang Gong, and Yang Zhang. Stealing links from graph neural networks. In *USENIX Security Symposium*, 2021.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

Minhua Lin, Enyan Dai, Junjie Xu, Jinyuan Jia, Xiang Zhang, and Suhang Wang. Stealing training graphs from graph neural networks. In *KDD*, 2025.

Jiale Liu, Jiahao Zhang, and Suhang Wang. Exposing privacy risks in graph retrieval-augmented generation. *arXiv preprint arXiv:2508.17222*, 2025.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, et al. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*, 2023.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. Graphgpt: Graph instruction tuning for large language models. In *ACM SIGIR*, 2024.

Vayer Titouan, Nicolas Courty, Romain Tavenard, Chapel Laetitia, and Rémi Flamary. Optimal transport for structured data with application on graphs. In *ICML*, 2019.

Duo Wang, Yuan Zuo, Guangyue Lu, and Junjie Wu. Unigte: Unified graph-text encoding for zero-shot generalization across graph tasks and domains. In *NeurIPS*, 2025.

Zhihao Wen and Yuan Fang. Augmenting low-resource text classification with graph-grounded pre-training and prompting. In *ACM SIGIR*, 2023.

Fan Wu, Yunhui Long, Ce Zhang, and Bo Li. Linkteller: Recovering private edges from graph neural networks via influence analysis. In *IEEE Symposium on Security and Privacy*, 2022.

Zhaohan Xi. All your knowledge belongs to us: Stealing knowledge graphs via reasoning apis. *arXiv preprint arXiv:2503.09727*, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

Jiaxuan You, Rex Ying, Xiang Ren, William L. Hamilton, and Jure Leskovec. Graphrnn: Generating realistic graphs with deep auto-regressive models. In *ICML*, 2018.

Chiyuan Zhang, Daphne Ippolito, Katherine Lee, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. Counterfactual memorization in neural language models. In *NeurIPS*, 2023.

Zaixi Zhang, Qi Liu, Zhenya Huang, Hao Wang, Chengqiang Lu, Chuanren Liu, and Enhong Chen. Graphmi: Extracting private graph data from graph neural networks. In *IJCAI*, 2021.

Zhongjian Zhang, Xiao Wang, Mengmei Zhang, Jiarui Tan, and Chuan Shi. Toward graph-tokenizing large language models with reconstructive graph instruction tuning. In *ACM WWW*, 2026.

A Experimental Details

A.1 Dataset and Graph Construction

We evaluate on Cora, PubMed, CiteSeer, and an ogbn-products subset. For each dataset, we construct connected induced target graphs with $n \in \{5, 10, 20, 40\}$ nodes. Each dataset contains 24 target graphs, distributed as $9/9/3/3$ across the four graph sizes, together with one size-matched unexposed control graph for each size. We allocate more targets to the two smaller graph sizes and fewer to the larger sizes to balance graph-scale coverage with the higher per-target training and evaluation cost of larger graphs. Since the allocation is unequal across sizes, aggregate results are computed by first averaging within each size stratum and then weighting the four strata equally. Target and control graphs are node-disjoint within each dataset. Allocated target and control identities are excluded from filler sampling so that target exposure is not reintroduced through background training records. All graph allocations are fixed before training and reused across exposure levels.

A.2 Exposure Protocol and Training Configuration

For each target graph G, the exposure level e controls how often target-derived records appear during training, replacing matched filler records to keep the total training budget fixed. One exposure includes a positive adjacency fact for every edge in G and an equal number of two-hop non-edge facts. Thus, a graph with $|E|$ edges contributes $2e|E|$ target-derived records at exposure level e. Each record pairs a binary adjacency question with the candidate nodes' textual descriptions and two 15-node local ego graphs. Training loss is applied only to the answer token. Repeated exposures retain the node pair and label while allowing the local graph context to vary. Filler records are matched by label, graph distance, and input format. Evaluation queries all candidate pairs, including those used in training, but removes graph context entirely or masks the queried edge as described in Appendix A.3. The main experiments use $e \in \{0, 20, 60\}$, and the fine-grained exposure analysis uses $e \in \{0, 10, 20, 30, 40, 50, 60\}$.

Across models, we freeze the graph encoder and train the graph projector together with LoRA adapters on the LLM backbone. We use AdamW with learning rates 2×10^{-3} for the projector and 2×10^{-4} for LoRA, rank 16, scaling factor 32, and dropout 0.05, applied to the attention and MLP projections. We use $60n$ optimizer steps, 300, 600, 1200, and 2400 steps for $n = 5, 10, 20, 40$, respectively, and clip the gradient norm at 1.0. We use training seed 13, with seeds 29 and 47 additionally used for the training-seed robustness analysis. Evaluation uses deterministic greedy decoding with a maximum of four newly generated tokens. GraphGPT-7B, Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct experiments run on 8 NVIDIA H200 GPUs (140 GB each) on a single node. The fine-grained sweep uses 16 H200 GPUs and takes roughly 6 hours, while the additional training-seed runs use 32 H200 GPUs and take roughly 3 hours.

A.3 Query Construction and Context Conditions

For every unordered candidate pair $\{u, v\}$, the model receives a binary adjacency query asking whether the two nodes are directly connected. The query template and decoding configuration are fixed across exposure levels. We evaluate two graph-context settings: no graph context and masked graph context, and use a fixed parsing rule to convert model generations into binary edge decisions.

No graph context: The attacker queries the model using only the identities and textual descriptions of the candidate nodes. Graph tokens and adjacency information are removed from the model input. This is the stricter setting: any recovered relation must be elicited from the model using node information alone, without inference-time structural cues.

Masked graph context: For each candidate pair, the evaluation pipeline independently constructs local graph context from the dataset topology. This pair-specific context supplies surrounding structure while withholding the queried relation. For a positive pair, both (u, v) and (v, u) are removed from the materialized adjacency before the model input is formed. Non-edge queries use the same construction without introducing artificial edges. The reconstructed edge set contains only the model's parsed positive decisions.

Response Parsing: Generation uses deterministic greedy decoding. We first accept an exact binary response and otherwise extract the first standalone binary digit. Responses containing no valid bi-

nary decision remain unparsed. During graph assembly, only successfully parsed positive decisions are added to $\widehat{E}_e$; an unresolved pair therefore contributes no predicted edge but is not counted as a correct negative decision. FGW and Reach_t are computed on the resulting reconstructed graph. For ExactGraph, any unresolved pair makes the reconstruction inexact. ExactGraph Rate is computed over all target graphs rather than only those with fully parsed responses. Formally, let $A_e^{(i)} = 1$ indicate that all candidate-pair responses for target i are successfully parsed. The per-graph ExactGraph indicator implemented in evaluation is

$$\mathbb{I}_{\mathrm{Exact}}^{(i)}(e) = \mathbb{I}\Big[A_e^{(i)} = 1 \ \wedge \ \widehat{E}_e^{(i)} = E^{(i)}\Big]. \tag{10}$$

Thus, the edge-set equality in equation 5 is applied only when all candidate-pair responses are parsed. Otherwise, the graph is counted as inexact.

A.4 Reference Predictors and Evaluation

We use six non-LLM reference predictors to measure how much target structure can be inferred from node information or local graph statistics. With *no graph context*, (i) the **Text-similarity based** predictor scores a candidate pair using the cosine similarity of its node-text embeddings, while (ii) the **Node-attribute based** predictor learns a link classifier from paired node attributes. With *masked graph context*, we additionally use (iii) the **Common-neighbor based** predictor, which counts shared neighbors; (iv) the **Jaccard based** predictor, which normalizes this overlap by the size of the neighbor union; (v) the **Preferential-attachment based** predictor, which uses the product of endpoint degrees; and (vi) the **Attribute–structure based** predictor, which jointly uses node attributes and these graph features. Learned predictor parameters and all decision thresholds are selected using validation data disjoint from the evaluated target graphs.

For evaluation, (i) **Identity-aware edge recovery** reports edge-level precision, recall, and F1 under the fixed candidate-node correspondence. These metrics directly compare predicted and target relations between the same node identities, providing an identity-aware measure of partial recovery:

$$P_E = \frac{|\widehat{E} \cap E|}{|\widehat{E}|}, \qquad R_E = \frac{|\widehat{E} \cap E|}{|E|}, \qquad F_{1,E} = \frac{2P_E R_E}{P_E + R_E}. \tag{11}$$

When $\widehat{E} = \varnothing$, we set edge precision and edge F1 to zero. Every evaluated target graph contains at least one edge, so edge recall remains well defined and is also zero in this case.

(ii) **Graph edit distance (GED)** reduces to the number of edge insertions and deletions required to transform the reconstructed graph into the target, with lower values indicating fewer identity-aware reconstruction errors:

$$\mathrm{GED}(\widehat{G}, G) = |\widehat{E} \triangle E|. \tag{12}$$

(iii) **Gromov–Wasserstein (GW) and fused Gromov–Wasserstein (FGW)** provide complementary graph-level measures. We use GW distance to compare graph structure without node-text feature and fused Gromov–Wasserstein (FGW) distance to jointly compare node semantics and graph structure. Let C^{sem} denote cosine distances between node-text embeddings, and let D and $\widehat{D}$ denote the all-pairs shortest-path distance matrices of the target and reconstructed graphs. For a graph with n candidate nodes, reachable entries are their unweighted hop distances, while each unreachable node pair is assigned the finite cost $n + 1$ before GW or FGW is evaluated. The target graphs are connected, but this rule is required for potentially disconnected reconstructions. These shortest-path costs are not rescaled across graph sizes. FGW is computed as

$$\mathrm{FGW}_\alpha = \min_{T \in \Pi(a,b)} (1-\alpha) \sum_{i,j} C_{ij}^{\mathrm{sem}} T_{ij} + \alpha \sum_{i,i',j,j'} \left| D_{ii'} - \widehat{D}_{jj'} \right|^2 T_{ij} T_{i'j'}, \tag{13}$$

where T is a transport coupling and $\alpha \in [0, 1]$ balances semantic and structural costs. We use $\alpha = 0.5$ for FGW in the main experiments, while GW corresponds to the structural term alone. We retain GW as a distance, for which lower is better, and convert FGW distance to the bounded similarity in equation 7, for which higher is better. Although the candidate-node identities are known and shared between $\widehat{G}$ and G, we retain the standard optimal-transport formulation of GW and FGW as a complementary graph-level measure. Since this comparison can tolerate node permutations, we additionally report identity-aware edge F1 and GED to verify recovery under the fixed node correspondence.

A.5 Aggregation and Statistical Analysis

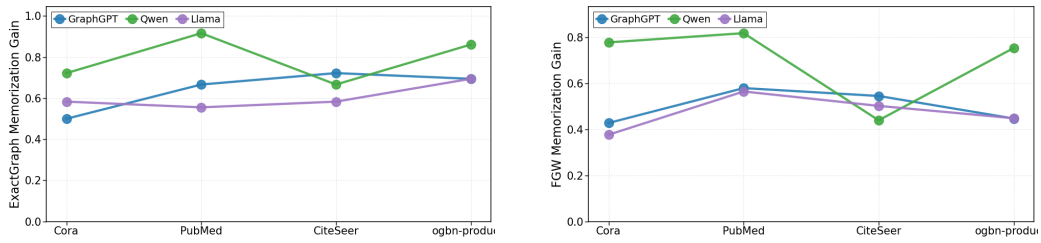
Because the target allocation is $9/9/3/3$ across $n \in \{5, 10, 20, 40\}$, aggregate quantities give equal weight to the four graph-size strata rather than pooling all target graphs directly. For exposure attribution, the target and control changes are first computed within each size and then averaged across sizes. Uncertainty for $\Gamma_R(e)$ is estimated with a stratified graph-level bootstrap. Within each graph-size stratum, target graphs are resampled with replacement while the $e = 0$ and $e > 0$ measurements for the same target remain paired. The same size-matched control is held fixed within a bootstrap draw, and the four size strata are then weighted equally. We use 10,000 bootstrap samples and report 95% intervals. These intervals quantify target-graph sampling uncertainty conditional on the fixed matched controls.

B Additional Memorization Quantification and Robustness

B.1 Results across Models and Datasets

Table 4: Target-specific exposure gains across models and datasets under masked and no graph context. Higher values indicate stronger target-specific recovery. Bold indicates the largest gain within each context for every dataset, exposure, and metric.

Context	Model	Exposure	Cora		PubMed		CiteSeer		ogbn-products	
			$\Gamma_{\text{ExactGraph}}$	Γ_{FGW}	$\Gamma_{\text{ExactGraph}}$	Γ_{FGW}	$\Gamma_{\text{ExactGraph}}$	Γ_{FGW}	$\Gamma_{\text{ExactGraph}}$	Γ_{FGW}
Masked graph	GraphGPT-7B	$e = 20$	0.250	0.217	0.361	0.282	0.389	0.358	**0.306**	**0.280**
		$e = 60$	0.500	0.428	0.667	0.580	**0.722**	**0.546**	0.694	0.448
	Qwen2.5-7B-Instruct	$e = 20$	**0.333**	**0.235**	0.306	0.237	0.306	0.254	0.194	0.177
		$e = 60$	**0.722**	**0.779**	**0.917**	**0.819**	0.667	0.440	**0.861**	**0.754**
	Llama-3.1-8B-Instruct	$e = 20$	**0.333**	0.184	**0.389**	**0.372**	**0.500**	**0.426**	0.250	0.188
		$e = 60$	0.583	0.377	0.556	0.565	0.583	0.503	0.694	0.449
No graph	GraphGPT-7B	$e = 20$	**0.167**	0.015	**0.250**	**0.219**	0.139	0.323	0.000	**0.112**
		$e = 60$	0.194	0.097	0.139	0.287	0.250	0.399	0.028	0.158
	Qwen2.5-7B-Instruct	$e = 20$	0.139	0.099	0.083	0.109	0.139	0.155	**0.083**	0.105
		$e = 60$	**0.528**	**0.561**	**0.417**	**0.437**	**0.306**	0.342	**0.250**	**0.208**
	Llama-3.1-8B-Instruct	$e = 20$	0.056	**0.127**	0.139	0.100	**0.389**	**0.505**	0.000	0.004
		$e = 60$	0.194	0.198	0.083	0.165	0.167	**0.465**	0.028	-0.020



(a) (b)

Figure 6: Exposure-controlled gains with masked graph context across models and datasets. (a) ExactGraph. (b) FGW.

Table 4 and Figure 6 extend the main-paper GraphGPT analysis with target-specific ExactGraph and FGW gains across three LLMs and four datasets. With masked graph context, all 48 gains are positive, and each $e = 60$ gain exceeds its $e = 20$ counterpart. Qwen2.5 has the largest ExactGraph gain, 0.917 on PubMed, and FGW gains of 0.819 on PubMed and 0.754 on ogbn-products. The largest ExactGraph gains for GraphGPT and Llama-3.1 are 0.722 and 0.694. Without graph context, mean gains on both metrics also increase from $e = 20$ to $e = 60$. Overall, these results support exposure-related recovery across model families, with effects varying by dataset and context.

B.2 Results for Reference Predictors

Table 5 reports the non-LLM reference predictors under their corresponding graph-context settings. The highest Edge F1 ranges from 0.343 to 0.395 across datasets, and the highest FGW similarity

from 0.441 to 0.570, while ExactGraph Rate is zero everywhere except a single 0.028 on ogbn-products. The three structure-only predictors produce identical binary edge sets after validation-based thresholding, so their aggregate results coincide. These predictors test whether generic node semantics and local graph statistics, without repeated target exposure, can recover the target graphs. Their moderate Edge F1 and FGW similarity show partial edge recovery, while the near-zero ExactGraph Rates show they rarely recover the complete topology. By comparison, Qwen2.5 at $e = 60$ already exceeds every reference predictor: without graph context, its ExactGraph Rate reaches 0.250–0.528 (Table 1) against zero for the reference predictors; with masked graph context, it reaches an FGW similarity of 0.824 and an ExactGraph Rate of 0.722 on Cora against 0.502 and zero for the strongest reference predictor. More broadly, every model on every dataset at $e = 60$ in Table 2 attains an ExactGraph Rate of at least 0.500, peaking at 0.917 on PubMed.

Intrinsic edge predictability alone does not explain the exposure-dependent gains. The target graphs, candidate nodes, and inference-time context remain fixed across exposure levels, so edges that are generally easy to predict would already be recoverable at $e = 0$. Such general prediction effects can appear on the matched unexposed controls, whose recovery changes are subtracted in the target-specific gain $\Gamma_R(e)$. The reference predictors and exposure-controlled comparison provide complementary evidence: generic semantic and structural cues explain part of the edge-level recovery, while repeated target exposure accounts for the additional shift toward exact whole-graph reconstruction.

Table 5: Whole-graph recovery of non-LLM reference predictors. Exact denotes ExactGraph Rate, and s_{FGW} denotes FGW similarity. Results are averaged within each graph-size stratum and then equally across $n \in \{5, 10, 20, 40\}$. Higher values are better.

Context	Predictor	Cora			PubMed			CiteSeer			ogbn-products		
		Exact	Edge F1	s_{FGW}	Exact	Edge F1	s_{FGW}	Exact	Edge F1	s_{FGW}	Exact	Edge F1	s_{FGW}
No graph	Text-similarity based	0.000	0.392	0.463	0.000	0.363	0.509	0.000	0.364	0.528	0.000	0.343	0.441
	Node-attribute based	0.000	0.394	0.484	0.000	0.365	0.446	0.000	0.395	0.570	0.000	0.334	0.417
Masked graph	Common-neighbor based	0.000	0.394	0.484	0.000	0.368	0.479	0.000	0.362	0.524	0.000	0.333	0.417
	Jaccard based	0.000	0.394	0.484	0.000	0.368	0.479	0.000	0.362	0.524	0.000	0.333	0.417
	Preferential-attachment based	0.000	0.394	0.484	0.000	0.368	0.479	0.000	0.362	0.524	0.000	0.333	0.417
	Attribute–structure based	0.000	0.394	0.502	0.000	0.256	0.121	0.000	0.378	0.521	0.028	0.319	0.297

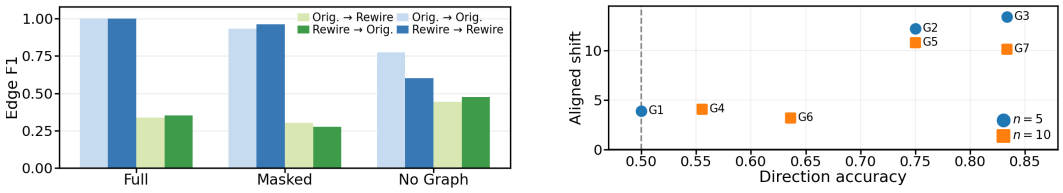
B.3 Fixed (V, X) Rewiring Control

We further test structural recovery while fixing node identities and textual information. Using GraphGPT-7B, we evaluate seven Cora target graphs, with three at $n = 5$ and four at $n = 10$. For each $G = (V, E, X)$, we construct a degree-preserving rewired counterpart $G^{\mathrm{rw}} = (V, E^{\mathrm{rw}}, X)$ with the same V and X, allowing us to vary relational structure while keeping node information unchanged. For each graph, we train matched $e = 60$ models on E and E^{rw}, sharing the $e = 0$ reference, and evaluate them under full, masked, and *no graph context*. We exchange original and rewired contexts to examine the effect of inference-time structure. Recovery is measured with ExactGraph and edge F1, while aligned log-odds on changed edge pairs are used as a diagnostic analysis.

Table 6: Recovery under fixed (V, X) rewiring across graph-context settings. Results are aggregated over seven Cora target graphs, with three graphs at $n = 5$ and four graphs at $n = 10$. **Bold** indicates the best matched-structure result within each context.

Context	Training structure	ExactGraph on E	ExactGraph on E^{rw}	Edge F1 on E	Edge F1 on E^{rw}
Full	$e = 0$	0.000	0.000	0.399	0.385
	Original E, $e = 60$	**1.000**	0.000	**1.000**	0.337
	Rewired E^{rw}, $e = 60$	0.000	**1.000**	0.353	**1.000**
Masked	$e = 0$	0.000	0.000	0.344	0.361
	Original E, $e = 60$	**0.714**	0.000	**0.935**	0.303
	Rewired E^{rw}, $e = 60$	0.000	**0.857**	0.276	**0.963**
No graph	$e = 0$	0.000	0.000	0.331	0.319
	Original E, $e = 60$	**0.286**	0.000	**0.774**	0.444
	Rewired E^{rw}, $e = 60$	0.000	**0.286**	0.474	**0.602**

Table 6 and Figure 7 show that recovery follows the relational structure encountered during training even when V and X are fixed. Under full graph context, both original- and rewired-trained models exactly recover their matched structures (100% ExactGraph), while cross-structure ExactGraph remains 0%, indicating that the recovered structure is tied to the training configuration rather



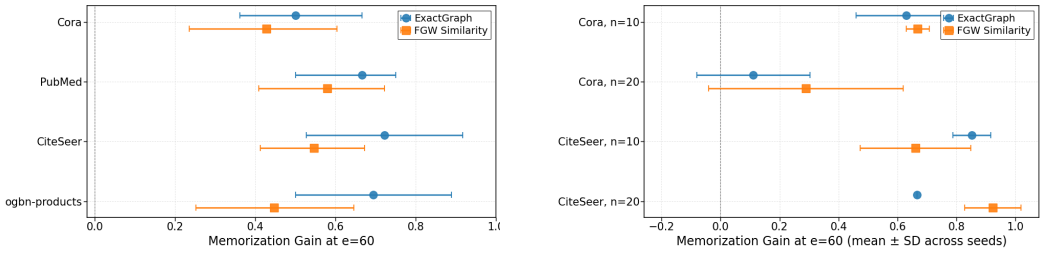
(a) (b)

Figure 7: Fixed rewiring analysis with GraphGPT-7B on Cora. (a) Matched and cross-structure edge recovery across graph-context settings. Arrows indicate training-to-evaluation structure. (b) Consistency and strength of no-graph structure-aligned prediction shifts across target graphs.

than to the shared node identities or textual information. The same separation persists with masked graph context, where matched ExactGraph reaches 71.4% for the original structure and 85.7% for the rewired structure, with edge F1 of 0.935 and 0.963. Because the queried relations are hidden at inference time, this result cannot be explained by simply reproducing the supplied graph context. With *no graph context*, recovery becomes weaker but remains structure-specific: matched ExactGraph reaches 28.6% for both structures, and predictions move toward the corresponding training structure on $61/88$ changed pairs (69.3%). Moreover, the aligned log-odds shift is positive for all seven graphs, with a pair-weighted mean of $+7.30$, showing that the effect is also visible as a graded pair-level preference rather than only through the thresholded ExactGraph metric. The progressive weakening from full to masked to no graph context suggests that inference-time graph context facilitates the readout of learned structural information, but does not fully account for it. Together, these results provide stronger evidence that structural recovery tracks the exposed relation structure itself rather than familiarity with the shared nodes and textual information alone.

B.4 Statistical and Training-Seed Robustness

We evaluate two distinct sources of variability. The stratified bootstrap in Appendix A.5 measures variation across target graphs, while a multi-seed subset evaluates sensitivity to training randomness. The training-seed analysis uses GraphGPT on Cora and CiteSeer for $n \in \{10, 20\}$, $e \in \{0, 20, 60\}$, and seeds $\{13, 29, 47\}$. Target allocation, exposure construction, optimization budget, and evaluation are fixed across seeds.



(a) (b)

Figure 8: Robustness of the exposure-dependent recovery effect. (a) Graph-level bootstrap intervals. (b) Training-seed robustness.

The bootstrap intervals in Figure 8a quantify variation across target graphs. For the main training seed, the 95% intervals at $e = 60$ remain strictly positive for both metrics on all four datasets: ExactGraph spans $[0.361, 0.667]$ on Cora and $[0.528, 0.917]$ on CiteSeer, while the corresponding FGW-similarity intervals are $[0.236, 0.604]$ and $[0.413, 0.672]$. This indicates that the observed attribution effect is not driven by a small subset of target graphs, although the interval widths reflect graph-level heterogeneity. Table 7 and Figure 8b separately evaluate sensitivity to optimization randomness. At $e = 60$, the exposure effect remains strong and relatively stable across seeds for Cora with $n = 10$ and for both CiteSeer settings. Cora with $n = 20$ shows greater variation across training seeds than the other settings, but the mean gain at $e = 60$ remains positive for both metrics.

Table 7: Training-seed sensitivity of the target-specific exposure effect at $e = 60$.

Dataset	Size	$\Gamma_{\mathrm{ExactGraph}}(e = 60)$	$\Gamma_{\mathrm{FGW}}(e = 60)$
Cora	$n = 10$	0.630 ± 0.170	0.668 ± 0.039
	$n = 20$	0.111 ± 0.192	0.289 ± 0.330
CiteSeer	$n = 10$	0.852 ± 0.064	0.661 ± 0.187
	$n = 20$	0.667 ± 0.000	0.923 ± 0.096

B.5 Robustness to Recovery Metric

ExactGraph provides a strict criterion for complete reconstruction but does not distinguish among incomplete graphs. We therefore examine the same reconstructions using identity-aware edge F1 and GED, together with topology-based GW. These metrics require no additional model queries.

Table 8: Edge F1 across models, datasets, and exposure levels. Higher values indicate better identity-aware edge recovery under masked graph context. Bold indicates the best result for each dataset.

Model	Exposure	Cora	PubMed	CiteSeer	ogbn-products
GraphGPT-7B	$e = 0$	0.143	0.099	0.113	0.134
	$e = 20$	0.409	0.450	0.564	0.576
	$e = 60$	0.615	0.667	**0.946**	0.874
Qwen2.5-7B-Instruct	$e = 0$	0.192	0.119	0.106	0.136
	$e = 20$	0.393	0.432	0.618	0.547
	$e = 60$	**0.885**	**0.917**	0.910	**0.902**
Llama-3.1-8B-Instruct	$e = 0$	0.151	0.133	0.143	0.128
	$e = 20$	0.425	0.547	0.628	0.642
	$e = 60$	0.604	0.820	0.801	0.883

Table 8 reports edge F1 across all three models and four datasets. We further report GED and GW for every model–dataset combination. GED preserves node identities and measures edge-edit errors, whereas GW evaluates structural similarity independently of node-text features.

Table 9: Graph recovery measured by GED and GW across models and datasets. Results use *masked graph context*.

Model	Exposure	Cora		PubMed		CiteSeer		ogbn-products	
		GED ↓	GW ↓	GED ↓	GW ↓	GED ↓	GW ↓	GED ↓	GW ↓
GraphGPT-7B	$e = 0$	35.972	357.439	39.833	322.347	59.083	221.628	23.639	417.332
	$e = 20$	30.361	412.678	26.333	435.559	21.194	391.319	16.083	271.328
	$e = 60$	22.000	372.484	20.833	377.074	**3.250**	**86.262**	5.333	134.208
Qwen2.5-7B-Instruct	$e = 0$	33.500	395.774	32.056	418.864	47.556	324.246	23.056	422.156
	$e = 20$	27.806	433.193	26.028	429.133	17.333	299.133	14.056	281.789
	$e = 60$	**6.500**	**49.714**	**4.333**	**113.284**	6.167	138.316	**3.778**	**105.882**
Llama-3.1-8B-Instruct	$e = 0$	32.639	425.744	35.333	376.562	54.000	222.540	42.750	260.683
	$e = 20$	27.750	430.007	24.139	405.573	20.278	379.555	11.278	242.472
	$e = 60$	22.500	376.517	9.750	233.949	11.000	237.826	4.472	114.787

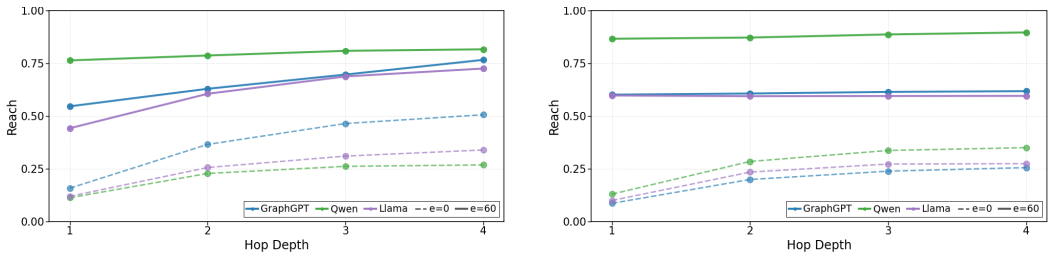
Tables 8 and 9 provide identity-aware and topology-only checks of the recovery pattern. Edge F1 increases from $e = 0$ to $e = 60$ in all 12 model–dataset settings; for example, GraphGPT rises from 0.113 to 0.946 on CiteSeer, and Qwen rises from 0.119 to 0.917 on PubMed. Raw GED also decreases in every setting, with GraphGPT–CiteSeer falling from 59.083 to 3.250. GW, in contrast, is less uniformly monotone at $e = 20$ and even remains above its $e = 0$ value for GraphGPT on Cora and PubMed at $e = 60$, because it couples graphs purely through all-pairs shortest paths, so a few misplaced edges in a partially-reconstructed graph can distort this structure even as identity-aligned overlap with the target keeps improving. FGW avoids this by anchoring the coupling with a semantic term computed from node-text embeddings, which stay fixed across exposure levels; as a result, FGW similarity rises monotonically across $e \in \{0, 20, 60\}$ in every model–dataset setting in Tables 1 and 2, and we use it as our main partial-recovery metric for this reason, keeping GW here

only as a topology-only reference. Overall, the recovery conclusion is best supported by Edge F1 and GED, which improve consistently across all settings, rather than by any single metric.

C Additional Graph Recovery Results

C.1 Local-Graph Recovery

We extend the local-recovery analysis to all three models and both graph-context settings on Cora. Since local recovery can vary across start nodes, we first average $\mathrm{Reach}_t(s)$ over all start nodes within each graph, average targets within each size, and then give the four size strata equal weight. We show hops 1–4, which are defined for every evaluated graph size.



(a) (b)

Figure 9: Additional local-graph recovery results. (a) No graph context. (b) Masked graph context.

Figure 9 extends the local-recovery comparison beyond the representative setting in the main paper. At hop 1 without graph context, the $e = 60$ gain over $e = 0$ is 0.388, 0.651, and 0.323 for GraphGPT, Qwen, and Llama, respectively; the corresponding masked-context gains are 0.515, 0.737, and 0.498. The effect persists at hop 4: Qwen improves from 0.269 to 0.817 without graph context and from 0.351 to 0.898 with masked context. Masking therefore facilitates local readout but is not necessary for it, while the variation across models shows that exposure sensitivity is strongest for Qwen rather than being a fixed consequence of hop depth alone.

C.2 Complete Exposure Sweep by Graph Size

We complement the main exposure analysis with a fine-grained sweep across graph sizes for GraphGPT on Cora with *masked graph context*. The overlapping $e \in \{0, 20, 60\}$ points use the same accepted checkpoints and target reconstructions as the main results; the remaining exposure levels complete the sweep. Since the attribution analysis has already established the target-specific exposure effect, this experiment focuses on when partial and exact recovery emerge as exposure increases and how this trajectory changes with graph size.

Table 10: ExactGraph Rate and FGW similarity s_{FGW} across graph sizes and fine-grained exposure levels. Entries are rounded individually to three decimals; size-balanced results in the main table are computed from the unrounded target-level values.

Size	Metric	Exposure						
		$e=0$	$e=10$	$e=20$	$e=30$	$e=40$	$e=50$	$e=60$
$n=5$	ExactGraph ↑	0.000	0.111	1.000	1.000	1.000	0.778	0.889
	s_{FGW} ↑	0.443	0.523	1.000	1.000	1.000	0.900	0.901
$n=10$	ExactGraph ↑	0.000	0.000	0.000	0.444	0.333	0.556	0.778
	s_{FGW} ↑	0.142	0.316	0.348	0.652	0.663	0.665	0.891
$n=20$	ExactGraph ↑	0.000	0.000	0.000	0.000	0.000	0.000	0.333
	s_{FGW} ↑	0.006	0.006	0.012	0.008	0.006	0.240	0.644
$n=40$	ExactGraph ↑	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	s_{FGW} ↑	0.003	0.001	0.001	0.001	0.001	0.001	0.012

Table 10 reveals a clear scale-dependent recovery trajectory. For $n = 5$, ExactGraph Rate reaches 1.000 at $e = 20$, while FGW similarity remains around 0.900 at $e = 50$ and $e = 60$ even when

ExactGraph Rate fluctuates, reflecting the sensitivity of exact recovery to a small number of edge errors. Recovery develops more gradually at $n = 10$, where ExactGraph Rate increases from 0 to 0.778 and FGW similarity from 0.142 to 0.891. At $n = 20$, FGW similarity reaches 0.240 at $e = 50$ while ExactGraph Rate remains zero; at $e = 60$, they rise to 0.644 and 0.333, respectively. Larger graphs require substantially more exposure to enter the same recovery regime. Overall, increasing graph size shifts recovery toward higher exposure and extends the transition from partial structural recovery to exact reconstruction.

D Additional Analysis of Training-Graph Stealing

D.1 Attack Cost and Query Complexity

The main paper reports how missing and spurious relations evolve with training exposure. Formally, for a reconstructed edge set $\widehat{E}$ and target edge set E, we define the two normalized error components as

$$\mathrm{Miss}(\widehat{G}, G) = \frac{|E \setminus \widehat{E}|}{|E|}, \qquad \mathrm{Spurious}(\widehat{G}, G) = \frac{|\widehat{E} \setminus E|}{|E|}. \tag{14}$$

Both quantities are computed per graph before aggregation across target graphs. This common denominator makes the two components directly comparable: Miss measures unrecovered target structure, whereas $\mathrm{Spurious}$ measures relations introduced by the attack. The attack queries every unordered pair of candidate nodes. Reconstructing a graph with n candidate nodes therefore requires

$$Q(n) = \binom{n}{2} \tag{15}$$

black-box queries. For $n \in \{5, 10, 20, 40\}$, this corresponds to 10, 45, 190, and 780 queries, respectively. Once the recovered edge set is assembled into $\widehat{G}$, multi-hop neighborhoods and reachable regions are obtained through offline graph traversal and require no additional victim-model queries. Thus, the query cost is determined entirely by candidate-pair enumeration, whereas multi-hop reconstruction adds only offline graph-processing cost.

D.2 Edge Vulnerability And Error Decomposition

We extend the main-paper attack analysis to Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct on the same Cora and PubMed targets. Following the same protocol, true edges are ranked within each target graph by their shortest-path usage and divided into four near-equal groups. Recovery and normalized edge errors are computed per graph before averaging across the 24 targets in each model–dataset–exposure setting.

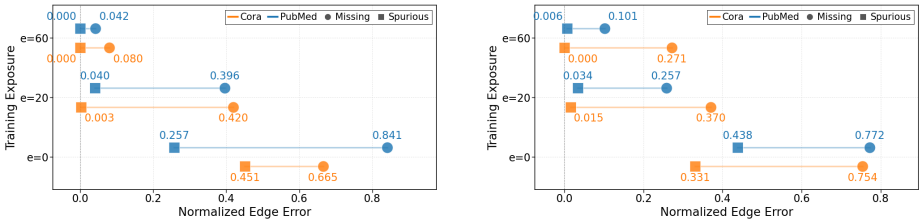
Table 11 shows that high-exposure recovery extends across structural positions for both model families. At $e = 60$, Qwen's four recovery rates range from 0.906 to 0.944 on Cora and reach 0.958 uniformly on PubMed. Llama reaches values from 0.708 to 0.753 and from 0.859 to 0.915, respectively. Range does not decrease consistently, as Llama on PubMed has a smaller value at $e = 20$ than at $e = 60$. Exposure therefore broadens recovery without creating a stable preference for any structural group. Figure 10 shows that remaining errors are mainly missing relations. From $e = 0$ to $e = 60$, Qwen's missing-edge error falls from 0.665 to 0.080 on Cora and from 0.841 to 0.042 on PubMed, while its spurious error reaches zero on both datasets. Llama retains a missing-edge error of 0.271 on Cora at $e = 60$, with zero spurious error. On PubMed, the two errors fall to 0.101 and 0.006. Across models, ExactGraph failures therefore resemble incomplete target subgraphs with few fabricated relations, although the recovered structure still varies across models and datasets.

E Limitations and Future Work

Our study focuses on graph-structure recovery in the setting where the candidate nodes and their textual information are known while the queried relations remain hidden. This setting isolates relational memorization and enables controlled evaluation, while future work could extend the attack to incomplete candidate sets and improve its scalability. Our analysis is primarily behavioral and characterizes when training graph structure becomes recoverable and how such recovery can be

Table 11: Mean edge-recovery rates by shortest-path usage for Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct on Cora and PubMed with masked graph context.

Model	Dataset	Exposure	Shortest-path usage				Range ↓
			Lowest	*Low*	*High*	*Highest*	
Qwen2.5-7B-Instruct	Cora	$e = 0$	0.338	0.385	0.375	0.240	0.145
		$e = 20$	0.581	0.562	0.583	0.594	**0.031**
		$e = 60$	0.921	0.906	0.919	0.944	0.039
	PubMed	$e = 0$	0.233	0.118	0.089	0.189	0.143
		$e = 20$	0.580	0.594	0.631	0.623	0.051
		$e = 60$	0.958	0.958	0.958	0.958	**0.000**
Llama-3.1-8B-Instruct	Cora	$e = 0$	0.244	0.198	0.260	0.306	0.108
		$e = 20$	0.640	0.615	0.656	0.611	0.045
		$e = 60$	0.725	0.753	0.708	0.729	**0.044**
	PubMed	$e = 0$	0.342	0.196	0.174	0.193	0.168
		$e = 20$	0.750	0.750	0.729	0.737	**0.021**
		$e = 60$	0.906	0.915	0.859	0.911	0.056



(a) (b)

Figure 10: Normalized missing and spurious edge errors across exposure for (a) Qwen2.5-7B-Instruct and (b) Llama-3.1-8B-Instruct on Cora and PubMed.

exploited through black-box queries. The next step is to investigate the internal mechanisms underlying structural memorization and to develop mitigation strategies that reduce unintended relational leakage while preserving graph-reasoning capabilities.

F Reproducibility Statement

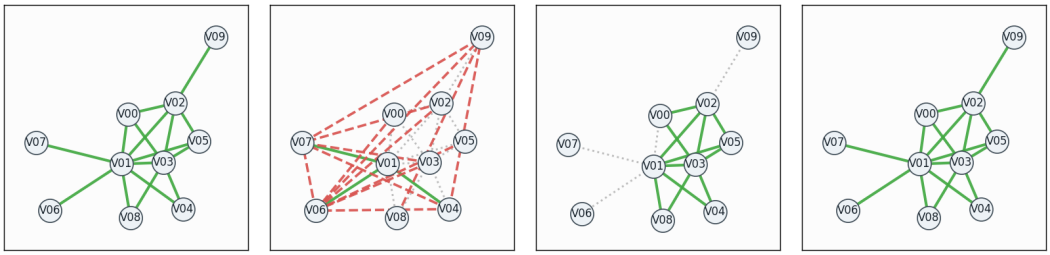
We provide the formulation of graph structural memorization, the exposure-controlled evaluation protocol, the training-graph stealing procedure, and the recovery criteria in Section 3. Section 4.1 describes the datasets, model backbones, exposure levels, graph-context settings, baselines, and main evaluation setup. Appendix A further documents target and control graph construction, exposure and training configuration, query construction and response parsing, baseline and evaluation implementation, and aggregation and statistical analysis. The remaining appendix sections provide additional details on the rewiring control, training-seed and metric robustness, local and exposure-scale analyses, and attack query complexity. These descriptions, together with the reported experimental settings and evaluation procedures, are intended to make the complete experimental pipeline transparent and reproducible. Our code will be publicly available upon paper acceptance.

G Qualitative Analysis

G.1 Training-Graph Reconstruction Case Study

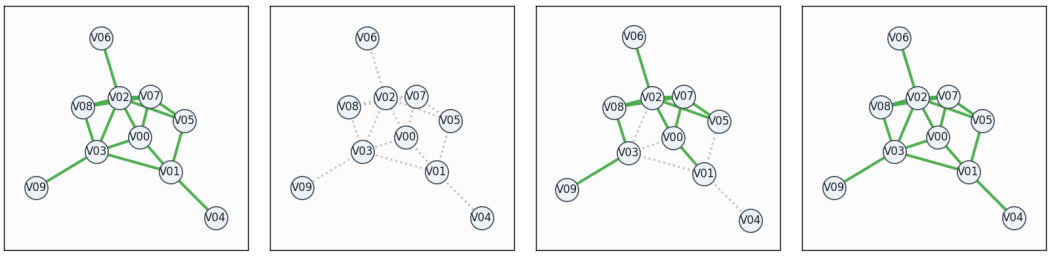
Figures 11–13, 14–16, and 17–19 provide qualitative counterparts to the aggregate results. Across GraphGPT cases, recovery is not simply a monotonic addition of correct edges: reconstructions may evolve from empty or error-dominated graphs to coherent partial topologies before matching the target. With node positions fixed, the panels show that convergence involves both recovering missing relations and removing spurious ones. The Qwen and Llama cases show similar behavior across model families and datasets. Some reconstructions improve progressively, whereas others remain stable at intermediate exposure before a sharper transition or retain only a few residual er-

rors near exact recovery. Thus, exposure-dependent recovery can be gradual or abrupt. From an attacker's perspective, partial reconstructions may already reveal connected regions, while missing and spurious edges determine the reliability of the stolen topology.



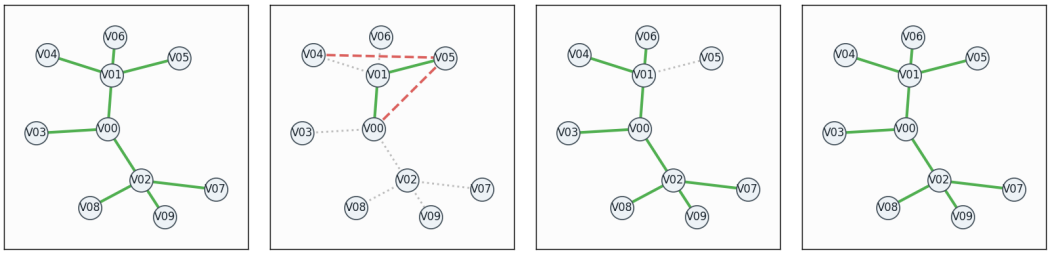
(a) Ground truth (b) $e = 0$ (c) $e = 20$ (d) $e = 60$

Figure 11: GraphGPT Case 1 on Cora: gradual transition from weak partial reconstruction to exact recovery as exposure increases.



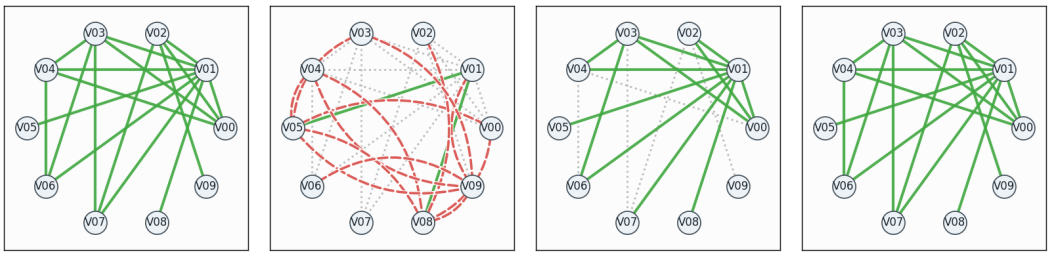
(a) Ground truth (b) $e = 0$ (c) $e = 20$ (d) $e = 60$

Figure 12: GraphGPT Case 2 on Cora: emergence of the target structure from no recovered relations to exact reconstruction.



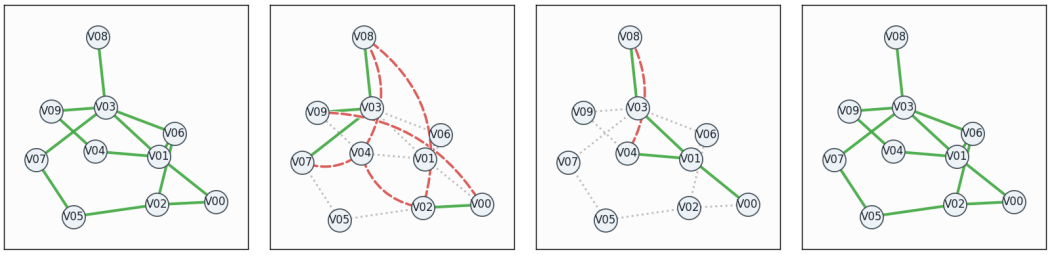
(a) Ground truth (b) $e = 0$ (c) $e = 20$ (d) $e = 60$

Figure 13: GraphGPT Case 3 on Cora: partial recovery at zero exposure followed by near-exact and exact reconstruction.



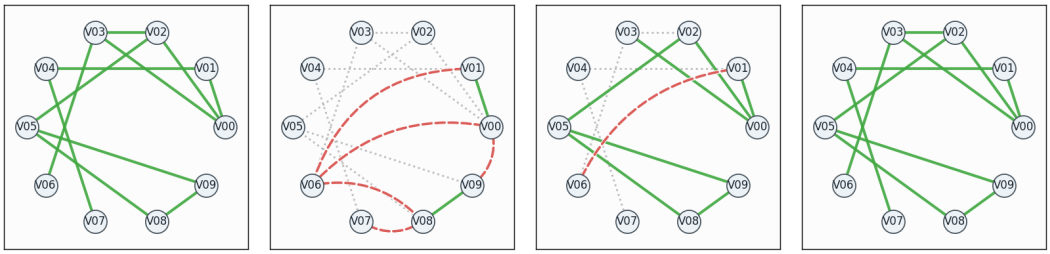
(a) Ground truth (b) $e = 0$ (c) $e = 20$ (d) $e = 60$

Figure 14: Qwen Case 1 on Cora: progressive recovery from a weak initial reconstruction.



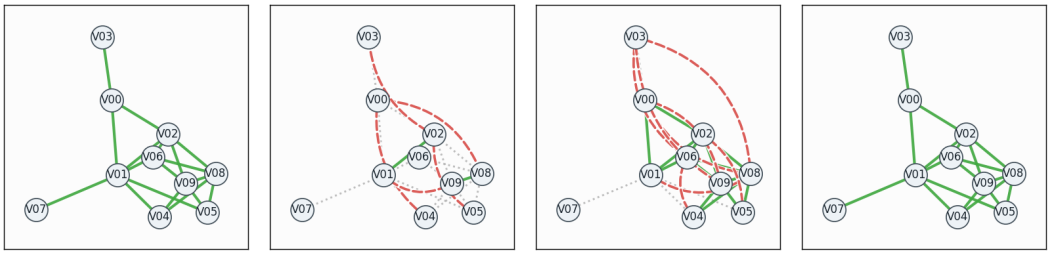
(a) Ground truth (b) $e = 0$ (c) $e = 20$ (d) $e = 60$

Figure 15: Qwen Case 2 on PubMed: limited intermediate improvement followed by exact recovery.



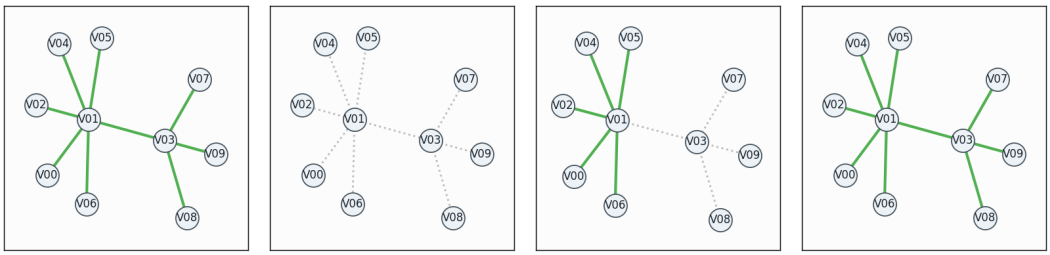
(a) Ground truth (b) $e = 0$ (c) $e = 20$ (d) $e = 60$

Figure 16: Qwen Case 3 on CiteSeer: steady transition from partial to exact reconstruction.



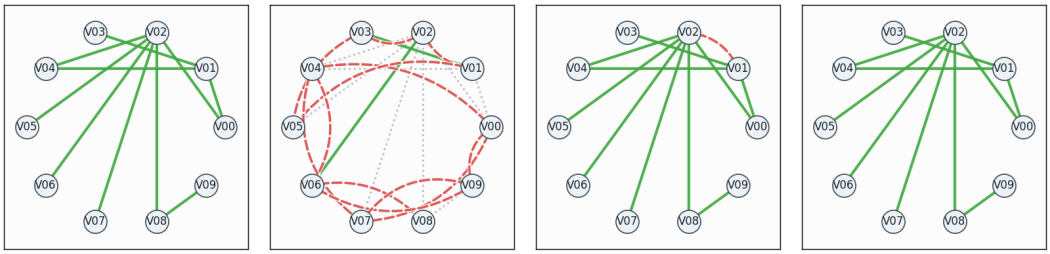
(a) Ground truth (b) $e = 0$ (c) $e = 20$ (d) $e = 60$

Figure 17: Llama Case 1 on PubMed: gradual reduction of missing and spurious relations.



(a) Ground truth (b) $e = 0$ (c) $e = 20$ (d) $e = 60$

Figure 18: Llama Case 2 on ogbn-products: target structure emerging from an empty reconstruction.



(a) Ground truth (b) $e = 0$ (c) $e = 20$ (d) $e = 60$

Figure 19: Llama Case 3 on CiteSeer: near-exact intermediate recovery followed by exact reconstruction.