Model Extraction Without Graphs Structure: How Homophily Drives Attack Effectiveness

Xuehai Wu
Rutgers University
xuehai.wu@rutgers.edu

Qiong Wu*
AT&T CDO
qiongwu9@gmail.com

Abstract

Model extraction attacks on Graph Neural Networks (GNNs) have traditionally been studied under assumptions of high homophily and full access to the graph structure, which oversimplifies real-world attack scenarios. In practice, attackers often lack access to the original graph topology, making structure-free settings more realistic and critical to study, as they reflect common constraints in privacy-sensitive or proprietary graph-based systems. This study investigates model extraction under such constraints and identifies graph homophily as the central factor driving attack success. Through extensive empirical evaluation, we show that homophily between the training and test node partitions is the primary driver of extraction success: higher homophily markedly increases attack fidelity. Counterintuitively, we find that heterophily-resilient GNN architectures are more vulnerable to these attacks than homophily-sensitive models. Furthermore, while Graph Structure Learning (GSL) methods can improve extraction fidelity by inferring proxy graph structures, their benefits are strongly dependent on underlying homophily levels and are most pronounced in high-homophily scenarios. Our findings establish homophily as a central factor in GNN security, providing new insights for designing robust architectures and defenses in structure-limited environments.

1 Introduction

Graph Neural Networks (GNNs) have emerged as powerful tools for learning on graph-structured data, achieving state-of-the-art performance across various tasks [1, 2, 3, 4]. Their wide deployment in critical applications has made understanding their security and privacy implications increasingly important [5, 6, 7], particularly the threat of model extraction attacks [8, 9, 10, 11]. In these attacks, an adversary aims to reconstruct a surrogate model that approximates the functionality of a target GNN, thereby compromising intellectual property and facilitating downstream attacks. Existing works [9, 12, 13, 14, 15] on GNN model extraction have predominantly assumed the attacker has full or partial access to the graph structure, leveraging node connectivity to achieve high-fidelity extraction. Recently, however, growing attention has been focused on more realistic structure-free settings [13], where the underlying topology is private. In these scenarios, attackers must rely solely on node features and model outputs, necessitating new strategies for querying and surrogate training. This has spurred interest in Graph Structure Learning (GSL) methods [15, 16, 17, 18, 19, 20, 21, 22], which infer latent graph structures from feature information alone to enhance extraction performance under these constraints [13, 14, 23, 24], as illustrated in the Figure 1.

While prior approaches have addressed the absence of structural information in model extraction, the role of target graph homophily in shaping attack performance remains poorly understood. Homophily, which is the tendency of connected nodes to share labels or features [25, 26], directly

*Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: NeurIPS 2025 Workshop on New Perspectives in Advancing Graph Machine Learning.

affects how much information a target GNN encodes in node representations. In a high-homophily graph, label signals propagate smoothly across neighborhoods, meaning that even without access to the original topology, an attacker's queries to the target model can capture rich correlations between node features and labels [27, 28]. In contrast, low-homophily (heterophilic) graphs, where connected nodes have dissimilar labels, pose a significant challenge for extraction because node features alone provide weaker cues about neighborhood label distributions [3, 27, 26]. Without access to the graph structure, an attacker loses this critical source of information, making surrogate training far less effective. Thus, homophily levels go beyond being a general property of graphs to fundamentally influence the success of extraction attacks in structure-limited settings. Traditional GNNs such as Graph Convolutional Networks (GCNs) rely on neighborhood aggregation [2, 29, 30], effectively smoothing node features to exploit homophily. However, this mechanism fails under heterophily, leading to reduced model accuracy and potentially easier extraction if the model collapses to trivial solutions. In contrast, advanced architectures such as Frequency Adaptive GCNs (FAGCNs)[29] are specifically designed to handle heterophilic graphs by decomposing node features into low- and high-frequency components and adaptively combining them through learnable gating mechanisms. Such models maintain higher accuracy in heterophilic settings, but their complex adaptation may introduce new challenges or opportunities for extraction attacks.

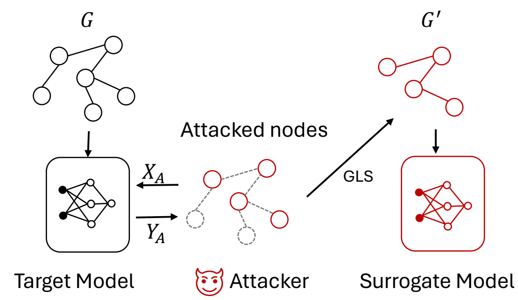

Figure 1: A general framework of model extraction attacks without graph structure. The attacker first probes the target model (trained on the private (G, X, Y)) by sending queries with node features X_A. The model's predictions are collected to form a new set of labels Y_A. Without access to the true graph structure G, the attacker uses a GSL method to infer a proxy structure G' and the trains a surrogate model on this completely synthetic graph (G', X_A, Y_A).

In this work, we investigate structure-free model extraction attacks, where the attacker has no access to the graph topology, and analyze how graph homophily and the target model's architectural capacity to handle heterophily collectively govern attack performance. Through comprehensive experiments spanning ten datasets and six baseline methods, across a range of homophily levels and model architectures, our results demonstrate that: (1) Train-test homophily is a pivotal factor governing extraction fidelity, with higher homophily substantially favoring the attacker. (2) Contrary to what might be expected, target models designed for heterophily resilience, are generally more vulnerable to these attacks than standard GCNs. (3) The fidelity gains from using GSL methods to infer missing structure are highly homophily-dependent, offering significant improvements primarily in high-homophily settings.

These findings underscore homophily not only as a fundamental property shaping graph learning but also as a critical security determinant, revealing nuanced and architecture-specific vulnerabilities in GNNs. In particular, our results provide new insights into GNN vulnerabilities under practical, structure-limited constraints by highlighting how the key factors of graph homophily and target model architecture jointly govern extraction susceptibility. Both dimensions must therefore be carefully considered when designing effective attacks and developing robust defenses against model extraction. The main contributions of this paper are summarized as follows:

- To the best of our knowledge, we are the first to establish that train-test homophily is a pivotal factor that substantially governs the success of structure-free model extraction attacks on GNNs.
- We provide the novel and counterintuitive finding that GNN models specifically designed to be resilient to heterophily are more vulnerable to these extraction attacks than standard GCNs; and we further demonstrate that the effectiveness of using GSL methods to infer missing topology for the attack is highly dependent on homophily, offering significant gains mostly in high-homophily settings.
- We conduct extensive experiments across multiple benchmark datasets and GNN architectures, providing comprehensive empirical evidence to support our findings and ensure the robustness of our conclusions.

2 Background and Notations

Graph Neural Networks. We model the target system as an attributed graph $G = (V, E, X, Y)$, where V is the set of n nodes, $E \subseteq V \times V$ the original edge set, and $X = \{x_i \mid v_i \in V\}$ the node-attribute set with $x_i \in \mathbb{R}^d$. Each node $v_i \in V$ is associated with a label $y_i \in Y$ drawn from a finite label space Y. A target GNN is parameterized by θ and defines a mapping $f_\theta : V \to Y$ that produces a prediction $p_i = f_\theta(v_i) \in P$ for every node. We assume the attacker has black-box query access to f_θ but no knowledge of the true edge set E.

Graph Homophily (Total Homophily). Let $G = (V, E)$ be a graph with node labels $\{y_u \mid u \in V\}$. The (total) homophily level [15, 25, 27] $h \in [0, 1]$ is defined as

$$h_{\text{total}} = \frac{|\{(u, v) \in E \mid y_u = y_v\}|}{|E|}. \tag{1}$$

A high value ($h_{\text{total}} \approx 1$) indicates that connected nodes are likely to share the same label, whereas a low value ($h_{\text{total}} \approx 0$) reflects heterophily.

Train–Test Homophily. Let $V_{\text{train}} \subseteq V$ and $V_{\text{test}} \subseteq V$ denote the sets of training and test nodes, respectively. The train–test homophily is computed over edges connecting training and test nodes:

$$h_{\text{train-test}} = \frac{|\{(u, v) \in E \mid u \in V_{\text{train}},\, v \in V_{\text{test}},\, y_u = y_v\}|}{|\{(u, v) \in E \mid u \in V_{\text{train}},\, v \in V_{\text{test}}\}|}. \tag{2}$$

Model Extraction Attack Against Graph Learning Model. The attacker creates a surrogate dataset $G' = (V_A, E_A^*, X_A)$ consisting of an attack node set $V_A \subseteq V$ selected by the attacker, synthetic edges $E_A^* \subseteq V_A \times V_A$ (possibly different from the true edges $E_A \subseteq E$ among V_A), and the corresponding attributes $X_A = \{x_i \mid v_i \in V_A\}$. The attacker queries the target model to obtain predictions $P_A = \{f_\theta(v_i) \mid v_i \in V_A\}$, and trains an extraction model $f_{\theta'}$ on the surrogate data (G', P_A). The goal is to minimize the generalization gap $\mathbb{E}_{v \sim V}[\ell(f_{\theta'}(v), f_\theta(v))]$ under the constraint that E is unknown.

Fidelity. Given a test node set $V_{\text{test}} \subseteq V$, fidelity measures the agreement between the surrogate model $f_{\theta'}$ and the target model f_θ on V_{test}. Formally, fidelity [13, 31, 32, 8, 11] is defined as

$$\text{Fid}(f_{\theta'}, f_\theta) = \frac{1}{|V_{\text{test}}|} \sum_{v \in V_{\text{test}}} \mathbf{1}[\arg\max f_{\theta'}(v) = \arg\max f_\theta(v)]\,, \tag{3}$$

where $\mathbf{1}[\cdot]$ denotes the indicator function. We report the mean fidelity together with its standard deviation across multiple runs.

Graph Structure Learning. To compensate for the absence of the true edge set E, the attacker may optionally refine the synthetic edges E_A^* via a GSL module. Concretely, the GSL module takes the surrogate attributes X_A and an initial synthetic edge set $\tilde{E}_A^{(0)}$ (commonly a k-nearest-neighbor graph or a fully-connected graph over V_A) and learns an optimized structure E_A^* by minimizing

$$\mathcal{L}_{\text{GSL}} = \mathcal{L}_{\text{Align}}(Z_A^*, P_A) + \lambda \mathcal{L}_{\text{Struct}}(E_A^*, Z_A^*, (V_A, \tilde{E}_A^{(0)}, X_A)). \tag{4}$$

Here, Z_A^* denotes the node representations produced by an auxiliary GNN g_ϕ operating on (V_A, E_A^*, X_A), $\mathcal{L}_{\text{Align}}$ measures the discrepancy between g_ϕ's predictions and the target model's predictions $P_A = \{f_\theta(v_i) \mid v_i \in V_A\}$, and $\mathcal{L}_{\text{Struct}}$ imposes structural priors (such as sparsity, smoothness, feature-structure consistency, or homophily) on the learned edges E_A^* and representations Z_A^*. The hyper-parameter λ balances the fidelity of target model imitation (via $\mathcal{L}_{\text{Align}}$) against structural plausibility (via $\mathcal{L}_{\text{Struct}}$). The resulting E_A^* is then used in the surrogate dataset $G' = (V_A, E_A^*, X_A)$ for downstream model extraction.

3 Research Questions

While recent work has explored structure-free model extraction attacks on GNNs, many fundamental aspects of their efficacy remain poorly understood. In particular, the role of graph homophily, which refers to the degree to which connected nodes share labels or features, has not been systematically examined in the context of extraction attacks where attackers lack access to the graph structure. This work aims to fill that gap by addressing the following research questions:

RQ1: How does graph homophily influence the effectiveness of structure-free model extraction attacks? Given that homophily affects how label information propagates in a graph, we investigate how different forms of homophily, such as global homophily and train-test homophily, affect the fidelity of surrogate models trained without access to the graph structure.

RQ2: How does the choice of a homophily-sensitive (GCN) versus a heterophily-resilient (FAGCN) architecture affect extraction susceptibility under structure-free conditions? We analyze whether the robustness of target models to heterophilic graphs translates to greater resilience or vulnerability to extraction attacks, and how this relationship changes across the homophily spectrum.

RQ3: Can GSL methods enhance extraction fidelity in structure-free settings, and how does their effectiveness depend on the underlying graph homophily? By leveraging GSL techniques, we assess whether inferred or synthesized topologies can meaningfully improve attack performance in low-structure, low-homophily scenarios.

4 Empirical Investigation

In this section, we perform extensive experiments across ten datasets and six baseline methods to answer the three research questions mentioned in Sec 3.

Table 1: Fidelity comparison against GCN and FAGCN models. On highly homophilic datasets, most methods achieve high fidelity when targeting GCN, whereas on datasets with low homophily, the overall attack performance is relatively lower. When targeting FAGCN, the overall fidelity is higher with smaller performance variance, particularly on low-homophily datasets where multiple methods exhibit significant improvements compared to the GCN results.

Target Model	Dataset	GCN-true	GCN-kNN	MLP	LDS	SLAPS	HESGSL
GCN	actor	30.41 ±1.11	41.38 ±1.35	**46.34 ±1.39**	OOM	43.93 ±1.75	44.92 ±0.74
	chameleon	26.54 ±2.82	**62.72 ±1.61**	50.35 ±0.98	39.01 ±2.22	51.67 ±1.37	50.92 ±1.50
	citeseer	55.84 ±0.88	76.98 ±0.08	67.94 ±0.32	76.07 ±0.71	**77.16 ±0.64**	75.32 ±2.35
	cora	62.06 ±0.65	69.48 ±0.25	62.78 ±0.44	71.05 ±0.35	**72.82 ±0.80**	69.76 ±1.65
	cornell	48.65 ±6.89	63.24 ±5.27	**68.65 ±2.42**	61.86 ±9.09	65.95 ±3.08	64.32 ±8.42
	minesweeper	69.81 ±0.16	**69.81 ±0.16**	69.81 ±0.16	OOM	69.78 ±0.18	69.79 ±0.17
	pubmed	78.66 ±0.23	79.46 ±0.36	79.82 ±0.71	OOM	80.10 ±0.94	**81.60 ±1.31**
	squirrel	14.02 ±0.63	33.76 ±2.06	**39.31 ±1.10**	35.62 ±1.39	38.46 ±1.55	38.69 ±1.67
	texas	45.41 ±15.22	55.68 ±8.88	**61.08 ±6.78**	54.93 ±10.32	60.00 ±8.84	60.54 ±4.10
	wisconsin	61.57 ±2.97	62.55 ±4.75	60.00 ±8.23	62.45 ±3.64	**67.06 ±6.85**	62.75 ±5.06
FAGCN	actor	48.34 ±0.92	54.59 ±1.19	67.78 ±1.20	OOM	**68.86 ±1.35**	67.70 ±0.69
	chameleon	28.25 ±2.02	45.61 ±0.92	53.60 ±1.88	46.40 ±1.63	**54.47 ±1.77**	53.99 ±2.25
	citeseer	57.40 ±2.16	**79.36 ±0.23**	71.70 ±0.19	76.73 ±1.21	75.50 ±0.80	76.22 ±1.29
	cora	64.02 ±0.53	70.68 ±0.28	65.72 ±0.61	71.45 ±0.84	72.50 ±1.88	**75.00 ±0.73**
	cornell	70.27 ±13.78	75.14 ±4.01	81.62 ±4.83	73.99 ±6.56	**82.16 ±6.51**	77.84 ±8.42
	minesweeper	61.87 ±0.45	61.88 ±0.47	61.87 ±0.45	OOM	**61.88 ±0.46**	61.84 ±0.43
	pubmed	84.10 ±0.69	84.70 ±0.29	**86.38 ±0.29**	OOM	85.12 ±1.70	84.64 ±0.25
	squirrel	18.00 ±1.62	37.64 ±1.69	**54.27 ±1.18**	37.67 ±1.65	47.72 ±2.19	49.84 ±0.57
	texas	54.05 ±22.61	81.08 ±5.41	84.32 ±5.86	80.07 ±4.07	82.16 ±6.22	**85.95 ±5.20**
	wisconsin	78.43 ±8.43	83.92 ±2.91	85.10 ±3.28	82.13 ±4.78	84.31 ±2.40	**85.88 ±0.88**

Datasets. To evaluate model extraction attacks across different homophily conditions, we select 10 benchmark datasets ranging from strongly homophilic to strongly heterophilic graphs. This includes three homophilic citation networks: Cora, Citeseer, and Pubmed, where connected nodes often share class labels. For heterophilic scenarios, we use social networks: Actor, Chameleon, and Squirrel, where connected nodes typically differ in labels, and webpage networks: Cornell, Texas, and Wisconsin, which exhibit moderate heterophily. We also include Minesweeper as a non-social graph with unique structural patterns. App. A.2 provides dataset descriptions and summary statistics.

Protocol of Experiments. We evaluated how graph homophily impacts structure-free model extraction attacks on GNNs. We measured surrogate model fidelity using various GSL methods in a transductive node classification task under Topology Inference. We also analyzed total and train-test homophily to assess their impact on extraction effectiveness across GNN architectures, targeting GCN (sensitive to homophily) and FAGCN (resilient to heterophily).

Baselines. We employ five GSL and structure-free methods: (1) *GCN-true*: Uses the original graph structure as input for the GCN; (2) *GCN-kNN* [5]: Constructs a k-nearest neighbor graph based on node feature similarity for GCN input; (3) *HESGSL* [15]: Enhances GSL with homophily regularization during structure generation, encouraging edges between same-class nodes to improve predictions in homophilic settings; (4) *LDS* [23]: Learns a pseudo-adjacency matrix from node features to create a surrogate graph for structure-aware message passing without true topology; (5) *MLP*: A baseline using a multi-layer perceptron to predict node labels based on features and target model outputs, without graph structure; (6) *SLAPS* [19]: Jointly optimizes structure learning and self-supervised feature prediction, inferring graph topology from node features and using auxiliary tasks to stabilize training and enhance classification. The App. A.3 shows the hyperparameter tuning process.

Homophily as a Key Factor in Structure-Free Graph Model Extraction. To address RQ1, we investigate the role of graph homophily in structure-free model extraction attacks. Our key observation is a strong, consistent relationship between train-test homophily and attack success, as measured by extraction fidelity (see Figure 2). We find that attacks are highly effective on high-homophily graphs (e.g., Cora, CiteSeer), achieving up to 77% fidelity, but perform poorly on low-homophily graphs (e.g., Actor, Chameleon). This is because high homophily enables the learned proxy graph to reliably connect test nodes to like-labeled training nodes, allowing the surrogate model to better approximate the target GNN's behavior. In low-homophily settings, this label propagation is disrupted, leading to ineffective surrogate models. For homophily measures of proxy graphs and their correlation with fidelity, see App. A.4.

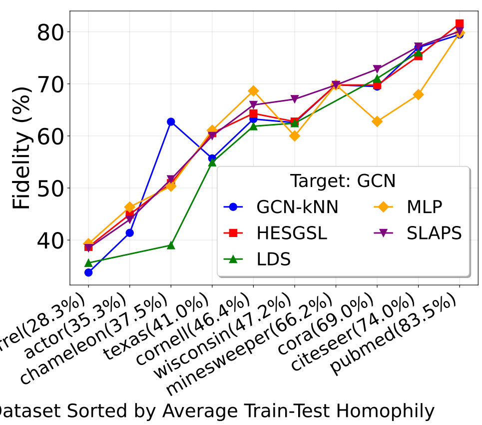

Figure 2: GSL methods' fidelity vs. dataset homophily shows a positive correlation: higher homophily boosts fidelity to 70–90% across most attacks, while low-homophily datasets show lower fidelity. Notably, attack fidelity is often higher for FAGCN than GCN, especially on low-homophily datasets, indicating that both the target model's heterophily adaptability and proxy graph's train-test homophily influence extraction difficulty.

Architectural Vulnerability to Extraction. To address RQ2, we compare the extraction susceptibility of a homophily-sensitive GCN with a heterophily-resilient FAGCN. Our key finding is a counterintuitive security trade-off: the FAGCN target model is consistently more vulnerable to extraction than the GCN target across most datasets (Table 1). This vulnerability gap is most pronounced on low-homophily, heterophilic graphs. We posit that because FAGCN is designed to be less reliant on potentially noisy graph structure and more reliant on node features in these settings, its decision boundary becomes easier for an attacker to learn from feature signals alone. This is evidenced by the MLP surrogate, which performs poorly against GCN but becomes highly competitive against FAGCN. On high-homophily graphs, the disparity between the two targets narrows, as both architectures can be approximated effectively.

Homophily-Dependent Fidelity Gains in GSL-Enhanced Extraction. To address RQ3, we evaluate whether GSL methods improve extraction fidelity and how this depends on homophily. Our findings reveal that the effectiveness of GSL is highly dependent on the underlying graph homophily. According to Table 1, in high-homophily settings (e.g., Cora, CiteSeer), GSL methods consistently and significantly outperform feature-based MLP surrogates. This is because the learned pseudo-structures successfully approximate the homophilic neighborhood aggregation of the target GNN, leading to substantial fidelity gains. Conversely, in low-homophily environments (e.g., Squirrel, Chameleon), GSL methods offer mixed or even negative returns compared to a simple MLP. The learned structures often incorrectly inject homophily where little exists, creating a mismatch with the target model's behavior. This misalignment is particularly detrimental when attacking heterophily-resilient architectures like FAGCN. The key insight is that the benefit of GSL is contingent on its ability to infer a structure that aligns with the homophily pattern the target model was trained on. When this alignment is achieved, fidelity improves; when it is not, GSL can be counterproductive. Therefore, GSL enhances extraction primarily in high-homophily scenarios.

5 Related Work

Advanced GNN Architectures for Heterophily. Traditional Graph Convolutional Networks (GCNs) inherently smooth node features by aggregating information from neighbors, which performs well on homophilic graphs where connected nodes tend to share similar labels [2, 9, 33, 34, 35, 36]. However, in heterophilic scenarios [37, 15, 25, 27], where connected nodes often have dissimilar labels, this smoothing effect can obscure important discriminative signals, leading to performance degradation. To address these limitations, FAGCN[29] decomposes node features into low-frequency (smoothed via neighborhood aggregation) and high-frequency (residual components) signals, adaptively combining them using a learnable gate. Additionally, it introduces signed edge weights, enabling both low-frequency smoothing and high-frequency enhancement through negative connections. This dual mechanism, which combines frequency-adaptive gating and flexible edge weighting, effectively handles heterophilic graphs by preventing oversmoothing and preserving discriminative node features, thereby improving node classification performance.

Structure-Free Model Extraction via Graph Structure Learning. GNNs are effective primarily due to topology-driven message passing [2]. However, in many practical scenarios like privacy-preserving or proprietary-data settings, the true graph is unavailable. This forces GNNs to rely heavily on intrinsic feature–label correlations such as homophily [1, 12, 9]. A naïve structure-free extraction approach simply treats a target GNN as a feature-to-label black box and trains an MLP surrogate on node features and outputs; however, this ignores the structural priors that GNNs exploit and thus limits surrogate fidelity. To address this limitation, LDS [23], and more broadly GSL [3, 5, 24, 30, 20, 21, 22], aim to infer a pseudo-adjacency matrix from node features so that a surrogate GNN can perform structure-aware aggregation even in the absence of the true topology [13], thereby substantially improving extraction success compared with MLP baselines. Representative GSL methods further refine this idea: SLAPS [19] jointly optimizes structure generation with self-supervised feature-prediction objectives under the hypothesis that a structure suitable for reconstructing features will also support label prediction, while auxiliary reconstruction losses stabilize end-to-end learning when labeled data are scarce. HESGSL [15] extends this framework by introducing explicit homophily regularization to encourage connections between nodes of the same class, making it particularly effective in settings where the homophily assumption holds. Collectively, these approaches illustrate a common strategy for structure-free settings: by reconstructing or regularizing a learnable pseudo-graph through self-supervision and domain-specific inductive biases such as homophily, surrogate GNNs can recover structure-aware behaviors of complex targets and achieve higher fidelity model extraction than feature-only surrogates.

6 Conclusion

We investigate factors influencing structure-free model extraction fidelity in GNNs. We identify homophily as a pivotal factor: high train-test homophily enables high-fidelity extraction, heterophily-resilient architectures like FAGCN are unexpectedly more vulnerable, and GSL methods only provide consistent gains in homophilic settings. These findings reveal that a model's ability to handle heterophily does not confer security and may even increase risk. We discuss the future work and the limitations in App. B.

References

[1] Deepak Bhaskar Acharya and Huaming Zhang. Feature selection and extraction for graph neural networks. In *Proceedings of the 2020 ACM southeast conference*, pages 252–255, 2020.

[2] Vijay Prakash Dwivedi, Chaitanya K Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *Journal of Machine Learning Research*, 24(43):1–48, 2023.

[3] Zhiyao Zhou, Sheng Zhou, Bochao Mao, Xuanyi Zhou, Jiawei Chen, Qiaoyu Tan, Daochen Zha, Yan Feng, Chun Chen, and Can Wang. Opengsl: A comprehensive benchmark for graph structure learning. *Advances in Neural Information Processing Systems*, 36:17904–17928, 2023.

[4] Yushun Dong, Song Wang, Zhenyu Lei, Zaiyi Zheng, Jing Ma, Chen Chen, and Jundong Li. A benchmark for fairness-aware graph learning. *arXiv preprint arXiv:2407.12112*, 2024.

[5] Zhixun Li, Liang Wang, Xin Sun, Yifan Luo, Yanqiao Zhu, Dingshuo Chen, Yingtao Luo, Xiangxin Zhou, Qiang Liu, Shu Wu, et al. Gslb: the graph structure learning benchmark. *Advances in Neural Information Processing Systems*, 36:30306–30318, 2023.

[6] Kiavash Satvat, Rigel Gjomemo, and VN Venkatakrishnan. Extractor: Extracting attack behavior from threat reports. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 598–615. IEEE, 2021.

[7] Zaixi Zhang, Qi Liu, Zhenya Huang, Hao Wang, Chengqiang Lu, Chuanren Liu, and Enhong Chen. Graphmi: Extracting private graph data from graph neural networks. *arXiv preprint arXiv:2106.02820*, 2021.

[8] David DeFazio and Arti Ramesh. Adversarial model extraction on graph neural networks. *arXiv preprint arXiv:1912.07721*, 2019.

[9] Lichao Sun, Yingtong Dou, Carl Yang, Kai Zhang, Ji Wang, Philip S Yu, Lifang He, and Bo Li. Adversarial attack and defense on graph data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):7693–7711, 2022.

[10] Kaixiang Zhao, Lincan Li, Kaize Ding, Neil Zhenqiang Gong, Yue Zhao, and Yushun Dong. A systematic survey of model extraction attacks and defenses: State-of-the-art and perspectives. *arXiv preprint arXiv:2508.15031*, 2025.

[11] Yuanxin Zhuang, Chuan Shi, Mengmei Zhang, Jinghui Chen, Lingjuan Lyu, Pan Zhou, and Lichao Sun. Unveiling the secrets without data: Can graph neural networks be exploited through {Data-Free} model extraction attacks? In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 5251–5268, 2024.

[12] Bang Wu, Xiangwen Yang, Shirui Pan, and Xingliang Yuan. Adapting membership inference attacks to gnn for graph classification: Approaches and implications. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 1421–1426. IEEE, 2021.

[13] Bang Wu, Xiangwen Yang, Shirui Pan, and Xingliang Yuan. Model extraction attacks on graph neural networks: Taxonomy and realisation. In *Proceedings of the 2022 ACM on Asia conference on computer and communications security*, pages 337–350, 2022.

[14] Qitian Wu, Wentao Zhao, Chenxiao Yang, Hengrui Zhang, Fan Nie, Haitian Jiang, Yatao Bian, and Junchi Yan. Sgformer: Simplifying and empowering transformers for large-graph representations. *Advances in Neural Information Processing Systems*, 36:64753–64773, 2023.

[15] Lirong Wu, Haitao Lin, Zihan Liu, Zicheng Liu, Yufei Huang, and Stan Z Li. Homophily-enhanced self-supervision for graph structure learning: Insights and directions. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[16] Yu Chen, Lingfei Wu, and Mohammed Zaki. Iterative deep graph learning for graph neural networks: Better and robust node embeddings. *Advances in neural information processing systems*, 33:19314–19326, 2020.

[17] Donghan Yu, Ruohong Zhang, Zhengbao Jiang, Yuexin Wu, and Yiming Yang. Graph-revised convolutional network. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 378–393. Springer, 2020.

[18] Danning Lao, Xinyu Yang, Qitian Wu, and Junchi Yan. Variational inference for training graph neural networks in low-data regime through joint structure-label estimation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 824–834, 2022.

[19] Bahare Fatemi, Layla El Asri, and Seyed Mehran Kazemi. Slaps: Self-supervision improves structure learning for graph neural networks. *Advances in Neural Information Processing Systems*, 34:22667–22681, 2021.

[20] Chengcheng Yu, Jiapeng Zhu, and Xiang Li. Class-balanced and reinforced active learning on graphs. *arXiv preprint arXiv:2402.10074*, 2024.

[21] Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 66–74, 2020.

[22] Yixin Liu, Yu Zheng, Daokun Zhang, Hongxu Chen, Hao Peng, and Shirui Pan. Towards unsupervised deep graph structure learning. In *Proceedings of the ACM Web Conference 2022*, pages 1392–1403, 2022.

[23] Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. Learning discrete structures for graph neural networks. In *International conference on machine learning*, pages 1972–1982. PMLR, 2019.

[24] Qingyun Sun, Jianxin Li, Hao Peng, Jia Wu, Xingcheng Fu, Cheng Ji, and Philip S Yu. Graph structure learning with variational information bottleneck. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 4165–4174, 2022.

[25] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in neural information processing systems*, 33:7793–7804, 2020.

[26] Jiajun Zhou, Shengbo Gong, Xuanze Chen, Chenxuan Xie, Shanqing Yu, Qi Xuan, and Xiaoniu Yang. Clarify confused nodes via separated learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[27] Yao Ma, Xiaorui Liu, Neil Shah, and Jiliang Tang. Is homophily a necessity for graph neural networks? *arXiv preprint arXiv:2106.06134*, 2021.

[28] Hongliang Chi, Qiong Wu, Zhengyi Zhou, and Yao Ma. Shapley-guided utility learning for effective graph inference data valuation. *arXiv preprint arXiv:2503.18195*, 2025.

[29] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 3950–3957, 2021.

[30] Fuli Feng, Weiran Huang, Xiangnan He, Xin Xin, Qifan Wang, and Tat-Seng Chua. Should graph convolution trust neighbors? a simple causal inference method. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1208–1218, 2021.

[31] Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. High accuracy and high fidelity extraction of neural networks. In *29th USENIX security symposium (USENIX Security 20)*, pages 1345–1362, 2020.

[32] Xu Zheng, Farhad Shirani, Tianchun Wang, Wei Cheng, Zhuomin Chen, Haifeng Chen, Hua Wei, and Dongsheng Luo. Towards robust fidelity for evaluating explainability of graph neural networks. *arXiv preprint arXiv:2310.01820*, 2023.

[33] Xin Zheng, Yi Wang, Yixin Liu, Ming Li, Miao Zhang, Di Jin, Philip S Yu, and Shirui Pan. Graph neural networks for graphs with heterophily: A survey. *arXiv preprint arXiv:2202.07082*, 2022.

[34] Emanuele Rossi, Bertrand Charpentier, Francesco Di Giovanni, Fabrizio Frasca, Stephan Günnemann, and Michael M Bronstein. Edge directionality improves learning on heterophilic graphs. In *Learning on graphs conference*, pages 25–1. PMLR, 2024.

[35] Yilun Zheng, Sitao Luan, and Lihui Chen. What is missing for graph homophily? disentangling graph homophily for graph neural networks. *Advances in Neural Information Processing Systems*, 37:68406–68452, 2024.

[36] Sitao Luan, Chenqing Hua, Qincheng Lu, Liheng Ma, Lirong Wu, Xinyu Wang, Minkai Xu, Xiao-Wen Chang, Doina Precup, Rex Ying, et al. The heterophilic graph learning handbook: Benchmarks, models, theoretical analysis, applications and challenges. *arXiv preprint arXiv:2407.09618*, 2024.

[37] 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? *arXiv preprint arXiv:2302.11640*, 2023.

[38] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[39] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. *arXiv preprint arXiv:2002.05287*, 2020.

[40] Microsoft. Neural Network Intelligence, 1 2021.

A Appendix

A.1 Experiment Setting

All experiments were implemented using PyTorch and PyTorch Geometric for GNN and GSL method implementations. Experiments were conducted on a machine equipped with an Intel Core i7-13620H CPU and an NVIDIA GeForce RTX 4070 GPU to support the computational requirements of graph processing and structure learning.

A.2 Dataset Description

We give the number of nodes, edges, homophily ratios and distinct classes of datasets we used in this paper in Table 2.

Table 2: Benchmark dataset summary statistics from PyTorch Geometric [38]. The homophily ratio h measures the proportion of edges connecting nodes of the same class, where values closer to 1 indicate strong homophily and values closer to 0 indicate heterophily.

	Actor	**Chameleon**	**Citeseer**	**Cornell**	**Cora**	**Minesweeper**	**Pubmed**	**Squirrel**	**Texas**	**Wisconsin**
# Nodes (|V|)	7,600	2,277	3,327	183	2,708	10,000	19,717	5,201	183	251
# Edges (|E|)	26,752	31,421	4,676	280	5,278	39,402	44,327	198,493	295	466
Homophily Ratio (h)	0.22	0.23	0.74	0.30	0.81	0.71	0.80	0.22	0.11	0.21
# Classes (|C|)	5	5	6	5	7	2	3	5	5	5

For all datasets, we follow the experimental setting provided in Pei et al. [39], which consists of 10 random splits with proportions 48/32/20% corresponding to training/validation/test for each graph.

A.3 Hyperparameter Tuning

We conduct extensive hyperparameter tuning for all models using Microsoft's Neural Network Intelligence (NNI) framework [40], with each model having a tailored search space designed to explore its specific architectural characteristics and optimization requirements: for the HESGSL model, we tune eight parameters including learning rate (0.0001 to 0.01, uniform), weight decay (0.00005 to 0.005, uniform), classifier hidden dimension (32 or 128), DAE hidden dimension (512 or 1024), neighborhood size (2 or 3), number of hops (2 or 3), alpha (0.1 to 1.0, uniform), and beta (1.0 to 10.0, uniform); the SLAPS model optimization focuses on five parameters including learning rate (0.0001 to 0.01, uniform), weight decay (0.00005 to 0.005), adjacency learning rate (0.0001 to 0.01, uniform), and two adjacency dropout rates (both ranging from 0.00005 to 0.5, uniform); for the GCN and MLP baselines, we optimize three core parameters including learning rate (0.0001 to 0.01, uniform), weight decay (0.00005 to 0.005, uniform), and hidden dimension (256, 512, or 1024); and the LDS model tuning includes learning rate (0.0001 to 0.01, uniform), weight decay (0.00005 to 0.005, uniform), and hidden size (64 or 128). The tuning process employs Bayesian optimization with Tree-structured Parzen Estimator (TPE) for efficient exploration of each parameter space. We allocate 50 trials per model for comprehensive search, with early stopping applied when performance plateaus. Optimal configurations are selected based on validation set performance using our primary evaluation metric, ensuring fair comparison across all models.

A.4 The Homophily Measures of Proxy Graphs

Table 3: The homophily measures of proxy graphs against an GCN model directly correlate with fidelity results in Table 1. On high-homophily datasets, higher train-test homophily corresponds to significantly higher extraction fidelity, whereas on low-homophily datasets, both homophily metrics remain relatively low, resulting in generally poor attack performance.

Dataset	**Global Homophily (%)**				**Train-Test Homophily (%)**			
	GCN-kNN	LDS	SLAPS	HESGSL	GCN-kNN	LDS	SLAPS	HESGSL
actor	36.24	OOM	40.94 ±2.09	**52.70 ±2.72**	29.10	OOM	37.36 ±1.94	**39.57 ±2.60**
chameleon	34.18	38.16 ±1.96	**54.92 ±4.59**	62.04 ±2.74	28.34	27.35 ±1.89	**48.69 ±2.13**	45.74 ±1.77
citeseer	67.57	**71.46 ±1.03**	60.96 ±0.36	67.59 ±3.98	61.79	66.20 ±3.68	**88.35 ±2.40**	79.71 ±7.34
cora	62.38	**63.57 ±2.00**	59.25 ±0.34	66.27 ±3.28	58.73	56.83 ±2.87	**84.89 ±1.38**	75.67 ±7.97
cornell	55.84	54.80 ±1.01	25.40 ±0.29	**69.65 ±7.29**	51.57	50.26 ±4.06	25.29 ±2.09	**58.65 ±11.65**
minesweeper	**70.78**	OOM	61.51 ±0.04	68.79 ±0.00	**70.44**	OOM	60.92 ±0.66	67.21 ±2.25
pubmed	77.97	OOM	76.76 ±2.17	**79.79 ±0.84**	82.76	OOM	**88.14 ±4.43**	79.59 ±3.11
squirrel	32.05	32.36 ±0.41	46.09 ±10.77	**46.13 ±7.21**	23.24	24.91 ±0.78	**33.65 ±3.73**	31.47 ±3.29
texas	52.36	51.97 ±0.50	28.48 ±0.14	**60.94 ±8.13**	40.11	47.03 ±4.42	28.66 ±2.17	**48.03 ±9.84**
wisconsin	54.13	55.22 ±2.37	27.80 ±0.08	**76.91 ±2.06**	50.00	44.80 ±2.67	27.75 ±0.96	**66.20 ±6.29**

Table 4: Homophily measures of proxy graphs against an FAGCN model correlate with fidelity results in Table 1. Compared to GCN, FAGCN yields higher global and train-test homophily, showing a stronger correlation with extraction fidelity. On high-homophily datasets, improved train-test homophily significantly boosts attack success rates.

Dataset	**Global Homophily (%)**				**Train-Test Homophily (%)**			
	GCN-kNN	LDS	SLAPS	HESGSL	GCN-kNN	LDS	SLAPS	HESGSL
actor	48.13	OOM	65.04 ± 0.72	**71.27 ± 1.04**	41.04	OOM	63.52 ± 0.68	**66.43 ± 0.77**
chameleon	37.01	38.04 ± 2.9	**59.84 ± 2.52**	**60.7 ± 1.77**	30.46	29.42 ± 1.44	**53.51 ± 1.86**	45.86 ± 2.68
citeseer	66.89	**68.14 ± 2.69**	60.75 ± 0.9	67.45 ± 4.18	61.79	63.51 ± 4.54	**86.97 ± 4.58**	74.63 ± 11.18
cora	62.54	**63.26 ± 1.93**	59.47 ± 0.66	61.03 ± 2.58	61.64	58.88 ± 4.58	**85.4 ± 1.48**	**89.12 ± 5.38**
cornell	67.57	66.74 ± 1.29	26.98 ± 0.14	**69.76 ± 8.76**	66.67	63.67 ± 4.71	27.31 ± 1.97	**68.1 ± 8.97**
minesweeper	**63.49**	OOM	55.21 ± 0.04	55.99 ± 0.0	**61.64**	OOM	55.2 ± 0.5	56.29 ± 0.35
pubmed	81.79	OOM	82.62 ± 1.89	**83.56 ± 1.02**	82.76	OOM	**93.71 ± 1.37**	87.18 ± 8.26
squirrel	32.89	34.38 ± 1.61	**60.66 ± 1.25**	**54.47 ± 9.86**	23.96	25.95 ± 0.7	**46.53 ± 2.03**	42.9 ± 4.7
texas	73.24	72.33 ± 0.81	42.13 ± 0.15	**86.44 ± 1.26**	70.06	68.54 ± 2.05	43.85 ± 2.5	**78.58 ± 8.44**
wisconsin	78.62	79.76 ± 0.96	33.99 ± 0.1	**89.44 ± 2.67**	73.21	74.56 ± 3.74	33.52 ± 2.45	**79.67 ± 6.42**

B Future Works and Limitations

Future work could explore additional factors affecting extraction under diverse graph structures and develop principled strategies for improving robustness across varying homophily levels.

While our study provides new insights into the role of homophily in structure-free model extraction attacks, several limitations remain. First, our study primarily considers surrogate training under classification tasks. Other learning paradigms, such as link prediction or graph regression, may exhibit different vulnerabilities, and their extraction dynamics remain largely unexplored. Second, our evaluation is restricted to a fixed set of datasets and homophily regimes, which may not fully capture the diversity of real-world graphs, particularly those with dynamic or evolving structures. Extending our analysis to temporal graphs or multi-relational settings would provide a more comprehensive understanding. Future work should examine structure-free extraction in dynamic and multi-relational graphs, and extend the analysis beyond node classification to tasks like link prediction and regression.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In the 5-8 sentences of abstract and last two paragrah of introduction section.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of the work in Appendix B.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: Our work is entirely empirical and does not include theoretical results. Therefore, there are no assumptions or proofs to report.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: In the experimental settings and implementation sections, details are fully described, including the data sources, model origins, the optimization ranges for hyperparameters, and other specific methodological procedures.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We can provide data and code.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: It is provided in the experimental settings and implementation section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: In the Table 1-4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: It is provided in the experimental settings and implementation section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: No ethical violations are identified in the study.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: No negative societal impacts

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: This paper does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.