MEMBERSHIP INFERENCE ATTACKS IN NEURAL NETWORKS ON SIMPLICIAL COMPLEXES

Anonymous authors
Paper under double-blind review

ABSTRACT

Simplicial Complex Neural Networks (SCNs) learn from relational data with group-level structure, including triangles, cliques, loops, and higher-order interactions. Boundary operators and Hodge Laplacians support this representation learning and encode membership-relevant cues when constructed from training simplices. These construction-dependent cues create a distinctive privacy risk alongside model memorization. We present the first security study of membership inference attacks on SCNs. Our exposure-aware analysis connects higher-order utility to membership signals carried by operator construction and model responses. We introduce a Structure-Aware Black-Box attack (SA-BB) that compares a queried simplex's prediction probability with those of neighbouring candidates in a public topology. A membership classifier trained on shadow models uses these local response patterns to infer target membership through a prediction API. To mitigate this risk, we propose Dynamic Topological Pruning and Injection (DTPI), which perturbs selected high-impact topological relations during training while preserving useful higher-order structure. Our evaluation examines membership leakage and the privacy–utility trade-off across datasets, backbones, and defenses.

1 INTRODUCTION

Geometric Deep Learning Bronstein et al. (2021) has become the standard for processing non-Euclidean data. Graph Neural Networks (GNNs) have achieved great success in social network analysis Yin et al. (2023); Awasthi et al. (2023), molecular generation Gao et al. (2025); Zhang et al. (2023), and recommendation systems Ozbay et al. (2024); Amara et al. (2025). Real-world systems also contain higher-order interactions: scientific collaborations form cliques and object trajectories contain loops Money et al. (2023); Kerimov et al. (2024). Simplicial Complex Neural Networks (SCNs) Roddenberry & Segarra (2019); Ebli et al. (2020); Bunch et al. (2020); Yang et al. (2022a) capture these structures through Hodge-Laplacian convolutions over edges, triangles, and tetrahedra.

When higher-order interactions are relevant to the task, SCNs improve utility by exploiting signals beyond the graph 1-skeleton, such as triangles, cliques, and higher-dimensional cofaces. This advantage is realized through topological operators, including boundary operators and Hodge Laplacians, which transport, aggregate, and filter signals across faces, cofaces, and same-order simplices. Operators therefore play a dual role: they enable higher-order message passing and encode membership cues through their construction from the training complex. A training k-simplex materialized in this complex contributes to operator entries, index sets, and induced adjacencies. The operator construction thus supports both utility and membership exposure, as illustrated in Figure 1(a).

This dual role leads to two forms of leakage. First, operators directly reveal membership structure (leakage **1**): boundary matrices and Hodge Laplacians record incidence and adjacency relations induced by training simplices. Releasing these operators with query-aligned indices allows an adversary to infer whether a candidate simplex belongs to the training complex. We study this construction dependence through the *operator exposure ratio*, the fraction of training-aligned structure materialized in the model's operators. Second, operators carry membership signals into model outputs through propagation (leakage **2**): SCN message passing mixes training-specific signals across adjacent faces, cofaces, and same-order simplices. The resulting prediction probabilities provide an observable channel for membership inference through a prediction API. Figure 1(b–c) illustrates the coupling between operator construction, model utility, and membership leakage. This coupling raises three

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.

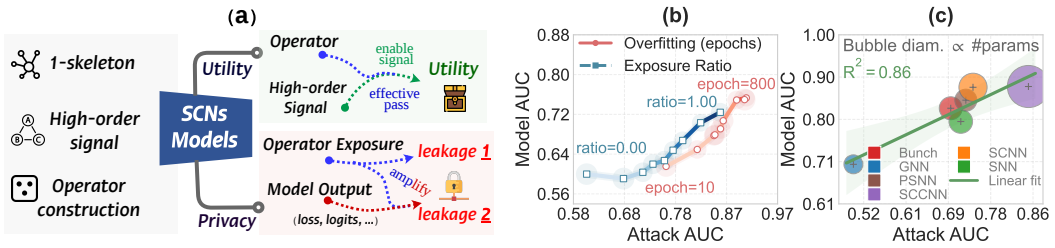


Figure 1: Panel (a) illustrates the dual role of SCN operators in utility and privacy, highlighting direct operator exposure (leakage 1) and model-output leakage (leakage 2). Panels (b)–(c) preview the resulting utility–privacy coupling across exposure, fitting, and architecture settings.

central questions, spanning risk characterization, attack design, and defense: ① *How do higher-order operators expose and amplify membership privacy risk in SCNs?* ② *How can an adversary exploit the resulting prediction patterns through black-box access?* ③ *How can topological edits reduce this risk while preserving the utility of higher-order modeling?*

To answer these questions, we conduct the *first security study of MIA on SCNs*. We focus on membership inference over candidate simplices and compare SCNs against graph-restricted baselines under aligned datasets and training protocols. We make three contributions: **(I) Exposure-Aware View and Risk Quantification.** We introduce an exposure-aware decomposition that relates higher-order utility to membership information in operator structure and model responses. This view connects direct exposure under operator release with leakage through prediction APIs.

(II) Structure-Aware Black-Box Attack. We design **SA-BB**, which compares a queried simplex's prediction probability with those of neighbouring candidates in a public topology. Shared boundary faces define the query neighbourhood, and local probability differences and ranks characterize the query's response pattern. A classifier trained on independently generated shadow models transfers these membership patterns to the target prediction API.

(III) Topological Defense. To reduce privacy risk while preserving higher-order structure, we propose **D**ynamic **T**opological **P**runing and **I**njection (**DTPI**). DTPI selectively removes and adds edges so as to perturb high-impact higher-order relations, prioritizing those that affect more surrounding simplices. These edits are further moderated based on node degree to avoid disproportionately disrupting the model's propagation behavior. The resulting mechanism suppresses membership-revealing cues while preserving useful higher-order structure, leading to improved privacy–utility trade-offs.

2 Related Work

Neural Networks on Simplicial Complexes. Simplicial Complex Neural Networks (SCNs) extend graph learning to simplicial complexes, enabling modeling of higher-order interactions beyond pairwise edges Grady & Polimeni (2010). Early studies mainly generalized spectral graph convolutions from graphs to simplices: edge-Laplacian filters were used to model flow data Roddenberry & Segarra (2019), and Simplicial Neural Networks (SNNs) leveraged the full Hodge Laplacian for spectral-domain convolutions Ebli et al. (2020). Several spectral approaches process simplices of different orders independently, combine them via attention, or organize them in blockwise architectures Yang et al. (2022a); Goh et al. (2022); Battiloro et al. (2024); Chen et al. (2022). Later work shifted toward architectures that explicitly model *inter-simplicial couplings*, enabling information exchange between simplices and their adjacent faces (lower adjacency) and cofaces (upper adjacency) through boundary operators Bunch et al. (2020); Yang et al. (2022b). Message Passing Simplicial Networks (MPSNs) further extended this idea to cell complexes Hajij et al. (2020; 2021), while recent alternatives explore simplified simplicial-aware models, message-passing-free designs, continuous formulations, and random-walk-based facilitation of higher-order structure Gurugubelli & Chepuri (2024); Ramamurthy et al. (2023); Einizade et al. (2025); Zhou et al. (2023). More recent unified models improve expressivity by separately modeling lower/upper adjacencies in gradient and curl subspaces Yang & Isufi (2023); Wu et al. (2024); Yang et al. (2025). Related work also studies community structure directly on simplicial complexes Ermolaev (2025). Sheaf-theoretic topological perspectives have likewise been used to analyze feature diffusion and attention in graph-based models Hu (2026).

Membership Inference Attacks on Graph Data. Extending MIAs to non-Euclidean graph data has revealed privacy risks rooted in topological dependency. Early work quantified leakage in graph embeddings and showed that GNNs are vulnerable to simple confidence-based attacks Duddu et al. (2020); Hu et al. (2023). Beyond overfitting, structural propagation has been identified as a major driver of membership exposure, and exploiting multi-hop neighborhood aggregation can further strengthen inference Olatunji et al. (2021); He et al. (2021). Recent *structure-level* attacks move beyond node inference to infer subgraphs, cliques, and k-hop paths in the training set Zhang et al. (2021); Wang & Wang (2024). Closest to our setting, a study evaluated privacy leakage of simplicial neural network representations via graph reconstruction attacks and found that higher-order embeddings can exacerbate leakage Zhan et al. (2023). Relatedly, privacy-preserving representation learning with simplicial complexes has also been explored from an application-driven perspective Zhan & Sheng (2023). Our work studies membership of candidate simplices and connects this risk to the operators used by SCNs. SA-BB uses public face-sharing neighbourhoods to organize prediction-API queries and learns membership patterns from their local response statistics.

3 PRELIMINARIES

Simplicial Complexes. A simplicial complex S over a finite vertex set V is a collection of simplices closed under taking faces. A k-simplex $s^k \subseteq V$ contains $k+1$ vertices; thus 0-, 1-, 2-, and 3-simplices correspond to nodes, edges, triangles, and tetrahedra, respectively. We denote by S_k the set of k-simplices, with $N_k = |S_k|$.

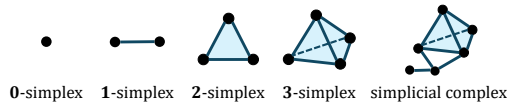


Figure 2: Simplices and simplicial complex.

Hodge Laplacian. For an oriented simplicial complex, let $B_k \in \mathbb{R}^{N_{k-1} \times N_k}$ be the boundary incidence matrix between $(k-1)$- and k-simplices. The k-th Hodge Laplacian is defined as $L_k = B_k^\top B_k + B_{k+1} B_{k+1}^\top$, where $L_{k,\ell} = B_k^\top B_k$ and $L_{k,u} = B_{k+1} B_{k+1}^\top$ denote the lower and upper Laplacians, respectively Lim (2015).

Neural Networks on Simplicial Complex. SCNs generalize Graph Neural Networks to higher-order topologies. Recent advancements unify various SCN architectures under a general message-passing paradigm defined by the following update rule for a k-simplicial signal x_k^l at layer l:

$$x_k^l = \rho\left(H_{k,\ell}^l B_k^\top x_{k-1}^{l-1} + H_k^l x_k^{l-1} + H_{k,u}^l B_{k+1} x_{k+1}^{l-1}\right), \tag{1}$$

Here the operators $H_{(\cdot)}^l$ are Simplicial Convolutional Filters (SCFs) defined as polynomials of the Hodge Laplacians. For consistency, we use the lower/upper notation (ℓ, u) throughout. The update decomposes topological information flow into three components: (i) **Information from Faces**, where $B_k^\top$ projects signals from $(k-1)$-simplices to be processed by the lower filter $H_{k,\ell}^l$, a polynomial of the lower Laplacian $L_{k,\ell}$; (ii) **Intra-simplicial Information**, where H_k^l performs spectral filtering via polynomials of both $L_{k,\ell}$ and $L_{k,u}$ to capture locality within the k-th order; and (iii) **Information from Cofaces**, where B_{k+1} projects signals from $(k+1)$-simplices via the upper filter $H_{k,u}^l$, a polynomial of the upper Laplacian $L_{k,u}$. By incorporating these inter-simplicial couplings, Eq. (1) also subsumes or closely approximates several existing architectures. SNN Ebli et al. (2020), PSNN Glaze et al. (2021), and SCNN Yang et al. (2022a) arise as restricted cases that omit explicit inter-simplicial couplings and retain only the intra-order filtering term. SCCNN Yang & Isufi (2023); Yang et al. (2025) fits the same cross-order message-passing template after absorbing degree normalizations into the boundary transport operators.

Membership Inference Attacks. Membership inference attacks determine whether a query sample z belongs to the private training set $\mathcal{D}_{\text{train}}$ of a target model f_θ. Given a query sample z and access to the target model f_θ, the goal of the adversary is to construct an inference function $\mathcal{A}(z, f_\theta) \in \{0, 1\}$, where $\mathcal{A}(z, f_\theta) = 1$ denotes member and $\mathcal{A}(z, f_\theta) = 0$ denotes non-member. For simplex prediction, the query is a candidate simplex σ. In a probability-based black-box setting, the target exposes a prediction API that returns $p_\sigma = \Pr_\theta(Y = 1 \mid \sigma)$ for each admissible query. The attacker combines these responses with public auxiliary data to estimate membership.

4 EXPOSURE, UTILITY, AND PRIVACY

This section introduces an *exposure-aware information decomposition* that connects higher-order utility to membership leakage. We then develop SA-BB, which uses public simplex neighbourhoods to probe this leakage through prediction probabilities.

4.1 EXPOSURE-AWARE INFORMATION DECOMPOSITION

Let $S = (V, S_1, \ldots, S_K) \sim \mathcal{D}$ be a simplicial complex of maximum order $K \geq 2$. Write $S_1 = E$, $\mathcal{H} = (S_2, \ldots, S_K)$, and $G = \pi(S) = (V, E)$ for its edges, higher-order simplices, and 1-skeleton.

Definition 4.1. Let $S = (G, \mathcal{H}) \sim \mathcal{D}$ be a simplicial complex of maximum order $K \geq 2$. A ***graph-restricted encoder*** is a possibly randomized measurable mapping f that produces $Z_G = f(G)$, with $Z_G \perp \mathcal{H} \mid G$. A ***simplicial encoder*** produces $Z_S = h(G, \mathcal{H})$ from S.

For the downstream task and privacy analysis, fix a query order $q \in \{1, \ldots, K\}$ and let σ be a candidate q-simplex drawn from the query distribution associated with S. Its task label is Y, and $M \in \{0, 1\}$ indicates whether it belongs to the private training set. Let R denote the construction-dependent operator representation associated with σ, including the relevant boundary matrices and Hodge Laplacians used by the SCN. This representation determines how the model propagates signals around the query and provides structural information when released with query-aligned indices. Let O denote prediction probabilities for the query and its neighbours.

Proposition 4.2 (Exposure-aware information decomposition)**.** *Under Definition 4.1, the excess task-relevant information beyond the graph view G satisfies*

$$\sup_{\phi} I\big(Y; \phi(G, \mathcal{H}, R)\big) - \sup_{f} I\big(Y; f(G)\big) = I(Y; \mathcal{H}, R \mid G), \tag{2}$$

where ϕ and f range over measurable encoders on their respective input views. For joint access to the operator representation R and model responses O, membership information decomposes as

$$I(M; R, O \mid G, Y) = \underbrace{I(M; R \mid G, Y)}_{\text{(*) protocol-side operator exposure}} + \underbrace{I(M; O \mid G, Y, R)}_{\text{(**) operator-conditioned model-side leakage}}. \tag{3}$$

The first term measures membership information in the operator representation. The second measures the additional information in model responses after conditioning on that representation. Conditioning on Y controls for the task label in both terms.

The proof follows the chain rule for mutual information (Appendix B.1). Under operator release, an attacker observes structural membership cues in R. Under prediction-API access, R remains internal to the target, and output probabilities reflect its effects together with learned parameters. The following attacks examine these structural and output channels.

4.2 PROTOCOL-SIDE STRUCTURAL ATTACK (PSA)

The term $I(M; R \mid G, Y)$ captures membership information exposed by query-aligned operators. For a queried k-simplex σ, a released boundary matrix B_k provides the materialization cue

$$z_{B_k}(\sigma) = \mathbf{1}\{\sigma \text{ is explicitly encoded in the released } B_k\}. \tag{4}$$

The upper Laplacian on boundary faces, $L_{k-1,u} = B_k B_k^\top$, provides the corresponding support cue

$$z_{L_{k-1,u}}(\sigma) = \mathbf{1}\{\exists\, \alpha_a \neq \alpha_b \in \partial\sigma : |(L_{k-1,u})_{\alpha_a, \alpha_b}| > 0\}. \tag{5}$$

Two distinct boundary faces determine a unique k-simplex, so this test identifies its materialization in the released upper operator. Under training-only construction, materialization supplies a membership witness. Appendix C.1 details this operator-release setting.

4.3 STRUCTURE-AWARE BLACK-BOX MEMBERSHIP INFERENCE (SA-BB)

SA-BB probes membership through local prediction patterns shaped by SCN message passing (Figure 3b). The attacker queries an API returning the positive-class probability $p_\nu = f_\theta(\nu)$ for a

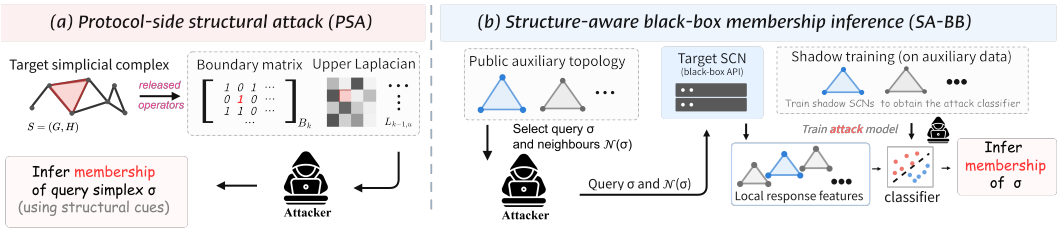


Figure 3: Attack pipelines. (a) PSA uses structural cues in released boundary matrices or upper Laplacians to infer membership. (b) SA-BB queries prediction probabilities for a simplex and its public face-sharing neighbours, then applies a shadow-trained classifier to six local response features.

candidate ν. Its auxiliary information comprises a public candidate pool $\mathcal{C}_k$ with at least two elements, their boundary faces, and labelled data for training shadow SCNs. The pool and face relations are fixed before the target membership split; all queries use a fixed model and topology.

Public neighbourhood queries. For a query $\sigma \in \mathcal{C}_k$, define its face-sharing neighbourhood as

$$\mathcal{N}(\sigma) = \{\tau \in \mathcal{C}_k \setminus \{\sigma\} : \partial\tau \cap \partial\sigma \neq \varnothing\}. \tag{6}$$

Neighbours share a complete $(k-1)$-face, such as an edge for triangle queries. The attacker selects up to $B \geq 1$ neighbours in a fixed public ordering. For an empty neighbourhood, it samples other public candidates with a fixed seed. Writing the selected set as $\mathcal{N}_B(\sigma)$ and its size as n_σ, querying σ and its neighbours costs $1 + n_\sigma \leq B + 1$ API calls.

Local response features. SA-BB constructs the six-dimensional response vector

$$\phi^{\mathrm{SA-BB}}(\sigma) = [p_\sigma,\ H(p_\sigma),\ \mu_\sigma,\ s_\sigma,\ d_\sigma^{\max},\ r_\sigma]. \tag{7}$$

Here p_σ is the query confidence and $H(p) = -p\log p - (1-p)\log(1-p)$ is binary entropy. For probability gaps $d_{\sigma,\tau} = |p_\sigma - p_\tau|$ over $\tau \in \mathcal{N}_B(\sigma)$, μ_σ and $d_\sigma^{\max}$ denote their mean and maximum. Their standard deviation s_σ and the query's relative confidence rank r_σ are

$$s_\sigma = \left[\frac{1}{n_\sigma}\sum_{\tau\in\mathcal{N}_B(\sigma)}(d_{\sigma,\tau} - \mu_\sigma)^2\right]^{1/2}, \quad r_\sigma = \frac{1 + \sum_{\tau\in\mathcal{N}_B(\sigma)} \mathbf{1}\{p_\tau \leq p_\sigma\}}{n_\sigma + 1}. \tag{8}$$

Shadow training and target inference. The attacker trains shadow SCNs on independent auxiliary splits using a known model family and training procedure. It extracts Eq. (7) for shadow members and held-out candidates, then fits feature standardization and a logistic-regression membership classifier on the shadow attack-training partition. Applying this fixed pipeline to target API responses yields membership scores; the evaluator uses target membership labels to measure attack performance.

5 DTPI Defense

To reduce privacy risk while preserving higher-order structure, we propose **D**ynamic **T**opological **P**runing and **I**njection (**DTPI**) as a defense method. During training, it applies dynamic topological edits that weaken reliance on specific training simplices. These edits also perturb the Hodge Laplacians and boundary operators, obscuring construction-dependent operator cues and targeting both model-side leakage and protocol-side exposure.

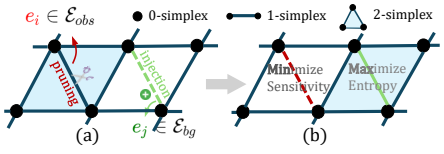


Figure 4: **Overview of DTPI.**

5.1 Motivation

The core challenge in defense is identifying which edges to edit. An effective defense must maximize disruption of membership cues while preserving the stability required for message passing.

We characterize the reach of an edge edit through its cascade size $T(e)$, which counts the affected higher-order simplices. Such edits change the operators shaping the probabilities queried by SA-BB.

Condition on the baseline graph G and learned edge-retention probabilities p. Fix a query order q, an ambient candidate family Σ_q, and an edge e. We compare coupled complexes $\widetilde{S}^{(1,e)}$ and $\widetilde{S}^{(0,e)}$ obtained by forcing e on or off with all other sampled masks held fixed. The following theorem bounds their query-level loss difference:

Theorem 5.1 (Query-Level Leverage of $T(e)$)**.** *Under the assumptions in Appendix B.3, there exist constants $c_1, c_2, c_3, c_4 > 0$, independent of e, such that for any $\delta \in (0,1)$, with conditional probability at least $1-\delta$ over the randomized edge-mask realization given (G, p), the magnitude of the query-level loss change induced by toggling an edge e satisfies*

$$\sup_{\sigma \in \Sigma_q} \left| \ell\big(f_{\widetilde{S}^{(1,e)}}(\sigma), y_\sigma\big) - \ell\big(f_{\widetilde{S}^{(0,e)}}(\sigma), y_\sigma\big) \right| \le c_1 T(e) + c_2 \sqrt{T(e)\log(c_3/\delta)} + c_4 \log(c_3/\delta). \quad (9)$$

Proof note. Appendix B.3 provides the detailed derivation of Theorem 5.1.

The bound motivates screening high-cascade edges for their potential effect on query responses. For binary simplex prediction, the task loss is computed from the prediction probability and task label, linking the bound to the output channel used by SA-BB. To preserve utility, DTPI also accounts for degree-dependent propagation stability. SCNs rely on normalized operators, whose sensitivity to editing $e = (u, v)$ is described by the degree factor $1/\sqrt{d(u)+\epsilon} + 1/\sqrt{d(v)+\epsilon}$ (Appendix B.2.4). This prioritizes well-connected endpoints with greater structural redundancy.

5.2 The DTPI Mechanism

Candidate filtering and masking. We define the raw observed-edge set $\mathcal{E}_{\text{obs}}^{\text{raw}}$ as the edges in the training complex's 1-skeleton, and the raw background-edge set $\mathcal{E}_{\text{bg}}^{\text{raw}}$ as non-existent candidate links for decoy injection. DTPI combines cascade size and propagation stability through a two-stage lexicographic filter. For both $\mathcal{E}_{\text{obs}}^{\text{raw}}$ and $\mathcal{E}_{\text{bg}}^{\text{raw}}$, we first rank candidate edges in decreasing order of $T(e)$ to form a high-leverage pool. Within that pool, we re-rank edges by increasing structural impact $1/\sqrt{d(u)+\epsilon} + 1/\sqrt{d(v)+\epsilon}$ to select the active sets $\mathcal{E}_{\text{obs}}$ and $\mathcal{E}_{\text{bg}}$.

For each selected edge in these structurally safe sets, we assign a learnable mask logit w_e. We parameterize the corresponding continuous mask by $m_e = \text{sigmoid}(w_e)$ and sample discrete edge masks $\hat{m}_e \in \{0, 1\}$ via the Gumbel-Softmax reparameterization Jang et al. (2017) for the forward pass. To ensure a valid simplicial complex, the masks of higher-order simplices are tied to their boundary edges via $m_\sigma = \prod_{\alpha \in \partial\sigma} m_\alpha$ in the continuous objective, and analogously $\hat{m}_\sigma = \prod_{\alpha \in \partial\sigma} \hat{m}_\alpha$ in the sampled forward pass. The forward pass uses the sampled masked Laplacians, which forces the model to learn representations that remain robust under these targeted topological perturbations.

Training objective. With f_θ denoting the SCN model, q_w the mask-sampling distribution induced by the logits w, and $\mathcal{D}$ the empirical training distribution over query–label pairs (σ, y), DTPI minimizes the expected masked empirical risk together with cascade-guided regularization:

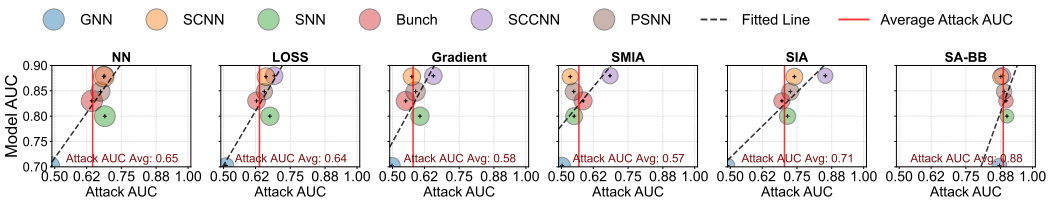
$$\min_{\theta, w} \mathbb{E}_{(\sigma,y)\sim\mathcal{D}} \mathbb{E}_{\hat{\mathcal{M}}\sim q_w} \Big[\ell\Big(f_\theta(S \odot \hat{\mathcal{M}}, \sigma), y \Big) \Big] + \lambda_1 \sum_{e \in \mathcal{E}_{\text{obs}}} m_e \bar{T}(e) + \lambda_2 \sum_{e \in \mathcal{E}_{\text{bg}}} (1 - m_e) \bar{T}(e) \quad (10)$$

The first term is the expected task loss under the sampled masked complex, whereas $\bar{T}(e)$ is the fixed, normalized cascade score. The λ_1 term penalizes retaining high-cascade observed edges by biasing their masks toward zero, while the λ_2 term encourages activating high-cascade background edges as decoys by biasing their masks toward one.

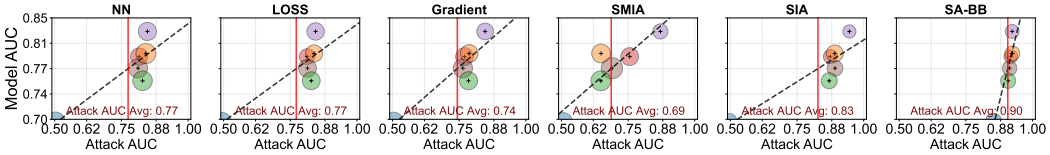
6 Experiments

6.1 Experimental Setup

Settings. We evaluate on four datasets and consider two tasks: *simplicial prediction* and *trajectory prediction*. For simplicial prediction, we use three datasets, Semantic Ammar et al. (2018), Cora, and Citeseer Yang et al. (2016). Following prior work Zhang & Chen (2018); Yang & Isufi (2023); Yang et al. (2025), the task is to predict whether a queried simplex instance is positive; protocol details are deferred to Appendix D.1. For trajectory prediction, we use the synthetic dataset of Glaze et al. (2021); we report trajectory-prediction accuracies and baseline MIA results in Appendix E.7. Our MIA infers membership of this queried simplex instance, i.e., whether it is in the private



(a) Attack results on the SEMANTIC



(b) Attack results on the CITESEER

Figure 5: Privacy–utility trade-off of membership inference attacks across backbones and attack methods. In each panel, the x-axis is the attacker AUC and the y-axis is the target model AUC.

training set. Target models use a `train/val/test` split of 8:1:1 with validation for early stopping. Unless otherwise specified, the simplicial operators B_k and L_k are constructed from the union of positive and negative *training* queries only, excluding validation/test candidates, following standard simplex-prediction practice. Appendix E.1 evaluates structural exposure under operator release.

Baselines & Threat Models. We evaluate six backbones spanning graph models (`GNN` Defferrard et al. (2017)) and simplicial complex networks (`SNN` Ebli et al. (2020), `SCNN` Yang et al. (2022a), `PSNN` Glaze et al. (2021), `Bunch` Bunch et al. (2020), and `SCCNN` Yang & Isufi (2023); Yang et al. (2025)). For privacy evaluation, we consider six membership inference attacks under both **black-box** (NN Salem et al. (2018), LOSS Sablayrolles et al. (2019), and our SA-BB) and **white-box** (GRADIENT Rezaei & Liu (2020), SIA Zhang et al. (2021), SMIA Wang & Wang (2024)) threat models; architectural details and attack-training protocols are provided in Appendices D.2 and C.2.

Defense Baselines. We compare seven defense strategies against membership inference. (i) ADV-REG Ding et al. (2024) trains an adversarial regularization framework that encourages intermediate representations of training samples to be indistinguishable from those of the validation set, thereby reducing memorization. (ii) SHAP Wang & Wang (2024) identifies influential embedding dimensions/features and injects calibrated noise to obfuscate membership cues. (iii) GEP Zhang et al. (2021) perturbs the original graph embeddings with noise to mitigate leakage from learned representations. (iv) DP-SGD Abadi et al. (2016) applies differentially private stochastic gradient descent as a principled privacy baseline. (v) KD Zheng & Peng (2022) employs knowledge distillation to transfer knowledge to a student model with reduced overfitting. (vi) EARLY-STOP uses early stopping based on validation performance to curb memorization. Finally, (vii) DTPI is our proposed defense, which combines cascade-guided candidate screening, structurally safe edge selection, and differentiable topological pruning/injection to suppress leakage while preserving utility.

Metrics. We report the area under the receiver operating characteristic curve (**AUC**) for both task utility and attacker performance. We also measure **TPR@1% FPR** and **TPR@5% FPR**, the member-detection rates at false-positive rates of 1% and 5%; Appendix E.3 reports these operating points for attacks and defenses. Unless otherwise specified, all reported means are averaged over 10 random seeds.

6.2 PRIVACY–UTILITY TRADE-OFF ACROSS BACKBONES

Figure 5 characterizes the privacy–utility trade-off of membership inference across six backbone architectures spanning graph models and simplicial complex networks with increasing expressive power. Each subfigure corresponds to one dataset, and each column corresponds to one attack method. For every column, the horizontal axis reports the attacker AUC and the vertical axis reports the target model AUC. Each circle is one backbone trained under the same protocol, where the circle center is the mean over repeated runs and the radius reflects the standard deviation. We also evaluate multiple training horizons, and Figure 5 aggregates results across epochs, while Appendix E.4 reports baseline trajectories over training. The red text in each panel reports the attack AUC averaged over all backbones. Figure 5 supports two key observations discussed below.

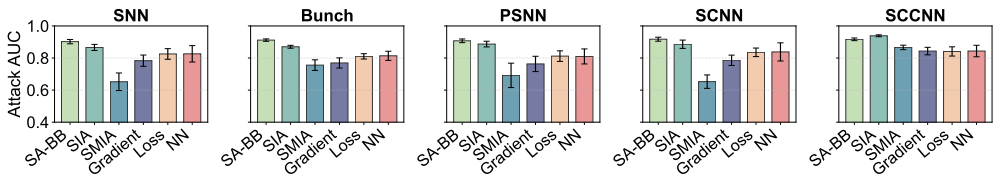


Figure 6: Attack AUC comparison across different backbones on the CITESEER benchmark

Observation I: ***Higher expressive power often coincides with higher MIA leakage.*** For each backbone, we compute mean model AUC and mean attack AUC as the black cross, then fit a least squares trend line across mean points. In the NN, Loss, Gradient, and SIA panels, higher model AUC generally accompanies higher attack AUC. SA-BB already attains high AUC across backbones, with a narrower spread in attack performance. This trend is consistent with Proposition 4.2: higher-order signal and construction-dependent operator cues can jointly improve utility while increasing membership distinguishability.

This pattern also reflects differences in how much higher-order structure different backbones can use. Traditional graph-based models tend to achieve lower model AUC and weaker MIA leakage. Models equipped with simplicial operators show intermediate utility and leakage, while backbones with richer lower/upper Hodge interactions often achieve higher model AUC and correspondingly stronger attacks. Appendix E.2 further complements this cross-backbone evidence with a controlled Full/Order-1/Graph-only ablation under the same backbone and data splits. The accompanying decline in utility and attack AUC supports the role of higher-order context in membership leakage.

Observation II: ***Local prediction patterns strengthen black-box attacks.*** SA-BB achieves average attack AUCs of approximately 0.88 on SEMANTIC and 0.90 on CITESEER in Figure 5. The corresponding NN averages are 0.65 and 0.77, and Loss reaches 0.64 and 0.77. Figure 6 provides the per-backbone comparison. These results support the design in Section 4.3: comparing a query with public face-sharing candidates reveals membership patterns in local prediction probabilities. Appendix E.5 further evaluates independently trained shadow models, low-FPR performance, and neighbour-selection controls.

Feature ablation of SA-BB. Figure 7 evaluates the six response features on CITESEER using SCCNN, $B = 20$, and three target seeds, each with three independent shadow models. We fix the API responses and shadow splits and refit logistic regression for each feature subset. The full vector achieves AUC 0.8853. Combining p_σ and r_σ gives the highest two-feature AUC, 0.8676, compared with 0.8413 for p_σ alone. Removing p_σ reduces AUC by 6.91 points on average; removing r_σ or μ_σ reduces it by 1.26 or 1.02 points. The remaining features have smaller average removal effects. These results show complementary membership signals in query confidence and neighbourhood responses.

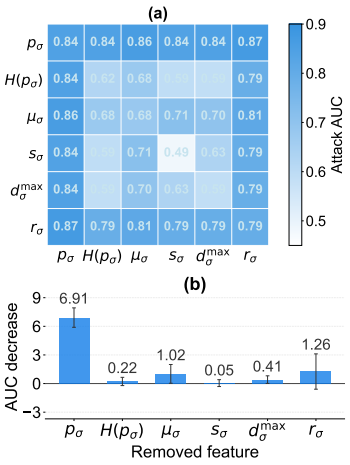


Figure 7: SA-BB feature ablation. (a) Single-feature (diagonal) and two-feature AUC. (b) AUC decrease after feature removal.

6.3 MITIGATING LEAKAGE WHILE PRESERVING UTILITY

Figure 8 summarizes defense behaviors against diverse MIA methods on a fixed SCCNN backbone. All target models are trained for 1000 epochs with early stopping and model selection based on validation AUC to mitigate overfitting. Each panel shows the trajectory of a defense as its hyperparameters vary, with attack AUC on the x-axis and model AUC on the y-axis. The preferred region is the upper-left corner, corresponding to lower attack success and higher task utility; the yellow star marks this reference point. The shaded bands indicate the standard deviation of model AUC across runs.

DTPI defense performance. Figure 8 shows that DTPI achieves a favorable privacy–utility trade-off on CITESEER. In the NN and SA-BB panels, its trajectory extends toward the upper-left region, reducing attack AUC while retaining high model AUC. Under SA-BB, this behavior indicates that topological edits weaken local response patterns used for membership inference. The shaded bands show the variation in utility across runs, which grows for several noise-based baselines as defense strength increases. Appendix E.7 reports additional baseline attack results across datasets, backbones.

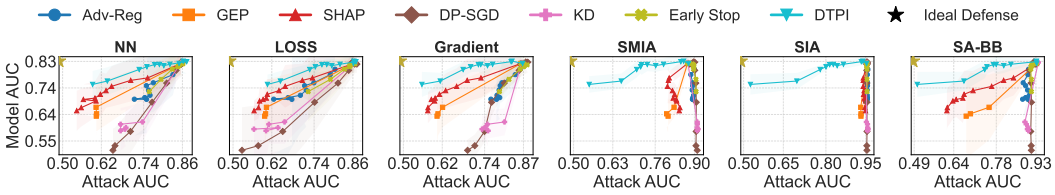


Figure 8: Defense trade-offs under different MIA methods on the CITESEER benchmark with SCCNN.

Defense dynamics and failure modes. We also observe two recurring patterns shared by multiple defenses. **(i)** For several baselines, increasing the defense strength causes a sharp drop in model AUC without further reducing attack AUC, suggesting that the training procedure collapses under over-regularization, excessive distillation, or overly strong noise injection. **(ii)** Some trajectories are non-monotonic, notably early stopping, where longer training can reduce attack AUC. Appendix E.6 examines the accompanying changes in member and non-member gradient roughness over training.

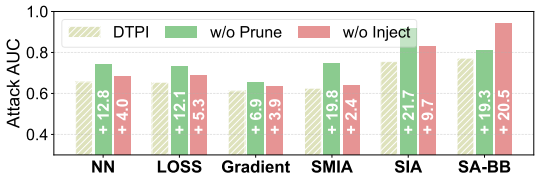


Table 1: Ablation of screening rules in DTPI.

Screening rule	2-simplex		3-simplex	
	$\text{AUC}_{\text{Model}}\uparrow$	$\text{AUC}_{\text{Attack}}\downarrow$	$\text{AUC}_{\text{Model}}\uparrow$	$\text{AUC}_{\text{Attack}}\downarrow$
DTPI	0.779	0.676	0.675	0.716
Degree-only	**0.780**	0.687	**0.689**	0.726
$T(e)$-only	0.767	**0.670**	0.665	**0.712**
Random	0.750	0.699	0.632	0.736

Figure 9: Ablation study of DTPI. The left panel evaluates pruning and injection under six baseline attacks, while the right table compares different edge-screening strategies.

Ablation of DTPI strategies. Figure 9 reports two complementary ablations. The left panel removes pruning or injection while keeping candidate screening and training fixed. Across the five baseline attacks (without SA-BB), removing pruning raises attack AUC by 14.7% on average, while removing injection raises it by 5.1%. Both components help suppress membership cues: pruning removes high-impact observed relations, and injection adds structurally safe decoys.

We further ablate the edge-screening strategy to verify the rationale behind DTPI's candidate selection. Under a fixed edit budget, DTPI is compared with three alternatives: degree-only, $T(e)$-only, and random screening. Degree-only screening best preserves utility, while $T(e)$-only screening most reduces attack AUC, confirming that the two signals capture complementary privacy–utility factors. DTPI balances them by using $T(e)$ to identify high-leverage edges and degree information to preserve propagation stability. Random screening degrades both utility and privacy.

Operator-update frequency. Table 2 shows that operator-update frequency is an important factor in DTPI. More frequent rebuilding strengthens perturbation and privacy protection, but increases training cost and reduces model utility. In contrast, longer rebuild intervals improve efficiency and utility at the cost of weaker defense, revealing a clear privacy–utility trade-off. Complexity and runtime comparisons with other baselines are provided in Appendix C.3.3.

Table 2: DTPI operator-update frequency ablation.

Rebuild interval	Train time (s/epoch)	Model AUC	SA-BB AUC	TPR@1% FPR
every epoch	0.174	0.780	0.803	0.563
every 5 epochs	0.066	0.821	0.853	0.622
every 10 epochs	0.051	0.836	0.874	0.657
every 50 epochs	0.038	0.839	0.915	0.673

7 CONCLUSION

This work connects membership leakage in SCNs to their topological operators and learned responses. Our analysis identifies protocol-side structural exposure and model-side leakage, and SA-BB uses public neighbourhoods and prediction probabilities to infer membership. DTPI reduces leakage through targeted pruning and injection while preserving useful higher-order structure. Experiments across datasets and backbones characterize the resulting privacy–utility trade-off. The results highlight the need to assess operator construction and attacker access together, and to balance task utility, residual leakage, and training cost when selecting defenses.

Future Work and Limitations. Our evaluation focuses on benchmark-scale complexes. Future work includes incremental updates for larger complexes and geometric constraints on injected relations.

REFERENCES

Martín Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS'16, pp. 308–318. ACM, October 2016. doi: 10.1145/2976749.2978318. URL http://dx.doi.org/10.1145/2976749.2978318.

Amina Amara, Mohamed Ali Hadj Taieb, and Mohamed Benaouicha. A multi-view gnn-based network representation learning framework for recommendation systems. *Neurocomputing*, 619: 129001, 2025.

Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hsu-Han Ooi, Matthew Peters, Joanna Power, Sam Skjonsberg, Lucy Lu Wang, Chris Wilhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. Construction of the literature graph in semantic scholar. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pp. 84–91, 2018. doi: 10.18653/v1/N18-3011.

Aravind K. Awasthi, Arun Kumar Garov, Minakshi Sharma, and Mrigank Sinha. Gnn model based on node classification forecasting in social network. *2023 International Conference on Artificial Intelligence and Smart Communication (AISC)*, pp. 1039–1043, 2023.

Claudio Battiloro, Lucia Testa, Lorenzo Giusti, Stefania Sardellitti, Paolo Di Lorenzo, and Sergio Barbarossa. Generalized simplicial attention neural networks. *IEEE Transactions on Signal and Information Processing over Networks*, 10:833–850, 2024.

Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *ArXiv*, abs/2104.13478, 2021.

Eric Bunch, Qian You, Glenn Fung, and Vikas Singh. Simplicial 2-complex convolutional neural networks. In *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond*, October 2020. URL https://openreview.net/forum?id=TLbnsKrt6J-. Poster.

Yuzhou Chen, Yulia R. Gel, and H. Vincent Poor. Bscnets: Block simplicial complex neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 6333–6341, 2022. doi: 10.1609/aaai.v36i6.20583.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *ArXiv*, abs/1606.09375, 2017.

Zehua Ding, Youliang Tian, Guorong Wang, and Jinbo Xiong. Regularization mixup adversarial training: A defense strategy for membership privacy with model availability assurance. *2024 2nd International Conference on Big Data and Privacy Computing (BDPC)*, pp. 206–212, 2024.

Vasisht Duddu, Antoine Boutet, and Virat Shejwalkar. Quantifying privacy leakage in graph embedding. In *MobiQuitous 2020 - 17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pp. 76–85, 2020. doi: 10.1145/3448891.3448939.

Stefania Ebli, Michaël Defferrard, and Gard Spreemann. Simplicial neural networks. *ArXiv*, abs/2010.03633, 2020.

Aref Einizade, Dorina Thanou, Fragkiskos D. Malliaros, and Jhony H. Giraldo. Cosmos: Continuous simplicial neural networks. *ArXiv*, abs/2503.12919, 2025.

Egor Ermolaev. Community detection on simplicial complexes. *Applied Network Science*, 10 (1):30, 2025. doi: 10.1007/s41109-025-00695-x. URL https://doi.org/10.1007/s41109-025-00695-x.

David A. Freedman. On tail probabilities for martingales. *The Annals of Probability*, 3(1):100–118, 1975. doi: 10.1214/aop/1176996452.

Xu Gao, Chengwei Zhang, Mengfan Yan, Xianyu Zuo, and Kecheng Yang. Fragment-enhanced graph neural networks for target-specific molecular generation and property prediction. In *Proceedings of the 2025 International Conference on Artificial Intelligence and Computational Intelligence*, pp. 153–158, 2025. doi: 10.1145/3730436.3730461.

Nicholas Glaze, T. Mitchell Roddenberry, and Santiago Segarra. Principled simplicial neural networks for trajectory prediction. *ArXiv*, abs/2102.10058, 2021.

Christopher Wei Jin Goh, Cristian Bodnar, and Pietro Liò. Simplicial attention networks. *ArXiv*, abs/2204.09455, 2022.

Leo J. Grady and Jonathan R. Polimeni. *Discrete Calculus: Applied Analysis on Graphs for Computational Science*. Springer London, 2010. ISBN 978-1-84996-289-6. doi: 10.1007/978-1-84996-290-2.

Sravanthi Gurugubelli and Sundeep Prabhakar Chepuri. Sann: Simple yet powerful simplicial-aware neural networks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=eUgS9Ig8JG.

Mustafa Hajij, Kyle Istvan, and Ghada Zamzmi. Cell complex neural networks. *ArXiv*, abs/2010.00743, 2020.

Mustafa Hajij, Ghada Zamzmi, and X. Cai. Simplicial complex representation learning. *ArXiv*, abs/2103.04046, 2021.

Xinlei He, Rui Wen, Yixin Wu, Michael Backes, Yun Shen, and Yang Zhang. Node-level membership inference attacks against graph neural networks. *ArXiv*, abs/2102.05429, 2021.

Chuan-Shen Hu. A sheaf-theoretic and topological perspective on complex network modeling and attention mechanisms in graph neural models. *ArXiv*, abs/2601.21207, 2026.

Yuke Hu, Wei Liang, Ruofan Wu, Kai Y. Xiao, Weiqiang Wang, Xiaochen Li, Jinfei Liu, and Zhan Qin. Quantifying and defending against privacy threats on federated knowledge graph embedding. In *Proceedings of the ACM Web Conference 2023*, pp. 2306–2317, 2023. doi: 10.1145/3543507.3583450.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *ArXiv*, abs/1611.01144, 2017.

Bulat Kerimov, Vincent Pons, Spyros Pritsis, Riccardo Taormina, and Franz Tscheikner-Gratl. Sensor placement and state estimation in water distribution systems using edge gaussian processes. *Engineering Proceedings*, 69(1):150, 2024. doi: 10.3390/engproc2024069150.

Lek-Heng Lim. Hodge laplacians on graphs. *ArXiv*, abs/1507.05379, 2015.

Dilip B. Madan and Eugene Seneta. Chebyshev polynomial approximations and characteristic function estimation. *Journal of the royal statistical society series b-methodological*, 49:163–169, 1987.

Rohan T. Money, Joshin P. Krishnan, Baltasar Beferull-Lozano, and Elvin Isufi. Online edge flow imputation on networks. *IEEE Signal Processing Letters*, 30:115–119, 2023.

Iyiola E. Olatunji, Wolfgang Nejdl, and Megha Khosla. Membership inference attack on graph neural networks. *2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, pp. 11–20, 2021.

Begum Ozbay, Resul Tugay, and Sule Gunduz Oguducu. A gnn model with adaptive weights for session-based recommendation systems. In *2024 9th International Conference on Machine Learning Technologies (ICMLT)*, pp. 258–264, 2024. doi: 10.1145/3674029.3674070.

Karthikeyan Natesan Ramamurthy, Aldo Guzmán-Sáenz, and Mustafa Hajij. Topo-mlp: A simplicial network without message passing. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023. doi: 10.1109/ICASSP49357.2023.10094803.

Shahbaz Rezaei and Xin Liu. Towards the infeasibility of membership inference on deep models. *ArXiv*, abs/2005.13702, 2020.

T. Mitchell Roddenberry and Santiago Segarra. Hodgenet: Graph neural networks for edge data. *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pp. 220–224, 2019.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5558–5567. PMLR, 2019. URL https://proceedings.mlr.press/v97/sablayrolles19a.html.

Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. 2018. URL https://arxiv.org/abs/1806.01246.

Joel A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012. doi: 10.1007/s10208-011-9099-z.

Xiuling Wang and Wendy Hui Wang. Subgraph structure membership inference attacks against graph neural networks. *Proc. Priv. Enhancing Technol.*, 2024:268–290, 2024.

Hanrui Wu, Andy M. Yip, Jinyi Long, Jia Zhang, and Michael K. Ng. Simplicial complex neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46:561–575, 2024.

Maosheng Yang and Elvin Isufi. Convolutional learning on simplicial complexes. *ArXiv*, abs/2301.11163, 2023.

Maosheng Yang, Elvin Isufi, and Geert Leus. Simplicial convolutional neural networks. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8847–8851, 2022a. doi: 10.1109/ICASSP43922.2022.9746017.

Maosheng Yang, Geert Leus, and Elvin Isufi. Hodge-aware convolutional learning on simplicial complexes. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=Nm5sp09Q25.

Ruochen Yang, Frederic Sala, and Paul Bogdan. Efficient representation learning for higher-order data with simplicial complexes. In *Proceedings of the First Learning on Graphs Conference*, volume 198 of *Proceedings of Machine Learning Research*, pp. 13:1–13:21. PMLR, 2022b. URL https://proceedings.mlr.press/v198/yang22a.html.

Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 40–48. PMLR, 2016. URL https://proceedings.mlr.press/v48/yanga16.html.

Xiaoyan Yin, Wanyu Lin, Kexin Sun, Chun Wei, and Yanjiao Chen. A2s2-gnn: Rigging gnn-based social status by adversarial attacks in signed social networks. *IEEE Transactions on Information Forensics and Security*, 18:206–220, 2023.

Huixin Zhan and Victor S. Sheng. Privacy-preserving representation learning for text-attributed networks with simplicial complexes. *ArXiv*, abs/2302.04383, 2023.

Huixin Zhan, Kun Zhang, Keyi Lu, and Victor S. Sheng. Measuring the privacy leakage via graph reconstruction attacks on simplicial neural networks (student abstract). *ArXiv*, abs/2302.04373, 2023.

Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. In *Advances in Neural Information Processing Systems*, volume 31, pp. 5171–5181, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/53f0d7c537d99b3824f0f99d62ea2428-Abstract.html.

Odin Zhang, Tianyue Wang, Gaoqi Weng, Dejun Jiang, Ning Wang, Xiaorui Wang, Huifeng Zhao, Jialu Wu, Ercheng Wang, Guangyong Chen, Yafeng Deng, Peichen Pan, Yu Kang, Chang-Yu Hsieh, and Tingjun Hou. Learning on topological surface and geometric structure for 3d molecular generation. *Nature Computational Science*, 3:849 – 859, 2023.

Zhikun Zhang, Min Chen, Michael Backes, Yun Shen, and Yang Zhang. Inference attacks against graph neural networks. *ArXiv*, abs/2110.02631, 2021.

Zhenzhu Zheng and Xi Peng. Self-guidance: Improve deep neural network generalization via knowledge distillation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3203–3212, January 2022.

Cai Zhou, Xiyuan Wang, and Muhan Zhang. Facilitating graph neural networks with random walk on simplicial complexes. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/345208bdbbb6104616311dfc1d093fe7-Abstract-Conference.html.

APPENDIX OVERVIEW

Our appendix is organized as follows:

- Appendix A: Notation . . . 14
- Appendix B: Supplementary Proofs . . . 16
  - Appendix B.1: Proof of Proposition 4.2 . . . 16
  - Appendix B.2: Perturbation of Hodge Laplacians under Edge Operations . . . 16
  - Appendix B.3: Proof of Theorem 5.1 . . . 19
- Appendix C: Implementation Details: Protocol-Side Structural Attack, SA-BB, and DTPI . . . 25
  - Appendix C.1: Protocol-Side Structural Attack . . . 25
  - Appendix C.2: Structure-Aware Black-Box (SA-BB) MIA . . . 25
  - Appendix C.3: Dynamic Topological Pruning and Injection (DTPI) . . . 26
- Appendix D: Baselines, Datasets, and Additional Experiments . . . 29
  - Appendix D.1: Datasets . . . 29
  - Appendix D.2: Detailed Backbones and Threat Models . . . 30
  - Appendix D.3: Defense Baselines and Hyperparameters . . . 30
- Appendix E: Additional Results and Analyses . . . 32
  - Appendix E.1: Protocol-Side Structural Leakage . . . 32
  - Appendix E.2: Full/Order-1/Graph-only Ablation . . . 33
  - Appendix E.3: Low-FPR Evaluation of Attacks and Defenses . . . 34
  - Appendix E.4: Early-Stopping Controls and Non-Monotonic Privacy Trajectories . . . 34
  - Appendix E.5: SA-BB Shadow Transfer and Neighbour Ablation . . . 36
  - Appendix E.6: Biphasic Leakage Dynamics: Inductive Bias vs. Overfitting . . . 36
  - Appendix E.7: Supplementary Experiments . . . 37
  - Appendix E.8: Additional Statements . . . 38

A NOTATION

We list only the symbols that are central to our theoretical analysis, attacks, and defense mechanisms. All other symbols follow standard conventions and are defined at their first appearance in the main text.

Table 3: Key notations used throughout the paper.

Symbol	Meaning
Simplicial complex and skeleton	
$S = (V, E, T)$	Simplicial complex up to order 2: vertices V, edges E, triangles T.
$G = \pi(S)$	1-skeleton (graph) of S; $\pi(\cdot)$ is the skeletonization map.
Hodge Laplacians and perturbations	
B_k	Oriented boundary/incidence matrix between $(k-1)$- and k-simplices.
L_k	k-th Hodge Laplacian.
$L_{k,\ell}$, $L_{k,u}$	Lower/upper terms: $L_{k,\ell} = B_k^\top B_k$, $L_{k,u} = B_{k+1} B_{k+1}^\top$.
$L_{k-1,\ell}$, $L_{k-1,u}$	Lower/upper Hodge Laplacians on boundary $(k-1)$-faces.
$\Delta_e(\cdot)$	Perturbation of an operator induced by operating on an edge e (add/delete).
Representations and exposure-aware information	
Z_G, Z_S	Learned representations from G only and from full S, respectively.
$I(Y; \mathcal{H} \mid G)$	Conditional MI: intrinsic higher-order task signal beyond the skeleton G.

Symbol	Meaning
R	Construction-dependent operator cues made available by the chosen operators.
$I(Y;\mathcal{H},R \mid G)$	Task-relevant information beyond the baseline view G, equivalently the excess achievable information of exposure-aware encoders over graph-restricted ones.
Prediction responses and SA-BB	
M	Membership indicator of a target instance.
O	Prediction probabilities for the query and its neighbours.
$\mathcal{C}_k$	Public candidate k-simplices, fixed before the membership split.
$\mathcal{N}(\sigma)$	Public candidates sharing a complete boundary face with σ.
$\mathcal{N}_B(\sigma), n_\sigma$	Selected neighbour set and its size, bounded by budget B.
$p_\sigma, H(p_\sigma)$	Query probability and binary prediction entropy.
$\mu_\sigma, s_\sigma, d_\sigma^{\max}$	Mean, standard deviation, and maximum absolute probability gap to selected neighbours.
r_σ	Query confidence rank among its selected neighbours and itself.
$\phi^{\mathrm{SA-BB}}(\sigma)$	Six-dimensional local response feature vector.
DTPI defense	
w_e	Learnable logit parameter for edge e.
q_w	Mask-sampling distribution induced by the logits w.
m_e	Continuous edge mask, parameterized as $m_e = \mathrm{sigmoid}(w_e) \in (0,1)$.
p_e	Edge-retention probability induced by the learned mask logit; in the stochastic appendix view we condition on $p = (p_e)_e$.
$\hat{m}_e$	Sampled binary edge mask used in the forward pass, $\hat{m}_e \in \{0,1\}$.
$\hat{m}_\sigma$	Sampled higher-order simplex mask induced by the boundary-edge masks.
$\mathcal{M}$	Mask operator applied to the complex (edge-level masks extended to higher-order simplices via boundary consistency).
$\hat{\mathcal{M}}$	Sampled mask realization used in the stochastic forward pass.
$\bar{T}(e)$	Normalized cascade score used to weight DTPI regularization.
$\mathcal{E}_{\mathrm{obs}}^{\mathrm{raw}}, \mathcal{E}_{\mathrm{bg}}^{\mathrm{raw}}$	Raw observed-edge and raw background-edge candidate pools before lexicographic filtering.
$\mathcal{E}_{\mathrm{obs}}, \mathcal{E}_{\mathrm{bg}}$	Active observed-edge and background-edge candidate sets retained after DTPI's lexicographic filtering.
Query perturbation, privacy, and cascade quantities	
Σ_q	Ambient candidate family of queried q-simplices used for common pre/post-operation evaluation.
ξ_u	Random binary mask realization for edge u in the stochastic DTPI analysis.
$\mathcal{N}(e)$	Local dependency neighborhood of edge variables whose reveal can affect the coupled perturbation induced by editing e.
$\mathcal{F}_t$	Reveal filtration over the local edge-mask variables in $\mathcal{N}(e)$.
$\widetilde{S}^{(1,e)}, \widetilde{S}^{(0,e)}$	Coupled randomized masked complexes obtained by forcing edge e on/off while keeping all other sampled edge masks identical.
$R_{k,*}(e), V_{k,*}(e)$	One-step influence and predictable quadratic-variation bounds used in the matrix-Freedman argument.
C_{stab}	Query-uniform stability constant transferring operator perturbation to query-level loss perturbation in the randomized analysis.
$m_k(e)$	Number of affected k-simplices induced by operating on edge e.
$t_r(e), T(e)$	Clique-count cascade at level r and aggregate cascade size; under flag completion, $T(e) = 1 + \sum_{r=0}^{K-2} t_r(e)$.
$\Delta^{\max}$	Maximum coface degree used in the perturbation bound.

B SUPPLEMENTARY PROOFS

B.1 PROOF OF PROPOSITION 4.2

We briefly outline the information-theoretic reasoning behind Proposition 4.2, mainly as a conceptual justification for the exposure-aware view used in the main text.

Proof. For the utility term, consider the difference between the best exposure-aware information and the best graph-restricted information:

$$\sup_{\phi} I\big(Y;\phi(G,\mathcal{H},R)\big) - \sup_{f} I\big(Y;f(G)\big), \tag{11}$$

where ϕ ranges over measurable exposure-aware encoders on $(G,\mathcal{H},R)$ and f ranges over measurable graph-restricted encoders on G.

By the Data Processing Inequality, for any measurable ϕ and f,

$$I\big(Y;\phi(G,\mathcal{H},R)\big) \le I(Y;G,\mathcal{H},R), \qquad I\big(Y;f(G)\big) \le I(Y;G). \tag{12}$$

Both upper bounds are tight by taking the identity map on the corresponding view. Hence

$$\sup_{\phi} I\big(Y;\phi(G,\mathcal{H},R)\big) - \sup_{f} I\big(Y;f(G)\big) = I(Y;G,\mathcal{H},R) - I(Y;G) = I(Y;\mathcal{H},R \mid G). \tag{13}$$

Applying the chain rule for mutual information yields

$$I(Y;\mathcal{H},R \mid G) = I(Y;\mathcal{H} \mid G) + I(Y;R \mid G,\mathcal{H}). \tag{14}$$

For the privacy term, under a label-controlled view, the chain rule directly gives

$$I(M;R,O \mid G,Y) = I(M;R \mid G,Y) + I(M;O \mid G,Y,R). \tag{15}$$

This proves the proposition. □

B.2 PERTURBATION OF HODGE LAPLACIANS UNDER EDGE OPERATIONS

Let S be a (finite) simplicial complex up to maximal order K with vertex set V and k-simplices S_k for each $k \in \{0,1,\dots,K\}$. Let $B_k \in \mathbb{R}^{|S_{k-1}|\times|S_k|}$ be the (oriented) boundary matrix. The k-th Hodge Laplacian decomposes as

$$L_k \;=\; L_{k,\ell} + L_{k,u} \;=\; B_k^\top B_k \;+\; B_{k+1}B_{k+1}^\top, \tag{16}$$

with the convention $B_0 = 0$ and $B_{K+1} = 0$.

We consider *edge operations* on a target pair $e = (u,v)$ where $u,v \in V$. We view an edge operation as modifying the 1-skeleton and then updating the complex by removing/adding all simplices that require the presence of e. Concretely:

- **Deletion:** if $e \in S_1$, remove the 1-simplex e and remove every simplex $\sigma \in S$ such that $e \subseteq \sigma$. Denote the resulting complex by S^{-e}.
- **Addition:** add the 1-simplex e (if absent) and add a collection of higher-order simplices consistent with the modeling assumption (e.g., from data or a completion rule). Denote the resulting complex by S^{+e}.

Both operations induce a cascade of simplex removals/additions across dimensions.

Affected-simplex counts and a unified cascade size. For each $q \in \{1,\dots,K\}$, let

$$m_q(e) \;:=\; \#\{q\text{-simplices that are added/removed by the edge operation on } e\}. \tag{17}$$

We define the aggregate cascade size as

$$T(e) \;:=\; \sum_{q=1}^{K} m_q(e), \tag{18}$$

so that $m_q(e) \le T(e)$ for all $q \in \{1,\dots,K\}$.

B.2.1 (Special case) Cascade counts via common-neighborhood cliques under flag completion

When S is generated by *flag (clique) completion* from an underlying graph, $T(e)$ has a simple closed form.

Let $G = (V, E)$ be a simple undirected graph and let $S = \mathrm{Flag}(G)$ denote its flag (clique) simplicial complex up to maximal order K. Fix a target pair $e = (u, v)$ and define the (open) neighborhood

$$N(u) \;:=\; \{\, w \in V \;:\; (u, w) \in E \,\}. \tag{19}$$

Define the common neighborhood $C(e) := N(u) \cap N(v)$ and the induced subgraph $H_e := G[C(e)]$. For $r \geq 0$, define the clique-count

$$t_r(e) \;:=\; \#\Big\{\, W \subseteq C(e) \;:\; |W| = r + 1, \;\; H_e[W] \text{ is complete} \,\Big\}. \tag{20}$$

A q-simplex is a clique on $q + 1$ vertices. If it contains $e = (u, v)$, then it is of the form $\{u, v\} \cup W$ where $W \subseteq C(e)$, $|W| = q - 1$, and W forms a clique in H_e. Thus $m_1(e) = 1$ and, for $q \geq 2$, $m_q(e) = t_{q-2}(e)$ under edge deletion/addition. Consequently,

$$T(e) = \sum_{q=1}^{K} m_q(e) = 1 + \sum_{r=0}^{K-2} t_r(e),$$

up to the chosen truncation order.

B.2.2 Upper Laplacian perturbation

For $k \in \{0, \ldots, K\}$, the upper Laplacian is $L_{k,u} = B_{k+1} B_{k+1}^\top$. Let b_τ denote the column of B_{k+1} corresponding to a $(k + 1)$-simplex τ. Then

$$L_{k,u} \;=\; \sum_{\tau \in S_{k+1}} b_\tau b_\tau^\top. \tag{21}$$

Each $(k + 1)$-simplex has $(k + 2)$ k-faces, hence b_τ has exactly $(k + 2)$ nonzero entries in $\{\pm 1\}$ and therefore

$$\|b_\tau\|_2^2 \;=\; k + 2, \qquad \|b_\tau b_\tau^\top\|_2 \;=\; \|b_\tau\|_2^2 \;=\; k + 2. \tag{22}$$

Lemma B.1 (Upper Laplacian sensitivity under edge operations)**.** *Let Δ_e denote either edge deletion ($S \mapsto S^{-e}$) or edge addition ($S \mapsto S^{+e}$). Then for every $k \in \{0, \ldots, K - 1\}$,*

$$\big\|\Delta_e L_{k,u}\big\|_2 \;\leq\; (k + 2)\, m_{k+1}(e), \tag{23}$$

where $m_{k+1}(e)$ is the number of $(k + 1)$-simplices added/removed by the operation. In particular, since $m_{k+1}(e) \leq T(e)$ by (18),

$$\big\|\Delta_e L_{k,u}\big\|_2 \;\leq\; (k + 2)\, T(e) \;\leq\; (K + 2)\, T(e). \tag{24}$$

Proof. Let $\mathcal{T}$ be the set of $(k + 1)$-simplices that are added/removed by Δ_e. Since

$$L_{k,u} = \sum_{\tau \in S_{k+1}} b_\tau b_\tau^\top,$$

the perturbation $\Delta_e L_{k,u}$ is obtained by adding/removing exactly the rank-one terms $b_\tau b_\tau^\top$ for $\tau \in \mathcal{T}$, namely

$$\Delta_e L_{k,u} \;=\; \sum_{\tau \in \mathcal{T}} \pm\, b_\tau b_\tau^\top. \tag{25}$$

Taking operator norms and using the triangle inequality,

$$\big\|\Delta_e L_{k,u}\big\|_2 \leq \sum_{\tau \in \mathcal{T}} \big\|b_\tau b_\tau^\top\big\|_2 \;=\; \sum_{\tau \in \mathcal{T}} \|b_\tau\|_2^2 \;=\; \sum_{\tau \in \mathcal{T}} (k + 2) \;=\; (k + 2)\, |\mathcal{T}|. \tag{26}$$

Since $|\mathcal{T}| = m_{k+1}(e)$ by definition, this gives $\|\Delta_e L_{k,u}\|_2 \leq (k + 2) m_{k+1}(e)$. Finally, (24) follows from $m_{k+1}(e) \leq T(e)$ and $(k + 2) \leq (K + 2)$. □

B.2.3 LOWER LAPLACIAN PERTURBATION

For the lower Laplacian $L_{k,\ell} = B_k^\top B_k$, an edge operation modifies B_k by adding/removing columns corresponding to the affected k-simplices. Let B_k' be the post-operation boundary matrix and define $\Delta B_k := B_k' - B_k$ (after embedding into a common ambient index set).

We upper bound $\|B_k\|_2$ via $\|B_k\|_2 \le \sqrt{\|B_k\|_1 \|B_k\|_\infty}$. Since each column of B_k has exactly $(k+1)$ nonzero entries in $\{\pm 1\}$, we have $\|B_k\|_1 \le k+1$. Let $\Delta_{k-1}^{\max}$ denote the maximum coface degree of a $(k-1)$-simplex (number of incident k-simplices). Then each row of B_k has at most $\Delta_{k-1}^{\max}$ nonzeros, so $\|B_k\|_\infty \le \Delta_{k-1}^{\max}$. Therefore,

$$\|B_k\|_2 \;\le\; \sqrt{(k+1)\Delta_{k-1}^{\max}}. \tag{27}$$

Lemma B.2 (Lower Laplacian perturbation bound). *For any $k \in \{1, \dots, K\}$,*

$$\begin{aligned} \Delta_e L_{k,\ell} &= B_k'^\top B_k' - B_k^\top B_k && (28)\\ &= B_k^\top \Delta B_k + (\Delta B_k)^\top B_k + (\Delta B_k)^\top (\Delta B_k), && (29) \end{aligned}$$

and therefore

$$\left\|\Delta_e L_{k,\ell}\right\|_2 \;\le\; 2\|B_k\|_2 \,\|\Delta B_k\|_2 \;+\; \|\Delta B_k\|_2^2. \tag{30}$$

Moreover, if $m_k(e)$ k-simplices are added/removed, then

$$\|\Delta B_k\|_2 \;\le\; \|\Delta B_k\|_F \;=\; \sqrt{(k+1)\, m_k(e)}. \tag{31}$$

In particular, combining (30)–(31) with (27) yields

$$\left\|\Delta_e L_{k,\ell}\right\|_2 \;\le\; 2(k+1)\sqrt{\Delta_{k-1}^{\max}\, m_k(e)} \;+\; (k+1) m_k(e). \tag{32}$$

Consequently, since $m_k(e) \le T(e)$ by (18), we also have

$$\left\|\Delta_e L_{k,\ell}\right\|_2 \;\le\; 2(k+1)\sqrt{\Delta_{k-1}^{\max}\, T(e)} \;+\; (k+1) T(e). \tag{33}$$

Since an edge operation changes the edge itself, $T(e) \ge 1$, and therefore the same bound admits the linear envelope

$$\left\|\Delta_e L_{k,\ell}\right\|_2 \;\le\; \Big(2(k+1)\sqrt{\Delta_{k-1}^{\max}} + (k+1)\Big) T(e). \tag{34}$$

Proof. Using $B_k' = B_k + \Delta B_k$, we expand

$$\begin{aligned} \Delta_e L_{k,\ell} &= B_k'^\top B_k' - B_k^\top B_k \\ &= (B_k + \Delta B_k)^\top (B_k + \Delta B_k) - B_k^\top B_k \\ &= B_k^\top \Delta B_k + (\Delta B_k)^\top B_k + (\Delta B_k)^\top (\Delta B_k). \end{aligned} \tag{35}$$

Taking operator norms and using subadditivity together with $\|X^\top\|_2 = \|X\|_2$, we obtain

$$\begin{aligned} \|\Delta_e L_{k,\ell}\|_2 &\le \|B_k^\top \Delta B_k\|_2 + \|(\Delta B_k)^\top B_k\|_2 + \|(\Delta B_k)^\top (\Delta B_k)\|_2 \\ &\le \|B_k\|_2 \,\|\Delta B_k\|_2 + \|\Delta B_k\|_2 \,\|B_k\|_2 + \|\Delta B_k\|_2^2 \\ &= 2\|B_k\|_2 \,\|\Delta B_k\|_2 + \|\Delta B_k\|_2^2, \end{aligned} \tag{36}$$

which is (30).

To relate $\|\Delta B_k\|_2$ to the number of affected k-simplices, note that an edge operation adds/removes exactly $m_k(e)$ columns of B_k (corresponding to the affected k-simplices). Each such column has exactly $(k+1)$ nonzero entries in $\{\pm 1\}$, hence

$$\|\Delta B_k\|_2 \;\le\; \|\Delta B_k\|_F \;=\; \sqrt{\sum_{j=1}^{m_k(e)} \|c_j\|_2^2} \;=\; \sqrt{m_k(e)\,(k+1)}, \tag{37}$$

which is (31).

Finally, combining (30) with (31) and the coface-degree bound $\|B_k\|_2 \le \sqrt{(k+1)\Delta_{k-1}^{\max}}$ from (27), we obtain

$$\begin{aligned}\|\Delta_e L_{k,\ell}\|_2 &\le 2\|B_k\|_2\sqrt{(k+1)m_k(e)} + \left(\sqrt{(k+1)m_k(e)}\right)^2 \\ &\le 2\sqrt{(k+1)\Delta_{k-1}^{\max}}\sqrt{(k+1)m_k(e)} + (k+1)m_k(e) \\ &= 2(k+1)\sqrt{\Delta_{k-1}^{\max} m_k(e)} + (k+1)m_k(e),\end{aligned} \tag{38}$$

which is (32). The bound (33) then follows immediately from $m_k(e) \le T(e)$. Since $T(e) \ge 1$, we also have $\sqrt{T(e)} \le T(e)$, which gives the linear envelope (34). □

B.2.4 Degree-sensitive impact under normalized propagation

The degree heuristic used by DTPI follows from the direct local contribution of normalized incidence operators. Fix the degree normalizer $D_0 = \mathrm{diag}(d(v)+\epsilon)$ with $\epsilon > 0$, define $\bar{B}_1 := D_0^{-1/2} B_1$, and let b_e be the incidence column of $e = (u, v)$. The normalized column $\bar{b}_e = D_0^{-1/2} b_e$ has entries $\pm 1/\sqrt{d(u)+\epsilon}$ and $\pm 1/\sqrt{d(v)+\epsilon}$. The direct contribution of toggling e is bounded, for any node signal x, by

$$\left|\bar{b}_e^\top x\right| \le \left(\frac{1}{\sqrt{d(u)+\epsilon}} + \frac{1}{\sqrt{d(v)+\epsilon}}\right)\|x\|_\infty. \tag{39}$$

If the normalizer is recomputed after the edit, neighboring columns also change; Eq. (39) isolates the direct endpoint-dependent term used by DTPI's screening rule. Thus low-degree endpoints yield larger local sensitivity under normalized propagation.

B.3 Proof of Theorem 5.1

This subsection gives the full argument behind Theorem 5.1. We first derive a deterministic query-level perturbation bound for a fixed edge operation. We then turn to the randomized mask mechanism used by DTPI, apply matrix Freedman to the resulting operator perturbations, and reduce the result to a high-probability envelope indexed by $T(e)$. The deterministic lemma supplies the operator-to-loss stability constants; the randomized part adds the moment and local-dependence control needed for masks. Fix a query order q and an ambient candidate family Σ_q. We evaluate the same target query $\sigma \in \Sigma_q$ before and after the edge operation, even if the operation changes whether σ belongs to the post-operation complex. Assume throughout that $\ell(\cdot, y)$ is L_ℓ-Lipschitz and that the model is query-uniformly Lipschitz with respect to the lower/upper Hodge operators:

$$\begin{aligned}&|\ell(a,y) - \ell(b,y)| \le L_\ell\|a-b\|_2, \\ &\|f_S(\sigma) - f_{S'}(\sigma)\|_2 \le K_{\mathrm{net}}\left[\sum_{k=1}^{K} a_{k,\ell}\|\Delta_e L_{k,\ell}\|_2 + \sum_{k=0}^{K-1} a_{k,u}\|\Delta_e L_{k,u}\|_2\right],\end{aligned} \tag{40}$$

and define

$$\Gamma := \max\left\{\max_{1\le k\le K} a_{k,\ell}, \max_{0\le k\le K-1} a_{k,u}\right\}, \qquad \Delta^{\max} := \max_{0\le r\le K-1}\Delta_r^{\max}. \tag{41}$$

Lemma B.3 (Deterministic fixed-complex query perturbation)**.** *Fix a query order q, an ambient candidate family Σ_q, and a target query $\sigma \in \Sigma_q$ with label y_σ. Under the assumptions above, for an edge operation on e (deletion or addition) with cascade size $T(e)$,*

$$|\ell(f_S(\sigma), y_\sigma) - \ell(f_{S'}(\sigma), y_\sigma)| \le L_\ell\, K_{\mathrm{net}}\,\Gamma\left[2(K+1)\sqrt{K\,\Delta^{\max}\,T(e)} + (2K+3)\,T(e)\right]. \tag{42}$$

Proof. By the L_ℓ-Lipschitzness of the loss and the query-uniform operator stability assumption (40),

$$\begin{aligned}|\ell(f_S(\sigma), y_\sigma) - \ell(f_{S'}(\sigma), y_\sigma)| &\le L_\ell\,\|f_S(\sigma) - f_{S'}(\sigma)\|_2 \\ &\le L_\ell\, K_{\mathrm{net}}\left[\sum_{k=1}^{K} a_{k,\ell}\|\Delta_e L_{k,\ell}\|_2 + \sum_{k=0}^{K-1} a_{k,u}\|\Delta_e L_{k,u}\|_2\right].\end{aligned} \tag{43}$$

Substituting the deterministic lower- and upper-Laplacian perturbation bounds from Lemmas B.2 and B.1, and using $a_{k,\ell}, a_{k,u} \le \Gamma$ together with $(k+1) \le K+1$ and $(k+2) \le K+2$, yields

$$\begin{aligned}
&|\ell(f_S(\sigma), y_\sigma) - \ell(f_{S'}(\sigma), y_\sigma)| \le L_\ell \, K_{\mathrm{net}} \, \Gamma \\
&\quad \times \left[2(K+1) \sum_{k=1}^{K} \sqrt{\Delta_{k-1}^{\max} \, m_k(e)} + (K+1) \sum_{k=1}^{K} m_k(e) + (K+2) \sum_{k=0}^{K-1} m_{k+1}(e) \right] \\
&= L_\ell \, K_{\mathrm{net}} \, \Gamma \left[2(K+1) \sum_{k=1}^{K} \sqrt{\Delta_{k-1}^{\max} \, m_k(e)} + (2K+3) \, T(e) \right],
\end{aligned} \tag{44}$$

because $\sum_{k=1}^{K} m_k(e) = T(e)$ and $\sum_{k=0}^{K-1} m_{k+1}(e) = T(e)$. Finally, Cauchy–Schwarz gives

$$\begin{aligned}
\sum_{k=1}^{K} \sqrt{\Delta_{k-1}^{\max} \, m_k(e)} &\le \sqrt{\Big(\sum_{r=0}^{K-1} \Delta_r^{\max} \Big) \Big(\sum_{k=1}^{K} m_k(e) \Big)} \\
&= \sqrt{\Big(\sum_{r=0}^{K-1} \Delta_r^{\max} \Big) T(e)} \le \sqrt{K \, \Delta^{\max} \, T(e)}.
\end{aligned} \tag{45}$$

Substituting this estimate into the previous display gives (42). □

Lemma B.3 is the deterministic precursor of the randomized-mask bound below. Its proof uses Lemmas B.2–B.1 to convert $T(e)$ into Hodge-operator perturbations, and then uses the stability factor $L_\ell K_{\mathrm{net}} \Gamma$ to pass these perturbations to query-level loss. In the randomized setting, Assumption B.6 is an operator-level moment envelope for masked perturbations: its mean term is the randomized analogue of the deterministic operator bounds, while its variation and increment terms encode the local-dependence control needed for matrix Freedman. Assumption B.7 then applies the same loss/network stability mechanism to the coupled masked complexes.

Randomized mask model and coupled edge edit. We now move to the randomized edge-mask mechanism used by DTPI. Condition on the ambient operator space G and the learned edge-retention probabilities $p = (p_u)_{u \in \mathcal{E}_{\mathrm{all}}}$ induced by the mask logits. For each edge $u \in \mathcal{E}_{\mathrm{all}}$, let

$$\xi_u \in \{0, 1\}, \qquad \Pr(\xi_u = 1 \mid G, p) = p_u,$$

and assume $\{\xi_u\}_{u \in \mathcal{E}_{\mathrm{all}}}$ are conditionally independent given (G, p). For a target edge e, write $\xi_{-e} := (\xi_u)_{u \ne e}$ and define the coupled masked complexes

$$\widetilde{S}^{(1,e)} := \widetilde{S}(\xi_e = 1, \xi_{-e}), \qquad \widetilde{S}^{(0,e)} := \widetilde{S}(\xi_e = 0, \xi_{-e}),$$

which respectively force e on and off while keeping all other sampled edge masks identical. For each k, define the random operator differences

$$\widetilde{\Delta}_e L_{k,\ell} := L_{k,\ell}\big(\widetilde{S}^{(1,e)}\big) - L_{k,\ell}\big(\widetilde{S}^{(0,e)}\big), \qquad \widetilde{\Delta}_e L_{k,u} := L_{k,u}\big(\widetilde{S}^{(1,e)}\big) - L_{k,u}\big(\widetilde{S}^{(0,e)}\big).$$

To capture the dependence induced by shared edge masks, define the local dependency neighborhood

$$\mathcal{N}(e) := \Big\{ u \in \mathcal{E}_{\mathrm{all}} \backslash \{e\} : \exists \text{ a realization of } \xi_{-e} \text{ such that toggling } \xi_u \text{ changes } \widetilde{\Delta}_e L_{k,*} \text{ for some } (k, *) \Big\}. \tag{46}$$

This definition is safe even for normalized operators, as it includes indirect changes through degree normalizers. Let $\mathcal{N}(e) = \{u_1, \dots, u_m\}$ and define the reveal filtration $\{\mathcal{F}_t\}_{t=0}^{m}$, where each $\mathcal{F}_t$ is generated by $\xi_{u_1}, \dots, \xi_{u_t}$. For each k and $* \in \{\ell, u\}$, define the Doob matrix martingale

$$Y_t^{(k,*)} := \mathbb{E}\Big[\widetilde{\Delta}_e L_{k,*} \mid \mathcal{F}_t, G, p \Big] - \mathbb{E}\Big[\widetilde{\Delta}_e L_{k,*} \mid G, p \Big], \qquad X_t^{(k,*)} := Y_t^{(k,*)} - Y_{t-1}^{(k,*)}.$$

Write $N_k := |S_k|$ for the relevant operator dimension.

Assumption B.4 (One-step influence and predictable variation)**.** For each k and $* \in \{\ell, u\}$, there exist edge-dependent quantities $R_{k,*}(e)$ and $V_{k,*}(e)$ such that

$$\|X_t^{(k,*)}\|_2 \le R_{k,*}(e) \quad \text{a.s. for all } t = 1, \dots, m, \tag{47}$$

and

$$\left\| \sum_{t=1}^{m} \mathbb{E}\Big[\big(X_t^{(k,*)}\big)^2 \mid \mathcal{F}_{t-1}, G, p \Big] \right\|_2 \le V_{k,*}(e) \quad \text{a.s.} \tag{48}$$

Assumption B.4 isolates the probabilistic part of the masked edit: after all deterministic geometry has been absorbed into local influence and variation terms, the remaining task is to control the spectral norm of the random operator difference. The next lemma performs exactly this concentration step.

Lemma B.5 (Matrix-Freedman operator concentration)**.** *Under Assumption B.4, for every k and $* \in \{\ell, u\}$ and every $\delta \in (0,1)$, the matrix martingale Freedman inequality (Tropp, 2012, Theorem 1.6), a noncommutative analogue of the classical martingale bound of Freedman Freedman (1975), gives*

$$\Pr\left(\|\widetilde{\Delta}_e L_{k,*}\|_2 \le \left\| \mathbb{E}\Big[\widetilde{\Delta}_e L_{k,*} \mid G, p \Big] \right\|_2 + \sqrt{2V_{k,*}(e) \log \frac{2N_k}{\delta}} + \frac{2R_{k,*}(e)}{3} \log \frac{2N_k}{\delta} \;\middle|\; G, p \right) \ge 1 - \delta. \tag{49}$$

Proof. Condition on (G, p). Revealing the masks in the local dependency neighborhood $\mathcal{N}(e)$ through the filtration $\{\mathcal{F}_t\}_{t=0}^m$ defines a self-adjoint Doob matrix martingale

$$Y_t^{(k,*)} = \mathbb{E}\Big[\widetilde{\Delta}_e L_{k,*} \mid \mathcal{F}_t, G, p \Big] - \mathbb{E}\Big[\widetilde{\Delta}_e L_{k,*} \mid G, p \Big],$$

with increments $X_t^{(k,*)} = Y_t^{(k,*)} - Y_{t-1}^{(k,*)}$. In particular,

$$Y_m^{(k,*)} = \widetilde{\Delta}_e L_{k,*} - \mathbb{E}\Big[\widetilde{\Delta}_e L_{k,*} \mid G, p \Big].$$

Assumption B.4 gives the two hypotheses required by matrix Freedman Tropp (2012): the almost-sure increment bound $\|X_t^{(k,*)}\|_2 \le R_{k,*}(e)$ and the predictable quadratic-variation bound

$$\left\| \sum_{t=1}^{m} \mathbb{E}\Big[\big(X_t^{(k,*)}\big)^2 \mid \mathcal{F}_{t-1}, G, p \Big] \right\|_2 \le V_{k,*}(e).$$

Applying the self-adjoint matrix Freedman bound in dimension N_k and using the standard dilation/sign argument to control spectral norm rather than only the maximum eigenvalue therefore yields, with conditional probability at least $1 - \delta$,

$$\|Y_m^{(k,*)}\|_2 \le \sqrt{2V_{k,*}(e) \log \frac{2N_k}{\delta}} + \frac{2R_{k,*}(e)}{3} \log \frac{2N_k}{\delta}.$$

The triangle inequality then gives

$$\|\widetilde{\Delta}_e L_{k,*}\|_2 \le \left\| \mathbb{E}\Big[\widetilde{\Delta}_e L_{k,*} \mid G, p \Big] \right\|_2 + \|Y_m^{(k,*)}\|_2,$$

which is exactly (49). □

Assumption B.6 (Cascade-controlled moment envelope)**.** For each k and $* \in \{\ell, u\}$, there exist constants $\bar{a}_{k,*}, \bar{b}_{k,*}, \bar{r}_{k,*} \ge 0$, independent of the edited edge e but allowed to depend on K, the operator type, normalization constants, coface-degree bounds, and local overlap, such that

$$\left\| \mathbb{E}\Big[\widetilde{\Delta}_e L_{k,*} \mid G, p \Big] \right\|_2 \le \bar{a}_{k,*}\, T(e), \qquad V_{k,*}(e) \le \bar{b}_{k,*}\, T(e), \qquad R_{k,*}(e) \le \bar{r}_{k,*}. \tag{50}$$

The mean term in Assumption B.6 is anchored by the deterministic operator bounds above. For the upper channel, Lemma B.1 allows $\bar{a}_{k,u} = k + 2$ (or the coarser uniform choice $K + 2$). For the lower channel, the linear envelope (34) allows

$$\bar{a}_{k,\ell} = 2(k+1)\sqrt{\Delta_{k-1}^{\max}} + (k+1).$$

The variation and increment constants capture the randomness of the coupled masks rather than loss stability. A sufficient local condition is that each revealed mask variable affects at most ρ local simplex atoms and that each atom has operator norm at most $C_{k,*}$, with

$$C_{k,u} = k + 2, \qquad C_{k,\ell} = 2(k+1)\sqrt{\Delta_{k-1}^{\max}} + (k+1).$$

Then one can take edge-independent bounds of the form $\bar{r}_{k,*} = \rho C_{k,*}$ and $\bar{b}_{k,*} = \rho C_{k,*}^2$, up to normalization-dependent constants.

The concentration lemma controls operator perturbations, but the theorem needs an attacker-visible loss perturbation. We therefore state the randomized analogue of the network stability condition used in Lemma B.3.

Assumption B.7 (Query-uniform randomized operator stability)**.** There exists a constant $C_{\mathrm{stab}} \geq 0$ such that for every $\sigma \in \Sigma_q$,

$$\left|\ell\big(f_{\widetilde{S}^{(1,e)}}(\sigma), y_\sigma\big) - \ell\big(f_{\widetilde{S}^{(0,e)}}(\sigma), y_\sigma\big)\right| \leq C_{\mathrm{stab}} \left[\sum_{k=1}^{K} \|\widetilde{\Delta}_e L_{k,\ell}\|_2 + \sum_{k=0}^{K-1} \|\widetilde{\Delta}_e L_{k,u}\|_2\right]. \tag{51}$$

In the SCN stability setting used for the deterministic bound, Assumption B.7 is obtained from (40) by absorbing the channel weights into Γ, namely

$$C_{\mathrm{stab}} = L_\ell K_{\mathrm{net}} \Gamma. \tag{52}$$

With the concentration envelope and the query-uniform stability condition in place, the following theorem combines them into the finite-δ version of the main $T(e)$ leverage statement.

Theorem B.8 (Sharpened finite-δ form of Theorem 5.1)**.** *Condition on the ambient operator space G and the learned edge-retention probabilities p. Fix a query order q, an ambient candidate family Σ_q, and a candidate edge e. Under Assumptions B.4–B.7, there exist constants $c_1, c_2, c_3, c_4 > 0$, independent of e, such that for any $\delta \in (0,1)$, with conditional probability at least $1-\delta$ over the randomized edge-mask realization given (G, p),*

$$\sup_{\sigma \in \Sigma_q} \left|\ell\big(f_{\widetilde{S}^{(1,e)}}(\sigma), y_\sigma\big) - \ell\big(f_{\widetilde{S}^{(0,e)}}(\sigma), y_\sigma\big)\right| \leq c_1 T(e) + c_2\sqrt{T(e)\log\frac{c_3}{\delta}} + c_4 \log\frac{c_3}{\delta} \tag{53}$$

Proof. Apply Lemma B.5 to each lower and upper operator with the budget

$$\delta_{k,*} := \frac{\delta}{2K}.$$

Since there are K lower operators and K upper operators, a union bound shows that with conditional probability at least

$$1 - \sum_{k,*} \delta_{k,*} = 1 - \delta,$$

the bound (49) holds simultaneously for every relevant pair $(k, *)$. On this event, let

$$N_{\max} := \max_{0 \leq k \leq K} N_k, \qquad L_\delta := \log\frac{4K\, N_{\max}}{\delta}.$$

Because $\delta_{k,*} = \delta/(2K)$ and $N_k \leq N_{\max}$, Assumption B.6 gives, for every $(k, *)$,

$$\|\widetilde{\Delta}_e L_{k,*}\|_2 \leq \bar{a}_{k,*}\, T(e) + \sqrt{2\bar{b}_{k,*}\, T(e)\, L_\delta} + \frac{2\bar{r}_{k,*}}{3} L_\delta. \tag{54}$$

Substituting these simultaneous operator bounds into the randomized stability assumption (51) yields

$$\begin{aligned}\sup_{\sigma \in \Sigma_q} \left|\ell\big(f_{\widetilde{S}^{(1,e)}}(\sigma), y_\sigma\big) - \ell\big(f_{\widetilde{S}^{(0,e)}}(\sigma), y_\sigma\big)\right| &\leq C_{\mathrm{stab}} \sum_{k=1}^{K} \left(\bar{a}_{k,\ell} T(e) + \sqrt{2\bar{b}_{k,\ell} T(e) L_\delta} + \tfrac{2\bar{r}_{k,\ell}}{3} L_\delta\right) \\ &\quad + C_{\mathrm{stab}} \sum_{k=0}^{K-1} \left(\bar{a}_{k,u} T(e) + \sqrt{2\bar{b}_{k,u} T(e) L_\delta} + \tfrac{2\bar{r}_{k,u}}{3} L_\delta\right).\end{aligned} \tag{55}$$

Since $T(e)$, $\sqrt{T(e)L_\delta}$, and L_δ are common across all summands, collecting coefficients gives

$$\sup_{\sigma\in\Sigma_q}\left|\ell\big(f_{\widetilde{S}^{(1,e)}}(\sigma),y_\sigma\big)-\ell\big(f_{\widetilde{S}^{(0,e)}}(\sigma),y_\sigma\big)\right| \le \underbrace{C_{\mathrm{stab}}\left(\sum_{k=1}^{K}\bar{a}_{k,\ell}+\sum_{k=0}^{K-1}\bar{a}_{k,u}\right)}_{=:c_1} T(e)$$
$$+\underbrace{C_{\mathrm{stab}}\left(\sum_{k=1}^{K}\sqrt{2\bar{b}_{k,\ell}}+\sum_{k=0}^{K-1}\sqrt{2\bar{b}_{k,u}}\right)}_{=:c_2}\sqrt{T(e)L_\delta}$$
$$+\underbrace{\frac{2C_{\mathrm{stab}}}{3}\left(\sum_{k=1}^{K}\bar{r}_{k,\ell}+\sum_{k=0}^{K-1}\bar{r}_{k,u}\right)}_{=:c_4} L_\delta. \tag{56}$$

Setting $c_3 := 4K\,N_{\max}$ gives (53). With the SCN stability instantiation (52), the constants c_1, c_2, c_4 inherit the common factor $L_\ell K_{\mathrm{net}}\Gamma$. □

Theorem 5.1 is used as a guide to edge editing: local cascade size controls a high-probability envelope on attacker-visible query perturbations, so DTPI allocates pruning budget toward high-leverage observed edges and injection budget toward high-leverage background decoys. The theorem is stated at the level of loss variation, whereas common MIAs often act by thresholding a scalar attack score such as loss, confidence, or a gradient-derived statistic. The next assumption and corollary translate the same $T(e)$-controlled perturbation into stability of threshold-based attack advantage.

Assumption B.9 (Threshold-score regularity)**.** Let $S(\sigma;\widetilde{S})\in\mathbb{R}$ be a scalar attack score on the queried simplex σ, and let

$$A_t(\sigma;\widetilde{S}) := \mathbf{1}\{S(\sigma;\widetilde{S})\ge t\}$$

be the threshold attack at level t. Assume there exists $L_S\ge 0$ such that

$$\left|S\Big(\sigma;\widetilde{S}^{(1,e)}\Big)-S\Big(\sigma;\widetilde{S}^{(0,e)}\Big)\right| \le L_S\left|\ell\big(f_{\widetilde{S}^{(1,e)}}(\sigma),y_\sigma\big)-\ell\big(f_{\widetilde{S}^{(0,e)}}(\sigma),y_\sigma\big)\right| \quad \text{for all } \sigma\in\Sigma_q. \tag{57}$$

Corollary B.10 (Threshold-based attack advantage stability)**.** *Under Theorem 5.1 and Assumption B.9, define*

$$\eta_e(\delta) := L_S\left(c_1T(e)+c_2\sqrt{T(e)\log\frac{c_3}{\delta}}+c_4\log\frac{c_3}{\delta}\right).$$

Let $(\Sigma,M,Y)\sim P_{\Sigma,M,Y|G,p}$ be a random queried simplex, its membership indicator, and its task label. For $m\in\{0,1\}$, define the conditional margin mass

$$q_{m,t}(\eta) := \Pr\Big(\Big|S\Big(\Sigma;\widetilde{S}^{(0,e)}\Big)-t\Big|\le\eta\;\Big|\;M=m,G,p,Y\Big). \tag{58}$$

If the balanced-prior absolute threshold advantage is

$$\mathrm{Adv}_t(\widetilde{S}) := \frac{1}{2}\left|\Pr(A_t=1\mid M=1,G,p,Y)-\Pr(A_t=1\mid M=0,G,p,Y)\right|, \tag{59}$$

then for any $\delta\in(0,1)$,

$$\left|\mathrm{Adv}_t\Big(\widetilde{S}^{(1,e)}\Big)-\mathrm{Adv}_t\Big(\widetilde{S}^{(0,e)}\Big)\right| \le \frac{1}{2}\Big(q_{1,t}(\eta_e(\delta))+q_{0,t}(\eta_e(\delta))\Big) \tag{60}$$

with conditional probability at least $1-\delta$ over the randomized edge-mask realization given (G,p).

Proof. Let

$$\mathcal{E}_e(\delta) := \left\{\sup_{\sigma\in\Sigma_q}\left|\ell\big(f_{\widetilde{S}^{(1,e)}}(\sigma),y_\sigma\big)-\ell\big(f_{\widetilde{S}^{(0,e)}}(\sigma),y_\sigma\big)\right| \le c_1T(e)+c_2\sqrt{T(e)\log\frac{c_3}{\delta}}+c_4\log\frac{c_3}{\delta}\right\}$$

be the high-probability event from Theorem 5.1, which has conditional probability at least $1-\delta$ given (G,p). On $\mathcal{E}_e(\delta)$, Assumption B.9 implies

$$\left|S\Big(\sigma;\widetilde{S}^{(1,e)}\Big)-S\Big(\sigma;\widetilde{S}^{(0,e)}\Big)\right|\le\eta_e(\delta)\qquad\forall\sigma\in\Sigma_q.$$

For a fixed queried simplex σ, define

$$A_t^{(1)}(\sigma):=\mathbf{1}\Big\{S(\sigma;\widetilde{S}^{(1,e)})\ge t\Big\},\qquad A_t^{(0)}(\sigma):=\mathbf{1}\Big\{S(\sigma;\widetilde{S}^{(0,e)})\ge t\Big\}.$$

If the two scores differ by at most $\eta_e(\delta)$, then the threshold decision can change only when the baseline score lies inside the margin band $t\pm\eta_e(\delta)$. Equivalently,

$$\left|A_t^{(1)}(\sigma)-A_t^{(0)}(\sigma)\right|\le\mathbf{1}\Big\{\left|S(\sigma;\widetilde{S}^{(0,e)})-t\right|\le\eta_e(\delta)\Big\}.$$

Taking conditional expectations under $M=m$ gives

$$\Big|\Pr(A_t^{(1)}=1\mid M=m,G,p,Y)-\Pr(A_t^{(0)}=1\mid M=m,G,p,Y)\Big|\le q_{m,t}(\eta_e(\delta))$$

on the event $\mathcal{E}_e(\delta)$. Applying this bound for $m=1$ and $m=0$, then taking the difference of the two class-conditional acceptance probabilities and dividing by two, yields

$$\left|\mathrm{Adv}_t\Big(\widetilde{S}^{(1,e)}\Big)-\mathrm{Adv}_t\Big(\widetilde{S}^{(0,e)}\Big)\right|\le\frac{1}{2}\Big(q_{1,t}(\eta_e(\delta))+q_{0,t}(\eta_e(\delta))\Big)$$

on $\mathcal{E}_e(\delta)$, which proves the stated high-probability inequality. □

B.3.1 Averaged implication for the pruning regularizer.

Under the linear margin-mass condition

$$q_{m,t}(\eta)\le\kappa_t\eta,\qquad m\in\{0,1\},$$

for any fixed $\delta\in(0,1)$, Corollary B.10 yields a per-edge local threshold-attack variation bound of the form

$$\Delta_e^{(t)}\ \le\ a_{t,\delta}\,T(e)+b_{t,\delta}\,\sqrt{T(e)}+c_{t,\delta},\qquad e\in\mathcal{E}_{\mathrm{obs}},$$

where $\Delta_e^{(t)}:=\left|\mathrm{Adv}_t(\widetilde{S}^{(1,e)})-\mathrm{Adv}_t(\widetilde{S}^{(0,e)})\right|$, and the constants $a_{t,\delta},b_{t,\delta},c_{t,\delta}>0$ absorb the quantities in Corollary B.10 (in particular, the logarithmic factor and the linear margin-mass coefficient). Therefore, averaging over observed edges with weights p_e gives

$$\frac{1}{P_{\mathrm{obs}}}\sum_{e\in\mathcal{E}_{\mathrm{obs}}}p_e\,\Delta_e^{(t)}\ \le\ a_{t,\delta}\,\frac{\sum_{e\in\mathcal{E}_{\mathrm{obs}}}p_eT(e)}{P_{\mathrm{obs}}}\ +\ b_{t,\delta}\,\frac{\sum_{e\in\mathcal{E}_{\mathrm{obs}}}p_e\sqrt{T(e)}}{P_{\mathrm{obs}}}\ +\ c_{t,\delta},$$

where $P_{\mathrm{obs}}:=\sum_{e\in\mathcal{E}_{\mathrm{obs}}}p_e$. Since $x\mapsto\sqrt{x}$ is concave, Jensen's inequality implies

$$\frac{\sum_{e\in\mathcal{E}_{\mathrm{obs}}}p_e\sqrt{T(e)}}{P_{\mathrm{obs}}}\ \le\ \sqrt{\frac{\sum_{e\in\mathcal{E}_{\mathrm{obs}}}p_eT(e)}{P_{\mathrm{obs}}}}.$$

Hence the p_e-weighted average local threshold-attack variation over observed edges is upper bounded by

$$a_{t,\delta}\,\frac{\sum_{e\in\mathcal{E}_{\mathrm{obs}}}p_eT(e)}{P_{\mathrm{obs}}}\ +\ b_{t,\delta}\sqrt{\frac{\sum_{e\in\mathcal{E}_{\mathrm{obs}}}p_eT(e)}{P_{\mathrm{obs}}}}\ +\ c_{t,\delta},$$

which is a monotone function of the p_e-weighted average cascade size

$$\frac{\sum_{e\in\mathcal{E}_{\mathrm{obs}}}p_eT(e)}{P_{\mathrm{obs}}}.$$

Since $\bar{T}(e)$ is the normalized version of $T(e)$ used by DTPI, this provides analytical support for interpreting the pruning regularizer as controlling an averaged local attackability surrogate, rather than merely serving as a heuristic topology penalty.

C Implementation Details: Protocol-Side Attack, SA-BB, and DTPI

C.1 Protocol-Side Structural Attack (PSA)

To instantiate the protocol-side term $I(M; R \mid G, Y)$, we use the most direct query-aligned cue available in the revealed operators. Let σ be a queried k-simplex and let B_k denote the released boundary/incidence matrix. We define

$$z_{B_k}(\sigma) = \mathbf{1}\{\sigma \text{ is explicitly encoded in the revealed } B_k\}\,. \tag{61}$$

Under the training-complex operator construction used in our supplementary evaluation, $z_{B_k}(\sigma) = 1$ is a sufficient witness that σ was materialized into the training-side operator space. The converse need not hold, since some member queries may not appear as explicit k-simplex columns in the released operator.

The same protocol-side cue can also be extracted from raw query-aligned Hodge Laplacians. In particular, the upper Laplacian on boundary faces,

$$L_{k-1,u} = B_k B_k^\top, \tag{62}$$

preserves the coface support of B_k. If the rows and columns are labeled by $(k-1)$-simplices, then for $\partial\sigma = \{\alpha_0, \ldots, \alpha_k\}$ we can define

$$z_{L_{k-1,u}}(\sigma) = \mathbf{1}\{\exists\, \alpha_a \neq \alpha_b \in \partial\sigma : |(L_{k-1,u})_{\alpha_a,\alpha_b}| > 0\}\,. \tag{63}$$

For a fixed queried σ, any two distinct boundary faces determine the same k-simplex, so this cue is support-equivalent to $z_{B_k}(\sigma)$ under raw query-aligned release. Normalization changes the values but not this nonzero support.

Other Laplacian releases can also leak, but they are not always equivalent to B_k as value-based cues. If a k-simplex-indexed $L_{k,\ell} = B_k^\top B_k$ or $L_k = L_{k,\ell} + L_{k,u}$ is released with row/column labels, then the presence of the row/column for σ (or a positive diagonal in an ambient embedding) directly reveals that σ was materialized. By contrast, $L_{k,u} = B_{k+1} B_{k+1}^\top$ only captures super-coface relations and may be zero for a materialized k-simplex with no $(k+1)$-cofaces, although the labeled index set itself can still leak. When only the summed lower-order operator $L_{k-1} = L_{k-1,\ell} + L_{k-1,u}$ is released, the upper support cue may be mixed with lower-adjacency terms, so it remains a protocol-side signal but is not strictly equivalent to exposing B_k.

We treat these cues first as standalone protocol-side attack signals: when the operator is released, membership is inferred directly from revealed operator information before any model output is queried. Because the signal is one-sided, this standalone attack is strongest when the released operator materializes more training-aligned simplices. We additionally study a protocol-assisted variant that appends this cue to a standard attack feature vector:

$$\phi_\sigma^{\mathrm{PSA}} = \left[\phi_\sigma^{\mathrm{base}},\, z_{B_k}(\sigma)\right]. \tag{64}$$

In Appendix E.1, we evaluate this cue both as an operator-only attack signal and as an auxiliary feature for standard attacks. We therefore treat PSA as a supplementary protocol-side attack, not part of the main black-box/white-box benchmark.

C.2 Structure-Aware Black-Box (SA-BB) MIA

SA-BB uses the six response features in Eq. (7). The target interface returns a positive-class probability for each candidate simplex. Public candidates and their boundary faces define the neighbourhoods before the target membership split. Queries retain the target's trained parameters and inference topology.

Neighbour selection and features. A face-to-candidate index retrieves candidates sharing a complete $(k-1)$-face with the query. A fixed public ordering selects up to B neighbours, excluding the query itself. For an empty neighbourhood, fixed-seed sampling selects other candidates from the public pool. The selected responses determine the mean, standard deviation, and maximum absolute probability gap, together with the query's relative confidence rank. We use the population standard deviation and count ties through $p_\tau \leq p_\sigma$, as in Section 4.3. For numerical entropy evaluation, $0 \log 0$ is defined as zero.

Algorithm 1: Structure-Aware Black-Box (SA-BB) MIA

Input: Target prediction API f_θ; public candidate pool $\mathcal{C}_k$ and boundary faces; auxiliary data; neighbour budget B.
Output: Membership scores for target candidates.
Train shadow SCNs on independent auxiliary training splits
foreach *shadow model* f_j **do**
  foreach *shadow member or held-out candidate* σ **do**
    Select up to B public face-sharing neighbours; use fixed-seed public candidates if the neighbourhood is empty
    Query f_j on σ and its selected neighbours
    Compute $\phi_j^{\mathrm{SA-BB}}(\sigma)$ using Eq. (7)
    Record the feature vector and shadow membership label
Fit feature standardization and logistic regression on the shadow attack-training partition
foreach *target candidate* σ **do**
  Select neighbours with the same public rule and query f_θ
  Compute and standardize $\phi^{\mathrm{SA-BB}}(\sigma)$
  Return the logistic-regression membership score

Attack training and shadow transfer. SA-BB assumes task-labelled auxiliary data from the target domain and trains independent shadow models. The attacker labels each shadow's training and held-out queries as members and non-members, respectively. These examples form a balanced 1:1 dataset with a stratified 8:1:1 attack-training/validation/test split. Feature standardization and logistic-regression fitting use the attack-training partition. The fixed pipeline scores target queries; target membership labels are used only to compute AUC and low-FPR TPR.

Matched controls. The controlled comparison in Appendix E.5 trains three independent shadow SCCNNs using the protocol above. NN, Loss, and SA-BB use the same logistic-regression classifier in this comparison: NN uses the query probability, and Loss uses the task loss with the known label in the positive-simplex protocol. Each method receives a budget of 21 API calls per query; SA-BB uses the query and $B = 20$ neighbours.

Query cost. A candidate with n_σ selected neighbours requires $1 + n_\sigma \leq B + 1$ prediction calls. Given these responses, computing the six features takes $\mathcal{O}(B)$ time. Public neighbour lists can be indexed once and reused across shadow and target inference. The same procedure applies to a graph backbone exposing a simplex-prediction API.

C.3 Dynamic Topological Pruning and Injection (DTPI)

C.3.1 Implementation note

Construction of Edge Sets ($\mathcal{E}_{\mathrm{obs}}$ and $\mathcal{E}_{\mathrm{bg}}$). In our implementation, we distinguish an *ambient operator space* from the *training complex*. The ambient operator space determines the candidate edge universe $\mathcal{E}_{all}$ and is built from the pre-defined boundary operators. Within this space, we partition the total set of candidate edges into two disjoint sets:

- **Raw Observed Edges ($\mathcal{E}_{\mathrm{obs}}^{\mathrm{raw}}$):** Defined as the set of edges present in the training complex (i.e., the 1-skeleton induced by the training data). These edges encode training-instance-specific structural patterns and are the target of *pruning*.
- **Raw Background Candidate Edges ($\mathcal{E}_{\mathrm{bg}}^{\mathrm{raw}}$):** Defined as the complement set of candidate edges that are absent from the training complex, i.e., $\mathcal{E}_{\mathrm{bg}}^{\mathrm{raw}} := \mathcal{E}_{all} \setminus \mathcal{E}_{\mathrm{obs}}^{\mathrm{raw}}$. These edges form the candidate pool for *injection*. In our experiments, we use all available non-training edges within the pre-defined boundary operators as potential topological decoys.

Cascade-Size Guidance and Lexicographic Filtering. For each candidate edge e, DTPI pre-computes a topology-only guidance score $T(e)$ on the ambient operator space, where $T(e)$ is defined

in Appendix B.2 as the total number of simplices affected by operating on e. We then normalize it to $T(e) \in [0, 1]$ and use it to guide both modules of DTPI. In the 2-simplex setting, this quantity reduces to

$$T(e) = 1 + d_{\triangle}(e), \tag{65}$$

where $d_{\triangle}(e)$ is the number of triangle cofaces incident to edge e.

However, DTPI does not operate on raw $T(e)$ rankings alone. For both $\mathcal{E}_{\text{obs}}^{\text{raw}}$ and $\mathcal{E}_{\text{bg}}^{\text{raw}}$, we first sort edges descendingly by $T(e)$ to form a high-leverage pool. Inside that pool, we re-rank edges ascendingly by the structural-impact score

$$s_{\text{imp}}(e = (u, v)) = \frac{1}{\sqrt{d(u) + \epsilon}} + \frac{1}{\sqrt{d(v) + \epsilon}}, \tag{66}$$

which captures how strongly editing e perturbs normalized message passing. This lexicographic rule favors high-cascade edges that are simultaneously safer for utility preservation. The retained edges constitute the active pruning and injection sets $\mathcal{E}_{\text{obs}}$ and $\mathcal{E}_{\text{bg}}$ used by the learnable mask module below.

Mask Parameterization and Differentiable Sampling. We assign a learnable logit parameter w_e only to edges in the active pruning and injection sets $\mathcal{E}_{\text{obs}} \cup \mathcal{E}_{\text{bg}}$, and define the *continuous* edge mask as

$$m_e := \text{sigmoid}(w_e) \in (0, 1). \tag{67}$$

During training, to approximate the binary nature of edge presence in the forward pass while keeping the optimization differentiable, we sample a *discrete* edge mask $\hat{m}_e \in \{0, 1\}$ via the Gumbel-Softmax (Binary-Concrete) reparameterization Jang et al. (2017) with temperature τ (default $\tau = 1.0$). We emphasize that **the forward propagation uses the sampled masks** $\hat{m}_e$, while **the DTPI regularization terms are defined on the continuous masks** m_e inside the overall objective in Eq. (10), which enables stable optimization of the mask parameters.

Dynamic Laplacian Reconstruction. A key aspect of DTPI is the dynamic reconstruction of Hodge Laplacians at each training step. The mathematical form remains the standard *normalized* Hodge Laplacian; the difference is that the boundary operators are *masked* by the sampled edge/triangle masks, allowing gradients to flow from the task loss back to the mask parameters.

Let $\hat{M}_1 = \text{diag}(\hat{m}_e)$ be the sampled edge mask matrix. For each triangle (2-simplex) σ, we tie its mask to its boundary edges by the product rule, i.e.,

$$\hat{m}_\sigma = \prod_{e \in \partial\sigma} \hat{m}_e, \qquad \hat{M}_2 = \text{diag}(\hat{m}_\sigma). \tag{68}$$

The effective boundary matrices are updated as

$$\tilde{B}_1 = B_1 \hat{M}_1^{1/2}, \qquad \tilde{B}_2 = \hat{M}_1^{1/2} B_2 \hat{M}_2^{1/2}. \tag{69}$$

Consequently, the degree matrices $\tilde{D}_1$ and $\tilde{D}_2$ are re-computed from $\tilde{B}_1$ and $\tilde{B}_2$. We use the standard absolute-incidence degree definition:

$$(\tilde{D}_1)_{vv} = \sum_e \left|\tilde{B}_1(v, e)\right| + \epsilon, \qquad (\tilde{D}_2)_{\sigma\sigma} = \sum_e \left|\tilde{B}_2(e, \sigma)\right| + \epsilon, \tag{70}$$

where $\epsilon > 0$ is a small constant for numerical stability. The dynamic Laplacians are then computed as

$$\tilde{L}_{1,\ell} = \tilde{B}_1^\top \tilde{D}_1^{-1} \tilde{B}_1, \qquad \tilde{L}_{1,u} = \tilde{B}_2 \tilde{D}_2^{-1} \tilde{B}_2^\top, \tag{71}$$

and passed to the SCN layers for message passing.

C.3.2 Hyperparameter Sensitivity and Loss Balancing

To ensure reproducibility, we detail the optimization and defense configurations used in our experiments. All backbone models (including `SCNN`, `PSNN`, and `SCCNN`) are trained with Adam at learning rate 10^{-3} and hidden dimension 32. We use simplex order $K = 2$ for the main results; additional experiments with $K = 3$ are reported in the appendix. For polynomial-based SCNs, we

use Chebyshev filter order $P = 2$. Training uses either full-batch updates or subgraph sampling, depending on the backbone, for at most 1000 epochs with early stopping based on validation AUC.

An important part of DTPI is balancing task utility against the cascade-weighted regularization terms in the total objective. Writing $\mathcal{D}$ for the empirical training distribution over query–label pairs and q_w for the mask-sampling distribution induced by the logits w, we use the full DTPI objective

$$\min_{\theta,w} \quad \mathbb{E}_{(\sigma,y)\sim\mathcal{D}}\mathbb{E}_{\hat{\mathcal{M}}\sim q_w}\left[\ell\Big(f_\theta(S\odot\hat{\mathcal{M}},\sigma),y\Big)\right] \quad +\lambda_1\sum_{e\in\mathcal{E}_{\text{obs}}} m_e\bar{T}(e)+\lambda_2\sum_{e\in\mathcal{E}_{\text{bg}}}(1-m_e)\bar{T}(e).$$

In implementation, we estimate the expectation term with a minibatch-average, single-mask-draw stochastic estimator at each training step. The lexicographic screening stage removes the need for a scalarized privacy–utility score when constructing candidate sets, so λ_1 and λ_2 only control the optimization strength after the structurally safe candidates have been fixed. Empirically, a 1:1 ratio between pruning and injection is suboptimal because overly aggressive edge activation can overwhelm the semantic topology needed for the main task. We therefore set the pruning weight to $\lambda_1 = 10^{-2}$, which is strong enough to suppress high-$T(e)$ structural shortcuts. We set the injection weight to $\lambda_2 = 10^{-3}$, one order of magnitude lower. This lower weight is enough to introduce high-cascade decoys without excessively densifying the simplicial structure or degrading predictive utility.

The absolute magnitudes of these coefficients govern the overall privacy–utility trade-off: scaling λ_1 and λ_2 proportionally while maintaining the $\approx 10{:}1$ ratio tunes the defense strength. Higher values provide stronger privacy at the cost of utility, while lower values prioritize model performance.

C.3.3 Complexity Analysis

A common concern with spectral graph/simplicial methods is the theoretical $\mathcal{O}(|V|^3)$ cost associated with eigendecomposition. We clarify here that our implementation avoids this bottleneck, ensuring scalability.

While classical spectral methods require full diagonalization of the Laplacian L, modern SCNs employ **Chebyshev polynomial approximations** Madan & Seneta (1987). Following the open-source implementations of previous works Defferrard et al. (2017); Yang & Isufi (2023), we adopt this technique to avoid explicit eigendecomposition. Instead, the convolution is computed via recursive sparse matrix-vector multiplications. For a filter of order P, the cost is $\mathcal{O}(P\cdot(|\mathcal{E}|+|\mathcal{T}|)\cdot F)$, which is linear in the number of simplices rather than cubic in the number of nodes.

DTPI Overhead. The computational overhead of DTPI mainly arises from the additional operator update and mask optimization performed during training. In particular, each step updates the topological masks, rebuilds the masked operators used by the backbone, and optimizes the corresponding auxiliary regularizers. The $T(e)$ scores are pre-computed once from the boundary operators and therefore add negligible cost compared with the per-epoch operator update and network training.

Table 4: Mean wall-clock time (s) per 10 training epochs.

Dataset	Early-Stop	GEP	SHAP	KD	Adv-Reg	DP-SGD	DTPI
Cora	0.350	0.350	0.372	0.517	0.527	169.227	2.540
CiteSeer	0.320	0.320	0.326	0.431	0.494	87.015	1.742

A100 training cost. Table 4 reports mean wall-clock time per 10 training epochs on an A100 GPU. DTPI takes 2.540 s on Cora and 1.742 s on CiteSeer, compared with 169.227 s and 87.015 s for DP-SGD. Reconstructing masked Hodge operators raises DTPI's cost above the lightweight baselines. Table 2 examines how less frequent reconstruction changes this cost and the privacy–utility trade-off.

Empirical Runtime and Resources. Our benchmark-scale training also fits on a single NVIDIA GeForce RTX 4090 GPU (24 GB memory). In addition to the A100 comparison in Table 4, we measure the average *training-loop* time per epoch on this single-GPU setup, excluding attack evaluation and other post-training analysis. For SCCNN+DTPI, the average steady-state epoch time is 0.026s on Semantic (352 nodes, 1,474 edges, 3,285 triangles), 0.534s on Cora (2,708 nodes, 5,278 edges, 1,630 triangles), and 0.331s on CiteSeer (3,327 nodes, 4,552 edges, 1,167 triangles). These measurements suggest that DTPI remains sub-second per epoch on the tested benchmarks,

with a modest wall-clock cost on the smallest graph and a more visible constant-factor increase on larger ones.

C.3.4 DTPI TRAINING ALGORITHM

The complete training procedure, which explicitly links the topological masks to the task optimization, is outlined in Algorithm 2.

Algorithm 2: DTPI Joint Optimization Strategy

Input: Initial Complex S (boundaries B_1, B_2), training query–label set $\mathcal{D}$, pre-computed normalized cascade scores $\bar{T}(e)$, epochs E, hyperparameters λ_1, λ_2.
Output: Robust SCN model f_θ, Optimized mask operator $\mathcal{M}$.
`/* Initialization */`
Partition edges into raw sets $\mathcal{E}_{\text{obs}}^{\text{raw}}$ (observed/train) and $\mathcal{E}_{\text{bg}}^{\text{raw}}$ (background), then retain active sets $\mathcal{E}_{\text{obs}}$ and $\mathcal{E}_{\text{bg}}$ after lexicographic filtering;
Rank each set by $T(e)$, then re-rank the high-leverage pool by structural impact $1/\sqrt{d(u)+\epsilon} + 1/\sqrt{d(v)+\epsilon}$;
Retain the active pruning/injection sets and initialize mask logits w_e only on those edges;
for *epoch* $t = 1, \ldots, E$ **do**
  `/* 1. Mask Generation (Gumbel-Softmax) */`
  Sample discrete edge masks: $\hat{m}_e \sim \text{Gumbel}(w_e)$;
  Compute continuous masks for regularization: $m_e \leftarrow \text{sigmoid}(w_e)$;
  Compute triangle masks (Simplicial Consistency): $\hat{m}_\sigma \leftarrow \prod_{e\in\sigma} \hat{m}_e$;
  `/* 2. Dynamic Structure Update (` $S \odot \mathcal{M}$ `) */`
  Update weighted boundaries: $\tilde{B}_1 \leftarrow B_1 \text{diag}(\hat{m}_e)^{1/2}, \tilde{B}_2 \leftarrow \text{diag}(\hat{m}_e)^{1/2} B_2 \text{diag}(\hat{m}_\sigma)^{1/2}$;
  Reconstruct dynamic Laplacians: $\tilde{L}_{1,\ell} \leftarrow \tilde{B}_1^\top \tilde{D}_1^{-1} \tilde{B}_1, \tilde{L}_{1,u} \leftarrow \tilde{B}_2 \tilde{D}_2^{-1} \tilde{B}_2^\top$;
  `/* 3. Forward with Masked Topology */`
  Sample a minibatch $\mathcal{B}_t \subset \mathcal{D}$;
  Compute embeddings and logits on the masked complex $S \odot \hat{\mathcal{M}}$;
  Compute minibatch task-loss estimator: $\widehat{\mathcal{L}}_{\text{task}} \leftarrow \frac{1}{|\mathcal{B}_t|} \sum_{(\sigma,y)\in\mathcal{B}_t} \ell(f_\theta(S \odot \hat{\mathcal{M}}, \sigma), y)$;
  `/* 4. Cascade-Guided Objective */`
  `/* 5. Optimization Step */`
  $\widehat{\mathcal{L}}_{total} \leftarrow \widehat{\mathcal{L}}_{\text{task}} + \lambda_1 \sum_{e\in\mathcal{E}_{\text{obs}}} m_e \bar{T}(e) + \lambda_2 \sum_{e\in\mathcal{E}_{\text{bg}}} (1-m_e)\bar{T}(e)$;
  Update θ and w_e via Adam optimizer;
end
return $f_\theta, \mathcal{M}$

D BASELINES, DATASETS, AND ADDITIONAL EXPERIMENTS

D.1 DATASETS

We use three benchmark datasets to evaluate SCNs and our privacy defense.

SEMANTIC. Derived from the Semantic Scholar database Ammar et al. (2018), this dataset represents a co-authorship network where nodes correspond to authors and edges indicate collaboration. The dataset contains 352 nodes and 1,474 edges (average degree 8.375). Following prior work, for 2-simplex prediction it provides 1,482 positive triangles and 1,803 negative triplets.

CORA. A standard citation network benchmark from the Planetoid suite Yang et al. (2016). Nodes represent machine learning papers and edges represent citation links; node features are 1,433-dimensional bag-of-words vectors. We treat the 1-skeleton as undirected for candidate construction and enumerate candidate 2-simplices as triangles (3-cliques) in the graph. The resulting candidate triangle set contains 2,437 instances.

CITESEER. Another Planetoid citation network Yang et al. (2016) with 3,327 nodes and 4,552 edges; nodes have 3,703-dimensional bag-of-words features. We construct candidate 2-simplices in the

same way as CORA, yielding 1,590 candidate triangles. Compared to CORA, CITESEER is sparser (average degree 2.736), enabling evaluation under different sparsity regimes.

Simplex prediction protocol. We study k-simplex prediction ($k \in \{2, 3\}$), where each queried k-simplex instance σ (a triangle for k=2 and a tetrahedron for k=3) is assigned a binary label and models predict whether σ forms a positive k-simplex under a dataset-specific criterion. Unless otherwise specified, we split positive and negative instances separately into train/val/test with an 8:1:1 ratio and apply the same training pipeline across backbones.

Labeling. On SEMANTIC (following prior work Yang & Isufi (2023)), we define positive 2-simplices as triangles with high collaboration impact, $\mathcal{T}_P = \{t \mid [x_2]_t > 7\}$, and define negatives as triplets with low impact, $\mathcal{T}_N = \{t \mid [x_2]_t \leq 7\}$. For the supplementary 3-simplex (tetrahedron) prediction, we construct balanced positive/negative 3-simplices using the higher-order signal x_3 as in prior work. On CORA and CITESEER, we label a triplet $t = \{u, v, w\}$ as positive if the three nodes share the same class label. We define negatives to include triplets whose nodes are not all from the same class, as well as triplets that do not satisfy the positive condition.

Operator construction. When constructing the simplicial operators used by SCN backbones (e.g., B_k and the corresponding Hodge Laplacians), we use the union of positive and negative *training* query instances only, excluding validation/test candidates, following standard simplex-prediction practice. This query set determines the operator-construction pool, but only materialized simplices appear explicitly as boundary-matrix columns. Validation/test queries are reserved for supervision, early stopping, and MIA evaluation and do not provide auxiliary protocol-side operator cues to the main benchmark attacks. For graph-only baselines, the same query split is used, but simplex logits are produced from boundary node/edge embeddings without higher-order Hodge operators. For graph-only baselines (e.g., `GNN`), simplex logits are obtained by applying the same simplex readout head p_ψ to the embeddings of each queried k-simplex (constructed from its boundary node/edge embeddings).

D.2 DETAILED BACKBONES AND THREAT MODELS

Backbones. We compare six backbones that span graph baselines and SCNs with increasing expressive power. The graph baseline `GNN` Defferrard et al. (2017) is a node-level polynomial graph convolution that operates only on the 1-skeleton and does not use simplicial couplings. `SNN` Ebli et al. (2020) is a simplicial neural network baseline that performs within-order propagation without explicit lower and upper Hodge Laplacian factorization. `SCNN` Yang et al. (2022a) applies separate lower and upper Hodge Laplacian filtering within a fixed simplex order. `PSNN` Glaze et al. (2021) is the one-step special case of this family with filter order one. `Bunch` Bunch et al. (2020) introduces cross-order interaction via boundary-based projections while keeping one-step shifting style operators. `SCCNN` Yang & Isufi (2023); Yang et al. (2025) further generalizes these designs and propagates information across adjacent simplex orders with extended inter-simplicial locality.

Attacks Baselines. We evaluate three **black-box** attacks: NN Salem et al. (2018), which trains an attacker on prediction outputs; LOSS Sablayrolles et al. (2019), which scores membership from the task loss; and SA-BB, which uses local response features from public candidate queries. The three **white-box** baselines are GRADIENT Rezaei & Liu (2020), based on loss gradients with respect to input features; SIA Zhang et al. (2021), based on subgraph edge embeddings; and SMIA Wang & Wang (2024), based on similarity rankings of final-layer node embeddings. These attacks assume access to the trained model and its internal activations or gradients, the query's task label, and an alignment between the queried simplex and the model's internal representation. Such access is generally unavailable through a standard prediction API and is difficult to obtain in deployed systems. We therefore use white-box attacks as high-access security audits, rather than as the default deployment threat model. Appendix C.2 gives the SA-BB implementation and the matched shadow-transfer controls.

D.3 DEFENSE BASELINES AND HYPERPARAMETERS

We compare DTPI against a diverse set of privacy defenses. For each baseline, we specify the implementation details and the hyperparameter search space used to trace the privacy–utility trade-off in our experiments.

ADV-REG (Adversarial Regularization) Ding et al. (2024). This method introduces a min-max adversarial game where a differentiable discriminator is trained to distinguish between member and non-member samples based on their gradient or spectral features, while the encoder is simultaneously trained to fool it. The defense strength is controlled by the adversarial loss weight λ (implemented as `adv_lambda_max`). To capture the full trade-off curve, we sweep λ using 10 evenly spaced values in the range $[0, 500]$. A larger λ enforces stronger indistinguishability (privacy) but imposes stricter constraints on the task performance.

GEP (Graph Embedding Perturbation) Zhang et al. (2021). We implement embedding perturbation by injecting Laplace noise into the learned edge embeddings. The noise is applied with a relative scale parameter `noise_b` (clipped to $[0, 1]$), where the noise scale is proportional to the magnitude of the embedding dimensions:

$$\tilde{h} = h + \text{Lap}(\texttt{noise_b} \cdot |h|) \tag{72}$$

In our experiments, we fix the perturbation ratio to `perturb_r`=1.0 (perturbing all dimensions) and sweep the noise scale `noise_b` across 10 evenly spaced values in $[0, 1]$.

SHAP (SHAP-based Noise Injection) Wang & Wang (2024). This method uses a gradient-based importance proxy (inspired by SHAP values) to identify low-importance embedding dimensions and selectively adds Laplace noise to them. The trade-off is governed by the noise scale `noise_b` and the perturbation ratio `perturb_r` (the fraction of dimensions receiving noise). In our parameter sweep, we fix the perturbation ratio at `perturb_r`=0.8 and vary the noise scale `noise_b` over 10 evenly spaced values in $[0, 1]$.

DP-SGD (Differentially Private SGD) Abadi et al. (2016). We include a DP-SGD-style baseline to compare against a widely used gradient-clipping and Gaussian-noise defense. At each optimization step, we clip the aggregate gradient norm by C and add Gaussian noise with multiplier σ to the clipped gradients before the parameter update. Because a single edge or simplex can affect multiple messages and gradients through the SCN propagation operators, standard per-example DP-SGD accounting does not directly certify edge-level differential privacy for our setting without an additional structural sensitivity bound. We therefore treat this baseline as an empirical Gaussian-noise defense rather than as a formal Edge-DP mechanism. For reference only, we also report the nominal full-batch Gaussian RDP values under add/remove adjacency with $\delta = 1/N$, where N is the number of training samples.

The privacy–utility trade-off is controlled by the noise multiplier σ. To avoid over-specializing the baseline to a single operating point, we tune the clipping bound C on the validation split over a small grid and sweep $\sigma \in \{10^{-4}, 0.005, 0.02, 0.1, 0.5, 5, 10, 20, 50, 100, 200\}$. For reference, the corresponding ϵ values in SEMANTIC dataset for the above σ values are around ∞, 2.2×10^7, 1.4×10^6, 5.5×10^4, 2277.5, 45.0, 17.5, 7.5, 2.5, 1.5, and 0.5, respectively.

KD (Knowledge Distillation) Zheng & Peng (2022). The student model f_S is trained to match the teacher f_T's soft outputs using a temperature T, alongside the standard supervised loss:

$$\mathcal{L}_{student} = (1-\alpha)\mathcal{L}_{CE}(y, f_S(x)) + \alpha T^2 \mathcal{L}_{KL}\left(\sigma\left(\frac{f_T(x)}{T}\right), \sigma\left(\frac{f_S(x)}{T}\right)\right) \tag{73}$$

We fix the temperature at $T = 2.0$ and sweep the distillation weight α over the set $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. The teacher model is pretrained for 300 epochs. Additionally, the student training includes mild structural/logit noise injection that scales with α to further reduce overfitting.

EARLY-STOP. We use validation-based early stopping and select the checkpoint with the best validation AUC while limiting the maximum training duration. We sweep the maximum training budget over $\{10, 20, 40, 50, 60, 80, 100, 400, 800, 1000\}$ epochs. For budgets below 100 epochs, we evaluate the final checkpoint at the specified budget. For larger budgets, we evaluate checkpoints every 50 epochs and select the checkpoint with the best validation AUC within the allowed budget. This lets us trace the transition from generalization to memorization under different training budgets.

DTPI (Ours). As proposed in Section 5, DTPI combines a fixed lexicographic candidate-screening stage with two optimization weights: λ_1 for pruning and λ_2 for injection. We sweep these weights proportionally to navigate the Pareto frontier, using values for λ_1 in $\{10^{-5}, 10^{-4}, 10^{-3}, 5\cdot10^{-3}, 10^{-2}, 5\cdot10^{-2}, 10^{-1}, 0.5, 1.0\}$ while maintaining $\lambda_2 \approx 0.1\lambda_1$.

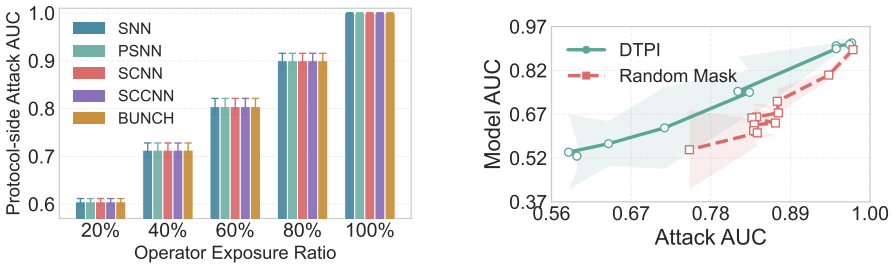


Figure 10: **Standalone protocol-side attack under operator release.** Left: operator-only membership inference AUC versus *operator exposure ratio*, the fraction of positive/negative training queries sampled into released-operator construction. Right: privacy–utility trade-off under the same operator-only threat model; each DTPI point corresponds to a different privacy intensity, and the random baseline is evaluated at matched intensities.

E Additional Results and Analyses

We report protocol-side leakage, controlled structural ablations, SA-BB shadow transfer, low-FPR operating points, and membership leakage over training.

E.1 Protocol-Side Structural Leakage

This subsection studies the protocol-side structural attacker defined in Appendix C.1. We first evaluate it as a standalone operator-only attack under operator release, then report a protocol-assisted variant in which the same cue augments a standard attack.

We first clarify the operator exposure ratio used in the main text. This ratio should be understood as the amount of training-complex-induced operator information that is both exposed by the protocol and available to the model for operator-guided message passing. At exposure ratio zero, no operator information derived from private training queries is revealed or used, so the adversary has no direct protocol-side inclusion cue and the model cannot obtain utility gains from training-aligned higher-order propagation. At exposure ratio one, the model has access to the operators induced by the training complex, such as boundary matrices or Hodge Laplacians, which maximizes the available operator-guided structural signal but also exposes construction-dependent membership information. Intermediate ratios interpolate between these endpoints by revealing only part of the training-aligned operator structure. Importantly, this notion is tied to model utility: operators that are sparsely exposed, heavily masked, or constructed independently of the training data may reduce protocol-side leakage, but they also provide little or no task-relevant message-passing benefit.

Standalone protocol-side attack. Figure 10 shows that protocol-side leakage is controlled primarily by how much private training structure is materialized into the released operator. In this operator-only setting, as the operator exposure ratio increases from 20% to 100%, the attack AUC rises monotonically from near 0.60 to almost perfect leakage. This supports the same exposure-based intuition as Figure 1(b): more operator exposure yields stronger membership cues even before any model output is observed.

Laplacian-based protocol cues. Figure 11 evaluates whether the same operator-only leakage can be obtained from Hodge Laplacians rather than from B_k directly. The upper boundary-face Laplacian $L_{k-1,u} = B_k B_k^\top$ gives exactly the same AUC as the explicit B_k cue at every exposure ratio on both datasets, because a nonzero off-diagonal entry between two boundary $(k-1)$-faces is equivalent to materializing their unique k-coface. In contrast, $L_{k-1,\ell}$ provides only a weaker indirect cue through lower-adjacency and normalization effects. The summed operator L_{k-1} remains attackable, but it is not strictly equivalent to B_k because lower and upper terms are mixed in the observed values.

DTPI against the operator-only threat. The right panel of Figure 10 shows that DTPI remains effective even against this operator-only threat model. Because this leakage channel comes from the released operator itself, defenses that only modify model training or post hoc attacker inputs do not

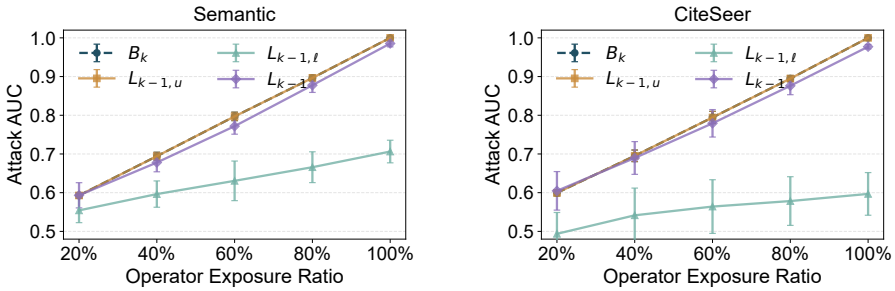


Figure 11: **Protocol-side cue comparison under raw query-aligned operator release.** We compare the direct B_k cue against cues extracted from $L_{k-1,u}$, $L_{k-1,\ell}$, and their sum L_{k-1} on SEMANTIC (left) and CITESEER (right), averaged over 10 seeds. The B_k and $L_{k-1,u}$ curves coincide on both datasets, confirming their support-equivalence for detecting materialized queried k-simplices.

Table 5: **B_k-assisted calibration of representative attacks.** We evaluate a single SCNN backbone on SEMANTIC, CORA, and CITESEER over 5 seeds. For each base attack, we train the same logistic-regression attacker on the original feature vector and on the augmented feature vector $[\phi_\sigma^{\text{base}}, z_{B_k}(\sigma)]$, where the auxiliary signal $z_{B_k}(\sigma)$ is computed from the operator released at **40% operator exposure**. Values are mean AUCs; Δ denotes the gain from the auxiliary B_k cue.

Attack	SEMANTIC			CORA			CITESEER		
	Base	$+B_k$	Δ	Base	$+B_k$	Δ	Base	$+B_k$	Δ
NN	0.598	0.768	+0.171	0.832	0.894	+0.062	0.752	0.861	+0.110
LOSS	0.597	0.769	+0.171	0.832	0.896	+0.064	0.752	0.864	+0.112
GRADIENT	0.633	0.767	+0.135	0.883	0.927	+0.044	0.827	0.907	+0.080
SIA	0.574	0.744	+0.170	0.797	0.845	+0.049	0.750	0.804	+0.054
SMIA	0.572	0.740	+0.168	0.622	0.779	+0.157	0.587	0.784	+0.197

directly perturb the protocol-side signal. We therefore use random topology editing as the baseline and compare it against DTPI at matched privacy intensities. DTPI traces a better privacy–utility frontier than this random baseline, which indicates that its benefit does not come from indiscriminately degrading the operator. Instead, it edits topology in a targeted way that weakens protocol-side leakage while preserving more utility.

Protocol-assisted use of the same cue. The same protocol-side operator information can also be used as auxiliary information on top of a standard attack. Table 5 confirms that appending $z_{B_k}(\sigma)$ improves every representative black-box and white-box attack across all three datasets. The gains are most pronounced for weaker similarity-based attacks such as SMIA (e.g., +0.157 on CORA and +0.197 on CITESEER).

Separation from the main benchmark. The main-text experiments never provide this protocol-side operator information to the attacker. This design choice is important because it lets us evaluate protocol-side exposure and model-side leakage independently, rather than allowing one privacy source to amplify or mask the other. We therefore treat Table 5 as a supplementary stress test under revealed operator information, not as part of the main benchmark.

Deployment interpretation. Taken together, these supplementary results show that released operators can support standalone membership inference and strengthen standard attacks. The main benchmark therefore withholds direct operator-release cues and evaluates the remaining risk through model-side outputs, gradients, and representations.

E.2 FULL/ORDER-1/GRAPH-ONLY ABLATION

To probe how much of the observed triangle-membership leakage can already be explained without explicit higher-order message passing, we repeat a representative ablation on SEMANTIC, CORA,

Table 6: **Controlled Full/Order-1/Graph-only ablation.** We evaluate triangle membership inference on SCCNN for 20 epochs and report mean AUC over 5 seeds. **Order-1** removes upper/coface message passing but retains the original order-2-derived operators, while **Graph-only** rebuilds the operators from B_1 only.

View	Model AUC			NN AUC			Raw-Emb. AUC		
	SEM.	CORA	CITE.	SEM.	CORA	CITE.	SEM.	CORA	CITE.
Full	0.790	0.683	0.771	0.590	0.766	0.647	0.643	0.866	0.799
Order-1	0.677	0.536	0.528	0.507	0.527	0.527	0.516	0.539	0.551
Graph-only	0.550	0.456	0.574	0.490	0.494	0.490	0.500	0.514	0.496

and CITESEER using the strongest simplicial backbone, SCCNN. We compare three encoder views under the same triangle-prediction protocol: **Full**, the standard SCCNN with node–edge–triangle coupling; **Order-1**, which removes the upper/coface message-passing branch but keeps the original order-2-derived normalization operators; and **Graph-only**, a stricter lower-order view that rebuilds the edge operators from B_1 alone, using $D_1 = \mathrm{diag}(|B_1|\mathbf{1})$, $D_2 = I$, $L_{1,u} = 0$, and dummy higher-order tensors that carry no B_2 information. Table 6 reports one representative black-box attack (NN, instantiated by an attacker trained on prediction confidences) and one representative white-box attack (RAW-EMBEDDING), averaged over **5 seeds** at epoch 20. We report these non-shadow attacks because they preserve the lower-order versus higher-order contrast without the saturation behavior observed for stronger gradient-oracle attacks. Across all three datasets, removing higher-order context consistently lowers both representative attack AUCs, while utility also drops substantially relative to the full simplicial model. This suggests that lower-order boundary information alone already explains a non-trivial fraction of leakage, although explicit higher-order context still contributes materially to both utility and attack success.

E.3 LOW-FPR EVALUATION OF ATTACKS AND DEFENSES

TPR@1% FPR and TPR@5% FPR measure how many members an attack detects when the false-positive rate on non-members is 1% or 5%. Table 7 compares attacks on three datasets, and Table 8 evaluates defenses on CITESEER. These comparisons use SCCNN. The separate SA-BB shadow-transfer evaluation in Table 9 reports the same low-FPR metrics under prediction-API access.

Table 7: Attack AUC and low-FPR TPR on SCCNN. TPR columns use FPR thresholds of 1% and 5%.

Attack	SEMANTIC			CORA			CITESEER		
	AUC	TPR@1%	TPR@5%	AUC	TPR@1%	TPR@5%	AUC	TPR@1%	TPR@5%
NN	0.7089	0.0201	0.0930	0.9445	0.2141	0.7016	0.8317	0.0637	0.1439
Loss	0.6838	0.0143	0.0905	0.9371	0.2855	0.5894	0.8321	0.0495	0.1626
Gradient	0.8295	0.1106	0.2982	0.9518	0.2204	0.6515	0.9170	0.1416	0.3973
SIA	0.8448	0.2051	0.4145	0.9801	0.5155	0.8506	0.9343	0.5067	0.6754
SMIA	0.6643	0.0756	0.2389	0.9303	0.4237	0.6786	0.8640	0.4004	0.5145

Low-FPR behavior. The low-FPR comparison shows substantial variation in detection rates across attacks at the same FPR. On CITESEER, SIA detects 0.5067 of members at 1% FPR, compared with 0.0637 for NN. Across defenses, DTPI combines model AUC 0.7807 with the lowest reported TPR under NN, Gradient, SIA, and SMIA at both operating points. These operating points expose differences in the privacy–utility trade-off that a single attack AUC can obscure.

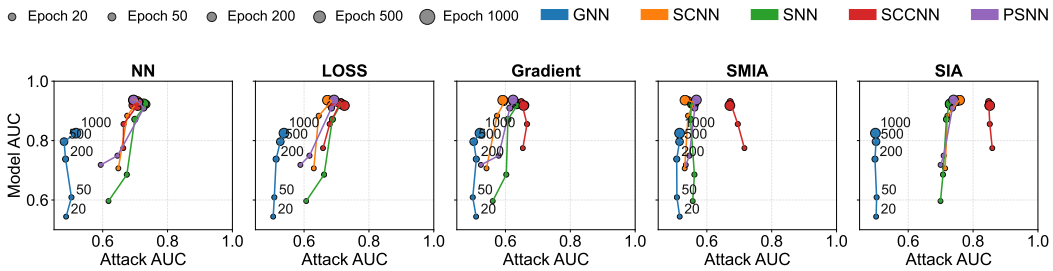
E.4 EARLY-STOPPING CONTROLS AND NON-MONOTONIC PRIVACY TRAJECTORIES

Epoch checkpoints. To evaluate MIAs under varying degrees of fitting, we probe models at fixed training checkpoints epoch $\in \{20, 50, 200, 500, 1000\}$. This is necessary because both utility and membership leakage can evolve along training and may not be well represented by a single near-converged snapshot.

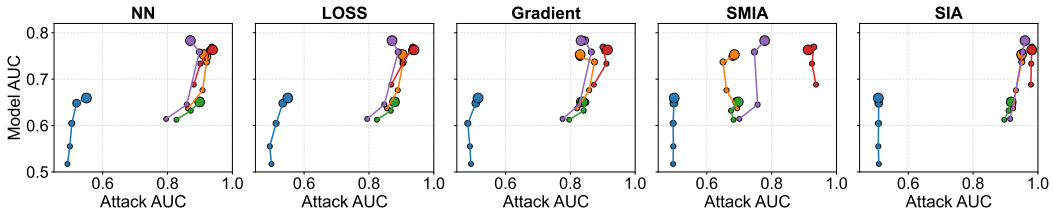
Table 8: Low-FPR defense comparison on CITESEER with SCCNN. Both panels use the same defense operating points.

Defense	Model AUC	NN		Loss		Gradient	
		TPR@1%	TPR@5%	TPR@1%	TPR@5%	TPR@1%	TPR@5%
Early-Stop	0.7291	0.0827	0.1377	0.0434	0.1175	0.2727	0.5169
Adv-Reg	0.7585	0.0973	0.1700	0.0737	0.1797	0.1949	0.5148
DP-SGD	0.6593	0.1184	0.2021	0.0648	0.1881	0.3314	0.6041
KD	0.6723	0.1475	0.2297	0.0031	0.1432	0.3156	0.5173
GEP	0.7598	0.0659	0.1323	0.0450	0.1355	0.2102	0.5097
SHAP	0.7648	0.1144	0.2063	0.0663	0.2012	0.2024	0.4833
DTPI	0.7807	0.0595	0.0926	0.0421	0.1037	0.1921	0.4277

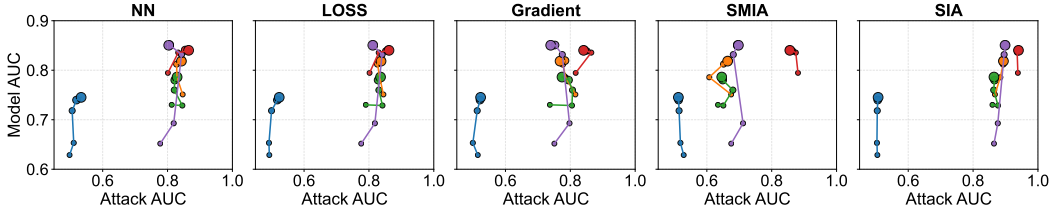
Defense	Model AUC	SIA		SMIA	
		TPR@1%	TPR@5%	TPR@1%	TPR@5%
Early-Stop	0.7291	0.4841	0.6854	0.3882	0.4840
Adv-Reg	0.7585	0.4923	0.6597	0.4196	0.4919
DP-SGD	0.6593	0.4898	0.6882	0.4029	0.5053
KD	0.6723	0.5135	0.6832	0.3853	0.5317
GEP	0.7598	0.4689	0.6760	0.4043	0.4986
SHAP	0.7648	0.4888	0.6883	0.4084	0.5100
DTPI	0.7807	0.4154	0.6536	0.3227	0.4434



(a) Privacy–utility trajectories across epochs on SEMANTIC.



(b) Privacy–utility trajectories across epochs on CORA.



(c) Privacy–utility trajectories across epochs on CITESEER.

Figure 12: Privacy–utility trajectories over training for SEMANTIC, CORA, and CITESEER. Each subfigure plots model AUC versus attack AUC across epochs for five baseline attacks; curves are colored by backbone and marker size indicates epoch.

Trajectories reveal joint effects of architecture and fitting. Figure. 12 complements Figure. 5 by plotting privacy–utility trajectories across epochs, making explicit that attack AUC depends on both (i) the backbone architecture (graph-only processing versus SCNs with richer simplicial operators/coupling) and (ii) the training stage. For example, under the NN attack, GNN moves

from $(0.487, 0.544)$ at epoch 20 to $(0.518, 0.825)$ at epoch 1000 (attack AUC, model AUC), while SC-based backbones reach higher utility with substantially higher attack success at later epochs, e.g., SNN attains $(0.727, 0.924)$ at epoch 1000 and SCCNN attains $(0.708, 0.918)$ at epoch 1000. Consistent patterns appear for LOSS and Gradient, where SNN increases from attack AUC 0.607 to 0.714 as model AUC rises from 0.596 to 0.924 (epoch 20→1000). Figure. 5 summarizes these dynamics by averaging attack/model AUC over the selected checkpoints to provide a compact view of the architecture-driven ordering.

Training budget in the main text. We set the main-text training budget to 1000 epochs because model AUC largely saturates after epoch 500 across backbones. For instance, under NN, PSNN changes only from 0.929 to 0.937 in model AUC from epoch 500 to 1000, while SNN changes from 0.922 to 0.924 and SCNN from 0.935 to 0.936; similar saturation holds under LOSS and Gradient. Using 1000 epochs therefore ensures near-converged utility without materially shifting the utility frontier relative to 500 epochs, while still allowing non-monotonic privacy behavior (e.g., early-stopping effects) to be observed and analyzed (Appendix E.6).

E.5 SA-BB Shadow Transfer and Neighbour Ablation

Independent shadow transfer. We train three independent shadow SCCNNs and transfer a logistic-regression attacker to the target, following Appendix C.2. This controlled positive-simplex evaluation gives NN, Loss, and SA-BB the same classifier, balanced attacker split, and 21-call budget. Table 9 reports attack AUC and TPR at low FPR. SA-BB reaches AUC 0.7286 on Semantic and 0.8853 on CiteSeer, exceeding both output-only controls. On CiteSeer, its TPR at 1% FPR rises to 0.1840, compared with 0.0637 for NN and 0.0495 for Loss.

Table 9: SA-BB under independent shadow transfer on SCCNN. All attacks receive 21 prediction calls per query. TPR columns give detection rates at the stated FPR.

Attack	Semantic			CiteSeer		
	AUC	TPR@1%	TPR@5%	AUC	TPR@1%	TPR@5%
NN	0.7089	0.0201	0.0930	0.8317	0.0637	0.1439
Loss	0.6838	0.0143	0.0905	0.8321	0.0495	0.1626
SA-BB	**0.7286**	**0.0214**	**0.1329**	**0.8853**	**0.1840**	**0.3770**

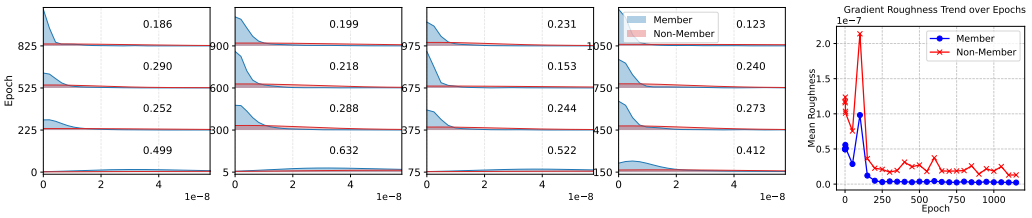
Neighbour selection and query budget. Table 10 compares three public neighbour-selection rules. **Topology** selects candidates sharing a complete boundary face with the query. **Random** samples candidates with no shared face. **Matched** applies the same no-shared-face restriction and matches local degree statistics. Each setting uses $B + 1$ prediction calls. On Semantic, increasing B from 1 to 20 raises topology-based AUC from 0.6954 to 0.7286. Topology outperforms random sampling on both datasets. The matched control performs best on CiteSeer, reaching 0.8972 at $B = 20$. These results show that local degree structure also carries useful membership signals.

Table 10: SA-BB attack AUC across neighbour-selection rules and budgets. B counts neighbour queries; each attack also queries the candidate itself.

B	Semantic			CiteSeer		
	Topology	Matched	Random	Topology	Matched	Random
1	**0.6954**	0.6876	0.6885	0.8537	**0.8590**	0.8366
5	**0.7021**	0.6926	0.6961	0.8762	**0.8817**	0.8479
10	**0.7171**	0.7013	0.7020	0.8822	**0.8943**	0.8449
20	**0.7286**	0.7023	0.6996	0.8853	**0.8972**	0.8490

E.6 Biphasic Leakage Dynamics: Inductive Bias vs. Overfitting

Membership leakage can vary non-monotonically over training, as illustrated by the baseline attack trajectories in Appendix E.4. Figure 13 examines gradient roughness as a diagnostic of this behavior.



(a) Distribution Evolution. (b) Roughness Trend.

Figure 13: (a) The ridgeline plot visualizes the evolution of gradient roughness distributions. The numerical value annotated in the top-right corner of each subplot represents the *Overlap Coefficient* ($\int \min(p, q)dx$) between member and non-member distributions; a lower overlap indicates higher distinguishability and stronger leakage. (b) The mean roughness trend shows an early spike for non-members and its decline during training.

The early spike in non-member roughness and the subsequent change in distribution overlap suggest that optimization affects member and non-member responses differently.

Early training. Non-member roughness is elevated during the initial epochs and decreases as the model fits the task. This separation provides a possible source of early leakage for gradient-based baselines. Its evolution is consistent with adaptation of the model to the training topology.

Later training. The member and non-member roughness distributions change as optimization continues. Their overlap remains low in the late regime, reaching 0.123 in the displayed diagnostic. This residual separation is consistent with persistent membership cues after task performance has largely converged.

E.7 Supplementary Experiments

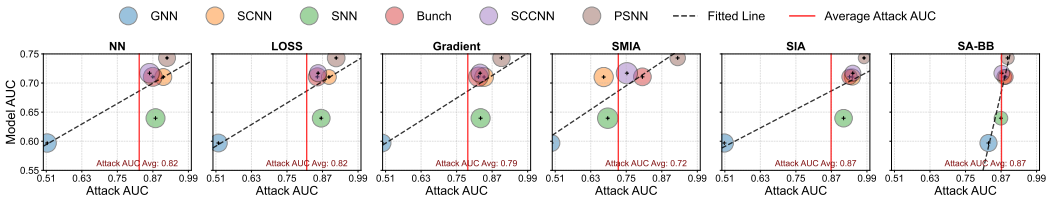


Figure 14: Additional privacy–utility trade-offs for the five baseline attacks on CORA.

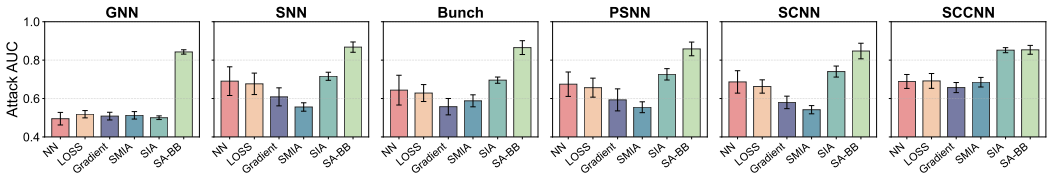


Figure 15: Comparison of Attack AUC across different backbones on the dataset SEMANTIC.

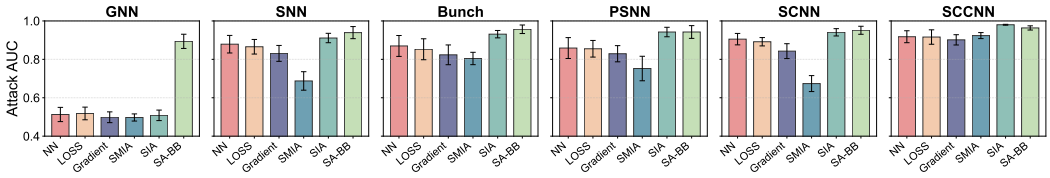


Figure 16: Comparison of Attack AUC across different backbones on the dataset CORA.

This subsection reports additional experimental results omitted from the main text due to space constraints. Figure 14 and Figure 16 present the additional privacy–utility trade-off on CORA and

the detailed attack performance on SEMANTIC. Figure 17 extends the defense evaluation to the SCCNN backbone on SEMANTIC and CORA, and further includes results on other backbones and on 3-simplex (tetrahedron) prediction tasks. The same broad trends appear across these settings: higher topological expressivity correlates with higher membership leakage, and DTPI yields the most favorable privacy–utility trade-off among the compared baselines.

Table 11: **Defense comparison on Trajectory-prediction task.** We report trajectory-prediction accuracy and attack AUC under five baseline attacks. Higher **Acc.** is better, while lower attack AUC is better. Best and second-best *defended* results in each column are shown in **bold** and underlined, respectively; the undefended `None` row is included only for reference.

Defense	Acc.	NN	Loss	Gradient	SIA	SMIA
None	0.652 ± 0.041	0.892 ± 0.038	0.702 ± 0.015	0.952 ± 0.015	0.956 ± 0.011	0.965 ± 0.005
DTPI	$\mathbf{0.572 \pm 0.050}$	0.791 ± 0.036	$\mathbf{0.588 \pm 0.009}$	$\mathbf{0.874 \pm 0.061}$	0.756 ± 0.009	$\mathbf{0.838 \pm 0.016}$
DP-SGD	0.526 ± 0.077	$\mathbf{0.788 \pm 0.044}$	0.684 ± 0.074	0.920 ± 0.024	0.761 ± 0.007	0.844 ± 0.011
GEP	0.568 ± 0.051	0.800 ± 0.040	0.598 ± 0.012	0.877 ± 0.041	0.792 ± 0.025	0.841 ± 0.009
SHAP	0.565 ± 0.043	0.801 ± 0.041	0.601 ± 0.014	0.884 ± 0.036	0.790 ± 0.021	0.845 ± 0.007
Adv-Reg	0.295 ± 0.144	0.798 ± 0.046	0.594 ± 0.035	0.893 ± 0.036	0.864 ± 0.014	0.874 ± 0.007
KD	0.294 ± 0.142	0.795 ± 0.041	0.596 ± 0.029	0.923 ± 0.011	$\mathbf{0.743 \pm 0.013}$	0.870 ± 0.012
Early-Stop	0.356 ± 0.194	0.826 ± 0.040	0.608 ± 0.081	0.900 ± 0.030	0.861 ± 0.018	0.858 ± 0.004

Trajectory Prediction. We evaluate DTPI against five baseline membership attacks on synthetic trajectory prediction. Each trajectory sample is represented by a sparse edge-flow prefix on the constructed simplicial complex, and the model predicts the next node from the local neighborhood of the prefix's last node. We use the same membership protocol as in simplex prediction: training trajectories are treated as members, held-out trajectories as non-members, and validation trajectories are excluded from membership labels. The MIA-simple/NN attack uses the model's next-node probability vector, while the other attacks use the corresponding loss, gradients, or hidden representations. Table 11 shows that DTPI largely achieves the lowest attack AUCs across different attacks while maintaining model utility, supporting its use across simplex and trajectory prediction.

E.8 ADDITIONAL STATEMENTS

Availability (Reproducibility). The implementation package includes SA-BB and DTPI, training and evaluation scripts, and dataset preprocessing code. The code is available in an *anonymous repository* at MIAsonSCNs.

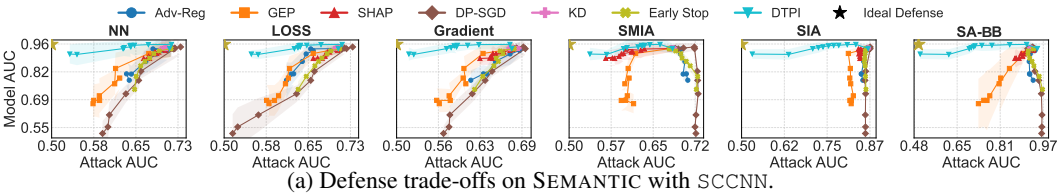
Ethical Statement. This work studies privacy risks and defenses for simplicial complex neural networks under membership inference attacks. All experiments are conducted on publicly available benchmark datasets and are intended to improve the safe and responsible deployment of learning systems. The proposed attack is presented solely for security analysis and to motivate effective mitigation strategies; it does not introduce harmful content.

Broader Impacts. This paper mainly contributes a more careful understanding of privacy risk in higher-order geometric learning. Its most direct positive impact is to help researchers and practitioners evaluate leakage in SCNs more systematically and to provide a defense that can reduce that leakage while preserving utility. As with many privacy-auditing studies, the main risk is that attack techniques may also inform stronger evaluations of vulnerable systems; for that reason, we present the attack together with a mitigation method and evaluate only on public benchmark datasets.

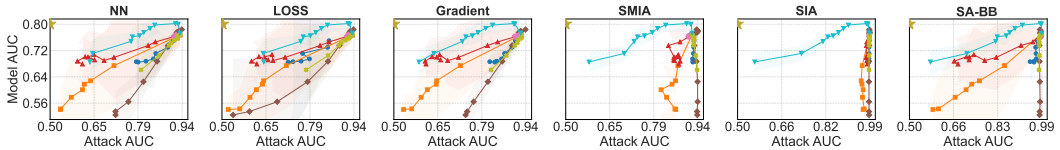
Assets and Licensing. We use publicly available benchmark datasets and cite their original sources in the main text and appendix. This manuscript does not provide a complete in-paper inventory of dataset and code licenses or terms of use; these details should accompany the public release package.

AI USE STATEMENT

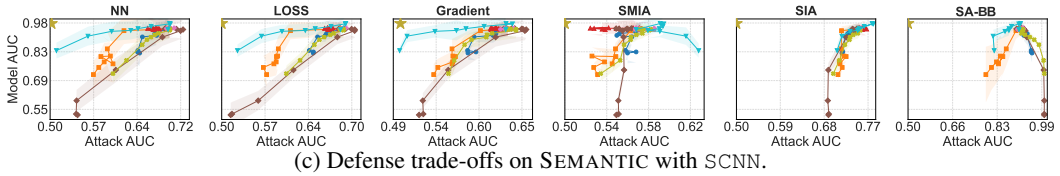
An LLM was used to polish writing.



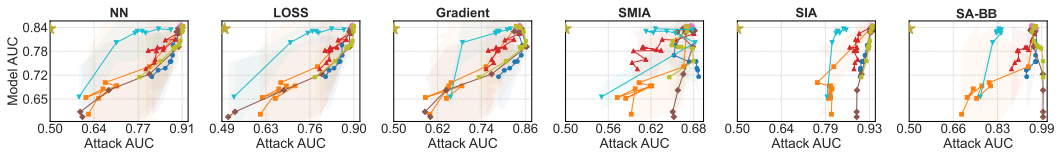
(a) Defense trade-offs on SEMANTIC with SCCNN.



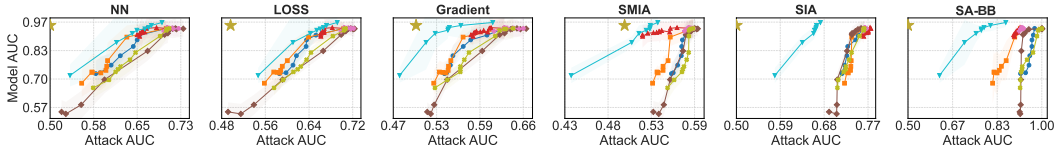
(b) Defense trade-offs on CORA with SCCNN.



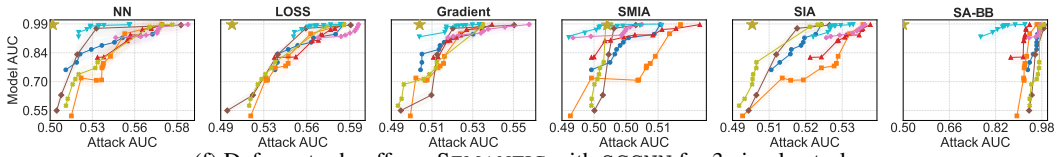
(c) Defense trade-offs on SEMANTIC with SCNN.



(d) Defense trade-offs on CORA with SCNN.



(e) Defense trade-offs on SEMANTIC with PSNN.



(f) Defense trade-offs on SEMANTIC with SCCNN for 3-simplex tasks.

Figure 17: Extended evaluation of defense trade-offs. We report privacy–utility trajectories under five baseline attacks for SCCNN on SEMANTIC and CORA (a–b), SCNN on SEMANTIC and CORA (c–d), PSNN on SEMANTIC (e), and SCCNN on 3-simplex prediction (f). The defense dynamics align with those reported in Section 6.3, confirming the robustness of DTPI across different backbones and tasks.