ROBUST TO THE WRONG ATTACKER: SEPARATING MODEL FROM ATTACKER IN GRAPH TRANSFORMER ROBUSTNESS

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph-transformer robustness comparisons can use edge edits chosen by a linearized GCN surrogate. Poor surrogate fit can understate vulnerability and reverse model rankings. We introduce a paired, reference-relative estimand: the surrogate's shortfall against a stated strong attack on the victim minus the corresponding shortfall on a named GCN baseline, evaluated on common targets and paired over ten seeds. Our contribution is a controlled evaluation protocol and paired estimand for measuring surrogate-induced bias; the attack procedure is deliberately conventional.

On Cora and Citeseer, linearized-surrogate GCN-to-transformer gaps are 16 to 24 points versus 4 to 13 under full-access greedy search. The excess is positive in all eight victim-graph cells against a plain GCN and separated from zero in six; against an accuracy-matched GCN it is positive in six and separated above zero in five. In three of these sixteen comparisons the surrogate inverts the ranking. On Pubmed, at nearly equal clean accuracy, the excess is 9.9 and 9.4 points against the two GCNs, both separated from zero. An exhaustive per-step candidate reference yields positive estimates separated from zero for the GPS-style transformer against both baselines on both graphs. No excess is detected for the official GPSConv layer on Cora or for an SGFormer-style victim, so the claim concerns surrogate fit, not attention.

With an attacker-owned shadow and at most 50 score queries per target, our diagnostic falls no more than 5 points below greedy white-box search across 16 complete-record conditions and sometimes exceeds it; in the official pipeline of Foth et al. (2025) it flips 61 percent of targets against 63 for their adaptive white-box attack. The protocol also separates defences: adversarial training lowers attack success, whereas a Jaccard edge filter mainly reduces transferability. Of its apparent 32.3-point protection against blind transfer on Cora, filter-aware transfer recovers 24.7 without victim weights or victim queries.

1 INTRODUCTION

A robustness result reflects both the model and the attack used to evaluate it. For message-passing networks, common inexpensive attacks such as Nettack (Zügner et al., 2018) and GOttack (Alom et al., 2025) select edge edits with a linearized GCN surrogate. Graph transformers (Ying et al., 2021; Rampášek et al., 2022) are also vulnerable to small structural perturbations when evaluated with attacks that have full model access (Foth et al., 2025). The linearized attacks suggest a different conclusion; in three of the sixteen paired comparisons of Section 5, they reverse the ranking given by search with full model access. We examine whether such differences come from the victim or from the surrogate used to choose edits.

We use a standard transfer attack supplemented by score queries. A shadow model trained on public data proposes candidate edits, and the victim's output scores rank them. This is the discrete analogue of query attacks that use a transfer prior (Cheng et al., 2019). We use this procedure as an evaluation instrument, not as a new attack method. The analysis asks how a reported model gap changes when a weak surrogate is replaced by a stated strong reference, whether a limited query budget changes the

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.

comparison, and whether a defence removes vulnerability or only transferability. Controls with GAT and GIN also show surrogate deficits, smaller than the transformer's on Cora (Section J). Removing global attention from our victim changes mean clean accuracy by only 0.2 points and leaves a similar deficit, whereas removing the message-passing branch reduces the success of every tested attack (Section Q). These results support a general concern about surrogate fidelity; they do not identify attention as the cause.

Reference-based comparison. Eq. (1) defines Δ relative to a chosen strong reference. It subtracts the linearized surrogate's deficit on a stated GCN baseline from the same deficit on the victim. Changing the reference or baseline can change the result, so Δ is not a causal share intrinsic to the model. Section 5 reports the estimates on shared targets across ten seeds with paired intervals. Our contributions are (i) a paired, reference-relative estimand that makes an architecture comparison explicit about its baseline and its reference; (ii) evidence that this excess is large for some graph transformers and not detected for others, and that in three comparisons the surrogate inverts which model looks more robust; (iii) a shadow-and-query diagnostic that, without victim weights, falls no more than 5 points below greedy white-box search across 16 complete-record conditions; and (iv) a defence decomposition showing that an inference-time filter can reduce transferability far more than vulnerability, with a check that uses neither victim weights nor victim queries. The evidence concerns mismatch between surrogate and victim, not an intrinsic property of attention.

Why not only run an adaptive attack? An evaluator who holds the weights should run one, and we do. Δ answers a different question: how far a gap measured with a linearized surrogate, including any gap reported that way, moves when a stated strong reference replaces the surrogate. Prediction agreement, the fidelity number usually reported for surrogates, shows no detectable rank association with how much of the gap a trained shadow recovers (Spearman 0.14 over thirteen cells, Section S). The practical check is to run one attacker that does not linearize the victim, and an attacker-trained shadow with the stated data access needs neither victim weights nor victim queries.

Attacker access. A score query submits a counterfactual adjacency and reads the victim's class scores. This access is literal in an inductive deployment and serves as an evaluator's instrument in the transductive settings used for most experiments. With a shadow and at most 50 queries per target, but without victim weights, gradients, or representations, the diagnostic falls no more than 5.0 points below greedy white-box search across 16 complete-record conditions and exceeds it in some (Section P). Several of the larger differences occur when the reference is weak or local message passing is absent, so we treat the method as a diagnostic and do not claim dominance. On the GPS-style CLUSTER victim and in the official pipeline of Foth et al. (2025), score queries add only about 1.0 and 1.6 points over shadow transfer without queries.

Defence evaluation. Adversarial training lowers success for every attacker tested: on the Cora transformer the strongest result falls from 64 to 45 percent. An inference-time Jaccard edge filter changes no parameters, yet on targets shared across conditions it reduces shadow transfer without queries on Cora from 58 to 26 percent. Its apparent cost is 32.3 points against a blind transfer attacker and 7.7 against an attacker that knows the filter identity and threshold and proposes only edges that survive it. Measured directly, the informed attacker recovers 24.7 points without victim weights or victim queries.

What the controls change. Each control changes a conclusion. On each victim's own targets the white-box gap reads $+2$ and -2 points, not 8.3 and 5.7 as on shared targets. The accuracy-matched GCN lowers Δ in all eight cells, on Cora from 14.2 to 7.7 with an interval containing zero; the exhaustive per-step reference separates that cell from zero, as does a retuned cohort (Sections 7 and M). Greedy search lands 7.0 and 8.7 points below that reference, so the white-box references are not upper bounds. Without multiplicity correction the negative GRIT-style cell on Cora would count as separated. No excess is detected for the official `GPSConv` layer on Cora or for the SGFormer-style victim, so we make no claim for graph transformers in general. On ogbn-arxiv, where shadow transfer is 0.7 point below the linearized proposal, the cohorts use separate targets, so we report no Δ.

2 Related Work

Structure attacks and black-box attacks on GNNs. Nettack (Zügner et al., 2018) introduced targeted evasion and poisoning through a linearized GCN surrogate; later work relaxes the discrete perturbation (Xu et al., 2019b; Zügner & Günnemann, 2019; Bojchevski & Günnemann, 2019; Chen et al., 2018; Li et al., 2023; Wang & Gong, 2019), scales it (Geisler et al., 2021), or learns it (Dai et al., 2018). GOttack (Alom et al., 2025) restricts candidate endpoints to nodes in peripheral graphlet orbits; BoCloak (Mukherjee et al., 2026) adds plausibility constraints. Black-box attacks transfer from a surrogate (Papernot et al., 2016; 2017), query the victim (Mu et al., 2021; Wan et al., 2021; Gao et al., 2026), or work in stricter settings, the strictest allowing no model output at all (Chang et al., 2020; Ma et al., 2020; 2022; Wang et al., 2024; Xu et al., 2022); we run Chang et al. (2020) and Wan et al. (2021) under our protocol (Section O). In vision, query attacks (Ilyas et al., 2018; Guo et al., 2019) can take a surrogate's gradient as a prior (Cheng et al., 2019). All of that graph work attacks message-passing or embedding models, much of it through GCN surrogates. We build on DeepRobust (Li et al., 2020) and the official GOttack code; Sun et al. (2023) and Jin et al. (2021) survey the field.

Evaluation and defences. Ngo et al. (2026) document that adversarial GNN studies do not share experimental settings, that victim models are often left unoptimized, and that common target selection underrepresents high-degree nodes, and they propose the stratified protocol we adopt; Zheng et al. (2021), Gosch et al. (2023) and Chen et al. (2022b) make related cases. Athalye et al. (2018) showed in vision that defences breaking the attacker's gradient are credited with robustness they do not have, and Carlini et al. (2019), Tramèr et al. (2020) and Mujkanovic et al. (2022) argue a defence is only as strong as the attack used to test it; Mujkanovic et al. (2022) is the closest to Section 6: under adaptive white-box attacks, most of the seven graph defences they test, Jaccard-GCN among them, show little or no gain over an undefended model. Our results identify a related failure mode that needs neither the victim's weights nor an adaptive gradient to expose: a defence can reduce transferability more than the underlying vulnerability, and knowledge of the deployed filter, or score queries to the deployed system, reveals it. Structural defences include GCN-Jaccard (Wu et al., 2019), low-rank reconstruction (Entezari et al., 2020), GNNGuard (Zhang & Zitnik, 2020) and Pro-GNN (Jin et al., 2020); for graph transformers Zhu et al. (2024) counter the low-pass behaviour of attention with a learned flexible-pass filter.

Graph transformers and their robustness. Graph transformers use local or global attention, with or without message passing (Dwivedi & Bresson, 2020; Kreuzer et al., 2021; Ying et al., 2021; Chen et al., 2022a; Rampášek et al., 2022; Ma et al., 2023), and Laplacian or random-walk encodings (Dwivedi et al., 2022); sparse or linear variants (Choromanski et al., 2021; Shirzad et al., 2023; Chen et al., 2023; Wu et al., 2023) scale them. Foth et al. (2025) study the adversarial robustness of graph transformers. They relax components that otherwise obstruct gradient attacks, find the tested models highly fragile in many settings, report that adversarial training with their attack substantially improves the robustness of Graphormer, and find stronger transfer between graph transformer models than from a GCN. They also report their adaptive attack ahead of Nettack on targeted CLUSTER at small and moderate budgets, with the ordering depending on the architecture at larger ones. Where Foth et al. (2025) ask whether graph transformers are robust once gradient attacks are made to work, we measure how much a between-model robustness gap moves across attackers by pairing both models on identical targets and subtracting baseline-side surrogate error; the diagnostic can be run without victim weights.

3 Threat Model and Method

Setting. Let $G = (\mathbf{A}, \mathbf{X})$ be a graph with adjacency $\mathbf{A} \in \{0,1\}^{N\times N}$ and features $\mathbf{X}$, and let f_θ be a graph transformer trained for node classification. For a target node t with label y_t, the attacker may flip at most k entries of row (and, symmetrically, column) t of $\mathbf{A}$: a direct, evasion-time structure attack. Indirect edits, which Nettack also allows, are outside the protocol we adopt, since Ngo et al. (2026) and the targeted setting of Foth et al. (2025) are both direct, and one edit set for every attacker is what makes a comparison across surrogates mean anything. Success means $\arg\max f_\theta(\mathbf{A}', \mathbf{X})_t \neq y_t$. We write $m_\theta(t; \mathbf{A}) = \log f_\theta(\mathbf{A}, \mathbf{X})_{t,y_t} - \max_{c\neq y_t} \log f_\theta(\mathbf{A}, \mathbf{X})_{t,c}$ for the victim's margin.

The quantity we measure. Fix a victim v, a target list, a budget, a baseline b, and a specified strong reference attack. Write $\mathrm{succ}_a(v)$ for the fraction of targets attack a flips, and $\mathrm{succ}_{\mathrm{lin}}(v)$ for the mean over the three linearized-GCN attacks (GOttack, Nettack, and the linearized GCN over all candidates) unless stated otherwise. The surrogate deficit on a model is $D(v) = \mathrm{succ}_{\mathrm{ref}}(v) - \mathrm{succ}_{\mathrm{lin}}(v)$, and we report

$$\Delta(v; b, \mathrm{ref}) = D(v) - D(b). \tag{1}$$

Equivalently, Δ is the linearized attacks' GCN-to-victim gap minus the reference attack's; a positive Δ means the surrogate makes the victim look more robust, relative to the baseline, than the reference does. The quantity is relative to the stated baseline and reference; it is not a causal share intrinsic to the model. Changing either can change Δ. We estimate it per seed, attacking both models on one target list drawn from nodes every compared model classifies correctly, and report the mean with a paired t interval over seeds; without such a list we report only D for each model.

Attacker knowledge. All attacks use the graph, features, task, and selected target's ground-truth label y_t; this is an evaluator-oriented threat model, not a claim about an unlabeled deployed adversary. Victim access has three levels: zero-query and score-query attackers lack victim weights; full-access references have them. A *shadow* is trained by the attacker on public data with a stated label set. The main experiments use the victim's training labels. An attacker with no queries has the graph, features, task, label set, and shadow but never evaluates the victim. The score-query attacker also receives the victim's class-score vector for target node t on each submitted graph, with one submission counted as one query. This access is literal in an inductive deployment, where each target arrives with its own graph, and an evaluation instrument in the transductive experiments. Neither the zero-query nor the score-query attacker has the victim's parameters, gradients, or internal representations; Table 14 tests architecture mismatch. The attacker is unaware of the defence in every condition except the informed filter evaluation in Section 6. There, our attacker knows the filter identity and threshold but gets no victim weights and makes no victim queries.

Shadow model. The attacker trains a shadow f_ϕ on the same task; in the main experiments a graph transformer of the victim's family, and Table 14 shows a trained GCN also works. One backward pass through the shadow with a relaxed adjacency scores flipping edge (t, j) as $s_j = (1 - 2A_{tj})\,\partial m_\phi(t; \mathbf{A})/\partial A_{tj}$, the first-order predicted change in margin, with positional encodings held fixed.

Shadow transfer without queries. Greedily flip the edge with the most negative s_j, recompute scores on the perturbed graph, and repeat k times. The perturbed graph is then applied to the victim.

Score-query attack. At each of k steps, form a candidate set C of size M and query the victim's margin for each single-flip candidate, choosing the one with the smallest margin. We define three candidate priors: random (M random nodes), shadow (the M nodes with the most negative s_j), and *hybrid* ($M/2$ shadow top-scored plus $M/2$ random). Total queries are at most Mk; with $M = 10$ and $k = 5$ this is 50 queries per target, against N queries per step for a naive score-based search. Algorithm 1 gives the procedure. An allowed-edit mask Ω_t imposes realistic constraints in the spirit of BoCloak (Mukherjee et al., 2026): additions only, or additions within two hops of the target.

Baselines and full-access references. GOttack (Alom et al., 2025) with its official code (orbit-periphery candidates, linear GCN surrogate) is the black-box baseline. Random edge additions give a floor. Two white-box attacks with full access to the victim serve as references, not ceilings, since both are first-order searches a query attack can exceed: greedy gradient on the victim, and a PRBCD-style relaxed attack (projected gradient ascent on a continuous perturbation with $\sum_j p_j \le k$, discretized by top-k and Bernoulli sampling (Geisler et al., 2021)). Both are run with the encoding frozen and, in Table 7, Section 6, Section Q and Section M, differentiated, the adaptive setting of Foth et al.

4 Experimental Setup

Victims and data. The main victim is a GPS-style graph transformer (Rampášek et al., 2022): two layers of a local GCN block (Kipf & Welling, 2017), global four-head attention, and a feed-forward block with layer normalization, hidden size 64, and an 8-step random-walk structural

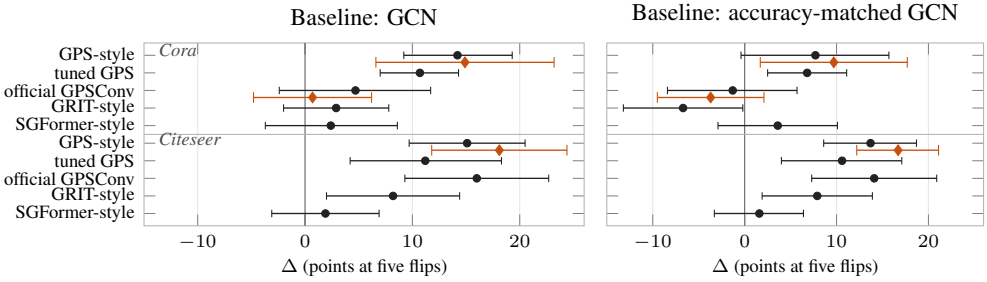


Figure 1: Surrogate excess Δ for every Cora and Citeseer victim-graph cell with both arms on one common target list, ten paired seeds (nine for the SGFormer-style Cora cell against the matched GCN), 95 percent t intervals (Tables 15 and 20). Circles: greedy white-box reference (through the encoding for GRIT-style cells, relaxed white-box for SGFormer-style cells); diamonds: exhaustive per-step reference on both arms (Sections F and 7).

Table 1: Official Foth et al. (2025) pipeline on their 16-layer GraphGPS victim, 90 targets, five splits. Victim-side cost is measured for the white-box attack; for the hybrid it is a normalized estimate for 50 victim forward passes that excludes attacker-owned shadow training and search. Rows with zero victim-side cost still incur attacker-owned computation, so the table compares victim access and victim-side cost, not total compute.

			Success		Victim-side cost	
Attacker	Access	Queries	3.6 flips	9.8 flips	s/target	peak MiB
Random edge flips	none	0	8	16	0	0
Nettack, linearized GCN surrogate	none	0	31	65	0	0
Their PRBCD on a GCN, transferred	weights (other model)	0	54	88	0	0
Their PRBCD on an attacker-owned GCN, transferred	shadow	0	52	88	0	0
Their PRBCD on an attacker-owned GT, transferred	shadow	0	60	89	0	0
Hybrid with up to 50 score queries (ours)	shadow + scores	49	61	88	2.8	21
Their adaptive PRBCD, white-box	weights	0	63	88	36.4	187

encoding (Dwivedi et al., 2022) on the input. Models are trained with Adam and early stopping on the DeepRobust validation split, reaching 70 to 74 percent on Cora (2,485 nodes, largest component) and 66 to 71 on Citeseer (2,110) (Sen et al., 2008), below GCN; Polblogs (Adamic & Glance, 2005), where no attacker exceeds 48 percent at five flips, appears only in Section Q. Shadow models use a different seed from the victim and, in the main experiments, the victim's training labels (test agreement 82.0 and 81.7 percent over the ten seeds of Section 5). In Section N the shadow is trained by the pipeline of Foth et al. (2025) on the public data, with no contact with the victim's run.

Targets and protocol. The main experiments use the stratified target design of Ngo et al. (2026), with six nodes per stratum rather than their ten: 6 highest-degree, 6 lowest-degree, 6 highest-margin, 6 lowest-margin and 6 random correctly classified test nodes per seed, so 30 targets per seed against their 50 (mean target degree 11 to 18). Ten seeds give 300 attacked nodes per condition. Greedy attacks run over budgets $k = 1, \ldots, 5$, evaluated on the fixed victim after each flip; success is the fraction of targets flipped after exactly k flips, their convention, which need not be monotone in k; Section Q reports what the alternative does.

5 Results

All results in this section use the fair-target protocol of Ngo et al. (2026) and five edge flips unless stated, with ten seeds on Cora and Citeseer (30 targets per seed; five to ten seeds elsewhere). Intervals are paired two-sided 95 percent t intervals over seeds; one that contains zero leaves the cell unresolved and is not evidence of equivalence.

Surrogate and victim combinations. Table 14 crosses the surrogates and shadows with the transformer and a two-layer GCN of the same width, ten seeds.

Shared targets. Table 7 and Fig. 3 compare three victims on the identical 30 targets per seed: the transformer, a GCN, and an accuracy-matched GCN. Greedy white-box search gives GCN-to-

Table 2: Reference-relative surrogate excess on common targets. $D(v)$ is greedy-white-box success minus the mean success of the three linearized attacks on the victim; $D(b)$ is the same deficit on the baseline. Percentage points over ten paired seeds, rounded independently. SD uses `ddof=1`; CIs are two-sided paired t intervals. `paper_evidence/reproduce.py` regenerates each row.

Condition	$D(v)$	$D(b)$	Δ	SD	95% CI
Cora, GPS-style vs GCN	21.1	6.9	14.2	7.1	$[9.2, 19.3]$
Cora, GPS-style vs matched GCN	21.1	13.4	7.7	11.3	$[-0.4, 15.7]$
Citeseer, GPS-style vs GCN	16.4	1.3	15.1	7.6	$[9.7, 20.5]$
Citeseer, GPS-style vs matched GCN	16.4	2.8	13.7	7.0	$[8.6, 18.7]$

transformer gaps of 8.3 and 12.7 points on Cora and 5.7 and 4.0 on Citeseer. The corresponding 95 percent t intervals are $[2.0, 14.6]$, $[4.3, 21.1]$, $[0.8, 10.6]$, and $[-1.5, 9.5]$. Differentiating the encoding in the adaptive PRBCD-style attack moves the gaps by 1.0 to 1.3 points. On each victim's own correctly classified nodes the white-box gap reads $+2$ and -2, showing that target selection itself materially changes the reported comparison.

Reference-based excess. Across the three linearized attacks, the gap between the GCN and transformer victims ranges from 16 to 24 points on shared targets. The estimates of Δ (Table 2) are 14.2 and 15.1 points relative to the plain GCN, and 7.7 and 13.7 relative to the GCN matched for accuracy, on Cora and Citeseer. The Cora interval for the accuracy-matched baseline, 7.7 points with $[-0.4, 15.7]$, contains zero: that cell neither establishes a nonzero excess nor supports equivalence, and we draw no conclusion from it. Two further measurements bear on it: the per-step oracle on both arms of this cohort gives 9.7 with $[1.7, 17.7]$ over ten seeds (Section 7), and the retuned cohort (Section E) is separated from zero under the greedy reference, 6.8 with $[2.5, 11.1]$, and under a pre-specified adaptive reference on both arms, 6.8 with $[3.2, 10.3]$ (Table 15 and Section M). Table 15 carries every greedy-reference Cora and Citeseer condition with both arms on a common target list: eight victim-graph cells, each read against both baselines; Fig. 1 plots them with the SGFormer-style and per-step-oracle cells. Against the plain GCN Δ is positive in all eight and separated from zero in six; against the accuracy-matched GCN it is positive in six and separated above zero in five. All eleven positive separations survive a Benjamini-Hochberg correction over the sixteen cells at a false discovery rate of 0.05, and six survive the stricter Holm correction (Section L). Of the two negative point estimates, official `GPSConv` on Cora is not separated from zero under either reference (the per-step oracle on both arms: -3.7 with $[-9.5, 2.1]$, Section F); the GRIT-style victim on Cora, which has no local message passing, is separated on the negative side (-6.7 with $[-13.2, -0.2]$) but not after the Benjamini-Hochberg correction (adjusted $p = 0.061$). Matching lowers Δ in all eight cells; the accuracy-matched baseline is the demanding one. The surrogate can also invert which model looks more robust: in three of the sixteen cells (tuned GPS on Cora against the plain GCN, official `GPSConv` on Citeseer against both baselines) the linearized attacks find the transformer 5 to 10 points harder to attack than the GCN and the reference finds it 4 to 9 points easier, both paired intervals excluding zero. Random edge addition, which uses no surrogate, also finds the transformer 5 to 17 points harder to attack; that gap reflects lower noise sensitivity, not surrogate error.

Shadow transfer and score queries. Without queries, a trained shadow of either family reaches 51 to 52 percent on Cora and 45 to 50 on Citeseer; greedy full-access search reaches 57 and 53. The hybrid with 50 queries reaches 56 to 57 and 53 on the same targets. The hybrid's gap to the GCN victims is similar to the full-access gap: 12 and 13 points on Cora, compared with 8 and 13, and 6 and 3 on Citeseer, compared with 6 and 4 (Table 7; differences use unrounded means). Shadow agreement is 97 to 98 percent on the GCN against 82 on the transformer.

Victim accuracy controls. Our main transformers reach 71.9 and 67.7 percent, below the GCN controls. Section E repeats the comparison after a sweep of 72 configurations, and Section F uses the official GraphGPS layer. The remaining robustness difference depends on the graph. Same-family trained and querying attackers leave it unresolved on Cora, whereas full-access search still finds the tuned Citeseer transformer about 12 points harder to attack than its GCN controls. For the official layer on Cora we detect no larger surrogate deficit on the transformer. We therefore do not treat it as a fixed property of the architecture.

Table 3: Six attackers, same victims, three conditions, ten seeds, success at five flips. Surrogate is the best linearized-GCN attack; transfer is the shadow's gradient with no queries. Hybrid spends at most 50 score queries; *hidden* has the undefended model answer them while deployment still filters. White-box is greedy gradient on the unfiltered model. *Defence-aware* means the attacker knows the filter identity and threshold and proposes only edges the filter keeps. Targets are re-drawn per condition among nodes that condition's deployed system classifies correctly; Section G.1 shares one list across conditions.

			No victim weights			White-box	
Victim	Condition	Surrogate	Transfer	Hybrid	Hybrid, hidden	greedy	defence-aware
GT, Cora	undefended	44	61	64		64	
GT, Cora	adversarially trained	32	34	45		41	
GT, Cora	Jaccard filter	34	28	52	35	34	59
GT, Citeseer	undefended	46	51	59		60	
GT, Citeseer	adversarially trained	30	28	36		34	
GT, Citeseer	Jaccard filter	35	41	54	50	51	57

Shadow training and architecture. A trained GCN shadow (Table 7) is within 1 point of a trained transformer shadow with 50 queries and, without queries, 1 point above it on Cora and 5 above on Citeseer, whereas the linearized GCN is substantially weaker with the same candidate set. A converse control attacks victims that the surrogate does not linearize. The best linearized attack falls 7.7 and 5.7 points below greedy full-access search on GAT, and 8.7 and 10.7 on GIN. The GCN comparison is 6.0 and 1.0 points over the same ten seeds. For the GPS-based transformer, the corresponding deficits are 19.7 and 14.7. The paired intervals in Section J separate the GIN and transformer deficits from zero on both graphs; the comparisons support a general mismatch effect and a larger deficit for the Cora transformer, and they do not establish equivalence between the Citeseer transformer and GIN.

Official Foth et al. pipeline. Table 1 repeats the comparison in the released code of Foth et al. (2025). On their GraphGPS victim, shadow transfer without queries, our attack with 50 score queries, and their adaptive full-access attack flip about 60, 61, and 63 percent of targets at the small budget. The paired comparison across splits shows numerical closeness, not equivalence. The point is not parity: the fragility Foth et al. (2025) exposed with full model access is approached on their own GraphGPS victim by an attacker holding neither victim weights nor victim gradients. Their adaptive attack remains ahead on the GCN victim, 71 against 67. The costs in Table 1 are victim-side: a shadow search at the normal budget adds 36.5 seconds per target on the attacker's side, and the hybrid's larger shadow search was not timed separately, so the method reduces victim access and victim-side computation, not total computation.

Published black-box baselines. The benefit of 50 queries depends on the candidate set. Random candidates reach 46 and 47 percent, compared with 64 and 59 for the hybrid. The Bayesian-optimization attack of Wan et al. (2021), given the same budget but no shadow, reaches 51 percent on each graph. It remains 13 and 8 points below the hybrid, which indicates that the shadow proposals, rather than the queries alone, account for most of the gain. GF-Attack (Chang et al., 2020) uses no queries and falls below random additions on all four victims, including the GCNs; its objective is untargeted. Section O reports both baselines.

6 Defences and Further Settings

Two defences with comparable headline reductions are doing different things, and only one largely survives an attacker that knows the defence. Table 3 puts adversarial training against a PRBCD-style inner attack at a 5 percent budget (Section G) and GCN-Jaccard edge filtering at inference (Wu et al., 2019), which drops edges whose endpoints have feature Jaccard similarity below 0.01 before every prediction at a cost of at most one accuracy point, against the same attackers.

Adaptive filter evaluation. For the filtered Cora transformer, transferred edits reach 28 percent, an attacker that differentiates the unfiltered model reaches 34, our attacker with 50 score queries reaches 52, and a full-access attacker that knows the filter identity and threshold and proposes only surviving edges reaches 59. On shared targets, the apparent cost of the filter against blind transfer

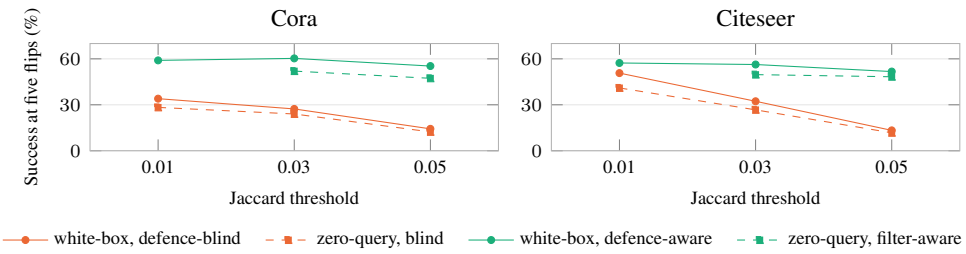


Figure 2: Attack success at five flips on each GPS-based victim's fair targets across ten seeds as the GCN-Jaccard threshold increases. The two attackers that do not model the filter fall to about 13 percent; the two that know the filter identity and threshold stay between 47 and 60. The zero-query filter-aware attacker was run at 0.03 and 0.05 on these targets; at 0.01 it appears on the shared targets of Table 11. Deployed clean accuracy falls from 70.9 to 67.5 on Cora and from 67.3 to 62.5 on Citeseer.

is 32.3 points on Cora and 9.0 on Citeseer. An attacker that uses no queries but has the same filter knowledge reduces those costs to 7.7 and -0.3 points. Measured directly on the filtered condition, awareness alone is worth 24.7 and 9.3 points without victim weights or victim queries. The three figures are paired means rounded independently, so they are not intended to subtract. The Citeseer cost of -0.3 neither shows a benefit nor establishes equivalence. A control in Section H deploys the filter at one threshold and lets the attacker assume another. Awareness degrades under every wrong guess and the graph decides how far: on Cora even the worst wrong guess stays well above blind transfer, while on Citeseer a badly wrong guess is close to worthless, so the threshold itself matters.

Mechanism check. Fig. 2 is consistent with a preprocessing mask invalidating a gradient computed without the filter. The filter changes no parameters; it changes which proposed edits reach the model. When the same 50 queries are answered by the undefended model while the deployed system still filters, attack success falls from 52 to 35 on Cora and from 54 to 50 on Citeseer. Recovery therefore depends on consulting the deployed system, not on changing the shadow, and the querying attacker is not told that a defence is present.

Adversarial training behaves differently. It lowers agreement from 82.0 to 78.5 percent and lowers success for every tested attacker, including the strongest. On the Cora transformer, no attack exceeds 45 percent, compared with 64 percent without the defence. The GCN results change by at most 5.3 points under adversarial training (Table 10). Under the filter the Cora GCN attacks lose 6 to 9 points, except shadow transfer without queries and greedy white-box search, which lose 19 and 18; Citeseer changes by at most 3.3 points. The shared-target control gives the same interpretation. Adversarial training reduces transfer, the score-query attack, and full-access search by 19 to 25 points. The filter separates those attacks. When the 50 queries are answered by the unfiltered model, the query attack loses 31.7 and 9.0 points, close to the blind transfer losses of 32.3 and 9.0 (Section G.1).

A filter evaluated only with transferred perturbations, or with gradients from the unfiltered model, can therefore receive credit for protection that does not persist against an informed evaluation. This is the graph structure analogue of the gradient masking concern described by Athalye et al. (2018). Fifty score queries to the deployed system provide a simple check.

Edit constraints. Following the plausibility constraints considered by BoCloak (Mukherjee et al., 2026), forbidding deletions costs the attacker at most two points. Restricting new links to nodes within two hops reduces every attack on the transformer to within 4 to 12 points of random additions. The GCN victims under the two-hop constraint show the same pattern, ten seeds throughout.

Additional datasets. Across Pubmed, Chameleon, ogbn-arxiv and CLUSTER the surrogate mismatch ranges from near zero to large. On Pubmed a paired run on common targets, added after the Cora and Citeseer analysis, puts the transformer, the GCN and the accuracy-matched GCN at 85.5, 85.4 and 85.3 percent clean accuracy. Against the plain GCN, the GCN-to-transformer gap is 9.2 points under the linearized attacks and -0.7 with $[-6.1, 4.8]$ under greedy white-box search; over ten seeds, Δ is 9.9 with $[5.3, 14.5]$ against the GCN and 9.4 with $[5.4, 13.5]$ against the matched GCN (Section C). The other settings have no paired GCN arm, so we report surrogate deficits there;

Section B reports Chameleon in full. On Chameleon the GCN surrogate falls 32 points short of greedy white-box search on the sparse GPS victim and 37 points short on a GCN victim of the same graph, so this deficit is not specific to the transformer; the zero-query shadow recovers 26 of those points on either victim, and on the filtered split of Platonov et al. (2023) the deficits are 20 and 34 with the shadow recovering 11 and 27 (Section B). On ogbn-arxiv (169,343 nodes) (Hu et al., 2020) Nettack is infeasible, so the surrogate is a linear GCN used as a proposal generator. At five flips the linearized proposal reaches 62.7 percent and the shadow 62.0, a difference of -0.7, with white-box search at 64.0 and the hybrid at 65.3 (Table 6). Scale alone does not produce the mismatch, and the protocol does not manufacture a positive effect where the reference shows little headroom.

Inductive setting. CLUSTER (Dwivedi et al., 2023) is inductive. The victim and shadow share neither graphs nor weights but draw data from the same public generator, which favours transfer. On a GPS-style victim, shadow transfer without queries reaches 66 percent over ten seeds, within one point of the full-access means, and queries add one point. A Graphormer-style victim, whose only edge paths are a shortest-path attention bias and a degree encoding, is similarly vulnerable: full access reaches 69 and 71 percent and the hybrid 69, although there the zero-query shadow (59) gains little over the linearized surrogate and the queries do the work. Single edges therefore affect the prediction without local message passing (Section D).

7 Ablations and Limitations

Section Q collects the ablations. Against a hard-label interface the attacker keeps about half the query gain. The per-step oracle, greedy search that evaluates every candidate edit on the victim at each step rather than a global optimum, places the others: on fair targets over ten seeds, greedy white-box lands 7.0 and 8.7 below it and the hybrid 6.7 and 10.0. Used on both arms of the common-target control, it raises Δ on Citeseer from 15.1 to 18.1 against the plain GCN, $[11.8, 24.4]$, and from 13.7 to 16.7 against the matched one, $[12.2, 21.1]$. On Cora it moves Δ from 14.2 to 14.9 against the plain GCN, $[6.6, 23.2]$, and from 7.7 to 9.7 against the matched one, $[1.7, 17.7]$; all four intervals exclude zero.

The main victims are our own implementations, 7 to 12 points below the GCNs and below published accuracies for their architectures; Sections E and F control for this, and on Pubmed, at nearly equal clean accuracy, the paired excess is also positive and separated from zero (Section C). All attacks are evaluated as evasion, including GOttack, which its authors evaluate under poisoning. The zero-query baseline is a first-order shadow: a shadow-side lookahead costing no victim queries leaves the 50-query gain at $+0.3$ and $+3.0$ instead of $+3.3$ and $+7.7$ (Section Q). Another transformer victim shows no detectable excess. With an SGFormer-style victim in place of the GPS-style one, ten seeds (nine in one Cora cell), relaxed white-box reference on both arms, Δ is 2.4 and 3.6 on Cora and 1.9 and 1.6 on Citeseer; no interval excludes zero. The shadow adds little there: it recovers 1.1 of 13.4 points on Cora against 15.1 of 22.4 for GPS, and on Citeseer it falls below the linearized attacks. Sections R and S report the cells and checks. We study one filter, GCN-Jaccard at three thresholds, with a defence-aware attacker; low-rank reconstruction and random edge dropping are evaluated without one in Section H. The hybrid issues up to 50 score queries per target node, which a deployment could rate-limit or coarsen; the hard-label output is the extreme of that coarsening, we do not model detection, and the zero-query shadow remains when queries are refused.

8 Conclusion

Reported robustness depends on both the victim and the attack used for evaluation, and a linearized surrogate can reverse which model looks more robust. The quantity in Eq. (1) is relative to a stated reference and baseline, not a causal share intrinsic to the model. GAT, GIN and the control without global attention indicate that surrogate mismatch is the broader concern; it can be tested without victim weights, but the attacker and its access must be reported with the model.

What an evaluator can adopt. Four practices follow: name the edit-proposing surrogate; compare architectures on common targets; include an attacker not based on a linearized GCN; and test each defence with an informed attacker, because reduced transferability is not reduced vulnerability.

REPRODUCIBILITY STATEMENT

The supplementary material is an anonymized package with the raw per-seed result files, the run scripts, the third-party pipeline logs, and the checker described here; model checkpoints are omitted for size, and every result file records the seed and configuration that regenerate its victim and shadow. A canonical checker, `paper_evidence/reproduce.py`, maps each headline claim to exact input paths, hashes, included and excluded seeds, formulas, metrics, and units. It regenerates a machine-readable manifest and a compact table with n and sample standard deviation using `ddof=1`, fails if an input hash changes, and flags incomplete cohorts rather than imputing missing results. On the released tree it validates 1,124 inputs and regenerates 1,061 of 1,069 claim rows exactly, including the sign reversals and the multiplicity adjustment of Sections 5 and L and the rank correlations of Section S; the remaining 8 are the PRBCD-style gap rows of the tuned common-target cohort, whose GCN files carry no PRBCD run; no table reports them, and the Cora accuracy-matched arm of that cohort is instead supplied by the locked validation records of Section M, which the checker also regenerates. Every dataset reported in the paper, Cora, Citeseer, Pubmed, Polblogs, Chameleon, ogbn-arxiv and CLUSTER, has claim rows in the checker. No incomplete row supports a claim in this paper, and none contradicts a reported value. The GAT, GIN and GCN point deficits use ten seeds, as does the attention-only control. Additional per-table scripts regenerate appendix rows; `delta_all.py` rebuilds every row of Table 15 from the same per-seed files the other tables read, pairing seed by seed and printing the sample standard deviation and the t interval, and `delta_oracle.py` recomputes Δ with the per-step oracle of Section 7 on both arms, reporting the white-box comparison restricted to the oracle's own seeds so the two references are compared on identical runs. The retained common-target white-box intervals are those printed by `aggregate_common3.py`; `sdrows.py` uses Bessel's correction for the defence rows. Seeds are fixed: victim seed $1000 + s$ and shadow seed $2000 + s$ for index s. The Laplacian positional encoding used for Pubmed and Chameleon fixes ARPACK's starting vector and pins each eigenvector's sign to its largest-magnitude entry; the ogbn-arxiv runs use ARPACK's default starting vector. CLUSTER graphs use seeds $3000 + s$ for victim data and $4000 + s$ for attacker data. The replication inside the Foth et al. (2025) pipeline is in `gtr/` and `logs/` of the package, with its configurations, per-split query results, and the archived runtime and memory log.

ETHICS STATEMENT

This work studies evasion attacks on node classifiers with the aim of correcting how their robustness is evaluated. The attacks require either a trained surrogate or score access to the victim, both of which are already assumed in the literature we build on, and all experiments are on public benchmark graphs. The defence results identify partial mitigations, adversarial training and a two-hop plausibility check on new edges, and show that an edge filter evaluated only against attackers unaware of it can overstate its protection.

AI USE STATEMENT

Generative AI tools were used as supporting tools to propose and refine hypotheses, give feedback on methodology and experimental design, assist with implementation and debugging, support experiment orchestration, literature navigation and result interpretation, draft sections of the manuscript and its figures, check and format references, and edit language. The authors independently reviewed and verified that work, made all final scientific decisions, and take full responsibility for the study design, experiments, analyses, claims, source attribution and manuscript.

REFERENCES

L. A. Adamic, N. Glance. The political blogosphere and the 2004 U.S. election: Divided they blog. *LinkKDD*, 2005.

Z. Alom, T. G. B. Ngo, M. Kantarcioglu, C. G. Akcora. GOttack: Universal adversarial attacks on graph neural networks via graph orbits learning. *ICLR*, 2025.

A. Athalye, N. Carlini, D. Wagner. Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples. *ICML*, 2018.

A. Bojchevski, S. Günnemann. Adversarial attacks on node embeddings via graph poisoning. *ICML*, 2019.

N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, A. Kurakin. On evaluating adversarial robustness. arXiv:1902.06705, 2019.

H. Chang, Y. Rong, T. Xu, W. Huang, H. Zhang, P. Cui, W. Zhu, J. Huang. A restricted black-box adversarial framework towards attacking graph embedding models. *AAAI*, 2020.

J. Chen, Y. Wu, X. Xu, Y. Chen, H. Zheng, Q. Xuan. Fast gradient attack on network embedding. arXiv:1809.02797, 2018.

D. Chen, L. O'Bray, K. Borgwardt. Structure-aware transformer for graph representation learning. *ICML*, 2022.

Y. Chen, H. Yang, Y. Zhang, K. Ma, T. Liu, B. Han, J. Cheng. Understanding and improving graph injection attack by promoting unnoticeability. *ICLR*, 2022.

J. Chen, K. Gao, G. Li, K. He. NAGphormer: A tokenized graph transformer for node classification in large graphs. *ICLR*, 2023.

S. Cheng, Y. Dong, T. Pang, H. Su, J. Zhu. Improving black-box adversarial attacks with a transfer-based prior. *NeurIPS*, 2019.

K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser, D. Belanger, L. Colwell, A. Weller. Rethinking attention with Performers. *ICLR*, 2021.

H. Dai, H. Li, T. Tian, X. Huang, L. Wang, J. Zhu, L. Song. Adversarial attack on graph structured data. *ICML*, 2018.

V. P. Dwivedi, X. Bresson. A generalization of transformer networks to graphs. arXiv:2012.09699, 2020.

V. P. Dwivedi, A. T. Luu, T. Laurent, Y. Bengio, X. Bresson. Graph neural networks with learnable structural and positional representations. *ICLR*, 2022.

V. P. Dwivedi, C. K. Joshi, A. T. Luu, T. Laurent, Y. Bengio, X. Bresson. Benchmarking graph neural networks. *Journal of Machine Learning Research*, 24(43):1-48, 2023.

N. Entezari, S. A. Al-Sayouri, A. Darvishzadeh, E. E. Papalexakis. All you need is low (rank): Defending against adversarial attacks on graphs. *WSDM*, 2020.

P. Foth, L. Gosch, S. Geisler, L. Schwinn, S. Günnemann. Adversarial robustness of graph transformers. *Transactions on Machine Learning Research*, 2025. arXiv:2407.11764.

H. Gao, J. Ren, L. Zhao, Y. Yang, J. Chang, G. Xiao. Blackknife: Hard-label query-limited black-box attacks on heterogeneous graph neural networks. arXiv:2606.29240, 2026.

S. Geisler, T. Schmidt, H. Şirin, D. Zügner, A. Bojchevski, S. Günnemann. Robustness of graph neural networks at scale. *NeurIPS*, 2021.

L. Gosch, D. Sturm, S. Geisler, S. Günnemann. Revisiting robustness in graph machine learning. *ICLR*, 2023.

C. Guo, J. R. Gardner, Y. You, A. G. Wilson, K. Q. Weinberger. Simple black-box adversarial attacks. *ICML*, 2019.

W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, J. Leskovec. Open Graph Benchmark: Datasets for machine learning on graphs. *NeurIPS*, 2020.

A. Ilyas, L. Engstrom, A. Athalye, J. Lin. Black-box adversarial attacks with limited queries and information. *ICML*, 2018.

W. Jin, Y. Ma, X. Liu, X. Tang, S. Wang, J. Tang. Graph structure learning for robust graph neural networks. *KDD*, 2020.

W. Jin, Y. Li, H. Xu, Y. Wang, S. Ji, C. Aggarwal, J. Tang. Adversarial attacks and defenses on graphs: A review, a tool and empirical studies. *SIGKDD Explorations*, 22(2):19-34, 2021.

T. N. Kipf, M. Welling. Semi-supervised classification with graph convolutional networks. *ICLR*, 2017.

D. Kreuzer, D. Beaini, W. L. Hamilton, V. Létourneau, P. Tossou. Rethinking graph transformers with spectral attention. *NeurIPS*, 2021.

J. Li, T. Xie, L. Chen, F. Xie, X. He, Z. Zheng. Adversarial attack on large scale graph. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):82-95, 2023.

Y. Li, W. Jin, H. Xu, J. Tang. DeepRobust: A PyTorch library for adversarial attacks and defenses. arXiv:2005.06149, 2020.

J. Ma, S. Ding, Q. Mei. Towards more practical adversarial attacks on graph neural networks. *NeurIPS*, 2020.

J. Ma, J. Deng, Q. Mei. Adversarial attack on graph neural networks as an influence maximization problem. *WSDM*, 2022.

L. Ma, C. Lin, D. Lim, A. Romero-Soriano, P. K. Dokania, M. Coates, P. H. S. Torr, S.-N. Lim. Graph inductive biases in transformers without message passing. *ICML*, 2023.

J. Mu, B. Wang, Q. Li, K. Sun, M. Xu, Z. Liu. A hard label black-box adversarial attack against graph neural networks. *ACM CCS*, 2021.

F. Mujkanovic, S. Geisler, S. Günnemann, A. Bojchevski. Are defenses for graph neural networks robust? *NeurIPS*, 2022.

K. Mukherjee, Z. Alom, T. G. B. Ngo, C. G. Akcora, M. Kantarcioglu. BoCloak: Optimal transport-guided adversarial attacks on graph neural network-based bot detection. *ICML*, 2026. arXiv:2602.00318.

T. G. B. Ngo, Z. Alom, F. Errica, M. Kantarcioglu, C. G. Akcora. Adversarial graph neural network benchmarks: Towards practical and fair evaluation. arXiv:2605.05534, 2026.

N. Papernot, P. McDaniel, I. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. arXiv:1605.07277, 2016.

N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, A. Swami. Practical black-box attacks against machine learning. *AsiaCCS*, 2017.

O. Platonov, D. Kuznedelev, M. Diskin, A. Babenko, L. Prokhorenkova. A critical look at the evaluation of GNNs under heterophily: Are we really making progress? *ICLR*, 2023.

L. Rampášek, M. Galkin, V. P. Dwivedi, A. T. Luu, G. Wolf, D. Beaini. Recipe for a general, powerful, scalable graph transformer. *NeurIPS*, 2022.

B. Rozemberczki, C. Allen, R. Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2), 2021.

P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3), 2008.

H. Shirzad, A. Velingker, B. Venkatachalam, D. J. Sutherland, A. K. Sinop. Exphormer: Sparse transformers for graphs. *ICML*, 2023.

L. Sun, Y. Dou, C. Yang, K. Zhang, J. Wang, P. S. Yu, L. He, B. Li. Adversarial attack and defense on graph data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):7693-7711, 2023.

F. Tramèr, N. Carlini, W. Brendel, A. Madry. On adaptive attacks to adversarial example defenses. *NeurIPS*, 2020.

P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio. Graph attention networks. *ICLR*, 2018.

X. Wan, H. Kenlay, B. Ru, A. Blaas, M. A. Osborne, X. Dong. Adversarial attacks on graph classification via Bayesian optimisation. *NeurIPS*, 2021.

B. Wang, N. Z. Gong. Attacking graph-based classification via manipulating the graph structure. *ACM CCS*, 2019.

B. Wang, M. Lin, T. Zhou, P. Zhou, A. Li, M. Pang, H. Li, Y. Chen. Efficient, direct, and restricted black-box graph evasion attacks to any-layer graph neural networks via influence function. *WSDM*, 2024.

H. Wu, C. Wang, Y. Tyshetskiy, A. Docherty, K. Lu, L. Zhu. Adversarial examples for graph data: Deep insights into attack and defense. *IJCAI*, 2019.

Q. Wu, W. Zhao, C. Yang, H. Zhang, F. Nie, H. Jiang, Y. Bian, J. Yan. SGFormer: Simplifying and empowering transformers for large-graph representations. *NeurIPS*, 2023.

K. Xu, W. Hu, J. Leskovec, S. Jegelka. How powerful are graph neural networks? *ICLR*, 2019.

K. Xu, H. Chen, S. Liu, P.-Y. Chen, T.-W. Weng, M. Hong, X. Lin. Topology attack and defense for graph neural networks: An optimization perspective. *IJCAI*, 2019.

J. Xu, Y. Sun, X. Jiang, Y. Wang, C. Wang, J. Lu, Y. Yang. Blindfolded attackers still threatening: Strict black-box adversarial attacks on graphs. *AAAI*, 2022.

C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, T.-Y. Liu. Do transformers really perform badly for graph representation? *NeurIPS*, 2021.

X. Zhang, M. Zitnik. GNNGuard: Defending graph neural networks against adversarial attacks. *NeurIPS*, 2020.

Q. Zheng, X. Zou, Y. Dong, Y. Cen, D. Yin, J. Xu, Y. Yang, J. Tang. Graph Robustness Benchmark: Benchmarking the adversarial robustness of graph machine learning. *NeurIPS Datasets and Benchmarks*, 2021.

Y. Zhu, J. Huang, Y. Chen, R. Amor, M. Witbrock. A graph transformer defence against graph perturbation by a flexible-pass filter. *Information Fusion*, 107:102296, 2024. https://doi.org/10.1016/j.inffus.2024.102296.

D. Zügner, A. Akbarnejad, S. Günnemann. Adversarial attacks on neural networks for graph data. *KDD*, 2018.

D. Zügner, S. Günnemann. Adversarial attacks on graph neural networks via meta learning. *ICLR*, 2019.

Appendix Overview

The sections are grouped by the claim they support. Estimand and intervals: Sections L and M. Reference attacks and the query attack's distance to them: Sections P and Q. Victim-strength and architecture controls: Sections E, F, I to K and R. Further graphs: Sections B to D. The official pipeline of Foth et al. (2025) and published black-box baselines: Sections N and O. Defences: Sections G and H. Surrogate fidelity: Section S. The algorithm: Section A.

Algorithm 1 Shadow-guided few-query attack on target t with budget k, candidate size M, prior $\pi \in \{\text{random}, \text{shadow}, \text{hybrid}\}$

Require: target label y_t, shadow model f_ϕ, graph $\mathbf{A}$, allowed-edit mask Ω_t; for the query variant, victim class-score oracle $f_\theta(\cdot, \mathbf{X})_t$

1: $\mathbf{A}' \leftarrow \mathbf{A}$; $D \leftarrow \emptyset$ ▷ D: nodes already edited
2: **for** $i = 1, \ldots, k$ **do**
3: $s_j \leftarrow (1 - 2A'_{tj})\, \partial m_\phi(t; \mathbf{A}')/\partial A'_{tj}$ for all j ▷ one backward pass on the shadow
4: $C_{\text{shadow}} \leftarrow$ the M (or $M/2$ if hybrid) nodes in $\Omega_t \setminus D$ with smallest s_j
5: $C_{\text{rand}} \leftarrow M$ (or $M/2$) nodes drawn uniformly from $\Omega_t \setminus D$
6: $C \leftarrow C_{\text{shadow}}$, C_{rand}, or $C_{\text{shadow}} \cup C_{\text{rand}}$ according to π
7: $j^\star \leftarrow \arg\min_{j \in C} m_\theta\big(t; \text{flip}(\mathbf{A}', t, j)\big)$ ▷ $|C|$ queries
8: $\mathbf{A}' \leftarrow \text{flip}(\mathbf{A}', t, j^\star)$; $D \leftarrow D \cup \{j^\star\}$
9: **end for**
10: **return** $\mathbf{A}'$ ▷ total queries $\leq Mk$; zero-query variant replaces line 7 by $j^\star = \arg\min_{j \in \Omega_t \setminus D} s_j$

A The Few-Query Algorithm

Algorithm 1 gives the procedure of Section 3.

B Heterophilous Graphs

Chameleon (Rozemberczki et al., 2021) is evaluated on the standard Geom-GCN split and the filtered split of Platonov et al. (2023), with ten seeds and the fair protocol. The victim is the sparse GPS pipeline of Section C and the linearized attack is GOttack transfer from a GCN surrogate, so the deficit here is measured against one linearized attack rather than the mean of three. On the standard split (victim 50.9 percent, shadow agreement 77.7) five-flip success is 10 ± 5 for random edges, 18 ± 7 for GOttack transfer, 44 ± 10 for zero-query shadow transfer, 50 ± 9 for the 50-query hybrid and 50 ± 6 for greedy white-box search, mean $\pm$ std over seeds. The 32-point deficit is larger than any $D(v)$ in Table 15; the shadow recovers 26 of those points without a query and the hybrid reaches the white-box value. On the filtered split (victim 35.8 percent, agreement 61.8) the corresponding values are 27 ± 11, 52 ± 17, 63 ± 22, 67 ± 16 and 72 ± 12: a 20-point deficit of which the shadow recovers 11, with seed standard deviations of 11 to 22 points on this 890-node graph. A two-layer GCN victim on the same graph, attacked with a GCN shadow on its own fair targets over the same ten seeds, shows that the deficit is not specific to the transformer: on the standard split (victim 60.9 percent, agreement 79.3) the five attackers read 23 ± 7, 48 ± 10, 75 ± 13, 82 ± 10 and 85 ± 9, a 37-point deficit of the linearized attack to greedy white-box search of which the shadow recovers 26; on the filtered split (39.6 percent, agreement 66.7) they read 26 ± 13, 48 ± 13, 75 ± 17, 81 ± 19 and 83 ± 17, a 34-point deficit of which the shadow recovers 27. The linearized GCN surrogate is a poor proxy for either victim on this heterophilous graph, so the Chameleon deficits measure surrogate fidelity rather than an architecture-level Δ; the two victims draw their own targets, so no paired estimate is formed. `paper_evidence/reproduce.py` regenerates these values from the forty per-seed files.

C PUBMED AND OGBN-ARXIV DETAILS

Table 4: Pubmed, fair targets, success at 1 to 5 flips, mean ± std over seeds. Upper block: the sparse GPS victim (85.5 percent, shadow agreement 94.8, ten seeds). Lower block: a two-layer GCN victim on the same graph at nearly the same clean accuracy without intervention (85.4, five seeds), attacked with a GCN shadow on its own fair targets. Every attacker other than random edges has a five-flip mean within one point on the GCN, whereas on the transformer these attackers span 19 points; this pattern is consistent with victim-dependent surrogate mismatch rather than a graph-wide ceiling.

Method	Queries	1	2	3	4	5
Sparse GPS transformer victim, 85.5 percent, ten seeds						
Random edges	0	9 ± 6	11 ± 5	14 ± 4	16 ± 6	17 ± 7
GOttack transfer (GCN surrogate)	0	32 ± 5	41 ± 5	48 ± 8	51 ± 8	56 ± 7
Shadow transfer (ours)	0	43 ± 3	55 ± 6	63 ± 9	69 ± 9	71 ± 8
Hybrid (ours)	50	43 ± 3	57 ± 5	64 ± 8	72 ± 7	75 ± 6
White-box greedy	full access	42 ± 4	54 ± 4	62 ± 9	68 ± 8	73 ± 7
Two-layer GCN victim on the same graph, 85.4 percent, five seeds						
Random edges	0	19 ± 2	22 ± 6	23 ± 4	21 ± 5	21 ± 8
GOttack transfer (GCN surrogate)	0	44 ± 7	53 ± 4	56 ± 1	57 ± 3	58 ± 3
Shadow transfer (ours)	0	49 ± 5	54 ± 3	56 ± 3	58 ± 2	59 ± 2
Hybrid (ours)	50	51 ± 5	54 ± 3	57 ± 1	59 ± 2	59 ± 2
White-box greedy	full access	51 ± 5	55 ± 3	57 ± 1	59 ± 2	59 ± 2

A paired Pubmed control. The two blocks of Table 4 attack separately drawn targets, so they do not support a paired estimate of Δ. After the Cora and Citeseer analysis we therefore ran the common-target protocol of Table 7 on Pubmed. In each of ten seeds we draw one split and train three models on it with the recipes of Table 4: the sparse GPS victim, the two-layer GCN, and the same GCN stopped at the first epoch whose validation accuracy reaches the transformer's. All three are attacked on one list of 30 targets that all three classify correctly, with the three linearized attacks and greedy white-box search. On four seeds the GCN never reaches the transformer's validation accuracy, so there the matched GCN is the GCN itself. Clean test accuracy is 85.5, 85.4 and 85.3 percent. Against the plain GCN, the GCN-to-transformer gap averaged over the three linearized attacks is 9.2 points; under greedy white-box search it is -0.7 with $[-6.1, 4.8]$, an interval that contains zero and so shows no detected gap rather than equivalence. Table 5 gives Δ. The two Pubmed comparisons were added after the family of Table 15 was fixed, so their intervals are unadjusted and outside the multiplicity adjustment of Section L.

Table 5: Paired Pubmed control: one common list of 30 targets per seed, ten seeds, success at five flips. D is greedy white-box success minus the mean of the three linearized attacks and $\Delta = D(v) - D(b)$, paired by seed; intervals are two-sided 95 percent t intervals. Entries are rounded independently, so $D(v) - D(b)$ can differ from Δ by 0.1. `paper_evidence/reproduce.py` regenerates each value from raw files.

Baseline	$D(v)$	$D(b)$	Δ	95% CI
GCN	10.7	0.8	9.9	$[5.3, 14.5]$
Accuracy-matched GCN	10.7	1.2	9.4	$[5.4, 13.5]$

Table 6: ogbn-arxiv: the sparse GPS victim (69.9 percent, shadow agreement 89.7) and a two-layer GCN victim on the same graph (70.6, agreement 92.8), each on its own fair targets; success at 1 to 5 flips, mean ± std over five seeds. The linearized surrogate is a linear GCN used as a first-order proposal generator for both victims, Nettack's exact scores being infeasible at this scale. Frozen-encoding and recomputed-encoding success agree on every target of every attack. Because the victim target cohorts differ, no cross-victim Δ is reported.

		GPS victim (sparse attention)					GCN victim				
Method	Queries	1	2	3	4	5	1	2	3	4	5
Random edges	0	12 ± 4	13 ± 2	15 ± 3	20 ± 7	21 ± 6	17 ± 4	22 ± 6	25 ± 10	25 ± 8	27 ± 10
Linearized GCN surrogate, all candidates (first-order)	0	49 ± 6	54 ± 5	56 ± 4	59 ± 4	63 ± 1	50 ± 7	53 ± 9	56 ± 6	58 ± 5	60 ± 5
Shadow transfer (ours)	0	49 ± 3	55 ± 4	57 ± 3	59 ± 1	62 ± 3	53 ± 7	57 ± 5	59 ± 4	59 ± 4	61 ± 6
Hybrid (ours)	50	51 ± 1	55 ± 4	58 ± 4	62 ± 4	65 ± 4	53 ± 7	58 ± 6	59 ± 4	60 ± 5	61 ± 6
White-box greedy	full access	50 ± 4	54 ± 3	57 ± 3	61 ± 4	64 ± 3	53 ± 7	57 ± 7	60 ± 3	61 ± 4	62 ± 6

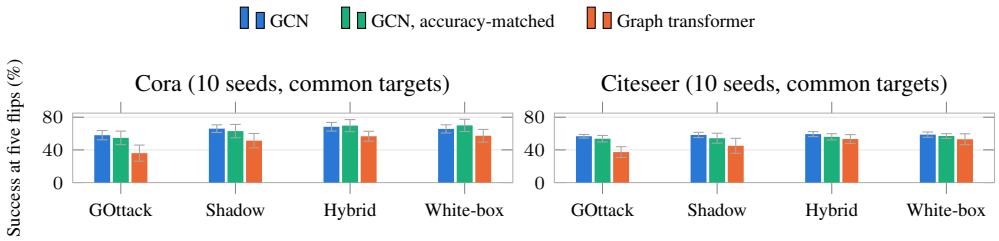


Figure 3: Attack success at five flips on shared targets, reported as the mean and sample standard deviation over seeds (Table 7). From the GCN victims to the transformer, GOttack loses 16 to 22 points, shadow transfer without queries loses 9 to 15, and the 50-query hybrid and full-access search lose 3 to 13.

D CLUSTER DETAILS

CLUSTER consists of 6-cluster SBM graphs from the official generator, attacked on fair targets of unseen test graphs, 30 targets per seed, ten seeds; seeds 0 to 2 were regenerated from scratch after the original files were lost, seeds 3 and 4 are the archived files, seeds 5 to 9 were run later, and `paper_evidence/reproduce.py` regenerates the values from all twenty per-seed files. On the GraphGPS-style victim (66.4 percent clean, shadow agreement 86.1) success at five flips is 17 for random edges, 53 for the linearized-GCN surrogate, 66 for the zero-query trained shadow, 67 for the 50-query hybrid and 66 to 67 for the three white-box variants, so transfer is numerically within one point of full access here. On the Graphormer-style victim (shortest-path attention bias and degree encoding, no message passing, 59.4 percent, agreement 69.2) the same five read 20, 59, 59, 69, and 69 to 71: here the linearized surrogate is 11 points behind greedy white-box search and the zero-query shadow gains only 0.3 points over it (59.0 against 58.7), so the 50 queries do the recovering; the hybrid reaches the greedy value, 2.0 points below the strongest variant. Over the first five seeds the shadow had recovered about half of the deficit (62 against 55 and 70); seeds 5 to 9 carry the weakest Graphormer victims (50.7 to 60.0 percent clean, agreement 58.8 to 76.9) and on those seeds the shadow falls below the linearized surrogate. A shadow of the other family, at seed 0, reduces success by 6.7 points at five flips on the GraphGPS-style victim (53 against 60) and leaves it unchanged on the other (63 against 63); with 50 queries the two shadows give the same value on both victims, which is descriptive and not an equivalence claim. These two cross-family files were regenerated alongside the same-family seeds and are also covered by the checker. No test node has GOttack's peripheral orbit signature, so it reduces to its linear surrogate over all candidates. The per-budget table is in the supplementary package.

Table 7: Common-target control with an accuracy-matched GCN: for each seed the transformer, a GCN, and the same GCN stopped at the transformer's validation accuracy are attacked on the identical 30 targets, drawn from nodes all three classify correctly and ranked by mean within-model margin percentile (ten seeds; clean accuracy GCN 83.6 and 75.0, GCN_m 74.6 and 69.3, GT 71.9 and 67.7). Success at five flips, mean $\pm$ sample standard deviation; p: paired t over seeds, GT against GCN / GT against GCN_m. The last two rows are the PRBCD-style attack with the encoding frozen and, on the transformer, differentiated through it.

	Cora				Citeseer			
Attack	GCN	GCN_m	GT	p	GCN	GCN_m	GT	p
Random edge additions	22 ± 4	29 ± 5	13 ± 4	<0.001 / <0.001	20 ± 7	29 ± 6	15 ± 6	0.08 / <0.001
GOttack (linearized GCN, periphery)	58 ± 6	55 ± 8	36 ± 10	<0.001 / <0.001	57 ± 2	54 ± 4	37 ± 7	<0.001 / <0.001
Linearized GCN, all candidates	59 ± 4	57 ± 10	35 ± 8	<0.001 / <0.001	58 ± 3	55 ± 3	35 ± 6	<0.001 / <0.001
Nettack (linearized GCN, direct)	60 ± 5	58 ± 10	38 ± 10	<0.001 / <0.001	58 ± 3	54 ± 4	37 ± 7	<0.001 / <0.001
Trained shadow, same family	66 ± 5	63 ± 8	51 ± 9	<0.001 / 0.008	58 ± 3	54 ± 6	45 ± 9	<0.001 / 0.006
Trained GCN shadow on the GT	n/a	n/a	52 ± 6		n/a	n/a	50 ± 6	
Hybrid, 50 queries, same-family shadow	68 ± 5	70 ± 7	57 ± 6	<0.001 / 0.001	59 ± 3	56 ± 4	53 ± 5	0.005 / 0.15
Hybrid, 50 queries, GCN shadow on the GT	n/a	n/a	56 ± 8		n/a	n/a	53 ± 4	
White-box greedy	66 ± 5	70 ± 8	57 ± 8	0.01 / 0.008	59 ± 3	57 ± 3	53 ± 7	0.03 / 0.13
Relaxed (PRBCD-style), encoding frozen	67 ± 6	71 ± 7	59 ± 8	0.01 / 0.005	59 ± 3	57 ± 3	56 ± 6	0.12 / 0.54
Relaxed, through the encoding (adaptive)	67 ± 6	71 ± 7	60 ± 9	0.04 / 0.01	59 ± 3	57 ± 3	55 ± 6	0.06 / 0.32

Table 8: Common-target control of Table 7 repeated with tuned transformers (ten seeds; clean accuracy GCN 83.6 and 75.0, GCN_m 78.7 and 72.2, GT 76.5 and 70.4; mean target degree 16.7 on Cora, 11.8 on Citeseer). Success at five flips, mean $\pm$ std over seeds; p: paired t-test over seeds, GT against GCN / GT against GCN_m.

	Cora				Citeseer			
Attack	GCN	GCN_m	GT	p	GCN	GCN_m	GT	p
Random edge additions	26 ± 8	31 ± 7	16 ± 5	0.002 / <0.001	23 ± 5	28 ± 5	13 ± 6	0.007 / <0.001
GOttack (linearized GCN, periphery)	61 ± 4	61 ± 5	54 ± 4	0.001 / 0.004	58 ± 5	57 ± 6	33 ± 13	<0.001 / <0.001
Linearized GCN, all candidates	60 ± 5	63 ± 6	53 ± 5	<0.001 / <0.001	58 ± 5	58 ± 5	34 ± 10	<0.001 / <0.001
Nettack (linearized GCN, direct)	61 ± 5	64 ± 6	56 ± 5	<0.001 / <0.001	58 ± 5	58 ± 5	38 ± 13	<0.001 / <0.001
Trained shadow, same family	65 ± 7	67 ± 7	67 ± 7	0.56 / 0.92	59 ± 5	59 ± 5	42 ± 11	0.002 / <0.001
Trained GCN shadow on the GT	n/a	n/a	65 ± 6		n/a	n/a	44 ± 13	
Hybrid, 50 queries, same-family shadow	67 ± 5	71 ± 8	71 ± 7	0.10 / 0.84	62 ± 5	60 ± 4	46 ± 9	<0.001 / <0.001
Hybrid, 50 queries, GCN shadow on the GT	n/a	n/a	67 ± 6		n/a	n/a	49 ± 11	
White-box greedy	66 ± 7	72 ± 8	71 ± 6	0.04 / 0.50	60 ± 4	60 ± 4	48 ± 10	0.003 / 0.003

E Common Targets with Tuned Transformers

The main victims reach 71.9 and 67.7 percent (Section 5). To test whether Table 7 depends on that, we swept 72 configurations of the same architecture over width, depth, dropout, learning rate and weight decay, three seeds each, keeping the configuration with the highest mean validation accuracy among those with no diverged seed. Table 8 repeats the common-target control with those victims over seeds 0 to 9 (76.5 and 70.4 percent), new matched GCNs (78.7 and 72.2) and new target lists.

The residual gap moves. On Cora the tuned transformer's retained white-box, shadow, and hybrid point gaps relative to the two GCNs range from -4 to $+2$: the white-box gaps are -4 and $+2$, the shadow gaps -2 and 0, and the hybrid gaps -4 and -1, while the linearized surrogate still trails the GCN columns by 5 to 11 points. On Citeseer the white-box gaps are 12 and 12, the shadow gaps 16 and 17, the hybrid gaps 16 and 15, and the linearized gaps 21 to 25. These direct means show that retuning the victim changes the residual comparison, and the paired Δ intervals for these victims regenerate from the raw per-seed files (Table 15): 10.7 $[7.0, 14.3]$ and 6.8 $[2.5, 11.1]$ on Cora against the two baselines, 11.2 $[4.2, 18.3]$ and 10.6 $[4.0, 17.1]$ on Citeseer. What retuning moves is the residual robustness gap, and it moves in a graph-dependent way rather than vanishing: on Cora no same-family trained or querying attacker separates the tuned transformer from either GCN (p from 0.10 to 0.92), while on Citeseer white-box search still leaves it 12 points harder than both ($p = 0.003$).

F The Official GraphGPS Layer

The main victims are our own implementation of the GraphGPS recipe (Section 4), so we repeated the fair grid rows (Table 9) and the common-target control with the layer as released by its authors in PyTorch Geometric, `GPSConv`. Its local block is a GCN propagation with the dense normalized adjacency, so every attack of Section 3 applies unchanged. A 72-configuration sweep chose four layers on Cora and two on Citeseer; retrained over seeds 0 to 9 the victims reach 74.4 and 67.1 percent against 71.9 and 67.7 for the main victims, with shadow agreement 80.0 and 75.5.

Two things change. First, the official layer is easier to attack than the GCN in the direct means on common targets: greedy white-box success is 70 percent on Cora against 64 and 69 for the two GCNs, and 64 on Citeseer against 58 and 55. Second, the surrogate and white-box comparisons can point in opposite directions on Citeseer: the linearized surrogate reports the transformer 4 to 13 points harder than the GCNs while white-box search finds it 6 to 9 points easier (two of the three sign reversals counted in Section 5). On Cora the corresponding point comparison is smaller. The paired estimand on the common target list is in Table 15: on Citeseer the official layer carries an excess of 16.0 points over the GCN and 14.1 over the accuracy-matched GCN, both intervals excluding zero; on Cora the excess is 4.7 and -1.3 and neither interval excludes zero. With the per-step oracle of Section 7 on both arms, ten seeds, the Cora cells read 0.7 with $[-4.8, 6.2]$ and -3.7 with $[-9.5, 2.1]$, so no excess is detected for the official layer on Cora under either reference. On the fair rows the hybrid reaches 63 and 61 against greedy white-box success of 66 and 66, and differentiating through the encoding moves the results by at most one point.

Table 9: Two further victims on their own fair targets, success at five flips, mean ± std over seeds; the counterpart of the GT columns of Table 14. *Official GPS*: the released `GPSConv` layer, ten seeds. *GRIT-style*: no message-passing block, RRWP pair representations bias the attention, ten seeds. Same-family shadow means a shadow of the victim's own kind. Hybrid uses 50 queries. n/a: not run, the PRBCD-style attacks costing about four hours per seed on the dense pair tensor.

	Official GPS layer		GRIT-style victim	
Attack	Cora	Citeseer	Cora	Citeseer
Random edge additions	26 ± 5	20 ± 5	10 ± 4	9 ± 4
GOttack: linearized GCN, periphery candidates	56 ± 5	49 ± 8	13 ± 5	14 ± 4
Linearized GCN, all candidates	58 ± 6	48 ± 10	20 ± 4	20 ± 8
Nettack (linearized GCN, direct)	58 ± 5	50 ± 9	15 ± 5	15 ± 8
Trained GCN shadow, all candidates	58 ± 5	52 ± 7	13 ± 6	15 ± 6
Same-family shadow, all candidates	61 ± 4	58 ± 7	15 ± 7	13 ± 4
Hybrid, GCN-shadow proposals	61 ± 5	58 ± 8	24 ± 7	24 ± 5
Hybrid, same-family-shadow proposals	63 ± 3	61 ± 7	28 ± 8	26 ± 7
White-box greedy, encoding frozen	66 ± 6	66 ± 7	16 ± 7	13 ± 4
White-box greedy, gradient through the encoding	66 ± 6	67 ± 7	22 ± 6	26 ± 3
White-box PRBCD-style relaxed, encoding frozen	67 ± 6	67 ± 7	n/a	n/a
White-box PRBCD-style relaxed, through the encoding	67 ± 6	67 ± 6	n/a	n/a

G Adversarially Trained Victims

Foth et al. (2025) report that adversarial training with their attack substantially improves the robustness of a graph transformer (Graphormer). We adversarially trained the GPS-style transformer and the GCN of Table 14 with a PRBCD-style inner attack: after 20 clean epochs, every epoch computes a global structure perturbation of the training loss at a budget of 5 or 2 percent of the edges and trains on the clean and the perturbed graph for 180 further epochs, keeping the epoch with the best mean of clean and perturbed validation accuracy (the released script has the full recipe). The attacker's shadow is the standard one, targets follow the fair protocol, ten seeds. Adversarial training costs the transformer 3.6 and 1.6 points of clean accuracy at the 5 percent budget (68.3 and 66.1 percent; 69.7 and 66.8 at 2 percent) and moves the GCN means by at most 0.5 point relative to the standard victim (83.8 and 75.2; 84.1 and 75.0).

The results show two main patterns (Table 10). First, adversarial training produces much larger attack-success reductions on the transformer than on the GCN. At the 5 percent budget the transformer's white-box success falls from 64 to 41 on Cora and 60 to 34 on Citeseer, the 50-query hybrid from 64 to 45 and 59 to 36, and the zero-query shadow from 61 to 34 and 51 to 28, while the GCN's numbers move by at most 5 points. Across the tested attacks, the remaining GCN-to-transformer point gaps are 21 and 26 points under white-box search and 16 and 24 under the hybrid. The surrogate's exaggeration shrinks accordingly, to 7 to 9 points on Cora ($p \leq 0.012$) and 4 to 8 on Citeseer, against 14 to 17 and 14 to 18 on the standard victims of the same seeds; at the 2 percent budget the transformer is protected less and the exaggeration is back at 10 to 15 points. Second, on the adversarially trained transformer the 50-query hybrid's point estimates are above greedy white-box search in all four cells (45 against 41 and 36 against 34 at 5 percent, 52 against 45 and 47 against 44 at 2 percent) and within one point of or above the PRBCD-style point estimates. These comparisons do not establish superiority; they indicate that the available first-order references may not be ceilings after adversarial training and motivate reporting a query-based stress test alongside them.

G.1 Does the defence protect, or does it change the targets?

Every defended number so far is measured on targets drawn among the nodes that condition's deployed system classifies correctly, so a defence that changes which nodes are correct also changes the target list, and a cross-condition difference mixes protection with that change. This control removes it. For each seed we draw one stratum of 30 targets from the nodes classified correctly by the undefended victim, the adversarially trained victim and the filtered victim at once, rank them by mean within-model margin percentile, and attack that one list under all conditions; 1211 nodes qualify on Cora at seed 0. Ten seeds, five flips, Table 11.

Table 10: Adversarially trained victims (PRBCD-style inner attack, budget 5 or 2 percent of the edges), fair targets, success at five flips, mean ± std over ten seeds, standard shadow. Clean accuracy at 5 percent: GT 68.3 and 66.1, GCN 83.8 and 75.2; at 2 percent: GT 69.7 and 66.8, GCN 84.1 and 75.0. Standard victims in Table 14.

	Budget 5 percent				Budget 2 percent			
	Cora		Citeseer		Cora		Citeseer	
Attack	GT	GCN	GT	GCN	GT	GCN	GT	GCN
Random edge additions	15 ± 5	28 ± 7	14 ± 4	24 ± 3	12 ± 3	27 ± 4	12 ± 7	25 ± 4
GOttack: linearized GCN, periphery candidates	31 ± 11	60 ± 2	25 ± 11	59 ± 3	30 ± 9	60 ± 3	28 ± 13	57 ± 4
Linearized GCN, all candidates	30 ± 8	60 ± 4	30 ± 11	59 ± 2	29 ± 9	61 ± 3	32 ± 13	58 ± 2
Nettack (linearized GCN, direct)	32 ± 9	59 ± 4	29 ± 13	59 ± 2	32 ± 11	60 ± 2	33 ± 16	58 ± 3
Trained shadow, same family, all candidates	34 ± 15	61 ± 2	28 ± 12	60 ± 2	43 ± 10	62 ± 3	39 ± 15	58 ± 2
Hybrid, 50 queries	45 ± 11	62 ± 2	36 ± 12	60 ± 2	52 ± 9	62 ± 2	47 ± 14	59 ± 2
White-box greedy gradient on the victim	41 ± 14	62 ± 2	34 ± 14	59 ± 1	45 ± 10	62 ± 2	44 ± 15	58 ± 2
White-box PRBCD-style relaxed	44 ± 12	62 ± 2	36 ± 12	60 ± 2	48 ± 11	62 ± 2	47 ± 16	58 ± 2

The target-set effect is a few points: on identical targets the 50-query attacker goes from 64.3 undefended to 48.0 filtered on Cora and from 58.7 to 56.0 on Citeseer, against 64 to 52 and 59 to 54 on the per-condition target lists. What the control does indicate is that the two defences protect against different attackers. Adversarial training takes nearly the same amount from every informed attacker, -22.7 from zero-query transfer, -19.3 from the 50-query hybrid and -22.7 from white-box search on Cora and -23.0, -23.3 and -24.7 on Citeseer (paired t over seeds, $p \leq 0.002$ throughout), leaving the ordering of attackers where it was. The filter pulls them apart: on Cora it takes -32.3 from transfer and -34.7 from white-box search but -16.3 from the query attacker, and on Citeseer -9.0 and -10.0 against -2.7, which ten seeds put at the edge of resolution ($p = 0.053$). Random-addition point changes are small under either defence (-0.3 to -2.7); the nonsignificant tests do not establish zero change, but the contrast with the 9 to 35 point reductions of shadow transfer and white-box search makes a target-strata explanation less plausible.

The hidden-filter condition suggests which part of the filter's protection the queries recover. When the 50 queries are answered by the unfiltered model while deployment still filters, the query attacker's loss becomes -31.7 on Cora and -9.0 on Citeseer, within 0.7 points and 0.0 of the zero-query transfer loss of -32.3 and -9.0 on the same targets. We do not treat that numerical closeness as equivalence. The pattern is consistent with queries recovering transferability that the filter destroys, but it does not isolate a unique causal mechanism; the attacker is never told a defence is present. One consequence is visible in the filter columns: a gradient attacker that holds the victim's weights but takes them through the unfiltered forward pass reaches 28.7 and 47.7 where our 50-query attacker reaches 48.0 and 56.0 on the identical targets, because the gradient ignores the filter and so describes a model that is not deployed. A gradient attacker told about the filter and restricted to edges it keeps does better than both (Table 11), and it is one of two comparisons in which the retained filter cost is small in point estimate: against the same gradient attacker on the undefended victim it loses 6.3 points on Cora ($p = 0.0002$) and 0.0 on Citeseer on average over the ten seeds. The zero mean is not evidence of zero cost.

The second is cheaper still and it is the sharpest version of the paper's point. Give the zero-query transfer attacker nothing but the knowledge that a Jaccard filter is deployed and at what threshold, and let it propose only edges the filter keeps. It spends no victim queries, holds no victim weights and takes no victim gradients. On the same target list it reaches 50.7 ± 13.6 on Cora and 51.7 ± 10.1 on Citeseer, against 26.0 and 42.3 for the same attacker left blind, a gain of 24.7 and 9.3 points from that knowledge (paired $p = 0.0001$ and 0.0004). Measured against itself on the undefended victim, the filter costs it 7.7 points on Cora ($p = 0.004$) and -0.3 on Citeseer, which ten seeds do not separate from zero ($p = 0.80$); that is a failure to resolve a cost, not a demonstration that there is none. So of the 32.3 and 9.0 points the filter appears to take from transfer, roughly three quarters on Cora and, on Citeseer, as much as ten seeds can resolve, is the attacker not modelling a defence it was never told about. Ten seeds do not show either correction dominating the other. On Cora the aware zero-query attacker reads 50.7 against the blind 50-query hybrid's 48.0, which ten seeds do not separate ($p = 0.48$); on Citeseer the hybrid is ahead, 56.0 against 51.7 ($p = 0.013$). They are different capabilities, not the same one twice: queries need access to the deployed system and no knowledge of it, awareness needs knowledge and no access. Adding gradients on top of awareness buys 6.3 and 6.0 more points ($p = 0.10$ and 0.045), so most of what the defence-aware gradient attacker achieves is available without victim gradients.

This attacker is weaker than it could be and we leave it that way: it constrains only its own proposals and does not model what the filter does to the rest of the graph, so the numbers likely understate what awareness is worth.

Fig. 2 sweeps the filter's own threshold, and the pattern is the one Athalye et al. (2018) describe. Raising the Jaccard threshold from 0.01 to 0.03 to 0.05 takes blind white-box search from 34.0 to 14.3 on Cora and 50.7 to 13.3 on Citeseer and the blind zero-query shadow from 28.3 to 12.3 and 41.0 to 11.7, so a defence-blind evaluation reports a defence that keeps getting stronger, while the aware attackers stay between 47 and 60 throughout. This is consistent with gradient masking by a preprocessing step the gradient does not see, rather than by a saturating activation. The tightening is not free: deployed clean accuracy falls from 70.9 to 67.5 on Cora and 67.3 to 62.5 on Citeseer against 71.9 and 67.7 undefended. We report it as a diagnostic and not as a recommended operating point: nothing here says which threshold a defender should choose, only that a blind attacker will misprice every one.

Table 11: Common-target defence control: one target list per seed, drawn from nodes the undefended, adversarially trained and filtered victims all classify correctly and ranked by mean within-model margin percentile, attacked under every condition. GPS-style victim, success at five flips, mean $\pm$ std over ten seeds. Clean accuracy on all test nodes: Cora 71.9, 68.3, 70.9; Citeseer 67.7, 66.1, 67.3. *adv*: PRBCD-style adversarial training at a 5 percent budget; *filter*: GCN-Jaccard at inference. The hidden row answers the attacker's 50 queries with the unfiltered model while deployment still filters; the two *filter-aware* rows restrict an attacker to edges the filter keeps, one a zero-query trained shadow and one greedy gradient, which is the strongest attacker we have against it. Filter awareness means knowing only that a Jaccard filter is deployed and at what threshold; the zero-query filter-aware attacker never holds the victim's weights or evaluates the victim. Differences quoted in the text are paired seed by seed.

	Cora			Citeseer		
Attack	undefended	adv	filter	undefended	adv	filter
Random edge additions	12.0 ± 5.0	10.3 ± 4.3	11.7 ± 4.8	13.3 ± 5.7	10.7 ± 5.2	12.0 ± 4.8
GOttack (linearized GCN)	38.0 ± 11.5	26.7 ± 11.7	30.0 ± 8.7	39.0 ± 9.9	22.0 ± 10.7	31.0 ± 11.6
Trained shadow, zero queries	58.3 ± 12.4	35.7 ± 15.2	26.0 ± 6.8	51.3 ± 9.6	28.3 ± 10.0	42.3 ± 9.9
Trained shadow, zero queries, filter-aware			50.7 ± 13.6			51.7 ± 10.1
Hybrid, 50 queries	64.3 ± 10.4	45.0 ± 12.2	48.0 ± 9.1	58.7 ± 8.5	35.3 ± 11.1	56.0 ± 7.7
Hybrid, 50 queries, filter hidden			32.7 ± 8.6			49.7 ± 9.5
White-box greedy gradient	63.3 ± 7.7	40.7 ± 14.3	28.7 ± 9.1	57.7 ± 10.3	33.0 ± 11.2	47.7 ± 7.9
White-box, defence-aware			57.0 ± 6.9			57.7 ± 8.3

H Threshold Mismatch and Two Further Defences

The filter-aware attacker in the main text is told the deployed Jaccard threshold. This control removes that. The filter is deployed at 0.05 while the attacker assumes 0.01, 0.03 or 0.10, on both citation graphs, ten seeds, GPS-style victim, fair targets, success at five flips. Paired differences are taken seed by seed against the matched-threshold attacker, with the sample SD at `ddof=1` and a two-sided 95 percent t interval on nine degrees of freedom (Table 12).

All six intervals exclude zero, so the threshold matters on both graphs. What the graphs disagree about is how much. On Cora every wrong guess still lands far above blind transfer, 28.0 at worst against 12.3, so much of what awareness buys survives a wrong threshold, though less than half at the worst guess. On Citeseer an attacker assuming 0.01 against a filter at 0.05 reads 16.3 against blind 11.7, which is close to worthless. The direction of the error also behaves differently: on Cora the overestimate is the best of the three wrong guesses and on Citeseer it is not, and ten seeds do not support reading anything into that. The claim we make is therefore the narrow one, that an attacker who knows the defence is much stronger than one who does not, and not that a rough guess suffices.

Two other preprocessing defences. The same harness was run with rank-15 low-rank reconstruction of the adjacency (Entezari et al., 2020), thresholded back to a binary graph, and with random edge dropping at rate 0.1 averaged over five draws at inference, on both victims and both graphs, ten seeds, own fair targets, without a defence-aware attacker. Low-rank reconstruction holds greedy white-box search at or below random additions at five flips: on the GPS-style victim 4.7 and 1.0 percent on Cora and Citeseer, with the hybrid at 5.3 and 9.3 and random additions at 4.7 and 3.7, against 63.7, 64.0

Table 12: Filter deployed at Jaccard 0.05; the attacker assumes the threshold in the first column. Success at five flips, ten seeds. The paired column compares the filter-aware zero-query attacker against the same attacker told the true threshold.

	Cora		Citeseer	
Attacker assumes	success	paired vs matched	success	paired vs matched
blind, no filter knowledge	12.3		11.7	
0.05, the true threshold	47.3	reference	48.3	reference
0.01	28.0	-19.3 $[-23.4, -15.3]$	16.3	-32.0 $[-37.8, -26.3]$
0.03	35.0	-12.3 $[-17.4, -7.3]$	29.0	-19.3 $[-28.0, -10.7]$
0.10	37.0	-10.3 $[-16.0, -4.7]$	22.0	-26.3 $[-33.3, -19.4]$

and 15.0 on Cora and 60.3, 59.0 and 18.3 on Citeseer undefended; the GCN victims read 4.7 and 1.0 for white-box search and 9.7 and 21.0 for the hybrid. It costs the transformer 8.3 and 12.1 points of clean accuracy (63.6 and 55.6 percent) and the GCN 9.3 and 2.5 (74.3 and 72.5). With white-box search at or below random additions the decomposition has little to separate; on Citeseer, however, the 50-query hybrid reaches 9.3 and 21.0 against 1.0 for white-box search, so we do not claim that this defence removes the structure the attacks need. Random edge dropping costs the GPS-style victim 0.2 points of accuracy on each graph and the attackers little: shadow transfer 55.3 and 51.3, the hybrid 57.7 and 56.3 and white-box search 61.3 and 59.7, so the randomized scores it returns to the 50 queries still leave the hybrid above the shadow on both graphs; the GCN victims read 66.3 and 58.3 for shadow transfer and 66.0 and 58.0 for the hybrid. Neither defence was run with a defence-aware attacker, so neither enters Table 3; both are regenerated by `paper_evidence/reproduce.py`. The filter of Zhu et al. (2024) was not run.

I A GRIT-STYLE VICTIM WITHOUT MESSAGE PASSING

Our GRIT-style control (Ma et al., 2023) has no local message-passing block: edges enter through relative random-walk pair encodings and a degree scaler. Over ten seeds it reaches 63.0 and 62.9 percent clean accuracy on Cora and Citeseer, below the GCN controls. On its own fair targets (Table 9) no attack exceeds 28 percent at five flips: encoding-frozen white-box search reaches 16 and 13, through-encoding white-box search 22 and 26, the 50-query hybrid 28 and 26, and zero-query shadow transfer 15 and 13. On common targets the GCN controls are flipped on 52 to 72 percent of targets while this victim remains at 7 to 25. Relative to the through-encoding reference (Table 15) the surrogate excess is positive and separated from zero on Citeseer, 8.2 with $[2.0, 14.4]$ against the plain GCN and 7.9 with $[1.9, 13.9]$ against the matched GCN, roughly half the GPS-style values on that graph, but not on Cora, where it is 2.9 with $[-2.0, 7.8]$ against the plain GCN and -6.7 with $[-13.2, -0.2]$ against the matched GCN: there the point estimate says the linearized surrogate fits this victim better than the matched GCN, although that cell does not survive the Benjamini-Hochberg correction (Section L). This isolates a boundary of the main result: removing local message passing reduces both clean accuracy and edge-attack success, the excess on this victim depends on the graph, and none of it establishes that attention itself causes robustness.

J GAT AND GIN VICTIMS

We add GAT (Veličković et al., 2018) and GIN (Xu et al., 2019a) victims to test whether the failure is specific to transformers. Each uses a same-family trained shadow and ten fair-target seeds (Table 13). The best of the three linearized-GCN attacks falls 7.7 and 5.7 points short of greedy white-box search on the Cora and Citeseer GAT victims, and 8.7 and 10.7 on the GIN victims. For comparison, the GCN deficits are 6.0 and 1.0 over the same ten seeds; the GPS-style transformer deficits are 19.7 and 14.7. Paired over the ten seeds, the 95 percent t intervals of the deficits are $[4.7, 12.6]$ and $[5.1, 16.3]$ for GIN, $[1.3, 14.0]$ and $[-1.5, 12.9]$ for GAT, $[1.8, 10.2]$ and $[-0.2, 2.2]$ for the GCN, and $[15.9, 23.5]$ and $[5.7, 23.6]$ for the GPS-style transformer (Cora then Citeseer). The Citeseer GAT and GCN intervals contain zero, which establishes neither equivalence nor a difference there; the other six exclude it. The 50-query gain over same-family shadow transfer ranges from 2.7 to 7.3 points across the four GAT/GIN conditions.

Table 13: GAT and GIN victims on their own fair targets: mean success at five flips over ten seeds. Deficit is greedy white-box minus the best linearized-GCN attack, computed per seed and then averaged. The interval is a two-sided 95 percent t interval on the ten paired per-seed deficits (nine degrees of freedom).

	GAT victim		GIN victim	
Attack	Cora	Citeseer	Cora	Citeseer
Clean accuracy	75.4	72.4	80.1	69.3
Shadow agreement	72.0	86.9	90.1	86.4
Random edge additions	32	28	25	25
GOttack: linearized GCN, periphery candidates	55	54	54	55
Linearized GCN, all candidates	59	62	57	59
Nettack (linearized GCN, direct)	64	62	58	59
Trained same-family shadow, no queries	67	64	65	65
Hybrid, 50 queries	74	67	68	68
White-box greedy gradient	72	68	66	70
Deficit to white-box	7.7	5.7	8.7	10.7
95 percent t interval	$[1.3, 14.0]$	$[-1.5, 12.9]$	$[4.7, 12.6]$	$[5.1, 16.3]$

K The Surrogate-by-Victim Grid

Table 14 is the full grid discussed in Section 5: every surrogate against every victim on each victim's own fair targets, ten seeds.

Table 14: Surrogate-by-victim grid: success (percent of targets flipped) at five edge flips under the fair-evaluation protocol, mean $\pm$ std over ten seeds (30 targets per seed); text differences use unrounded means. GCN victims: 83.6 and 75.0 percent clean accuracy; GT victims: 71.9 and 67.7. n/a: not run.

	Cora		Citeseer	
Attack	GCN victim	GT victim	GCN victim	GT victim
Random edge additions	29 ± 7	15 ± 2	27 ± 5	18 ± 7
GOttack: linearized GCN, periphery candidates	58 ± 4	42 ± 8	58 ± 2	41 ± 10
Linearized GCN, all candidates	59 ± 4	40 ± 7	58 ± 3	45 ± 9
Nettack (linearized GCN, direct)	59 ± 4	44 ± 7	58 ± 3	46 ± 9
Trained GCN shadow, all candidates	64 ± 3	59 ± 6	58 ± 2	54 ± 9
Trained GCN shadow, periphery candidates	61 ± 4	n/a	59 ± 2	n/a
GT shadow, all candidates	60 ± 4	61 ± 10	54 ± 5	51 ± 10
GT shadow, periphery candidates	n/a	51 ± 8	n/a	50 ± 11
Hybrid, GCN-shadow proposals, 50 queries	67 ± 4	63 ± 6	59 ± 2	61 ± 10
Hybrid, GT-shadow proposals, 50 queries	63 ± 4	64 ± 8	57 ± 3	59 ± 10
White-box greedy gradient on the victim	65 ± 5	64 ± 7	59 ± 2	60 ± 11
White-box PRBCD-style relaxed	n/a	64 ± 7	n/a	64 ± 10

L The Surrogate-Excess Estimand on Common Targets

Eq. (1) defines a difference of two surrogate deficits relative to a stated baseline and reference. For paired seed s, we compute $\Delta_s = [S_{\mathrm{ref},s}(v) - \overline{S}_{\mathrm{lin},s}(v)] - [S_{\mathrm{ref},s}(b) - \overline{S}_{\mathrm{lin},s}(b)]$ on the identical target set. Table 2 reports $\bar{\Delta}$, the sample SD with `ddof=1`, and $\bar{\Delta} \pm t_{0.975,9}\,\mathrm{SD}(\Delta_s)/\sqrt{10}$. All four comparisons have ten valid paired seeds. Table 2 is the headline subset. Table 15 carries every Cora and Citeseer condition that has both arms on one common target list under a greedy reference, and it is the unit of analysis: eight victim-graph cells, each read against two GCN baselines, so the sixteen rows are not sixteen independent conditions. Against a plain GCN Δ is positive in all eight and separated from zero in six; against the accuracy-matched GCN, it is positive in six and separated above zero in five. Matching lowers Δ in all eight. Treating the sixteen rows as one family of paired t tests, a Benjamini-Hochberg correction keeps all eleven positive separations at a false discovery rate of 0.05 (largest adjusted $p = 0.023$), and the negative GRIT-style cell does not survive it (adjusted $p = 0.061$). The stricter Holm correction keeps six: the GPS-style victim against both baselines on

Table 15: Every Cora and Citeseer condition carrying both arms on one common target list under a greedy reference, regenerated from the raw per-seed files by `delta_all.py`. $D(v)$ and $D(b)$ are the surrogate deficits of victim and baseline under the stated reference (greedy white-box, or the through-encoding attack for the GRIT-style victim), $\Delta = D(v) - D(b)$ paired by seed, SD is the sample standard deviation with Bessel's correction, and the interval is a 95 percent t interval. Entries are rounded independently, so $D(v) - D(b)$ can differ from Δ by 0.1. An interval containing zero is not evidence of equivalence.

Condition	n	$D(v)$	$D(b)$	Δ	SD	95% CI
Cora, GPS-style vs GCN	10	21.1	6.9	14.2	7.1	$[9.2, 19.3]$
Cora, GPS-style vs matched GCN	10	21.1	13.4	7.7	11.3	$[-0.4, 15.7]$
Cora, tuned GPS vs GCN	10	16.3	5.7	10.7	5.1	$[7.0, 14.3]$
Cora, tuned GPS vs matched GCN	10	16.3	9.6	6.8	6.0	$[2.5, 11.1]$
Cora, official GPSConv vs GCN	10	12.9	8.2	4.7	9.8	$[-2.4, 11.7]$
Cora, official GPSConv vs matched GCN	10	12.9	14.2	-1.3	9.8	$[-8.4, 5.7]$
Cora, GRIT-style vs GCN	10	9.3	6.4	2.9	6.9	$[-2.0, 7.8]$
Cora, GRIT-style vs matched GCN	10	9.3	16.0	-6.7	9.1	$[-13.2, -0.2]$
Citeseer, GPS-style vs GCN	10	16.4	1.3	15.1	7.6	$[9.7, 20.5]$
Citeseer, GPS-style vs matched GCN	10	16.4	2.8	13.7	7.0	$[8.6, 18.7]$
Citeseer, tuned GPS vs GCN	10	13.0	1.8	11.2	9.8	$[4.2, 18.3]$
Citeseer, tuned GPS vs matched GCN	10	13.0	2.4	10.6	9.2	$[4.0, 17.1]$
Citeseer, official GPSConv vs GCN	10	16.7	0.7	16.0	9.4	$[9.3, 22.7]$
Citeseer, official GPSConv vs matched GCN	10	16.7	2.6	14.1	9.5	$[7.3, 20.9]$
Citeseer, GRIT-style vs GCN	10	10.6	2.3	8.2	8.7	$[2.0, 14.4]$
Citeseer, GRIT-style vs matched GCN	10	10.6	2.7	7.9	8.4	$[1.9, 13.9]$

Citeseer and against the plain GCN on Cora, the tuned transformer against the plain GCN on Cora, and official `GPSConv` against both baselines on Citeseer. The two negative point estimates are the official `GPSConv` and GRIT-style victims on Cora against the matched GCN; the first is not separated from zero and the second is ($[-13.2, -0.2]$ over ten seeds), the one cell in which the reference attack finds the surrogate a better fit for the transformer-style victim than for its accuracy-matched GCN.

The Cora accuracy-matched interval contains zero, which does not establish equivalence; the tuned rows are separated from zero, Section M repeats the tuned cell under an adaptive reference on both arms, and the per-step oracle on both arms of the original cohort separates the cell from zero over ten seeds, 9.7 with $[1.7, 17.7]$ (Section 7). These values are not a causal decomposition intrinsic to the victim: changing the baseline or reference changes them.

M A Fixed-Protocol Validation of the Cora Accuracy-Matched Cell

The weakest cell in Table 2 is Cora against the accuracy-matched GCN, whose interval contains zero. This appendix reports a separate run that fixes the estimator in advance and uses an adaptive reference on both arms.

The protocol was written into a script before the run and not changed afterwards. It claims a fresh output directory, records a SHA-256 of every input it will read (75 files: code, data, the victim and shadow checkpoints, and the linearized-attack records), and refuses to aggregate if any of them changes. The victims and the common target list are those of the tuned cohort of Section E (76.5 and 78.7 percent clean accuracy), ten paired seeds, five flips. The reference on the transformer arm is the PRBCD-style attack differentiated through the positional encoding; on the GCN arm it is the same attack, which has no encoding to differentiate. Writing A for attack success on the stated arm, the per-seed quantity is

$$\Delta_s \;=\; \mathrm{mean}_{a\in\mathrm{LIN}}\big(A^{\mathrm{gcn_m}}_{a,s} - A^{\mathrm{gt}}_{a,s}\big) \;-\; \Big(A^{\mathrm{gcn_m}}_{\mathrm{pgd},s} - A^{\mathrm{gt}}_{\mathrm{pgd\text{-}pe},s}\Big).$$

The ten paired values, seeds 0 through 9, in points:

$$14.45,\ 7.78,\ 7.78,\ 13.34,\ 4.44,\ 8.89,\ 2.22,\ -2.22,\ 6.67,\ 4.44$$

giving $\bar{\Delta} = 6.78$, sample SD 4.96 with `ddof=1`, $t_{0.975,9} = 2.262$, and a paired interval of $[3.23, 10.32]$. One seed is negative. The adaptive references succeed on 71.7 percent of the tar-

gets on the transformer and 73.3 on the matched GCN, a reference gap of 1.7 points against a linearized gap of 8.4. The interval is marginal and unadjusted for the other comparisons in the paper, and it is conditional on this dataset and protocol.

This does not restate the Table 2 entry with more seeds. That entry is the original cohort under the greedy reference; this run is the tuned cohort, a different victim configuration and target draw, under an adaptive reference on both arms. The same tuned cohort under the greedy reference is the tuned row of Table 15, 6.8 with $[2.5, 11.1]$; the two per-seed series differ on four of the ten seeds, which is why their standard deviations differ (6.0 against 5.0), and share their mean to two decimals by coincidence. The three are separate measurements of the same cell rather than one estimate computed twice. What they establish is narrower than a resolution of the original cell and still useful: for the GPS-style victim on Cora against an accuracy-matched baseline, the excess does not vanish when the victims are retuned or when the reference attack is made adaptive to the victim's encoding.

N Replication in the Official Pipeline of Foth et al.

Every victim and attack in the body is our own implementation, so we repeated the comparison inside the released code of Foth et al. (2025) (`gt_robustness`, commit `28ed8b81`, one NVIDIA V100), without modifying any released model or original attack. The victims are their 16-layer GraphGPS and 16-layer GCN, trained by their pipeline from their own configuration files; the linearized surrogate is the two-layer linear GCN their Nettack configuration uses; their adaptive attack is PRBCD with the encoding differentiated. Their targeted protocol draws three victim nodes in each of the first 30 test graphs, so 90 targets, the same nodes for every victim and attacker, and allows at most a fraction b of a graph's edges to be flipped, 3.6 flips at $b = 0.002$ and 9.8 at $b = 0.005$. Their GCN trains poorly on split 123 (53.3 percent clean accuracy on the targets against 72 to 73 elsewhere), so we read no cross-victim conclusion out of it. Exactly two tracked files differ from that commit, each by the one line the release needs to import a BigBird module and register its configuration key. We added a shadow of each family trained by their pipeline at a different seed, an evaluation routine that replays every attacker's flips on every victim so all rows share one metric, and our query attacker.

The query attacker, and what it costs. The attacker forms proposals by attacking its shadow at budgets b and $3b$, then uses at most 50 victim score queries per target to select edits. The archived benchmark on one V100 and split 42 gives 3274.3 seconds for the complete adaptive victim-side run over 90 targets, or 36.4 seconds per target, with 187.3 MiB peak allocation. A normal-budget attacker-owned shadow search takes 3280.6 seconds, or 36.5 seconds per target, with 194.8 MiB peak allocation. That 36.5-second measurement does not include the additional triple-budget shadow search used by the hybrid, so it is not an end-to-end hybrid cost. Separately, the query log records 295.54 seconds for 5208 victim forwards and 20.9 MiB peak allocation; scaling that mean forward time to 50 queries gives 2.8 seconds. This is a normalized victim-forward estimate, not a measured full attack. The method saves victim-side access, time, and memory; total compute, including shadow training and both shadow searches, is not established by these measurements.

Results, and the noise floor. Table 16 averages over the five splits; the per-split values are in the supplementary package. The ordering of the main text survives the move into their code: random flips 8 to 9 percent, the linearized surrogate 31 to 34, a trained shadow 60 with no queries, and the strongest attacks 61 to 71. On the transformer our 50-query attacker stays close to their adaptive white-box attack on average over the five splits without victim weights or gradients: paired over splits the difference is -1.25 ± 1.37 ($t(4) = -2.04$, $p = 0.11$) at the small budget and $+0.17 \pm 4.07$ ($t(4) = 0.09$, $p = 0.93$) at the next, so neither budget separates the two attackers in this sample. That is numerical closeness, not a claim of equality. On the GCN the adaptive attack keeps a clear lead at the small budget, 71 against 67. One caveat on reproducing these numbers exactly: the 16-layer victim's forward pass is not bit-deterministic on a GPU, so the set of targets it classifies correctly varies, 65 to 67 of 90 across five repetitions of split 42, and per cell the run-to-run standard deviation is 0.7 to 1.7 points at $b = 0.002$. A one or two point difference between rows is inside the noise of a single row. Every comparison here is between rows of one run, which share the target set and the denominator.

Two controls. The two victims do not classify exactly the same targets correctly, so we recomputed every row on the 45 to 63 targets of each split that both classify correctly. On that common cohort their adaptive attack reports a GCN-to-GT gap of 11.5 points at the small budget, our query attacker 8.7 and the linearized surrogate 7.4, so here the cheap surrogate understates the difference the strong attackers find, the opposite of its direction on the citation graphs; the per-split values vary widely and split 123 has the poorly trained GCN, so we interpret only the direction, not these magnitudes. Sweeping the query budget over 10, 25, 50, 100 and 200 gives 59.1, 60.7, 61.3, 60.9 and 62.2 percent on their transformer against 62.5 for their adaptive attack and 59.7 for zero-query transfer: zero-query transfer already reaches 59.7 and two hundred queries add 2.5, so the shadow carries the attack. Their GCN victim shows the same sweep where transfer has more to close, 60.3, 63.8, 66.9, 68.2 and 69.7 against 70.9 and 60.4, so queries keep paying to 200. The sweep shows descriptively that more queries help most when zero-query shadow transfer leaves a larger gap to the white-box reference.

Table 16: Official pipeline of Foth et al. (2025) on CLUSTER: their 16-layer GCN and GraphGPS victims, their 90 targets and budget (3.6 flips on average), mean over their five splits (42, 670, 123, 7, 99). Success is the percentage of correctly classified targets an attack flips; accuracy is over all 90, the metric their code reports. Every row shares targets, budget and evaluation routine. Access: W victim weights and gradients, T a shadow trained on the same data, Q score queries.

Attacker	Access	Success		Attacked accuracy		Gap
		GCN	GT	GCN	GT	
Random edge flips	none	9	8	65	71	+1
Nettack on their linear surrogate	none	34	31	45	50	+3
Their PRBCD on the other victim, transferred	W (other model)	56	54	30	34	+2
Our shadow of the other family, transferred	T	58	52	29	35	+6
Our shadow of the same family, transferred	T	60	60	27	29	+0
Our hybrid, 50 score queries	T + Q	67	61	23	28	+6
Their adaptive PRBCD (white-box)	W	71	63	20	27	+8

O Two Published Black-Box Attacks as Baselines

The attackers this paper compares against are linearized-GCN surrogates and white-box searches, neither of which is a query-efficient black-box attack, so we ran two published ones under our protocol: GF-Attack (Chang et al., 2020), which is restricted black-box and spends no victim queries, and an adaptation of the Bayesian-optimization attacker of Wan et al. (2021) at the same 50 score queries per target our hybrid spends. Both run in the same harness on the same victims, seeds, targets and budget, so every column of Table 17 compares within one victim. We name the second a GRABNEL-style adapted baseline rather than GRABNEL, because it is not their released evaluation: their method attacks graph classification and we port the loop to targeted node classification, replace their sparse prior with a ridge prior because our feature map is eight dimensional, and fit that surrogate to a graph-only description of each candidate flip, with expected improvement choosing what to query and the victim's margin as the observation. Observations are pooled across the five steps of a target, the reading most favourable to the baseline. GF-Attack is left as published: it maximizes $\sum_i |\lambda_i|^{2T} \|U_i^\top X\|^2$ over the high-frequency end of the spectrum at 128 eigenpairs and $T = 2$, reading only the graph and the features, with candidates restricted to the flips our threat model allows.

The Bayesian-optimization baseline is real and not a straw one: at the same budget it beats querying random candidates by 5 and 4 points on the two transformers and the best linearized surrogate by 7 and 5. It is still below our hybrid on every victim, by 13.3 and 8.0 points on the transformers ($p < 0.001$ and 0.013) and 7.7 and 2.7 on the GCNs ($p = 0.001$ and 0.003), and below greedy white-box search by 13.0 and 9.3 on the transformers. Its deficit to our attacker is about twice as large on the transformers as on the GCNs; this cross-victim contrast is descriptive, not a paired estimate of Eq. (1). GF-Attack lands below random edge additions on all four victims ($p \leq 0.023$ throughout). We do not read that as evidence about graph transformers, because it holds on the GCNs just as strongly: its objective is untargeted, a global property of the spectrum, so under a protocol that names a node it selects the edge that most degrades the whole embedding rather than the one that hurts the target. We read the result as a task mismatch: the published query-free restricted black-box attack does not transfer to the targeted per-node setting of this paper, on either victim family.

The GF-Attack result is not a tuning artifact. Sweeping each of its free choices in turn on the transformer over seeds 0 to 4, $T \in \{1, 2, 3\}$ at $K = 128$ and $K \in \{32, 128, 512\}$ at $T = 2$, moves it by at most 4.0 points from the published setting, and the best number any of the ten variant-graph pairs reaches is 14.0 against 16.0 for random edge additions on the same seeds.

Table 17: Two published black-box attacks under this paper's protocol: success at five flips, mean $\pm$ std over ten seeds, fair targets drawn per victim. Every attacker in a column sees the same victim, seeds, targets and budget. Access: *scores* are class probabilities the victim returns, *shadow* a model the attacker trains on the same public data, *weights* the victim's parameters and gradients.

		Graph transformer		GCN	
Attacker	Access	Cora	Citeseer	Cora	Citeseer
Random edge additions	none	15 ± 2	18 ± 7	29 ± 7	27 ± 5
GF-Attack, spectral (Chang et al., 2020)	none	9 ± 4	12 ± 7	22 ± 6	15 ± 6
Best linearized-GCN surrogate	none	44 ± 7	46 ± 9	59 ± 4	58 ± 3
Random candidates, 50 score queries	scores	46 ± 5	47 ± 8	n/a	n/a
Trained shadow, no queries (ours)	shadow	61 ± 10	51 ± 10	64 ± 3	58 ± 2
GRABNEL-style BO, 50 score queries (Wan et al., 2021)	scores	51 ± 6	51 ± 12	59 ± 4	56 ± 2
Hybrid, 50 score queries (ours)	shadow + scores	64 ± 8	59 ± 10	67 ± 4	59 ± 2
White-box greedy gradient	weights	64 ± 7	60 ± 11	65 ± 5	59 ± 2

P White-Box Comparison by Victim

Table 18: One attacker across the victims that carry the full set of four attackers, on their own fair targets except the last vanilla row, which uses the 90 targets of Foth et al. (2025); the other victims are in their own appendices. Success is the percentage of targets flipped at five flips (3.6 for the last vanilla row, the convention of Foth et al. (2025)), over the seeds of the table cited. Surrogate is the best linearized-GCN attack available there; the reference is greedy gradient on the victim, except the last vanilla row, where it is the authors' own adaptive PRBCD. Targets, seeds, budget and reference differ by row, so rows compare attackers on one victim and columns do not compare across rows. The hybrid is above the reference in 5 of the 10 rows that have one and level in two more (Section P). The filtered rows carry no single reference: a defence-blind gradient attacker reaches 34 on Cora and 51 on Citeseer and a defence-aware one 59 and 57 (Table 3).

Victim and graph	Clean acc.	Seeds	Random	Surrogate	Shadow (0 q)	Hybrid (50 q)	Strong white-box ref.	Source
Vanilla graph transformers								
GPS-style, Cora (homophily)	71.9	10	15	44	61	64	64	Table 14
GPS-style, Citeseer (homophily)	67.7	10	18	46	51	59	60	Table 14
Sparse GPS, Chameleon (heterophily)	50.9	10	10	18	44	50	50	Section B
Sparse GPS, Chameleon-filtered (heterophily)	35.8	10	27	52	63	67	72	Section B
Sparse GPS, Pubmed (20k nodes)	85.5	10	17	56	71	75	73	Table 4
Sparse GPS, ogbn-arxiv (169k nodes)	69.9	5	21	63	62	65	64	Table 6
GPS-style, CLUSTER (inductive)	66.4	10	17	53	66	67	66	Section D
Their 16-layer GraphGPS, CLUSTER	67 to 79	5	8	31	60	61	63	Table 1
Defended graph transformers								
Adversarially trained, Cora	68.3	10	15	32	34	45	41	Table 10
Adversarially trained, Citeseer	66.1	10	14	30	28	36	34	Table 10
Jaccard filter at inference, Cora	70.9	10	15	34	28	52	n/a	Table 3
Jaccard filter at inference, Citeseer	67.3	10	14	35	41	54	n/a	Table 3

Table 18 places one attacker at a time across the victims that carry all four attackers, and Table 19 compares the 50-query attacker with the white-box references for conditions with complete records. The official-pipeline victims use the authors' adaptive attack and remain in Table 16; the GCN controls remain beside their transformers in Tables 7 and 14. Targets, seeds, budgets, and reference variants differ by row, so only within-row differences are interpretable. Across the 16 conditions with an encoding-frozen greedy reference, $H - G$ runs from -5.0 to $+13.0$; the 50-query attacker is sometimes above and sometimes below the first-order reference, with a largest shortfall of 5.0 points against greedy search and 6.3 against the strongest variant, official `GPSConv` on Citeseer. The attention-only control has ten seeds (Section Q); its graph enters only through the encoding, so the search through the encoding is its sole white-box reference, and the 50-query attacker is 4.7 points above it on Cora and 1.0 below it on Citeseer. Thus the table supports closeness in several conditions, not equivalence or dominance.

Table 19: Conditions with complete archived white-box rows: success at five flips on each victim's own fair targets, mean over the retained seeds. *Greedy* is greedy gradient on the victim; *strongest* is the highest-scoring white-box variant in that condition. $H - G$ and $H - S$ are simple within-victim attack-success differences, not the cross-model surrogate excess Δ of Eq. (1). The attention-only victim has no encoding-frozen search because its graph enters only through the encoding, so $H - G$ is not defined there and the search through the encoding is its only reference.

Victim	Seeds	Hybrid, 50 q	Greedy	$H - G$	Strongest	Variant	$H - S$
GRIT-style, Citeseer	10	26.0	13.0	$+13.0$	26.0	greedy, through encoding	$+0.0$
GRIT-style, Cora	10	27.7	15.7	$+12.0$	21.7	greedy, through encoding	$+6.0$
Adv. trained 2pc, Cora	10	51.7	45.3	$+6.3$	48.0	PRBCD-style	$+3.7$
Adv. trained 5pc, Cora	10	45.3	41.0	$+4.3$	44.3	PRBCD-style	$+1.0$
Adv. trained 2pc, Citeseer	10	47.0	44.0	$+3.0$	47.3	PRBCD-style	-0.3
Adv. trained 5pc, Citeseer	10	35.7	33.7	$+2.0$	36.3	PRBCD-style	-0.7
Sparse GPS, Pubmed	10	74.7	72.7	$+2.0$	72.7	greedy, encoding frozen	$+2.0$
Sparse GPS, ogbn-arxiv	5	65.3	64.0	$+1.3$	64.0	greedy, encoding frozen	$+1.3$
GPS-style, CLUSTER	10	67.0	66.0	$+1.0$	66.7	greedy, through encoding	$+0.3$
GPS-style, Cora	10	64.0	63.7	$+0.3$	64.3	PRBCD-style	-0.3
Graphormer-style, CLUSTER	10	69.3	69.3	$+0.0$	71.3	PRBCD-style, through encoding	-2.0
Sparse GPS, Chameleon	10	50.3	50.3	$+0.0$	50.3	greedy, encoding frozen	$+0.0$
GPS-style, Citeseer	10	59.0	60.3	-1.3	63.7	PRBCD-style	-4.7
Official GPSConv, Cora	10	62.7	66.3	-3.7	67.0	PRBCD-style	-4.3
Official GPSConv, Citeseer	10	61.0	66.0	-5.0	67.3	PRBCD-style, through encoding	-6.3
Sparse GPS, Chameleon-filt	10	67.3	72.3	-5.0	72.3	greedy, encoding frozen	-5.0
Attention-only, Cora	10	27.7	n/a	n/a	23.0	greedy, through encoding	$+4.7$
Attention-only, Citeseer	10	25.3	n/a	n/a	26.3	greedy, through encoding	-1.0

Q Ablations

Sensitivity to the success convention. Success is counted at exactly k flips. Counting a target as flipped if any budget up to k flipped it raises the linearized attacks on the transformer by 1.3 to 4.7 points, the periphery-candidate shadow by at most 1.3 and the other strong attackers by at most 0.7, and changes no GCN mean except random additions. The transformer figures use the seeds whose files store per-budget flags, all ten for the linearized GCN with all candidates and for Nettack and five for GOttack transfer; the GCN figures use all ten seeds. On those seeds the linearized gaps between the two victims then read 9 to 17 points instead of 12 to 20 and the white-box gaps are unchanged. The asymmetry is itself informative: the attacks that flip a target at an intermediate budget and lose it again at five are the weak ones, and they do that on the transformer and not on the GCN. One convention is used throughout, and the quantity the paper rests on, the linearized gap minus the white-box gap, shrinks by 1.3 to 4.7 points under the alternative without changing sign.

Sensitivity to the target-ranking rule. Table 7 ranks the common strata by the mean within-model margin percentile. Ranking instead by the mean of two raw margins, on seeds 0 to 4, gives a white-box gap of 9 points on Cora ($p = 0.28$) and 3 on Citeseer ($p = 0.18$) against 4 and 3 under the percentile rule on the same seeds, and both rules agree the linearized surrogate's gap is far larger at 16 to 21 points. Seeds 5 to 9 put the Cora white-box gap at 13 and all ten at 8, so the first five understated it, which is why the main text reports ten. The percentile rule is used because it does not favour the model with the wider margin distribution.

A hard-label oracle. Every query elsewhere returns the victim's class scores. An interface that returns only the predicted label is the weakest oracle worth naming, and the same attacker runs against it unchanged except that a query reports whether the target's label changed and nothing else: it keeps the first candidate that flips the target and otherwise the shadow's highest-ranked proposal. Over ten fair-target seeds on the main victims it reaches 62.3 ± 8.0 on Cora and 55.0 ± 8.8 on Citeseer, against 64.0 ± 8.0 and 59.0 ± 9.8 with scores and 60.7 ± 9.5 and 51.3 ± 10.2 with no queries at all. Hard labels recover about half of what scores buy, $+1.7$ of the $+3.3$ on Cora and $+3.7$ of the $+7.7$ on Citeseer, paired over seeds at $p = 0.052$ and $p = 0.012$ against zero queries, with the score advantage that remains at $p = 0.34$ and $p = 0.051$. Truncating an interface to top-1 therefore costs an evaluator about half the query gain in point estimate; what remains over transfer is separated from zero on Citeseer and not on Cora.

Architecture mismatch is not resolved by ten seeds. A GPS-style shadow attacking a Cora variant of the GPS-style victim whose attention adds a learnable bias toward connected node pairs (71 to 74 percent clean accuracy, 79 to 85 percent agreement) transfers at 61 ± 12 percent zero-query and

66 ± 6 with 50 queries over ten seeds (seeds 0 to 2 regenerated from scratch after the original files were lost), against 59 ± 10 and 64 ± 7 for a same-variant shadow on the same victim, seed and targets (paired t, $p = 0.24$ and 0.28). Seed by seed the mismatch costs at most 3.3 points zero-query and 6.7 with queries, helps on four of ten in each case, and ties on four. In the CLUSTER runs of Section D the cross-family cost is at most seven points, and its reported mean is smaller after adding queries.

A lookahead on the shadow buys a few points without victim queries. Replacing the shadow's first-order score with its exact margin after each candidate flip, which costs shadow evaluations and no victim queries, takes Cora from 61 to 64 and Citeseer from 51 to 56 over ten fair-target seeds, so some of what the queries buy is available to an attacker willing to spend its own compute instead.

Removing message passing sharply reduces first-order attack success in the GPS-style victim. Removing the local GCN block leaves global attention connected to the graph only through the random-walk encoding. Over ten fair-target seeds (seeds 0 to 2 regenerated from scratch after the original files were lost, seeds 3 to 9 archived) this victim reaches 62.4 percent clean accuracy on both graphs, against 71.9 and 67.7 for the full victim, and it is much less attackable: greedy white-box search through the encoding flips 23 ± 8 and 26 ± 6 percent of its targets on Cora and Citeseer against 64 and 60 for the full victim, the same-family shadow 15 ± 6 and 16 ± 4 against 61 and 51, the 50-query hybrid 28 ± 9 and 25 ± 4 against 64 and 59, and random additions 8 ± 3 and 9 ± 4 against 15 and 18. Every attack on this victim stays within 20 points of random additions. `paper_evidence/reproduce.py` regenerates these values from the ten per-seed files per graph. No standalone MLP result or checkpoint is present, so no MLP comparison is made. The ten-seed GRIT-style control in Section I provides a second no-message-passing boundary.

The mirror ablation: removing attention leaves the measured mismatch similar. The paragraph above removes the local block. The symmetric control removes the other branch, keeping the local GCN block and deleting global attention entirely, and it is the one that tests whether any of this is about attention. Over ten fair-target seeds that victim reaches 71.7 ± 1.2 and 67.5 ± 1.8 percent on Cora and Citeseer, within 0.2 points of the full victim's 71.9 and 67.7: on these runs removing the attention branch changes mean clean accuracy by only 0.2 points. Its attack success is also close to the full victim's. Greedy white-box search flips 64.3 ± 9.9 and 57.0 ± 8.2 percent of its targets against 64 and 60 for the full victim, the mean linearized-GCN attack 42.8 and 42.3 against 42 and 44, a trained same-family shadow 56.7 ± 6.7 and 48.7 ± 5.5, and shadow-victim agreement is 83.2 and 82.4 against 82.0 and 81.7. The surrogate's deficit D is 21.6 and 14.7 against 21.7 and 16.3 for the full victim on its own fair targets (Table 21). Thus the measured mismatch remains similar in a model with no global attention, while the model with nothing but global attention is much less attackable over ten seeds. Together, these results indicate that the evaluated first-order attacks rely predominantly, but not exclusively, on the message-passing block in this GPS-style victim. This points to a mismatch between the surrogate and victim rather than to a property of attention, which is what Section 1 claims and what Section J found on the GAT and GIN victims from the other direction.

Polblogs. On the GPS-style victim over five seeds with a common target list, five-flip success is 16.0 for random flips; 30.0, 28.7 and 32.7 for GOttack transfer, the linearized surrogate over all candidates, and Nettack; 38.7 for zero-query shadow transfer; 42.7 for greedy white-box search; and 48.0 for the 50-query hybrid. The query attack is the only one above white-box search. These rows are regenerated by `paper_evidence/reproduce.py`.

R An SGFormer-Style Transformer Victim

The headline estimates of Table 2 use the GPS-style victim. To ask whether the reference-relative excess measured for that victim also appears for a different transformer, we repeated the common-target protocol with an SGFormer-style victim in place of the GPS-style one, on both datasets, with ten seeds (nine in one Cora cell). Targets are the nodes classified correctly by the SGFormer-style victim and by both GCN baselines, so the three arms share a target list, except on one Cora seed for the matched GCN, which is dropped from those rows. The SGFormer-style victim reaches 79.8 and 74.0 percent clean accuracy against 83.6 and 75.0 for the plain GCN, and its accuracy-matched baseline is a GCN at 80.8 and 73.9 percent. Greedy white-box search does not terminate on this victim within our budget, so the reference attack throughout this appendix is the relaxed

white-box attack, applied to both arms of every pair; Table 2 is unchanged and uses the greedy reference. The eight SGFormer-style cells below are regenerated from the raw per-seed files by `paper_evidence/reproduce.py`, which refuses any pair whose two arms did not attack the same targets.

The excess is not resolved on this victim. Against a plain GCN the excess is 2.4 points on Cora and 1.9 on Citeseer; against the accuracy-matched GCN it is 3.6 and 1.6. All four intervals contain zero. Under the same reference attack the GPS-style victim gives 14.2, 17.4, 7.7 and 16.3, and three of those four exclude zero. An interval containing zero is not evidence that the excess is zero, and these intervals run about five to six and a half points either side of the estimate. What we can say is that the effect we report for the GPS-style victim is not resolved here. We therefore scope the claim in Section 1 to victims that a linearized surrogate approximates poorly, and not to graph transformers as a class.

The shadow transfers, but adds little beyond the linearized surrogate. This is a limitation of the method and not only of the framing. On the GPS-style victim the shadow attacker recovers 15.1 of the 22.4 points that separate the reference attack from the linearized attacks on Cora, and 8.4 of 19.4 on Citeseer. On the SGFormer-style victim it recovers 1.1 of 13.4 on Cora, and on Citeseer it lands 4.6 points below the linearized attacks. Panel B of Table 20 pairs this against the GCN arms: the shadow advantage is lower on the transformer victim than on its GCN baseline in all four cells, by 5.1 to 8.2 points, and every interval excludes zero.

This is not an implementation fault. The failing and the working cells come from the same runs, the same code path and the same target lists. Within the SGFormer-style family the shadow attacker gains 9.3 points over the linearized attacks on the Cora GCN arm, but only 1.1 on the transformer arm. Over ten seeds, random edge flips reach 16.0 percent success on the Cora transformer arm where the shadow attacker reaches 49.3, so the attacker is selecting edits, it is simply not selecting better ones than a linearized surrogate does.

Higher agreement did not mean better transfer. The explanation we expected was a less faithful shadow, and the data do not support it. The SGFormer-style shadow agrees with its victim on 89.7 percent of test predictions on Cora and 89.4 percent on Citeseer; the GPS-style shadow agrees on 82.0 and 81.7 percent. Nor do the other victims show a detectable association: Section S repeats the comparison on every victim we have trained and finds a Spearman rank correlation of 0.14 between agreement and the fraction of the gap a shadow attacker recovers, over thirteen cells and seven architectures, short of significance at that sample size. The surrogate that is closer on the fidelity measure usually reported for surrogate attacks is the one whose edits transfer worse. The account the architecture suggests is that the global branch of this victim never sees the graph, so a structure attack has to act through its local branch, which is propagation on the same normalized adjacency the GCN baselines use. A linearized GCN surrogate is then already a reasonable model of the attackable pathway, and a shadow of matching architecture adds little that linearization did not already have. That account fits both a small excess and a shadow attacker that gains little, and it is what the design of this victim implies rather than something we test directly here. It does not explain the rest: the reference attack still takes 13.4 points more than the linearized attacks on the Cora transformer arm, and neither surrogate recovers them, so that headroom appears to depend on the victim's particular weights rather than on its architecture. We do not claim a complete mechanism. The practical consequence for an evaluator is that agreement on labels is not sufficient evidence that a surrogate will propose effective edits, and that a surrogate should be judged by the edits it selects rather than by the predictions it reproduces.

S Prediction Agreement and Transfer

The mechanism this paper appeals to is surrogate fidelity, and the fidelity number reported for surrogate attacks is almost always prediction agreement between surrogate and victim. The SGFormer result in Section R is a case where that number is high and the shadow adds little over the linearized attacks, so we asked how the two relate across every victim we have trained.

Table 20: An SGFormer-style transformer victim. Panel A is the reference-relative excess of Eq. (1) with the relaxed white-box attack as the reference on both arms. Panel B replaces the reference with the zero-query shadow attacker, so $D(v)$ and $D(b)$ are what that attacker gains over the mean of the three linearized attacks. Values are percentage points at five flips over ten paired seeds; SD uses `ddof=1` and CIs are two-sided paired t intervals. Entries are rounded independently, so $D(v) - D(b)$ can differ from Δ by 0.1. The SGFormer-style and GPS-style families draw their common targets from different victim triples, so $D(b)$ is not comparable across families; each Δ is paired within its own family. One seed is dropped from the two Cora SGFormer-style matched-GCN rows (Panels A and B) because the two arms did not draw the same thirty targets, so those rows use nine seeds and every other row uses ten. An interval containing zero is not evidence of equivalence.

Condition	$D(v)$	$D(b)$	Δ	SD	95% CI
Panel A: excess relative to the relaxed white-box reference					
Cora, SGFormer-style vs GCN	13.4	11.0	2.4	8.7	$[-3.7, 8.6]$
Cora, SGFormer-style vs matched GCN	13.1	9.5	3.6	8.4	$[-2.9, 10.1]$
Citeseer, SGFormer-style vs GCN	6.1	4.2	1.9	6.9	$[-3.1, 6.9]$
Citeseer, SGFormer-style vs matched GCN	6.1	4.6	1.6	6.8	$[-3.3, 6.4]$
Cora, GPS-style vs GCN	22.4	8.2	14.2	7.6	$[8.8, 19.6]$
Cora, GPS-style vs matched GCN	22.4	14.8	7.7	12.1	$[-1.0, 16.3]$
Citeseer, GPS-style vs GCN	19.4	2.0	17.4	7.4	$[12.2, 22.7]$
Citeseer, GPS-style vs matched GCN	19.4	3.1	16.3	6.7	$[11.6, 21.1]$
Panel B: gain of the zero-query shadow attacker over the linearized attacks					
Cora, SGFormer-style vs GCN	1.1	9.3	-8.2	6.3	$[-12.7, -3.7]$
Cora, SGFormer-style vs matched GCN	0.9	7.3	-6.4	5.7	$[-10.8, -2.0]$
Citeseer, SGFormer-style vs GCN	-4.6	0.6	-5.1	4.9	$[-8.6, -1.6]$
Citeseer, SGFormer-style vs matched GCN	-4.6	1.6	-6.1	5.0	$[-9.7, -2.5]$
Cora, GPS-style vs GCN	15.1	7.2	7.9	10.4	$[0.5, 15.3]$
Cora, GPS-style vs matched GCN	15.1	6.4	8.7	11.5	$[0.4, 16.9]$
Citeseer, GPS-style vs GCN	8.4	1.0	7.4	10.0	$[0.3, 14.6]$
Citeseer, GPS-style vs matched GCN	8.4	0.1	8.3	8.6	$[2.2, 14.5]$

For a single victim, define the gap G left by linearization as the reference attack's success at five flips minus the mean success of the three linearized attacks, and define the shadow gain S the same way with the zero-query shadow attacker in place of the reference. Both are per seed; we report their means and the recovery fraction S/G, a ratio of means rather than a mean of ratios. This quantity needs no cross-victim pairing, so each victim is evaluated on its own fair targets wherever those runs exist. Table 21 lists every victim and dataset for which all three quantities are available.

The recovery fraction is only meaningful when there is a gap to recover. Three cells have $|G| < 5$ points and are reported but excluded from the analysis below: the Citeseer GCN victim, where the linearized surrogate is already close to the reference, and the GRIT control, where G is negative because the encoding-frozen greedy reference scores below the linearized attacks on that victim, which resists every attack we tested (Section I and Table 19). In both cases the measured headroom for a better surrogate is small or negative, which is the same mechanism read from the other end. For GRIT this depends on the reference: under the through-encoding reference used for this victim in Table 15 its G is positive, and including its two cells under that reference leaves the rank correlation below significance.

No detectable rank association between agreement and recovery. Over the thirteen remaining cells, spanning seven victim architectures and two datasets, the Spearman rank correlation between agreement and recovery is 0.14. Agreement ranges from 72.0 to 97.6 percent and recovery from -75 to 86 percent. With thirteen cells this is an absence of evidence rather than evidence of absence: a correlation below about 0.55 in absolute value would not reach significance at this sample size, so we can say we detect no association, not that none exists. The rank correlation between G and recovery is $+0.25$, also not significant, in the direction one would expect if a larger gap is easier to make progress on. The shadow runs on the other datasets each carry one linearized attack rather than three, so they are not in Table 21; computing G and S against GOttack transfer alone for those five cells (Pubmed, agreement 94.8 percent and recovery 90; Chameleon, 77.7 and 80; Chameleon-filtered, 61.8 and 53; CLUSTER GPS-style, 86.1 and 100; CLUSTER Graphormer-style, 69.2 and 3) and

Table 21: Recovery fraction by victim. G is the reference attack's success at five flips minus the mean of the three linearized attacks; S is the same with the zero-query shadow attacker. Recovery is S/G as a ratio of means over seeds. Agreement is shadow-victim prediction agreement on the test set. Rows marked n/a have $|G| < 5$ points, which makes the ratio unstable, and are excluded from the correlations quoted in the text. The reference is greedy white-box search except for the SGFormer-style rows, which use the relaxed white-box attack because greedy does not terminate on that victim. Victims use their own fair targets except the SGFormer-style rows, which use common targets.

Dataset	Victim	seeds	clean acc.	agreement	G	S	recovery
Cora	GPS-style	10	71.9	82.0	21.7	18.7	86%
Cora	official GraphGPS layer	10	74.4	80.0	9.0	3.3	37%
Cora	GAT	10	75.4	72.0	12.7	7.3	58%
Cora	GIN	10	80.1	90.1	10.1	8.4	84%
Cora	no global attention	10	71.7	83.2	21.6	13.9	64%
Cora	SGFormer-style	10	79.8	89.7	13.4	1.1	8%
Cora	GRIT control	10	63.0	79.8	−0.3	−1.0	n/a
Cora	GCN	10	83.6	97.6	6.4	5.1	79%
Citeseer	GPS-style	10	67.7	81.7	16.3	7.3	45%
Citeseer	official GraphGPS layer	10	67.1	75.5	16.8	8.8	52%
Citeseer	GAT	10	72.4	86.9	8.4	4.8	57%
Citeseer	GIN	10	69.3	86.4	12.2	7.6	62%
Citeseer	no global attention	10	67.5	82.4	14.7	6.3	43%
Citeseer	SGFormer-style	10	74.0	89.4	6.1	−4.6	−75%
Citeseer	GRIT control	10	62.9	79.2	−3.2	−3.2	n/a
Citeseer	GCN	10	75.0	97.4	1.0	0.7	n/a

adding them gives a rank correlation of 0.30 over eighteen cells ($p = 0.23$), still short of significance at that sample size.

A matched pair makes the point without any correlation. On Cora the GIN victim has 90.1 percent agreement and the SGFormer-style victim has 89.7 percent, a difference of four tenths of a point on the same dataset. Their recovery fractions are 84 and 8 percent. In the other direction, the GAT victim has the lowest agreement in Table 21 at 72.0 percent and recovers 58 percent. Two surrogates that the reported fidelity measure separates by four tenths of a point differ by a factor of ten in the quantity an evaluator cares about, and the least faithful surrogate by that measure is not the worst attacker.

What this does and does not license. Eleven of the thirteen cells use the greedy white-box reference; the two SGFormer-style cells use the relaxed white-box attack, because greedy search does not terminate on that victim within our budget. On the GPS-style victim, where both references exist on the same targets, the relaxed attack gives the larger G, 22.4 against 21.1 points, so using it for the SGFormer-style cells pulls their recovery fractions slightly toward zero, far too little to create the contrast. The SGFormer-style cells also use common targets rather than their own fair targets, since that is how they were run; a within-victim ratio needs no paired targets, but the target list can still move it, so we name the difference. It is also why the GPS-style figures here, 21.7 and 18.7 points on Cora, are not the common-target figures of 22.4 and 15.1 quoted in Section R. We draw one practical conclusion and no more: an evaluator who reports surrogate agreement has not thereby shown that the surrogate selects effective edits, and should report the edits' effect directly.