香港中文大學 The Chinese University of Hong Kong

香港浸會大學 HONG KONG BAPTIST UNIVERSITY

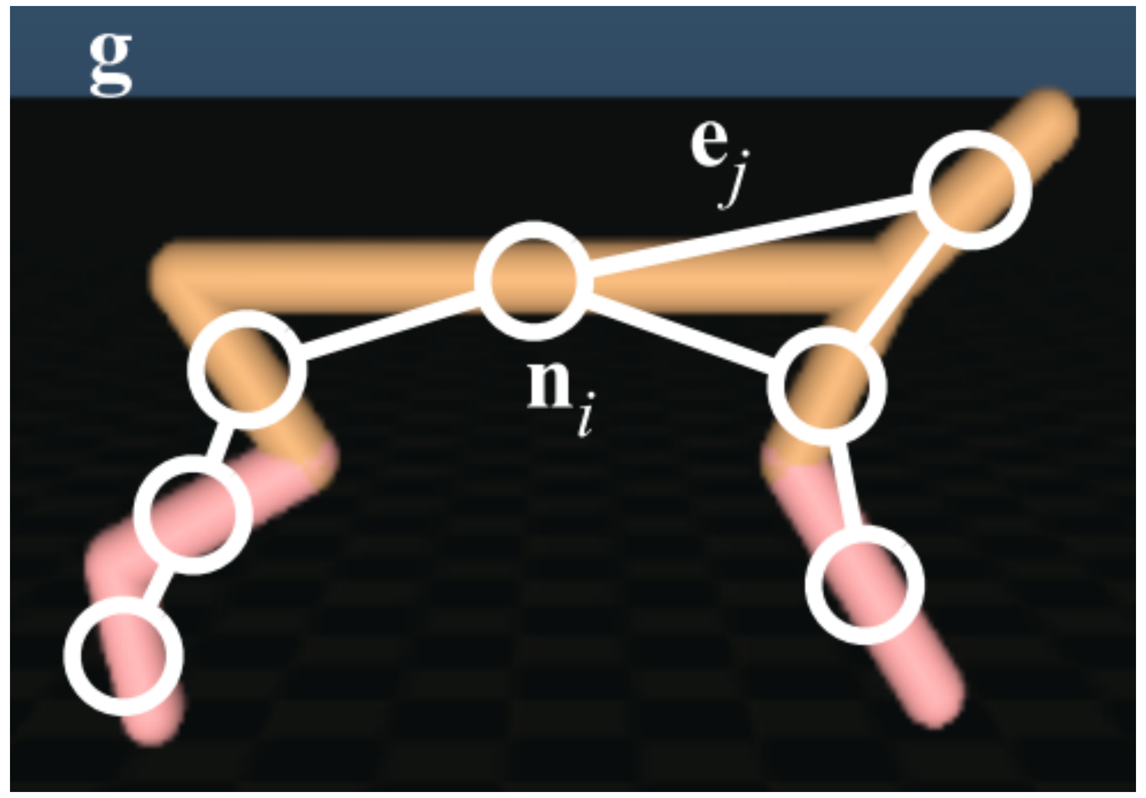
THE UNIVERSITY OF SYDNEY

Understanding and Improving Graph Injection Attack by Promoting Unnoticeability

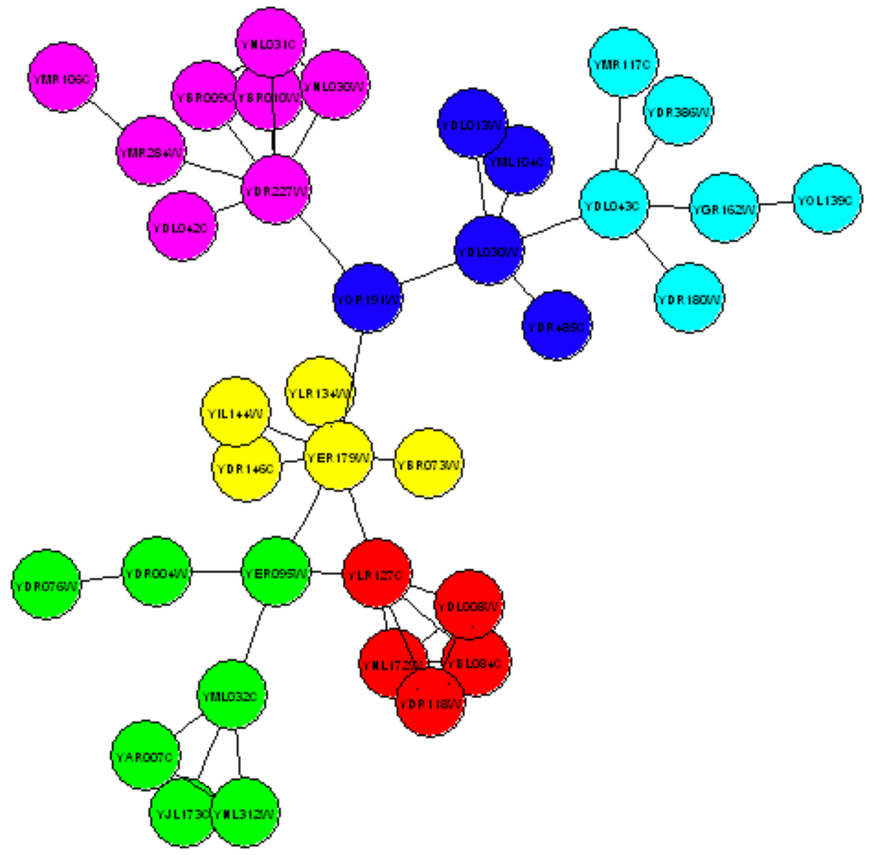
Yongqiang Chen
CUHK

with Han Yang, Yonggang Zhang, Kaili Ma, Tongliang Liu, Bo Han, and James Cheng

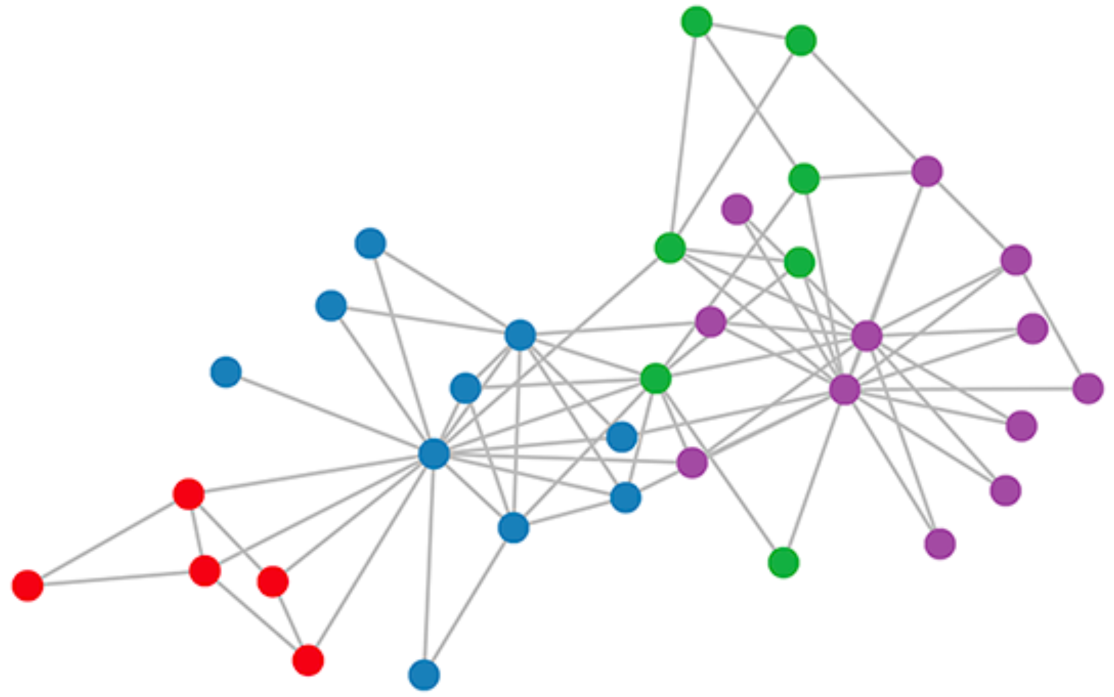
Graph Neural Networks (GNNs) Are Widely Applied



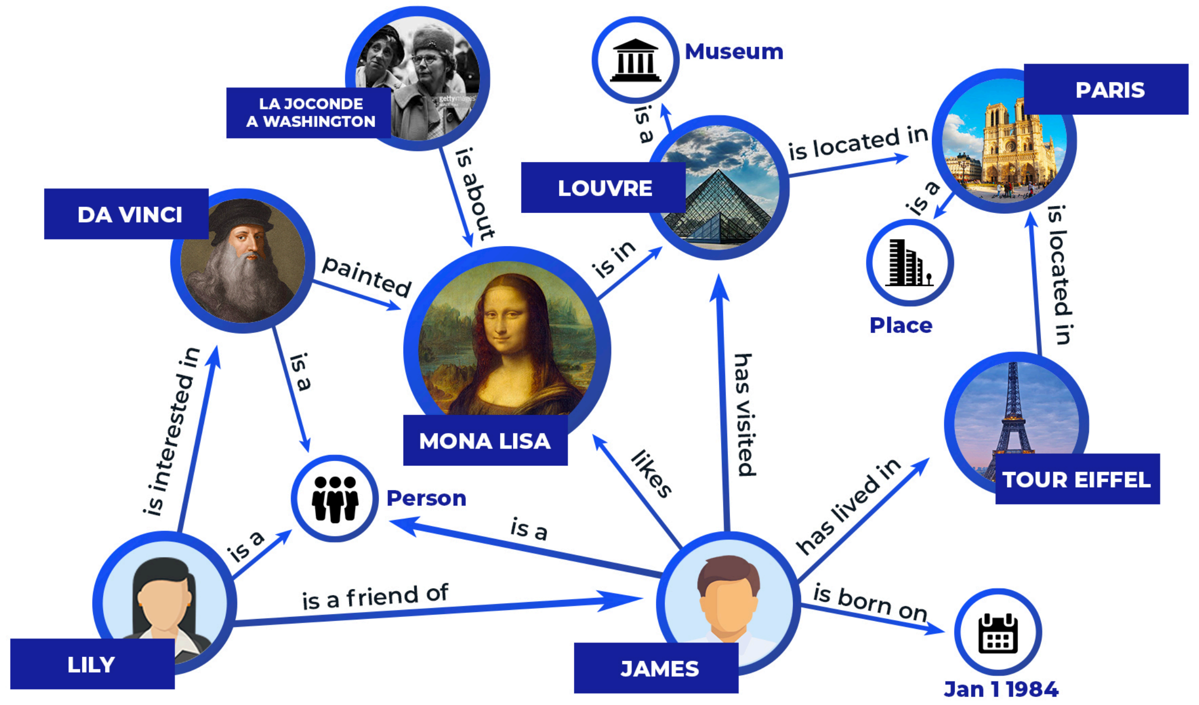
Model & Inference over the physical world



Protein Interaction Predictions

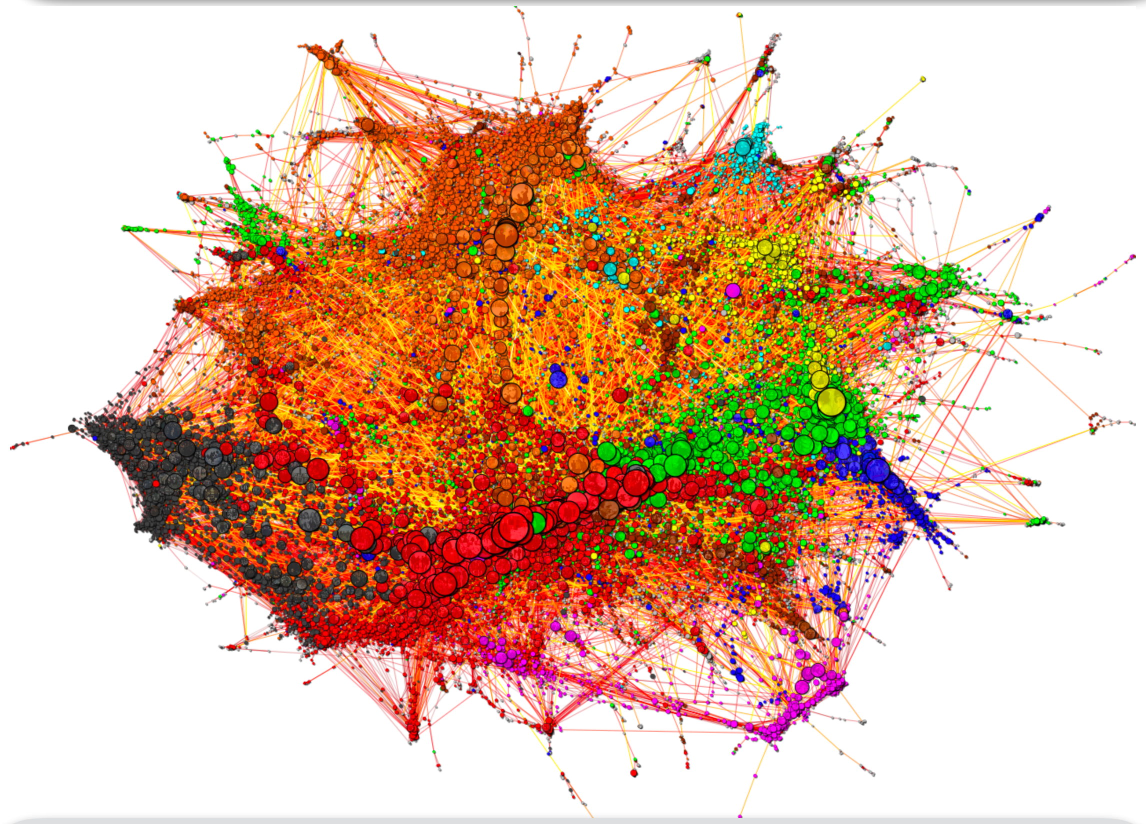


Social Network Analysis



Knowledge Graph Completion & Analysis

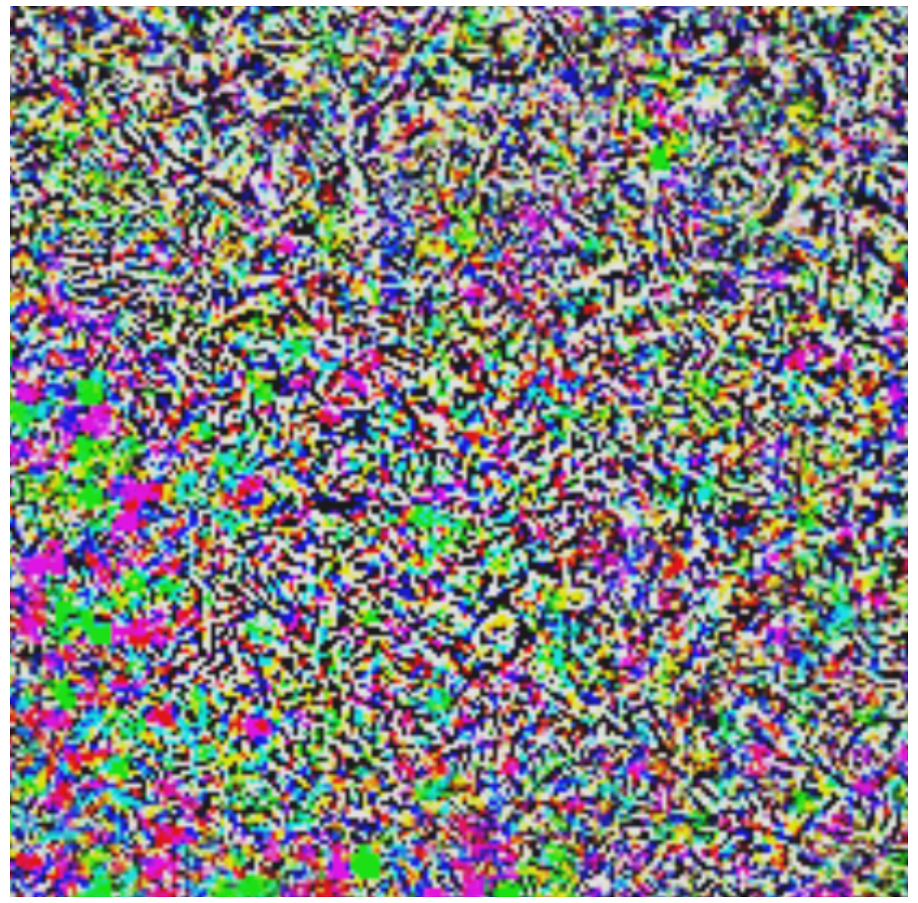
Besides, GNNs can also process structures like image and text…



Recommender Systems

GNNs Are Inherently Vulnerable

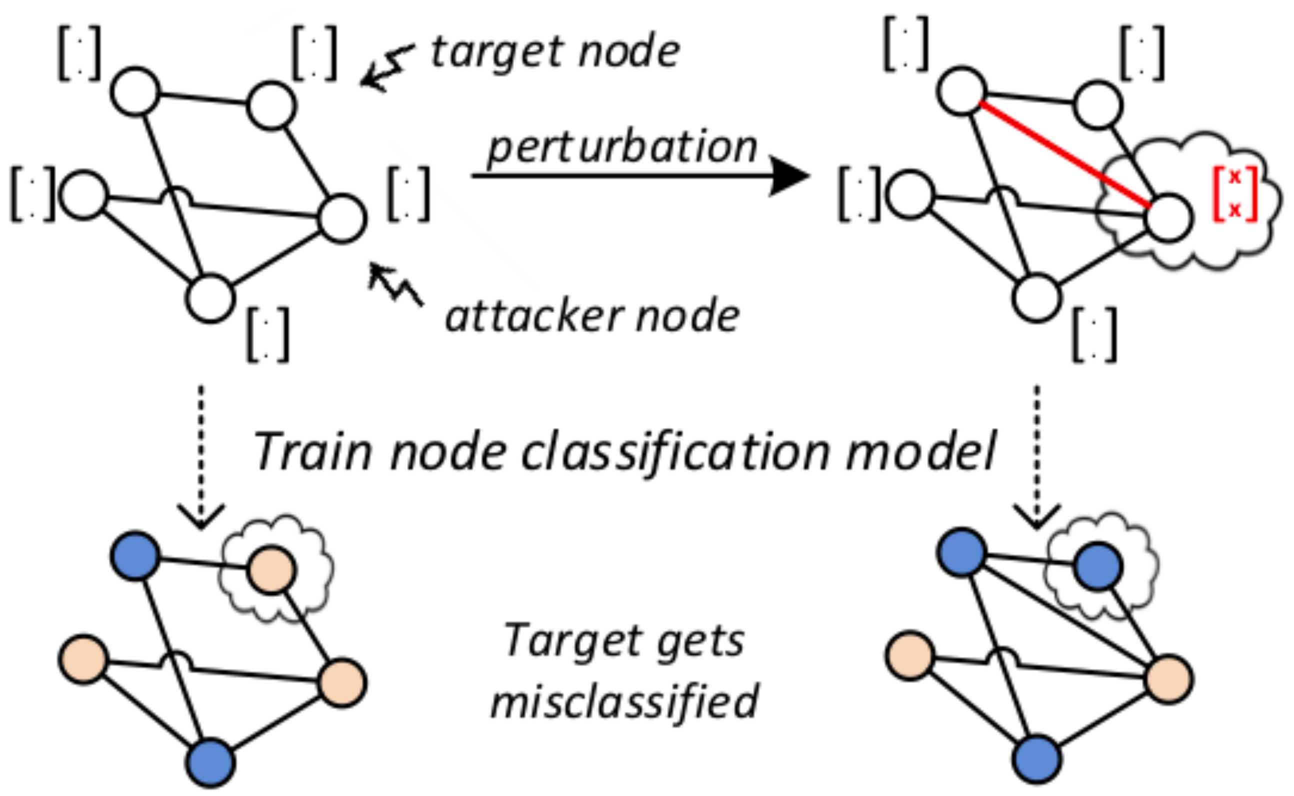
Prediction: Pig



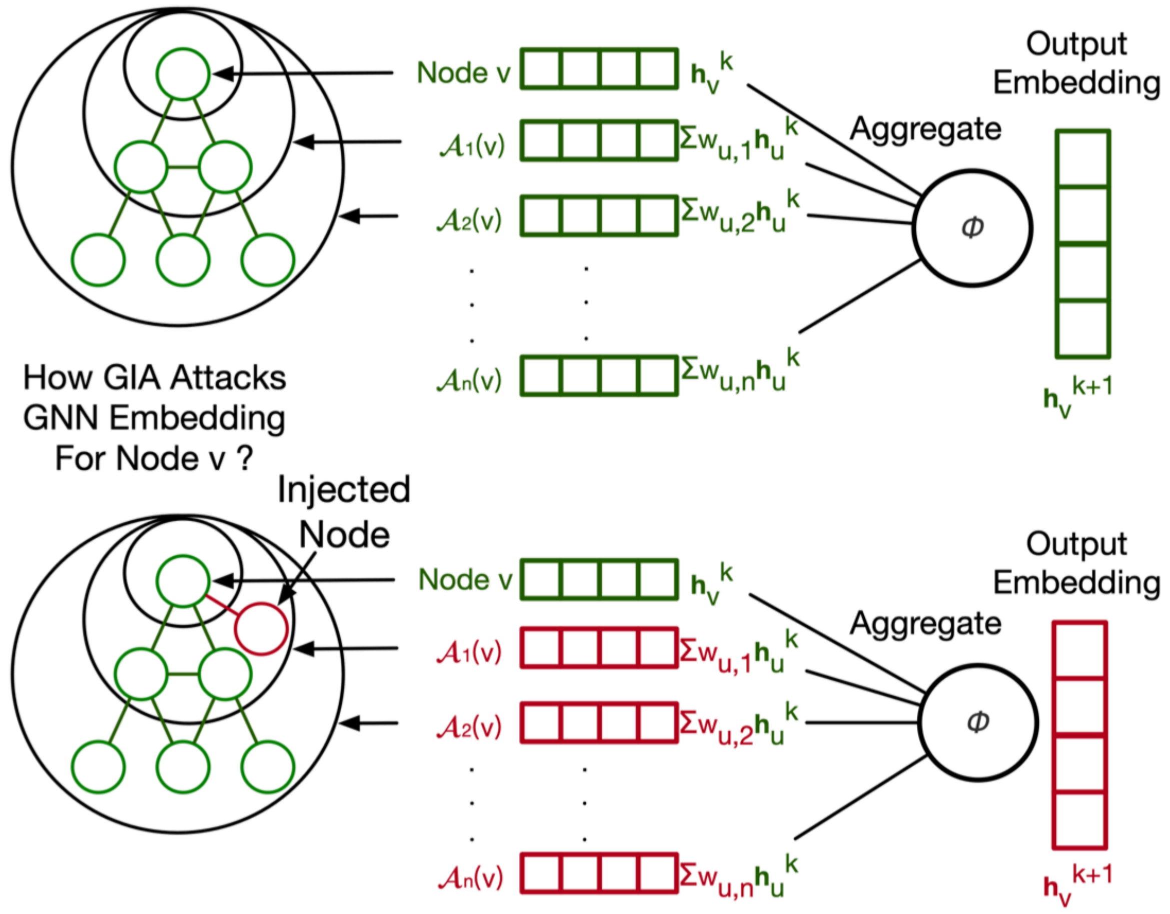
Adversarial noise

Prediction: Airliner

(Szegedy et al., 2014; Goodfellow et al., 2015; Kolter and Madry et al. 2019)

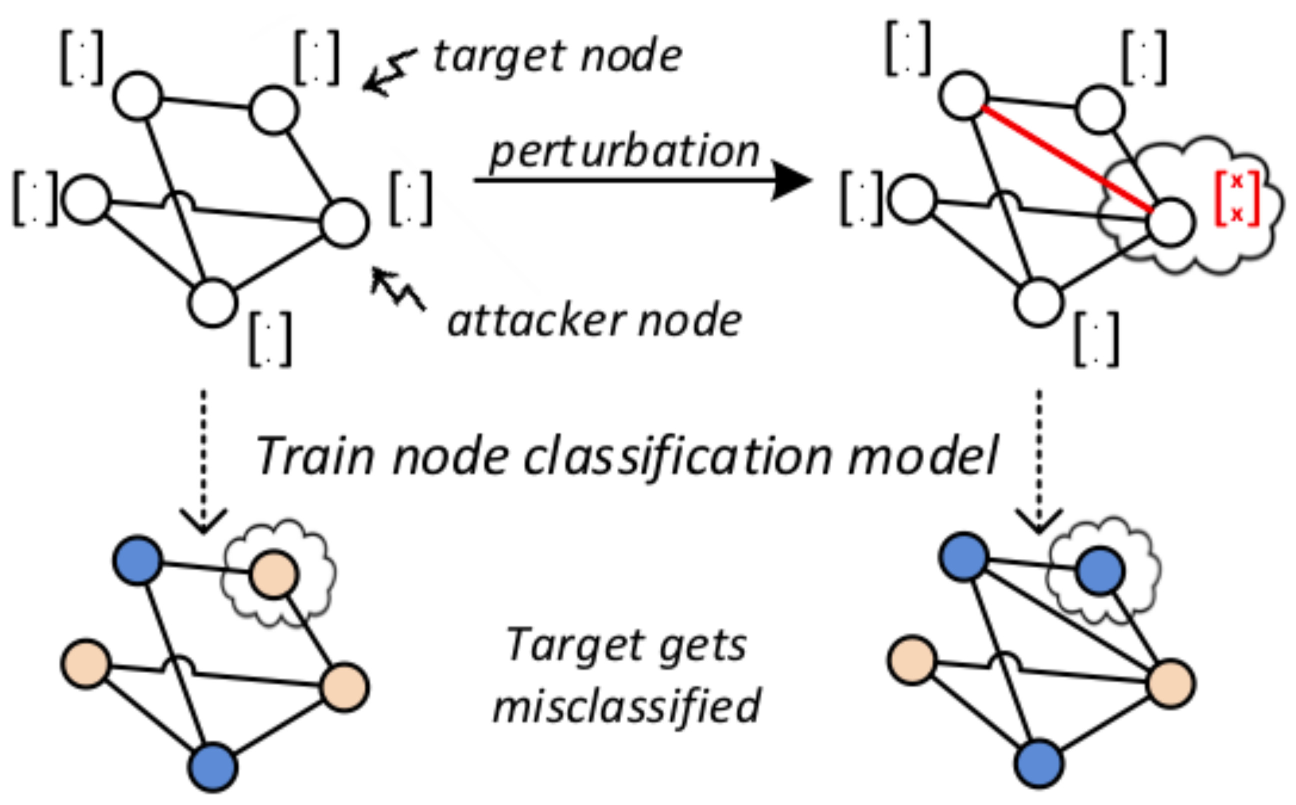


(Zügner et al., 2018)

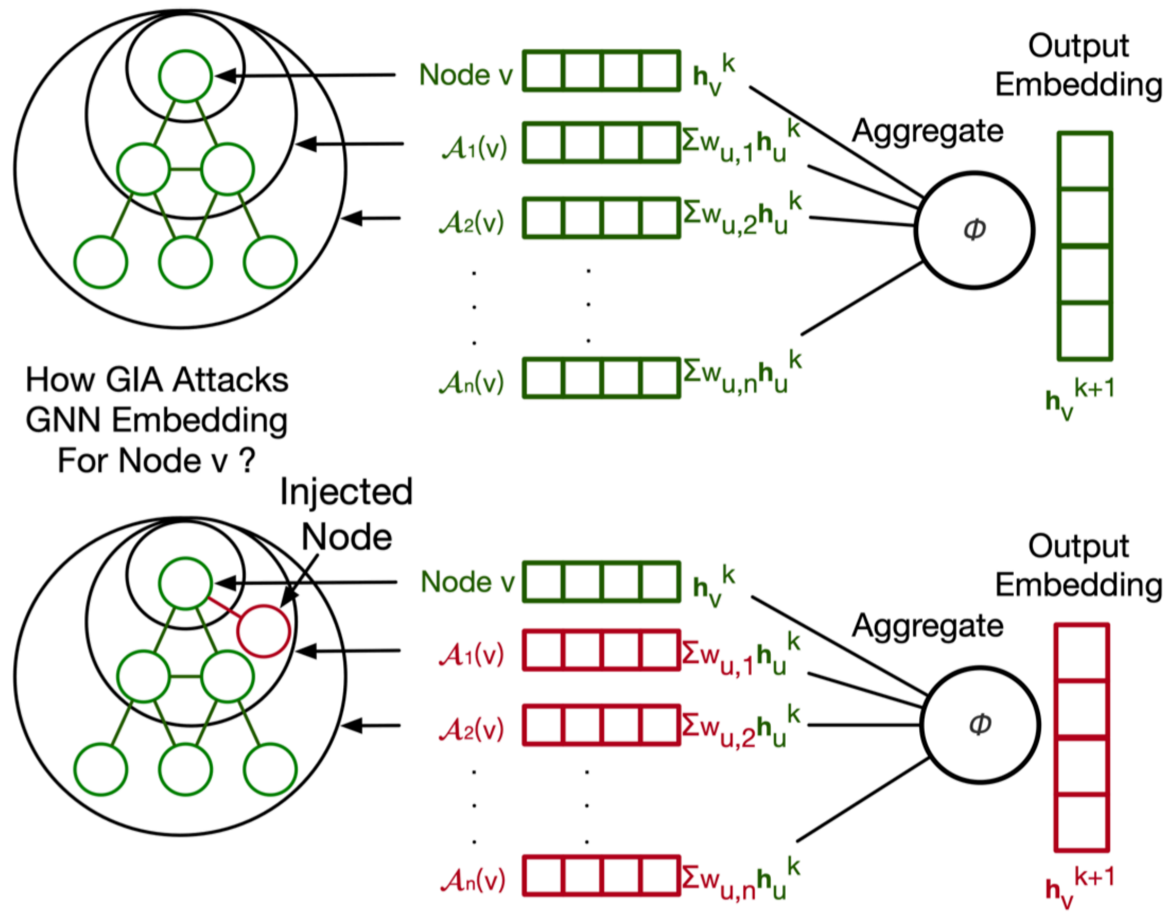


(Zou et al., 2020)

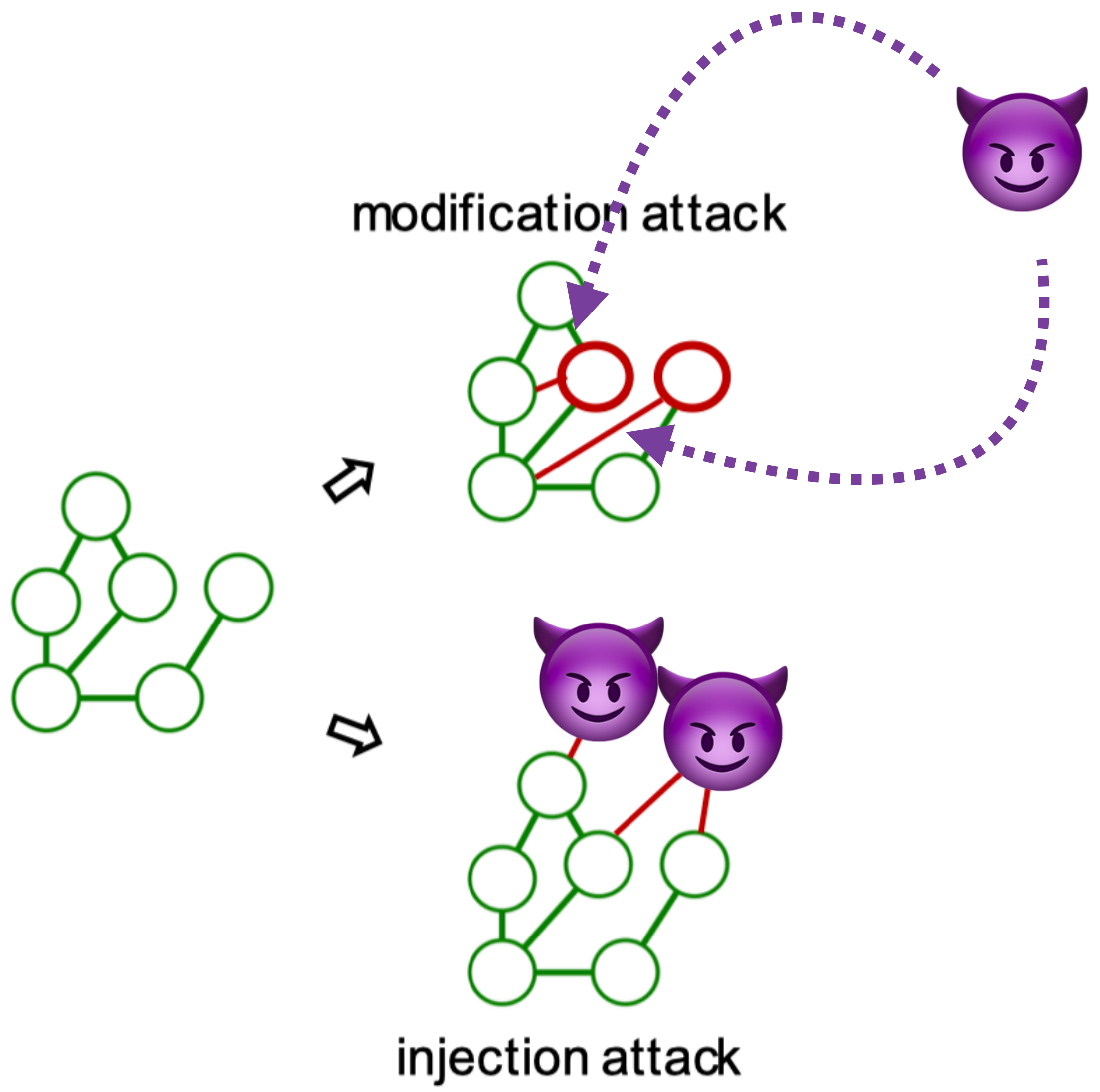
Adversarial Attacks on GNNs



(Zügner et al., 2018)



(Zou et al., 2020)

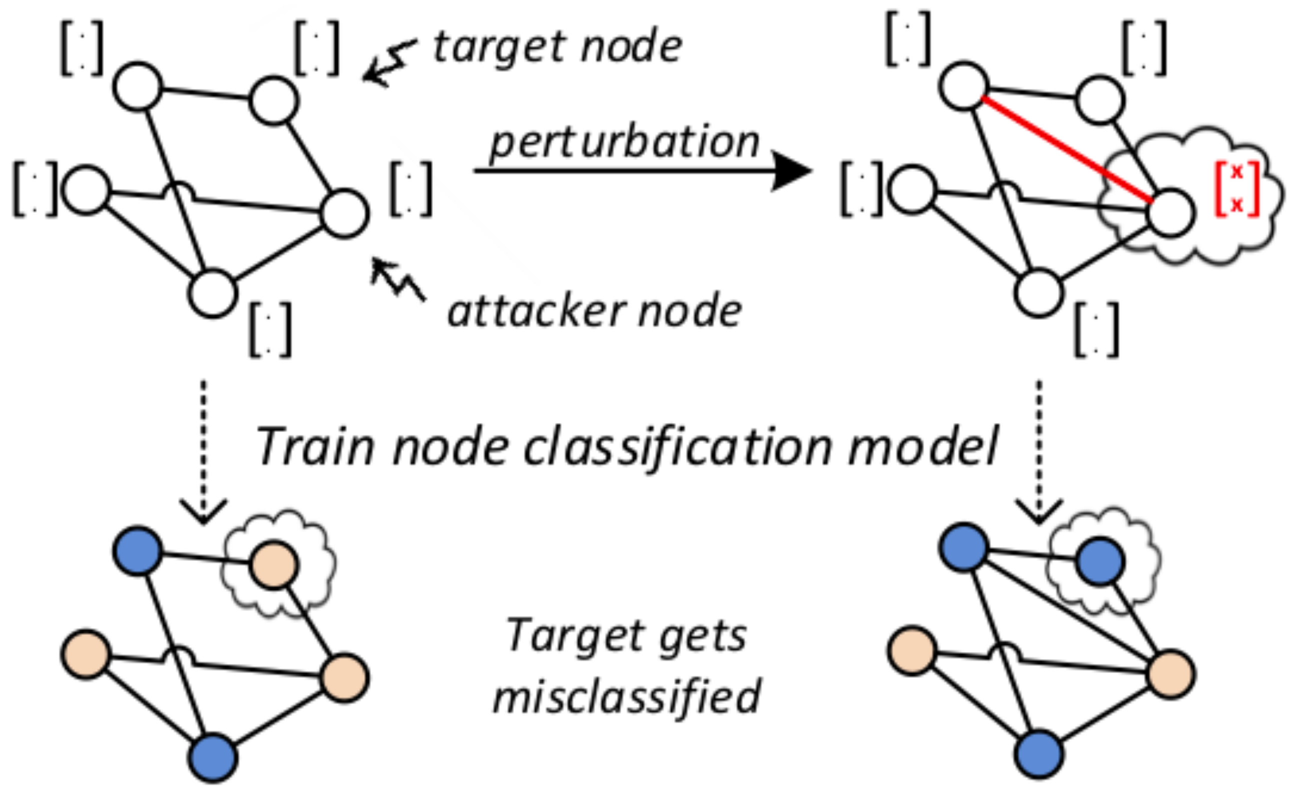


GMA vs. GIA

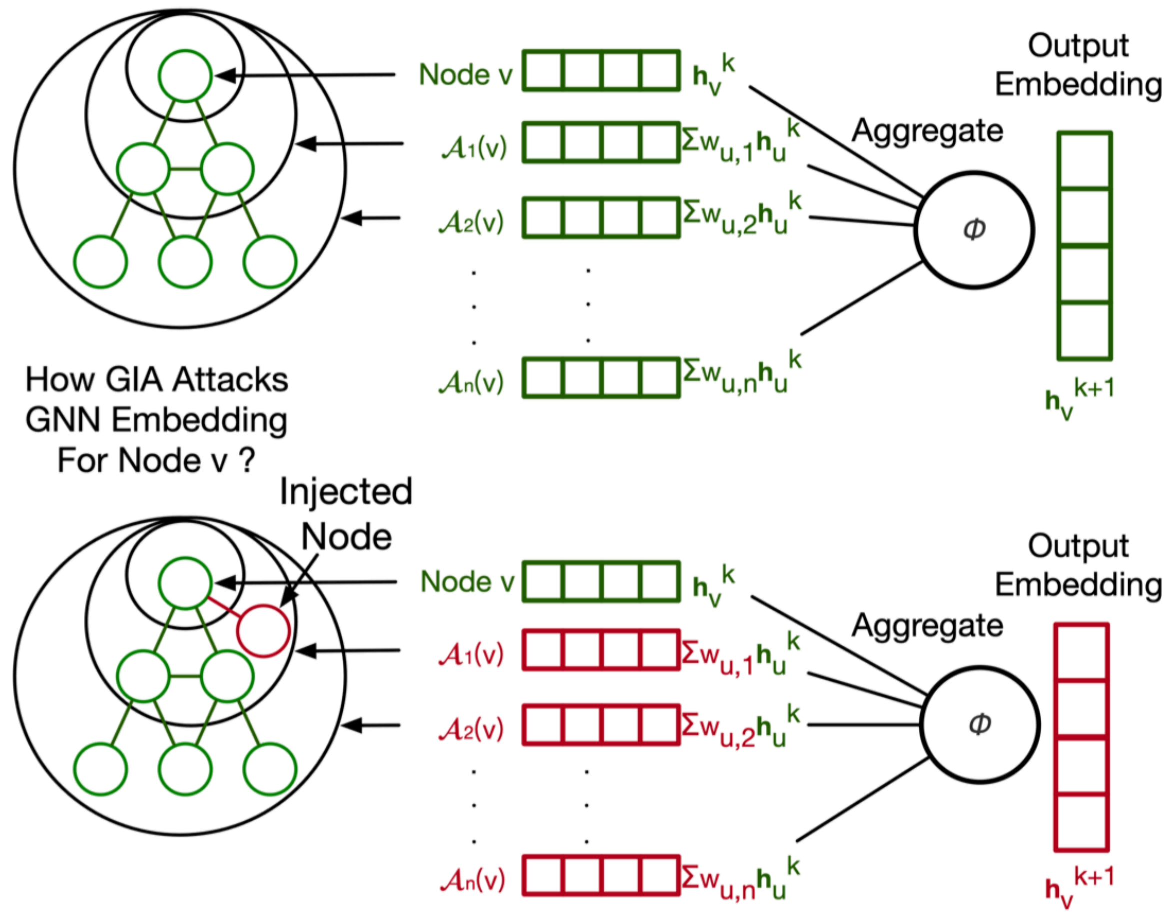
Adversarial Objective:

$$\min \underbrace{\mathcal{L}_{\text{atk}}}_{\text{Usually } -\mathcal{L}_{\text{sup}}}(f_{\theta^*}(G')), \quad \text{s.t.} \ \|G' - G\| \le \underbrace{\triangle}_{\text{perturbation budgets}}$$

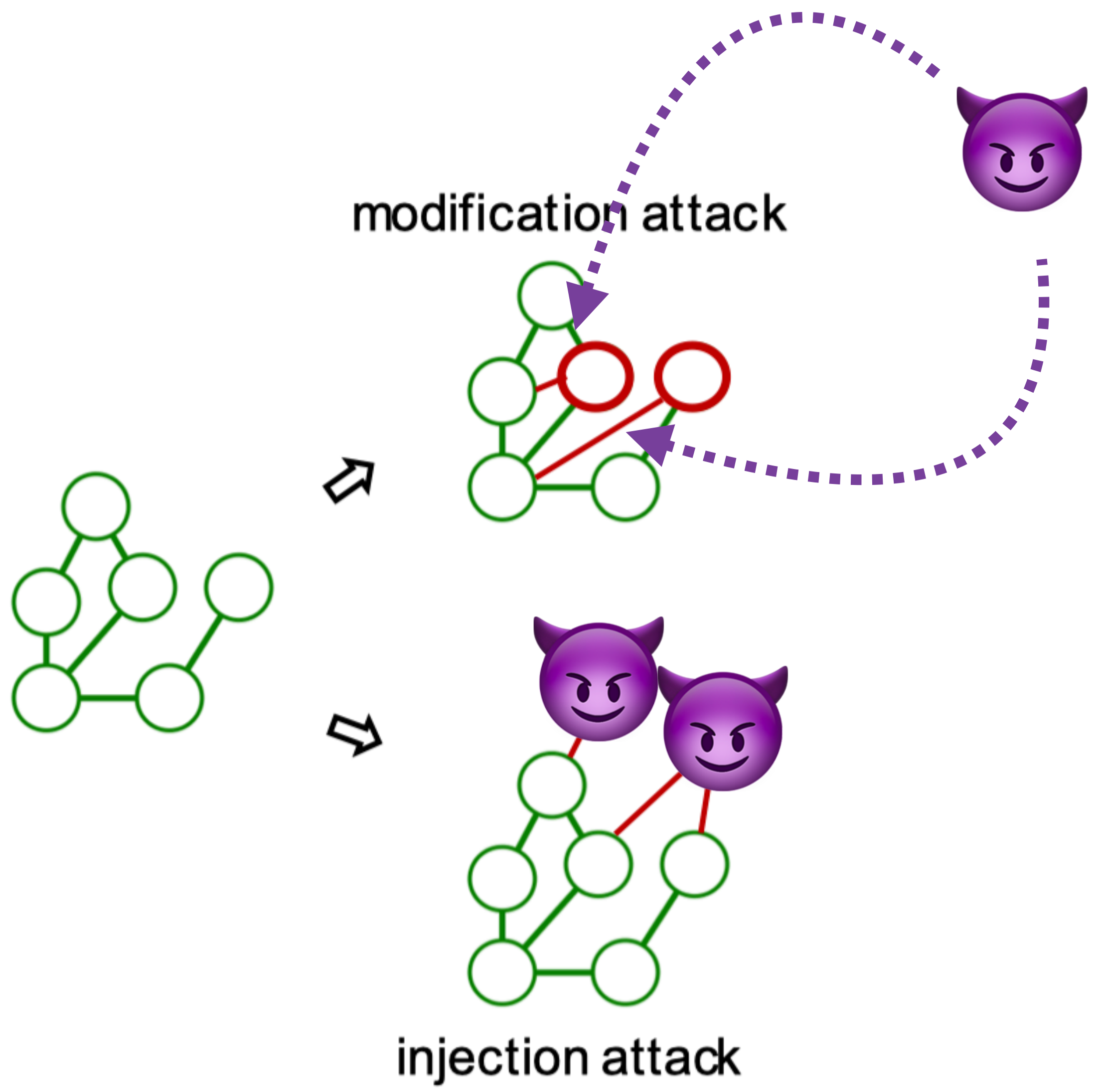
Adversarial Attacks on GNNs



(Zügner et al., 2018)



(Zou et al., 2020)



GMA vs. GIA

Adversarial Objective:

$$\min \mathcal{L}_{\text{atk}}(f_{\theta^*}(G')),\ \text{s.t.} \|G' - G\| \le \underbrace{\triangle}_{\text{perturbation budgets}}$$

Graph Modification Attack (GMA):

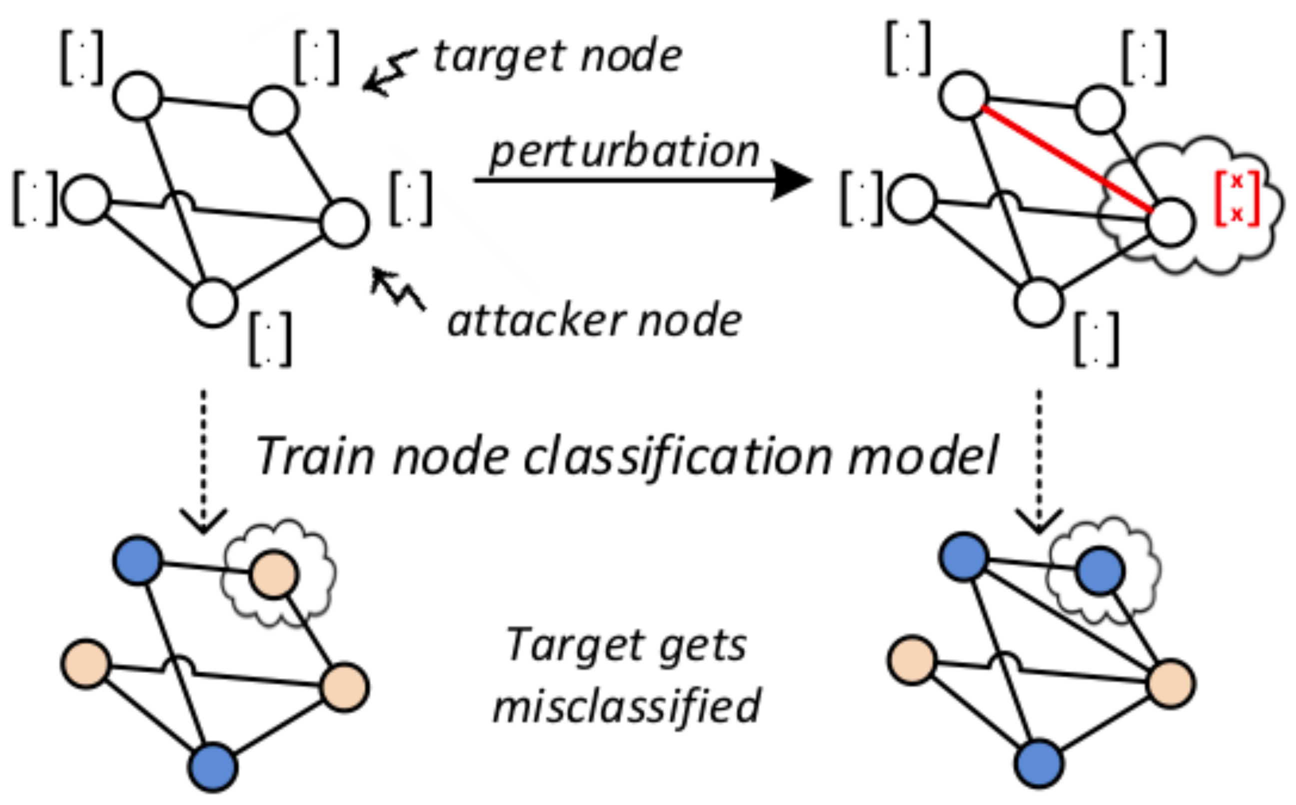
$$\triangle_A + \triangle_X \le \triangle \in \mathbb{Z},\ \|A' - A\|_0 \le \triangle_A \in \mathbb{Z},\ \|X' - X\|_\infty \le \epsilon \in \mathbb{R}$$

Sometimes Expensive

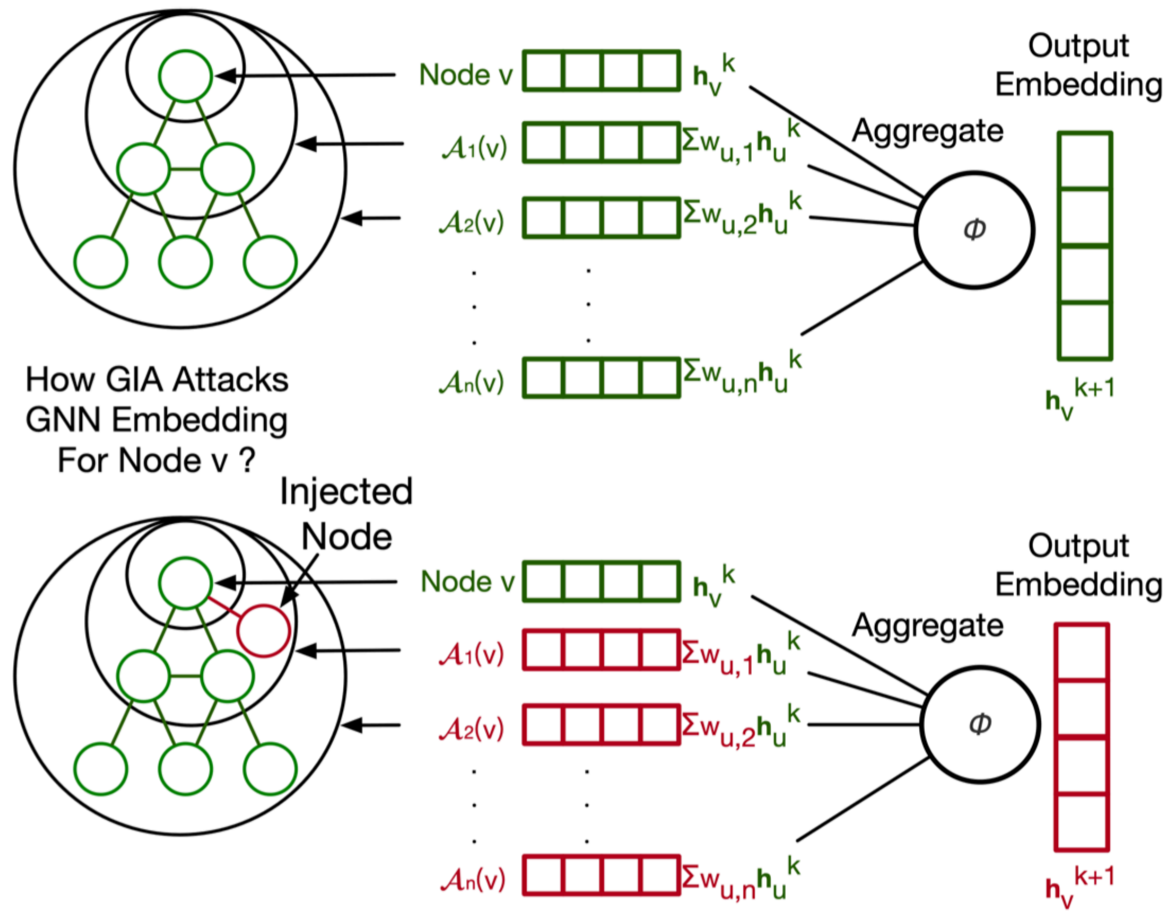
Modifying edges

Perturbing node features

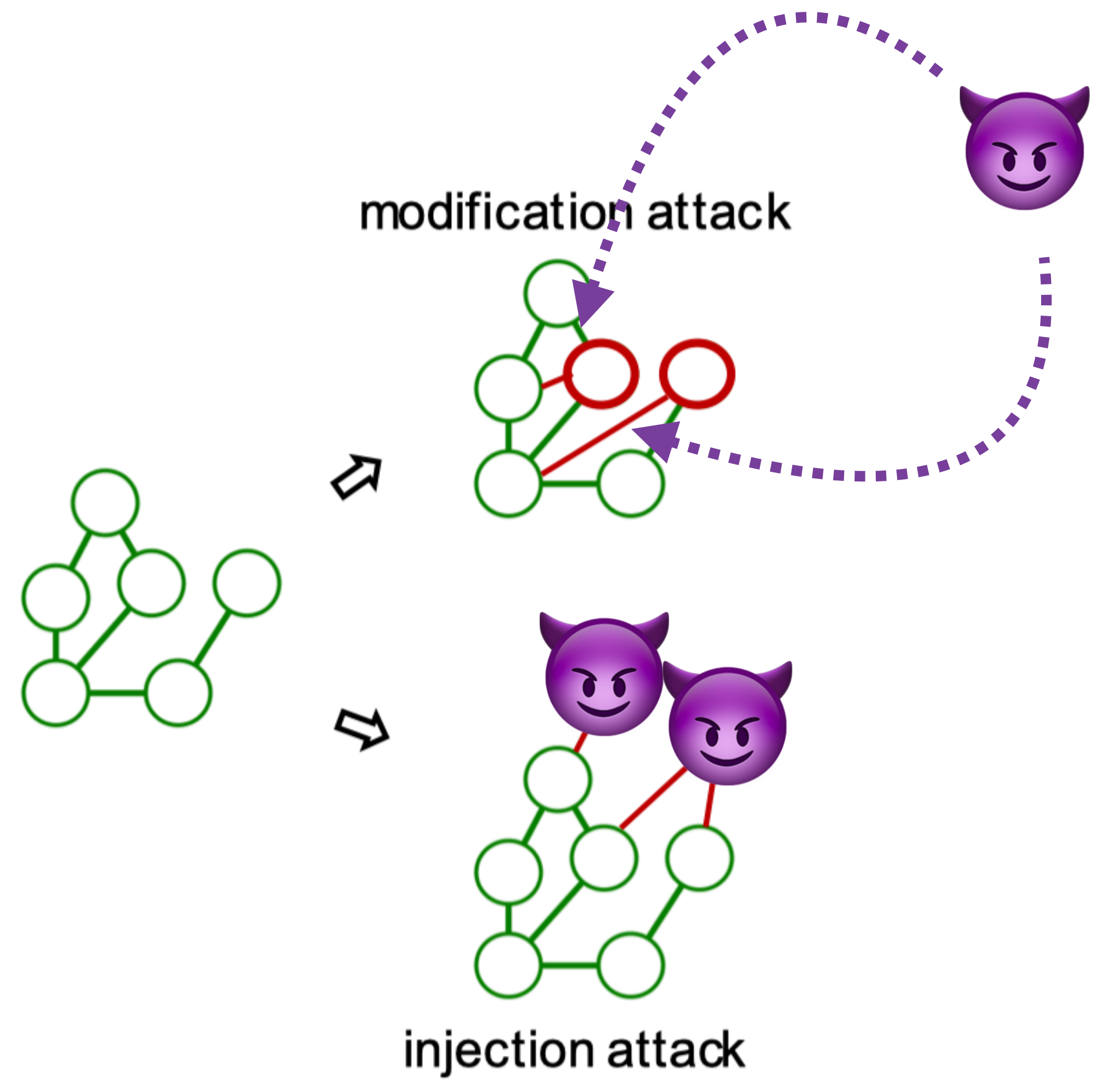
Adversarial Attacks on GNNs



(Zügner et al., 2018)



(Zou et al., 2020)



GMA vs. GIA

Adversarial Objective:

$$\min \mathscr{L}_{\mathrm{atk}}(f_{\theta^*}(G')), \ \ \mathrm{s.t.} \|G' - G\| \leq \underbrace{\triangle}$$

perturbation budgets

Graph Injection Attack (GIA):

$$X' = \begin{bmatrix} X \\ X_{\mathrm{atk}} \end{bmatrix}, A' = \begin{bmatrix} A & A_{\mathrm{atk}} \\ A_{\mathrm{atk}}^T & O_{\mathrm{atk}} \end{bmatrix}, \quad |V_{\mathrm{atk}}| \leq \triangle \in \mathbb{Z}, \ 1 \leq d_u \leq b \in \mathbb{Z}, X_u \in \mathscr{D}_X \subseteq \mathbb{R}^d, \forall u \in V_{\mathrm{atk}}$$

Practical

Injecting nodes

Carefully injected connections

Carefully crafted node features

Let’s find out more about GIA!

- through a friendly comparison -

🤺 GIA 🤔 🤺 GMA

The Power of Graph Injection Attack

Definition 1 (Threats)

Consider an adversary $\mathscr{A}$, given a perturbation budget $\triangle$, the threat of $\mathscr{A}$ to a GNN f_θ is defined as $\min_{\|G'-G\|\leq\triangle} \mathscr{L}_{\text{atk}}(f_{\theta^*}(G'))$, i.e., the **optimal objective value**.

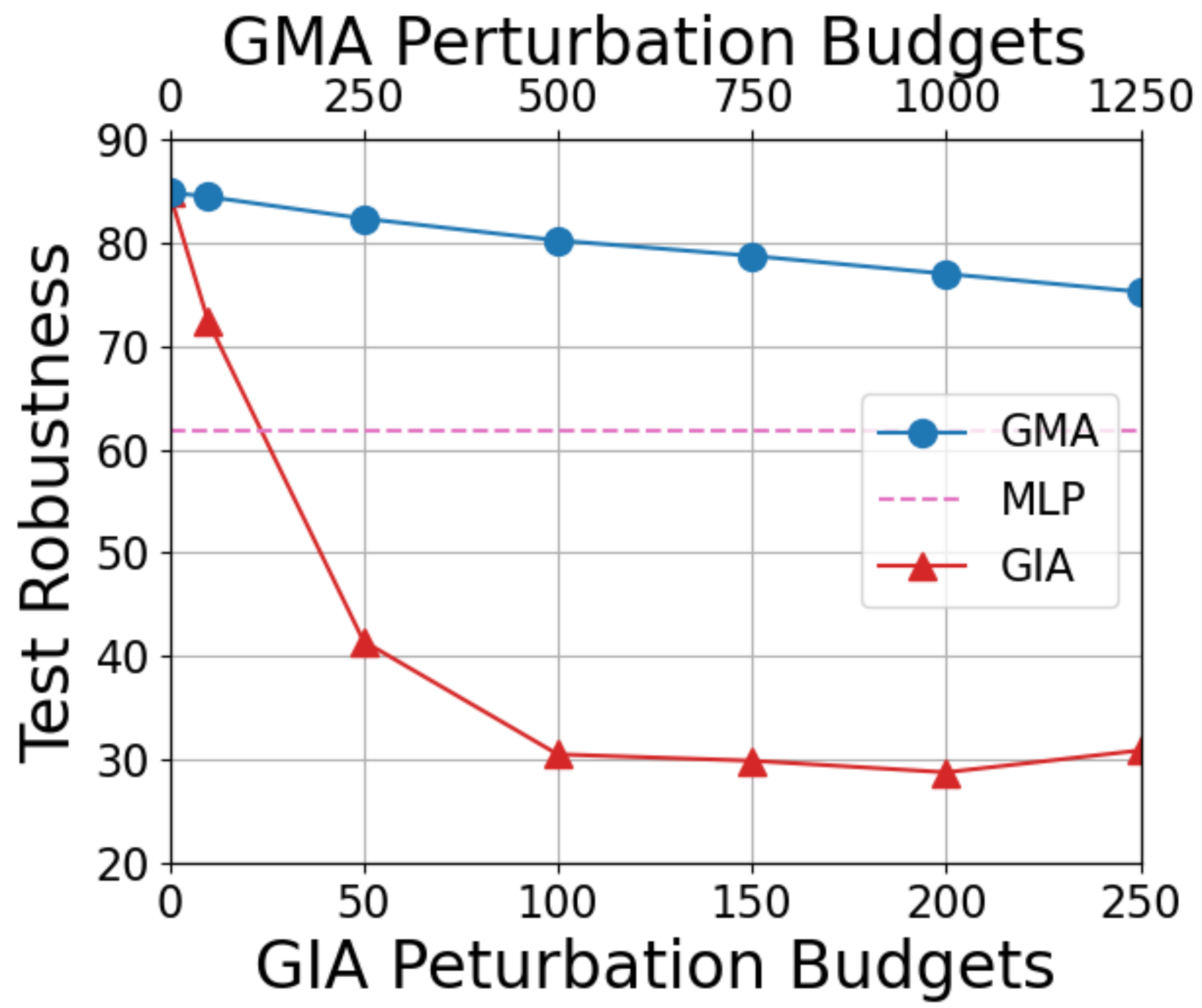
Theorem 1 (GIA is more harmful than GMA)

Given moderate perturbation budgets $\triangle_{\text{GIA}}$ for GIA and $\triangle_{\text{GMA}}$ for GMA, that is, let $\triangle_{\text{GIA}} \leq \triangle_{\text{GMA}} \ll |V| \leq |E|$, for a fixed linearized GNN f_θ trained on G, assume that G has no isolated nodes, and both GIA and GMA follow the **optimal strategy**, then, $\forall \triangle_{\text{GMA}} \geq 0, \exists \triangle_{\text{GIA}} \leq \triangle_{\text{GMA}}$,

$$\mathscr{L}_{\text{atk}}(f_\theta(G'_{\text{GIA}})) - \mathscr{L}_{\text{atk}}(f_\theta(G'_{\text{GMA}})) \leq 0,$$

where G'_{GIA} and G'_{GMA} are perturbed graphs generated by GIA and GMA, respectively.

The Power of Graph Injection Attack



GMA vs. GIA

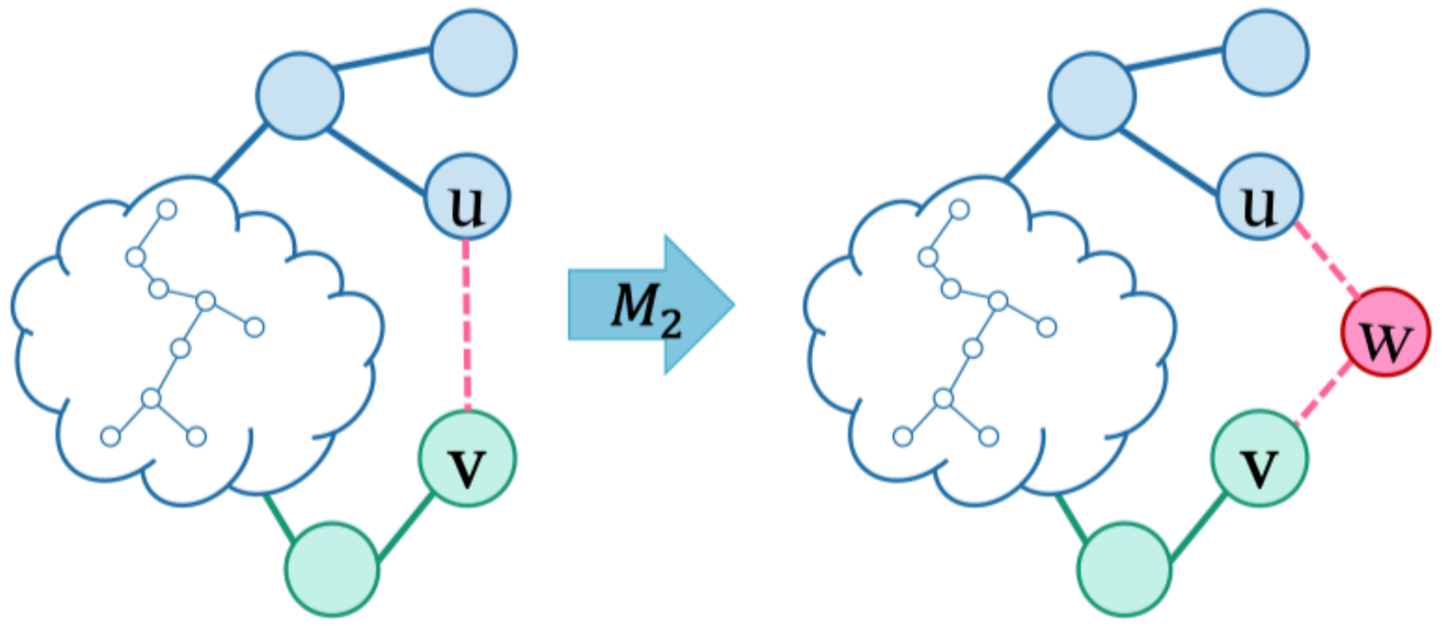
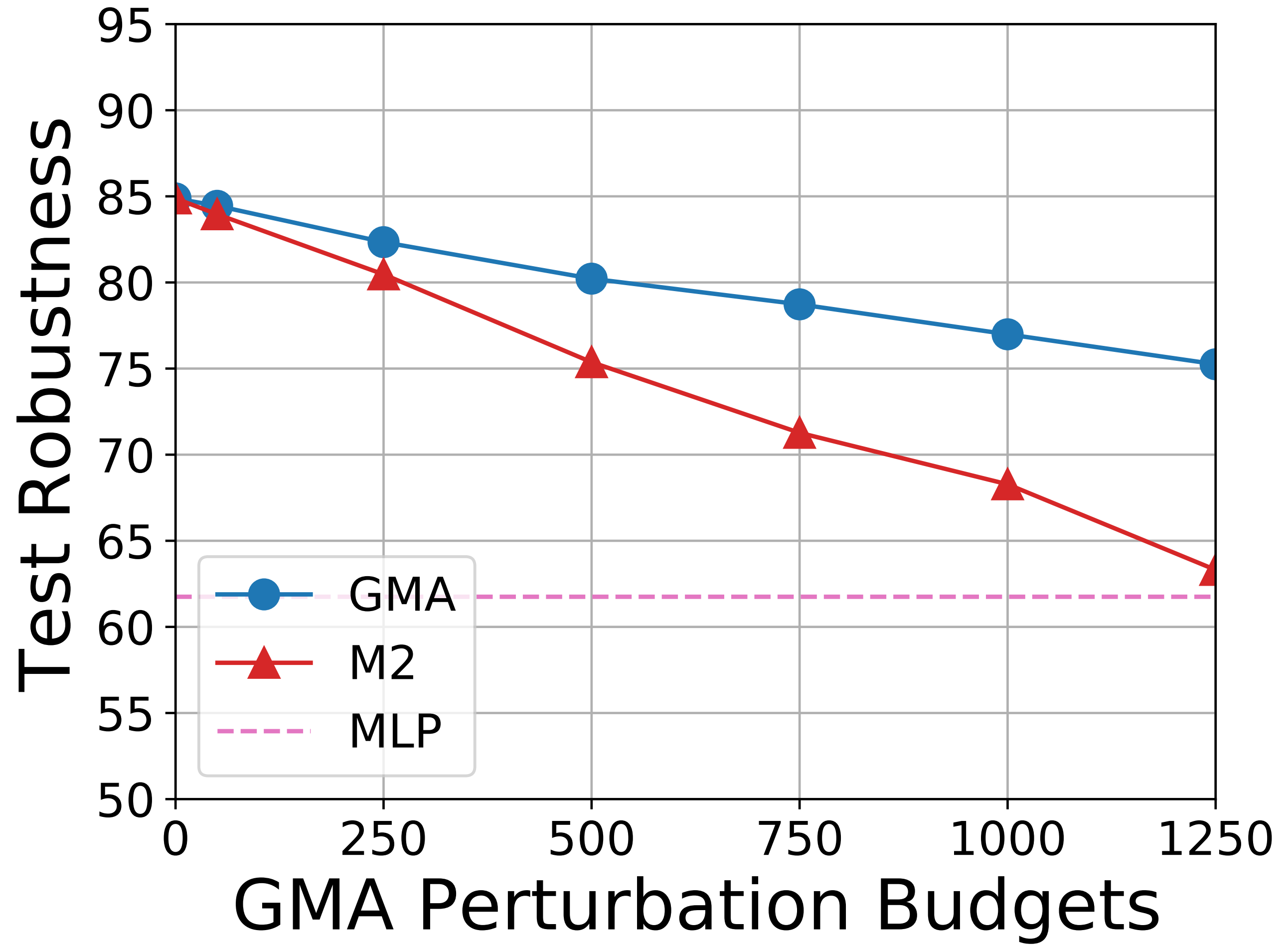


Illustration of $\mathscr{M}_2$ mapping



GMA vs. GIA with $\mathscr{M}_2$

Definition 2 (Plural Mapping $\mathscr{M}_2$)
A plural mapping $\mathscr{M}_2$ maps a perturbed graph G'_{GMA} generated by GMA with only edge addition perturbations*, to a GIA perturbed graph $G'_{\text{GIA}} = \mathscr{M}_2(G'_{\text{GMA}})$, such that:

$$f_\theta(G'_{\text{GIA}})_u = f_\theta(G'_{\text{GMA}})_u, \ \forall u \in V.$$

**We can also find such mappings for other perturbation actions of GMA.*

Is The Power of GIA A Free Lunch? 🤔

It turns out to be NO. 😅

The Pitfalls in Graph Injection Attack

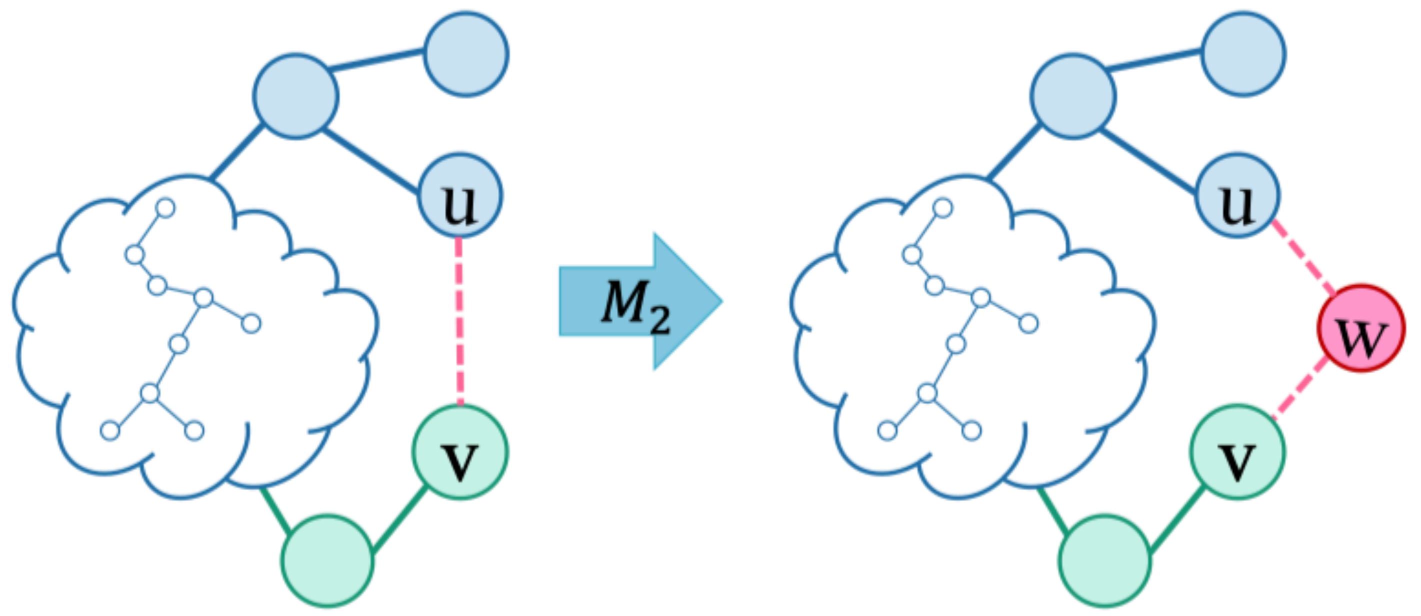


Illustration of $\mathscr{M}_2$ mapping

Given the example of $\mathscr{M}_2$, assume GIA uses PGD to optimize X_w iteratively, we find:

$$\text{sim}(X_u, X_w)^{(t+1)} \le \text{sim}(X_u, X_w)^{(t)},$$

where t is the number of optimization steps and $\text{sim}(\cdot)$ is the cosine similarity.

Definition 3 (Node-Centric Homophily)

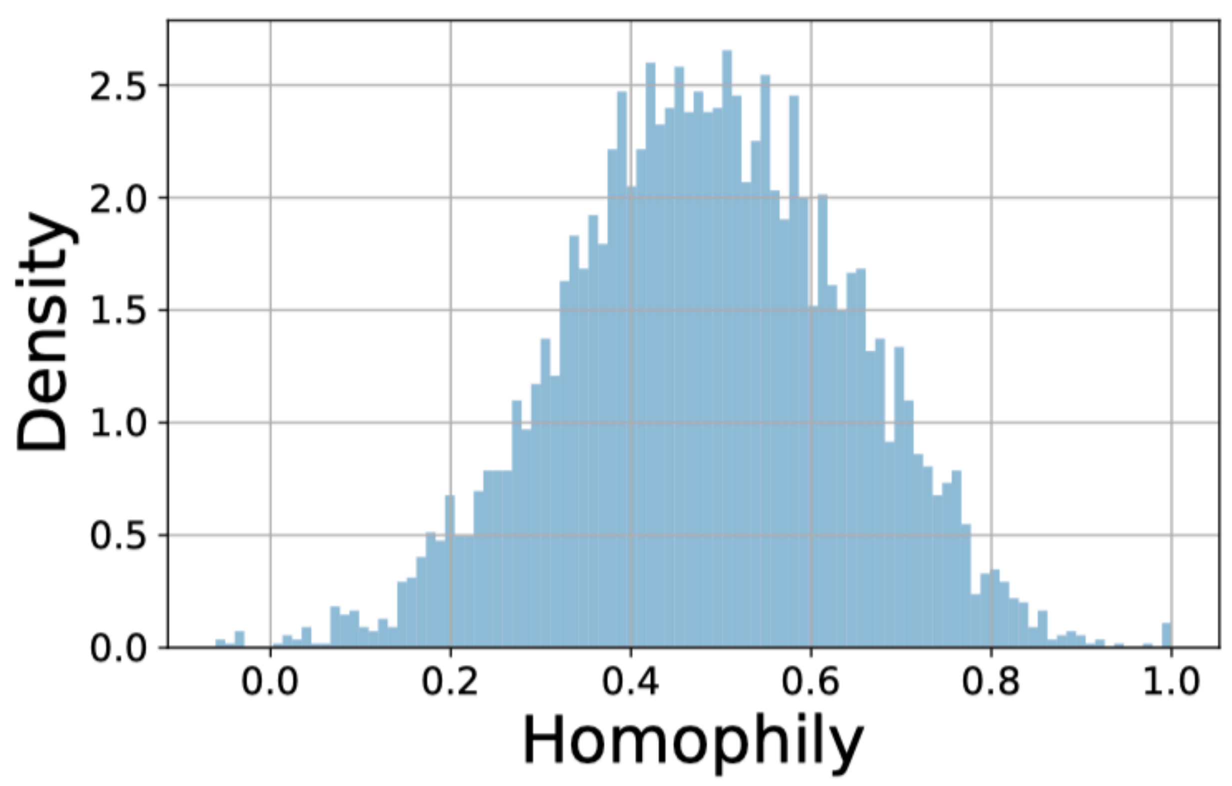
The homophily of a node u can be defined with the similarity between the features of node u and the aggregated features of its neighbors*:

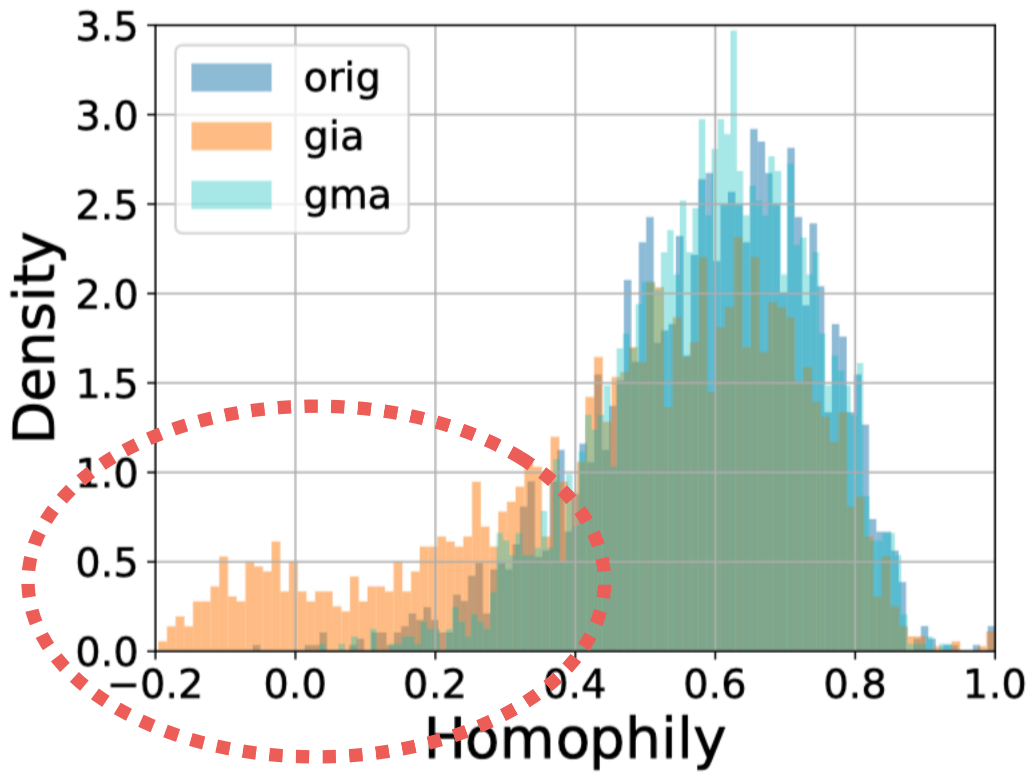
$$h_u = \text{sim}(r_u, X_u),\ r_u = \sum_{j \in \mathcal{N}(u)} \frac{1}{\sqrt{d_j d_u}} X_j,$$

where d_u is the degree of node u and $\text{sim}(\cdot)$ is a similarity metric, e.g., cosine similarity.

**We can also define edge-centric homophily, while we will focus on node-centric homophily.*

The Pitfalls in Graph Injection Attack





Homophily changes before and after attacks

GIA provably leads more damage to the homophily of the original graph than GMA

Definition 3 (Homophily Defenders)

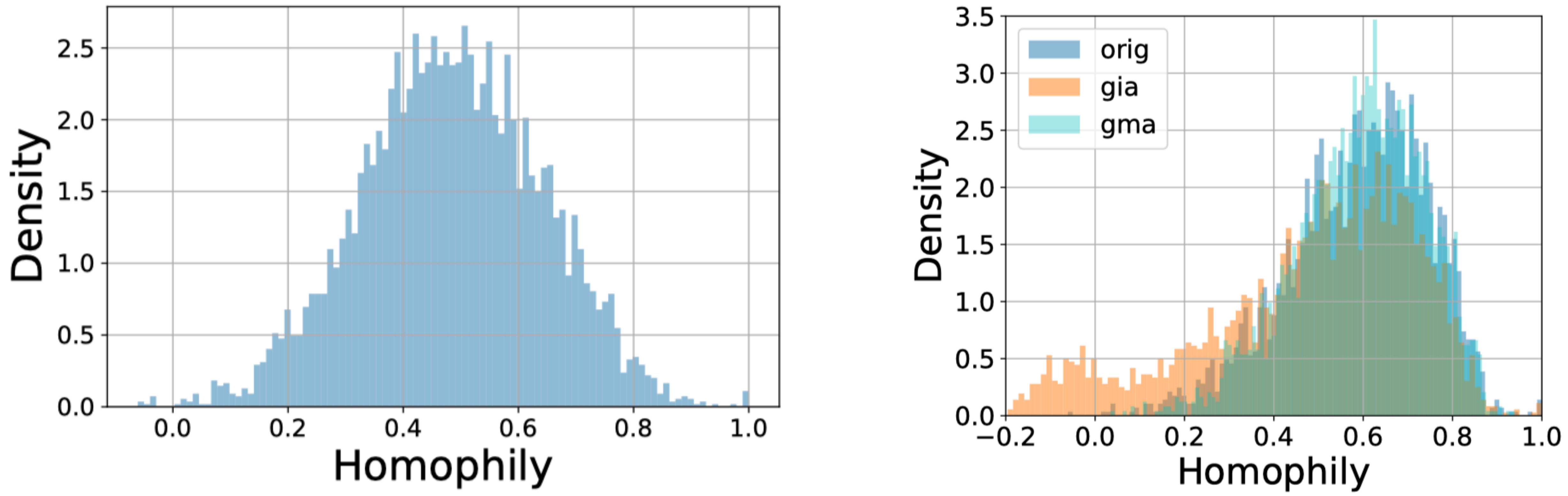
The homophily defenders can be implemented via edge pruning*:

$$H_u^{(k)} = \mathrm{READOUT}(W_k \cdot \mathrm{AGG}(\mathbb{1}_{\mathrm{con}}(u,v)\{H_v^{(k-1)} \mid v \in \mathcal{N}(u) \cup \{u\}\})),$$

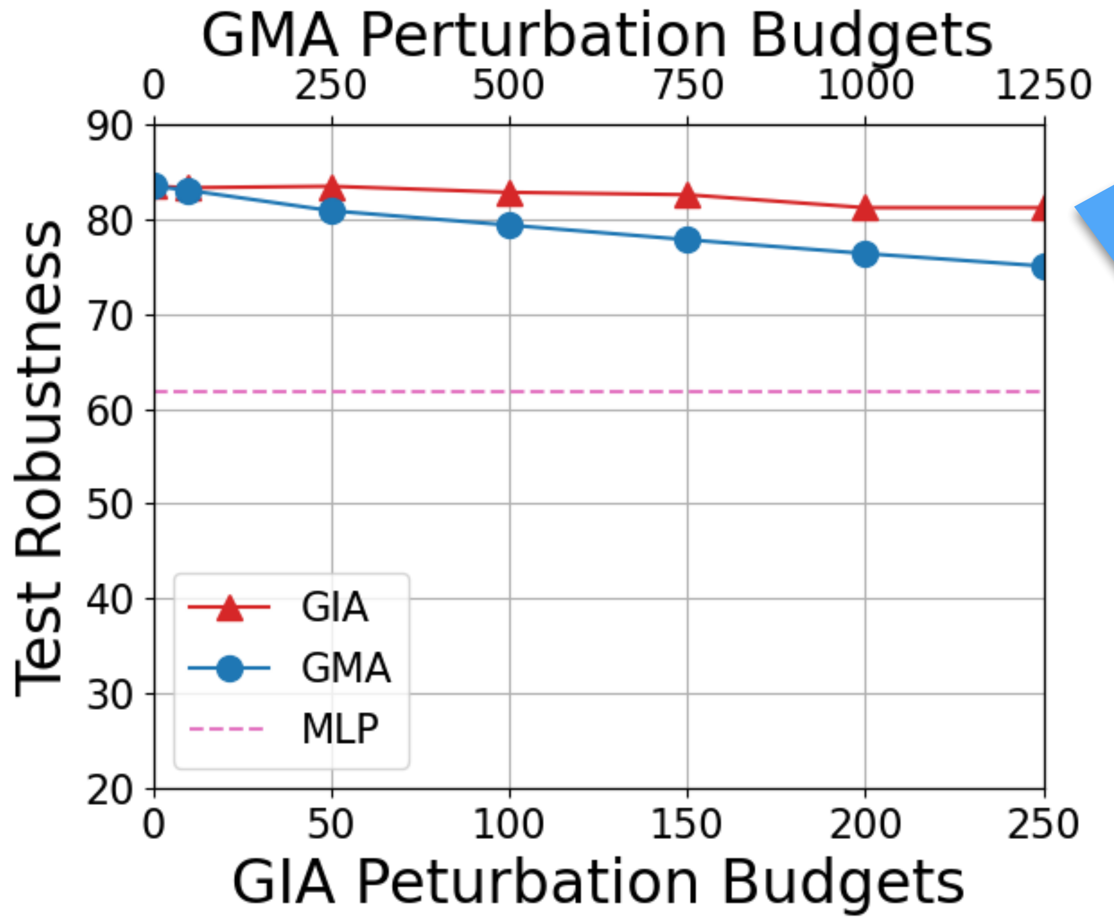
where $\mathbb{1}_{\mathrm{con}}(u,v)$ elaborates the pruning condition for edge (u,v).

**Essentially, homophily defenders can have other implementations than edge pruning.*

The Pitfalls in Graph Injection Attack



Homophily changes before and after attacks



GMA vs. GIA when with defense

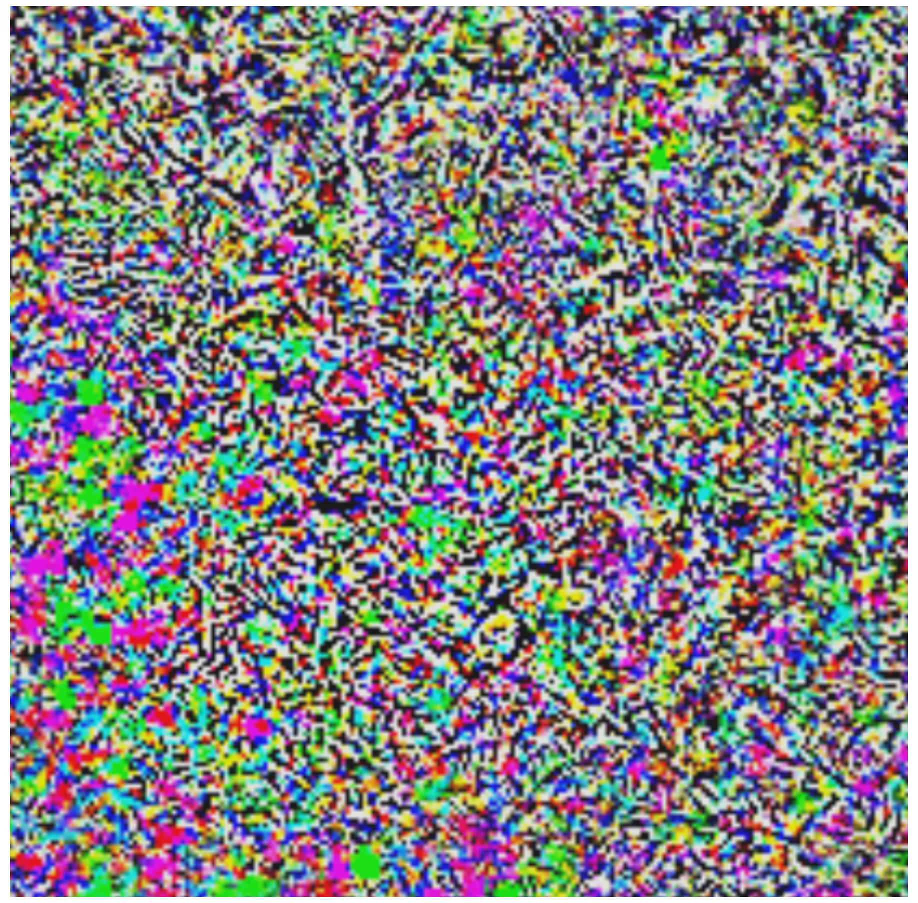
GIA almost loses its power!

Theorem 2 (GIA loses power when against homophily defenders)

Given conditions in Theorem 1, consider a GIA attack, which **(i)** is mapped by $\mathscr{M}_2$ from from a GMA attack that only performs edge addition perturbations, and **(ii)** uses a linearized GNN trained with at least one node from each class in G as the surrogate model, and **(iii)** optimizes the malicious node features with PGD. Assume that G has no isolated node, and has node features as $X_u = \frac{C}{C-1} e_{Y_u} - \frac{1}{C-1}\mathbf{1} \in \mathbb{R}^d$ where Y_u is the label of node u and $e_{Y_u} \in \mathbb{R}^d$ is a one-hot vector with the Y_u-th entry being 1 and others being 0. Let the minimum similarity for any pair of nodes connected in G be $s_G = \min_{(u,v)\in E} \mathrm{sim}(X_u, X_v)$ implemented with cosine similarity. For a homophily defender g_θ that prunes edges (u, v) if $\mathrm{sim}(X_u, X_v) \le s_G$, we have:

$$\mathscr{L}_{\mathrm{atk}}(g_\theta(\mathscr{M}_2(G'_{\mathrm{GMA}}))) - \mathscr{L}_{\mathrm{atk}}(g_\theta(G'_{\mathrm{GMA}})) \ge 0.$$

Unnoticeability in Graph Adversarial Attack

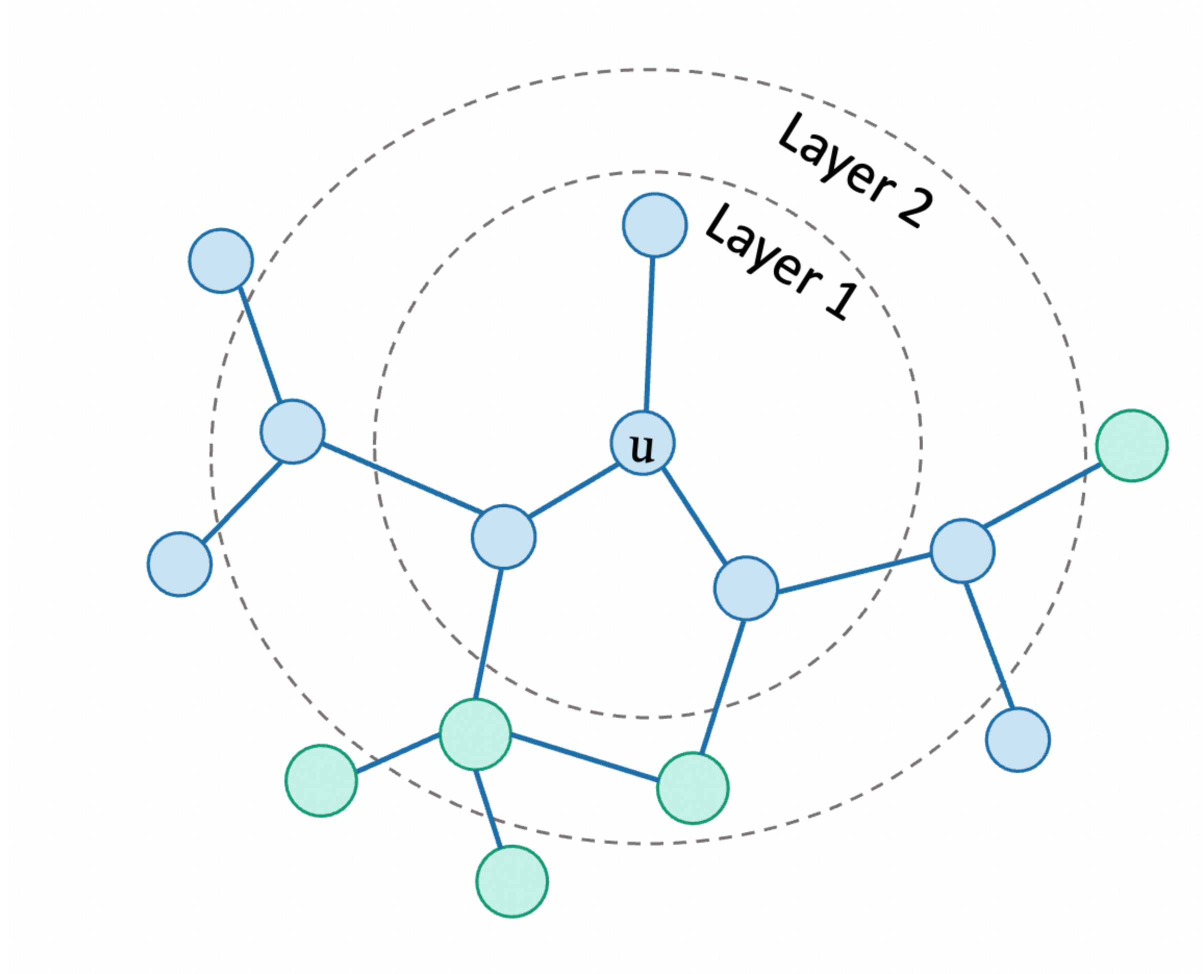


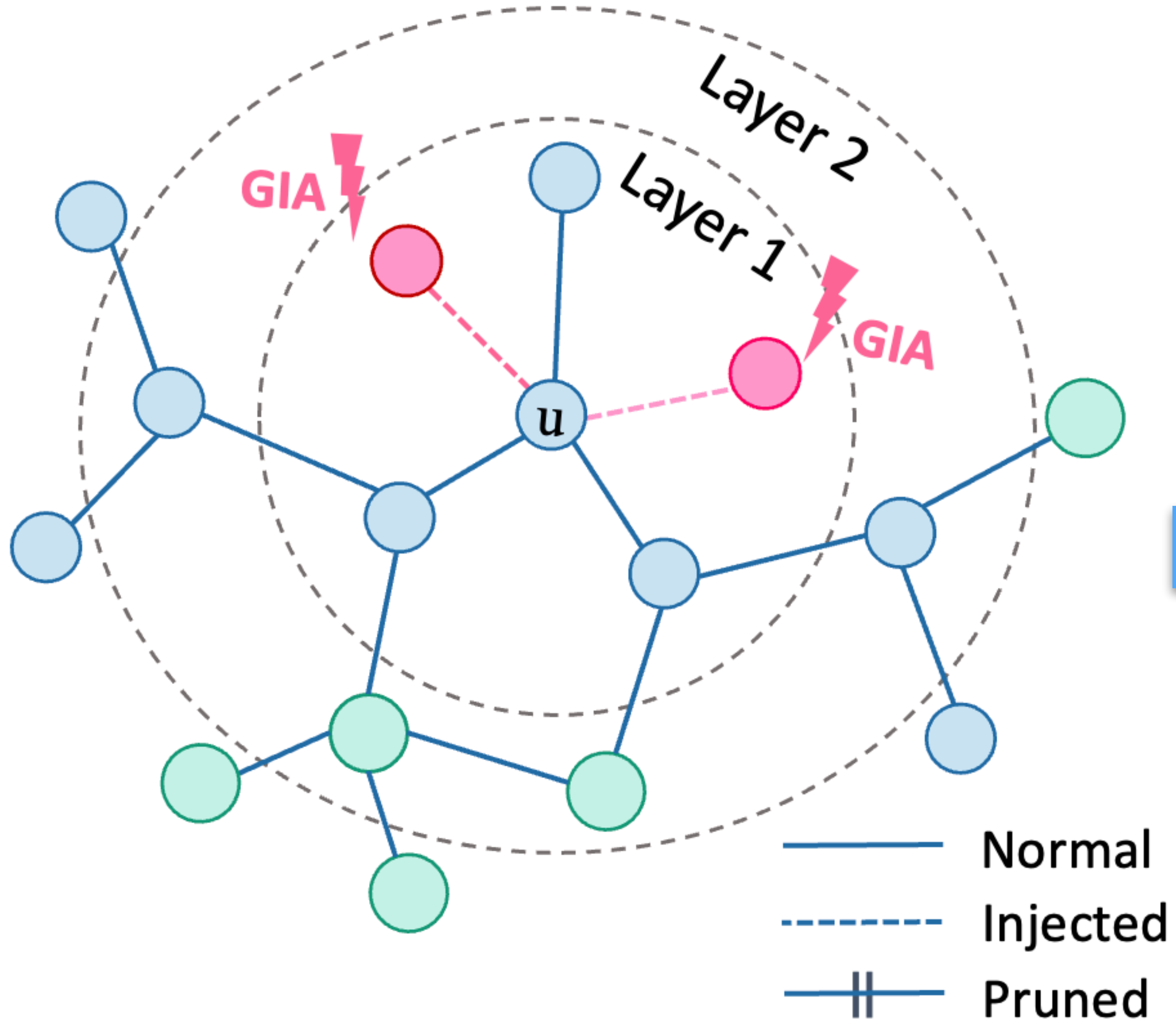
Prediction: Pig

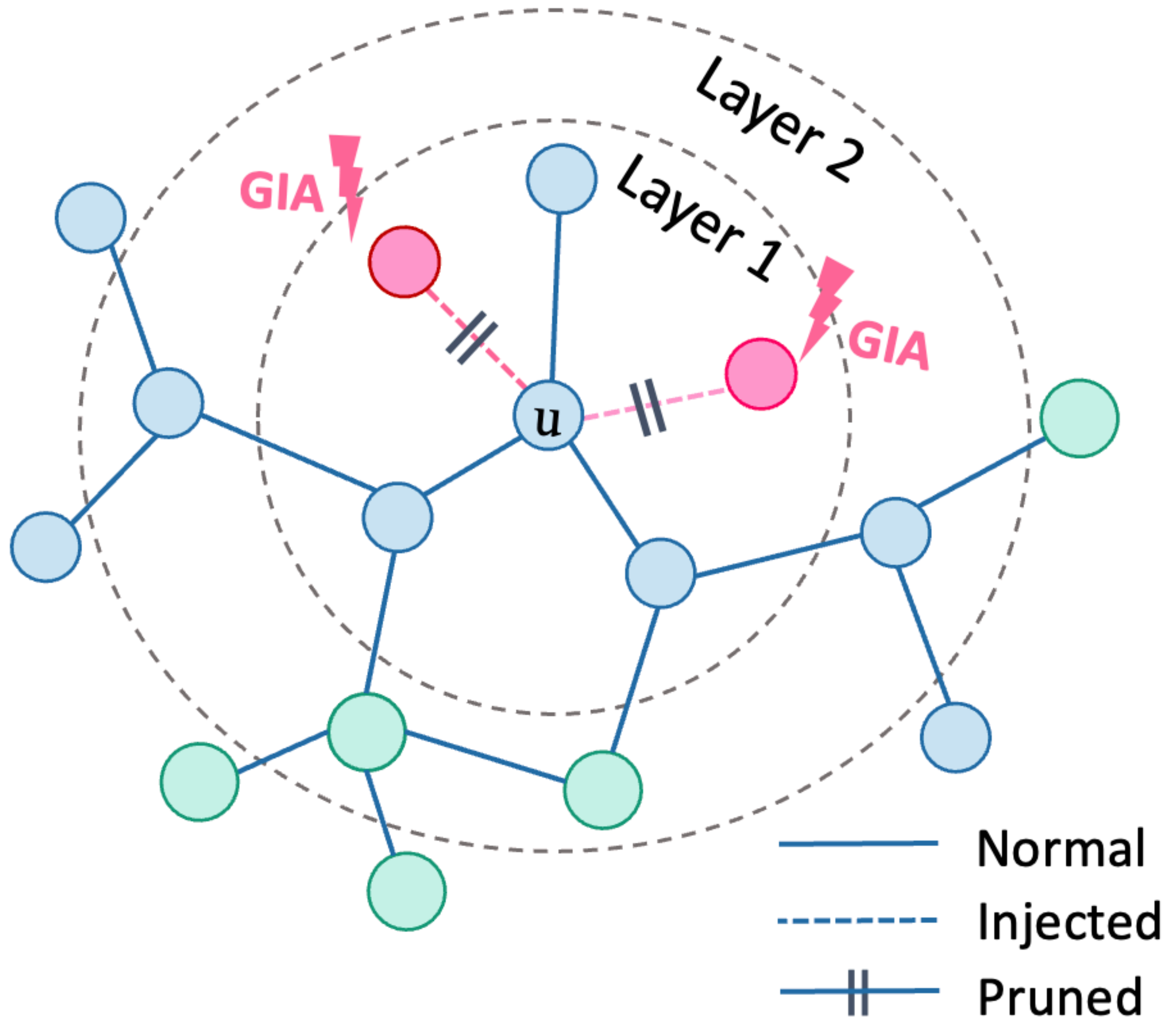
Unnoticeable Adversarial noise 😋

Prediction: Airliner

(Szegedy et al., 2014; Goodfellow et al., 2015; Kolter and Madry et al. 2019)







Unnoticeable Adversarial noise? 🤔

Homophily Unnoticeable Graph Injection Attack

Definition 4 (Homophily Unnoticeability)
Let the node-centric homophily distribution for a graph G be $\mathscr{H}_G$. Given the upper bound for the allowed homophily distribution shift $\triangle_{\mathscr{H}} \geq 0$, an attack $\mathscr{A}$ is **homophily unnoticeable** if:

$$m(\mathscr{H}_G, \mathscr{H}_{G'}) \leq \triangle_{\mathscr{H}},$$

where G' is the perturbed graph generated $\mathscr{A}$ and $m(\cdot)$ is a distribution distance measure.

Homophily Unnoticeability measures how likely the new connections between the malicious nodes and target nodes will appear *naturally*.

Homophily Defender provides efficient check for homophily unnoticeability serving as *external examiners*.

Homophily Unnoticeable Graph Injection Attack

Definition 5 (Harmonious Adversarial Objective (HAO))

Observing the homophily (Definition. 4) is differentiable with respect to X, we can integrate it into the original adversarial objective as*:

$$\min_{\|G'-G\|\leq\triangle} \mathcal{L}^h_{\text{atk}}(f_{\theta^*}(G')) = \mathcal{L}_{\text{atk}}(f_{\theta^*}(G')) - \lambda C(G, G'),$$

where $C(G, G')$ is a regularization term based on homophily and $\lambda \geq 0$ is the corresponding weight.

We only use HAO to solve for G' while still using the original objective to evaluate the threats.

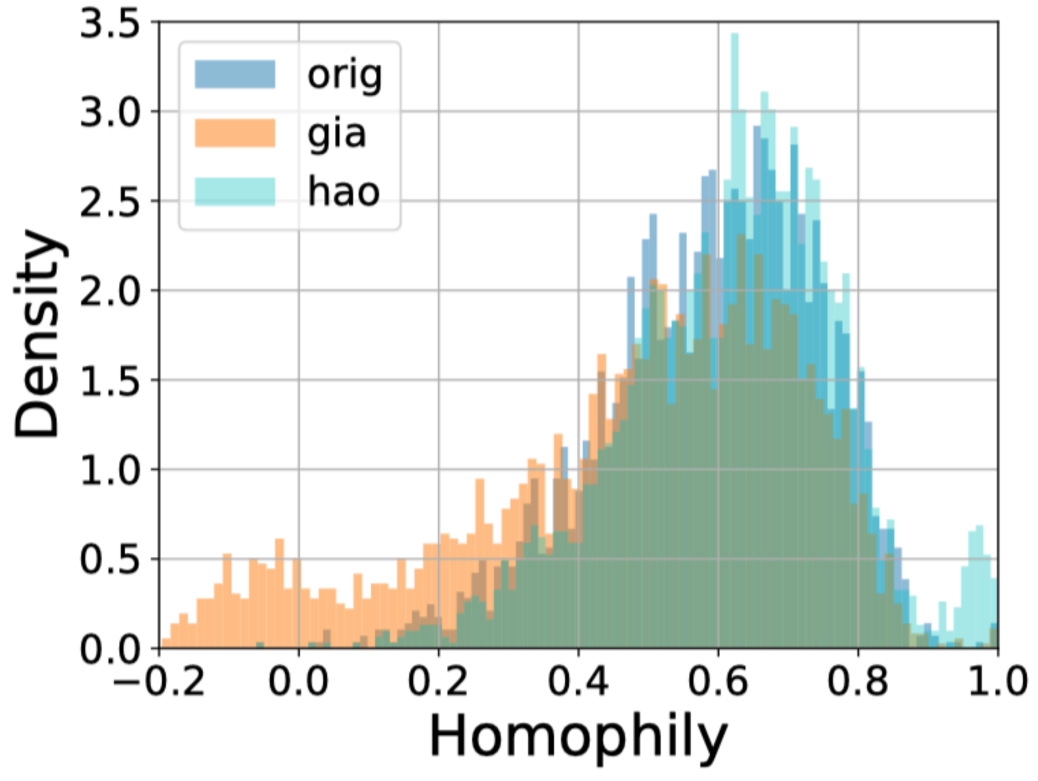
Theorem 3 (HAO re-empowers GIA)

Given conditions in Theorem 2, we have $m(\mathcal{H}_G, \mathcal{H}_{G'_{\text{HAO}}}) \leq m(\mathcal{H}_G, \mathcal{H}_{G'_{\text{GIA}}})$, hence:

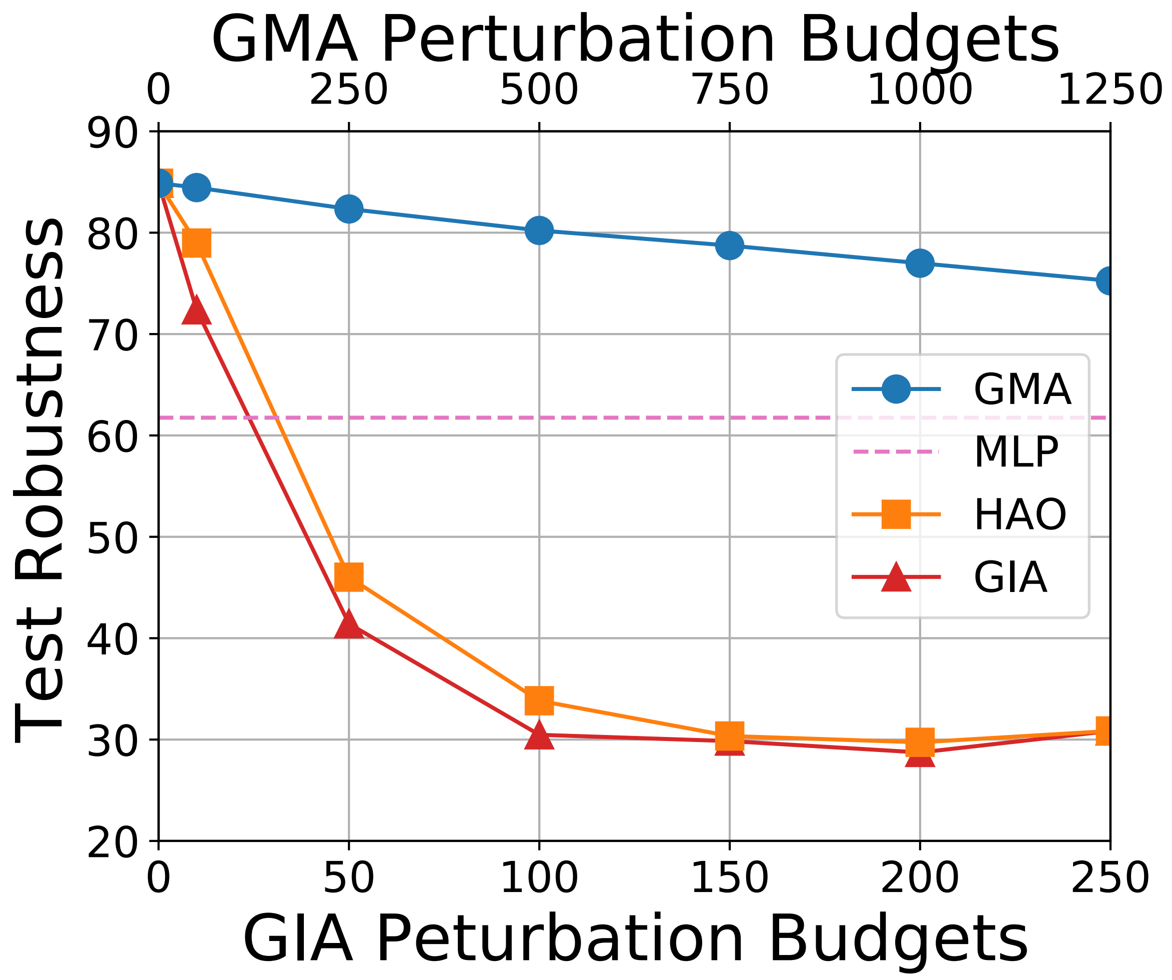
$$\mathcal{L}_{\text{atk}}(g_\theta(G'_{\text{HAO}})) - \mathcal{L}_{\text{atk}}(g_\theta(G'_{\text{GIA}})) \leq 0,$$

where G'_{HAO} and G'_{GIA} are perturbed graphs generated by GIA with and without HAO, respectively..

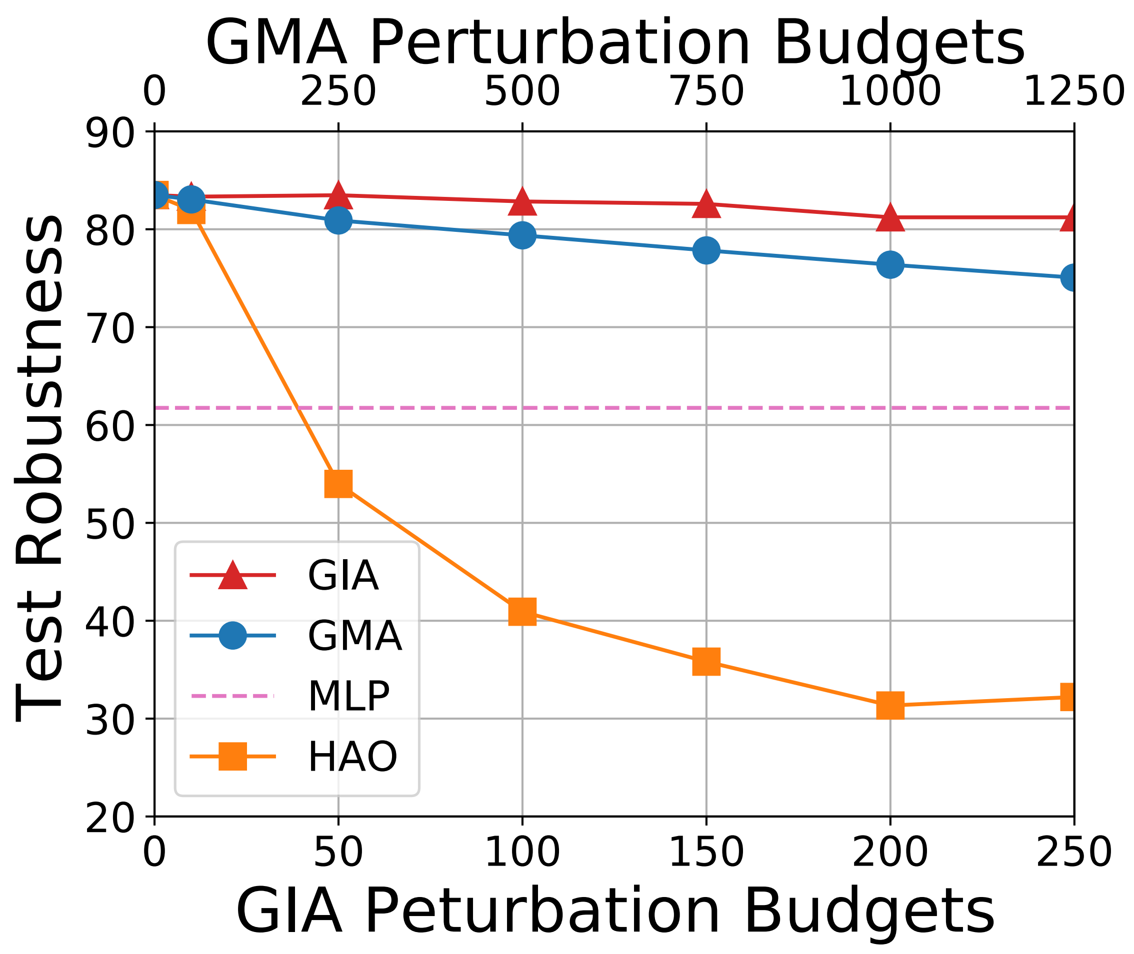
Homophily Unnoticeable Graph Injection Attack



Homophily changes



GMA vs. GIA without defense



GMA vs. GIA when with defense

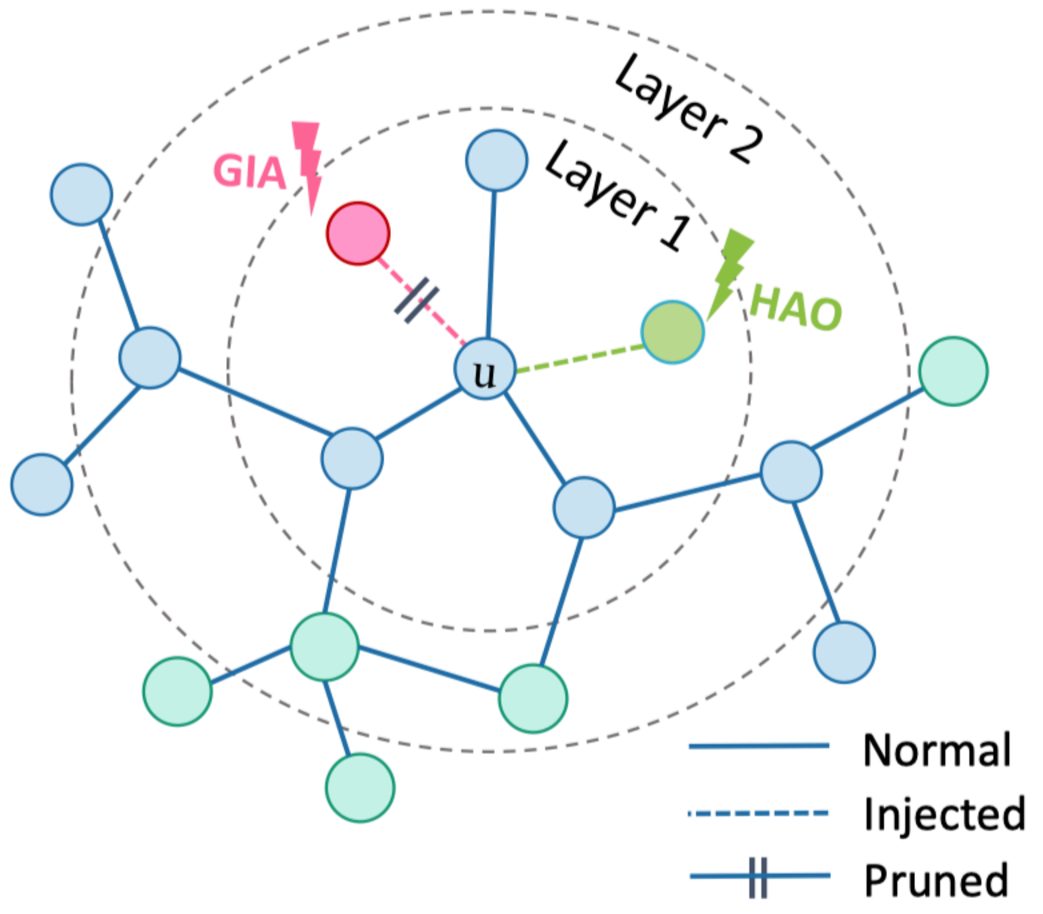


Illustration of GIA at node u

Theorem 3 (HAO re-empowers GIA)

Given conditions in Theorem 2, we have $m(\mathcal{H}_G, \mathcal{H}_{G'_{\mathrm{HAO}}}) \leq m(\mathcal{H}_G, \mathcal{H}_{G'_{\mathrm{GIA}}})$, hence:

$$\mathcal{L}_{\mathrm{atk}}(g_\theta(G'_{\mathrm{HAO}})) - \mathcal{L}_{\mathrm{atk}}(g_\theta(G'_{\mathrm{GIA}})) \leq 0,$$

where G'_{HAO} and G'_{GIA} are perturbed graphs generated by GIA with and without HAO, respectively..

HAO Re-empowers GIA

HAO significantly improves the performance of ***all* attacks on *all* datasets up to 30%**. Adaptive injection strategies can further advance the state of the art.

Homo: Homophily Defenders

Vanilla: Vanilla GNNs, e.g., GCN, GAT, GraphSage.

Robust: Robust GNN models, or GNN models with robust tricks such as layer normalisation, or adversarial training.

Combo: Robust GNN models with robust tricks such as layer normalisation, or adversarial training.

Table 1: Performance of non-targeted attacks against different models

	HAO	Cora (↓)			Citeseer(↓)			Computers(↓)			Arxiv(↓)		
		Homo	**Robust**	**Combo**	**Homo**	**Robust**	**Combo**	**Homo**	**Robust**	**Combo**	**Homo**	**Robust**	**Combo**
Clean		85.74	86.00	87.29	74.85	75.46	75.87	93.17	93.17	93.32	70.77	71.27	71.40
PGD		83.08	83.08	85.74	74.70	74.70	75.19	84.91	84.91	91.41	68.18	68.18	71.11
PGD	✓	52.60	62.60	77.99	69.05	69.05	73.04	79.33	79.33	87.83	55.38	62.89	68.68
MetaGIA†		83.61	83.61	85.86	74.70	74.70	75.15	84.91	84.91	91.41	68.47	68.47	71.09
MetaGIA†	✓	49.25	69.83	76.80	68.04	68.04	71.25	78.96	78.96	90.25	57.05	63.30	69.97
AGIA†		83.44	83.44	85.78	74.72	74.72	75.29	85.21	85.21	91.40	68.07	68.07	71.01
AGIA†	✓	47.24	**61.59**	**75.25**	70.24	70.24	71.80	75.14	75.14	86.02	59.32	65.62	69.92
TDGIA		83.44	83.44	85.72	74.76	74.76	75.26	88.32	88.32	91.40	64.49	64.49	70.97
TDGIA	✓	56.95	73.38	79.45	**60.91**	**60.91**	72.51	**74.77**	**74.77**	90.42	49.36	**60.72**	**63.57**
ATDGIA		83.07	83.07	85.39	74.72	74.72	75.12	86.03	86.03	91.41	66.95	66.95	71.02
ATDGIA	✓	**42.18**	70.30	76.87	61.08	61.08	**71.22**	80.86	80.86	**84.60**	**45.59**	63.30	64.31
MLP			61.75			65.55			84.14			52.49	

↓The lower number indicates better attack performance. †Runs with SeqGIA framework on Computers and Arxiv.

We evaluate with **38** defense models and report the *maximum* mean test robustness from multiple runs.

HAO Re-empowers GIA

HAO significantly improves the performance of ***all* attacks on *all* datasets up to 15%**. Adaptive injection strategies can further advance the state of the art.

Homo: Homophily Defenders

Vanilla: Vanilla GNNs, e.g., GCN, GAT, GraphSage.

Robust: Robust GNN models, or GNN models with robust tricks such as layer normalisation, or adversarial training.

Combo: Robust GNN models with robust tricks such as layer normalisation, or adversarial training.

Table 2: Performance of targeted attacks against different models

		Computers(↓)			Arxiv(↓)			Aminer(↓)			Reddit(↓)		
	HAO	**Homo**	**Robust**	**Combo**	**Homo**	**Robust**	**Combo**	**Homo**	**Robust**	**Combo**	**Homo**	**Robust**	**Combo**
Clean		92.68	92.68	92.83	69.41	71.59	72.09	62.78	66.71	66.97	94.05	97.15	97.13
PGD		88.13	88.13	91.56	69.19	69.19	71.31	53.16	53.16	56.31	92.44	92.44	93.03
PGD	✓	71.78	71.78	85.81	**36.06**	**37.22**	69.38	34.62	34.62	39.47	56.44	86.12	**84.94**
MetaGIA†		87.67	87.67	91.56	69.28	69.28	71.22	48.97	48.97	52.35	92.40	92.40	93.97
MetaGIA†	✓	70.21	71.61	85.83	38.44	38.44	48.06	41.12	41.12	45.16	**46.75**	90.06	90.78
AGIA†		87.57	87.57	91.58	66.19	66.19	70.06	50.50	50.50	53.69	91.62	91.62	93.66
AGIA†	✓	**69.96**	**71.58**	85.72	38.84	38.84	68.97	35.94	35.94	42.66	80.69	88.84	90.44
TDGIA		87.21	87.21	91.56	63.66	63.66	71.06	51.34	51.34	54.82	92.19	92.19	93.62
TDGIA	✓	71.39	71.62	**77.15**	42.56	42.56	42.53	25.78	25.78	29.94	78.16	**85.06**	88.66
ATDGIA		87.85	87.85	91.56	66.12	66.12	71.16	50.87	50.87	53.68	91.25	91.25	93.03
ATDGIA	✓	72.00	72.53	78.35	38.28	40.81	**39.47**	**22.50**	**22.50**	**28.91**	64.09	89.06	88.91
MLP			84.11			52.49			32.80			70.69	

↓The lower number indicates better attack performance. †Runs with SeqGIA framework.

We evaluate with **38** defense models and report the *maximum* mean test robustness from multiple runs.

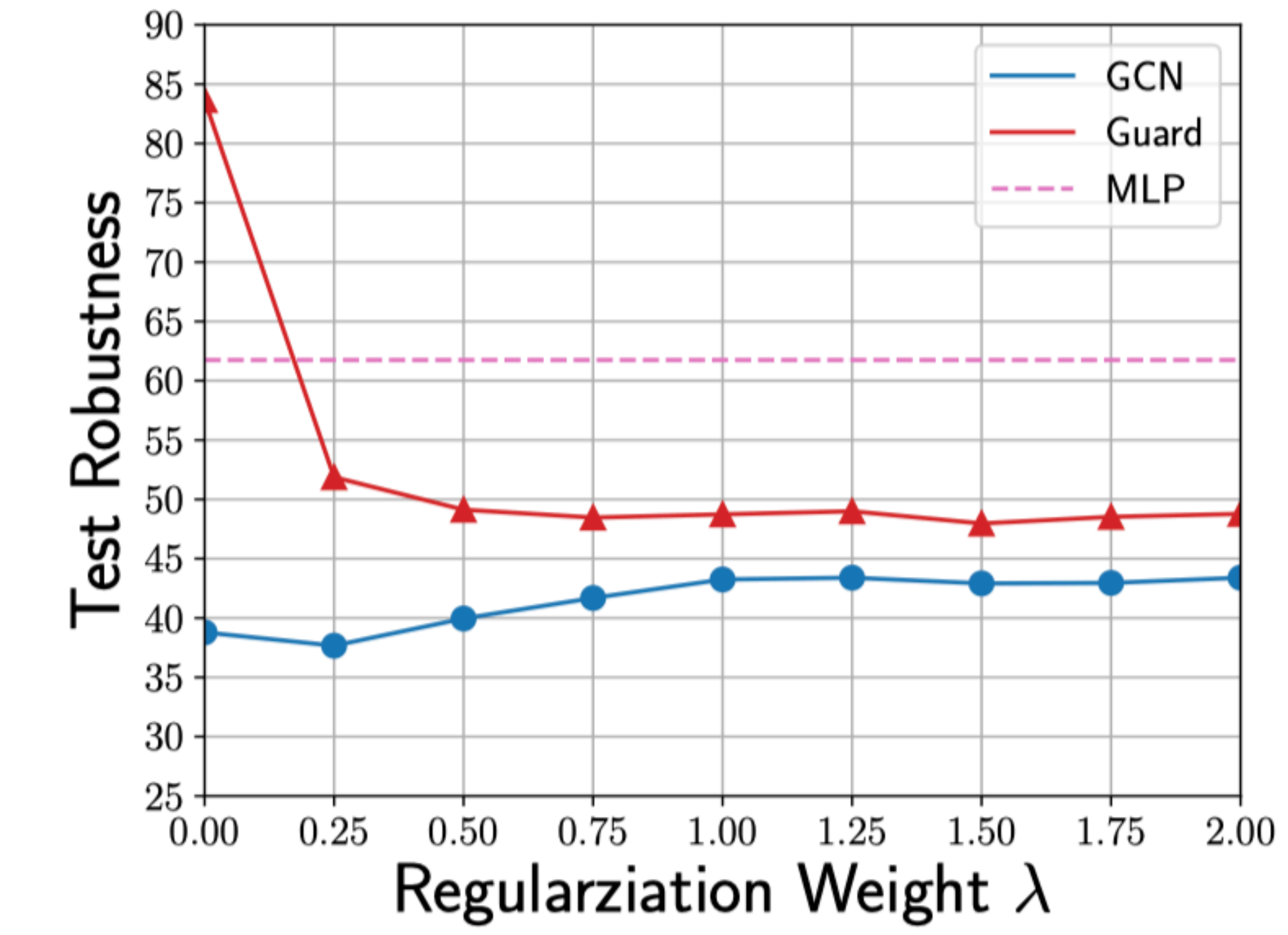
HAO Re-empowers GIA

HAO consistently improves the performances of ***all*** **attacks on** ***all*** **datasets up to 5%**. Adaptive injection strategies can further advance the state of the art.

Table 9: Full Averaged performance across all defense models

Model	Cora†	Citeseer†	Computers†	Arxiv†	Arxiv‡	Computers‡	Aminer‡	Reddit‡
Clean	84.74	74.10	92.25	70.44	70.44	91.68	62.39	95.51
PGD	61.09	54.08	61.75	54.23	36.70	62.41	26.13	62.72
+HAO	56.63	48.12	59.16	45.05	28.48	59.09	22.15	56.99
MetaGIA	60.56	53.72	61.75	53.69	28.78	62.08	32.78	60.14
+HAO	58.51	47.44	60.29	48.48	24.61	58.63	29.91	**54.14**
AGIA	60.10	54.55	60.66	48.86	32.68	61.98	31.06	59.96
+HAO	**53.79**	48.30	**58.71**	48.86	29.52	**58.37**	26.51	56.36
TDGIA	66.86	52.45	66.79	49.73	31.68	62.47	32.37	57.97
+HAO	65.22	46.61	65.46	49.54	**22.04**	59.67	22.32	54.32
ATDGIA	61.14	49.46	65.07	46.53	32.08	64.66	24.72	61.25
+HAO	58.13	**43.41**	63.31	**44.40**	29.24	59.27	**17.62**	56.90

The lower is better. †Non-targeted attack. ‡Targeted attack.



Varying λ in HAO

We evaluate with **38** defense models and report the mean test robustness of all models from multiple runs.

Summary

We provide a formal comparison between GIA and GMA in a unified setting and find that GIA can be provably more harmful than GMA due to its high flexibility (Theorem 1).

However, the flexibility of GIA will cause severe damage to the homophily which makes GIA easily defendable by homophily defenders (Theorem 2).

To mitigate the issue, we introduce the concept of homophily unnoticeability and a novel objective HAO to conduct homophily unnoticeable attacks (Theorem 3).

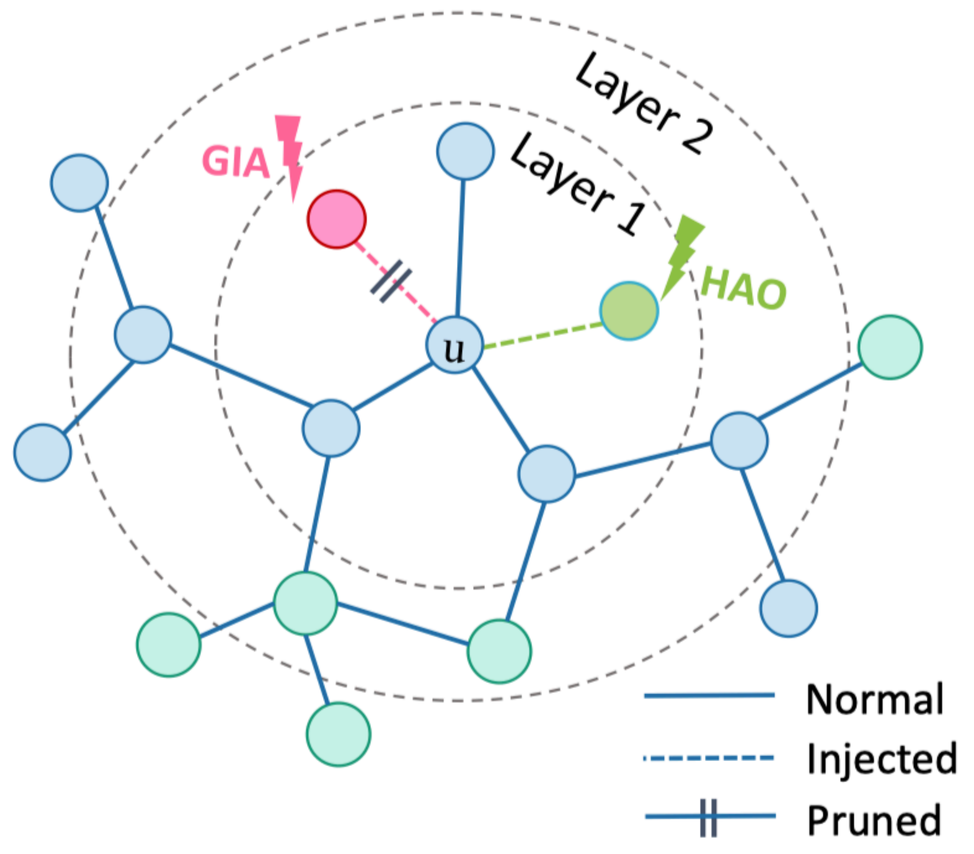
Paper

Code

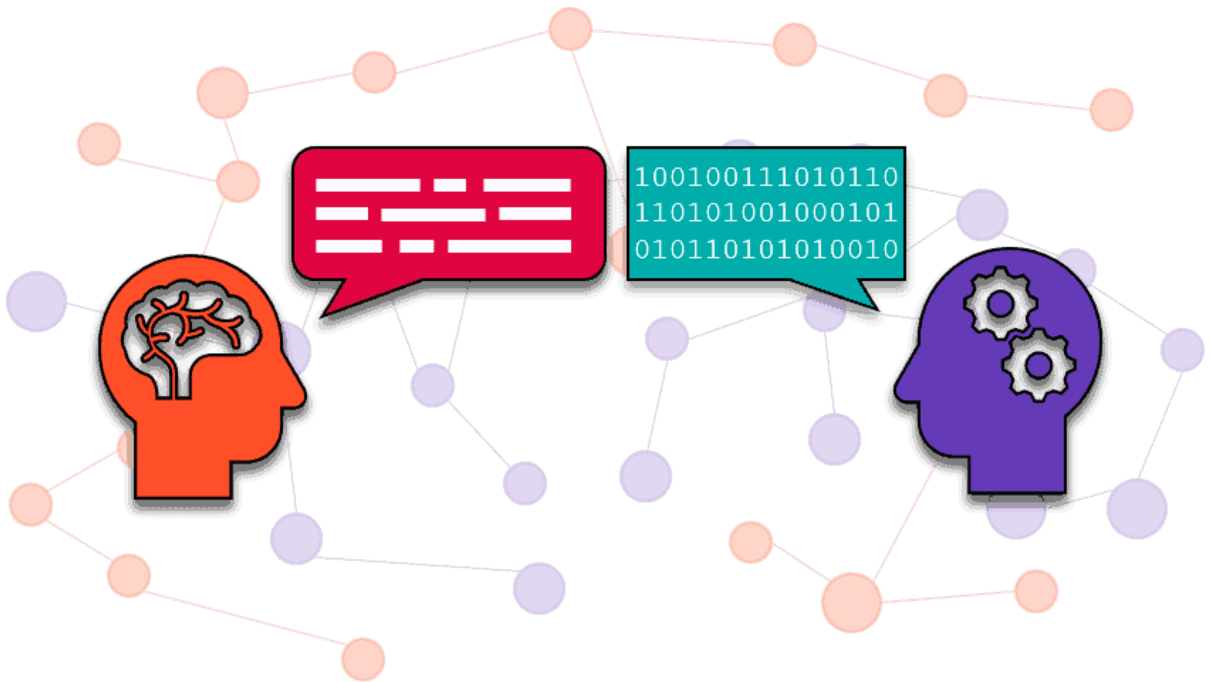
Thank you!

Contact: yqchen@cse.cuhk.edu.hk

Future Navigations



If you focus on graph domain: more advanced injection strategies, more downstream tasks, more attack scenarios, more robust GNNs…



If you focus on other domains: more unnoticeability constraints & the corresponding external examiners…

Paper

Code

Thank you!

Contact: yqchen@cse.cuhk.edu.hk

Figure source: navigate360