中国人民大學
RENMIN UNIVERSITY OF CHINA

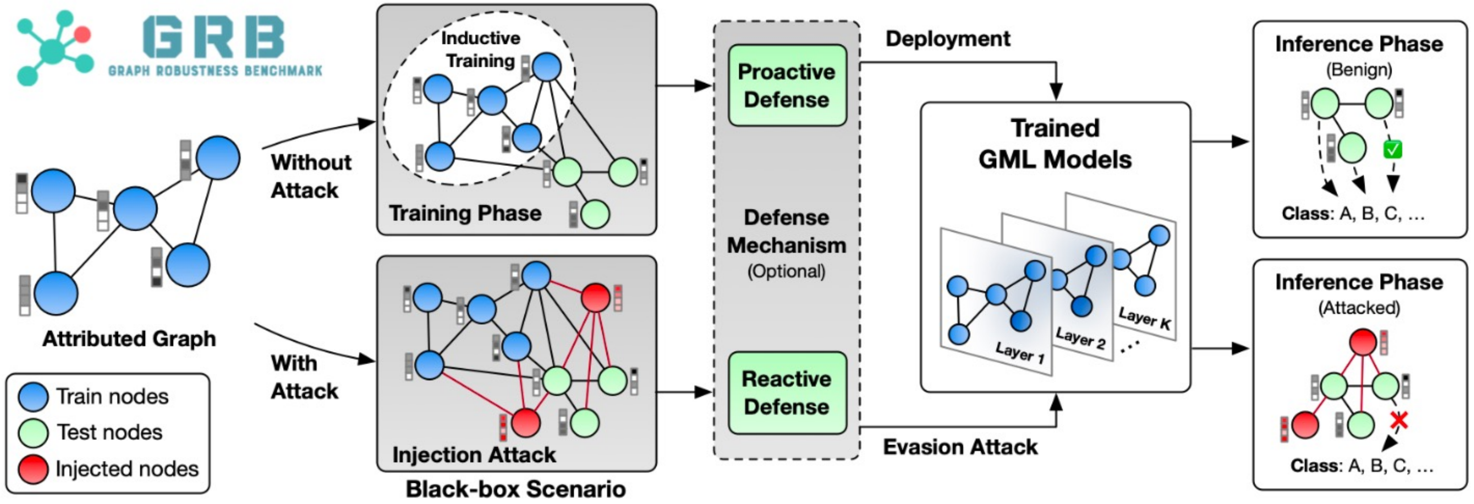
高瓴人工智能学院
Gaoling School of Artificial Intelligence

Intruding with Words: Towards Understanding Graph Injection Attacks at the Text Level

Runlin Lei, Yuwei Hu, Yuchen Ren,
Zhewei Wei

Background

- Graph Injection Attack (GIA)
  - Injecting “malicious” nodes, degrading GNN’s performance
  - More practical than Graph Modification Attacks [1]



An Illustration of GIA from [1]

[1] Qinkai Zheng, et al. Graph Robustness Benchmark: Benchmarking the Adversarial Robustness of Graph Machine Learning

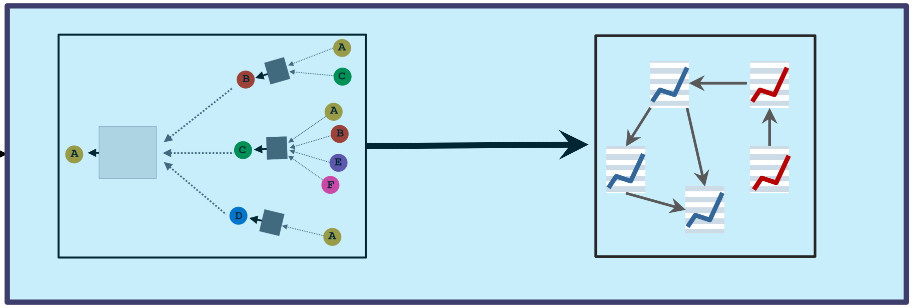
Background

- Text Attributed Graph (TAG)
  - Node attributes are typically text-based
  - Commonly found in networks like citation networks and social networks
- Current GNN Framework:

Authors	Abstract	Keys
……	……	……
……	……	……
……	……	……

Text Attributes

Word Embedding

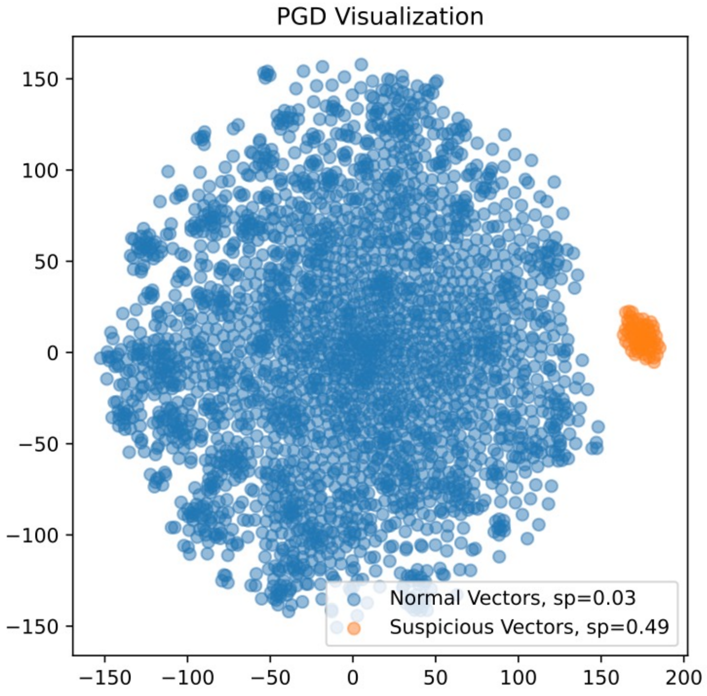


GNN given shallow embedding

Background

- For TAGs, existing GIAs:
  - are limited to embedding-level，not injecting interpretable text
  - are easily detected due to out of distribution
  - have embeddings that may be abnormal in structure

- Example: PGD-based GIA:
  - is still embedding (Orange points)
  - is largely different from blue points
  - holds abnormally high sparsity in embedding



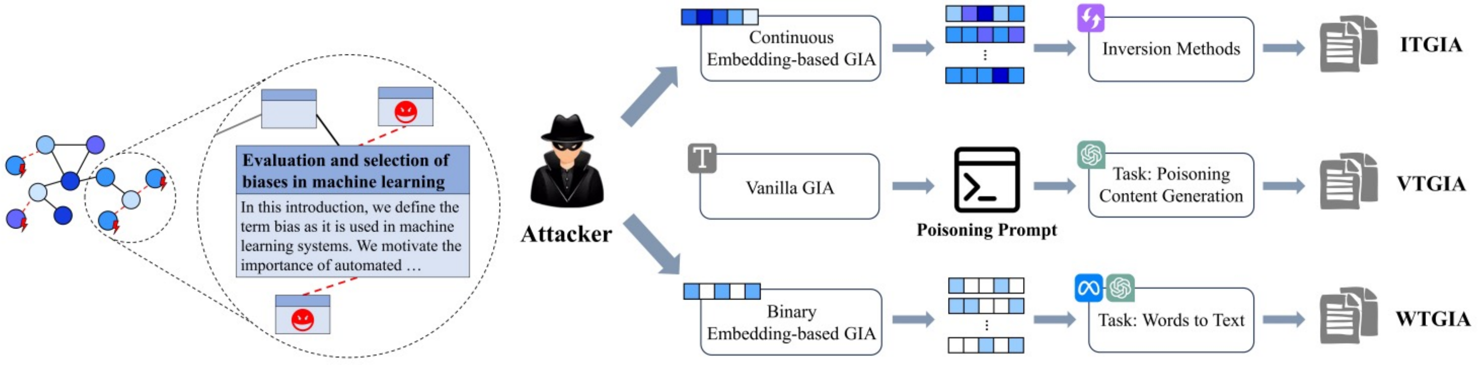
Intruding with Words: Towards Understanding Graph Injection Attacks at the Text Level [NeurIPS’24]

Exploring Text-level GIA

- How to design Text-level GIA?
- How does Text-level GIA perform?
  - Performance
  - Unnoticeability
  - Text Interpretability
- How to defense Text-level GIA?

Exploring Text-level GIA

- Text-level GIA



(a) The task of Text-Level GIAs (b) Framework of ITGIA, VTGIA and WTGIA

Figure 1: Illustration of the Text-Level GIA setup and the three designs explored.

- ITGIA: Based on text inversion, convert injected embedding to text
- VTGIA: Based on direct prompt design, let LLM generate poisoning text
- WTGIA: Based on 0-1 embedding, use word-filling task, let LLM generate poisoning text

ITGIA

- ITGIA: Based on Text Inversion， transferring Embedding into Text.
  - Conducting Embedding-level GIA
  - Using Inversion model [1] to transfer the injected embedding into text
- The text-level performance *degrades a lot* than embedding-level

Table 1: Performance of GCN on graphs under ITGIA. Raw text is embedded by GTR before being fed to GCN for evaluation. "Avg. cos" represents the average cosine similarity between the embeddings of the inverted text and their corresponding original embeddings across five ITGIAs. "Best Emb." represents the best attack performance across the five variants at the embedding level.

Dataset	Clean	HAO	Avg. cos	SeqGIA	MetaGIA	TDGIA	ATDGIA	AGIA	Best Emb.
Cora	87.19 ± 0.62	x	0.14	74.16 ± 1.76	**71.35 ± 1.14**	76.52 ± 1.45	76.73 ± 1.46	72.25 ± 1.32	31.14 ± 0.05
		✓	0.76	**66.70 ± 0.94**	67.83 ± 0.75	71.49 ± 1.71	74.63 ± 2.48	68.81 ± 1.39	
CiteSeer	75.93 ± 0.41	x	0.11	68.17 ± 0.94	69.39 ± 0.89	68.24 ± 1.30	69.72 ± 1.34	**66.18 ± 1.19**	21.45 ± 0.58
		✓	0.56	**64.79 ± 1.30**	65.11 ± 1.01	67.43 ± 0.89	71.89 ± 0.50	**64.79 ± 1.30**	
PubMed	87.91 ± 0.26	x	0.06	65.13 ± 1.67	**58.96 ± 1.25**	59.49 ± 1.08	69.81 ± 1.90	66.16 ± 0.97	38.32 ± 0.00
		✓	0.59	66.40 ± 2.33	**58.56 ± 1.22**	60.26 ± 1.32	76.23 ± 2.08	65.77 ± 0.91	

[1] Morris, John X., et al. "Text embeddings reveal (almost) as much as text." In EMNLP, 2023

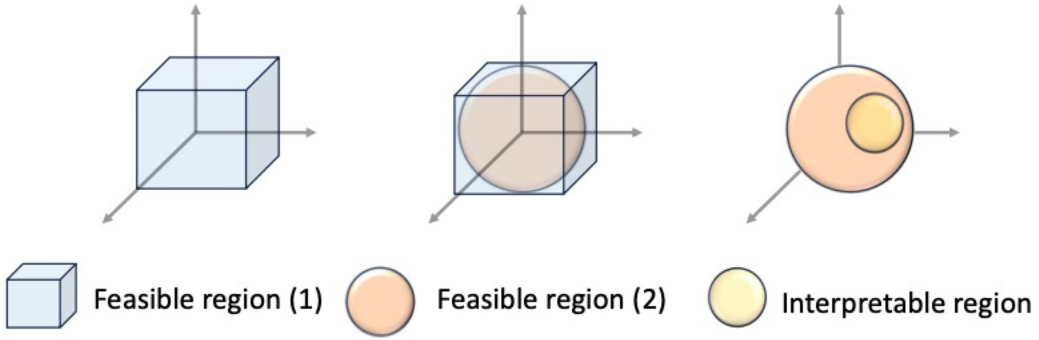
ITGIA

- Poor text interpretability
  - Example： *he liner notes of The MC6’s "Desirty Pigs": the relayed that some of the plaque*
  - High Perplexity:

Table 11: Average perplexity of raw text generated by VTGIA and ITGIA. Clean refers to the average perplexity of original dataset.

Dataset	Clean	VTGIA-Het.	VTGIA-Rand.	VTGIA-Mix.	ITGIA	ITGIA-HAO
Cora	110.47	14.02	18.12	16.63	623.65	546.89
CiteSeer	66.71	14.37	16.53	21.21	705.41	379.80
PubMed	30.85	8.21	16.76	13.52	503.14	348.07

- *Why?* Ill-defined feasible region for embedding-level GIA



VTGIA

- VTGIA： Based on direct prompt design, let LLM generate poisoning text
  - Random Text, Heterophily Text, Mixing Text
  - Readable Text, but bad attack performance

Table 2: Performance of GCN against VTGIA. Raw text is embedded by GTR before being fed to GCN for evaluation. "Best Emb." refers to the best-performing embedding-level GIAs that directly update embeddings across various injection strategies.

Dataset	Clean	Prompt	SeqGIA	MetaGIA	TDGIA	ATDGIA	AGIA	Best Emb.
Cora	87.19 ± 0.62	Heterophily	83.35 ± 0.49	**80.81 ± 0.37**	84.23 ± 0.80	82.05 ± 0.88	83.88 ± 0.83	31.14 ± 0.05
		Random	84.65 ± 1.11	**82.32 ± 0.66**	85.51 ± 0.81	84.73 ± 0.82	86.21 ± 0.77	
		Mixing	83.10 ± 0.80	**80.78 ± 0.66**	83.89 ± 1.32	83.91 ± 1.73	84.19 ± 1.21	
CiteSeer	75.93 ± 0.41	Heterophily	74.91 ± 0.55	**73.32 ± 0.39**	75.50 ± 0.44	73.73 ± 1.04	74.62 ± 0.86	21.45 ± 0.58
		Random	73.84 ± 0.79	73.28 ± 0.69	72.61 ± 1.30	71.43 ± 0.96	**70.81 ± 1.27**	
		Mixing	75.29 ± 0.67	**74.16 ± 0.51**	74.61 ± 0.71	74.74 ± 1.15	74.87 ± 1.03	
PubMed	87.91 ± 0.26	Heterophily	80.80 ± 0.83	77.50 ± 0.52	**75.41 ± 1.22**	75.78 ± 0.77	82.36 ± 0.53	38.32 ± 0.00
		Random	81.99 ± 2.34	**78.34 ± 2.08**	80.39 ± 2.87	82.26 ± 4.46	86.23 ± 0.87	
		Mixing	81.27 ± 1.91	**78.48 ± 1.59**	78.62 ± 2.78	80.37 ± 1.99	85.44 ± 0.80	

Intruding with Words: Towards Understanding Graph Injection Attacks at the Text Level [NeurIPS'24]

WTGIA

- WTGIA：Combining ITGIA and VTGIA
  - Based on 0-1 embedding （BoW），using FGSM with sparsity-budget，generate must-used-words and must-not-used-words lists
  - Based on used-words and not-used-words，let LLM do the word-filling task to generate poisoning text
- Sparsity Budget: *the ratio of must-used-words*
  - For a 500-dim BoW embedding，20% Sparsity Budget means 100 words must appear in the generated text
  - 【No arbitrary long text !!!】Noticeable in practice.
  - 【Given limited text length】，larger budget，lower text interpretability

Intruding with Words: Towards Understanding Graph Injection Attacks at the Text Level [NeurIPS'24]

WTGIA

- Under WTGIA setting，the relationship between Performance & Unnoticeability & Interpretabliity

Theorem 1. *Performance and Unnoticeability can be both satisfied* using larger sparsity budget, at the expense of text *Interpretability*

Theorem 1. *In the setting outlined in Definition 1, assume we apply a cosine similarity constraint with a threshold $c \in (0,1)$ for unnoticeability. Specifically, this constraint requires that the cosine similarity between x_t and x_i satisfies $\frac{x_t \cdot x_i}{\|x_t\| \|x_i\|} > c$. Let a denotes the number of words used by x_i from the set W_u , and b denotes the number of words used by x_i from W_n. If the budget is m words at most to ensure interpretability, then the maximum value of b is $\max(b) = \max\left(\lfloor(m - c\sqrt{mk}\rfloor, 0\right)$.*

WTGIA Experiments

- WTGIA：Balance performance and text interpretability
  - Recover the embedding-level performance，while maintaining text interpretability of generated text

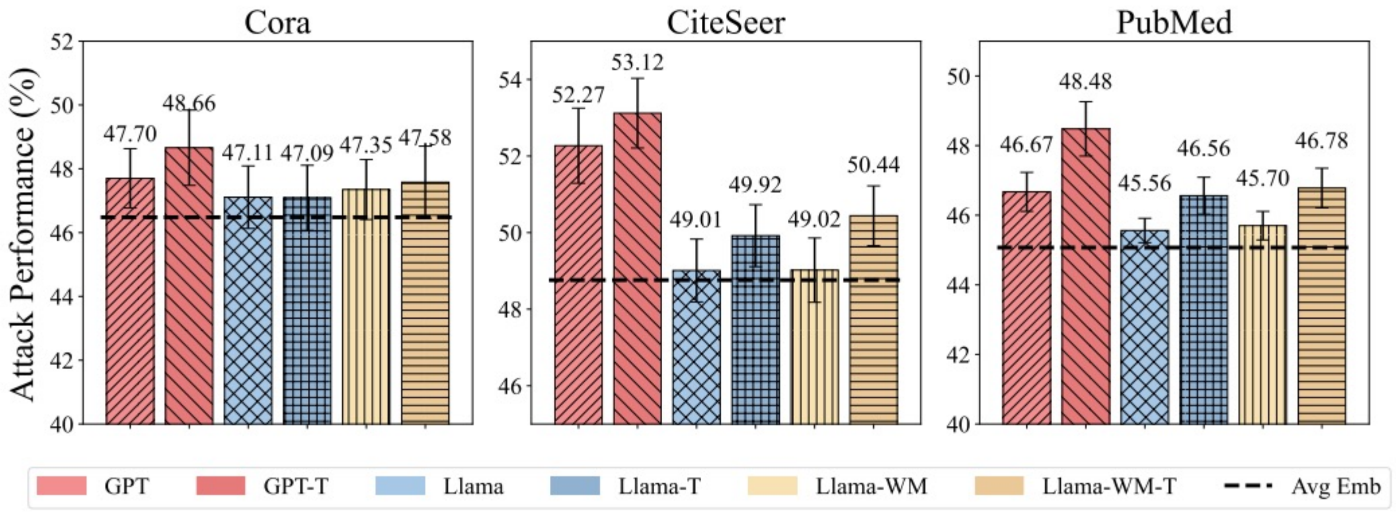
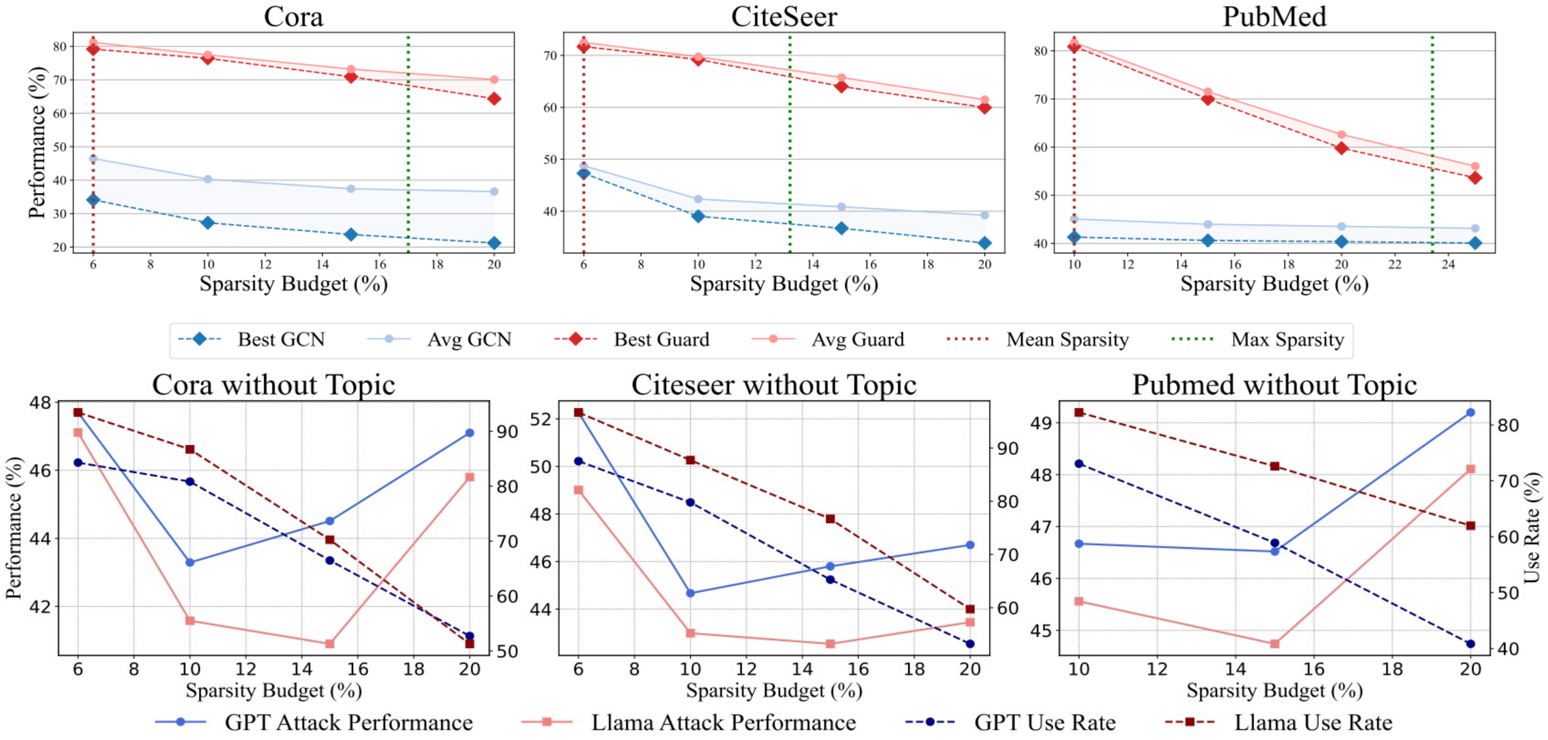


Figure 4: Performance of WTGIA against GCN. Sparsity budget is the average sparsity of the original dataset. Methods with -T include topic requirements in the prompt. Methods with -WM exclude masks for prohibited words in Llama. Avg Emb. represents the average FGSM attack performance at the embedding level. Lower values indicate better attack performance.

Intruding with Words: Towards Understanding Graph Injection Attacks at the Text Level **[NeurIPS'24]**

WTGIA Experiments

- Trade-off between performance and text interpretability
  - Performance & Unnoticeability ✅
  - Performance & Text Interpretability ❌



Intruding with Words: Towards Understanding Graph Injection Attacks at the Text Level [NeurIPS'24]

WTGIA Experiments

- WTGIA's bottleneck
  - Use Rate keeps decreasing，LLMs are unable to complete the task
  - Perplexity also decreases，LLMs use easier words in generating text

Table 12: Average perplexity (↓) and use rate of raw texts generated by WTGIA w.r.t sparsity budget on Cora dataset.

WTGIA Variant	Avg.	0.10	0.15	0.20
GPT Perplexity	53.88	43.11	39.08	35.60
GPT Use Rate (%)	84.29	80.84	66.48	52.72
GPT-Topic Perplexity	30.70	26.92	26.40	25.01
GPT-Topic Use Rate (%)	81.78	73.93	58.85	44.81
Llama Perplexity	90.23	75.95	58.17	54.73
Llama Use Rate (%)	93.43	86.71	54.60	51.29
Llama-Topic Perplexity	83.21	65.97	54.60	55.69
Llama-Topic Use Rate (%)	93.08	86.03	71.56	50.67

Intruding with Words: Towards Understanding Graph Injection Attacks at the Text Level [NeurIPS'24]

Defender Strategies

- Transferability
  - ITGIA: Continuous embedding, WTGIA: 0-1 embedding
  - Huge performance degradation， WTGIA slightly better

Table 3: Performance of ITGIA and WTGIA-Llama transferred to different embeddings on Cora.

Text-GIA	Embedding	Clean	SeqGIA	MetaGIA	TDGIA	ATDGIA	AGIA
ITGIA	BoW	86.48 ± 0.41	84.85 ± 0.76	**84.04 ± 0.78**	85.56 ± 0.61	86.49 ± 0.50	84.90 ± 0.73
	GTR	87.19 ± 0.62	66.70 ± 0.94	67.83 ± 0.75	71.49 ± 1.71	74.63 ± 2.48	**68.81 ± 1.39**
WTGIA	BoW	86.48 ± 0.41	48.32 ± 0.74	51.58 ± 0.78	52.49 ± 1.32	**35.33 ± 1.29**	47.81 ± 0.78
	GTR	87.19 ± 0.62	78.15 ± 1.70	**76.88 ± 0.96**	79.27 ± 1.24	83.77 ± 1.11	77.95 ± 1.51

- Ensemble multiple Word-Embedding can help

Intruding with Words: Towards Understanding Graph Injection Attacks at the Text Level [NeurIPS’24]

Defender Strategies

- LLM-based Predictor are strong defender
  - Directly use LLM as predictor
  - In some datasets (PubMed), perform extremely robust

Table 4: The performance of WTGIA against LLMs-as-predictor. The term "(w/o Nei.)" means the exclusion of neighborhood information in the prompt. Methods "Clean (w/o Nei.)" and "WTGIA (w Nei.)" can be used as LLM-based defenders. The best results for defenders are **bold**.

Dataset	Zero-shot			Few-shot		
	Clean (w Nei.)	Clean (w/o Nei.)	WTGIA (w Nei.)	Clean (w Nei.)	Clean (w/o Nei.)	WTGIA (w Nei.)
Cora	78.64	67.90	**74.81**	79.51	66.54	72.71
CiteSeer	69.18	59.53	67.71	73.90	66.67	**68.44**
PubMed	89.80	**89.80**	89.30	84.50	80.00	80.20

- In practice，LLM-based Methods need to be considered

Conclusion

- We propose:
  - □ The first text-level graph adversarial attack analysis. Discovering past limitations of embedding-level GIA in real-world applications
  - □ Three designs for Text-level GIA. Discovering the trade-off between text interpretability and performance
  - □ Challenges of Text-level GIA in practice with new defender strategies

- Future directions：
  - □ Further improvement for Text-level GIA
  - □ LLM-based defender design

Thanks!

Q&A