Evaluating the Robustness of Graph Neural Networks Against Text-Based Adversarial Attacks

Avviral Jain and Sriram Hebbale and Siddhartha Gotur
Birla Institute of Technology & Science
f20221632@pilani.bits-pilani.ac.in
f20220147@pilani.bits-pilani.ac.in
f2022070@pilani.bits-pilani.ac.in

Abstract

Graph Neural Networks (GNNs) are widely used for learning on text-attributed graphs, where node representations are derived from natural language content such as documents, posts, or descriptions. While prior work on adversarial robustness in GNNs has primarily focused on structural perturbations or numeric feature manipulation, the vulnerability of GNN pipelines to semantic-preserving textual attacks remains underexplored. In this work, we study the robustness of GNN-based node classification models to Unicode-level obfuscation and paraphrasing attacks applied to node text. Using the large-scale OGBN-ArXiv citation network, we construct node features by encoding paper titles and abstracts with multiple sentence embedding models and train GraphSAGE classifiers on the resulting text-attributed graph. We then introduce a suite of Unicode-based attacks - including homoglyph substitution, emoji injection, currency symbol insertion, and mixed-script punctuation - as well as paraphrasing-based perturbations, applied exclusively to test-time node text at varying intensities. Despite preserving semantic content and graph structure, these attacks induce significant drops in classification accuracy and prediction stability. Our analysis further shows that even small embedding-level deviations can lead to disproportionate performance degradation, revealing a brittle dependence on surface-level textual representations. These findings highlight a critical robustness gap in current text-attributed GNN systems and underscore the need for defense mechanisms that account for adversarial and noisy natural language inputs.

1 Introduction

Graph-structured learning has become a central paradigm in machine learning, enabling models to reason over relational data in domains such as citation networks, social media, recommendation systems, and knowledge graphs. In many real-world applications, nodes are not only connected through edges but are also associated with rich textual attributes, including document abstracts, user-generated content, or metadata descriptions. Graph Neural Networks (GNNs) effectively combine graph topology with node features, achieving strong performance on tasks such as node classification and link prediction.

As GNNs are increasingly deployed in practical settings, concerns regarding their robustness have gained prominence. A growing body of work has demonstrated that GNNs are vulnerable to adversarial perturbations, particularly those targeting graph structure (e.g., edge insertion or deletion) or continuous node features. However, these threat models only partially reflect the realities of text-attributed graphs, where node features are derived from natural language via tokenization and embedding pipelines. In such systems, adversarial manipulation can occur at the level of raw text, without altering the graph structure or the semantic meaning of the content.

Textual attacks present a distinct and underexplored challenge. Natural language admits a wide range of transformations - such as Unicode substitutions, symbol insertions, or paraphrasing - that are often imperceptible or innocuous to human readers but can substantially alter the input processed by downstream encoders. For example, visually identical Unicode characters from different scripts, the insertion of emojis or symbols, or syntactic paraphrasing can change token boundaries and embedding representations while preserving semantic intent, as demonstrated in non-Graph environments (Cooper et al., 2025). These perturbations are particularly realistic in open or noisy environments, where text may originate from diverse sources, platforms, or users.

In this work, we systematically investigate the robustness of GNN-based node classification

models to semantic-preserving text-level attacks. We focus on two broad classes of perturbations. The first consists of Unicode-based attacks, which modify the character-level representation of node text through homoglyph substitutions, emoji injection, currency symbol insertion, and mixed-script punctuation. The second class involves paraphrasing-based attacks, where node text is rewritten to maintain meaning while altering lexical and syntactic structure. Crucially, all attacks are applied at test time only, leaving the graph structure, training data, and label semantics unchanged.

Our experiments are conducted on the OGBN-ArXiv dataset (Hu et al., 2020), a large-scale citation network with over 169,000 nodes, where node features are derived from paper titles and abstracts encoded using state-of-the-art sentence embedding models (Reimers and Gurevych, 2019). We train GraphSAGE (Hamilton et al., 2017) classifiers on clean data and evaluate them under controlled adversarial perturbations of varying intensities. By isolating text-level modifications and keeping all other components fixed, we aim to directly measure how sensitive GNN performance is to surface-form variations in node attributes.

Through extensive evaluation, we show that even simple, non-adaptive textual perturbations can cause substantial accuracy degradation and prediction instability, despite inducing only modest changes in embedding space. These results reveal a mismatch between semantic robustness and representational robustness in text-attributed GNN pipelines. Our findings suggest that current approaches implicitly assume clean and well-formed textual inputs, an assumption that does not hold in many real-world or adversarial settings.

In summary, this work makes the following contributions:

- We provide a systematic robustness evaluation of text-attributed GNNs under Unicode-based and paraphrasing attacks.
- We demonstrate that semantic-preserving text perturbations can significantly degrade node classification performance.
- We analyze embedding-level changes to better understand the relationship between textual perturbations and downstream model brittleness.

2 Methodology

2.1 Dataset and Preprocessing

We conduct our experiments on the **OGBN-ArXiv** dataset (Hu et al., 2020), a large-scale citation network comprising 169,343 Computer Science papers from the Microsoft Academic Graph (MAG). Each node represents a paper, with directed edges indicating citations. The task is **node classification**, where papers are categorized into 40 subject areas. The dataset is used with the official split: 90,941 training nodes, 29,799 validation nodes, and 48,603 test nodes.

2.1.1 Text Data Alignment

The OGBN-ArXiv dataset provides graph structure and labels but requires external retrieval for textual content. We retrieve paper titles and abstracts from the official OGB supplementary data (`titleabs.tsv`). We align this text data with the graph nodes using the provided MAG ID to OGB node ID mapping (`nodeidx2paperid.csv.gz`). For each of the 169,343 nodes, we concatenate the paper's title and abstract to form a single text document, which serves as the content feature for the node.

2.2 Text Encoding Pipeline and GNN Training

To generate node features, we employ pretrained sentence embedding models and use these features to train a Graph Neural Network (GNN).

2.2.1 Text Encoders

We evaluate five Sentence-Transformer models to generate initial node features:

1. **all-MiniLM-L6-v2** (384-dim): A lightweight SBERT model.
2. **all-MiniLM-L12-v2** (384-dim): A larger MiniLM variant (used only for initial screening).
3. **all-mpnet-base-v2** (768-dim): A state-of-the-art SBERT model based on MPNet.
4. **paraphrase-multilingual-mpnet-base-v2** (768-dim): Multilingual MPNet variant (used only for initial screening).
5. **intfloat/e5-base-v2** (768-dim): A state-of-the-art LLM-based embedding model (used only for initial screening).

We process the entire corpus using batch encoding (batch size = 64) and apply mean pooling over token embeddings to generate the node feature matrices. Custom mean pooling and L2 normalization were implemented for the E5 model, following the original paper's specifications.

2.2.2 Graph Neural Network (GNN) Training

We train a **2**-layer GraphSAGE model (**?**) for node classification. The architecture uses mean aggregation, ReLU activation, and a 0.5 dropout rate. The input dimension is dependent on the encoder (384 or 768), the hidden dimension is 128, and the output dimension is 40 (number of classes).

Training uses the Adam optimizer with a learning rate of 0.01 and Cross-Entropy loss for up to 200 epochs, employing early stopping with a patience of 20. We save the model checkpoint with the highest validation accuracy.

2.2.3 Model Selection for Adversarial Evaluation

From the five trained models, we select **all-MiniLM-L6-v2** and **all-mpnet-base-v2** for comprehensive adversarial evaluation, as they achieved the highest clean test accuracies. This allows us to compare the adversarial robustness of a lightweight (384-dim) and a higher-capacity (768-dim) encoder under the same attack settings.

2.3 Attack Generation Experiments

We conduct two primary adversarial experiments, each focusing on a distinct mechanism of textual perturbation: character-level manipulation (Unicode) and sentence-level semantic preservation (Paraphrasing).

2.3.1 Experiment 1: Unicode-Based Adversarial Attacks

This experiment evaluates four adversarial strategies that modify the textual content of test nodes through Unicode character manipulation. All attacks preserve semantic meaning while altering the character-level representation.

Homoglyph Substitution Attack This attack replaces Latin characters with visually similar characters from other Unicode scripts (e.g., Cyrillic '' for Latin 'a'). We use a dictionary of 18 common Latin characters and their homoglyphs. Characters are randomly selected and replaced according to a defined attack rate.

Emoji Injection Attack We inject contextually relevant emojis adjacent to specific technical keywords (e.g., "network" $\rightarrow$ "network "). A dictionary maps 18 domain-specific terms to corresponding emojis, testing sensitivity to non-alphabetic Unicode additions.

Currency Symbol Injection Attack This attack targets numerical expressions by randomly injecting currency symbols (e.g., $, €, £) before or after numbers, identified via regular expressions.

Mixed Script Punctuation Attack Standard ASCII punctuation marks are replaced with visually similar punctuation from other scripts, primarily Chinese/Japanese (e.g., '.' $\rightarrow$ ''), while maintaining sentence structure.

Attack Rate Variations For each of the four Unicode strategies, we evaluate three perturbation intensities: **15%, 25%, and 35% attack rates**, which determine the proportion of applicable elements (characters, keywords, numbers, or punctuation) that are modified.

2.3.2 Experiment 2: Paraphrase-Based Adversarial Attacks

This experiment focuses on **semantic-preserving** adversarial attacks that modify node text via sentence-level paraphrasing and back-translation, inducing distributional shifts in the embedding space.

Direct Paraphrasing Attack The test document is rewritten using a neural paraphrasing model. We evaluate both:

1. **Single-step paraphrasing**: The original text is paraphrased once.

2. **Two-step paraphrasing**: The output of the first paraphrase is paraphrased again to compound the semantic and representational drift.

Back-Translation Attack This attack translates the English text into an intermediate pivot language and then translates it back to English. We evaluate three pivot languages:

- Chinese (`zh`)
- Hindi (`hi`)
- German (`de`)

We consider both single-step and selected two-step back-translation to modulate perturbation strength.

2.4 Attack Application Protocol

Due to the computational cost of generating high-quality semantic paraphrases and back-translations using large language models, we evaluate attacks on a fixed subset of **1,000 nodes** sampled from the OGBN-ArXiv test split.

After text perturbation, node embeddings are recomputed using a frozen pretrained sentence encoder. These embeddings replace the original node features, and inference is rerun using the unchanged GNN parameters and graph structure.

2.5 Evaluation and Analysis Metrics

We evaluate model performance under adversarial conditions using four key metrics:

1. **Clean Test Accuracy**: Baseline accuracy on the unmodified test set.
2. **Attacked Test Accuracy**: Accuracy after applying adversarial perturbations.
3. **Accuracy Drop**: Absolute difference between clean and attacked accuracy.
4. **Prediction Flip Rate**: Percentage of test nodes whose prediction changes from correct to incorrect due to the attack.

2.5.1 Embedding Perturbation Analysis

To quantify the impact of the textual modifications on the node representations, we compute:

- **Mean Squared Error (MSE)**: Average squared difference between clean and attacked embeddings (Unicode attacks).
- **L2 Distance**: Average Euclidean distance between embedding pairs (Unicode attacks).
- **Cosine Similarity**: Average, median, and maximum cosine similarity between corresponding clean and attacked embedding vectors (both attacks).

These metrics provide insight into whether observed performance drops correlate with significant representational shifts in the embedding space.

2.6 Experimental Design Summary

Our systematic evaluation covers all combinations of the selected encoders (all-MiniLM-L6-v2, all-mpnet-base-v2) and the defined attack strategies and intensities. This results in 24 conditions for the Unicode attacks (2 encoders $\times$ 4 attacks $\times$ 3 rates) and multiple conditions for the Paraphrasing attacks (Direct and Back-Translation variations). All experiments utilize the identical graph structure, data splits, and GNN architecture, isolating the impact of the textual adversarial perturbations on the model's robustness.

3 Results

We evaluate the robustness of text-augmented Graph Neural Networks (GNNs) on the OGBN-ArXiv dataset under two distinct classes of adversarial textual perturbations: character-level (Unicode) and semantics-preserving (Paraphrasing). All GraphSAGE models are trained on clean data.

3.1 Baseline Performance

Two GraphSAGE models were trained using different text encoders. The `all-MiniLM-L6-v2` (384-dimensional) model achieved a clean test accuracy of $\mathbf{70.48}\%$, while the `all-mpnet-base-v2` (768-dimensional) model achieved $\mathbf{72.33}\%$ on the OGBN-ArXiv test set. The 1.85 percentage point improvement in MPNet accuracy validates the benefit of larger embedding dimensions and more sophisticated pretraining, and these baselines serve as the reference point for all adversarial evaluations across 48,603 test nodes.

3.2 Results for Unicode-Based Adversarial Attacks

Table 1 presents the comprehensive results for four character-level attack strategies. The attacks demonstrate vastly different levels of effectiveness, with homoglyph substitution causing severe degradation while other perturbations have minimal impact.

3.2.1 Homoglyph Attack: Critical Vulnerability

The homoglyph substitution attack proved to be the only effective adversarial strategy. At the maximum 35% attack rate, MiniLM suffers a catastrophic $\mathbf{27.12}\%$ accuracy drop with cosine similarity plummeting to 0.405. MPNet, however,

Encoder	Attack	Rate	Clean	Attacked	Drop	Cos Sim
MiniLM	Homoglyph	15%	70.48%	61.39%	**9.09%**	0.716
		25%	70.48%	52.43%	**18.05%**	0.540
		35%	70.48%	43.36%	**27.12%**	0.405
	Emoji	15%	70.48%	70.21%	0.27%	0.981
		25%	70.48%	70.10%	0.38%	0.997
		35%	70.48%	70.00%	0.47%	0.995
	Mixed Script	15%	70.48%	70.38%	0.09%	0.999
		25%	70.48%	70.24%	0.23%	0.998
		35%	70.48%	70.09%	0.38%	0.997
	Currency	15%	70.48%	70.50%	-0.02%	0.999
		25%	70.48%	70.50%	-0.03%	0.999
		35%	70.48%	70.50%	-0.02%	0.999
MPNet	Homoglyph	15%	72.33%	69.72%	**2.60%**	0.899
		25%	72.33%	66.89%	**5.43%**	0.819
		35%	72.33%	62.92%	**9.41%**	0.727
	Emoji	15%	72.33%	72.24%	0.08%	0.999
		25%	72.33%	72.18%	0.15%	0.998
		35%	72.33%	72.09%	0.23%	0.996
	Mixed Script	15%	72.33%	72.33%	-0.00%	1.000
		25%	72.33%	72.25%	0.07%	0.999
		35%	72.33%	72.22%	0.11%	0.998
	Currency	15%	72.33%	72.29%	0.03%	1.000
		25%	72.33%	72.28%	0.04%	1.000
		35%	72.33%	72.29%	0.04%	1.000

Table 1: Attack effectiveness comparison across encoders and intensity levels for Unicode-based attacks. Homoglyph attacks cause severe degradation while other perturbations have negligible impact. MPNet demonstrates substantially greater robustness than MiniLM.

Encoder	Rate	Acc Drop	Flip Rate	Cos Sim	MSE
MiniLM	15%	9.09%	13.11%	0.716	0.00148
	25%	18.05%	22.17%	0.540	0.00239
	35%	27.12%	30.78%	0.405	0.00310
MPNet	15%	2.60%	6.04%	0.899	0.00026
	25%	5.43%	9.55%	0.819	0.00047
	35%	9.41%	14.22%	0.727	0.00071
Robustness Gain		**2.9**×	**2.2**×	–	**4.4**×

Table 2: Homoglyph attack impact across encoders. MPNet demonstrates 2.9× lower accuracy drop and 4.4× smaller embedding perturbations at maximum intensity (35% rate), revealing that larger models provide substantial robustness benefits.

demonstrates superior robustness, suffering only a $\mathbf{9.41}\%$ drop. This represents a $\mathbf{2.9}\times$ robustness advantage for MPNet.

The embedding perturbation analysis confirms the mechanism: MPNet's MSE is $4.4\times$ smaller than MiniLM's at the maximum rate (Table 2), demonstrating that the larger 768-dimensional space is substantially more resistant to character-level perturbations, a finding further supported by Table 3.

3.2.2 Minimal Impact Attacks: Additive Ineffectiveness

In contrast, the Currency, Mixed Script Punctuation, and Emoji Injection attacks resulted in a maximum accuracy drop of only 0.47% across all conditions. These **additive perturbations** are highly ineffective, showing high cosine similarity (> 0.997) and $100 - 1000\times$ smaller MSE values compared to the homoglyph attack (Table 3). This suggests encoders either strip or neutralize these additive symbols during tokenization.

Encoder	Attack (35%)	MSE	Relative
MiniLM	Homoglyph	0.00310	1.0×
	Emoji	0.000028	0.009×
	Mixed Script	0.000017	0.005×
	Currency	0.0000038	0.001×
MPNet	Homoglyph	0.00071	1.0×
	Emoji	0.0000097	0.014×
	Mixed Script	0.0000039	0.005×
	Currency	0.00000086	0.001×

Table 3: Mean Squared Error between clean and attacked embeddings at maximum attack intensity. Homoglyphs cause 100–$1000\times$ larger perturbations than other attacks.

Embeddings	Steps	Accuracy	ΔAcc	Flip Rate
MPNet	1	59.8%	–12.6%	29.4%
MPNet	2	48.7%	–23.7%	43.3%
MiniLM-L6	1	56.2%	–14.9%	31.2%
MiniLM-L6	2	46.7%	–24.4%	42.6%

Table 4: Performance under direct paraphrase attacks. Accuracy drops and flip rates increase consistently with attack strength across both embedding models. (Clean Baselines: MPNet: 72.4%, MiniLM: 71.7%)

Language Pair	Steps	Accuracy	ΔAcc	Flip Rate
ZH→EN	1	63.7%	–7.4%	22.6%
ZH→EN	2	61.2%	–9.9%	27.2%
DE→EN	1	70.3%	–1.4%	10.5%
HI→EN	1	42.7%	–29.0%	50.1%

Table 5: Back-translation attack results using MiniLM-L6 embeddings. Effectiveness varies substantially across languages, with Hindi resulting in a severe drop.

3.3 Results for Paraphrase-Based Adversarial Attacks

We evaluate robustness under semantics-preserving paraphrase attacks applied exclusively to test-node textual content, with performance reported using test accuracy, absolute accuracy drop (ΔAcc), and label flip rate.

3.3.1 Direct Paraphrase Attacks

A single paraphrasing step results in a substantial degradation in performance, as detailed in Table 4. Accuracy dropped by 12–15 percentage points across both encoders, flipping approximately 30% of node predictions. Two-step paraphrasing further compounds this effect, pushing accuracy below 50% and causing over $\mathbf{42}\%$ of predictions to change.

3.3.2 Back-Translation Attacks

Back-translation attacks show high variability based on the pivot language, as shown in Table 5 (using MiniLM-L6 embeddings). Back-translation through German (DE→EN) has minimal impact ($\Delta\text{Acc} = 1.4\%$), whereas Hindi–English back-translation (HI→EN) results in the largest accuracy drop ($\mathbf{29.0}$ percentage points) and flips over half of all predictions.

4 Discussion

4.1 Comparative Analysis of Attack Mechanisms and Vulnerabilities

Our experiments with character-level (Unicode) and sentence-level (Paraphrasing) attacks reveal two distinct, yet critical, classes of vulnerability in text-augmented GNNs.

4.1.1 Character-Level Vulnerability: The Tokenization Bottleneck

The extreme effectiveness of the **Homoglyph Substitution Attack** stems from fundamental characteristics of subword tokenization in SBERT models. Replacing Latin characters with visually identical Cyrillic/Greek homoglyphs forces tokenizers into:

1. **Out-of-vocabulary (OOV)** fragmentation, or
2. Mapping to **distinct token IDs**.

This tokenization divergence collapses the semantic alignment of the resulting embeddings, evidenced by cosine similarities as low as 0.405 in MiniLM. In contrast, **additive perturbations** (Emoji, Currency, Mixed Script Punctuation) are rendered ineffective because modern tokenizers either neutralize these tokens or the transformer attention mechanism learns to downweight their semantic contribution, demonstrating an implicit, though limited, robustness against non-substitutive noise.

4.1.2 Sentence-Level Vulnerability: Brittle Semantic Embeddings

In contrast to the tokenization failure of homoglyphs, the **Paraphrasing Attacks** succeed by exploiting the brittleness of the sentence embedding space itself. The results demonstrate that GNNs are highly sensitive to paraphrasing, even when the perturbations preserve semantic similarity (high cosine similarity). This indicates a reliance on surface-level textual representations rather than invariant semantics. The observed failures suggest that the graph context alone does not compensate for the instability inherent in fixed, pre-trained textual representations.

4.2 Cumulative Effects and Semantic Drift

The consistent degradation observed from **one-step to two-step paraphrasing** suggests a significant cumulative effect. Successive meaning-

Attack Class	Attack Type	Encoder	Attack Rate / Strength	ΔAcc	Flip Rate
Unicode (Injection)	Homoglyph Substitution	MiniLM-L6	35% chars replaced	–27.12%	30.78%
	Homoglyph Substitution	MPNet	35% chars replaced	–9.41%	14.22%
	Emoji Injection	MiniLM-L6	35% keywords injected	–0.47%	≈0%
	Emoji Injection	MPNet	35% keywords injected	–0.23%	≈0%
	Mixed-Script Punctuation	MiniLM-L6	35% punctuation replaced	–0.38%	≈0%
	Currency Symbol Injection	MPNet	35% numbers modified	–0.04%	≈0%
Semantic (Paraphrase)	Direct Paraphrase	MPNet	2.06% test nodes (1-step)	–12.6%	29.4%
	Direct Paraphrase	MPNet	2.06% test nodes (2-step)	–23.7%	43.3%
	Direct Paraphrase	MiniLM-L6	2.06% test nodes (1-step)	–14.9%	31.2%
	Direct Paraphrase	MiniLM-L6	2.06% test nodes (2-step)	–24.4%	42.6%
Semantic (Back-Translation)	ZH→EN	MiniLM-L6	2.06% test nodes (1-step)	–7.4%	22.6%
	DE→EN	MiniLM-L6	2.06% test nodes (1-step)	–1.4%	10.5%
	HI→EN	MiniLM-L6	2.06% test nodes (1-step)	–29.0%	50.1%
	ZH→EN	MPNet	2.06% test nodes (1-step)	–5.9%	22.1%
	DE→EN	MPNet	2.06% test nodes (1-step)	–0.7%	9.0%
	HI→EN	MPNet	2.06% test nodes (1-step)	–30.7%	53.0%

Table 6: Unified comparison of text-based adversarial attacks on OGBN-ArXiv. Unicode injection attacks are parameterized by the fraction of applicable elements modified, while paraphrasing and back-translation attacks affect 2.06% of test nodes (1,000 out of 48,603).

preserving rewrites progressively shift node representations into decision regions associated with different class labels. Notably, under these stronger attacks, a high cosine similarity between the clean and attacked embedding **does not** guarantee classifier stability, highlighting a crucial disconnect between embedding-space proximity and classifier output.

4.3 The Role of Encoder Capacity in Defense

The comparative results across the two attack mechanisms provide clear guidance on defense strategies:

- **Robustness against Homoglyphs**: The **MPNet** encoder (768-dim) exhibits significantly superior robustness ($\mathbf{2.9}\times$ lower accuracy drop and $\mathbf{4.4}\times$ smaller Mean Squared Error) against homoglyph attacks compared to MiniLM. This is attributed to its larger embedding space and superior pretraining, which "dilutes" character-level perturbations more effectively.

- **Vulnerability to Paraphrasing**: This advantage is marginal under Direct Paraphrase attacks. Both MPNet and MiniLM converge to similar failure modes under multi-step attacks, suggesting that higher embedding quality alone is **insufficient** to ensure robustness when embeddings are used as fixed node features against sophisticated semantic perturbations.

4.4 Linguistic Distance and Real-World Implications

The large variation in attack effectiveness across back-translation pivot languages emphasizes that **linguistic distance** and translation variability play a critical role in robustness. Back-translation through typologically distant languages (e.g., Hindi) introduces perturbations that are substantially more disruptive to downstream classification, despite preserving overall meaning.

4.4.1 Implications for Graph-Based NLP Security

These findings raise serious concerns for the security and integrity of real-world graph-based NLP systems. Scenarios such as academic paper revision, document versioning, or cross-lingual information flow can introduce similar perturbations. Since the graph structure remains static, the observed failures suggest that attackers do not need to exploit graph topology; targeting the text-feature generation pipeline is sufficient to severely undermine classification performance across a broad range of NLP + Graph ML applications.

4.5 Defense Strategies

Based on these findings, we propose that effective defense requires a multi-pronged approach:

1. **Use Larger Models:**: Deploy MPNet or larger encoders (2.9 times more robust than MiniLM). It would cost 2-3x inference time

and 2x the memory. The benefit is substantial robustness with no code changes.

2. **Adversarial Training**: Augment training data with homoglyph perturbations (10-20% of the samples). Force the model to learn homoglyph-invariant representations however the challenge would be that it would require us to retrain a GNN which is computationally expensive.

5 Conclusion

In this work, we conducted a systematic evaluation of the adversarial robustness of text-augmented Graph Neural Networks (GNNs) on the OGBN-ArXiv dataset across two distinct attack modalities: character-level Unicode manipulation and semantics-preserving paraphrasing.

We demonstrated that GNN-based node classification systems are vulnerable to both low-level and high-level textual perturbations.

5.1 Summary of Key Findings

5.1.1 Vulnerability to Character-Level Attacks (Homoglyphs)

We identify **homoglyph substitution** as a critical, encoder-dependent vulnerability stemming from the tokenization pipeline. While small encoders (`MiniLM`) suffer catastrophic $\mathbf{27.12}\%$ accuracy drops, larger models (`MPNet`) demonstrate $\mathbf{2.9}\times$ greater robustness with only $\mathbf{9.41}\%$ degradation at the maximum attack rate. This finding has immediate implications: $\mathbf{1}$. Larger models offer a crucial defense layer, and $\mathbf{2}$. Simple Unicode normalization can effectively neutralize this threat. Conversely, additive perturbations (Emoji, Currency, Punctuation) were found to be universally ineffective, highlighting the implicit robustness of attention mechanisms against non-substitutive noise.

5.1.2 Vulnerability to Semantics-Preserving Attacks (Paraphrasing)

By applying direct paraphrasing and back-translation exclusively at inference time, we demonstrated that **meaning-preserving textual perturbations** can cause substantial degradation in node classification performance, even when the underlying graph structure remains unchanged. Both single-step and multi-step paraphrase attacks lead to large accuracy drops and high label flip rates across different sentence embedding models. Notably, increasing attack strength compounds performance degradation despite moderate to high cosine similarity, indicating that embedding-space proximity alone is insufficient to guarantee prediction stability. Back-translation experiments further reveal that linguistic factors play a significant role in attack effectiveness, with typologically distant languages inducing the largest performance losses.

5.2 Implications and Future Work

These findings highlight a fundamental vulnerability in current text–graph fusion pipelines, where classifiers exhibit sensitivity to surface-level textual variation rather than robust semantic understanding. Addressing this security and reliability issue will require future work on:

- **Robustness-Aware Training Objectives**: Developing methods to jointly train GNNs and text encoders to learn representations that are invariant to both tokenization errors and semantic-preserving paraphrases.
- **Adaptive Fusion Mechanisms**: Creating models that can better integrate and cross-validate textual and structural signals, allowing the graph context to compensate for instability in the text embeddings.
- **Proactive Defenses**: Implementing normalization and character filtering for known low-level attacks, while focusing future research on complex, meaning-preserving adversarial data augmentation.

We hope this study motivates further research into the reliability and security of graph-based NLP systems under realistic linguistic perturbations.

References

P. Cooper, E. Blanco, and M. Surdeanu. 2025. The lies characters tell: Utilizing large language models to normalize adversarial unicode perturbations. In *Findings of the Association for Computational Linguistics (ACL)*, pages 18932–18944.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark:

Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.