Seeing Through the Mask: AI-Generated Text Detection with Similarity-Guided Graph Reasoning

Nidhi Gupta, Qinghua Li
Department of Electrical Engineering & Computer Science
University of Arkansas
Fayetteville, AR, USA
{nidhig, qinghual}@uark.edu

Abstract

The rise of generative AI has led to challenges in distinguishing AI-generated text from human-written content, raising concerns about misinformation and content authenticity. Detecting AI-generated text remains challenging, especially under various stylistic domains and paraphrased inputs. We introduce SGG-ATD, a novel detection framework that models structural and contextual relationships between LLM-predicted and original-input text. By masking parts of the input and reconstructing them using a language model, we capture implicit coherence patterns. These are encoded in a graph where cosine and contextual links between keywords guide classification via a Graph Convolutional Network (GCN). SGG-ATD achieves strong performance across diverse datasets and shows resilience to adversarial rephrasing and out-of-distribution inputs, outperforming competitive baselines.

1 Introduction

In an era where machines write as fluently as humans, we are entering a new chapter in how information is produced, consumed, and trusted. Large Language Models (LLMs) such as GPT-4 (Achiam et al., 2023), Claude (Anthropic, 2023), and LLaMA (Touvron et al., 2023) have made it nearly effortless to generate essays, news articles, reviews, and even research papers with human-like fluency. What was once an imaginative leap, a machine composing coherent and contextually accurate paragraphs, is now commonplace. The boundary between synthetic and authentic language is becoming indistinguishable to the naked eye.

As this generative capability becomes more accessible and widespread through models like GPT (Brown et al., 2020), BERT (Devlin et al., 2019), and T5 (Raffel et al., 2020), its applications have expanded rapidly to include content creation, conversational agents, and real-time translation (Vaswani et al., 2017; Open, 2023). However, this growing realism brings profound challenges: from misinformation and fake news propagation to academic dishonesty and erosion of digital trust (Bender et al., 2021; Weidinger et al., 2021; Zellers et al., 2019; Gupta et al., 2025). With AI-generated content becoming nearly indistinguishable from human writing, questions around authorship, authenticity, and accountability are now more urgent than ever.

As these models seamlessly blend into communication workflows, a new and urgent challenge emerges. Educators, journalists, policymakers, and even AI developers are increasingly grappling with a pressing question: How do we determine who—or what—authored a piece of text? From student assignments generated at the push of a button to fabricated news articles and automated spam campaigns, the misuse of LLMs has already begun to erode trust in written communication. Existing detection methods are increasingly ineffective. Traditional approaches (Gehrmann et al., 2019; Afroz et al., 2012) rely on shallow linguistic features or supervised classifiers trained on known model outputs. While effective on benchmarks, they often fail to generalize across domains or resist adversarial rewriting and stylistic obfuscation (Mitchell et al., 2023; Verma et al., 2023). As a result, AI-generated text can be easily manipulated to appear human, highlighting the need for deeper, structure-aware detection frameworks.

At the heart of this dilemma lies a deeper question—not just whether a piece of text is AI-generated, but whether its structure and predictability reveal traces of its origin. Human language, while flexible and expressive, carries with it natural irregularities and subtleties rooted in reasoning, creativity, and intent. AI-generated text, by contrast, is often more formulaic, exhibiting higher token-level predictability and stylistic consistency. Capturing this difference requires methods that can perceive and represent the interplay between mean-

Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 1350–1360

ing, context, and linguistic structure.

However, most existing detection methods fail to operationalize this structural perspective. Despite recent advances, two major limitations persist. (1) *Lack of structural reasoning:* While prior work recognizes that AI-generated text tends to exhibit higher predictability, many existing methods rely only on surface-level cues such as per-token probabilities (Solaiman et al., 2019; Ippolito et al., 2019) or shallow statistical features (Gehrmann et al., 2019; Jawahar et al., 2020), failing to model the deeper contextual and compositional structures that give rise to these patterns. (2) *Limited generalization across varied domains:* Existing detectors such as Mitchell et al. (2023) and Verma et al. (2023) often underperform when applied to unseen domains or writing styles.

Building on this intuition, we propose a new approach to AI-generated text detection that leverages masked language modeling to uncover patterns of semantic coherence and contextual regularity. We first extract content-rich keywords from the input text and mask a fixed subset. A pretrained language model predicts the masked keywords, and both the extracted and predicted keywords are used to construct a contextual graph. In this graph, nodes represent keywords, and edges encode lexical semantics and contextual similarity. This structure allows our framework to reason over meaning-based patterns and generative signals, enabling more accurate and robust classification.

Our method, SGG-ATD, a graph-based framework for AI-generated text detection, addresses the limitations outlined earlier by combining masked language modeling with graph-based reasoning:

- We construct a graph connecting original keywords and LLM-predicted keywords, enabling the model to capture how words relate in both meaning and context. This moves beyond isolated word-level analysis and captures structural flows, key areas where AI-generated text often diverges from human writing.
- We leverage masked keyword prediction to help the model learn contextual predictability across varied text types, including news, essays, technical descriptions, and creative writing. This facilitates robust detection of generative patterns that generalize across domains and styles.
- We conduct comprehensive empirical evaluation across four datasets, demonstrating strong generalization to out-of-distribution domains, robustness to adversarial paraphrasing, and effectiveness through ablation studies. Our framework consistently outperforms strong baselines in F1 score and robustness, validating the practical impact of our design.

By combining semantic meaning and LLM prediction behavior within a graph structure, SGG-ATD offers a unified framework for modeling contextual and structural signals, enabling more reliable detection of AI-generated content, even under prompt variation or domain shifts.

2 Related Work

Large language models (LLMs) have significantly improved machine-generated text, reducing the gap with human writing. Early models like GPT-2 and GPT-3 showcased few and zero-shot learning (Radford et al., 2019; Brown et al., 2020), with later scaled versions (Chowdhery et al., 2023; Zhang et al., 2022) enhancing tasks like instruction following and QA. Despite progress, studies (Jawahar et al., 2020; Dou et al., 2021) noted linguistic gaps, including lower factuality and coherence in early outputs.

To detect AI-generated text, prior work leveraged surface features, probabilities, or neural cues. Gehrmann et al. (2019) used token likelihoods, Mitchell et al. (2023) analyzed log-probability curvature, and Verma et al. (2023) scored tokens via weaker models. Others like Chen et al. (2023) combined DeBERTa with classifiers for strong results. However, many methods depend on access to scoring APIs or logits, limiting use with closed-source LLMs.

Recent work moved beyond token-level cues by incorporating structure and semantics. For example, Mao et al. (2024) used rewriting-based detection, measuring changes after text rewriting, while Valdez and Gómez-Adorno (2025) applied GNNs to capture word co-occurrence patterns. These methods aimed to overcome the limitations of shallow-feature detectors.

Domain generalization emerged a key challenge in detecting text from unseen models like GPT-4. Bhattacharjee et al. (2024) used domain-adversarial and contrastive learning to generalize without retraining, while Bhattacharjee et al. (2023) framed detection as domain adaptation, enabling transfer from older to newer LLMs without labels. Both

aim to future-proof detectors against rapid advances in generation technologies. Siddiq et al. (2022), though effective, still relied on feature alignment rather than deeper semantic grounding

In parallel, Watermarking-based detection also saw renewed interest. Kirchenbauer et al. (2023) introduced a soft watermark by biasing token distributions, while Zhao et al. (2023) proposed a statistically robust version resilient to paraphrasing. Sadasivan et al. (2023) depended on stylistic patterns or frequency-based features that can be evaded through prompt rephrasing or synonym substitution. A survey by Kamaruddin et al. (2018) reviewed earlier methods and noted challenges like multilinguality and adversarial robustness. These techniques offer post-hoc verifiability but rely on model-side cooperation.

Despite progress, many detectors remain vulnerable to simple evasion tactics like rephrasing, synonym swaps, or style shifts, which degrade performance even in strong models (Mitchell et al., 2023; Chen et al., 2023). Prompt-only attacks that preserve meaning also fooled multiple detectors (Zou et al., 2023; Zhang et al., 2024), raising concerns about long-term robustness.

Prompt engineering has also played a dual role—both in instructing models for tasks and in enabling or defeating detection. Chain-of-thought prompting, prefix tuning, and zero-shot reasoning enhanced reasoning fluency in LLMs (Wei et al., 2022; Li and Liang, 2021; Kojima et al., 2022). However, these same mechanisms can be exploited to disguise AI-generated text or control its stylistic fingerprint as in Zhou et al. (2022).

Finally, questions of fairness and bias in detection remain largely underexplored. Liang et al. (2023) found detectors often mislabel non-native English as AI-generated, raising equity concerns. Guo et al. (2023) showed AI text differs in tone and formality, influencing its acceptability across tasks.

Together, this body of work underscores that despite significant progress, AI-generated text detection remains challenging, particularly under adversarial, cross-domain, and stylistically diverse scenarios. In response, our framework shifts focus to the underlying structure and contextual predictability of the text by modeling relationships between original and LLM-predicted keywords. This alternative perspective aims to offer robustness in detection without relying on model-specific signatures. A preliminary version of this work is included in the author's thesis Gupta (2025).

3 Method

In this section, we present our AI Text Detection Framework, SGG-ATD (Figure 1), which combines masked language modeling with graph-based reasoning to detect AI-generated text. It captures semantic associations and contextual predictability through a context-enriched graph. Figure 2 illustrates the full pipeline with an example input. The framework consists of four key components.

1. **Masking and Keyword Extraction:** In our framework, we begin by randomly masking 30% of the input text keywords, to simulate partial context and expose underlying structural cues. Parallelly, we extract syntactically meaningful keywords (nouns and verbs) from the original input text using Part-of-Speech (POS) tagging (Church, 1989).

2. **Masked Keyword Prediction:** The masked input text is then passed through a pretrained ALBERT-base-v2 model (Lan et al., 2019), which predicts the missing keywords based on surrounding context. These predictions provide insight into keyword-level predictability, revealing structural regularities often present in AI-generated content.

3. **Graph Construction with Dual Similarity Encoding:** A graph is constructed where nodes represent original and LLM-predicted keywords. Edges are weighted using cosine similarity and contextual similarity, which are combined into a unified adjacency matrix for graph-based reasoning.

4. **Graph-Based Classification via GCN:** The constructed graph is processed using a two-layer GCN (Kipf and Welling, 2016), which propagates and aggregates information across keyword nodes. A global graph representation is then derived and passed to a classifier to determine whether the input text is AI-generated or human-written.

We highlight the novelty and contributions of this framework as follows. (1) *Predictive Masking for Structural Signal:* Unlike prior works, our approach probes contextual predictability by masking semantic keywords and reconstructing them using a pretrained language model, capturing generative patterns often indicative of AI-written text. (2) *Dual Similarity Graph Encoding:* The integration

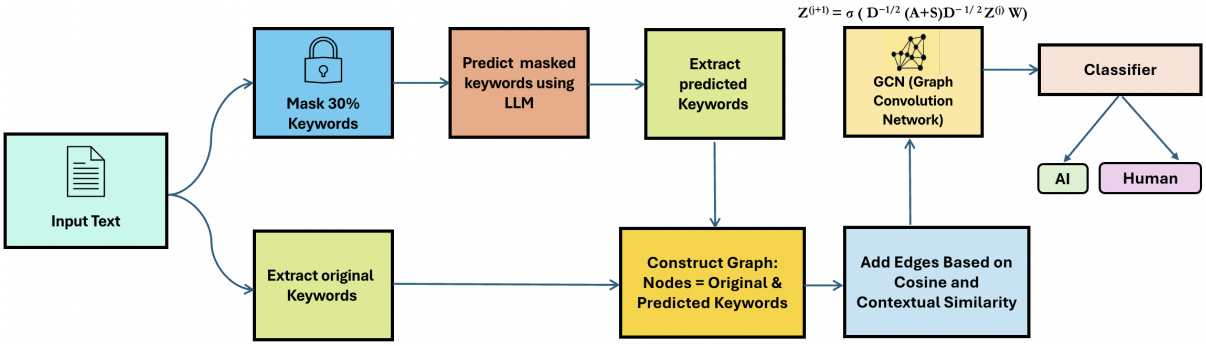

Figure 1: SGG-ATD detects AI-generated text by constructing a graph per input, where nodes are original and predicted keywords. Edges encode lexical semantics (cosine) and contextual (prediction-based) similarity. A GCN processes the graph for final classification.

of lexical semantics and contextual similarity into a single graph structure enables more expressive relational modeling. (3) *Graph-Based Reasoning over Prediction-Informed Graphs:* We leverage a Graph Convolutional Network (GCN) over the constructed similarity graph to model higher-order dependencies, supporting robust detection beyond surface-level textual patterns.

3.1 Masking and Keyword Extraction

Given an input text, we randomly select a subset of keywords $\mathcal{M}$ to be masked, where $\mathcal{M} \subset T$ and T is the full sequence. Specifically, we mask 30% of the keywords by replacing them with `<mask>` tokens:

$$|\mathcal{M}| = \lfloor \alpha |T| \rfloor, \quad \text{where } \alpha = 0.3 \tag{1}$$

This produces a masked version of the input text T_m, which is later used to probe contextual predictability through a language model. In parallel, we extract a set of syntactically meaningful keywords $\mathcal{K} = \{k_1, k_2, \ldots, k_n\}$ from the original text using part-of-speech (POS) tagging, focusing on nouns and verbs for their semantic importance.

3.2 Masked Keyword Prediction

To expose latent structural differences between AI-generated and human-written texts, we employ a prediction step inspired by masked language modeling (MLM). The masked input text is passed to a pretrained ALBERT-base-v2 model (Lan et al., 2019), which predicts the missing keywords based on surrounding context.

Our hypothesis is that language models demonstrate higher confidence and accuracy in reconstructing masked keywords in AI-generated text, due to its syntactic regularity and high dependency on keyword-level patterns. In contrast, human-written content—being more varied and context-rich—leads to greater prediction uncertainty.

As illustrated in Figure 3, this behavioral difference becomes evident when comparing prediction results across both text types. The figure shows that AI-generated texts result in more accurate predictions, while human-written texts often produce more incorrect keywords (incorrect predictions are highlighted in blue), supporting our hypothesis.

The predicted keywords are treated as contextual reconstructions and are later used to construct a graph alongside the original keywords. Formally, given a masked input text T_m, the predicted keywords $\hat{\mathcal{M}}$ are obtained as:

$$\hat{\mathcal{M}} = \text{ALBERT}(T_m) \tag{2}$$

To ensure high-quality predictions, we filter out punctuation and malformed outputs (e.g., incomplete tokens, symbols).

3.3 Graph Construction with Dual Similarity Encoding

A graph representation of the text is constructed, where nodes represent both original and LLM-predicted keywords. We construct a similarity graph where each node is connected to every other node, and edges are weighted using two key similarity measures: **(1) Lexical Semantic Adjacency Matrix (A):** Captures semantic similarity between words on subword-level lexical features using FastText embeddings (Bojanowski et al., 2017) via cosine similarity. **(2) Contextual Similarity Matrix (S):** Encodes contextual alignment between original and predicted keywords based on dot-product similarity. These two similarity measures are computed independently and reflect distinct aspects of

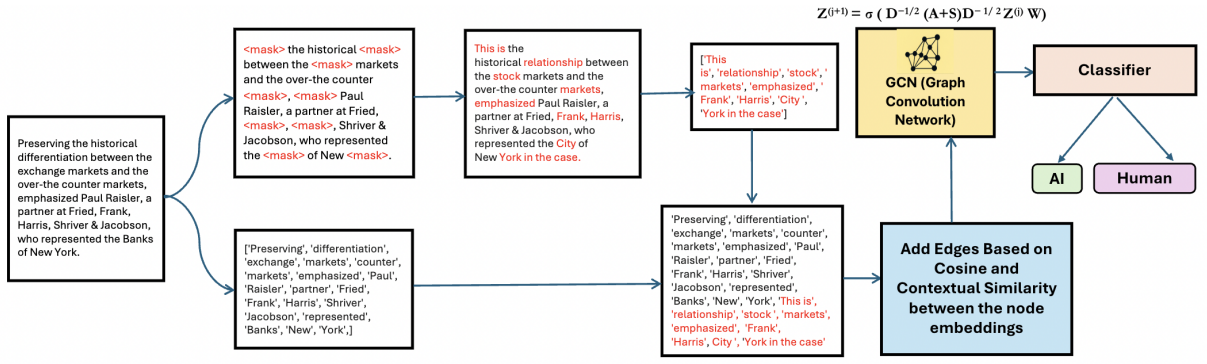

Figure 2: An example illustrating the process of SGG-ATD.

textual structure: lexical semantics and contextual predictability.

The lexical semantic adjacency matrix A and contextual similarity matrix S are computed as:

$$A_{ij} = \frac{\mathbf{w}_i \cdot \mathbf{w}_j}{\|\mathbf{w}_i\| \|\mathbf{w}_j\|} \tag{3}$$

$$S_{ij} = \mathbf{w}_i \cdot \mathbf{w}_j \tag{4}$$

where $\mathbf{w}_i$ and $\mathbf{w}_j$ are FastText embeddings of words i and j.

To form the final graph structure, we integrate both signals by summing the two matrices:

$$A' = A + S \tag{5}$$

The combined adjacency matrix A' is then used as input to the GCN for graph-based reasoning.

3.4 Graph-Based Classification via GCN

The constructed similarity graph is processed using a Graph Convolutional Network (GCN), which operates on the enhanced adjacency matrix A' that encodes both lexical semantics and contextual similarity. The GCN propagates information across nodes to refine their embeddings and model higher-order relationships relevant for classification.

Node embeddings are updated layer-wise as follows:

$$Z^{(i+1)} = \sigma\left(D^{-1/2}(A' + I)D^{-1/2}Z^{(i)}W\right) \tag{6}$$

where $Z^{(i)}$ is the node embedding at layer i, A' is the modified adjacency matrix, D is the degree matrix, W is a trainable weight matrix, and σ is a non-linear activation function (e.g., ReLU). The initial input $Z^{(0)} = X$ corresponds to the feature matrix composed of FastText embeddings of the original and predicted keywords.

After the final layer, the node representations are aggregated using mean pooling to form a global graph representation, and passed to a classifier:

$$\hat{y} = \text{softmax}(\text{Classifier}(\text{MeanPool}(Z))) \tag{7}$$

Here, $\hat{y}$ is the predicted label indicating whether the input text is AI-generated or human-written.

3.5 Training and Evaluation

The GCN-based classifier is trained using a binary cross-entropy loss function:

$$\mathcal{L} = -\sum_i y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \tag{8}$$

where $y_i \in \{0, 1\}$ is the true label (1 for AI-generated, 0 for human-written), and $\hat{y}_i$ is the predicted probability output from the model.

4 Evaluations

4.1 Datasets

To evaluate our approach, we use four diverse text datasets spanning different writing domains and linguistic styles:

- **News Dataset (News)** – Journalistic content with a formal tone and fact-based reporting, sourced from Verma et al. (2023).
- **Creative Writing (CW)** – Fictional and narrative-driven samples featuring diverse vocabulary and stylistic variation, also from Verma et al. (2023).
- **Student Essay (SE)** – Academic-style writing with structured reasoning and moderate complexity, derived from Verma et al. (2023).
- **Vulnerability Dataset (Vuln)** – Technical descriptions of software vulnerabilities. We

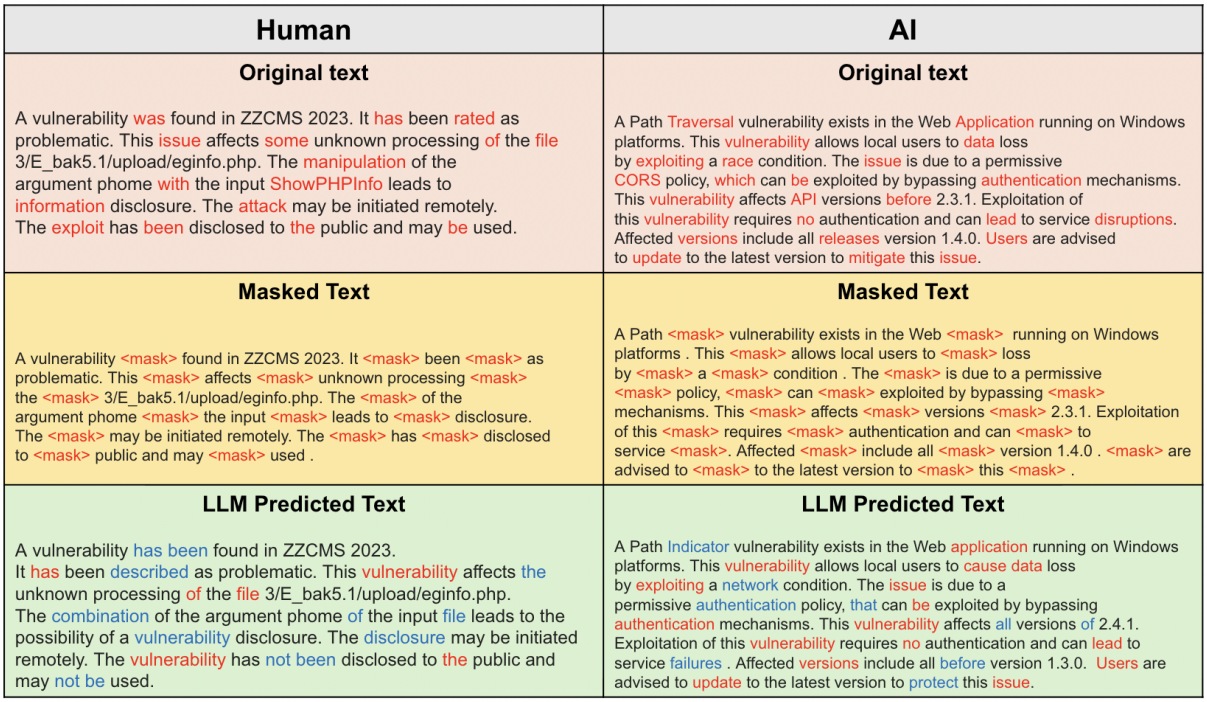

Figure 3: This illustration shows the rationale behind our masking strategy using examples from the vulnerability dataset. 30% of keywords are masked in both AI and human examples. The language model predicts the masked keywords, and differences in accuracy reveal predictability patterns across text types. Incorrect predictions (blue keywords) occur more often in human-written text, indicating lower contextual predictability.

constructed this dataset ourselves: human-written samples were taken from the NVD (National Institute of Standards and Technology), and AI-generated samples were created using ChatGPT (Open, 2023) to match the NVD style.

Table 1 summarizes dataset statistics across domains chosen to evaluate robustness across diverse writing styles, ranging from news and academic essays to creative narratives and technical vulnerability reports. This diversity exposes the model to varied linguistic patterns, domain-specific vocabulary, and stylistic complexity, making it effective for detecting AI-generated content in both general and specialized contexts. Each dataset is randomly split into 80% training and 20% testing.

Table 1: Dataset Statistics Across Domains

	News	CW	SE	Vuln
# Dataset Size	479	728	13629	946
# Median Length	45	38	82	30
# Min Length	3	2	2	4
# Max Length	208	354	291	429

4.2 Implementation Details

We implemented our model in PyTorch (Hu et al., 2021; Paszke et al., 2017), leveraging the HuggingFace Transformers library and pretrained ALBERT-Base v2 (Lan et al., 2019) for masked language modeling. Keyword extraction was performed using NLTK (Bird et al., 2009), and FastText embeddings were used to represent nodes in the graph. Each input sample was converted into a graph structure informed by lexical semantics and contextual similarity. A two-layer Graph Convolutional Network (GCN) processed the graph, and its output was passed through a fully connected layer for binary classification. The model was trained using binary cross-entropy loss with the Adam optimizer (Kingma and Ba, 2017), a learning rate of 0.01, and 100 epochs on an NVIDIA GPU.

4.3 Baselines

We compare our method against several state-of-the-art AI-generated text detection approaches that employ diverse strategies:

- **GPTZero (Tian, 2023):** A commercial tool that uses mathematical features like perplexity to assess whether text is human- or AI-written.

- **DetectGPT (Mitchell et al., 2023):** A zero-shot method that detects AI-generated text by analyzing the curvature of the log-probability landscape from a language model.
- **Ghostbuster (Verma et al., 2023):** Constructs feature representations from aggregated predictions of small language models to capture statistical patterns in AI content.
- **RAIDAR (Mao et al., 2024):** A rewriting-based method that uses the degree of change from language model rewrites, measured via edit distance, as a detection signal.

4.4 Main Results

Table 2 presents the core results of our model and baseline comparisons across all four datasets using F1 score as the evaluation metric, consistent with prior works (Verma et al., 2023; Mao et al., 2024) where it was the sole reported metric. Among existing models, RAIDAR and Ghostbuster demonstrate strong performance in structured and technical domains like Student Essay and Vulnerability dataset, reaching up to 0.69 and 0.75 respectively. However, our model, which integrates contextual graph modeling with masked keyword reconstruction, achieves the highest F1 scores across all domains, attaining 0.96 on the Vulnerability dataset and 0.88 on Student Essay using a masking ratio of 0.3.

Table 2: Performance comparison across all datasets

Methods	News	CW	SE	Vuln
GPTZero	0.43	0.61	0.48	0.66
DetectGPT	0.41	0.63	0.52	0.72
GhostBuster	0.59	0.57	0.64	0.75
RAIDAR	0.63	0.65	0.69	0.84
SGG-ATD (Ours)	**0.79**	**0.72**	**0.88**	**0.96**

Furthermore, our method significantly outperforms all baselines in challenging domains such as Creative Writing and News, where other detectors like GPTZero and DetectGPT struggle due to reliance on shallow statistical cues. The consistent performance of our model across diverse writing styles demonstrates the robustness of our graph-augmented detection framework.

4.5 Analysis

4.5.1 Effect of LLM Backbone

As shown in Table 3 (F1 Scores), we assess our framework using various backbone language models at a fixed 0.3 masking ratio. ALBERT-Base v2 offers the best overall trade-off, excelling in News and Creative Writing while maintaining strong performance in other domains. DeBERTa-Base (He et al., 2020) and RoBERTa (Liu et al., 2019) also perform well, especially on the Vulnerability dataset, and BERT-Base-Uncased (Devlin et al., 2019) shows strong results for Student Essay and Vulnerability dataset. These findings highlight the modularity and model-agnostic nature of our graph-based framework.

Table 3: Effect of LLM Choice on Performance

LLM	News	CW	SE	Vuln
BERT-Base-Uncased	0.75	0.66	0.88	0.97
ALBERT-Base v2	0.79	0.72	0.88	0.96
DeBERTa-Base	0.75	0.72	0.85	0.97
Roberta	0.73	0.70	0.86	0.96

4.5.2 Out-of-Distribution Generalization

Table 4 shows out-of-distribution results (F1 scores) using a *leave-one-domain-out* setup, where the model is trained on three domains and tested on the unseen fourth. These unseen domains differ notably in tone, structure, and syntax, making OOD a strong test of generalization. Our model consistently outperforms baselines, with notable gains in News (0.67 vs. 0.49, 0.58) and Vulnerability (0.75 vs. 0.62, 0.66), and leads in more stylistically diverse domains like Creative Writing and Student Essay. GPTZero and DetectGPT, being unsupervised, show identical in-domain and OOD performance, underscoring the advantage of our supervised, graph-based approach. Overall, SGG-ATD demonstrates stronger robustness to distributional shifts across domains.

Table 4: Out-of-Distribution (OOD) Evaluation

Methods	News	CW	SE	Vuln
GPTZero	0.43	0.61	0.48	0.66
DetectGPT	0.41	0.63	0.52	0.72
GhostBuster	0.49	0.52	0.50	0.62
RAIDAR	0.58	0.59	0.53	0.66
SGG-ATD (Ours)	**0.67**	**0.65**	**0.61**	**0.75**

4.5.3 Effect of Masking Ratio

Figure 4 shows how masking ratio impacts model performance (F1 scores) across domains. Vulnerability and Student Essay datasets remain stable, with Vulnerability consistently above 0.97 and peaking at 0.99. Creative Writing is more sensitive,

with F1 dropping from 0.77 to 0.71 at higher ratios. News improves up to 0.79 at 0.3 before leveling off. Based on this, we adopt 0.3 as the default ratio, it yields the best News score and competitive results elsewhere. This setting balances under-masking and over-masking, supporting both generalization and reconstruction learning.

Figure 4: Effect of masking ratio on F1 scores across four datasets. A 0.3 ratio offers the best balance, peaking in News and performing competitively in other domains, while higher ratios hurt performance in stylistically varied data like Creative Writing.

4.5.4 Ablation Study

To analyze the role of individual components, we perform ablation studies summarized in Table 5 (F1 Scores). Using only cosine or contextual similarity edges results in a slight performance drop, with cosine edges performing slightly better overall, suggesting lexical semantics offers more stable structural signals. Replacing the GCN with an MLP (Multi-Layer Perceptron) leads to a clear decline across all domains, especially in News and Creative Writing, confirming the value of graph-based reasoning in capturing higher-order dependencies.

Table 5: Ablation Study Results

Components	News	CW	SE	Vuln
Cosine Edges Only	0.76	0.69	0.84	0.93
Contextual Edges Only	0.75	0.67	0.83	0.91
GCN replaced with MLP	0.69	0.63	0.80	0.90

4.5.5 Single Prompt (Rephrasing Evasion)

Table 6 shows the vulnerability of our model to adversarial prompt engineering under a single training prompt setting. Following the work in Mao et al. (2024), the detector is trained using standard GPT prompt (Non-Adaptive) and tested against both the standard GPT prompt and rephrased versions generated with prompts designed to evade detection. For instance, Prompt 1 ("Help me rephrase it in human style") and Prompt 2 ("Help me rephrase it so that another GPT rewriting will cause a lot of modifications") significantly reduce detection performance, especially in Creative Writing and News domains. F1 scores drop from 0.72 to 0.39 and from 0.79 to 0.44 in Creative and News respectively. These results reveal that even structurally minor paraphrases can obscure generative signals learned during training, highlighting the fragility of detection systems trained on narrow prompt distributions.

Table 6: Rephrasing Evasion (Single Prompt Training)

Prompt	News	CW	SE	Vuln
Non-Adaptive	0.79	0.72	0.88	0.96
Prompt 1	0.53	0.42	0.57	0.77
Prompt 2	0.44	0.39	0.73	0.81

4.5.6 Multi-Prompt Training for Robustness

To mitigate the effects of adversarial rephrasing, we train our model using two prompts and evaluate it on an unseen third prompt. As shown in Table 7, this multi-prompt strategy yields notable F1 gains across domains when compared to single-prompt in Table 6. News improves from 0.44 to 0.71 and Creative Writing from 0.42 to 0.65. These results indicate the model better generalizes to diverse rephrasing styles. While performance on non-adaptive prompts slightly declines (e.g., News: 0.79 to 0.75, CW: 0.72 to 0.69, Vuln: 0.96 to 0.90, SE: 0.88 to 0.72), the trade-off is acceptable given the enhanced resilience to prompt-based evasion.

Table 7: Rephrasing Evasion (Multi-Prompt Training)

Prompt	News	CW	SE	Vuln
Non-Adaptive	0.75	0.69	0.72	0.90
Prompt 1	0.67	0.65	0.61	0.86
Prompt 2	0.71	0.68	0.79	0.91

5 Conclusions

We proposed SGG-ATD, a graph-based framework for AI text detection that combines masked keyword prediction with contextual reasoning. By modeling relationships between original and predicted keywords using a GCN, the approach captures structural differences between human and AI-generated text. Experiments across four datasets showed consistent gains over baselines and strong generalization to unseen domains.

Limitations

One limitation of SGG-ATD is its current focus on textual input alone. Extending the framework to handle multimodal content, such as combining text with associated images or metadata, offers a promising future direction, especially for detecting AI-generated content in news and social media. While SGG-ATD performs well against rephrasing, its robustness against more advanced adversarial tactics, remains an open challenge and an avenue for further enhancement.

Acknowledgement

This material is based upon work supported in part by the Department of Energy under Award Number DE-CR0000003.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Sadia Afroz, Michael Brennan, and Rachel Greenstadt. 2012. Detecting hoaxes, frauds, and deception in writing style online. In *2012 IEEE symposium on security and privacy*, pages 461–475. IEEE.

Anthropic. 2023. Claude (v1-v3). https://www.anthropic.com/index/claude. Large language model.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.

Amrita Bhattacharjee, Tharindu Kumarage, Raha Moraffah, and Huan Liu. 2023. Conda: Contrastive domain adaptation for ai-generated text detection. *arXiv preprint arXiv:2309.03992*.

Amrita Bhattacharjee, Raha Moraffah, Joshua Garland, and Huan Liu. 2024. Eagle: A domain generalization framework for ai-generated text detection. *arXiv preprint arXiv:2403.15690*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. 2023. Gpt-sentinel: Distinguishing human and chatgpt generated content. *arXiv preprint arXiv:2305.07969*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Kenneth Ward Church. 1989. A stochastic parts program and noun phrase parser for unrestricted text. In *International Conference on Acoustics, Speech, and Signal Processing,*, pages 695–698. IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Yao Dou, Maxwell Forbes, Rik Koncel-Kedziorski, Noah A Smith, and Yejin Choi. 2021. Is gpt-3 text indistinguishable from human text? scarecrow: A framework for scrutinizing machine text. *arXiv preprint arXiv:2107.01294*.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

Nidhi Gupta. 2025. Towards content authenticity: Multimodal fake news detection and ai-generated text identification. Master's thesis.

Nidhi Gupta, Qinghua Li, and Lu Zhang. 2025. Camfend: Credibility-aware multimodal fake news detection with rotational attention. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Jun Hu, Shengsheng Qian, Quan Fang, Youze Wang, Quan Zhao, Huaiwen Zhang, and Changsheng Xu. 2021. Efficient graph deep learning in tensorflow with tf_geometric. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3775–3778.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2019. Automatic detection of generated text is easiest when humans are fooled. *arXiv preprint arXiv:1911.00650*.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks VS Lakshmanan. 2020. Automatic detection of machine generated text: A critical survey. *arXiv preprint arXiv:2011.01314*.

Nurul Shamimi Kamaruddin, Amirrudin Kamsin, Lip Yee Por, and Hameedur Rahman. 2018. A review of text watermarking: theory, methods, and applications. *IEEE Access*, 6:8011–8028.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *International Conference on Machine Learning*, pages 17061–17084. PMLR.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Weixin Liang, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Zou. 2023. Gpt detectors are biased against non-native english writers. *Patterns*, 4(7).

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Chengzhi Mao, Carl Vondrick, Hao Wang, and Junfeng Yang. 2024. Raidar: generative ai detection via rewriting. *arXiv preprint arXiv:2401.12970*.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*, pages 24950–24962. PMLR.

National Institute of Standards and Technology. National vulnerability database (nvd). https://nvd.nist.gov/.

AI Open. 2023. Chatgpt (mar 14 version)[large language model].

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can ai-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*.

Mohammed Latif Siddiq, Shafayat H Majumder, Maisha R Mim, Sourov Jajodia, and Joanna CS Santos. 2022. An empirical study of code smells in transformer-based code generation techniques. In *2022 IEEE 22nd International Working Conference on Source Code Analysis and Manipulation (SCAM)*, pages 71–82. IEEE.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, and 1 others. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Edward Tian. 2023. Gptzero. AI-generated text detection tool.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Andric Valdez and Helena Gómez-Adorno. 2025. Text graph neural networks for detecting ai-generated content. In *Proceedings of the 1stWorkshop on GenAI Content Detection (GenAIDetect)*, pages 134–139.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 2023. Ghostbuster: Detecting text ghostwritten by large language models. *arXiv preprint arXiv:2305.15047*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, and 1 others. 2021. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, and 1 others. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Yuehan Zhang, Yongqiang Ma, Jiawei Liu, Xiaozhong Liu, Xiaofeng Wang, and Wei Lu. 2024. Detection vs. anti-detection: Is text generated by ai detectable? In *International Conference on Information*, pages 209–222. Springer.

Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. 2023. Provable robust watermarking for ai-generated text. *arXiv preprint arXiv:2306.17439*.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.