LLM-based Contrastive Self-Supervised AMR Learning with Masked Graph Autoencoders for Fake News Detection

Anonymous ACL submission

Abstract

The proliferation of misinformation in the digital age has led to significant societal challenges. Existing approaches often struggle with capturing long-range dependencies, complex semantic relations, and the social dynamics influencing news dissemination. Furthermore, these methods require extensive labelled datasets, making their deployment resource-intensive. In this study, we propose a novel self-supervised misinformation detection framework that integrates both complex semantic relations using Abstract Meaning Representation (AMR) and news propagation dynamics. We introduce an LLM-based graph contrastive loss (LGCL) that utilizes negative anchor points generated by a Large Language Model (LLM) to enhance feature separability in a zero-shot manner. To incorporate social context, we employ a multi view graph masked autoencoder, which learns news propagation features from social context graph. By combining these semantic and propagation-based features, our approach effectively differentiates between fake and real news in a self-supervised manner. Extensive experiments demonstrate that our self-supervised framework achieves superior performance compared to other state-of-the-art methodologies, even with limited labelled datasets while improving generalizability.[1]

1 Introduction

The spread of misinformation has become a significant problem in the digital age. It can lead to social unrest, foster hatred, erode trust, and ultimately impede the overall progress and stability of the society (Dewatana and Adillah, 2021). Hence, effectively detecting misinformation has become an essential challenge to solve.

Table 1: Comparison of different methods based on their utilization of various graph-based learning components. The table evaluates whether each method incorporates an AMR (Abstract Meaning Representation) graph, a Social Context Graph (SCG), a Graph Masked Autoencoder with augmentations (GMA^2), a Graph Masked Autoencoder with multi-view remasking (GMA^2+R), and Unsupervised Feature Generation (U).

Method	AMR	SCG	GMA^2	GMA^2+R	U
EA^{2}N	✓	✗	✗	✗	✗
GACL	✗	✓	✗	✗	✗
(UMD)^2	✗	✓	✗	✗	✓
GTUT	✗	✓	✗	✗	✓
GAMC	✗	✓	✓	✗	✓
Ours	✓	✓	✓	✓	✓

The concept of the "veracity problem on the web" was first introduced by (Yin et al., 2008) by designing a solution called *TruthFinder*. This method verified news content by cross-referencing it with information from reputable websites. Later, (Feng et al., 2012) employed manually crafted textual features for detecting misinformation. However, manually crafted features are time-consuming to create and fail to capture the complex semantic relations present in the text. Subsequently, many researchers turned to more advanced techniques, utilizing RNN's, and Transformer-based (Long et al., 2017; Liu and Wu, 2018) models to address this issue. For example, RNNs are employed to capture local and temporal dependencies within text data (Ma et al., 2016a; Li et al., 2022) and BERT has been increasingly utilized to improve the comprehension of contextual relationships in news articles (Devlin et al., 2019). Key limitations of these approaches are their struggle to maintain longer text dependencies and they do not capture complex semantic relations, such as events, locations, and trigger words. (Gupta et al., 2025) solves this problem but requires supervision. Additionally, these models often neglect the social context and dynamics that influence news propagation (Yuan et al., 2019). Acknowledging this, researchers have introduced graph-based approaches that integrate social con-

[1] Code repository: https://anonymous.4open.science/r/Fake1-3245/README.md

text into the detection process (Min et al., 2022; Sun et al., 2022; Li et al., 2024). Despite their effectiveness, these methods rely heavily on large, labelled datasets for training. Collecting and annotating such extensive datasets is time-consuming and resource-intensive, limiting their practical implementation. To address this (Yin et al., 2024) propose a model to generate unsupervised features from the social context graph but do not consider the semantic relationship within the text. Therefore, we require a model that is capable of incorporating semantic text features, a social context propagation graph and also perform well with minimal labelled data as highlighted in Table 1.

This paper proposes a novel self-supervised misinformation detection methodology that considers complex semantic relations among entities in the news and the propagation of the news as a social context graph. In order to identify the semantic relations, this method incorporates a self-supervised Abstract Meaning Representation (AMR) encoder using the proposed graph contrastive loss. This loss creates feature separation by sampling negative anchor points using LLM. The use of negative anchor points from LLM helps in increasing the separation between fake and real classes in the latent space. In order to integrate the social context and capture the propagation of the news, our methodology also integrates a multi-view Graph Masked Autoencoder that employs the context and content of the news propagation process as the self-supervised signal to enhance the final feature space. These features, even with limited labelled data, achieve performance comparable or better than supervised counterparts using a simple linear SVM layer. The key contributions of our research are as follows:

- A novel self-supervised learning based on AMR and social context graph is introduced in order to validate the veracity of news articles, eliminating dependence on labelled data.
- In order to segregate the feature space among real and fake classes, graph contrastive loss is proposed. An LLM-based negative sampler is designed to handle negatives in the loss.
- To capture the social context and propagation feature of the news, we propose an augmentation-based multi-view masked graph autoencoder module.
- Comprehensive evaluation with SOTA methods, demonstrating its superior performance.

2 Related Work

In this section, we provide a concise overview of the approaches utilized for detecting misinformation. The relevant studies are categorized into two main components: misinformation detection and self-supervised graph learning methodologies.

2.1 Misinformation Detection Methods

Early research on misinformation detection focused on manually crafted linguistic features (Feng et al., 2012; Ma et al., 2016b; Long et al., 2017), requiring significant effort for evaluation. EANN (Wang et al., 2018) is proposed to effectively extract event-invariant features from multimedia content, thereby enhancing the detection of misinformation on newly arrived events. In this line of work, recently, FakeFlow (Ghanem et al., 2021) classified news using lexical features and affective information. In a separate line of work, external knowledge was integrated to improve model performance. Different source of external knowledge was used. For example, Popat et al. (Popat et al., 2017) retrieved external articles to model interactions; KAN (Dun et al., 2021) and CompareNet (Hu et al., 2021) leveraged Wikidata for domain expansion, while KGML (Yao et al., 2021) bridged meta-training and meta-testing using knowledge bases. Further, researchers have developed graph-based methods that incorporate social context into the detection process, for example, authors of GTUT (Gangireddy et al., 2020) construct a graph for initial fake news spreader identification, (UMD)^2 (Silva et al., 2024) considers user credibility and propagation speed, GACL (Sun et al., 2022) constructs a tree of tweets for contrastive learning. All these methods do not leverage the complete propagation graph, and GACL requires supervision. Other graph-based methods like (Min et al., 2022; Li et al., 2024) rely heavily on manual annotation and external data.

Recently, Abstract Meaning Representation (AMR)-based methods emerged to mitigate long-text dependency. Abstract Meaning Representation (AMR), as introduced by (Banarescu et al., 2013), captures relationships between nodes using PropBank framesets, sentence vocabularies, and a wide range of over a hundred semantic relations, including negation, conjunction, command, and wikification. Its goal is to represent sentences with identical semantic meaning using the same AMR graph. Recently, Zhang et al. (Zhang et al., 2023) utilized AMR to detect out-of-context multimodal misinformation by identifying discrepancies between textual and visual data. In (Gupta et al., 2023), authors encoded textual information using AMR and explored how its semantic relations influence the veracity assessment of news. However, this study lacked sufficient evidence or justification for entity relationships within the AMR graph. Further, in the integration of evidence in AMR, EA^{2}N (Gupta et al., 2025) is proposed that effectively captures evidence among entities present in AMR. All of these approaches rely on supervised data for AMR training and have not explored the potential of unsupervised methods.

2.2 Self-Supervised Graph Learning

Self-supervised graph learning harnesses the structured richness of graph data to derive meaningful representations without relying on explicit labels (Wu et al., 2023). Kipf et al. introduced a Graph Auto-Encoder (GAE), a method that encodes a graph into a lower-dimensional space and reconstructs it back to its original form, surpassing traditional approaches based on manually crafted features (Kipf and Welling, 2016). Recognizing that many GAEs struggle to reconstruct node features, subsequent research has focused on reconstructing masked features to improve the efficiency of self-supervised GAEs for classification tasks (Hou et al., 2022). Further, (Hou et al., 2023) improved the performance by introducing multi-view random remasking. Recently, an unsupervised method for detecting misinformation GAMC (Yin et al., 2024) has been proposed by leveraging both the context and content of news propagation as self-supervised signals. However, GAMC does not effectively handle complex semantic relations for longer text dependencies.

3 Methodology

The overall methodology is presented in Figure 1. In this section we present these in more detail.

3.1 Self-supervised AMR Graph Learning

Given an input text T, we first create the AMR graph $\mathcal{G}^{amr}(\mathcal{V}^{amr}, \mathcal{E}^{amr})$ capturing the relationships between different entities. AMR generation process involves parsing the sentences to extract linguistic information, including semantic roles, relations, and core events. In order to incorporate reasoning through AMR, we have integrated the external evidence by using the Evidence Linking Algorithm (ELA) used in (Gupta et al., 2025). The graph after applying ELA is referred to as WikiAMR, represented as $\mathcal{G}^{WikiAMR}$. WikiAMR comprises interconnected undirected paths between entity nodes in $\mathcal{G}^{amr}$ generated from the text. The WikiAMR representation helps to distinguish the difference between real and fake articles.

AMR Graph Learning with Path Optimization: This module plays an important role in extracting meaningful features from the given WikiAMR graph. Features extracted here capture essential semantic relationships, enabling a deeper understanding of the underlying textual data. At the core of this module is a Graph Transformer (Cai and Lam, 2020), which employs various attention mechanisms to effectively process the graph representation. This allows the model to reason about and learn from the text more comprehensively.

The WikiAMR graph is first passed through a node initialization and relation encoder to transform it into a representation in $\mathbb{R}^{n \times k \times d}$, where n, k, and d denote the batch size, maximum sequence length, and the dimensionality of the graph encoding, respectively. To facilitate the model in identifying specific paths within $\mathcal{G}^{WikiAMR}$, the relation encoder computes the shortest path between two entities. This sequence of the path is subsequently converted into a relation vector using a Gated Recurrent Unit (GRU)-based RNN (Cho et al., 2014). q_t is the sequence encoding extracted from GRU to get the relation vector r_{uv}. The mathematical formulation for this encoding is given by:

$$\overrightarrow{q}_t = \text{GRU}_f(\overrightarrow{q}_{t-1}, sp_t)$$
$$\overleftarrow{q}_t = \text{GRU}_b(\overleftarrow{q}_{t+1}, sp_t) \quad (246)$$

Here, sp_t represents the shortest path between two entities. Formally, the shortest relation path $sp_{i \to j} = [e(u, k_1), e(k_1, k_2), \ldots, e(k_n, v)]$ between the node u and the node v, where $e(\cdot, \cdot)$ indicates the edge label and $k_{1:n}$ are the relay nodes. To compute the attention scores, the final relational encoding r_{uv} is split into two distinct components, $r_{u \to v}$ and $r_{v \to u}$, via a linear transformation with a parameter matrix W_r:

$$r_{uv} = [\overrightarrow{q}_n; \overleftarrow{q}_0], \quad [r_{u \to v}; r_{v \to u}] = W_r r_{uv}$$

Subsequently, attention scores β_{uv} are calculated by incorporating both entity and relation representations from the graph $\mathcal{G}^{WikiAMR}$:

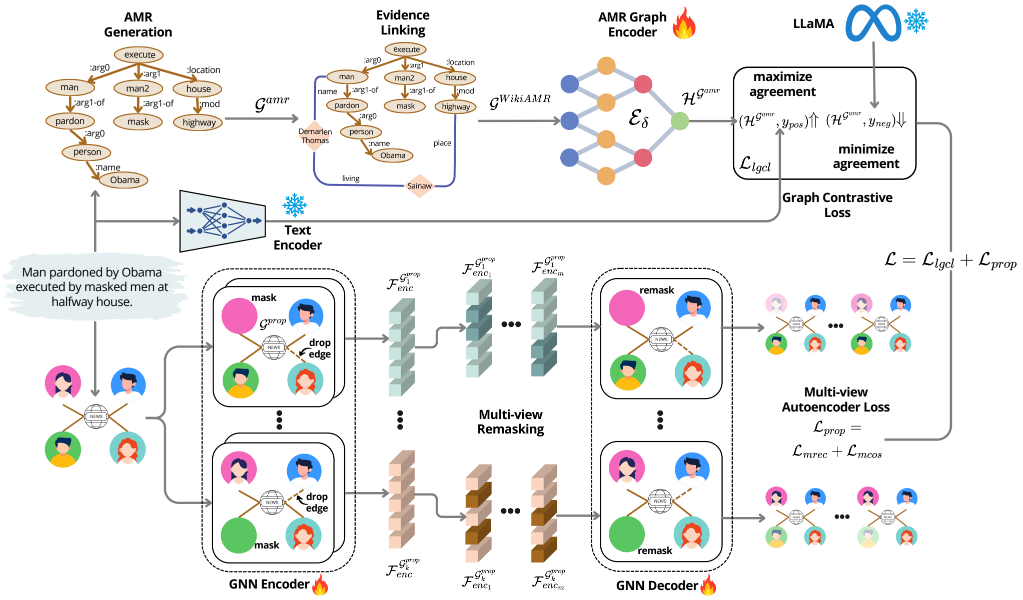

Figure 1: Overview of the proposed method: The news article is converted to an AMR graph $\mathcal{G}^{amr}$. $\mathcal{G}^{amr}$ is then linked to external evidences from Wikipedia represented as $\mathcal{G}^{WikiAMR}$. This $\mathcal{G}^{WikiAMR}$ graph is then converted to latent space features $\mathcal{H}^{\mathcal{G}^{amr}}$ by the graph transformer $\mathcal{E}_\delta$ based on $\mathcal{L}_{lgcl}$ optimization. The propagation graph of the same news article is then extracted and multiple augmentations are created. These augmented graphs are then passed to our multi-view remasked graph autoencoder which is optimized using $\mathcal{L}_{prop}$. The propagation graph feature $\mathcal{H}^{\mathcal{G}^{prop}}$ for each news is extracted from the trained GNN encoder. The final features for misinformation classification are obtained by concatenating $\mathcal{H}^{\mathcal{G}^{amr}}$ and $\mathcal{H}^{\mathcal{G}^{prop}}$.

$$\begin{aligned}\beta_{uv} &= h(e_u, e_v, r_{uv})\\ &= (e_u + r_{u\to v})W_p^\top W_k(e_v + r_{v\to u})\\ &= \underbrace{e_u W_p^\top W_k e_v}_{ⓐ} + \underbrace{e_u W_p^\top W_k r_{v\to u}}_{ⓑ}\\ &+ \underbrace{r_{u\to v} W_p^\top W_k e_v}_{ⓒ} + \underbrace{r_{u\to v} W_p^\top W_k r_{v\to u}}_{ⓓ}\end{aligned} \tag{1}$$

The attention weights computed here guide the focus on entities according to their relationships. Each term in Equation 1 serves a distinct purpose: (a) models content-based attention, (b) captures biases related to the source of the relationship, (c) addresses biases from the target, and (d) encodes a general relational bias, providing a comprehensive view of entity interactions. Finally, the Graph Transformer $(\mathcal{E}_\delta)$ encodes $\mathcal{G}^{WikiAMR}$, producing the final graph representation as follows:

$$\mathcal{H}^{\mathcal{G}^{amr}} = \mathcal{E}_\delta(\mathcal{G}^{WikiAMR}) \in \mathbb{R}^{n\times k\times d} \tag{2}$$

Here, $\mathcal{H}^{\mathcal{G}^{amr}}$ represents the output graph embeddings generated by the Graph Transformer, and d is the feature dimensionality.

Graph Contrastive Loss: Our proposed LLM-based graph contrastive loss (LGCL) function comprises two primary objectives. The first objective aims to ensure that the graph embedding remains close to its original embedding space by minimizing the reconstruction error between the predicted feature and the original feature. The second objective seeks to maximize the divergence between the predicted feature and the negative sample feature. To quantify the similarity between features, we utilize the Scaled Cosine Error (SCE) (Hou et al., 2022). Formally, given the original feature Y and the reconstructed output Y', SCE is defined as:

$$\mathcal{L}_{\text{SCE}} = \frac{1}{|\mathcal{N}|}\sum_{n\in\mathcal{N}}\left(1 - \frac{y_i^T y_i'}{\|y_i\|\cdot\|y_i'\|}\right)^\gamma, \quad \gamma \geq 1 \tag{3}$$

Here, γ is a scaling factor. When predictions have high confidence, the resulting cosine errors are generally less than 1 and diminish more quickly towards zero as the scaling factor $\gamma > 1$.

The contrastive loss requires both a positive sample feature y_{pos} and a negative sample feature y_{neg} to compare against the predicted feature. In the proposed formulation, $\mathcal{H}^{\mathcal{G}^{amr}}$ is used as y', y_{pos} is the original BERT-derived feature of the input text, while y_{neg} is a negative sample feature generated using an LLM-based negative sampler. The final contrastive loss for graph-based self-supervised learning (SSL) is formulated as follows:

$$\begin{aligned}\mathcal{L}_{lgcl} =& \mathcal{L}_{\text{SCE}}(y', y_{\text{pos}}) \\ &+ \lambda \cdot \max\left(0, m - \mathcal{L}_{\text{SCE}}(y', y_{\text{neg}})\right)\end{aligned} \tag{4}$$

Here, λ is a weighting factor, and m is the margin to ensure negatives are pushed apart in cosine space.

LLM-based Negative Sampler: We employ a large language model (LLM) in zero-shot to facilitate effective contrastive learning. Specifically, LLaMA3-7B is used to generate negative samples (y_{neg}). This approach leverages the reasoning capabilities of the LLM to distinguish between real and fake input samples, assigning them pseudo labels for the selection of the negative feature for the contrastive learning task.

Let $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ denote the set of input features. The input prompt and output format used for the LLM is mentioned in the end of the section. For each input $x_i \in \mathcal{X}$, the LLM assigns a pseudo label $\widetilde{y}_i \in \{0, 1\}$, where:

$$\widetilde{y}_i = \begin{cases} 1 & \text{if } x_i \text{ is labelled as real}, \\ 0 & \text{if } x_i \text{ is labelled as fake}. \end{cases}$$

Using the LLM's output labels, we partition the input samples into two groups:

$$\mathcal{X}_{\text{real}} = \{x_i \mid \widetilde{y}_i = 1\}, \quad \mathcal{X}_{\text{fake}} = \{x_i \mid \widetilde{y}_i = 0\}.$$

We compute the centroids of the real and fake samples as,

$$c_{\text{real}} = \frac{1}{|\mathcal{X}_{\text{real}}|} \sum_{x_i \in \mathcal{X}_{\text{real}}} \mathbf{f}_i, \quad c_{\text{fake}} = \frac{1}{|\mathcal{X}_{\text{fake}}|} \sum_{x_i \in \mathcal{X}_{\text{fake}}} \mathbf{f}_i.$$

where a feature vector $\mathbf{f}_i \in \mathbb{R}^{n \times k \times d}$ is the initial BERT feature corresponding to x_i. The negative sample (y_{neg}) is chosen to maximize the contrastive loss. In particular, we use c_{fake} as the representative negative sample for the real input sample, while c_{real} is used as the negative sample for the fake input sample. By leveraging the LLM to reason over input samples and compute these centroids, our approach effectively selects meaningful negative samples, enhancing the discriminative power of the contrastive learning model.

LLM's Zero Shot Input Prompt:
Write in one word among 'real' or 'fake' whether given text is real or fake. {text}

LLM's Output: fake/real

3.2 Multi-View Social Context and Propagation Graph Learning

Each news article is converted into a propagation graph $G^{prop} = (V, E, \mathcal{F})$ as in (Dou et al., 2021). Nodes in V represent one news article and users who forward that article. An edge in E exists between two nodes if there exists a forwarding relationship between them. The features for the news node are generated by passing the news article to a pre-trained language model (BERT), and the features for the user nodes are generated based on their recent 200 posts. The news and user node features are collectively referred to as $\mathcal{F}$.

Graph Augmentation: We use two augmentation strategies: ① feature masking and ② random edge removal for creating augmentations of the input graph as suggested in (Yin et al., 2024). For input feature masking, we randomly select 50% nodes in the graph and replace their features with a masked token. For ②, we randomly remove 20% edges from the graph. Each augmented graph for G^{prop} is denoted as $\mathcal{G}_i^{prop}$.

Graph Encoding: We encode each $\mathcal{G}_i^{prop}$ into a latent space representation using a GNN encoder. For this, we use GIN (Xu et al., 2019) represented using Equation 5 as it is theoretically proven to distinguish between graph structures.

$$f_v^{(k)} = \text{MLP}\left((1 + \epsilon) \cdot f_v^{(k-1)} + \sum_{u \in \mathcal{N}(v)} f_u^{(k-1)}\right) \tag{5}$$

Here, $f_v^{(k)}$ is embedding of node v at layer k, $\mathcal{N}(v)$ contains neighbors of node v and ϵ is a learnable scalar controlling residual connections. The final node embeddings from the encoder for each $\mathcal{G}_i^{prop}$ is represented as $\mathcal{F}_{enc}^{\mathcal{G}_i^{prop}}$.

For downstream classification tasks on G^{prop} we use the graph embedding $\mathcal{H}^{G^{prop}}$ calculated as:

$$\mathcal{H}^{G^{prop}} = \frac{1}{|V|} \sum_{v \in V} f_v \in \mathcal{F}_{enc}^{G^{prop}} \tag{6}$$

Multi-View Graph Decoding: Now, from the encoded node representations $\mathcal{F}_{enc}^{\mathcal{G}_i^{prop}}$, we decode the input node features $\mathcal{F}$ using GIN as a decoder. In (Yin et al., 2024) the authors use a single stage remasking for each $\mathcal{F}_{enc}^{\mathcal{G}_i^{prop}}$ to reconstruct the input features. But authors in (Hou et al., 2023) have shown that feature reconstruction is susceptible to congruence among the input features, which single remasking cannot address. To address this, we introduce multi-view feature remasking of each augmented graph $\mathcal{F}_{enc}^{\mathcal{G}_i^{prop}}$. Each remasked encoded feature is denoted by $\mathcal{F}_{enc_j}^{\mathcal{G}_i^{prop}}$. It acts as a regularizer for the decoder, making it robust against unexpected noises in input and helping to avoid overfitting. The final objective of the decoder is to reconstruct the actual node features $\mathcal{F}$ from these masked encoded node features using the multi-view autoencoder loss described next.

Multi-View Autoencoder Loss: Given k augmentations of the input graph $\mathcal{G}^{prop}$ represented as $\mathcal{G}_1^{prop}, \ldots, \mathcal{G}_k^{prop}$, and m remasked decoded output for each augmented graph represented as $\mathcal{F}_{dec_1}^{\mathcal{G}_1^{prop}}, \ldots, \mathcal{F}_{dec_m}^{\mathcal{G}_1^{prop}}, \ldots, \mathcal{F}_{dec_m}^{\mathcal{G}_k^{prop}}$, we define the multi-view reconstruction loss as

$$\mathcal{L}_{mrec} = \sum_{i=1}^{k} \sum_{j=1}^{m} (\mathcal{F} - \mathcal{F}_{dec_j}^{\mathcal{G}_i^{prop}}) \tag{7}$$

To minimize the divergence across the views of the decoded features, we define the multi-view cosine similarity loss as

$$\mathcal{L}_{mcos} = \underset{\substack{\forall l,i,j;\text{ if } l=l' \text{ then } i\neq j \\ l\leq k, i\leq m, j\leq m}}{\mathcal{M}} \frac{\mathcal{F}_{dec_i}^{\mathcal{G}_l^{prop}} . \mathcal{F}_{dec_j}^{\mathcal{G}_{l'}^{prop}}}{\left|\left|\mathcal{F}_{dec_i}^{\mathcal{G}_l^{prop}}\right|\right| \cdot \left|\left|\mathcal{F}_{dec_j}^{\mathcal{G}_{l'}^{prop}}\right|\right|} \tag{8}$$

Here, $\mathcal{M}$ is the mean operation. Our final propagation loss is $\mathcal{L}_{prop} = \mathcal{L}_{mrec} + \mathcal{L}_{mcos}$.

3.3 Final Loss

We combine the AMR and Propagation loss as $\mathcal{L} = \mathcal{L}_{lgcl} + \mathcal{L}_{prop}$. We train our model using this loss, and the final features of our model are $\mathcal{H}^{\mathcal{G}^{amr}} \cdot \mathcal{H}^{\mathcal{G}^{prop}}$. These features are then used for misinformation classification.

4 Experiments and Results

We perform experiments on the publicly available datasets FakeNewsNet (Shu et al., 2020) in order to assess the effectiveness of the model. This repository contains two separate benchmark datasets, namely, PolitiFact and GossipCop. We cover the datasets and supervised and unsupervised baselines in more details in the Appendix.

5 Results

We conducted a comparative analysis of our model against various unsupervised and supervised baselines on the PolitiFact and GossipCop datasets. As shown in Table 2, our model achieved the highest accuracy (0.919), precision (0.933), recall (0.903), and F1-score (0.918) among the unsupervised baselines. Compared to GAMC, the existing benchmark, our model outperforms it by a margin of 8.1% in accuracy and 8.7 points in F1-score (on the absolute scale). Also, our model surpasses GTUT and (UMD)^2 by significant margins, $12 \sim 14\%$ in accuracy and $14 \sim 15$ points in the F1-score, indicating a superior ability to differentiate between fake and real news. In a similar context, as shown in Table 3, our model significantly outperforms existing unsupervised baselines on the GossipCop dataset. It achieves the highest accuracy (0.968), precision (0.965), recall (0.967), and F1-score (0.966), outperforming GAMC, which attained an accuracy of 0.946 and an F1-score of 0.943. This represents a 2.2% improvement in accuracy and a 2.3 point improvement in the F1-score. This improvement can be attributed to the proposed model's unique design, which leverages a combination of self-supervised AMR semantic features and news propagation features from multi-view social context graph learning.

When we compare our model to supervised baselines on both PolitiFact and GossipCop datasets (Table 4), it consistently outperforms state-of-the-art approaches in terms of accuracy, while comparable results on F1 score are observed. On PolitiFact, our model achieves an accuracy of 0.919 and an F1-score of 0.933, surpassing EA^{2}N with BERT (0.911 accuracy, 0.915 F1-score), GACL (0.867 accuracy, 0.866 F1-score), and EANN (0.804 accuracy, 0.798 F1-score). However, it shows comparative performance with dEFENED in F1-score. On GossipCop, our model outperforms all supervised baselines, achieving the highest accuracy (0.968) and F1-score (0.966). It notably surpasses GACL (0.907 accuracy, 0.905 F1-score) and EA^{2}N (0.844 accuracy, 0.872 F1-score), as well as dEFEND, which lags significantly behind with 0.808 accuracy and 0.755 F1-score. These results highlight that while supervised models perform well, our self-supervised approach not only competes effectively on PolitiFact but outperforms all supervised baselines on GossipCop, demonstrating superior performance across datasets.

Table 2: Comparative study of our model w.r.t. different unsupervised baselines on PolitiFact dataset.

Methods	Acc	Pre	Rec	F1
TruthFinder	0.581	0.572	0.576	0.573
UFNDA	0.685	0.667	0.659	0.670
UFD	0.697	0.652	0.641	0.647
GTUT	0.776	0.782	0.758	0.767
$(\text{UMD})^2$	0.802	0.795	0.748	0.761
GAMC	0.838	0.836	0.827	0.831
Ours	**0.919**	**0.933**	**0.903**	**0.918**
variance	± 0.019	± 0.045	± 0.058	± 0.020

Table 3: Comparative study of our model w.r.t. different unsupervised baselines on GossipCop dataset.

Methods	Acc	Pre	Rec	F1
TruthFinder	0.668	0.669	0.672	0.669
UFNDA	0.692	0.687	0.662	0.673
UFD	0.662	0.687	0.654	0.667
GTUT	0.771	0.770	0.731	0.744
$(\text{UMD})^2$	0.792	0.779	0.788	0.783
GAMC	0.946	0.941	0.946	0.943
Ours	**0.968**	**0.965**	**0.967**	**0.966**
variance	± 0.015	± 0.026	± 0.039	± 0.015

Table 4: Comparative study of our model with supervised methods on PolitiFact and GossipCop datasets.

Dataset	PolitiFact		GossipCop	
	Acc	F1	Acc	F1
SAFE	0.793	0.775	0.832	0.811
EANN	0.804	0.798	0.836	0.813
dEFEND	0.904	**0.928**	0.808	0.755
GACL	0.867	0.866	0.907	0.905
EA^2N	0.911	0.915	0.844	0.872
Ours	**0.919**	0.918	**0.968**	**0.966**

6 Ablation Study

Change in classification result with different values of λ: Figure 2 shows the change in classification accuracy of the proposed method with the change in weightage to negative samples in Equation 4. It is evident that the accuracy improved initially with the value of λ and obtained the maximum result when $\lambda = 0.5$ for both datasets. With a further increase in λ, the accuracy decreases, indicating that our model overemphasizes negative samples compared to being close to positive samples, thus decreasing feature separability. Based on this study, we set the value of λ to 0.5 in our experiments.

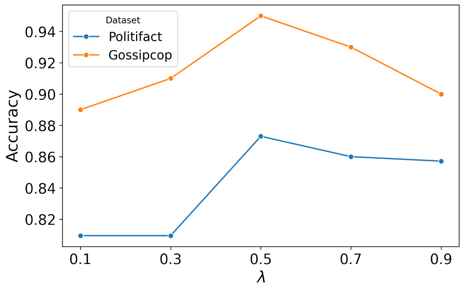

Figure 2: Change in classification result with different values of λ.

Table 5: Results on different split sizes for PolitiFact and GossipCop datasets.

Test Size %	PolitiFact		GossipCop	
	Acc	F1	Acc	F1
10	0.937	0.937	0.971	0.971
20	0.919	0.918	0.968	0.966
30	0.885	0.889	0.955	0.956
40	0.878	0.883	0.957	0.958
50	0.876	0.881	0.958	0.958
60	0.857	0.866	0.959	0.959
70	0.854	0.858	0.956	0.956
80	0.852	0.851	0.955	0.955
90	0.844	0.849	0.952	0.952

Change in classification result with training size: We study the effect of our features on misinformation classification with different training sizes for linear SVM. The results are shown in the Table 5. As expected, the accuracy decreases with an increase in test size; however, the proposed model results in better accuracy than the unsupervised methods with few training samples. It is evident from Tables 4-5 that with only 10% training, our result surpasses the results of supervised methods for GossipCop dataset, while it is better than in 3 out 5 models with only 50% training points for PolitiFact dataset.

Change in results with varying number of augmentations k and multi-view remaskings m: We study the change in classification accuracy with different numbers of augmentations and remaskings for the PolitiFact dataset (Figure 3). We can infer from the figure that the best results are obtained when we set $k = 2$ and $m \leq 6$. This shows that multi-view remaskings help the model achieve superior performance, but more than three remaskings do not bring considerable improvements.

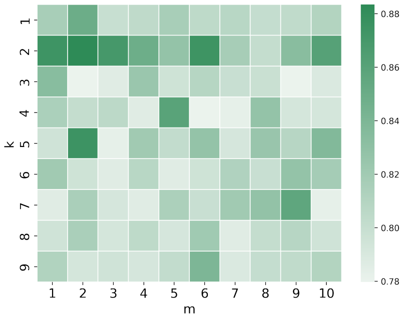

Figure 3: Change in accuracy with varying number of augmentation k and multi-view remasking m.

Table 6: Accuracy Score for different components of the model.

Model	PolitiFact		GossipCop	
	Acc	F1	Acc	F1
Mistral (Zero-shot)	0.747	0.636	0.610	0.320
LLaMA (Zero-shot)	0.804	0.749	0.680	0.535
Only $\mathcal{L}_{lgcl}$+ Mistral	0.822	0.830	0.934	0.932
Only $\mathcal{L}_{lgcl}$+ LLaMA	0.841	0.828	0.948	0.949
Only $\mathcal{L}_{prop}$	0.846	0.845	0.946	0.945
$\mathcal{L}_{lgcl} + \mathcal{L}_{prop}$+ Mistral	0.893	0.892	0.938	0.938
$\mathcal{L}_{lgcl} + \mathcal{L}_{prop}$+ LLaMA	**0.919**	**0.918**	**0.968**	**0.966**

Change in classification results with different components of our model: In Table 6, we show the importance of different components of our model. All the results shown here use 80% labelled data in the final linear SVM for training. As we can see from the table, $\mathcal{L}_{lgcl}$ and $\mathcal{L}_{prop}$ individually produce comparable results. But we get significant improvements in classification accuracy when we combine features generated using $\mathcal{L} = \mathcal{L}_{lgcl} + \mathcal{L}_{prop}$. We also compare the performance of our model with varying versions of the LLM. We use two popular models, Mistral-7B and LLaMA-7B. We show the results when we use the LLMs independently for zero-shot classification. Our model significantly improves the classification results using information from the LLM. One must also note that there is a significant difference between the results from the two LLMs when used independently without our model. But, when used with any component of our model, this difference reduces, thus showing the robustness of the extracted features by the proposed method.

Qualitative results at different stages of our proposed pipeline In Figure 4 we show the feature separation between the real and fake news at different stages of our proposed pipeline. In the first row of the Figure we see the results of PolitiFact dataset and the second row we show the results of the GossipCop dataset. The first column of each row shows the TSNE embedding of the initial features. The second column shows the TSNE plot of the original features after an MLP layer. The third column shows the TSNE plot of the features obtained after the self-supervised AMR graph learning ($\mathcal{H}^{\mathcal{G}^{amr}}$) phase trained with a linear layer. The last columns shows the TSNE plot of the final concatenated features after self-supervised AMR graph learning and multi-view propagation graph learning ($\mathcal{H}^{\mathcal{G}^{amr}}.\mathcal{H}^{\mathcal{G}^{prop}}$) with a linear layer. In all the cases we train the MLP with 80% labelled data. We can see from the results that the feature separation increases after each stage of the pipeline, thus showing the effectiveness of our model.

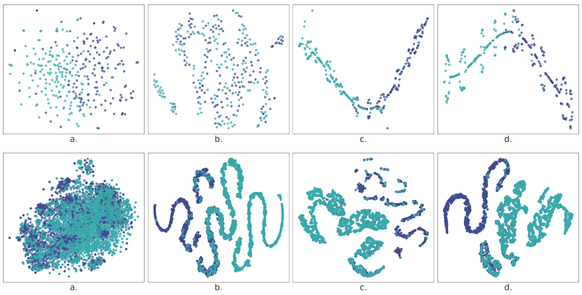

Figure 4: The TSNE plots showing the embeddings of PolitiFact (Row1) and GossipCop (Row2).

7 Conclusion

This study presents a novel self-supervised approach for misinformation detection. The LLM-based contrastive self-supervised AMR learning framework captures complex semantic relationships in text. This method enhances feature separation between real and fake news by leveraging an LLM-based negative sampler. Additionally, we introduce a multi-view graph-masked autoencoder that integrates social context and news propagation patterns for more robust detection. Through extensive experiments, the proposed method is found to produce state-of-the-art performance. Beyond misinformation detection, our methodology has broader applications in NLP. For instance, self-supervised AMR graph learning can be applied to tasks like question-answering and event detection, while multi-view social context and propagation graph learning can be leveraged for hate speech and aggression detection, etc. This work not only advances misinformation detection but also lays the groundwork for tackling various NLP challenges using graph-based learning in constraint settings.

8 Limitations

We have already mentioned the advantages of our proposed model in the previous sections. In this section we highlight some limitations of the proposed model. Our work is primarily dominated by the US centric dataset. This was primarily because propagation data for any other language was not available and we could not collect data due to restrictions from X. In the future works we would like to extend this work to datasets of other countries by collecting data from platforms other than X. Also, all the news articles here are in English, in the future we would like to extend our work to multi-lingual data. Finally, we would like to improve our self-supervised AMR graph learning by incorporating reasoning based agentic AI instead of LLMs for finding negative samples.

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kev Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria.

Deng Cai and Wai Lam. 2020. Graph transformer for graph-to-sequence learning. In *AAAI*, pages 7464–7471. AAAI Press.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *EMNLP*, pages 1724–1734, Doha, Qatar. ACL.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Hernawan Dewatana and Siti Ummu Adillah. 2021. The effectiveness of criminal eradication on hoax information and fake news. *Law Development Journal*, 3(3):513–520.

Yingtong Dou, Kai Shu, Congying Xia, Philip S. Yu, and Lichao Sun. 2021. User preference-aware fake news detection. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2051–2055, New York, NY, USA. Association for Computing Machinery.

Yaqian Dun, Kefei Tu, Chen Chen, Chunyan Hou, and Xiaojie Yuan. 2021. Kan: Knowledge-aware attention network for fake news detection. *AAAI*, 35(1):81–89.

Song Feng, Ritwik Banerjee, and Yejin Choi. 2012. Syntactic stylometry for deception detection. In *ACL (Volume 2: Short Papers)*, pages 171–175, Jeju Island, Korea. ACL.

Siva Charan Reddy Gangireddy, Deepak P, Cheng Long, and Tanmoy Chakraborty. 2020. Unsupervised fake news detection: A graph-based approach. In *Proceedings of the 31st ACM Conference on Hypertext and Social Media*, HT '20, page 75–83, New York, NY, USA. Association for Computing Machinery.

Bilal Ghanem, Simone Paolo Ponzetto, Paolo Rosso, and Francisco Rangel. 2021. Fakeflow: Fake news detection by modeling the flow of affective information. In *16th EACL*.

Shubham Gupta, Abhishek Rajora, and Suman Kundu. 2025. Ea2n: Evidence-based amr attention network for fake news detection. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–12.

Shubham Gupta, Narendra Yadav, Suman Kundu, and Sainathreddy Sankepally. 2023. Fakedamr: Fake news detection using abstract meaning representation network. In *International Conference on Complex Networks and Their Applications*, pages 308–319. Springer.

Zhenyu Hou, Yufei He, Yukuo Cen, Xiao Liu, Yuxiao Dong, Evgeny Kharlamov, and Jie Tang. 2023. Graphmae2: A decoding-enhanced masked self-supervised graph learner. In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 737–746, New York, NY, USA. Association for Computing Machinery.

Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. 2022. Graphmae: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 594–604, New York, NY, USA. Association for Computing Machinery.

Linmei Hu, Tianchi Yang, Luhao Zhang, Wanjun Zhong, Duyu Tang, Chuan Shi, Nan Duan, and Ming Zhou. 2021. Compare to the knowledge: Graph neural fake news detection with external knowledge. In *ACL-IJCNLP (Volume 1: Long Papers)*, pages 754–763, Online. ACL.

Thomas N. Kipf and Max Welling. 2016. Variational graph auto-encoders. *Preprint*, arXiv:1611.07308.

Dun Li, Haimei Guo, Zhenfei Wang, and Zhiyun Zheng. 2021. Unsupervised fake news detection based on autoencoder. *IEEE Access*, 9:29356–29365.

Shaohua Li, Weimin Li, Alex Munyole Luvembe, and Weiqin Tong. 2024. Graph contrastive learning with feature augmentation for rumor detection. *IEEE Transactions on Computational Social Systems*, 11(4):5158–5167.

Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. 2022. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12):6999–7019.

Yang Liu and Yi-Fang Wu. 2018. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. *AAAI*, 32(1).

Yunfei Long, Q Lu, Rong Xiang, Minglei Li, and Chu-Ren Huang. 2017. Fake news detection through multi-perspective speaker profiles. In *IJCNLP (Volume 2: Short Papers)*, pages 252–256, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J. Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016a. Detecting rumors from microblogs with recurrent neural networks. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 3818–3824. AAAI Press.

Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J. Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016b. Detecting rumors from microblogs with recurrent neural networks. In *IJCAI*, IJCAI'16, page 3818–3824. AAAI Press.

Erxue Min, Yu Rong, Yatao Bian, Tingyang Xu, Peilin Zhao, Junzhou Huang, and Sophia Ananiadou. 2022. Divide-and-conquer: Post-user interaction network for fake news detection on social media. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 1148–1158, New York, NY, USA. Association for Computing Machinery.

Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. 2017. Where the truth lies: Explaining the credibility of emerging claims on the web and social media. WWW '17 Companion, page 1003–1012, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. 2019. defend: Explainable fake news detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 395–405, New York, NY, USA. Association for Computing Machinery.

Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. 2020. Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big Data*, 8(3):171–188.

Amila Silva, Ling Luo, Shanika Karunasekera, and Christopher Leckie. 2024. Unsupervised Domain-Agnostic Fake News Detection Using Multi-Modal Weak Signals . *IEEE Transactions on Knowledge & Data Engineering*, 36(11):7283–7295.

Tiening Sun, Zhong Qian, Sujun Dong, Peifeng Li, and Qiaoming Zhu. 2022. Rumor detection on social media with graph adversarial contrastive learning. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 2789–2797, New York, NY, USA. Association for Computing Machinery.

Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. 2018. Eann: Event adversarial neural networks for multi-modal fake news detection. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 849–857, New York, NY, USA. Association for Computing Machinery.

Lirong Wu, Haitao Lin, Cheng Tan, Zhangyang Gao, and Stan Z. Li. 2023. Self-supervised learning on graphs: Contrastive, generative, or predictive. 35(4):4216–4235.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How powerful are graph neural networks? In *International Conference on Learning Representations*.

Ruichao Yang, Xiting Wang, Yiqiao Jin, Chaozhuo Li, Jianxun Lian, and Xing Xie. 2022. Reinforcement subgraph reasoning for fake news detection. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 2253–2262, New York, NY, USA. Association for Computing Machinery.

Huaxiu Yao, Ying-xin Wu, Maruan Al-Shedivat, and Eric Xing. 2021. Knowledge-aware meta-learning for low-resource text classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1814–1821, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shu Yin, Peican Zhu, Lianwei Wu, Chao Gao, and Zhen Wang. 2024. Gamc: An unsupervised method for fake news detection using graph autoencoder with masking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(1):347–355.

Xiaoxin Yin, Jiawei Han, and Philip S. Yu. 2008. Truth discovery with multiple conflicting information providers on the web. *IEEE Transactions on Knowledge and Data Engineering*, 20(6):796–808.

Chunyuan Yuan, Qianwen Ma, Wei Zhou, Jizhong Han, and Songlin Hu. 2019. Jointly Embedding the Local and Global Relations of Heterogeneous Graph for Rumor Detection . In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 796–805, Los Alamitos, CA, USA. IEEE Computer Society.

Sheng Zhang, Xutai Ma, Kev Duh, and Benjamin Van Durme. 2019. AMR parsing as sequence-to-graph transduction. In *ACL*, pages 80–94, Florence, Italy. ACL.

Yizhou Zhang, Loc Trinh, Defu Cao, Zijun Cui, and Yan Liu. 2023. Detecting out-of-context multimodal misinformation with interpretable neural-symbolic model. *Preprint*, arXiv:2304.07633.

Xinyi Zhou, Jindi Wu, and Reza Zafarani. 2020. Safe: Similarity-aware multi-modal fake news detection. In *Advances in Knowledge Discovery and Data Mining*, pages 354–367, Cham. Springer International Publishing.

A Details on Datasets, Baselines and Implementation

PolitiFact is dedicated to news coverage revolving around U.S. political affairs, while GossipCop delves into stories about Hollywood celebrities. These datasets also capture the broader social dynamics by including information about how news spreads through networks and the posting patterns of users. We evaluate our model using a set of metrics, including Precision (Pre), Recall (Rec), F1-score, and Accuracy (Acc). Comprehensive details of the datasets are provided in Table 7.

Table 7: Datasets Statistics

	# News	# True	# Fake	# Nodes	# Edges
PolitiFact	314	157	157	41054	40740
GossipCop	5464	2732	2732	314262	308798

Baselines: In our evaluation, we contrast our model with various state-of-the-art baselines, categorized into two groups. The first group utilizes only unsupervised methods (**TruthFinder** (Yin et al., 2008), **UFNDA** (Li et al., 2021), **UFD** (Yang et al., 2022), **GTUT** (Gangireddy et al., 2020), **(UMD)**[2] (Silva et al., 2024), **GAMC** (Yin et al., 2024)), while the second incorporates supervised methods (**SAFE** (Zhou et al., 2020), **EANN** (Wang et al., 2018), **dEFEND** (Shu et al., 2019), **GACL** (Sun et al., 2022), **EA^{2}N (BERT)** (Gupta et al., 2025)).

Implementation Details: In order to generate the AMR graph, we have used a pretrained STOG model (Zhang et al., 2019). For LGCL, we use $\alpha = 0.5$ and in order to integrate the evidence in the AMR graph, we use the same parameters described in (Gupta et al., 2025). For social context and propagation graph learning we use 2 encoder layers and 1 decoder layer. For multi-view remasking, we select $k = 2$ and $m = 2$. We selected Support Vector Machine (SVM) as the final classifier and reported the results from 5-fold cross-validation. Although we provided our results for each test size percentage in Table 5, our main results are based on an 80:20 train-test split to ensure consistency with other methods. We have trained our model on RTX A5000 Nvidia GPU with 24 GB GPU memory. The training of AMR took 1 hour for PolitiFact and took 3 hours for the GossipCop dataset with 50 epochs. Multi-view masked graph learning took 5 mins for the PolitiFact dataset and 15 minutes for the GossipCop dataset.