LLM-ENHANCED RUMOR DETECTION VIA VIRTUAL NODE INDUCED EDGE PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

The proliferation of rumors on social networks poses significant challenges to information credibility and public trust. The dissemination of rumors forms complex networks, yet existing rumor detection methods exhibit several limitations, including a limited capacity to capture complex propagation features. Representing each node solely through its textual embeddings neglects the textual coherence across the entire rumor propagation path, which undermines the accuracy of rumor identification on social platforms. To address these challenges, this study proposes a novel framework for rumor detection on social media, which captures latent characteristics of rumor propagation and enhances contextual correlation within rumor graphs through large language models (LLMs). We introduce a novel paradigm for effectively leveraging LLMs, utilizing their powerful linguistic capabilities to analyze complete information flows within subchains, assign rumor probabilities, and guide the construction of connections between virtual nodes and selected subchain nodes. This enables the modification of the original graph structure, which is a critical advancement for capturing subtle rumor signals. Given the inherent limitations of LLMs in rumor identification, we develop a structured prompt framework to mitigate model biases and ensure robust graph learning performance. Additionally, the proposed framework is model-agnostic, meaning it is not constrained to any specific graph learning algorithm or LLMs. Its plug-and-play nature allows for seamless integration with further fine-tuned LLMs and graph techniques in the future, potentially enhancing predictive performance without modifying original algorithms.

1 INTRODUCTION

While social media's rapid expansion has revolutionized information sharing, it has simultaneously accelerated the spread of rumors, threatening information credibility and societal stability Ziari and Charkari (2025). Detecting rumors in social networks is a critical yet challenging task, as rumors typically propagate downward along tree-like graph structures over time, with nodes (representing users or posts) and edges (indicating interactions such as replies or retweets) reflecting the flow of information. This temporal nature, combined with the semantic complexity of textual data, requires modeling approaches that can effectively capture structural patterns, temporal evolution, and deep semantic content.

Traditional rumor detection methods, while achieving some success, are constrained by notable limitations. Early machine learning approaches that rely on hand-crafted features struggle to adapt to the diversity and noise inherent in social media data (Zubiaga et al., 2018). Similarly, conventional Graph Neural Networks (GNNs), which are primarily designed to model local graph structures, frequently prove inadequate in capturing the subtle rumor propagation pathways and diffusion patterns evident in subchains of varying lengths. These shortcomings highlight the need for more semantically aware modeling techniques. LLMs excel at extracting rich semantic features from text, presenting a promising opportunity to address the semantic shortcomings of traditional GNNs (Wang et al., 2024). However, our experiments reveal a critical limitation of LLMs in rumor detection: when used solely to classify news as rumor or non-rumor, LLMs exhibit a significant bias, consistently favoring a "non-rumor" classification. By tolerating more false negatives to avoid false positives, this leniency increases the risk of the system accepting rumors as truth. Consequently, standalone LLM-based approaches may be inherently unreliable for robust rumor detection. To address these challenges, we propose a novel framework that integrates LLMs and GNNs, leveraging structured subchain propagation patterns to enhance rumor detection accuracy. Our method uses LLMs to analyze information flow within tweet subchains, capturing key propagation patterns and semantic cues. Through well-designed prompts, we enhance the LLMs' ability to process large-scale network data and reduce their bias toward rumor classification. Our framework begins with the introduction of a virtual node labeled "is rumor" and augments the graph based on fine-grained rumor probabilities generated by LLMs for each subchain. The enriched graph is subsequently processed to derive robust node representations, as demonstrated in Section 4 using a Bidirectional Graph Attention Network (Bi-GAT), to facilitate accurate rumor detection through link prediction between the root node and the virtual node. In summary, this study offers two principal contributions: (1) We propose a novel, model-agnostic framework that synergistically integrates LLMs and GNNs for

rumor detection, enabling effective fusion of semantic and structural information while mitigating LLM bias. (2) We develop an innovative approach to capture previously unexploited propagation features by restructuring the graph with a virtual node and subchain-based connections, enhancing the detection of complex rumor diffusion patterns.

2 RELATED WORK

2.1 RUMOR DETECTION

Rumor detection has become a vital research area, especially in social media. Early studies relied on handcrafted features and traditional machine learning methods, such as Support Vector Machines, Naive Bayes, and Decision Trees, using linguistic, user, and timing clues, though these methods struggled with the scale and complexity of social media. The introduction of deep learning marked major progress, as Convolutional Neural Networks were applied to text and image data to capture spatial features. Subsequently, Recurrent Neural Networks (RNNs), especially with LSTM units, excelled at modeling temporal rumor propagation, with Ma et al. (2016) pioneering RNNs for rumor detection in microblogs through modeling user interaction sequences. Further improvements came with Transformer-based models like BERT and GPT-2, whose pre-trained embeddings grasp subtle semantic features in rumor texts, boosting performance on datasets such as PHEME, Twitter15, Twitter16, and Weibo. Ma et al. (2017) extended this by incorporating propagation structures into kernel learning, revealing how rumors spread through networks.

2.2 LARGE LANGUAGE MODELS IN FAKE NEWS DETECTION

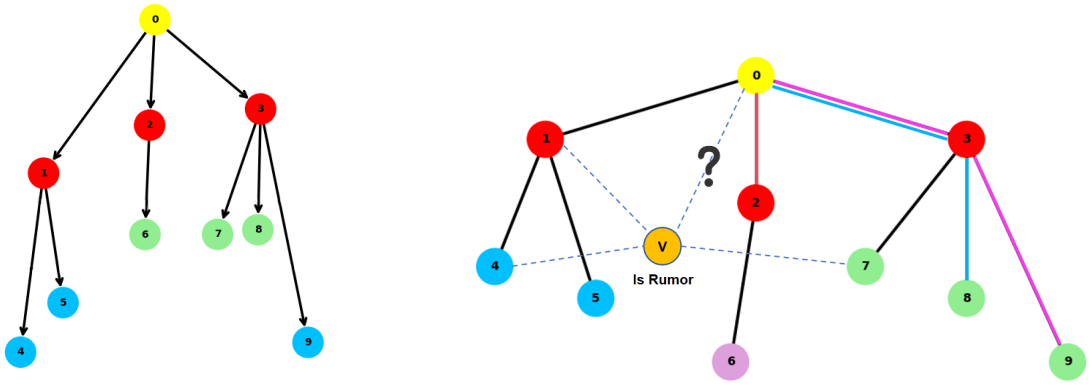
LLMs have emerged as a powerful tool in combating fake news, demonstrating capabilities beyond traditional machine learning methods. A common approach involves leveraging LLMs as judges to assess information credibility in a manner akin to human experts (Zhou et al., 2023), capitalizing on their deep understanding of language, context, and world knowledge, with the scalability of this judging role further demonstrated by Li et al. (2024), who showed that LLMs can perform consistent, large-volume credibility checks. Another line of research improves LLMs via parameter-efficient tuning; for example, Cheung and Lam (2023) applied Supervised Fine-Tuning with Low-Rank Adaptation to incorporate external knowledge. However, such techniques frequently encounter difficulties due to unreliable training samples. This issue was addressed by Tian et al. (2025) through their framework, which uses influence scores to selectively curate data and mitigate the impact of noisy data. Beyond serving as standalone assessors, LLMs also enhance classification systems through data augmentation to mitigate data scarcity, as demonstrated by Lai et al. (2024) through generating synthetic fake news samples to balance datasets and improve performance. Furthermore, their ability to model complex information supports novel detection frameworks, such as the method introduced by Ma et al. (2024b), which partitions crowd wisdom into competing viewpoints, uses LLMs to generate reasoned arguments for each perspective, and determines truthfulness through a defense-inspired verification process.

2.3 GRAPH METHODS IN RUMOR DETECTION

Graph-based methods have become essential for rumor detection on social media, as they effectively capture the natural propagation patterns of information through networks. Rumors often display distinctive structural features, such as retweet chains or branching reply threads, which can be naturally represented using graph models. Central to these approaches are Graph Convolutional Networks (GCNs), which aggregate features from connected nodes to learn robust node embeddings. For example, Bian et al. (2020) introduced a Bi-Directional GCN that improves upon previous models by capturing bidirectional information flow, aiding in the identification of critical nodes. Further advancing this line, Wu et al. (2020) developed propagation graph neural networks that incorporate attention mechanisms to model non-sequential propagation dynamics, enabling more accurate capture of complex rumor diffusion paths. Meanwhile, Sun et al. (2022) proposed a graph adversarial Contrastive Learning framework that enhances detection robustness through adversarial feature transformations on propagation graphs. More recently, Liu et al. (2024) designed a novel GNN that constructs a bipartite graph to model user correlations while simultaneously capturing propagation patterns via tree-structured graphs, effectively integrating social context with diffusion topology. A promising new direction involves combining LLMs with graph techniques to enable deeper semantic analysis of information diffusion within static structures. Reflecting this trend, Ma et al. (2024a) introduced a graph sampling and aggregation model (GSMA), which enhances GraphSAGE (Hamilton et al., 2017) with dynamic attention aggregation, modulated positional encodings, and sentiment-aware semantic features for improved rumor detection.

3 PROBLEM STATEMENT

Let $\mathcal{N} = \{N_1, N_2, \ldots, N_s\}$ denote the set of source news articles, where each $N_i \in \mathcal{N}$ represents an individual news item. Associated with each N_i is a set of reactions $\mathcal{R}_i = \{R_{i1}, R_{i2}, \ldots, R_{it_i}\}$, forming a multi-level discussion structure beneath each original news post. Specifically, 1) each news post contains a large number of user replies; 2) these replies may directly respond to the news post (first-level replies) or be nested under existing replies as sub-replies (multi-level replies), thus forming a tree-like dialogue structure. This tree propagation structure corresponds exactly to the way rumors spread. As shown in Figure 1(a), the yellow root node at the top represents the source news, and the child nodes below are all reaction posts. In rumor detection tasks, the



(a) Rumor propagation patterns in social media

(b) Virtual node and shortest path subchains

Figure 1: Combined figure showing two patterns

problem is formulated as a binary classification task. The objective is to train a model using news instances labeled with ground-truth values $y \in \{\text{Rumor}, \text{Non-Rumor}\}$, enabling accurate prediction of labels for unseen test news items. Our work focuses on developing a model-agnostic framework for rumor detection, designed to seamlessly integrate with various graph learning algorithms and LLMs.

4 ALGORITHM

For each source news, a directed graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ is constructed: $\mathbb{V} = \{v_0, v_1, \ldots, v_n\}$ is the set of nodes, where v_0 is the source news (root node) and each child node $v_i \in \mathbb{V} \setminus \{v_0\}$ represents a reply post. $\mathbb{E}$ is the set of edges and the direction of the edge is the same as the direction of information propagation. The text information associated with node v_i is denoted as $\text{text}(v_i)$. Each node v_i is assigned a feature vector x_i, extracted using BERT as follows:

$$x_i = \text{BERT}(\text{text}(v_i)).$$

Subchain Construction: For each child node $v_i \in \mathbb{V} \setminus \{v_0\}$, there exists a unique path (subchain) from v_i to the root node v_0, denoted as:

$$\text{path}(v_0, v_i) = (v_0, v_{p_1}, v_{p_2}, \ldots, v_{p_k}, v_i),$$

where v_{p_j} are intermediate nodes on the path, and $k \geq 0$. The text information of the subchain is concatenated using a separator token [SEP]:

$$\text{info}(\text{path}(v_0, v_i)) = \text{text}(v_0)\ \text{[SEP]}\ \text{text}(v_{p_1})\ \text{[SEP]}\ \cdots\ \text{[SEP]}\ \text{text}(v_i).$$

This chain-like information flow captures rich contextual information from the root to each child node. By leveraging this structure, our approach aims to enhance the capability of LLMs and graph learning algorithms to detect rumors effectively. As depicted in Figure 1(b), the red path from the root node to child node 2 represents the subchain associated with child node 2, while the purple path from the root node through child node 3 to child node 9 represents the subchain associated with child node 9. By analogy, the entire graph contains as many subchains as there are child nodes, with each subchain corresponding to the unique path from the root node to a child node.

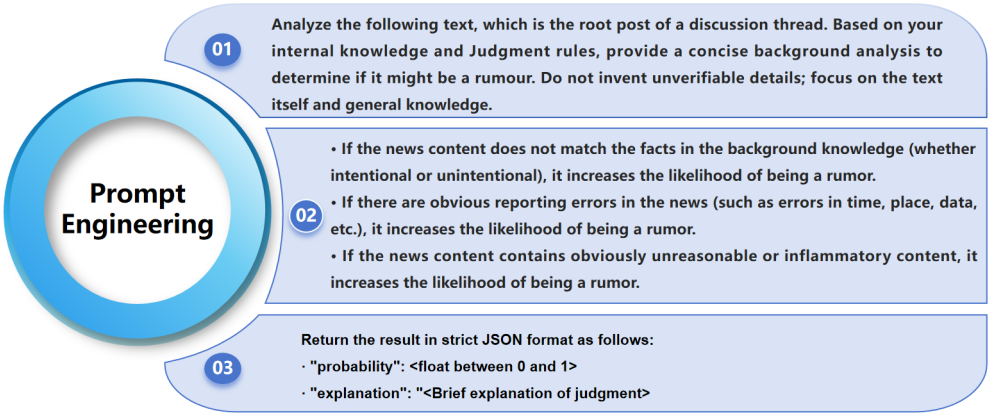


Figure 2. Prompt Engineering

Virtual Node and Edge Augmentation in Graph Structures: To leverage the capabilities of LLMs and the graphical structure of information flows, we introduced a virtual node v_{virtual}, labeled "is Rumor", which initially lacks feature information. We then use a LLM to process each subchain and outputs the probability that the source news is a rumor. For each child node v_i, the LLM takes the concatenated text information of the subchain and outputs a probability:

$$P_{\text{rumor}}(v_i) = \text{LLM}(\text{info}(\text{path}(v_0, v_i))), \tag{1}$$

where $P_{\text{rumor}}(v_i) \in [0, 1]$ represents the likelihood that the source news is a rumor based on the subchain from the root node to node v_i. For each child node v_i, if the LLM probability exceeds a predefined threshold θ:

$$P_{\text{rumor}}(v_i) > \theta. \tag{2}$$

For a node v_i satisfying (2), a bidirectional edge is established between v_i and v_{virtual}, denoted as $(v_i, v_{\text{virtual}}), (v_{\text{virtual}}, v_i) \in \mathbb{E}'$, where $\mathbb{E}'$ is the set of new edges added to the graph. The updated graph is denoted as $\mathbb{G}' = (\mathbb{V} \cup \{v_{\text{virtual}}\}, \mathbb{E} \cup \mathbb{E}')$. As shown in Figure 1(b), assume that the LLM assigns probabilities to the subchains of child nodes 1, 4, and 7 exceeding the threshold, edges are established between these nodes and the virtual node. Through these operations, we modify the graph structure by adding virtual nodes and leverage the logical reasoning and knowledge base of LLMs to establish edges between the virtual node and other subnodes. Intuitively, the virtual node's representation is derived by aggregating features from connected child nodes, which are identified as rumors with high probability based on the subchain's information. Consequently, the problem of rumor detection is reduced to predicting the linkage between the root node and the virtual node, which can be accomplished using various GNN approaches.

Next we provide further details on how to retrieve the probability (1) from a LLM. It has been shown that carefully designed prompt engineering can enhance the ability of LLMs to identify rumors (Yan et al., 2024; Shehata, 2024). In our paper, we propose a specialized prompt engineering framework, as illustrated in Figure 2. As depicted in Step 1 of Figure 2, the input prompts guide the LLM to leverage its internal knowledge and predefined rules to briefly analyze the root post's background and evaluate its potential as a rumor. However, LLMs may apply inconsistent standards across multiple evaluations of the same news item, which can significantly compromise accuracy (Huang, 2025; Mohanty, 2025). To address this, we prompt the model to thoroughly query and compile detailed background information on the event when processing the root node's original news post, storing it in a persistent knowledge base. In Step 2 of Figure 2, subsequent subchain evaluations strictly adhere to this knowledge base alongside the prompt's criteria, ensuring consistent and reliable assessments throughout the process. This prompting strategy enhances the LLM's reliability in rumor detection, which is critical for combating misinformation on social platforms (Ziari and Charkari, 2025). Figure 3 presents an example illustrating how the LLM evaluates rumor probabilities across subchains in a propagation tree, using breaking news coverage of the Airbus A320 Germanwings crash. The root input includes details like a gradual descent observed on flight radar over the final 10 minutes, linked to a source, yielding an initial LLM-assigned rumor probability of 0.3 which indicates a moderate uncertainty in early reporting. This flows into subsequent subchains: one appends a query about the specific flight number (4U9525), elevating the probability to 0.7 due to potential inconsistencies or unverified specifics; another adds observations of lost altitude with constant speed and a supporting link, resulting in a probability of 0.6, reflecting partial alignment with facts but lingering ambiguities. This sequential evaluation demonstrates the model's capability to process evolving information threads, dynamically adjusting probabilities based on accumulated context while maintaining consistency through the established knowledge base.

To address extreme scenarios where the LLM-augmented graph contains notably few virtual edges, which may result from limitations in the model's knowledge base or reasoning capabilities and could significantly impair prediction performance, we introduce a mitigation strategy. This approach aims to diminish the undue influence of the LLM in such outliers while leveraging graph learning methods to correct biases (Li et al., 2025; Hang,

2025). Specifically, if the number of virtual edges falls below a predefined minimum threshold (indicating sparse connections), we retain connections to some child nodes ranked top by the rumor probability assigned by LLM, ensuring a baseline level of virtual edges for graph propagation to refine and rectify potential LLM misjudgments. Formally, for child nodes $\{v_1, v_2, \ldots, v_n\}$ with corresponding rumor probabilities $P = \{p_1, p_2, \ldots, p_n\}$, the original method adds a virtual edge from the virtual node to v_i if $p_i > \theta$. However, if the resulting edge count $|E_v| < \lceil \gamma n \rceil$, we sort P in descending order and connect the virtual node to the top $k = \lceil \gamma n \rceil$ child nodes, even if some $p_i \leq \theta$. This can be expressed as:

$$E_v = \begin{cases} \{(v_{\text{virtual}}, v_i) \mid p_i > \theta\} & \text{if } |E_v| \geq \lceil \gamma n \rceil \\ \{(v_{\text{virtual}}, v_{\sigma(j)}) \mid 1 \leq j \leq k\} & \text{otherwise} \end{cases}$$

Here, $v_{\mathcal{D}_{\text{test}} virtual}$ denotes the virtual node, σ is the permutation sorting indices by $p_{\sigma(1)} \geq p_{\sigma(2)} \geq \cdots \geq p_{\sigma(n)}$, and $k = \lceil \gamma n \rceil$ ensures at least a proportional subset of edges. In our experiments, we found that setting the parameter γ to around 0.2 yields stable and relatively optimal performance. We used Youden's J statistic to determine the optimal classification threshold θ for edge classification tasks to optimize the performance of the classifier from the ROC curve. Experiments have shown that the threshold determined in this way is much better than the conventional 0.5 threshold. This formula ensures robust graph connectivity, enabling downstream GNNs to aggregate contextual signals, mitigate LLM-induced bias, and enhance the system's resilience and predictive accuracy for rumor detection tasks.

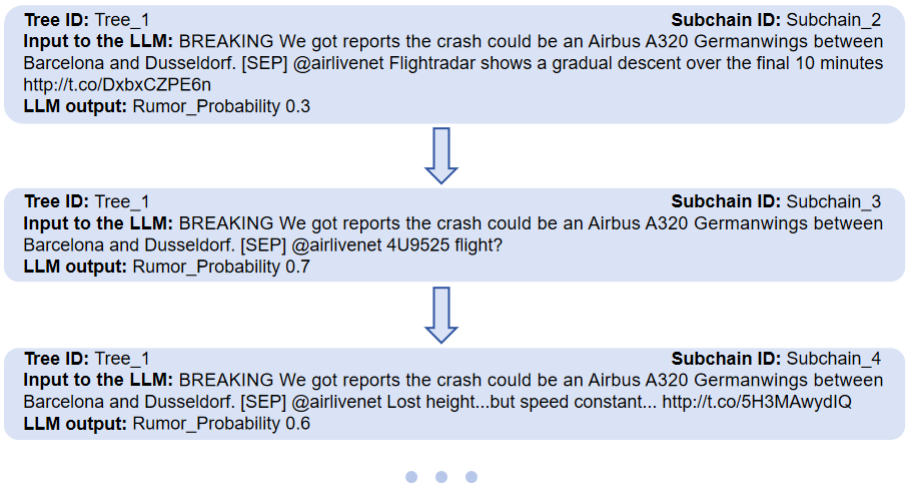


Figure 3. LLM processes information flow

To demonstrate the enhancing effect of our designed framework on graph learning methods, we will use Bidirectional Graph Attention Network (Bi-GAT) as an example to show how it is implemented in practice. Bi-GAT is a specialized model designed for processing graph-structured data, particularly for rumor detection in social media networks. The model uses BERT embeddings as input features and integrates Graph Attention Network (GAT) layers to learn node representations.

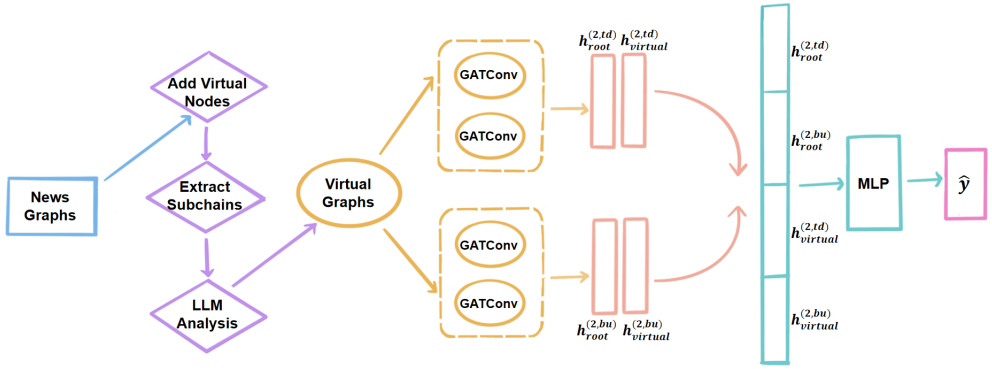


Figure 4. LLM-enhanced Bi-GAT model

Graph Attention Convolution (GATConv) The GATConv function implements the graph attention mechanism proposed by Veličković et al. (2018), which computes node representations by attending over neighboring nodes

with learned attention coefficients. For a graph $\mathbb{G}' = (\mathbb{V}', \mathbb{E}')$ with node features $\boldsymbol{x}_i \in \mathbb{R}^{F_{\text{in}}}$ for node $v_i \in \mathbb{V}'$, and edge set $\mathbb{E}'$, the `GATConv` layer computes attention scores for its neighbors $v_j \in \mathcal{N}(i)$, where $\mathcal{N}(i) = \{v_j \mid (v_i, v_j) \in \mathbb{E}'\}$ is the set of neighboring nodes. The attention score e_{ij} between nodes v_i and v_j is calculated as:

$$e_{ij} = \text{LeakyReLU}\left(\mathbf{a}^T[\mathbf{W}_q\mathbf{x}_i \parallel \mathbf{W}_k\mathbf{x}_j]\right),$$

where $\mathbf{W}_q$, $\mathbf{W}_k$, $\mathbf{W}_v \in \mathbb{R}^{F' \times F_{\text{in}}}$ are weight matrices that transform the input features to the hidden dimension. $\mathbf{a} \in \mathbb{R}^{2F'}$ is the attention parameter vector. $[\mathbf{W}_q\mathbf{x}_i \parallel \mathbf{W}_k\mathbf{x}_j] \in \mathbb{R}^{2F'}$ is the concatenation of the transformed features of nodes v_i and v_j. Then the attention coefficients α_{ij} are normalized across neighbors using the softmax function and use the attention coefficient to weight the neighbor value vector to update the feature of node i:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik})}, \quad \mathbf{h}_i = \sum_{j \in \mathcal{N}(i)} \alpha_{ij}\mathbf{W}_v x_j.$$

To stabilize and enhance the attention mechanism, multiple attention heads are employed. For head $h = 1, \ldots, H$, each head has its own weight matrix $\mathbf{W}_v^{(h)} \in \mathbb{R}^{F' \times F_{\text{in}}}$ and attention vector $\mathbf{a}^{(h)} \in \mathbb{R}^{2F'}$. The output of each head is:

$$\mathbf{h}_i^{(h)} = \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(h)}\mathbf{W}_v^{(h)}\mathbf{x}_j, \quad \mathbf{h}_i = \|_{h=1}^{H} \mathbf{h}_i^{(h)} \in \mathbb{R}^{HF'},$$

where $\alpha_{ij}^{(h)}$ is the attention coefficient for head h. $\boldsymbol{h}_i$ is the final output for node v_i concatenates the head outputs. The feature dimension after concatenation is HF'. Compared with averaging, this approach significantly increases the dimension of the output feature, which is equivalent to increasing the number of parameters and representation ability of the model.

Top-Down GAT (TD-GAT) Simulate the information transmission from the "high-level" nodes to the "low-level" nodes of the graph, which is suitable for capturing hierarchical or causal relationships.

- **First GATConv Layer**: $\mathbf{h}_i^{(1,\text{td})} = \text{ReLU}\left(\|_{h=1}^{H} \sum_{j \in \mathcal{N}_{out}(i)} \alpha_{ij}^{(h)}\mathbf{W}_{\text{td1}}^{(h)}\mathbf{x}_j\right)$, where $\mathcal{N}_{out}(i)$ is the out-neighbors of node i and $\mathbf{h}_i^{(1,\text{td})} \in \mathbb{R}^{HF'}$.
- **Second GATConv Layer**: $\mathbf{h}_i^{(2,\text{td})} = \frac{1}{K}\sum_{k=1}^{K} \sum_{j \in \mathcal{N}_{out}(i)} \alpha_{ij}\mathbf{W}_{\text{td2}}[\mathbf{h}_j^{(1,\text{td})} \parallel \mathbf{x}_{\text{root}}]$, where x_{root} is the feature of root node and $\mathbf{h}_i^{(2,\text{td})} \in \mathbb{R}^{F'}$.

Bottom-Up GAT (BU-GAT) Simulate the aggregation of features from low-level nodes (such as child nodes) to high-level nodes (such as root nodes) of the graph to better capture summary or global information.

- **First GATConv Layer**: $\mathbf{h}_i^{(1,\text{bu})} = \text{ReLU}\left(\|_{h=1}^{H} \sum_{j \in \mathcal{N}_{in}(i)} \alpha_{ij}^{(h)}\mathbf{W}_{\text{bu1}}^{(h)}\mathbf{x}_j\right)$, where $\mathcal{N}_{in}(i)$ is the in-neighbors of node i.
- **Second GATConv Layer**: $\mathbf{h}_i^{(2,\text{bu})} = \frac{1}{K}\sum_{k=1}^{K} \sum_{j \in \mathcal{N}_{in}(i)} \alpha_{ij}\mathbf{W}_{\text{bu2}}[\mathbf{h}_j^{(1,\text{bu})} \parallel \mathbf{x}_{\text{root}}]$.

Feature Fusion and Edge Classification

- **Feature Extraction**: Extract $\mathbf{h}_{\text{root}}^{(2,\text{td})}$, $\mathbf{h}_{\text{virtual}}^{(2,\text{td})}$, $\mathbf{h}_{\text{root}}^{(2,\text{bu})}$, and $\mathbf{h}_{\text{virtual}}^{(2,\text{bu})}$.
- **Fusion**: Concatenate into $\mathbf{c} = [\mathbf{h}_{\text{root}}^{(2,\text{td})}, \mathbf{h}_{\text{root}}^{(2,\text{bu})}, \mathbf{h}_{\text{virtual}}^{(2,\text{td})}, \mathbf{h}_{\text{virtual}}^{(2,\text{bu})}] \in \mathbb{R}^{4F'}$.
- **Classification**: $\mathbf{z} = \mathbf{W}_{\text{edge}}\mathbf{c} + \mathbf{b}_{\text{edge}}$, $\hat{y}_{\text{edge}} = \sigma(\boldsymbol{z})$, where $\mathbf{W}_{\text{edge}} \in \mathbb{R}^{2 \times 4F'}$, $\mathbf{b}_{\text{edge}} \in \mathbb{R}^2$, and σ is the sigmoid function.

Model and Training We use the binary cross-entropy function. For a batch size of N, the loss function is defined as

$$\textbf{Loss} = -\frac{1}{N}\sum_{i=1}^{N} \omega_i[y_i \log(\sigma(z_i)) + (1 - y_i)\log(1 - \sigma(z_i))]$$

where z_i is the positive class logit output by the model (i.e. the logit score of the virtual node and the root node with an edge). y_i is the true label of the source news, and ω_i is the weight determined by y_i's category and pos_weight. pos_weight is set to the negative-positive class ratio, reducing positive loss weight and increasing negative loss weight to prioritize negative class predictions and counteract class imbalance. Parameters are trained using the Adam optimizer with backpropagation to optimize all components of the Bi-GAT model. We set the training set, validation set, and test set in a ratio of 7:1:2. The hyperparameters were set as follows: learning rate

= 0.00005, weight decay = 1e-3, dropout rate = 0.3, and a maximum of 150 training epochs. An early stopping mechanism was used; during training, the F1 score on the validation set is continuously monitored, and training is automatically stopped if the score failed to exceed the historical best value for 20 consecutive epochs, thus preventing model overfitting.

5 EXPERIMENTS

5.1 DATASETS AND LLMS

We experiment on five news events from the PHEME dataset: Charlie Hebdo shooting (charlie Hebdo); Killing of Michael Brown (Ferguson); Germanwings Flight 9525 (Germanwings crash); 2014 shootings at Parliament Hill, Ottawa (Ottawa Shooting); Lindt Cafe siege (Sydney Siege); and Weibo dataset (Ma et al., 2016), as shown in Table 1. After data preprocessing, some invalid data are removed, so the actual number of news will be slightly less than the number in the dataset. This experiment uses the DeepSeek-V3 released by DeepSeek as the base model. All interactions with the model are performed programmatically through the official DeepSeek-Chat API.

Table 1: Statistics of PHEME and Weibo Datasets

News Event	Non-Rumor Count	Rumor Count	Total
Charlie Hebdo	1621	458	2079
Ferguson	859	284	1143
Germanwings Crash	231	238	469
Ottawa Shooting	420	470	890
Sydney Siege	699	522	1221
Weibo	2313	2351	4664

5.2 EVALUATION METRICS

Table 2 presents the fundamental metrics for assessing classification model performance: Accuracy, Precision, Recall, and F1 Score.

Table 2: Classification Metrics, Terms, and Formulas

Metric	Formula	Term	Definition
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$	TP	True Positives (correctly predicted positives)
Precision	$\frac{TP}{TP + FP}$	TN	True Negatives (correctly predicted negatives)
Recall	$\frac{TP}{TP + FN}$	FP	False Positives (incorrectly predicted as positive)
F1 Score	$2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$	FN	False Negatives (incorrectly predicted as negative)

We named our proposed LLM-Virtual Node framework as 'LLM-VN' and to demonstrate the superior performance and effectiveness of our model, we have implemented several existing methods as comparative baselines.

Bi-GCN (Bian et al., 2020): A bidirectional GCN that captures both top-down and bottom-up structural dependencies in tree-structured data, enhancing representation learning.

RvNN (Recursive Neural Network) (Socher et al., 2011): Processes hierarchical structures (e.g., parse trees) by recursively composing child node representations into parent nodes, suitable for syntax-aware tasks.

GAT (Veličković et al., 2018): A graph network using attention mechanisms to dynamically learn neighbor weights, particularly effective for handling heterogeneous graph structures such as social networks.

HD-TRANs (Ma and Gao, 2020): Combining Transformer with graph networks, it models temporal dependencies through multi-head attention and is suitable for dynamic graphs (such as user behavior prediction).

LINE (Tang et al., 2015): A foundational model for large-scale static network embedding, often used as a feature extraction module combined with models like GCN and LSTM in complex tasks.

DDGCN (Korban and Li, 2020): A dynamic directed GCN for action recognition that models spatiotemporal features via dynamic convolution sampling, weight allocation, and directed spatiotemporal extraction.

Graphsage (Hamilton et al., 2017):An inductive framework for generating node embeddings by sampling and aggregating features from local neighborhoods.

GSMA (Ma et al., 2024a): A graph sampling and aggregation model with attention and modulated position encoding for rumor detection, enhancing GraphSAGE to capture structural, positional, and sentiment features.

The 'LLM-VN' enhanced version of the above baseline approaches are denoted as '**LLM-VN+Baseline**'. We have also compare our approach to simple model average where the final probability of rumors is a weighted average of the probability from LLM and the probability based on the baseline method. The optimal weight is tuned via line search. We denote this as '**LLM+Baseline**'.

5.3 Results and analysis

As shown in Table 3, based on the PHEME dataset, LLMs exhibit poor performance in identifying rumors when they are not provided with carefully designed prompts. The Accuracy (R) is computed as the ratio of cases where the LLM correctly predicts a "rumor" (matching the true "rumor" label) to the total number of cases with a true "rumor" label. Accuracy (N) is calculated in a similar way for "non-rumor" cases. This confirms our previous observation that large language models inherently have biases and limitations in their ability to identify rumors.

Table 3: Performance of LLMs under basic prompts on PHEME

	Charlie Hebdo	Ferguson	Germanwings crash	Ottawa shooting	Sydney Siege
Accuracy (R)	15.54%	41.55%	23.11%	16.81%	33.14%
Accuracy (N)	83.20%	89.76%	91.77%	87.62%	78.25%

Next, we will present the performance of our proposed framework 'LLM-VN' combined with Bi-GAT on the PHEME dataset as shown below:

Table 4: Performance of LLM-VN+Bi-GAT on PHEME

News Event	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**AUC**
Charlie Hebdo	0.9229	0.9425	0.9669	0.9545	0.9619
Ferguson	0.8595	0.8182	0.8919	0.8534	0.9026
Germanwings Crash	0.8936	0.9136	0.9610	0.9367	0.8273
Ottawa Shooting	0.8764	0.8830	0.9869	0.9321	0.8455
Sydney Siege	0.8286	0.8663	0.9050	0.8852	0.8577

From Table 4, the model achieves strong performance on most news events, with accuracy often exceeding 85%, along with high precision, recall, and F1 scores. Table 5 provides a comprehensive comparison of graph learning methods for rumor detection on the Weibo and PHEME datasets. It evaluates several baseline models in three setups: the original baseline, LLM+Baseline, and LLM-VN+Baseline. From table 5, we can see that the LLM-VN-enhanced methods demonstrate significant advantages in rumor detection tasks on both the PHEME and Weibo datasets. Compared to the original baseline method and the model average approaches (LLM+Baseline), our framework shows substantial improvements in key metrics such as accuracy, precision, recall, and F1 score. Especially on the PHEME dataset, methods like LLM-VN+DDGCN exhibit particularly outstanding performance in accuracy and F1 score. On the Weibo dataset, methods such as LLM-VN+Bi-GCN and LLM-VN+GAT also outperform the baseline, particularly in terms of precision and recall for the non-rumor (N) category, demonstrating higher classification stability. In contrast, the LLM+Baseline method, which uses a weighted average of the LLM-processed news content and the baseline prediction, improves performance to some extent, but the improvement is limited. Its performance is less robust than the proposed LLM-VN enhanced approach, especially in cases of complex propagation structures or imbalanced data distributions. Overall, the LLM-VN framework significantly enhances the model's generalization ability and classification accuracy in rumor detection tasks by more effectively integrating the semantic understanding capability of LLMs with the structured information processing capability of graph learning models, particularly when dealing with complex event propagation patterns in social media.

6 Conclusion and Discussion

We propose a general framework to enhance the performance of graph learning methods for rumor prediction by leveraging LLMs, using DeepSeek and Bi-GAT as examples to demonstrate how the framework operates. By employing LLMs to analyze subchains and assign rumor probabilities, we augment the graph with a virtual "is Rumor" node. Intuitively, when LLMs predict high rumor probabilities across many subchains, the virtual node shares more neighbors with the root node, leading to convergent embeddings during bidirectional GNN propagation. This enhances the likelihood of link prediction between the root and virtual nodes, effectively classifying

Table 5: Comparison of Baseline, LLM-Enhanced, and LLM-VN-Enhanced Methods on PHEME and Weibo Datasets for Rumor Detection

	PHEME Dataset					Weibo Dataset				
Method	**Accu.**	**Class**	**Prec.**	**Rec.**	**F1**	**Accu.**	**Class**	**Prec.**	**Rec.**	**F1**
Bi-GCN	0.824	R	0.753	0.734	0.741	0.963	R	0.948	0.946	0.947
		N	0.861	0.872	0.865		N	0.970	0.972	0.971
LLM+Bi-GCN	0.830	R	0.758	0.744	0.750	0.971	R	0.952	0.949	0.950
		N	0.862	0.850	0.855		N	0.970	0.972	0.971
LLM-VN+Bi-GCN	**0.842**	R	0.772	0.752	0.761	**0.988**	R	0.976	0.978	0.976
		N	0.872	0.890	0.884		N	0.992	0.991	0.992
RVNN	0.763	R	0.689	0.587	0.631	0.763	R	0.689	0.587	0.631
		N	0.796	0.858	0.825		N	0.796	0.858	0.825
LLM+RVNN	0.783	R	0.777	0.754	0.761	0.775	R	0.692	0.595	0.638
		N	0.811	0.802	0.808		N	0.793	0.800	0.796
LLM-VN+RVNN	**0.807**	R	0.773	0.696	0.784	**0.781**	R	0.773	0.796	0.784
		N	0.835	0.896	0.842		N	0.791	0.766	0.778
Graphsage	0.842	R	0.772	0.820	0.795	0.963	R	0.956	0.953	0.954
		N	0.876	0.878	0.877		N	0.972	0.975	0.973
GSMA	0.848	R	0.834	0.823	0.840	0.974	R	0.967	0.953	0.960
		N	0.856	0.851	0.860		N	0.973	0.982	0.977
LLM-VN+Graphsage	**0.866**	R	0.822	0.825	0.820	**0.981**	R	0.978	0.933	0.955
		N	0.887	0.879	0.892		N	0.972	0.992	0.982
GAT	0.811	R	0.733	0.541	0.405	0.947	R	0.939	0.936	0.938
		N	0.877	0.798	0.833		N	0.961	0.943	0.962
LLM+GAT	0.823	R	0.799	0.796	0.804	0.960	R	0.944	0.939	0.941
		N	0.826	0.820	0.828		N	0.969	0.953	0.961
LLM-VN+GAT	**0.847**	R	0.823	0.860	0.800	**0.982**	R	0.977	0.972	0.981
		N	0.868	0.850	0.870		N	0.987	0.985	0.989
HD-TRANs	0.766	R	0.656	0.697	0.676	0.974	R	0.957	0.946	0.952
		N	0.783	0.755	0.768		N	0.979	0.978	0.979
LLM+HD-TRANs	0.779	R	0.767	0.766	0.780	0.974	R	0.957	0.946	0.952
		N	0.784	0.782	0.777		N	0.979	0.978	0.979
LLM-VN+HD-TRANs	**0.796**	R	0.696	0.737	0.716	**0.991**	R	0.963	0.960	0.800
		N	0.811	0.792	0.802		N	0.981	0.984	0.989
LINE	0.744	R	0.732	0.730	0.733	0.790	R	0.763	0.771	0.760
		N	0.749	0.750	0.749		N	0.802	0.811	0.796
LLM+LINE	0.759	R	0.753	0.750	0.755	0.803	R	0.768	0.775	0.763
		N	0.760	0.758	0.763		N	0.804	0.806	0.803
LLM-VN+LINE	**0.786**	R	0.747	0.745	0.746	**0.811**	R	0.802	0.804	0.800
		N	0.794	0.782	0.797		N	0.820	0.816	0.823
DDGCN	0.855	R	0.877	0.763	0.816	0.948	R	0.941	0.933	0.937
		N	0.831	0.892	0.860		N	0.965	0.970	0.967
LLM+DDGCN	0.860	R	0.855	0.852	0.856	0.954	R	0.950	0.949	0.952
		N	0.866	0.861	0.867		N	0.959	0.955	0.963
LLM-VN+DDGCN	**0.876**	R	0.858	0.832	0.845	**0.984**	R	0.979	0.978	0.982
		N	0.861	0.867	0.864		N	0.988	0.985	0.990

the source news as a rumor. Notably, the LLM and GNN-based linkage prediction can be flexibly substituted with alternative methods. Our numerical analysis demonstrates that our proposed framework significantly enhances the performance of existing GNN approaches by integrating LLMs, thereby improving rumor detection accuracy. We anticipate that performance could be further enhanced by leveraging more powerful LLMs, fine-tuned models, or advanced graph learning methods. The virtual node idea can also be applied to other tasks, transforming node classification tasks into link prediction problems, and effectively utilizing the strong language understanding and logical reasoning capabilities of large models.

7 Acknowledgment

This work received support from the Hong Kong RGC/GRF grant (15302924).

REFERENCES

Bian, T., Xiao, X., Xu, T., Zhao, P., Huang, W., Rong, Y., & Huang, J. (2020). Rumor Detection on Social Media with Bi-Directional Graph Convolutional Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, No. 1, pp. 549-556). AAAI Press.

Cheung, T. H., & Lam, K. M. (2023). FactLLaMA: Optimizing Instruction-Following Language Models with External Knowledge for Automated Fact-Checking. In *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)* (pp. 846-853). IEEE.

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems 30* (Vol. 30, pp. 1024-1034). Curran Associates.

Hang, C. H., Yu, P. D., & Tan, C. W. (2025). TrumorGPT: Graph-Based Retrieval-Augmented Large Language Model for Fact-Checking. *arXiv:2505.07891*.

Huang, T., Yi, J., Yu, P., & Xu, X. (2025). Unmasking Digital Falsehoods: A Comparative Analysis of LLM-Based Misinformation Detection Strategies. *arXiv:2503.00724*.

Korban, M. & Li, X. (2020). DDGCN: A Dynamic Directed Graph Convolutional Network for Action Recognition. In *European Conference on Computer Vision* (pp. 761-776). Springer.

Kumar, S. & Carley, K. M. (2019). Tree LSTMs with Convolution Units to Predict Stance and Rumor Veracity in Social Media Conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 5047-5058). Association for Computational Linguistics.

Lai, J., Yang, X., Luo, W., Zhou, L., Li, L., Wang, Y., & Shi, X. (2024). RumorLLM: A Rumor Large Language Model-Based Fake-News-Detection Data-Augmentation Approach. *Applied Sciences*, 14(8):3532. doi:10.3390/app14083532.

Li, H., Qian, D., Chen, J., Su, H., Zhou, Y., Ai, Q., Ye, Z., & Liu, Y. (2024). LLMs-as-Judges: A Comprehensive Survey on LLM-based Evaluation Frameworks. *arXiv:2412.05579*

Li, G., Hu, D., Liu, Z., Zhang, X., & Lyu, H. (2025). Semantic Reshuffling with LLM and Heterogeneous Graph Auto-Encoder for Enhanced Rumor Detection. In *Proceedings of the 31st International Conference on Computational Linguistics* (pp. 8557-8572). Association for Computational Linguistics.

Liu, T., Cai, Q., Xu, C., Hong, B., Ni, F., Qiao, Y., & Yang, T. (2024). Rumor Detection with A Novel Graph Neural Network Approach. *Academic Journal of Science and Technology*, 10(1):1-6.

Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B. J., Wong, K. F., & Cha, M. (2016). Detecting Rumors from Microblogs with Recurrent Neural Networks. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI 2016)* (pp. 3818-3824). AAAI Press.

Ma, J., Gao, W., & Wong, K. F. (2017). Detect Rumors in Microblog Posts Using Propagation Structure via Kernel Learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 708-717). Association for Computational Linguistics.

Ma, J., & Gao W. (2020). Debunking Rumors on Twitter with Tree Transformer. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)* (pp. 5455-5466). Association for Computational Linguistics.

Ma, M., Zhang, C., Li, Y., Chen, J., & Wang, X. (2024). Rumor Detection Model with Weighted GraphSAGE Focusing on Node Location. *Scientific Reports*, 14(1):27127. doi:10.1038/s41598-024-76738-7.

Ma, X., Zhang, Y., Ding, K., Yang, J., Wu, J., & Fan, H. (2024). On Fake News Detection with LLM Enhanced Semantics Mining. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing* (pp. 508-521). Association for Computational Linguistics.

Mohanty, S. (2025). Fine-Grained Bias Detection in LLM: Enhancing Detection Mechanisms for Nuanced Biases. *arXiv:2503.06054*.

Shehata, D., Cohen, R., &Clarke, C. (2024). Rumour Evaluation with Very Large Language Models. *arXiv:2404.16859*.

Socher, R., Lin, C. C., Ng, A. Y., & Manning, C. D. (2011). Parsing Natural Scenes and Natural Language with Recursive Neural Networks. In *Proceedings of the 28th International Conference on Machine Learning* (pp. 129-136). Omnipress.

Sun, T., Qian, Z., Dong, S., Li, P., & Zhu, Q. (2022). Rumor Detection on Social Media with Graph Adversarial Contrastive Learning. In *Proceedings of the ACM Web Conference 2022* (pp. 2789-2797). ACM.

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., & Mei, Q. (2015). LINE: Large-scale Information Network Embedding. In *Proceedings of the 24th International Conference on World Wide Web (WWW)* (pp. 1067-1077). ACM.

Tian, Z., Huang, J., He, Z., Huang, Z., Lu, M., Qiao, L., Mei, S., Wang, Y., & Li, D. (2025). LLM-based Rumor Detection via Influence Guided Sample Selection and Game-based Perspective Analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics* (pp. 1378-1391). Association for Computational Linguistics.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph Attention Networks. In *International Conference on Learning Representations* . OpenReview.net.

Wang, B., Ma, J., Lin, H., Yang, Z., Yang, R., Tian, Y., & Chang, Y. (2024). Explainable Fake News Detection With Large Language Model via Defense Among Competing Wisdom. In *Proceedings of the ACM Web Conference 2024* (pp. 2452-2463). ACM.

Wu, Z., Pi, D., Chen, J., Xie, M., & Cao, J. (2020). Rumor Detection Based on Propagation Graph Neural Network with Attention Mechanism. *Expert Systems with Applications*, 158:113595. doi:10.1016/j.eswa.2020.113595.

Yan, Y., Zheng, P., & Wang, Y. (2024). Enhancing Large Language Model Capabilities for Rumor Detection with Knowledge-Powered Prompting. *Engineering Applications of Artificial Intelligence*, 133:108259. doi:10.1016/j.engappai.2024.108259.

Zhou, C., Liu, P., Xu, P., Iyer, S., Sun, J., Mao, Y., Ma, X., Efrat, A., Yu, P., Yu, L., Zhang, S., Ghosh, G., Lewis, M., Zettlemoyer, L., & Levy, O. (2023). LIMA: Less Is More for Alignment. In *Advances in Neural Information Processing Systems 36*. Curran Associates.

Ziari, M., & Moghadam Charkari, N. (2025). Rumor Detection and Propagation on Social Networks: A Survey. *Expert Systems with Applications*, 263:128798. doi:10.1016/j.eswa.2025.128798.

Zubiaga, A., Aker, A., Bontcheva, K., Liakata, M., & Procter, R. (2018). Detection and Resolution of Rumours in Social Media: A Survey. *ACM Computing Surveys*, 51(2):1-36. doi:10.1145/3161603.