Leveraging Content and Structural Dynamics of Discourse for Rumor Verification

Gibson Nkhata, Susan Gauch
Department of Electrical Engineering and Computer Science
University of Arkansas
Fayetteville, AR 72701, USA
e-mails: {gnkhata, sgauch}@uark.edu

***Abstract*—The rapid spread of misinformation on social media platforms has heightened the need for effective rumor verification models. Traditional approaches to rumor verification primarily rely on textual content and transformer-based embeddings, but they often fail to incorporate conversational dynamics and stance evolution, limiting their effectiveness, necessitating the need for robust and interpretable rumor verification systems. This study introduces a novel framework that integrates semantic content, structural dynamics, and stance distribution features into a unified model for predicting rumor veracity. By employing hierarchical sequencing strategies, including Breadth-First Search, Depth-First Search, and temporal traversal, for structural dynamics, this work effectively captures both global and local relationships within conversational threads. This work further leverages an attention-based encoder to aggregate content embeddings, stance distributions, and reply-level features, overcoming limitations of sequence truncation and underutilized conversational structures found in existing methods. Experimental results on benchmark datasets, including SemEval-2017, RumorEval-2019, and PHEME, demonstrate that our approach achieves state-of-the-art performance, significantly improving Macro-F1 scores and accuracy over competing models. Ablation studies confirm the critical contributions of hierarchical encoding, stance aggregation, and attention mechanisms to the model's success. Thus, this work sets a new standard for efficient, interpretable, and scalable rumor verification, offering promising directions for mitigating misinformation on social media platforms.**

***Index Terms*—Rumor verification; stance-conditioned modeling; social media misinformation; early detection; embedding aggregation.**

I. Introduction

This paper extends our previous work, stance-conditioned modeling for rumor verification [1]. Specifically, the prior work focused on integrating source and reply post embeddings from Bidirectional Encoder Representations from Transformers (BERT) [2] with stance labels, encoded through Bidirectional Long Short Term Memory (BiLSTM) [3], for rumor classification.

Social media platforms play an essential role in information dissemination, news sharing, and communication in the modern Internet era [4][5], yet they also foster the proliferation of misinformation, which poses a significant challenge to societal trust, with far-reaching implications for public health, politics, and safety [6][7]. Therefore, automated rumor verification systems have emerged as crucial tools to combat misinformation, relying on machine learning and natural language processing (NLP) techniques [8][9]. Despite progress, existing systems face key limitations in effectively utilizing the entirety of conversational threads, particularly in capturing both semantic and structural dynamics.

For example, [8] segment lengthy threads into shorter subthreads and use BERT to individually encode each subthread. A global model layer is then applied to integrate the representations of all subthreads, with constraints on the maximum number of posts per thread and the maximum number of subthreads per thread. While existing work predominantly utilize pre-trained Large Language Models (LLMs) like BERT, these methods face inherent limitations. One major challenge is the sequence length constraint of LLMs, often used to encode full conversational threads. LLMs truncate inputs or subdivide threads into smaller chunks, resulting in the omission of replies that contain critical semantic information. Moreover, structural information, such as the hierarchy of replies, stance distributions (e.g., Support, Deny, Query, Comment), and the temporal sequencing of posts' features, is either underutilized or entirely neglected in a discourse. Such limitations hinder the development of robust models that can comprehensively analyze discourse for accurate rumor veracity prediction.

Recent advances in rumor verification, mostly breaking event rumor detection, has shown that stance-separated models have improved veracity prediction by analyzing distinct stance categories during breaking news events [10]. Furthermore, multi-task frameworks integrating stance classification with rumor detection have shown improved performance by leveraging shared features across tasks [11]. Additionally, modern graph-based enhancements include stance distributions as node-level features in graph attention mechanisms, underscoring its importance in contextualizing relationships within threads [12].

We propose a novel approach that integrates content embeddings, stance aggregation, and structural information into a unified model for rumor verification. By integrating hierarchical and temporal sequencing of discourse into a unified framework, this research bridges the gap between content-based and structure-aware rumor verification. We use stance-based posts' aggregation and graph-based traversal techniques for post-level vector encoding to ensure a holistic representation of conversational threads, overcoming the limitations of sequence truncation and incomplete conversational thread

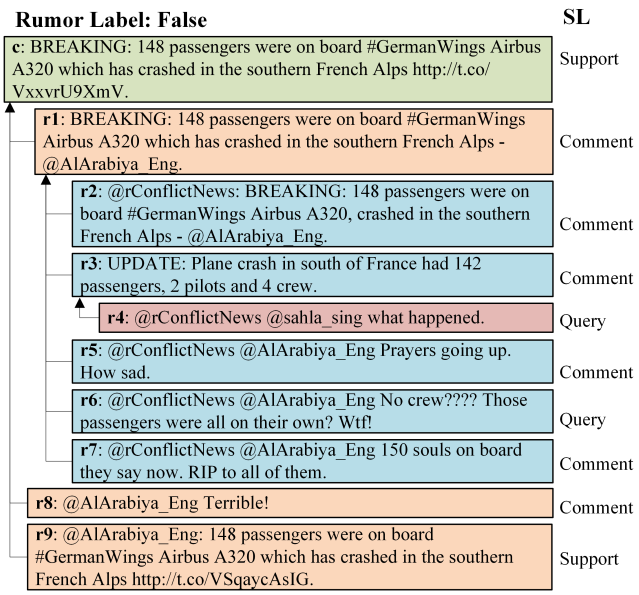

Fig. 1. A sample thread C with a false veracity label. SL stands for Stance Labels.

utilization. While methodologies such as GACN [13] and SAMGAT [12] focus on graph-based models, our approach departs from explicit graph neural networks and instead employs graph-inspired traversal strategies for sequencing. This provides a lightweight yet effective way to capture structural and temporal dynamics without the computational overhead of full graph-based models.

Furthermore, this study explores the task of early rumor detection, focusing on identifying and assessing the veracity of emerging rumors in real-time as they propagate online. By detecting rumors at an early stage, this approach aims to mitigate the rapid spread of misinformation, enabling timely interventions and fact-checking before false narratives gain widespread traction. Figure 1 presents a sample discourse, showcasing how stances evolve. Our work sets a new standard for efficient and interpretable systems that leverage discourse dynamics to combat misinformation.

The key contributions of this work are:

- **Rich Feature Representation**: Our framework integrates embeddings of a source post and replies grouped by stance type. Hierarchical reply levels and stance distributions are explicitly modeled as input features, enabling the model to learn from structural and temporal dynamics.
- **Hierarchical Sequencing of Discourse Trees**: This work models each conversation thread as a discourse tree and sequences posts using Breadth-First Search (BFS), Depth-First Search (DFS) traversals, and temporal sequencing, systematically organizing reply levels and robustly representing the hierarchical relationships within the discourse.
- **Enhanced Model Architecture**: We present an attention-based encoder that processes posts' embeddings, stance distributions, and reply-level vectors derived from graph traversal strategies.
- **Rigorous Experimental Validation**: The approach is evaluated on competitive and publicly available datasets, demonstrating significant improvements over state-of-the-art (SOTA) models in metrics such as Macro-F1 score and accuracy. Furthermore, we evaluate the model's ability to detect rumors at an early stage of the conversation, highlighting its real-world applicability for misinformation mitigation

The remainder of this paper unfolds as follows. Section II reviews the existing literature on rumor verification, and Section III delves into a comprehensive description of our approach. Section IV demonstrates experiments and provides a discussion of results. Finally, Section V concludes the paper.

II. Related Work

A rumor is defined as a widely circulated piece of information whose veracity is uncertain [14][15]. Rumours appear credible but lack immediate verification and often provokes skepticism, thus prompting users or automated systems to seek confirmation of its truthfulness. The field of rumor verification has witnessed significant advancements in recent years, driven by the need to combat the spread of misinformation on social media platforms. Most existing studies can be categorized into content-based approaches, structure-aware models, and methods integrating temporal and hierarchical features.

A. Content-Based Approaches

The early methods of rumor verification focused primarily on textual content using traditional machine learning and NLP techniques [16, 17, 18]. With the rise of deep learning, numerous models have been applied to the task of rumor verification. Early deep learning studies predominantly utilized Recurrent Neural Networks (RNNs), such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), to extract information from rumor conversations. These approaches typically arrange tweets chronologically to represent the entire conversation's features [19, 20, 21]. However, such methods face challenges in effectively capturing long-range dependencies within sequences and rely on sequential processing. Therefore, transformer-based pre-trained models like BERT and its variants became the backbone of many rumor verification systems [9]. Nevertheless, these models suffer from sequence length limitations, often truncating crucial replies or ignoring their hierarchical relationships within threads.

B. Structure-Aware Models

Several studies have incorporated discourse structures into their models, recognizing the importance of structural features [22, 23, 24, 25]. For instance, [22] employed tree-structured recursive neural networks to model hierarchical reply relationships in Twitter (rebranded as X) threads. More recently, graph-based approaches like SAMGAT [12] have introduced multilevel graph attention networks to integrate the semantic relationships between posts and their replies. In addition, a sequence graph network [26] has modeled conversations

as graphs and utilized a graph attention mechanism to process the interactions within them. While graph neural networks provide a powerful tool for capturing structural and semantic interactions, they often involve significant computational overhead and lack explicit focus on hierarchical sequencing.

C. Temporal and Hierarchical Features

Temporal dynamics and reply hierarchies have also been explored to enhance rumor verification systems. [27] highlighted the importance of temporal sequencing in understanding the progression of online discussions. Similarly, methods like DynamicGCN [28] use temporal information to model the propagation of rumors. Still, these approaches typically focus on global temporal patterns, overlooking the local reply-level interactions and structural nuances

To this end, existing rumor verification work often adopts flat sequence-based approaches or computationally intensive graph-based models, leaving key gaps in their ability to capture discourse dynamics holistically. Flat-sequence approaches fail to represent hierarchical and temporal relationships, while graph-based models focus on learning node and edge interactions but come with significant computational overhead. We further notice that many existing methods overlook the integration of nuanced structural features, such as reply hierarchies and stance distributions, alongside temporal patterns of replies. This research bridges these gaps by employing lightweight graph traversal strategies and temporal sequencing to sequence hierarchical relationships in discourse trees. Using these strategies, we create a solution that effectively encodes hierarchical, temporal, and semantic features without relying on computationally heavy graph-based neural networks, providing a scalable, interpretable, and comprehensive technique for capturing the interplay between content and structure within discourse.

III. Methodology

Our model consists of four main components: 1) **Post embedding representation**: BERT extracts contextual embeddings for the source and reply posts. 2) **Structural features extraction**: We encode hierarchical levels and stance distribution. 3) **An attention-based encoder**: The post embeddings and structural features are integrated. 4) **A decoder**: used for rumor prediction. Figure 2 illustrates our methodology. We begin by formally defining the problem of rumor verification, followed by a detailed explanation of the methodology.

A. Problem Definition

The dataset for rumor detection is defined as $E = \{e_1, e_2, \ldots, e_n\}$, where each e_i corresponds to a unique rumor event and n refers to the total number of rumor events. Each event e_i consists of (c, y, R, s), where c represents the claim, and y denotes the veracity label associated with c, such that $y \in \{\text{true, false, unverified}\}$ rumor. The responses in the discourse for claim c are arranged chronologically as $R(c) = \{r_1, r_2, \ldots, r_m\}$, where m indicates the number of posts r responding to a particular claim c. s encapsulates the structural information of the discourse tree. To this end, this task is formulated as a supervised classification problem, where the goal is to learn a function $f : E \rightarrow y$ that predicts the veracity label of each event e_i.

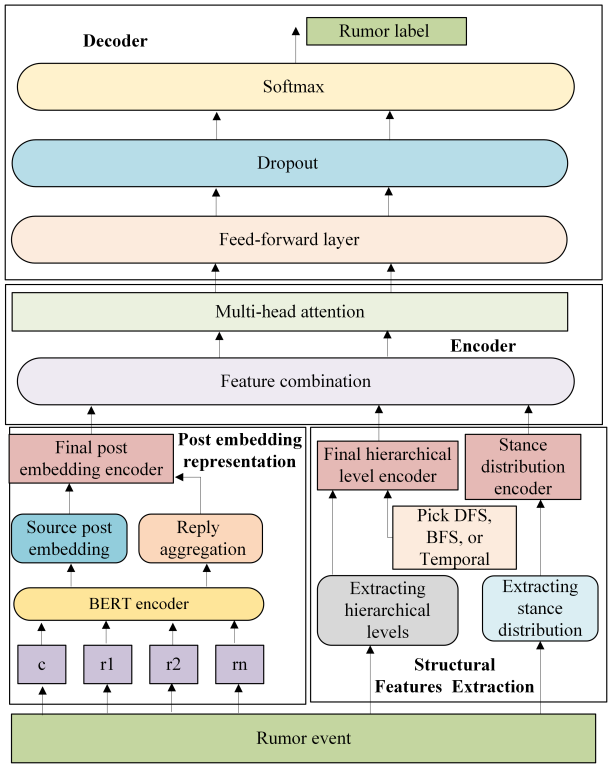

Fig. 2. The model framework.

B. Post embedding representation

Each post x_i (both c and r_i) in a conversational thread is firstly tokenized and passed through a pre-trained language model (e.g., BERT). This produces contextual embeddings for each token in the post. A pooling strategy is then applied to aggregate these token embeddings into a single vector representing the entire post as

$$\mathbf{e}_i = \text{BERT}(x_i) = \frac{1}{T}\sum_{t=1}^{T} h_t, \tag{1}$$

where h_t represents the hidden state at position t of a given post, and T is BERT input sequence length.

The stance of each reply plays a critical role in the discourse's overall meaning. Following prior work on stance aggregation [7][17][29], we categorize all replies in a thread stance labels (Support, Deny, Query, Comment). For each stance type, embeddings of all replies sharing that stance are aggregated to create a single vector representing the semantic contribution of that stance to the discourse. Let $V_{\text{stance}} \subseteq V$ be the set of nodes (replies) in the thread that share a particular stance. The aggregated embedding for this stance is:

$$\mathbf{e}_{\text{stance}} = \frac{1}{|V_{\text{stance}}|} \sum_{v_i \in V_{\text{stance}}} \mathbf{e}(v_i), \tag{2}$$

If no replies belong to a particular stance, a zero vector is used, following related practices in hierarchical discourse modeling [17].

After aggregating embeddings for all four stances, these vectors are concatenated with the embedding of the source post to create a composite feature vector for a rumor event:

$$f = [e_{src}, e_{support}, e_{deny}, e_{query}, e_{comment}] \tag{3}$$

In rumor verification tasks, aggregating embeddings efficiently and meaningfully is crucial to capturing both semantic content and the structural dynamics of conversations. Moreover, as illustrated in Figure 1, posts sharing the same stance tend to convey similar semantic information. Thus, we propose that aggregating these posts can provide a comprehensive representation of the semantic information within a conversational thread. Additionally, stance aggregation ensures dimensionality reduction, where aggregating embeddings by stance reduces computational overhead while preserving essential information [6], thereby overcoming the limitations of thread sequence truncation. Finally, aggregation enforces scalability, ensuring that the model handles long threads without exceeding memory constraints [17][30].

f from (3) is then concatenated with hierarchical and structural information to form the input features for the model.

C. Structural Features Extraction

Building on recent advances in discourse modeling [12][28], structural and temporal features are incorporated to enhance a composite representation. First, to capture the hierarchical relationships within conversational threads, we represent the discourse as a tree structure, where each node corresponds to a post, and edges represent reply relationships. Second, for traversal and sequencing, traversal techniques—BFS, DFS, and temporal sequencing—are applied to sequence replies and encode their hierarchical levels into a structured feature vector. Finally, we integrate stance distribution information.

Let the discourse tree $G = (V, E)$ represent a conversational thread. V is the set of nodes, where $v_i \in V$ corresponds to a post. E is the set of directed edges, where $e_{ij} \in E$ indicates that v_i is a reply to v_j. The root node v_r represents the source post c; the other nodes v_i are replies r_i.

1) Encoding hierarchical levels: The hierarchical level $\ell(vi)$ of each node v_i is defined recursively, $\ell(vi) = \ell(v_p)+1$, where v_p is the parent of v_i. Each node v_i is assigned a reply-level vector based on its position in the tree. The root node is by default promoted to level 1. For a tree with maximum depth D, the vector $h(v_i)$ is:

$$h(v_i) = [0, \ldots, 0, 1, 0, \ldots, 0], \tag{4}$$

where the $\ell(vi)$-th position is set to 1, and all others are 0; $h(v_i)$'s dimension is D. Therefore, we sequence $h(v_i)$ vectors by separately adopting the following three tree traversal strategies.

1) **Breadth-First Search:** BFS sequences nodes level by level, starting from the root node v_r. At each level, all nodes are processed before moving to the next level. This traversal preserves the hierarchy in a breadth-wise manner when applying the hierarchical level $\ell(v_i)$. For example, considering the discourse tree in Figure 1, the breadth-first order of the propagation structure would be $[c, r_1, r_8, r_9, r_2, r_3, r_5, r_6, r_7, r_4]$.
2) **Depth-First Search:** DFS explores as far as possible along each branch before backtracking. This traversal preserves the depth-first order of replies and captures long chains of interactions. This sequence highlights the reply chains within a thread. Referring again to Figure 1, the depth-first order of the propagation structure would be $[c, r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9]$.
3) **Temporal sequencing:** Reply timestamps are appended to capture the evolution of discussions over time [28]. To integrate the time dimension, each node v_i is assigned a timestamp $t(v_i)$, representing the post's creation time. The replies are then sorted by $t(vi)$ within each hierarchical level. Given two nodes (v_i, v_j),

$$\text{Order}(v_i, v_j) = \begin{cases} -1 & \text{if } t(v_i) < t(v_j), \\ 1 & \text{if } t(v_i) > t(v_j). \end{cases} \tag{5}$$

This ordering ensures that replies are chronologically processed while maintaining their hierarchical structure.

Whether BFS, DFS, or temporal sequencing is used, it results in a hierarchical vector $\mathbf{h}_{\text{input}}$.

2) Stance Distribution: We subsequently integrate a normalized vector representing the proportion of replies associated with each stance category. This design is inspired by prior work, such as SAMGAT [31], which employs graph-based aggregation to capture reply distribution patterns. The resulting feature serves as a global descriptor of the discourse's stance dynamics and is formally expressed as a normalized vector:

$$\mathbf{s} = \left[\frac{|V_{\text{support}}|}{|V|}, \frac{|V_{\text{deny}}|}{|V|}, \frac{|V_{\text{query}}|}{|V|}, \frac{|V_{\text{comment}}|}{|V|} \right], \tag{6}$$

where $|V_{stance}|$ is the count for each stance and $|V|$ is the total number of replies. This augmented feature representation adds three benefits to our rumor verification model. First, insight into credibility, since replies with negative stances (e.g., Deny) or critical engagement (e.g., Query) are often associated with false rumors, while supportive replies may reinforce credibility [29][12]. Next, cross-feature interaction, combined with hierarchical or temporal features, stance distribution amplifies the model's ability to learn patterns in rumor propagation [10]. Finally, signal amplification in sparse data, in threads with limited replies, stance distribution provides aggregate signals that are more robust than individual post features [11].

The final harmonic unified input feature combines the aforementioned components as:

$$F = [f, s, \mathbf{h}_{\text{input}}] \tag{7}$$

D. An attention-based Encoder

The encoder in this work is designed to process and integrate diverse features from conversational threads, including f, s, and $\mathbf{h}_{\text{input}}$. It employs specialized fully connected layers to encode each type of feature before combining them into a unified representation for subsequent processing by an attention layer.

1) Final Post Embedding Encoder: The input here is a concatenated vector of the source post embedding and stance-specific aggregated embeddings f from (3). A Fully-connected Feed-forward Layer (FFL) with ReLU activation and dropout regularization is used to process f as:

$$\mathbf{h}_{\text{post}} = \mathbf{ReLU}(\mathbf{W}_{\text{post}}\mathbf{f} + \mathbf{b}_{\text{post}}), \tag{8}$$

where $\mathbf{W}_{\text{post}}$ and $\mathbf{b}_{\text{post}}$ are learnable parameters. The output is a latent representation of stance embeddings with dimension H.

2) Stance Distribution Encoder: Likewise, here the input is a normalized stance distribution vector s from (6). We process s with an FFL, which transforms the structural vector into a latent representation:

$$\mathbf{h}_{\text{dist}} = \mathbf{ReLU}(\mathbf{W}_{\text{dist}}\mathbf{s} + \mathbf{b}_{\text{dist}}), \tag{9}$$

where $\mathbf{W}_{\text{dist}}$ and $\mathbf{b}_{\text{dist}}$ are trainable weights and $\mathbf{h}_{\text{dist}} \in \mathbb{R}^n$ is the encoded stance distribution output.

3) Final Hierarchical Level Encoder: This sub-encoder uses the structural information $\mathbf{h}_{\text{input}}$ encoded as one-hot or positional vectors. An FFL transforms the structural vector into a latent representation as:

$$\mathbf{h}_{\text{struct}} = \mathbf{ReLU}(\mathbf{W}_{\text{struct}}\mathbf{h}_{\text{input}} + \mathbf{b}_{\text{struct}}), \tag{10}$$

where $\mathbf{W}_{\text{struct}}$ and $\mathbf{b}_{\text{struct}}$ are learnable parameters ans $\mathbf{h}_{\text{struct}} \in \mathbb{R}^m$ encodes structural information.

4) Attention mechanism: Attention mechanisms enable models to focus on pertinent parts of input sequences dynamically, enhancing the capture of intricate dependencies and relationships [32, 33, 34]. We extend this aspect to model interactions among encoded features ($\mathbf{h}_{\text{post}}$, $\mathbf{h}_{\mathbf{dist}}$, $\mathbf{h}_{\text{struct}}$), leveraging a Multi-Head Attention (MHA) mechanism [35], which allows parallel attention computations to capture diverse relationships within the input feature space. Thus, it enhances feature fusion, by attending across multiple features, aiding the model to effectively integrate information from diverse sources, ensuring that stance semantics, hierarchical levels, and stance distributions are considered together for predicting rumor veracity. Hence, the three latent representations are concatenated along a sequence dimension to form the input for the MHA mechanism as:

$$\mathbf{H}_{\text{input}} = \mathbf{Concat}(\mathbf{h}_{\text{post}}, \mathbf{h}_{\text{dist}}, \mathbf{h}_{\text{struct}}), \tag{11}$$

where $\mathbf{H}_{\text{input}} \in \mathbb{R}^{n+m+H}$. For each attention head i, the scaled dot-product attention computes with ReLU activation function:

$$\text{head}_i = \text{ReLU}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i, \tag{12}$$

where Q_i, K_i, V_i are the query, key, and value matrices derived from input feature encodings through linear projections of $\mathbf{H}_{\text{input}}$ with the respective weight matrices as:

$$Q_i = \mathbf{H}_{\text{input}} W_i^Q, \quad K_i = \mathbf{H}_{\text{input}} W_i^K, \quad V_i = \mathbf{H}_{\text{input}} W_i^V \tag{13}$$

The outputs from all heads are concatenated and passed through a final linear transformation:

$$\text{MultiHead}(\mathbf{H}_{\text{input}}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_n) W^o, \tag{14}$$

where n represents number of attention heads and W^o is a weight matrix. The attended features from MHA are then aggregated using mean pooling to produce the final unified representation:

$$\mathbf{h}_{\text{final}} = \text{Dropout}\left(\frac{1}{n}\sum_{i=1}^{n} \text{MultiHead}(\mathbf{H}_{\text{input}})\right), \tag{15}$$

where $Dropout$ is used for regularization.

E. Decoder

The decoder aims to take the output ($\mathbf{h}_{\text{final}}$) from the MHA as input and makes the rumor veracity prediction. It consists of a single dense layer and a softmax activation function to compute class probabilities for rumor veracity prediction. The feed-forward layer is leveraged to map the high-dimensional attended representation $\mathbf{h}_{\text{final}}$ to the output space corresponding to the rumor classes (True, False, Unverified):

$$\mathbf{z} = \text{Dropout}\left(\mathbf{W}_{\text{out}}\mathbf{h}_{\text{final}} + \mathbf{b}_{\text{out}}\right), \tag{16}$$

where $\mathbf{W}_{\text{out}} \in \mathbb{R}^{C \times D}$ is a learnable weight matrix that maps the hidden representation $\mathbf{h}_{\text{final}}$ to the output classes *C*, while $\mathbf{b}_{\text{out}} \in \mathbb{R}^C$ is the bias term. In this case, *D* is the dimensionality of $\mathbf{h}_{\text{final}}$ and *C* = 3. The raw output $\mathbf{z}$ is passed through a softmax function to convert it into probabilities for each class:

$$\hat{y}_i = \text{softmax}(\mathbf{z_i}) = \frac{\exp(z_i)}{\sum_{j=1}^{C} \exp(z_j)}, \quad i = 1, \ldots, C, \tag{17}$$

where $\hat{y}_i$ is the predicted probability for class i.

F. Objective function

During training, the predicted probabilities $\hat{y}$ are compared with the ground truth labels using the cross-entropy loss function:

$$\mathcal{L} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i), \tag{18}$$

where y_i is the one-hot encoded ground truth label.

The objective function for the rumor verification task is designed to minimize the classification error $\mathcal{L}$. That is, it integrates the outputs from the attention mechanism, processed by the decoder, and compares the predictions with the ground truth labels. In detail, the objective function is expressed as

$$\mathcal{J}(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log \hat{y}_{i,c}, \tag{19}$$

where N is the total number of rumor events in a training batch.

IV. Experiments

This section assesses the performance of our model in comparison to SOTA baselines and conducts a comprehensive analysis to gain deeper insights into the model's effectiveness.

A. Datasets

Experiments are conducted on three widely used and publicly available challenging benchmark datasets: SemEval-2017 [14], RumorEval-2019 [36], and PHEME [37]. Among these, RumorEval-2019 and PHEME extend the SemEval-2017 task, which comprises 325 rumor-related events and 5,568 tweets collected from eight major breaking news events.

On the one hand, RumorEval-2019 extends SemEval-2017 by incorporating additional test data and new Reddit-based content while utilizing all SemEval-2017 rumor events for training. It consists of 446 rumor-related conversational threads and a total of 8,574 posts. The claims in both SemEval-2017 and RumorEval-2019 are annotated with three veracity labels: True, False, or Unverified. Each post within a thread is assigned a stance label: Support, Deny, Query, or Comment. On the other hand, PHEME enhances RumorEval-2017 by incorporating additional rumor events and data from nine major breaking news stories on Twitter. It contains 2,402 conversational threads and 105,354 tweets. Unlike RumorEval-2019, the additional data in PHEME is annotated solely with rumor veracity labels.

These datasets are widely recognized within the research community. Employing the same datasets as previous studies enables a fair and direct comparison between our approach and SOTA techniques, ensuring consistency and reliability in the experimental evaluation.

Tables I, II, and III provide detailed statistics of the datasets. In the tables, (F, T, U) represent (False, True, Unverified), (S, D, Q, C) represent (Support, Deny, Query, Comment), and NS-M stands for Non-Stance Comments, respectively.

B. Data Preprocessing

Alongside conventional data preprocessing techniques, like removing null entries, we implement hashtag processing and text normalization, adapting the approach outlined by [38]. We additionally substitute all hyperlinks in the text with \$url\$ and replace all *@user* mentions with \$mention\$, as these transformations demonstrated effectiveness in prior work [9].

C. Experimental setup

Our model utilizes the uncased BERT base [2] to generate word embeddings for both a claim c and replies R within a rumor event e_i. We also tested alternative pre-trained language models (PLMs), such as the Robustly Optimized BERT Approach (RoBERTa) [39] and the Generative Pre-trained Transformer (GPT) [40]. However, they yielded suboptimal performance compared to BERT and were therefore dropped in most of our experiments. During training, the model processes 16 rumor events per batch. The BERT tokenizer is configured with a maximum sequence length of 128. Optimization is performed using the Adam optimizer [41], with a learning rate of 0.0001. Other hyperparameters include a dropout probability of 0.35 and four attention heads. For encoding hierarchical levels and stance distributions, the embedding dimensions are set to [18, 19, 20] for [SemEval-2017, PHEME, RumorEval-2019], corresponding to the average thread lengths in these datasets. We also experimented with various dimensions to find optimal ones.

For SemEval-2017 and RumorEval-2019, we adhere to the standard train/validation/test split as defined in the original publications. Conversely, since PHEME does not provide an official dataset split, a conventional evaluation protocol is adopted, that follows a leave-one-out k-fold validation strategy, where each event is used as a test set in turn.

To address class imbalance, class weights are dynamically calculated based on label frequencies and incorporated into the cross-entropy loss function $\mathcal{L}$ during training. The model's performance is evaluated using the Macro-F1 score and accuracy metrics, with the best-performing model on the validation Macro-F1 score saved for final testing. All hyperparameters were meticulously fine-tuned using the development dataset. The reported results are aggregated from ten experimental runs. All experiments were conducted on two Quadro RTX 8000 GPUs, each equipped with 48 GB of VRAM, ensuring sufficient computational resources for our tasks.

As the PHEME dataset includes only partial stance annotations, we initially trained the model (omitting the stance-based embedding aggregation and stance distribution modules during this stage) using both the stance-labeled RumorEval-2019 and SemEval-2017 datasets. Given that these datasets exhibit a significant skew toward the *Comment* stance, we applied the SMOTE [42] oversampling technique to balance the stance distribution and improve the generalization of the model. The best-performing model from this training was then used to predict the stance labels for the PHEME dataset. On the same note, SMOTE was not used for rumor verification to ensure a fair comparison with baseline methods.

TABLE I
DETAILED STATISTICS OF RUMEVAL2017

Split	Rumor Statistics					Stance Distribution			
	#Threads	**AvgDepth**	**#F**	**#T**	**#U**	**#S**	**#D**	**#Q**	**#C**
Train set	272	3.2	50	127	95	841	333	330	2734
Development set	25	3.4	12	10	3	69	11	28	173
Test set	28	2.8	12	8	8	94	71	106	778
Total	**325**	**3.2**	**74**	**145**	**106**	**1004**	**415**	**464**	**3685**

TABLE II
DETAILED STATISTICS OF RUMEVAL2019

Split	Rumor Statistics					Stance Distribution			
	#Threads	**AvgDepth**	**#F**	**#T**	**#U**	**#S**	**#D**	**#Q**	**#C**
Twitter Train	325	2.2	74	145	106	1004	415	464	3685
Reddit Train	40	3.0	24	9	7	23	45	51	1025
Total Train	365	2.3	98	154	113	1027	460	515	4700
Twitter Test	56	0.9	30	22	4	141	92	62	771
Reddit Test	25	2.9	10	9	6	16	54	31	705
Total Test	81	1.7	40	31	10	157	146	93	1476
Total	**446**	**2.3**	**138**	**185**	**123**	**1184**	**606**	**608**	**6176**

TABLE III
DETAILED STATISTICS OF PHEME

Event	Rumor Statistics					Stance Distribution				
	#Threads	**AvgDepth**	**#F**	**#T**	**#U**	**#S**	**#D**	**#Q**	**#C**	**#NS-Com**
Charlie Hebdo	458	3.5	116	193	149	248	60	61	795	380
Sydney siege	522	3.3	86	382	54	225	90	110	769	448
Ferguson	284	5.2	8	10	266	191	95	116	784	234
Ottawa shooting	470	2.9	72	329	69	171	78	83	568	407
Germanwings-crash	238	3.1	111	94	33	80	16	43	244	209
Putin missing	126	2.2	9	0	117	18	6	5	33	117
Prince Toronto	229	2.3	222	0	7	21	7	11	64	217
Gurlitt	61	1.3	0	59	2	0	0	0	0	61
Ebola Essien	14	2.4	14	0	0	6	6	1	21	12
Total	**2402**	**3.6**	**638**	**1067**	**697**	**960**	**358**	**430**	**3278**	**2085**

D. Baseline Models

We present our model in three variations (CoSDD-Depth, CoSDD-Temp, and CoSDD-Breadth) each differentiated by the encoding strategy used for hierarchical levels. CoSDD-Depth and CoSDD-Breadth adopt DFS and BFS strategies, respectively, while CoSDD-Temp utilizes temporal sequencing. These variations are evaluated against several SOTA rumor detection models:

1) **Stance-Conditioned Modeling (S-CoM)** [1]: This is our prior work, that is extended in this paper. It aggregates post embeddings and separately models stance progression with a BiLSTM for rumor verification.
2) **eventAI** [43]: This method, which secured the first position in the RumorEval-2019 competition task [44], leverages multidimensional information and employs an ensemble strategy to enhance rumor verification.
3) **MTL2-Hierarchical Transformer** [8]: This approach segments conversational threads into multiple groups based on the hierarchical structure of conversations. Each group is processed using BERT to extract contextual information, and the aggregated information is fused using a Transformer for rumor verification.
4) **Coupled Hierarchical Transformer (CHT)** [8]: Building on the MTL2-Hierarchical Transformer, this model incorporates BERT to capture contextual nuances. It further enhances performance by integrating stance information.
5) **SEMTEC** [45]: This method introduces SEMTEC, a deep learning-based approach that combines emotion features, sentiment attributes, and contextual text analysis to improve rumor detection.
6) **Joint Rumor and Stance Model (JRSM)** [9]: This framework utilizes a graph transformer to encode input data and a partition filter network to model explicitly rumor-specific, stance-specific, and shared interactive features. These features are subsequently employed for

joint rumor and stance classification.

7) **SAMGAT** [31]: This model utilizes Graph Attention Networks (GATs) to capture contextual relationships between posts. While initially applied to the PHEME dataset for binary classification (excluding the *Unverified* class), we adapt and retrain SEMTEC and SAMGAT for our experimental setup (three classes) across all relevant datasets.
8) **Knowledge Graphs (KGs)** [46]: This study proposes a knowledge graph-based methodology that automatically retrieves evidence for rumor verification.

E. Results

1) Results and Discussion: Table IV provides a comparative analysis of the performance of the models. The findings demonstrate that our model significantly outperforms the best-competing baselines, as validated by McNemar's test with a p-value < 0.05 [47], with the *CoSDD-Breadth* variant delivering the best results. Furthermore, our results exhibit a standard deviation in the range of 0.006–0.02 across all three datasets over the 10 experimental runs, indicating robust and consistent performance.

From the table, while *eventAI* uses multidimensional information and ensemble learning to boost performance, it relies heavily on pre-defined multidimensional features, which could miss latent cues in data, and it does not account for the hierarchical structure of conversations or the distribution of stances, which are critical for understanding rumor propagation dynamics. *MTL2-Hierarchical Transformer* processes hierarchical conversational threads by segmenting them into groups and using BERT to extract contextual features. It further employs a Transformer to fuse group-level information. Still, the hierarchical structure is only partially encoded, and inter-group interactions may be underrepresented and does not leverage stance distributions as part of its input for understanding conversations in rumor detection and also fails to emphasize temporal dependencies, which can provide critical context for rumor dynamics. *Coupled Hierarchical Transformer* introduces an attention mechanism to integrate stance information, which adds contextual nuance but the stance information is not as comprehensive or dynamically aligned with structural and temporal features as in our model, the model lacks a systematic representation of hierarchical dependencies.

The *Joint Rumor and Stance* baseline model effectively combines rumor and stance-specific features, but it also lacks the temporal and structural encoding capabilities that our model achieves via *CoSDD-Breadth* and *CoSDD-Temp*. *SAMGAT's* reliance on localized graph attention limits its ability to capture global relationships across threads, which our model handles effectively through hierarchical encodings. Compared to SAMGAT and SEMTEC, our model is more adaptable to three-class classification, as demonstrated by the substantial performance boost, since these two baselines were primarily intended for binary rumor classification.

The evidence-retrieval approach of *Knowledge Graphs* is limited by the availability and quality of external evidence. S-Com is close to our model's performance, but it statically models stance progression with BiLSTM without accounting for dynamic stance sequencing strategies in a conversational tree. Our models leverage hierarchical encoding to capture the complex structural relationships in conversational threads effectively. Including stance distributions as part of the input features, concurrently applying graph traversal techniques to hierarchical encoding, enables the model to align stance dynamics with rumor classification. This aligns with methods like the *Joint Rumor and Stance* model but incorporates richer attention mechanisms, providing a significant edge. The MHA mechanism ensures that the model focuses on relevant parts of the input (e.g., stance signals and structural levels). Our models enhance this by modeling the entire structural hierarchy effectively, compared to models like *SAMGAT* that focus on localized graph attention. Our approach incorporates embeddings aggregated from source posts, stance distributions, and structural hierarchies, allowing for nuanced feature representations. This approach captures latent cues that other baselines (e.g., *eventAI* and *SEMTEC*) might overlook due to reliance on pre-determined features like sentiment or emotion.

It has been noticed from the table that the *CoSDD-Breadth* variant of our model performs better than the *CoSDD-Depth* and *CoSDD-Temp* variants, and we point out the following factors. To begin with, *CoSDD-Breadth* processes nodes level by level within the hierarchical structure of a conversation thread. This allows the model to capture the broad, overall structure and relationships across different levels of the hierarchy simultaneously. In contrast, *CoSDD-Depth* focuses on deeper, more localized paths first, which might cause the *CoSDD-Depth* model to miss or underrepresent cross-level interactions that are critical for understanding the broader context of rumor propagation. Secondly, rumors often involve interactions across multiple users at the same hierarchical level (e.g., multiple replies to a single source post or other replies). *CoSDD-Breadth* captures these parallel interactions effectively by processing all posts at the same level together. *CoSDD-Depth*, on the other hand, focuses on specific threads or chains of replies, which may lead to a narrow perspective on the overall conversational dynamics and off-topic diversions. Third, *CoSDD-Breadth* ensures that stance distributions across the hierarchical levels are aggregated and attended to in a balanced way, as the model processes the entire level before moving to the next. This aligns well with our model's attention mechanism, which uses these distributions to refine the representation of conversational threads. The *CoSDD-Depth* variant, conversely, may prioritize deeper paths without adequately integrating the stance dynamics across intermediate hierarchical levels. Finally, the *CoSDD-Temp* variant primarily relies on time-ordered posts, which may not always align with the hierarchical structure. In cases where multiple posts are created simultaneously or out of sequence, temporal encoding can misrepresent the structural relationships. However, the *CoSDD-Breadth* naturally respects the hierarchical structure

TABLE IV
PERFORMANCE COMPARISON WITH BASELINE MODELS.

Model	SemEval-2017		RumorEval-2019		PHEME	
	Macro-F1	Acc	Macro-F1	Acc	Macro-F1	Acc
eventAI	0.618	0.629	0.577	0.591	0.342	0.357
MTL2-Hierarchical Transformer	0.657	0.643	0.568	0.572	0.375	0.454
Coupled Hierarchical Transformer	0.680	0.678	0.579	0.611	0.396	0.466
SAMGAT	0.702	0.709	0.542	0.562	0.409	0.418
SEMTEC	0.711	0.727	0.581	0.592	0.421	0.437
Joint Rumor and Stance Model	0.754	0.767	0.598	0.623	0.448	0.479
Knowledge Graphs	0.758	0.759	0.584	0.593	0.489	0.523
S-Com	0.774	0.781	0.636	0.648	0.641	0.643
CoSDD-Depth	0.775	0.767	0.712	0.724	0.591	0.622
CoSDD-Temp	0.776	0.789	0.728	0.731	0.595	**0.659**
CoSDD-Breadth	**0.783**	**0.798**	**0.731**	**0.765**	**0.658**	0.638

while maintaining a balance between temporal and structural features. This makes it more robust for predicting rumors from online discourse, where hierarchical relationships are crucial.

Although only Twitter and Reddit data are used in our experiments, this work can be customized and extended to any social media platform actively engaging in fact-checking and where users participate in the subsequent conversations about a source claim. Therefore, our stance-conditioned modeling for rumor verification can also be generalized to Facebook, Instagram, Threads, etc. This will be incorporated into future work.

TABLE V
ABLATION STUDY OF OUR MODEL ON THE RUMOREVAL-2019 AND PHEME DATASETS.

Model	RumorEval-2019		PHEME	
	Macro-F1	Acc	Macro-F1	Acc
-structs & emb aggreg	0.540	0.566	0.345	0.582
-structs	0.549	0.573	0.417	0.590
-emb aggreg	0.552	0.579	0.410	0.623
-hier levels	0.561	0.582	0.450	0.593
-stance distr	0.573	0.594	0.461	0.628
-MHA	0.595	0.642	0.474	0.631
CoSDD-Breadth	**0.731**	**0.765**	**0.658**	**0.638**

F. Ablation Study

The ablation study, as illustrated in Table V, provides insights into the contributions of different modules within our model. By systematically removing individual components and observing the changes in Macro-F1 and accuracy on the RumorEval-2019 and PHEME datasets, we can evaluate the importance of each module. In the table, *-structs & emb aggreg* indicates that our model relies solely on the claim post c, omitting both the structural dynamics from replies R and the embeddings aggregation $\mathbf{e}_{\text{stance}}$. The *-structs* configuration excludes the stance distribution $\mathbf{h}_{\text{dist}}$ and the hierarchical levels encoding $\mathbf{h}_{\text{struct}}$. In *-emb aggreg*, the embeddings aggregation mechanism is removed; instead, the entire rumor event is encoded as a single BERT embedding, constrained by the maximum sequence length of the language model (i.e., 512). The variant *-hier levels* excludes the hierarchical levels encoding feature, while *-stance distr* removes only the stance distribution module. The *-MHA* configuration does not utilize the MHA mechanism; instead, it directly employs the combined feature vector F for predicting rumor veracity. Lastly, *CoSDD-Breadth* represents our most comprehensive model, incorporating all modules and employing BFS traversal to encode hierarchical levels. Here is an analysis of the results:

1) Impact of Structural Dynamics and Embedding Aggregation (-structs & emb aggreg): Removing both structural dynamics and embedding aggregation mechanisms results in the worst performance across both datasets (Macro-F1 drops to 0.540 on RumorEval-2019 and 0.345 on PHEME). The model, in this configuration, relies solely on the claim post c, completely ignoring the hierarchical structure and stance information from the replies R. Additionally, excluding the embedding aggregation only, meanwhile encoding the entire rumor event as a single BERT embedding (*-emb aggreg*), further limits the model's ability to distinguish nuanced relationships between posts. This confirms that capturing relationships within conversation threads and integrating stance distributions is essential for effective rumor verification.

2) Impact of Structural Dynamics (-structs, -hier levels, -stance distr): Excluding both/either stance distribution and/or hierarchical level encodings results in a significant performance drop compared to the full model. The hierarchical structure of a conversation plays a critical role in understanding rumor propagation. Without this structural context, the model cannot effectively capture the interplay between a claim post c and replies R, and the model cannot represent how information propagates through different levels of the conversation. The stance distribution is crucial for incorporating the stance dynamics of replies into the rumor veracity decision. Without it, the model loses critical features that differentiate between supportive, commenting, denying, and querying stances. This highlights the importance of structural dynamics ($\mathbf{h}_{\text{struct}}$ and $\mathbf{h}_{\text{dist}}$) in improving veracity prediction.

3) Impact of Multi-Head Attention (-MHA): Removing the MHA mechanism equally reduces the performance of the model. MHA enhances the model by allowing it to focus on different aspects of the conversation, such as structural relationships, stance, and temporal cues. Directly using the

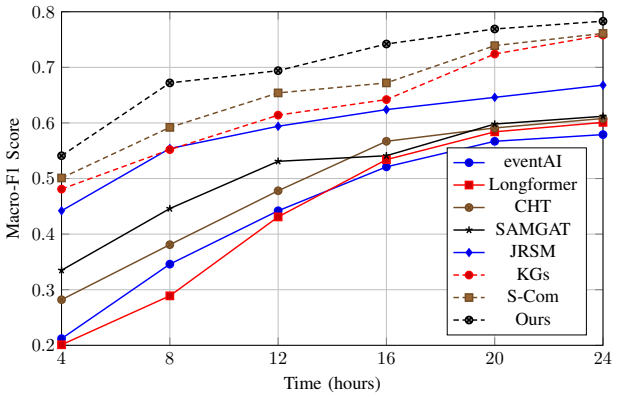

Fig. 3. Early Rumor Detection Performance on SemEval-2017.

combined feature vector reduces the model's flexibility and representational power.

The full model (*CoSDD-Breadth*) achieves the highest performance on both datasets, with a Macro-F1 of 0.731 on RumorEval-2019 and 0.658 on PHEME. Combining all modules ensures that the model captures both the global and local dynamics of a discourse. The BFS traversal specifically enhances the representation by effectively encoding hierarchical relationships across levels.

G. Early Rumor Detection

Timely detection of rumors can mitigate their widespread dissemination. To assess early rumor detection capabilities, we define detection checkpoints based on the elapsed time, spanning 24 hours, since the initial post. At each checkpoint, only replies accumulated up to that point are considered for model evaluation, and performance is measured using the Macro-F1 score at each detection interval.

Figure 3 illustrates Macro-F1 and accuracy scores over time for early rumor detection on the SemEval-2017 dataset. Our model consistently outperforms all baselines throughout the 24-hour period, demonstrating superior effectiveness in detecting rumors early. While all models improve as more information becomes available, our model achieves significantly higher Macro-F1 scores early on, starting with an advantage at 4 hours and maintaining superior performance throughout. This suggests that our approach is more responsive to limited initial data, making it highly effective for early-stage rumor identification and particularly valuable in real-world misinformation scenarios where timely intervention is crucial.

H. Performance of different pre-trained language models

To identify the most effective pre-trained language model for generating contextual embeddings of rumor events, we conducted additional experiments comparing BERT, RoBERTa, and GPT. These models were chosen due to their strong presence and performance in the existing literature. For each language model, we applied all three configurations of our approach: *Depth*, *Temporal*, and *Breadth*. The performance results for each configuration across the SemEval-2017, RumorEval-2019, and PHEME datasets are summarized in Table VI.

The results indicate a clear performance advantage for BERT-based configurations, which consistently outperform both GPT and RoBERTa counterparts across all datasets and metrics. Among the three configurations, CoSDD-Breadth-BERT achieves the highest overall scores on the SemEval-2017 dataset, with a Macro-F1 of 0.783 and an accuracy of 0.798, while also attaining top performance on RumorEval-2019 (Macro-F1 = 0.731, Acc = 0.765). On the PHEME dataset, BERT models also dominate, with CoSDD-Temp-BERT achieving the best accuracy (0.659) and CoSDD-Breadth-BERT obtaining the highest Macro-F1 (0.658). In contrast, GPT-based models show relatively lower performance, with Macro-F1 scores generally below 0.73 for SemEval-2017 and below 0.70 for RumorEval-2019. RoBERTa models perform better than GPT but still fall short of BERT, suggesting that BERT's embedding representations are more effective for this rumor verification task, particularly when combined with the *Breadth* configuration.

V. Conclusion

This study presents a novel approach to rumor verification that leverages hierarchical structural encoding, stance distributions, and MHA mechanisms to capture the intricate dynamics of rumor propagation in conversational threads. Our model demonstrates SOTA performance across multiple benchmark datasets, including SemEval-2017, RumorEval-2019, and PHEME, outperforming existing approaches with statistically significant improvements in Macro-F1 and accuracy metrics. Our findings highlight several key insights: the explicit incorporation of stance information significantly improves rumor verification, demonstrating that user reactions provide crucial contextual cues. Our experiments further show that breadth-based graph traversal outperforms depth-based and temporal-based sequencing strategies for hierarchical encodings. Early rumor detection analysis demonstrates that our model achieves faster and more accurate misinformation detection than competing methods, underscoring its practical utility in real-world misinformation detection; meanwhile, BERT indicated stronger performance over RoBERTa and GPT when encoding discourse embeddings.

While the model has shown success, its limitations include a heavy reliance on accurate stance annotations—which might not be consistently available—and training on datasets that may not fully represent real-world misinformation trends across diverse social media platforms. Additionally, the focus on textual content ignores the visual aspects (such as images, memes, and videos) that often accompany online rumors. Future work could reduce dependence on manually labeled data through weakly supervised and self-supervised learning, improve generalization via cross-platform adaptation, incorporate multi-modal data, and further explore extra structural

TABLE VI
PERFORMANCE COMPARISON OF PRE-TRAINED LANGUAGE MODELS.

Model	SemEval-2017		RumorEval-2019		PHEME	
	Macro-F1	Acc	Macro-F1	Acc	Macro-F1	Acc
CosDD-Depth-GPT	0.709	0.719	0.682	0.687	0.522	0.551
CosDD-Temp-GPT	0.712	0.723	0.689	0.691	0.523	0.536
CosDD-Breadth-GPT	0.725	0.731	0.692	0.703	0.541	0.545
CosDD-Temp-RoBERTa	0.721	0.739	0.698	0.713	0.558	0.575
CosDD-Depth-RoBERTa	0.734	0.742	0.706	0.710	0.545	0.552
CosDD-Breadth-RoBERTa	0.748	0.761	0.712	0.715	0.557	0.601
CoSDD-Depth-BERT	0.775	0.767	0.712	0.724	0.591	0.622
CoSDD-Temp-BERT	0.776	0.789	0.728	0.731	0.595	**0.659**
CoSDD-Breadth-BERT	**0.783**	**0.798**	**0.731**	**0.765**	**0.658**	0.638

dynamics like stance distribution, hierarchical level encoding, and attention mechanisms.

Finally, while the work has positive implications, ethical challenges and risks persist. False negatives and false positives could respectively suppress credible information or allow misinformation to spread, so human validation of predictions is recommended. The system's success could also enable misuse, such as censorship or targeting, requiring transparent deployment and strict ethical guidelines. Additionally, training data biases may lead to unfair outcomes; therefore, evaluating and mitigating these biases is crucial.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation (NSF) under Award number OIA-1946391, Data Analytics that are Robust and Trusted (DART).

REFERENCES

[1] G. Nkhata and S. Gauch, "Stance-conditioned modeling for rumor verification," in *The Seventeenth International Conference on Information, Process, and Knowledge Management (eKNOW 2025)*. Nice, France: IARIA, May 2025, pp. 30–36, copyright © IARIA, 2025.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019. [Online]. Available: https://arxiv.org/abs/1810.04805

[3] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[4] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.aap9559

[5] Y. Özkent, "Social media usage to share information in communication journals: An analysis of social media activity and article citations," *PLOS One*, vol. 17, no. 2, p. e0263725, 2022.

[6] E. Kochkina, M. Liakata, and A. Zubiaga, "All-in-one: Multi-task learning for rumour verification," *arXiv preprint arXiv:1806.03713*, 2018.

[7] J. Manurung, P. Sihombing, M. A. Budiman *et al.*, "Dynamic rumor control in social networks using temporal graph neural networks," in *2023 International Conference of Computer Science and Information Technology (ICOSNIKOM)*. IEEE, 2023, pp. 1–5.

[8] J. Yu, J. Jiang, L. M. S. Khoo, H. L. Chieu, and R. Xia, "Coupled hierarchical transformer for stance-aware rumor verification in social media conversations." Association for Computational Linguistics, 2020.

[9] N. Luo, D. Xie, Y. Mo, F. Li, C. Teng, and D. Ji, "Joint rumour and stance identification based on semantic and structural information in social networks," *Applied Intelligence*, vol. 54, no. 1, pp. 264–282, 2024.

[10] H. Gong, M. Zhang, Q. Liu, S. Wu, and L. Wang, "Breaking event rumor detection via stance-separated multi-agent debate," *arXiv preprint arXiv:2412.04859*, 2024.

[11] J. Bian *et al.*, "Multi-task learning for stance and rumor detection in online conversations," in *Proceedings of ICWSM*, 2023.

[12] Y. Bai, C. Han, and Y. Jia, "Samgat: Structure-aware multilevel graph attention networks for automatic rumor detection," *PeerJ Computer Science*, vol. 9, p. e1032, 2024.

[13] P. Yang, J. Leng, G. Zhao, W. Li, and H. Fang, "Rumor detection driven by graph attention capsule network on dynamic propagation structures," *The Journal of Supercomputing*, vol. 79, pp. 15 674–15 690, 2023.

[14] L. Derczynski, K. Bontcheva, M. Liakata, R. Procter, G. W. S. Hoi, and A. Zubiaga, "Semeval-2017 task 8: Rumoureval: Determining rumour veracity and support for rumours," *arXiv preprint arXiv:1704.05972*, 2017.

[15] J. Li, Y. Bin, Y. Ma, Y. Yang, Z. Huang, and T.-S. Chua, "Filter-based stance network for rumor verification," *ACM Transactions on Information Systems*, vol. 42, no. 4, pp. 1–28, 2024.

[16] E. Elmurngi and A. Gherbi, "An empirical study on detecting fake reviews using machine learning techniques," in *2017 Seventh International Conference on Innovative Computing Technology (INTECH)*. IEEE, 2017, pp. 107–114.

[17] J. Ma, W. Gao, and K.-F. Wong, "Rumor detection on twitter with tree-structured recursive neural networks," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, 2018.

[18] Y. Yang, L. Zheng, J. Zhang, Q. Cui, Z. Li, and P. S. Yu, "Ti-cnn: Convolutional neural networks for fake news detection," *arXiv preprint arXiv:1806.00749*, 2018.

[19] Q. Lv, Y. Wang, B. Zhang, and Q. Jin, "Rv-ml: An effective rumor verification scheme based on multi-task learning model," *IEEE Communications Letters*, vol. 24, no. 11, pp. 2527–2531, 2020.

[20] Y. Luo, J. Ma, and C. K. Yeo, "Bcmm: A novel post-based augmentation representation for early rumour detection on social media," *Pattern Recognition*, vol. 113, p. 107818, 2021.

[21] Y. Wang, B. Zhang, J. Ma, and Q. Jin, "Marv: Multi-task learning and attention based rumor verification scheme for social media," in *2022 IEEE/CIC International Conference on Communications in China (ICCC)*. IEEE, 2022, pp. 94–98.

[22] J. Ma, W. Gao, and K.-F. Wong, "Rumor detection on twitter with tree-structured recursive neural networks," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018, pp. 1980–1989.

[23] C. Yuan, Q. Ma, W. Zhou, J. Han, and S. Hu, "Jointly embedding the local and global relations of heterogeneous graph for rumor detection," in *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 796–805.

[24] Y.-J. Lu and C.-T. Li, "Gcan: Graph-aware co-attention networks for explainable fake news detection on social media," *arXiv preprint arXiv:2004.11648*, 2020.

[25] L. Wei, D. Hu, W. Zhou, Z. Yue, and S. Hu, "Towards propagation uncertainty: Edge-enhanced bayesian graph convolutional networks for rumor detection," *arXiv preprint arXiv:2107.11934*, 2021.

[26] Q. Mai, S. Gauch, D. Adams, and M. Huang, "Sequence graph network for online debate analysis," 2025. [Online]. Available: https://arxiv.org/abs/2406.18696

[27] H. Sahni *et al.*, "Temporal dynamics in rumor propagation: Challenges and solutions," in *Proceedings of the International Conference on Web and Social Media (ICWSM)*, 2022.

[28] J. Choi, T. Ko, Y. Choi, H. Byun, and C.-k. Kim, "Dynamic graph convolutional networks with attention mechanism for rumor detection on social media," *PLOS One*, vol. 16, no. 8, p. e0256039, 2021.

[29] A. Zubiaga, M. Liakata, R. Procter, G. Wong Sak Hoi, and P. Tolmie, "Analysing how people orient to and spread rumours in social media by looking at conversational threads," *PLOS One*, vol. 11, no. 3, p. e0150989, 2016.

[30] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the EMNLP-IJCNLP 2019*, 2019. [Online]. Available: https://arxiv.org/abs/1908.10084

[31] Y. Li, Z. Chu, C. Jia, and B. Zu, "Samgat: structure-aware multilevel graph attention networks for automatic rumor detection," *PeerJ Computer Science*, vol. 10, p. e2200, 2024.

[32] X. Li, Y. Zhou, and J. Liu, "Weighted graphsage with attention mechanism for rumor detection," *Scientific Reports*, vol. 13, no. 1, p. 12345, 2023.

[33] H. Wang, L. Zhao, and K. Liu, "Multi-attention neural interaction networks for rumor detection," in *Proceedings of the IEEE International Conference on Big Data*. IEEE, 2023, pp. 1023–1034.

[34] W. Zhang, Y. Chen, and Z. Li, "Attention graph adversarial dual contrast learning for rumor detection in social media," *PLOS One*, vol. 18, no. 8, p. e0290291, 2023.

[35] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[36] G. Gorrell, E. Kochkina, M. Liakata, A. Aker, A. Zubiaga, K. Bontcheva, and L. Derczynski, "SemEval-2019 task 7: RumourEval, determining rumour veracity and support for rumours," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, J. May, E. Shutova, A. Herbelot, X. Zhu, M. Apidianaki, and S. M. Mohammad, Eds. Minneapolis, Minnesota, USA: Association for Computational Linguistics, Jun. 2019, pp. 845–854. [Online]. Available: https://aclanthology.org/S19-2147

[37] A. Zubiaga, M. Liakata, and R. Procter, "Learning reporting dynamics during breaking news for rumour detection in social media," *arXiv preprint arXiv:1610.07363*, 2016.

[38] B. Han and T. Baldwin, "Lexical normalisation of short text messages: Makn sens a# twitter," in *Proceedings of the 49th Annual meeting of The Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 368–378.

[39] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," in *arXiv preprint arXiv:1907.11692*, 2019.

[40] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[41] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[42] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[43] Q. Li, Q. Zhang, and L. Si, "eventAI at SemEval-2019 task 7: Rumor detection on social media by exploiting content, user credibility and propagation information," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, J. May, E. Shutova, A. Herbelot, X. Zhu, M. Apidianaki, and S. M. Mohammad, Eds. Minneapolis, Minnesota, USA: Association for

Computational Linguistics, Jun. 2019, pp. 855–859. [Online]. Available: https://aclanthology.org/S19-2148

[44] G. Gorrell, K. Bontcheva, L. Derczynski, E. Kochkina, M. Liakata, and A. Zubiaga, "Rumoureval 2019: Determining rumour veracity and support for rumours," *arXiv preprint arXiv:1809.06683*, 2018.

[45] D. Sharma and A. Srivastava, "Detecting rumors in social media using emotion based deep learning approach," *PeerJ Computer Science*, vol. 10, p. e2202, 2024.

[46] J. Dougrez-Lewis, E. Kochkina, M. Liakata, and Y. He, "Knowledge graphs for real-world rumour verification," in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 2024, pp. 9843–9853.

[47] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947.