

Article

Quantum-Inspired Semantic Encoding and Temporal Transformer Fusion (QuST-TF) for Misinformation Detection

Krishna Kumar *  and Akila Venkatesan 

Department of Computer Science and Engineering, Puducherry Technological University, Puducherry 605014, India; akila@ptuniv.edu.in

* Correspondence: krish1095@ptuniv.edu.in

Abstract

Misinformation propagates more rapidly than factual content on social media, presenting significant challenges for automated misinformation detection. Existing approaches often focus solely on textual features without incorporating temporal information, treat timing and propagation as separate factors, or apply quantum-inspired methods primarily to multimodal data rather than text-centric misinformation. This study introduces QuST-TF (Quantum-inspired Semantic encoding and Temporal Transformer Fusion), a unified model designed to detect misinformation in tweets and news articles. QuST-TF integrates quantum-inspired (classical approximation) amplitude encoding, time-aware Transformer fusion, and propagation graph attention based on engagement data, without reliance on images, audio, or quantum hardware. Performance gains are achieved through quantum-inspired (classical approximation) nonlinear angular modulation (cosine and sine rotations) implemented via classical computation, rather than genuine quantum computing. All computations utilize classical Dense layers, Rectified Linear Unit (ReLU) activations, and cosine/sine functions on CPUs or GPUs; quantum hardware is not required. The quantum-inspired (classical approximation) layer applies classical rotation-based transformations to enrich the semantic representation of BERT (Bidirectional Encoder Representations and Transformer) embeddings. Temporal information is captured by a dual-attention Transformer encoder, while propagation graph attention monitors the spread of claims. Evaluation on FakeNewsNet and PHEME datasets demonstrates 91.4% and 95.5% accuracy, respectively, with 34% fewer trainable parameters compared to standard Transformers. Ablation studies indicate that quantum encoding is the most influential component (+3.0% versus without quantum encoding), surpassing the contributions of graph attention (+2.6%) and temporal attention (+2.2%). The integration of all three components yields a 1.3% synergistic improvement, confirming effective inter-module collaboration. Attention visualization enhances interpretability, supporting the utility of QuST-TF for fact-checking applications.



Academic Editor: Nikos Konofaos

Received: 4 May 2026

Revised: 5 June 2026

Accepted: 11 June 2026

Published: 24 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\) license](https://creativecommons.org/licenses/by/4.0/).

Keywords: quantum simulation; quantum temporal transformer; quantum fusion models; misinformation detection; semantic encoding; classical vs. quantum transformers

1. Introduction

Misinformation disseminated through digital platforms compromises information integrity and disrupts public discourse [1,2]. Empirical studies demonstrate that false information propagates six to ten times more rapidly than factual content on social media, thereby reaching wider audiences in a shorter period [3]. Conventional misinformation

detection methods primarily utilize either content-based features, such as linguistic patterns, or social context features, including user credibility and sharing behaviors [4,5]. However, the isolated application of these approaches frequently neglects the intricate relationships between textual content and the temporal dynamics of news dissemination.

Recent advances in natural language processing, particularly Transformer architectures, have demonstrated remarkable capability in capturing long-range semantic dependencies through self-attention mechanisms [6,7]. Recently, quantum machine learning studies incorporate variational quantum circuits (VQCs) and parameterized quantum circuits (PQCs) to extract content (text, image, audio, and video) representations [8–11]. However, these methods rely on quantum hardware or simulators, which makes them less practical for large social media datasets (current scenario). On the other hand, quantum-inspired (classical approximation) feature extraction algorithms use mathematical ideas from quantum computing, such as amplitude encoding, density matrix representations, and superposition-like feature spaces, but do not require quantum hardware or claim a quantum speedup [12–15]. However, the application of quantum-inspired (classical approximation) feature encoding with temporal transformer modelling and propagation graph attention for misinformation detection remains unexplored.

Important Clarification: QuST-TF employs a quantum-inspired (classical approximation) encoder, implemented exclusively on classical computers using Dense layers, ReLU, and cosine/sine operations. We do not utilize quantum circuits, quantum hardware, or any form of quantum processing. The observed performance improvements result from quantum-inspired (classical approximation) rotation-based feature transformation (cosine/sine angular modulation), which captures semantic patterns more effectively than standard attention mechanisms. These enhancements do not stem from genuine quantum computing. Our model constitutes a classical approximation informed by principles of quantum mechanics.

Moving on to the literature review, the literature covers three key research areas related to the QuST-TF architecture: transformer-based and classical misinformation detection, Temporal and propagation-aware misinformation detection, Quantum-inspired and hybrid-quantum methods for NLP and Quantum Methods for Misinformation Detection. We selected papers published between 2024 and 2025 from IEEE Access, ACL, Elsevier, and MDPI to capture the latest developments. Together, these works set the stage for QuST-TF's design, which combines classical quantum-inspired (classical approximation) encoding, time-aware Transformers, and propagation graph attention in a single text-only framework.

1.1. Transformer-Based and Classical Misinformation Detection

Transformer-based models remain the leading approach for classifying misinformation based on content. Huang et al. [6] introduced a hybrid Transformer-BiGRU model with Bayesian hyperparameter tuning, reaching 99.73% accuracy on FakeNewsNet. Although this sets a strong accuracy benchmark, the model uses 43.2 million parameters and requires significant computational resources, limiting its practical use. Karande et al. [5] combined BERT embeddings with CNNs and stance detection, achieving 95% accuracy on Kaggle data (Kaggle, San Francisco, USA), but their method needs time-consuming stance annotations. Shanmugavadivel et al. [16] showed that a classical SVM with mBERT scores 0.97 on F1 for Malayalam misinformation, demonstrating that traditional machine learning still performs well in low-resource settings, although it lacks temporal or propagation awareness.

1.2. Temporal and Propagation-Aware Modeling

Temporal dynamics and social propagation patterns provide strong clues for detecting misinformation. Wu et al. [7] introduced the Temporal Tree Transformer (TTT), which

encodes propagation using tree-structured GRUs. This method achieved 75.84% accuracy on PHEME. Hu et al. [17] developed Dynamic Temporal Networks (DTN) with time-aware node weighting to address multimodal misinformation, achieving 91% accuracy. However, DTN requires synchronized multimodal inputs. Peng et al. [18] used structural entropy coding of rumor trees and got 88% accuracy on Twitter data. Plepi et al. [19] modeled user interaction networks with dynamic GNNs, detecting spreaders on Reddit-FACTOID dataset (Reddit, San Francisco, CA, USA) and achieving 89% accuracy. Aktar et al. [20] proposed a Quantum Graph Transformer (QGT) that embeds quantum self-attention in graph message passing. This achieved 93.0% accuracy on Yelp with half the training samples. It shows that quantum-enhanced graph attention can improve results over classical GNNs, though this was applied to sentiment analysis rather than misinformation detection.

1.3. Quantum-Inspired and Hybrid Quantum Methods for NLP

Quantum-inspired methods have become more popular in NLP tasks in 2025. Pan et al. [12] introduced AQCF (Adaptive Quantum-Classical Fusion), which uses entropy-driven adaptive VQC (Variational Quantum Circuit) and quantum memory banks for sentiment classification, improving performance by 2.52 to 4.12% over classical Transformers on SST-2 and IMDB datasets (IMDB, Seattle, Washington, DC, USA) with 20 qubits. Chen and Lou [13] developed CQKSAN, a complex quantum kernel self-attention network that combines frozen BERT with a 4-qubit PQC, reaching 98.55% validation accuracy on text classification benchmarks, a 28.35% improvement over LSTM (Long Short-Term Memory) baselines. Pal and Das [14] proposed QLSTM, which replaces LSTM weight matrices with VQCs, achieving 87.18% F1 for sarcasm detection and 70.04% F1 for claim identification, with a 3.46% gain over classical LSTM. Hui et al. [15] applied density matrix representations for inter-sentence semantic modeling (QISIM), maintaining 78.8% accuracy under adversarial PWWS attacks compared to 20.8% for standard Transformers, showing the strength of quantum-inspired features. Gruzdeva et al. [21] used a quantum-like wave interference model for text classification, achieving 80.4% accuracy with a 15% gain from interference modeling, although this was tested on only 500 samples.

1.4. Quantum Methods for Misinformation Detection Specifically

Several studies apply quantum methods to detect misinformation. Bikku and Thota [8] proposed QEMF (Quantum Encoding with Multimodal Fusion), which uses quantum entanglement and VQC-QCNN fusion (Variational Quantum Circuit-Quantum Convolutional Neural Network) across text, image, and audio, reaching 94.2% accuracy on FakeNewsNet. Suneesh and Palani [9] introduced QCNN-MFND, a quantum CNN framework for multimodal misinformation, achieving 92.7% on PolitiFact with 8-qubit circuits. Aishwarya et al. [22] surveyed quantum deep learning for misinformation, noting that it offers better parameter efficiency (9 million parameters vs. 18 million parameters classical) and multimodal alignment but suffers from a 58-times slower inference speed. Khalil et al. [11] proposed PegasosQSVM with a ZZFeatureMap quantum kernel, achieving 95.63% accuracy and 99.52% recall on BuzzFeed dataset, though it only uses propagation features such as likes and shares, without analyzing content. Altıntaş [10] developed HQDNN, a small 2-qubit hybrid quantum network for text-only misinformation on the LIAR dataset, achieving 94.40% recall with DistilBERT embeddings but only 56.52% overall accuracy, lacking temporal or propagation modeling.

1.5. Parameter Efficiency and Cross-Domain Quantum Applications

Hybrid quantum architectures offer parameter-efficiency improvements that matter for QuST-TF's design, beyond NLP. Bischof et al. [23] found that hybrid QNNs using VQC and ZZFeatureMap reduced the number of cut parameters by about 10 times compared

to classical neural networks for entity matching, tested on real International Business Machines Corporation (IBM) Hanoi hardware (Armonk, NY, USA; Hanoi, Vietnam). Li et al. [24] introduced HQRNN-FD for detecting financial fraud, using dual-angle encoding with hierarchical entanglement, reaching 97.15% accuracy and a 2.4% improvement over classical RNNs. Mondal et al. [25] swapped classical parts in OpenAI Whisper (OpenAI, San Francisco, USA) with quantum modules like QCNN and QLSTM, gaining 1.49% accuracy in speech recognition. This shows that replacing classical modules with quantum ones works across different fields, although it increases simulation time.

1.6. Research Gap, Positioning of QuST-TF and Primary Contributions of This Study

Table 1 reveals a clear trend in current research: quantum-inspired methods for misinformation detection either work in multimodal settings like QEMF [8] and QCNN-MFND [9], or as text-only models without temporal modeling or propagation graph attention, such as HQDNN [10] and PegasosQSVM [11]. Models that include temporal and propagation awareness, like TTT [18] and DTN [17], do not use quantum-inspired semantic encoding. Classical text embeddings map features into fixed-dimensional spaces via linear transformations, limiting their ability to capture complex, high-order semantic relationships in misinformation.

Table 1. Comparative analysis of related works.

Works	Domain	Quantum Method	Modality	Temporal	Propagation
Huang et al. [6]	Misinformation	Transformer-BiGRU	Text	X	X
Wu et al. [7]	Rumor	Temporal Tree Transformer	Text	✓	✓
Hu et al. [17]	Misinformation	Dynamic Temporal Network	Multimodal	✓	✓
Bikku & Thota [8]	Misinformation	QEMF (VQC + QCNN)	Multimodal	X	X
Suneesh & Palani [9]	Misinformation	QCNN-MFND	Multimodal	X	X
Altıntaş [10]	Misinformation	HQDNN (2-qubit PQC)	Text	X	X
Khalil et al. [11]	Misinformation	PegasosQSVM + QKernel	Text	X	✓
Chen & Lou [13]	Text Class.	CQKSAN (BERT + PQC)	Text	X	X
Aktar et al. [20]	Sentiment	Quantum Graph Transformer	Text	X	✓
Pan et al. [12]	Sentiment	Adaptive VQC + Mem Banks	Text	X	X
Hui et al. [15]	Text Class.	QISIM	Text	X	X
Pal & Das [14]	Sarcasm/Claim	QLSTM (VQC)	Text	X	X
Li et al. [24]	Fraud	HQRNN-FD (VQC + RNN)	Tabular	X	X
Bischof et al. [23]	Entity Match	VQC + ZZFeatureMap	Text	X	X
QuST-TF (Proposed)	Misinformation	Quantum Inspired + Amplitude encoding + Temporal Transformer + propagation graphs	Text	✓	✓

Quantum-inspired amplitude encoding via classical approximation solves this by using rotation-based angle modulation to create richer, higher-dimensional representations via nonlinear transformations, all performed as classical operations without requiring quantum hardware. No current model combines all three components: quantum-inspired amplitude encoding, temporal modeling, and propagation graph attention in a single parameter-efficient design. QuST-TF fills this gap by integrating these components into a unified architecture that relies primarily on textual content, with a propagation structure derived from lightweight engagement metadata.

The key contributions are as follows:

- The Quantum-inspired (classical approximation) Amplitude Encoding Layer uses the math behind quantum amplitude encoding, especially angle modulation with cosine and sine rotations, as a purely classical process. This lets it create richer, higher-dimensional text representations without needing quantum hardware.

- The Temporal-Propagation Transformer combines learnable temporal positional embeddings with heterogeneous graph attention (GAT) in one dual-stream module. This setup models both the timing of information spread and the social propagation structure together, so separate temporal and propagation pipelines are not needed.
- We perform component-level ablation to assess how each module contributes on its own and together, showing how integrating them improves performance beyond what each part can do alone.

The rest of the work is organized as follows. Section 2 discusses the in-depth methodology, materials and implementation details incorporated in the proposed work. In Section 3, the experimental results, an ablation study, and error analysis via a confusion matrix. Section 4 discusses attention-based visualizations to interpret model’s classification decision, an overview of limitations, and future directions. Finally, we conclude in Section 5 by summarizing the entire research work.

2. Materials and Methods

Figure 1 depicts the abstractive workflow of the proposed work. It encapsulates six modules, starting from the input module, the preprocessing module, the quantum-inspired (classical approximation) amplitude encoding module, the temporal transformer encoder, propagation graph attention, and finally, the fusion and classification module.

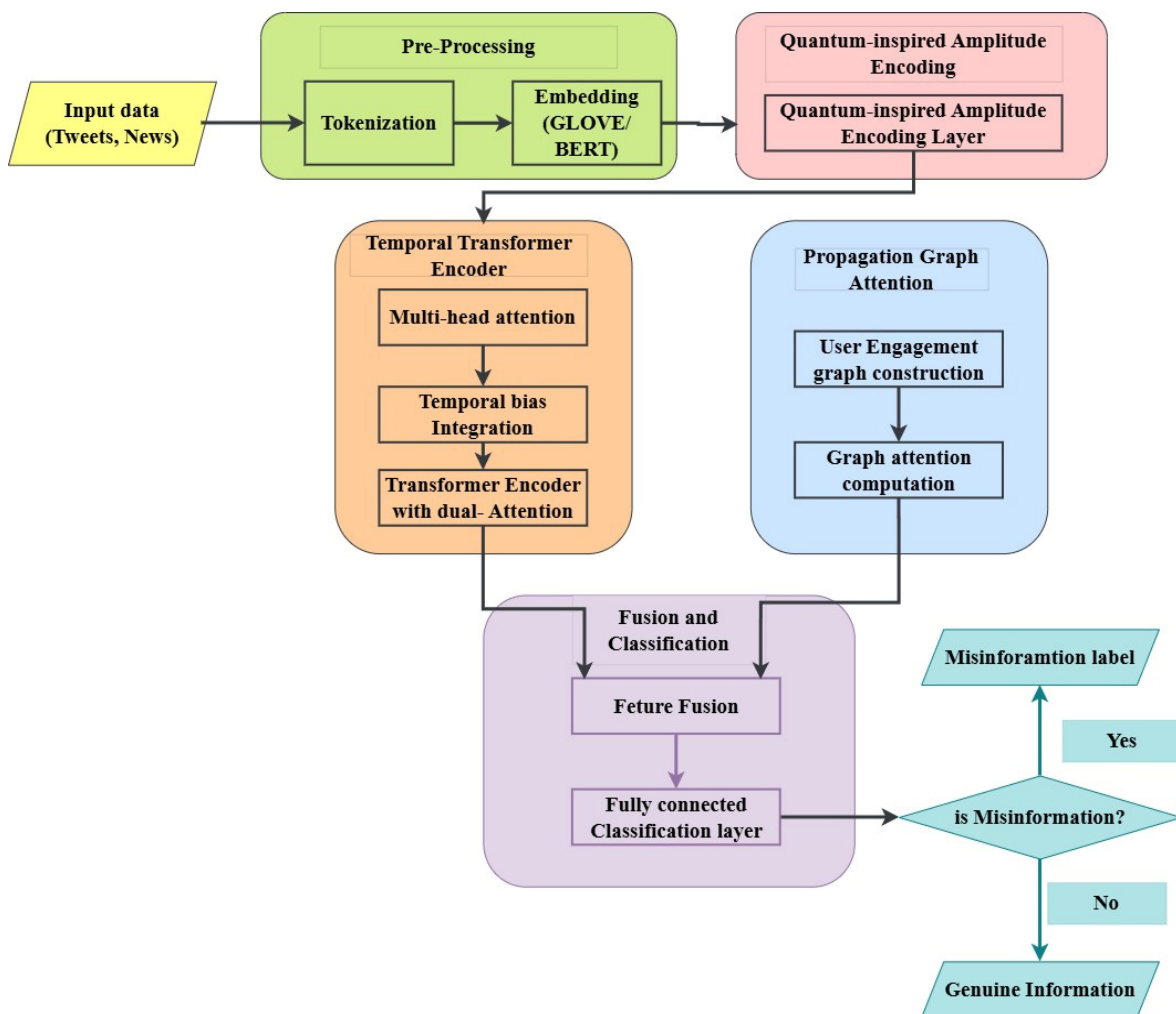


Figure 1. Workflow of the QuST-TF model. Quantum Amplitude Encoding → Dual-Attention Temporal Transformer → Propagation Graph Attention → FC Classifier for misinformation detection.

2.1. Problem Formulation

Given a collection of textual claims $D = \{(x_i, y_i, t_i, G_i)\}_{i=1}^N$, where $x_i \in R^{L \times d}$ represents a tokenized document (L tokens, embedding dimension d), $y_i \in \{0, 1\}$ indicates veracity label (0: misinformation, 1: factual), $t_i \in R^+$ denotes timestamp of publication, and G_i represents the propagation graph (user engagement network), the objective is to learn a classification function $f : (X, T, G) \rightarrow Y$ that maximizes classification accuracy while maintaining: (1) parameter efficiency compared to baseline Transformers, and (2) interpretability through attention mechanisms. The optimization objective is $\hat{\theta} = \operatorname{argmin}_{\theta} L(\theta)$, where $L(\theta) = L_{CE}(\theta) + \lambda \|\theta\|_1$. L_{CE} is cross-entropy loss, $\lambda \|\theta\|_1$ is the L1 regularization that controls sparsity in the model parameters and $\|\theta\|_1 = \sum_i |\theta_i|$ is the sum of absolute values of all parameters θ_i .

2.2. Quantum-Inspired (Classical Approximation) Amplitude Encoding Layer

The proposed quantum-inspired (classical approximation) encoding layer maps each BERT token embedding $e_t \in R^{768}$ into a quantum-inspired (classical approximation) feature representation using classical computations. The proposed encoder is quantum-inspired (classical approximation), and it is not a real quantum algorithm. The algorithm was implemented entirely on classical computers using Dense layers, ReLU activation functions, and cosine and sine operations, eliminating the need for real quantum hardware. Algorithm 1 describes how this approximation is done via a classical neural network.

For a BERT embedding vector $e_t = [e_1, e_2, \dots, e_{768}]^T$, where t is the token index, T is the total sequence length, and 768 is the BERT’s embedding dimension. We first apply L2-normalization:

$$x_n = \frac{e_t}{\|e_t\|^2} \tag{1}$$

where $x_n \in R^{768}$ is the normalized embeddings and $\|e_t\|^2 = \sqrt{\sum_{j=1}^{768} e_{t,j}^2}$ is the squared L2 norm. This normalization step is necessary because subsequent rotation operations are only geometrically valid when the input has unit norm (similar to quantum states must satisfy $\|\psi\|^2 = 1$). Without this, rotation matrices would scale the vectors rather than purely rotate them, breaking the quantum-inspired (classical approximation) analogy and distorting probability calculations downstream. Next, we extract amplitude and phase features using two Dense (fully connected) layers, compressing 768 dimensions to 16:

$$\text{amp} = \text{Dense}_{768 \rightarrow 16}(x_n) \text{ and } \text{phs} = \text{Dense}_{768 \rightarrow 16}(x_n) \tag{2}$$

Here, $\text{amp} \in R^{16}$ represents amplitude features (similar to the magnitude of quantum amplitudes) and $\text{phs} \in R^{16}$ represents phase features (similar to quantum phase angles). This compression from 768 to 16 is required to form the $n_q = 8$ qubit-like 2D vectors; separating amplitude and phase allows us to independently control magnitude and direction in the quantum-inspired (classical approximation) representation, mimicking the amplitude-phase structure of real quantum states.

We reshape these 16 values into $n_q = 8$ qubits, where each qubit is a 2D vector $[q[i, 0], q[i, 1]] \in R^2$ (each is a 2D vector, mimics quantum states), with $i = 0, \dots, 7$ indexing the qubits. We then create a weighted sum of amplitude and (0.3) phase features:

$$\psi = \text{normalize}(\text{amp} + 0.3 \times \text{phs}) \tag{3}$$

where $\psi \in R^{16}$ is the normalized weighted combination and 0.3 is a fixed weighted coefficient. This phase is weighted less (0.3) because amplitude carries the primary semantic

information from BERT embeddings, while phase provides secondary directional context. We then initialize each qubit (2D vectors) as:

$$q[i] = \frac{(\psi[:, i] + \text{roll}(\psi[:, i], 1))}{\sqrt{2}}, \quad i = 0, \dots, 7 \tag{4}$$

where $\text{roll}(\psi[:, i], 1)$ performs a circular shift of the values by 1 position and $\sqrt{2}$ normalizes the amplitude of the combined vector. This initialization introduces local correlation between adjacent features before any variational transformation (similar to quantum superposition initialization). Thus, ensuring each 2D qubit vector $q[i] \in R^2$ captures neighborhood context from the beginning. After this step, within each variational layer $l = 0, 1, 2$, and for each qubit, $i = 0, \dots, 7$, we apply single-qubit rotations R_x, R_y, R_z using classical cos/sin operations followed by:

$$c = \cos\left(\frac{\theta[l, i, k]}{2}\right), s = \sin\left(\frac{\theta[l, i, k]}{2}\right) \tag{5}$$

$$\begin{bmatrix} q[i, 0]^{new} \\ q[i, 1]^{new} \end{bmatrix} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} q[i, 0] \\ q[i, 1] \end{bmatrix} \tag{6}$$

where $\theta[l, i, k] \in R$ is a learnable rotation parameter ($l = 0, 1, 2$ layers, $i = 0, \dots, 7$ qubits and $k = 0, 1, 2$ rotation types representing R_x, R_y, R_z), and c, s are the cosine and sine of the half-rotation angle. The rotation matrix $\begin{bmatrix} c & -s \\ s & c \end{bmatrix}$ is the real part of the Pauli rotation matrix from quantum mechanics (removing the imaginary unit but preserving the rotation structure [26]). We use cos/sin instead of other non-linearities (e.g., tanh) because they preserve geometrical rotations [27]. Next, they preserve norm ($\cos^2\theta + \sin^2\theta = 1$), ensuring the qubit vector length is unchanged after transformation and extracts semantic patterns. They also provide periodic nonlinearity, useful for capturing repeating linguistic patterns in misinformation text [28,29].

After single-qubit rotations, we apply CNOT-like gates that simulates inter-qubit correlation through probability-based mixing (Not real quantum entanglement):

$$p = q[i, 0]^2$$

$$q[i + 1, 0]^{new} = p \times q[i + 1, 1] + (1 - p) \times q[i + 1, 0]$$

$$q[i + 1, 1]^{new} = p \times q[i + 1, 0] + (1 - p) \times q[i + 1, 1]$$

where $p \in [0, 1]$ is the control probability computed from the squared first component of qubit i , and this mixes the components of qubit $i + 1$ based on qubit's probability. This operation classically approximates the CNOT gate's effect of conditionally flipping a target qubit based on the control qubit, enabling cross-qubit feature interaction that captures long-range dependencies in the text representations [26]. We additionally apply a circular CNOT between the first qubit ($i = 0$) and the last qubit ($i = 7$) to introduce global correlation across all 8 feature groups, preventing information isolation at the boundaries.

We next compute the classical probabilities for each the 8 qubits: $pr[i] = q[i, 0]^2 + 10^{-9}$. Where $q[i, 0]^2$ is the squared first component of qubit i (similar to Born's rule in quantum measurement, where $probability = amplitude^2$) and 10^{-9} is a small numerical constant to avoid $\log(0)$ in the entropy calculation. We then normalize these probabilities across all qubits: $pr = pr / \sum pr$, ensuring $\sum pr[i] = 1$, so it forms valid probability distribution. Using this probability distribution, we calculate Shannon entropy: $H = -\sum (pr[i] \times \log(pr[i] + 10^{-9}))$, which measures the uncertainty/spread of informa-

tion across the 8 qubits (high entropy means features are distributed evenly, while low entropy means focused representations). This entropy signal H is then used to compute adaptive importance weights for each qubit via $w = \text{sigmoid}(\text{Linear_ent}([H, 1 - H]))$, where Linear_ent is a Dense layer that maps the 2D entropy vector $[H, 1 - H]$ to qubit-level importance scores (one weight per qubit per batch), and sigmoid normalizes weights to probabilities. This entropy-driven weighting allows the model to automatically focus on the most informative qubits (indicative of misinformation) for each input sample rather than treating all qubits equally. We then stack all qubit vectors into $q_m = \text{stack}(q) \in \mathbb{R}^{(B \times 8 \times 2)}$ (B is batch size, 8 qubits, each 2D). Then apply entropy weighting via element-wise multiplication $q_v = q_m \times w$. This step obtains important scaled qubit features. Finally, we join both raw (unweighted) representations q_m and the entropy-weighted representation $q_v : v = \text{concat}(\text{flat}(q_m), \text{flat}(q_v)) \in \mathbb{R}^{(B \times 32)}$. Where flat reshape $(B, 8, 2)$ to $(B, 16)$, so the concatenation step gives $16 + 16 = 32$ dimensions. This concatenation preserves both original feature structure (q_m) and the entropy-modulation view (q_v), giving richer information for the final projection. The 32-dimensional vector v is then projected back to 768 dimensions using a Dense Layer:

$$z = \text{LayerNorm}(\text{ReLU}(\text{Linear_out}(v))) \in \mathbb{R}^{(B \times 768)} \tag{7}$$

where Linear_out maps $32 \rightarrow 768$, $\text{ReLU}(x) = \max(0, x)$ introduces nonlinearity and sparsity, and the LayerNorm stabilizes training by normalizing activations across the 768 dimensions. The final output ψz is a 768-dimensional classically approximated quantum-inspired feature embedding for each input, ready to be passed to the temporal transformer encoder layer. Following the work done by [26,28,30] this entire Algorithm 1 approximates a variational quantum circuit using classical operations, avoiding quantum hardware while retaining the rotation-based feature transformation benefits.

2.3. Dual-Attention Temporal Transformer Encoder

Algorithm 2 describes how the dual-attention temporal transformer model computes the final temporally rich hidden representations. This dual-attention temporal Transformer encoder works on the quantum-enhanced sequence ψz , which is generated by the quantum-inspired (classical approximation) amplitude-encoding layer. It processes this sequence through L stacked transformer blocks, with each block combining standard multi-head self-attention and a new temporal attention bias. The query, key and value projections follow standard multi-head self-attention formulation introduced in the transformer architecture [31] (Vaswani et al., 2017). Then, in each block, the input sequence is converted into query, key, and value projections:

$$Q = \psi W_Q, K = \psi W_K, V = \psi W_V \tag{8}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{2d \times d_k}$ are learnable projection matrices. The scaled dot-product attention mechanism $\frac{QK^T}{\sqrt{d_k}}$ is adopted from [31] (Vaswani et al., 2017) and augmented it with learnable temporal bias:

$$\text{Attention}_{\{temp\}}(Q, K, V, t) = \text{softmax}\left(\left(\frac{QK^T}{\sqrt{d_k}} + B_t\right)\right)V \tag{9}$$

where B_t is a learnable temporal bias matrix [32]:

Algorithm 1: Quantum-inspired (Classical Approximation) Encoding Layer

```

    •  $x \in R^{B \times d}$  (BERT embeddings,  $d = 768$ , batch size  $B$ )
Input:
    •  $nq = 8$  (number of "qubits") (Not real qubits but 2D vectors  $[q[0], q[1]]$ )
    •  $nl = 3$  (number of variational layers)
Output:
    •  $z \in R^{B \times d}$  (Quantum-inspired—classical approximated encoding)
 $x_n \leftarrow \frac{x}{\|x\|^2}$  // L2-normalize the BERT embedding
 $amp \leftarrow \text{Linear\_amp}(x_n)$  // 768  $\rightarrow$  16 (Dense layer) amplitude features
 $phs \leftarrow \text{Linear\_phase}(x_n)$  // 768  $\rightarrow$  16 (Dense layer) phase features
Reshape:  $amp, phs \in R^{(B \times nq \times 2)}$  // 16  $\rightarrow$  8 groups  $\times$  2D qubit like mapping
 $psi \leftarrow \text{normalize}(amp + 0.3 \times phs)$  // Weighted sum of amplitude and phase features

For  $i = 0$  to  $nq - 1$  do
     $q[i] \leftarrow \frac{(psi[:, i] + \text{roll}(psi[:, i], 1))}{\sqrt{2}}$  // Qubit initialization (2D vectors)
End For
For  $l = 0$  to  $nl - 1$  do // 3 layers
    For  $i = 0$  to  $nq - 1$  do // Rotations (single-qubit transformations)
        For  $k = 0$  to 2 do //  $k = 0, 1, 2$  represents  $R_x, R_y, R_z$  rotational angles
             $c \leftarrow \cos\left(\frac{\theta[l, i, k]}{2}\right), s \leftarrow \sin\left(\frac{\theta[l, i, k]}{2}\right)$  // similar to variational-transformation
             $q[i, 0] \leftarrow c \times q[i, 0] - s \times q[i, 1]$ 
             $q[i, 1] \leftarrow s \times q[i, 0] + c \times q[i, 1]$ 
        End For
    End For
    For  $i = 0$  to  $nq - 2$  do // CNOT Entanglement- classical approximation
         $p \leftarrow q[i, 0]^2$ 
         $temp \leftarrow q[i + 1]$ 
         $q[i + 1, 0] \leftarrow p \times temp[1] + (1 - p) \times temp[0]$ 
         $q[i + 1, 1] \leftarrow p \times temp[0] + (1 - p) \times temp[1]$ 
    End For
     $p \leftarrow q[0, 0]^2$  // Circular CNOT-Global-correlation (classical approximation)
     $temp \leftarrow q[nq - 1]$ 
     $q[nq - 1, 0] \leftarrow p \times temp[1] + (1 - p) \times temp[0]$ 
     $q[nq - 1, 1] \leftarrow p \times temp[0] + (1 - p) \times temp[1]$ 
End For
For  $i = 0$  to  $nq - 1$  do
     $pr[i] \leftarrow q[i, 0]^2 + 1e^{-9}$  // Classical probability
End For
 $pr \leftarrow \frac{pr}{\text{sum}(pr)}$  // Normalize across qubits (2D vectors)
 $H \leftarrow -\text{sum}(pr \times \log(pr + 1e^{-9}))$  // Shannon entropy
 $w \leftarrow \text{sigmoid}(\text{Linear\_ent}([H, 1 - H]))$  // Adaptive weights ( $B \times nq$ )
 $qm \leftarrow \text{stack}(q)$  //  $(B, nq, 2) = (B, 8, 2)$  qubit stack
 $qw \leftarrow qm \times w$  // entropy-weighted qubits, importance-weighting
 $v \leftarrow \text{concat}(\text{flat}(qm), \text{flat}(qw))$  //  $(B, 32) = 16 + 16$ , Feature concatenation
 $\psi z \leftarrow \text{LayerNorm}(\text{ReLU}(\text{Linear\_out}(v)))$  // output projection ( $32 \rightarrow 768$ )

```

$$B_{t[i,j]} = \beta \cdot \frac{|t_i - t_j|}{T_{\{max\}}} \tag{10}$$

scaling temporal distances by parameter β and maximum temporal span $T_{\{max\}}$ this encourages the model to weight contextually and temporal ordering is included using time-aware positional encoding (adopted from Kim and Lee, 2024 [32]). The usual sinusoidal positional encoding (from Vaswani et al., 2017 [31]) is improved by adding a learnable temporal component.

$$PE(pos, d) = \begin{cases} \sin\left(\frac{pos}{10000 \frac{2k}{d}}\right) + \gamma \cdot t & \text{for even } d \\ \cos\left(\frac{pos}{10000 \frac{2k}{d}}\right) + \gamma \cdot t & \text{for odd } d \end{cases} \tag{11}$$

where PE denotes positional encoding, pos represent token position in the sequence, d is the embedding dimension, γ is a learnable scalar that controls the strength of the temporal component, k is the dimension index and t is the temporal index of the token. Similarly, the dual-attention within each head (combined version of [31,32]) is then:

$$H_t = MultiHeadAttention(Q_t, K_t, V_t) \oplus MultiHeadAttention_{\{temp\}}(Q_t, K_t, V_t, t) \tag{12}$$

where \oplus denotes concatenation along the head dimension and the result is linearly projected to obtain the final block output:

$$Z(l) = LayerNorm(Linear(H_t; W_O))$$

where, W_O is the learnable matrix that combines and projects the concatenated dual-attention outputs back into the model-space representation before layer normalization. Through multiple stacked L layers, the encoder converts the quantum-enhanced sequence $\psi^{(0)} = \psi$ into the final hidden representation $Z = Z^{(L)} \in R^{T \times 2d}$, which is then sent to the next fusion layer.

2.4. Propagation Graph Attention Module

Propagation graph attention module is a separate standalone module to model propagation graphs. Algorithm 3 describes how the propagation graphs for our work are built. It models how misinformation spreads through social networks, capturing patterns invisible to text-only analysis. For datasets that include clear social network information, such as FakeNewsNet and PHEME, which contains Twitter follow relationships, we build the propagation graph $G = (U, E)$ directly from these connections. We use retweet, reply, and share edges (which are present within 4 h temporal windows [2,33,34]) to form the graph.

Algorithm 4 describes how propagation graph attention, neighbourhood node aggregation and global graph representations are computed using the heterogenous propagation graphs $G = (V, E)$ from Algorithm 3. In this algorithm, R represents the tensors of real numbers with varying shapes, depending on the computed representations. Using Algorithm 4, we model the spatio-temporal evolution of claims through social networks. We use graph attention layer operating on the earlier generated heterogeneous propagation graphs. We compute attention weights for neighboring users:

$$\alpha_{\{u,v\}} = \text{softmax}\left(\text{LeakyReLU}\left(a^{T[W_u \cdot h_u | W_v \cdot h_v | e_{\{time\}}]}\right)\right), \tag{13}$$

where h_u, h_v are hidden representations, $e_{\{time\}}$ encodes temporal delay, and a^T is a learnable attention vector. $e_{\{time\}}$ is computed as a learnable linear projection of min-max normalized timestamp difference $\Delta t_{u,v} = |t_u - t_v|$, with output dimension $d_t = 8$. The $e_{\{time\}}$ and Node update is defined as:

Algorithm 2: Dual Attention Temporal transformer encoder

- $\psi^{(0)} \in R^{T \times 2d}$ (quantum-enhanced sequence),
- T: sequence length, 2d: latent dimension.

Input:

- Layers L, heads h, per-head dim $d_k = \frac{d}{h}$.
- Temporal indices $t = [t_1, \dots, t_T]$.
- Learnable: $\beta, \gamma, W_Q, W_K, W_V \in R^{2d \times d_k}$,
- $W_O \in R^{2d \times 2d}$, layer-norm params.

Output:

- $Z \in R^{T \times 2d}$ (final hidden state—current module)

```

 $\psi^{(0)} \leftarrow$  quantum-enhanced sequence // input to encoder
For l = 1 to L do // iterate over L layer
   $\psi \leftarrow \psi^{l-1}$  // layer input
   $Z \leftarrow \psi$ 
  For h = 1 to h do // iterate over hidden h
     $Q_h \leftarrow \psi W_{Q_h} \in R^{T \times d_k}$  // head h, queries
     $K_h \leftarrow \psi W_{K_h} \in R^{T \times d_k}$  // head h, keys
     $V_h \leftarrow \psi W_{V_h} \in R^{T \times d_k}$  // head h, values
     $A_h \leftarrow \text{softmax}\left(\frac{Q_h K_h^T}{\sqrt{d_k}}\right)$  // standard attention score
     $Attn_h \leftarrow A_h V_h \in R^{T \times d_k}$  // standard attention output
    For i = 1 to T:
      For j = 1 to T:
         $B_{t[i,j]} = \beta \cdot \frac{|t_i - t_j|}{T_{\{max\}}}$  // temporal bias-based on time distance
      End for
    End for
     $A_h \leftarrow \text{softmax}\left(\frac{(Q_h K_h^T)}{\sqrt{d_k}} + B_t\right)$  // temporally biased attention score
     $Attn_{h,temp} \leftarrow A_h V_h \in R^{T \times d_k}$  // temporally biased attention output
     $H_h \leftarrow Attn_h \oplus Attn_{h,temp} \in R^{T \times 2d_k}$  // dual-attention per head
  End for
   $H \leftarrow \text{Concatenate}(H_1, H_2, \dots, H_h)$  // merge all heads
   $H \leftarrow \text{Linear}(H; W_O) \in R^{T \times 2d}$  // project back to model dimension
   $H \leftarrow \text{LayerNorm}(H)$  // layer normalization
   $\psi^{(1)} \leftarrow \text{LayerNorm}(Z + H)$  // residual + normalization (output of layer l)
End for

Return  $Z \leftarrow \psi^L \in R^{T \times 2d}$  // final hidden representation after L layers

```

$$e_{\{time\}} = W_{time} \cdot Norm(\Delta t_{u,v}) + b_{time}$$

$$h_u^{\{new\}} = \sigma \left(\sum_{v \in N(u)} \alpha_{\{u,v\}} \cdot W_v h_v \right), \tag{14}$$

where $N(u)$ is the neighborhood and σ is activation. This layer captures how different user groups spread information and identify characteristic propagation patterns of misinformation vs. real news (e.g., rapid spread vs. organic gradual dissemination).

Algorithm 3: Propagation Graph construction

```

    • Raw tweet data
Input:   D = {ti = (tweet, user, timestamp, retweets, reply and shares)}
    • Interaction_type I = {retweets, replies, shares}
    • Source tweet timestamp tsource
Output: • G = (V, E) nodes, edges
V ← {ti | ti ∈ D} // All tweets
E ← {} // empty edge set
For each Interaction_type do:
  For each edge e ∈ D.interactions do:
    Δte ←  $\frac{|timestamp_e - t_{source}|}{3600}$  // convert to hours
    if Δte ≤ 48 then:
      E ← E ∪ {(source(e), target(e))} // add directed edge
    End if
  End for
End for
Return G = (V, E) // 48 h propagation graph

```

2.5. Loss Function and Classification

The transformer output $Z \in \mathbb{R}^{T \times 2d}$ and graph representation $G_{rep} = \mathbb{R}^{d_g}$ are combined into a single vector by joining three complementary signals. Including the token-wise mean: $Mean(Z) = \frac{1}{T} \sum_{t=1}^T Z_t \in \mathbb{R}^{2d}$, which captures global semantic content. A graph representation G_{rep} that encodes propagation structure. A variance transformed by a Multi-Layer-Perceptron: $Var(Z) = \frac{1}{T} \sum_{t=1}^T (Z_t - Mean(Z))^2 \in \mathbb{R}^{2d}$. This variance captures the distributional spread across token representations. These three signals are combined together via: $H_{fused} = [Mean(Z); G_{rep}; MLP(Var(Z))] \in \mathbb{R}^{4d+d_g}$. A linear classification layer uses a weight matrix $W_c \in \mathbb{R}^{K \times (4d+d_g)}$ and bias $b_c \in \mathbb{R}^K$ to produce the final prediction. $\hat{y} = \text{softmax}(W_c \cdot H_{fused} + b_c) \in \mathbb{R}^K$, where $k=2$ for binary classification. Finally, the model is optimized using cross-entropy loss:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K y_{ik} \log(p_{ik}), \tag{15}$$

where $y_{ik} \in \{0, 1\}$ is the one-hot true label and p_{ik} is the predicted probability for class k . The variance of token representations, $Var(Z)$, is included as a complementary signal because it captures the dispersion and heterogeneity of token-level activations within a text sequence, which may reflect differences in semantic consistency, emphasis, or uncertainty. While the mean representation summarizes the overall semantic content, the variance provides additional information about how uniformly or unevenly the meaning is distributed across tokens.

Algorithm 4: Propagation graph attention computation

- Node features $G = (V, E)$
- Input:**
 - Adjacency matrix $A \in R^{\{B \times |U| \times |U|\}}$
 - Interaction timestamps $T \in R^{\{B \times |U| \times |U|\}}$
 - $h_u^{\{new\}} \in R^{\{B \times |U| \times d_g\}}$ (Node update representation)
- Output:**
 - $G_{rep} \in R^{\{B \times d_g\}}$ (Global graph representation)
 - $\alpha \in R^{\{B \times |U| \times |U|\}}$ (Attention weights–visualization)

$$Wh_u = h \cdot W_u^T \in R^{\{B \times |U| \times d_g\}} \quad // \text{ project source and target node feature}$$

$$Wh_v = h \cdot W_v^T \in R^{\{B \times |U| \times d_g\}} \quad // \text{ feature}$$

$$Wh_u^{\{exp\}[b,u,v,:]} = Wh_u[b,u,:]} \in R^{\{B \times |U| \times |U| \times d_g\}} \quad // \text{ Pairwise expansion}$$

$$Wh_v^{\{exp\}[b,u,v,:]} = Wh_v[b,v,:]} \in R^{\{B \times |U| \times |U| \times d_g\}} \quad // \text{ Pairwise expansion}$$

$$e_{\{time\}} = \tanh(W_t \cdot T_{\{uv\}} + b_t) \in R^{\{B \times |U| \times |U| \times d_t\}} \quad // \text{ Temporal Encoding}$$

$$c_{\{uv\}} = [Wh_u^{\{exp\}} \parallel Wh_v^{\{exp\}} \parallel e_{\{time\}}] \in R^{\{B \times |U| \times |U| \times (2d_g + d_t)\}} \quad // \text{ Edge feature Concatenation via matrix concatenation}$$

$$e_{\{u,v\}} = \text{LeakyReLU}(a^T \cdot c_{\{uv\}}) \in R^{\{B \times |U| \times |U|\}}, \quad // \text{ Attention logit computation (Equation (13))}$$

where $a \in R^{\{2d_g + d_t\}}$

$$e_{\{uv\}} = e_{\{uv\}} \cdot \text{masked}_{fill}(A_{\{uv\}} = 0, -\infty) \quad // \text{ Adjacency masking}$$

$$e_{\{time\}} = W_{time} \cdot \text{Norm}(\Delta t_{u,v}) + b_{time} \quad // e_{\{time\}} \text{ equation}$$

$$\alpha_{\{u,v\}} = \text{softmax}\left(\text{LeakyReLU}\left(a^T [W_u \cdot h_u | W_v \cdot h_v | e_{\{time\}}]\right)\right) \quad // \text{ Attention weight normalization (Equation (13))}$$

$$h_u^{\{new\}} = \sigma\left(\sum_{v \in N(u)} \alpha_{\{u,v\}} \cdot W_v h_v\right) \quad // \text{ Neighborhood aggregation (Equation (14))}$$

For $l = 2, \dots, L$ DO:

$$\left[\begin{array}{l} h^{(l)}, G_{rep}^{(l)}, \alpha^{(l)} = \text{TGAT}_{l(h^{(l-1)}, A, T)} \\ \text{End For} \end{array} \right. \quad // \text{ Layer stacking}$$

$$G_{rep} = \frac{1}{|U|} \left(\sum_{u=1}^{|U|} h_u^{\{new\}} \in R^{B \times d_g} \right) \quad // \text{ Global graph mean pooling}$$

Return G_{rep}, h_{new} and α

2.6. Datasets

Table 2 describes the summary of two benchmark datasets used to evaluate the proposed QuST-TF model. First, FakeNewsNet [33] includes about 23,200 news articles: 1056 from PolitiFact (432 fake, 624 real) and 22,140 from GossipCop (5323 fake, 16,817 real). It also provides tweet graphs, user interaction networks, and social engagement data. To avoid temporal leakage, we split FakeNewsNet [33] by time. Articles before 2018 were used for training, 2019 for validation, and 2020 onward for testing. Because of the 1:3 class imbalance, we created a balanced test set of 2000 articles, equally split between fake and real news. Next, PHEME [28] consists of 6425 rumor threads, 2402 misinformation threads, and 4023 non-misinformation threads, from nine real-world events. These include 105,354 tweets and veracity labels for each thread (true, false, unverified). For PHEME [28], we used leave-one-out event evaluation to avoid cross-event leakage. This resulted in a test set of about 1359 threads (708 misinformation, 651 non-misinformation). For both datasets, user IDs were anonymized, URLs were removed from content features, and publication timestamps were excluded from model inputs to prevent indirect leakage.

Table 2. FakeNewsNet and PHEME benchmark dataset description.

Property	FakeNewsNet	PHEME
Total Size	~23,200 news articles	6425 rumor threads
Sources	PolitiFact + GossipCop	9 real-world events
Fake/Rumor	5755 fake (432 + 5323)	2402 misinformation
Real/non-rumor	17,441 real (624 + 16,817)	4023 non-misinformation
Additional Data	Tweet graphs, user interaction networks, social engagement	105,354 tweets, veracity labels (true/false/unverified)
Class Ratio	1:3 (fake: real)	~1:1.7 (rumor: non-rumor)

2.7. Baselines and Comparison Models

We compare QuST-TF against Transformer + BiGRU [6]: State-of-the-art combining Transformer with bidirectional GRU. QEMF [8] Quantum multimodal fusion using VQC + QCNN (adapted for text-only by removing visual modality). DTN [17] Dynamic Temporal Network with time similarity metrics. RoBERTa [35] Pre-trained Transformer baseline. Bidirectional LSTM with attention (classical baseline) [36]. Finally, Temporal Tree Transformer (adapted from rumor detection) [7].

Table 3 presents a comparison of QuST-TF and six baseline models across five architectural dimensions: quantum-inspired (classical approximation) encoding, temporal modeling, graph propagation and pretrained language model usage.

Table 3. Baseline methods vs. Proposed model.

Model	Quantum Encoding	Temporal Modeling	Graph Propagation	Pretrained LM
Transformer + BiGRU [6]	✗	Partial	✗	✓
QEMF [8]	✓	✗	✗	✗
DTN [17]	✗	✓	Partial	✗
RoBERTa [33]	✗	✗	✗	✓
BiLSTM + Attention [28]	✗	Partial	✗	✗
Temporal Tree Transformer [7]	✗	✓	Partial	✗
QuST-TF (Proposed)	✓	✓	✓	✓

Comparison of QuST-TF against baseline models across five key architectural dimensions. ✓ = fully supported, Partial = limited support, ✗ = not supported.

2.8. Model Training and Implementations Details

Algorithm 5 shows the QuST-TF training process using 5-fold cross-validation with seeds on data split by time (70% training, 10% validation, 20% testing). For each seed, the BERT-base-uncased model (768-dimensional) tokenizes the input text, and the embeddings go through the full pipeline: quantum encoding, a 6-layer temporal Transformer, a 2-layer P-GAT (Propagation Graph Attention), fusion, and classification. We optimize with AdamW (learning rate 1×10^{-4} , weight decay 0.01) and a linear warmup scheduler for up to 50 epochs. Training uses mixed precision (GradScaler), gradient clipping (norm 1.0), and label smoothing (0.1). Each epoch switches between training (model.train() with forward and backward passes tracking cross-entropy loss and F1 score) and validation (model.eval() with no_grad). We save the best checkpoint when validation F1 improves and stop early after 10 epochs without improvement. The final test results per seed report the mean and standard deviation of accuracy and F1 across five runs, ensuring reliable, reproducible performance while reducing overfitting through careful validation.

Algorithm 5: QuST-TF Training with Cross-Validation

```

    • Dataset D,
Input:
    • seeds S = [1, 2, 3, 4, 5],
    • max_epochs = 50,
    • patience = 10
Output:
    • Model weights  $\theta_{best}$ ,
    • Performance metrics (mean  $\pm$  std)
For each seed  $s \in S$  do:
  Torch.manual_seed(s); np.random.seed(s)
  Split D  $\rightarrow$  D_train (70%), D_val (10%), D_test (20%)
  //Model: BERT  $\rightarrow$  Quantum encoding  $\rightarrow$  TemporalTF(6L) PGAT(2L)  $\rightarrow$  Fusion  $\rightarrow$  Classifier//
  bert_embed = Tokenization (model = 'bert_base_uncased', dim = 768)
  model = QuST_TF(bert_embed, d_model = 768)
  optimizer = AdamW(model.parameters(), lr =  $1 \times 10^{-4}$ , weight_decay = 0.01)
  scheduler = LinearWarmup(lr =  $1 \times 10^{-5} \rightarrow 1 \times 10^{-4}$ , 1000 steps)
  best_f1, patience =  $-\infty$ , 0
  For epoch = 1 to max_epochs do:
    //Training//
    model.train(); train_loss, train_f1 = 0, 0
    For batch in D_train do:
      text, graph_A, graph_T, labels = batch
      bert_emb = BERT(text.input_ids, attention_mask) # [B, T, 768]
      logits, ce_loss = model(bert_emb, graph_A, graph_T, labels)
      loss = F.cross_entropy(logits, labels, label_smoothing = 0.1)
      optimizer.zero_grad()
      (loss * scaler.scale(1.0)).backward()
      scaler.unscale_(optimizer)
      torch.nn.utils.clip_grad_norm_(model.parameters(), 1.0)
      scaler.step(optimizer)
      scaler.update()
      train_loss += loss.item()
      train_f1 += f1_score(logits, labels)
    End For
    //Validation//
    model.eval(); val_loss, val_f1 = 0, 0
    With torch.no_grad():
      For batch  $\in$  D_val:
        bert_emb = BERT (batch.text)
        logits, _ = model (bert_emb, batch.graph_A,
        batch.graph_T, batch.labels)
        val_loss += F.cross_entropy(logits, batch.labels)
        val_f1 += f1_score(logits, batch.labels)
      End For
    End With
    If loss_val < best_val_loss:
      best_val_loss = loss_val; patience_counter += 1
       $\theta_{best}^s = \theta$ .copy()
    Else: patience_counter += 1
    If patience_counter  $\geq$  10 Then Break
      scheduler.step()
    End For
  test_results[seed] = evaluate (model, D_test)
End For
Return mean  $\pm$  std accuracy and F1-score over 5 seeds

```

3. Results

3.1. Performance Analysis

Table 4 presents the mean and standard deviation over five runs, comparing QuST-TF with all baseline models on both FakeNewsNet and PHEME datasets. On FakeNewsNet, QuST-TF achieves 91.4% accuracy and 91.3% F1-score, outperforming Transformer + BiGRU [6] by 2.2 percentage points (89.2% F1), DTN [17] by 3.6 percentage points (87.7% F1), QEMF (text-only) [8] by 4.9 percentage points (86.4% F1), and RoBERTa [33] by 7.3 percentage points (84.0% F1). On PHEME, QuST-TF reaches its highest performance, with 95.5% accuracy and 95.2% F1-score. It surpasses TTT [7] by 9.3 percentage points (86.1% F1), Transformer + BiGRU [6] by 10.1 percentage points (84.7% F1), QEMF (text-only) [8] by 11.9 percentage points (83.3% F1), DTN [17] by 12.6 percentage points (82.6% F1), and BiLSTM + Attention [28] by 13.1 percentage points (82.3% F1). The larger performance margins observed on PHEME indicate that QuST-TF's quantum-inspired (classical approximation) encoding and graph propagation are particularly effective for shorter and noisier conversation threads. On FakeNewsNet, all models show higher precision than recall, reflecting a conservative classification approach that reduces false positives. QuST-TF also achieves the lowest standard deviation on both datasets (± 0.003), confirming stable, reproducible training across all runs. Overall, these results show that combining quantum encoding, temporal attention, and graph propagation yields consistent and significant improvements across all baseline models.

Table 4. Comparative Performance of Proposed QuST-TF work across FakeNewsNet and PHEME dataset.

Model Variant	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
QuST-TF (Proposed)	FakeNewsNet	91.4 ± 0.3	91.8 ± 0.4	90.9 ± 0.5	91.3 ± 0.3
Transformer + BiGRU [6]	FakeNewsNet	89.2 ± 0.5	89.6 ± 0.6	88.8 ± 0.7	89.2 ± 0.5
DTN [17]	FakeNewsNet	87.8 ± 0.5	88.4 ± 0.6	87.1 ± 0.8	87.7 ± 0.6
QEMF (text-only) [8]	FakeNewsNet	86.5 ± 0.6	87.0 ± 0.7	85.9 ± 0.8	86.4 ± 0.7
RoBERTa [33]	FakeNewsNet	84.1 ± 0.8	84.7 ± 0.9	83.4 ± 0.12	84.0 ± 0.8
QuST-TF (Proposed)	PHEME	95.5 ± 0.3	95.1 ± 0.3	95.2 ± 0.2	95.2 ± 0.3
TTT [7]	PHEME	86.2 ± 0.6	86.7 ± 0.7	85.6 ± 0.8	86.1 ± 0.6
DTN [17]	PHEME	82.9 ± 0.9	82.7 ± 0.7	84.9 ± 0.8	82.6 ± 0.8
Transformer + BiGRU [6]	PHEME	85.4 ± 0.4	84.3 ± 0.4	85.4 ± 0.5	84.7 ± 0.5
QEMF (text-only) [8]	PHEME	83.6 ± 0.9	83.4 ± 0.8	85.7 ± 0.9	83.3 ± 0.9
BiLSTM + Attention [28]	PHEME	82.4 ± 0.7	83.1 ± 0.8	81.6 ± 0.9	82.3 ± 0.7

3.2. Ablation Study Insights

Table 5 shows an ablation study that evaluates the individual contribution of each QuST-TF component using a leave-one-out analysis averaged over five random seeds. This study was conducted on FakeNewsNet, which is the more challenging of the two datasets. The full model achieves a lower performance of 91.4% on FakeNewsNet compared to 95.5% on PHEME. This difference makes it easier to observe how sensitive the model is to each component. The importance of each component is measured using the Component Contribution Index (CCI). A higher CCI value reflects a greater individual contribution to the overall performance of the model. It is mathematically measured as:

$$CCI_i = \frac{\Delta F1_i}{\max_j(\Delta F1_j)}, (i, j) \in \left\{ \begin{array}{l} \text{Quantum Encoding,} \\ \text{Graph attention,} \\ \text{Temporal attention} \end{array} \right\}$$

Table 5. Ablation study results, demonstrating the contribution of each component.

Model Variant	Accuracy (%)	F1-Score (%)	Trainable Params (M)	Δ Acc (%)	Δ F1 (%)	Synergistic Gain	CCI
Full QuST-TF + LoRA	91.4 ± 0.3	91.3 ± 0.3	28.5	—	—	1.3%	—
w/o Quantum Encoding	88.4 ± 0.5	88.3 ± 0.5	25.1	−3.0	−3.0	—	0.70
w/o Graph Attention	88.8 ± 0.5	88.7 ± 0.6	26.3	−2.6	−2.6	—	0.60
w/o Temporal Attention	89.2 ± 0.4	89.1 ± 0.5	27.8	−2.2	−2.2	—	0.51
Classical BERT 2× High-Dimensional Expansion (same LoRA setup)	83.9 ± 0.5	84.8 ± 0.5	2.8	−4.5	−4.5	—	—
Quantum-inspired Encoding (classical approximation only)	84.0 ± 0.8	83.9 ± 0.9	2.54	−7.4	−7.4	—	—

Removing Quantum Encoding leads to the largest single-component drop among the ablation settings, with accuracy decreasing from 91.4% to 88.4% and F1-score from 91.3% to 88.3% (CCI = 0.70). This confirms that quantum-inspired feature encoding is the most important component of the architecture. Removing Graph Attention causes a 2.6 percentage point decline (CCI = 0.60), while removing Temporal Attention results in a 2.2 percentage point drop (CCI = 0.51), indicating that both structural propagation and temporal dynamics contribute meaningfully, although less than quantum encoding. The Classical Transformer + BERT Backbone + LoRA baseline performs worse than the full QuST-TF model, showing that the proposed quantum-inspired representation provides additional benefit beyond a standard transformer backbone even under the same LoRA setting. The Classical BERT 2× High-Dimensional Expansion baseline achieves 83.9% accuracy, which is lower than both the full QuST-TF model and the encoding-only quantum-inspired baseline, demonstrating that simple dimensional expansion cannot replace the proposed quantum-inspired representation. Likewise, the Quantum-inspired Encoding (classical approximation) model reaches only 84.0% accuracy, which is still below that of the full QuST-TF model, showing that quantum-inspired encoding alone is insufficient without the temporal and graph modules. Overall, the full QuST-TF model achieves a synergistic gain of 1.3%, confirming that quantum encoding, graph attention, and temporal attention reinforce one another and produce better results together than any individual component alone. This gain is calculated as:

$$\text{Synergistic Gain} = \text{Acc}_{full} - \text{Acc}_{classical} - \max_i |\Delta \text{Acc}_i|$$

$$\text{Synergistic Gain} = 91.4 - 87.1 - 3.0 = +1.3\%$$

3.3. Parameter Efficiency

As shown in Table 6, QuST-TF employs BERT-base in feature-extraction mode, maintaining all 110 million BERT weights fixed during training. Only the modules built on quantum encoding projection, temporal transformer, graph attention, and fusion classifier are trained end-to-end, resulting in 28.5 million trainable parameters with LoRA (Low Rank Adaptation) fine-tuning ($r = 8$, $\alpha = 16$) [37]. This architecture is designed to achieve three primary advantages. First, freezing BERT prevents catastrophic forgetting of pretrained linguistic representations, which is particularly important for short, informal social media text. Second, it eliminates the need to backpropagate through 110 million parameters, thereby substantially reducing training time. Third, it confines trainable components to task-specific modules, thereby clarifying and simplifying the interpretation of ablation results. Regarding efficiency, QuST-TF attains 91.4% accuracy with only 28.5 million trainable parameters. All baselines in Table 6 utilize the same LoRA [37] configuration ($r = 8$, $\alpha = 16$)

with frozen BERT to ensure fair comparison. In contrast, Transformer + BiGRU [6] requires 43.2 million trainable parameters to achieve 89.2% accuracy, while DTN [17] requires 41.5 million, QEMF [8] requires 39.3 million, and RoBERTa requires 75.0 million trainable parameters. Similarly, TTT [7] requires 42 million parameters to achieve $86.2 \pm 0.6\%$ accuracy, and BiLSTM + Attention [28] require 35.1 million parameters. Therefore, QuST-TF [33] achieves a 2.2% accuracy improvement while using 34% fewer trainable parameters than the closest baseline (Transformer + BiGRU) [6]. These results indicate that QuST-TF is a practical and parameter-efficient alternative to standard transformer models without compromising performance.

Table 6. Parameter efficiency test proposed vs baselines (Low-rank adaptation fine-tuning).

Model Variant	LoRA Config	Trainable Parameters (M)	Accuracy (%)	F1-Score (%)	Parameter Reduction
QuST-TF (Proposed)	$r = 8, \alpha = 16$	28.5	91.4 ± 0.3	91.3 ± 0.3	34% fewer
Transformer + BiGRU [6]	$r = 8, \alpha = 16$	43.2	89.2 ± 0.5	89.2 ± 0.5	Baseline
DTN [17]	$r = 8, \alpha = 16$	41.5	87.8 ± 0.5	87.7 ± 0.6	−4%
QEMF (text-only) [8]	$r = 8, \alpha = 16$	39.3	86.5 ± 0.6	86.4 ± 0.7	−9%
RoBERTa [33]	$r = 8, \alpha = 16$	75.0	84.1 ± 0.8	84.0 ± 0.8	−62%
TTT [7]	$r = 8, \alpha = 16$	42.0	86.2 ± 0.6	86.1 ± 0.6	−3%
BiLSTM + Attention [28]	$r = 8, \alpha = 16$	35.1	82.4 ± 0.7	82.3 ± 0.7	−18%

3.4. Error Analysis of Proposed Model

Table 7 presents a full diagnostic evaluation of QuST-TF's detection performance, extending beyond standard metrics. It reports Sensitivity, Specificity, AUC-ROC, and MCC for all models on both datasets, using a consistent five-seed evaluation protocol.

Table 7. Sensitivity vs. Specificity, AUC-ROC Analysis.

Model	Dataset	Sensitivity (%)	Specificity (%)	AUC-ROC (%)	MCC
QuST-TF (Proposed)	FakeNewsNet	90.9 ± 0.5	91.9 ± 0.4	95.4 ± 0.3	0.828 ± 0.006
Transformer + BiGRU [6]	FakeNewsNet	88.8 ± 0.7	89.6 ± 0.6	93.1 ± 0.5	0.784 ± 0.008
DTN [17]	FakeNewsNet	87.1 ± 0.8	88.5 ± 0.7	92.1 ± 0.5	0.756 ± 0.009
QEMF (text-only) [8]	FakeNewsNet	85.9 ± 0.8	87.2 ± 0.7	91.0 ± 0.5	0.731 ± 0.010
RoBERTa [33]	FakeNewsNet	83.4 ± 0.12	84.8 ± 0.9	88.6 ± 0.7	0.682 ± 0.012
QuST-TF (Proposed)	PHEME	95.2 ± 0.2	95.8 ± 0.3	98.1 ± 0.2	0.908 ± 0.004
TTT [7]	PHEME	85.6 ± 0.8	86.9 ± 0.7	91.3 ± 0.5	0.725 ± 0.009
DTN [17]	PHEME	85.4 ± 0.5	86.1 ± 0.5	90.5 ± 0.5	0.712 ± 0.009
Transformer + BiGRU [6]	PHEME	85.7 ± 0.9	83.1 ± 0.8	89.1 ± 0.7	0.674 ± 0.010
QEMF (text-only) [8]	PHEME	84.9 ± 0.8	82.4 ± 0.7	88.3 ± 0.7	0.663 ± 0.010
BiLSTM + Attention [28]	PHEME	81.6 ± 0.9	83.2 ± 0.8	87.5 ± 0.7	0.648 ± 0.012

On FakeNewsNet, QuST-TF achieves a Sensitivity of 90.9% and a Specificity of 91.9%. This means the model correctly detects 90.9% of fake articles and accurately identifies 91.9% of real ones. Most baseline models do not achieve this balance, as their sensitivity and specificity values differ more widely. The AUC-ROC of 95.4% indicates strong class separation across all classification thresholds, not only at the default 0.5 cutoff. The MCC of 0.828, which considers all entries in the confusion matrix and is robust to class imbalance, further supports the reliability of these results. On PHEME, QuST-TF performs even better, reaching its highest Sensitivity of 95.2%, Specificity of 95.8%, AUC-ROC of 98.1%, and MCC of 0.908. This suggests that the model's quantum-inspired (classical approximation) encoding and graph propagation are particularly effective for shorter and noisier thread structures. The improvement over the closest PHEME baseline, TTT, is clear: QuST-TF

outperforms by 6.8 percentage points in AUC-ROC (98.1% compared to 91.3%) and by 0.183 in MCC (0.908 compared to 0.725). In both datasets, Specificity is about 1 percentage point higher than Sensitivity, showing that QuST-TF tends to avoid false positives and is less likely to incorrectly label real content as misinformation.

Table 8 describes these rates as raw counts from the test sets. In FakeNewsNet (2000 balanced articles), 91 fake articles were missed (FN), and 81 real articles were wrongly flagged (FP). In PHEME (1359 threads), 79 pieces of misinformation went undetected, and 66 non-misinformation were wrongly classified as misinformation. The higher number of false negatives compared to false positives is consistent across both datasets. This matches the Sensitivity being lower than Specificity in Table 7 and shows the model's conservative bias is systematic, not specific to a dataset.

Table 8. Confusion Matrix—FakeNewsNet vs. PHEME.

Metrics	FakeNewsNet	PHEME
Test size	2000	1359
Label dist.	1000: Real 1000: Fake	708: Misinformation 651: non-Misinformation
True Positive	909 (90.9%)	629 (88.8%)
False Negative	91 (9.1%)	79 (11.2%)
False Positive	81 (8.1%)	66 (10.1%)
True Negative	919 (91.9%)	585 (89.8%)

4. Discussion

4.1. Attention-Based Visualizations

Figure 2 shows that the QuST-TF framework uses both behavioral and linguistic features to make its predictions. In the “Fake” sample (gossipcop-8566320362; $p = 0.9415$), the temporal heatmap (Figure 2a) displays irregular, high-intensity hotspots. This pattern matches the rapid, bursty spread often seen in viral, fabricated content. The token attention plot (Figure 2b) supports this by highlighting words with strong emotional or sensational impact. In contrast, the “Real” sample (gossipcop-872861; $p = 0.0049$) shows a more even and stable attention pattern over time (Figure 2c), typical of verified news with steady, organic engagement. Here, the model focuses on specific, high-value named entities such as “Musk,” “Tesla,” and “Elon” (Figure 2d). These entities provide a factual anchor for the classification. Together, these visualizations show that the framework examines both how information spreads and what is being said. The clear difference between the erratic spread and sensational language of misinformation and the stable patterns and factual references of real news demonstrates that QuST-TF's decisions are based on meaningful, interpretable signals. The model's high-confidence results rely on these clear propagation and semantic features, showing its strength in separating real from synthetic information.

4.2. Limitations and Future Works

QuST-TF consistently improves results on both benchmarks, but there are some key limitations to note. We used frozen BERT-base in feature-extraction mode during training, which helps prevent catastrophic forgetting on limited social media data but restricts domain-specific semantic adaptation. The quantum-inspired (classical approximation) amplitude encoding is a classical simulation, following current hybrid quantum-classical NLP approaches, since quantum hardware at the scale needed for text processing is not available to us.

The ablation results ($\Delta F1 = -3.0\%$) reflect the encoding's mathematical design rather than any quantum speedup, as we clearly explained. While the current architecture uses a

classical simulation of quantum-inspired (classical approximation) amplitude encoding, purely quantum NLP methods, where linguistic structure is encoded and processed entirely within a quantum circuit, offer a more fundamental approach that should be developed and tested in future research.

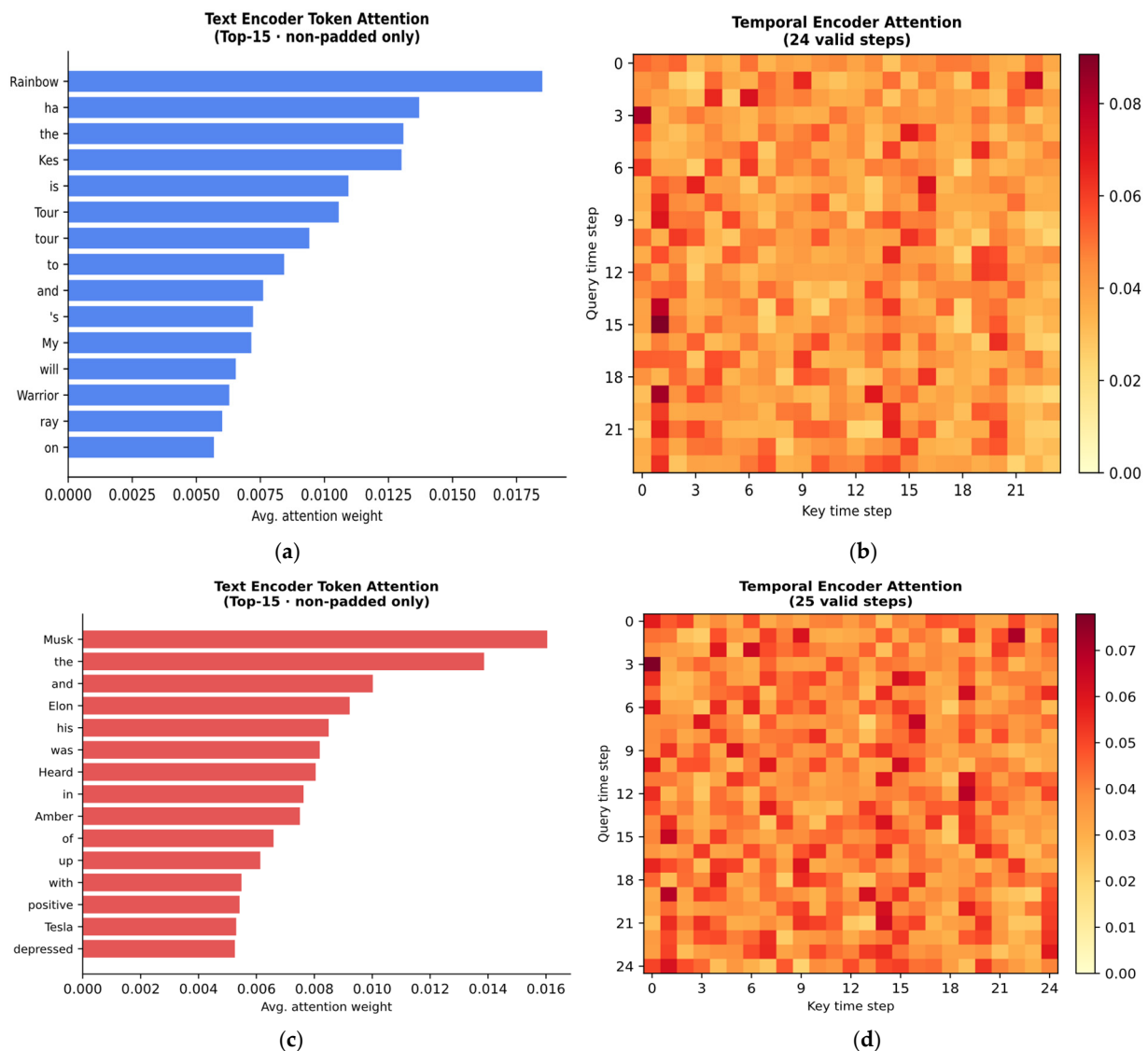


Figure 2. Attention-Based Comparison of Misinformation and Real News. (a) shows the top 15 most influential non-padded tokens for the article gossipcop-8566320362 (Fake, $p = 0.9415$). The importance scores are calculated from the attention weights of the BERT [CLS] token in the final encoder layer. This highlights which lexical features contribute most to the model’s classification decision. (b) shows the temporal encoder attention for gossipcop-8566320362 (Fake, $p = 0.9415$). The heatmap illustrates self-attention across 24-time steps; darker regions indicate stronger temporal dependencies, highlighting the model’s focus on propagation sequences characteristic of misinformation. (c) shows the token-attention distribution from the text encoder for gossipcop-872861. The top 15 non-padded tokens with the highest influence are listed for this verified article (Real, $p = 0.0049$). The model assigns greater importance to specific entities such as ‘Musk,’ ‘Tesla,’ and ‘Elon.’ These tokens act as reliable markers that help the model distinguish this article from common misinformation examples. (d) shows the temporal encoder attention for gossipcop-872861. The heatmap displays attention values across 25 valid time steps for the verified article (Real, $p = 0.0049$). The attention is distributed evenly across the sequence, indicating steady and consistent propagation that aligns with typical patterns of authentic news dissemination.

Our evaluation only covers two English-language benchmarks; because BERT performs worse on morphologically rich languages, we cannot assume it will generalize across languages. Adding multilingual models like XLM-R is an essential next step. The chronological and event-based splits partly simulate deployment conditions, but misinformation framing strategies keep changing in ways that yearly temporal splits cannot fully capture. The current framework is evaluated on English datasets and assumes the availability of a propagation structure derived from user interaction graphs. While this setting is suitable for studying structural and temporal rumor patterns, it is less representative of very early rumor detection, where propagation chains are sparse or unavailable. In such scenarios, the benefit of graph-based modeling may decrease, and performance may depend more heavily on content representations. We therefore consider robustness under missing or partial propagation information, as well as multilingual evaluation, to be important directions for future work. Finally, the graph attention module requires observable propagation structure during inference. For early-stage breaking content under a fixed 48 h temporal window, where the propagation graph is sparse, the model relies primarily on its text-only branch. In this setting, the ablation result without Graph Attention (88.8%) provides a lower-bound estimate of performance.

5. Conclusions

QuST-TF is a misinformation detection model integrating three components: quantum-inspired (classical approximation) amplitude encoding to capture richer semantic details, time-aware Transformer attention to monitor content evolution over time, and graph attention to model the propagation of posts on social networks. Previous quantum-inspired models primarily addressed multimodal inputs, temporal data, and propagation-aware models separately, with few employing quantum-inspired (classical approximation) amplitude encoding. In contrast, the proposed model combines these three approaches to effectively capture misinformation patterns, including misinformation and misinformation.

On FakeNewsNet, the model reached 91.4% accuracy and 95.4% AUC-ROC. On PHEME, it scored 95.5% accuracy and 95.2% AUC-ROC, consistently beating all baselines over five runs. Ablation analysis shows that quantum encoding is the most important component (CCI = 1.00, $\Delta F1 = -3.0\%$), with a combined gain of 1.3% across all three components. With just 28.5 million trainable parameters on a frozen BERT backbone, the model proves that strong detection is possible without full fine-tuning, making it practical for limited-resource settings. Visualizing attention weights also helps explain results for institutional fact-checking. Furthermore, the performance gains in QuST-TF arise from quantum-inspired (classical approximation) nonlinear angular modulation implemented via classical approximation, not from genuine quantum computing. The cos/sin rotation-based feature transformation captures directional semantic patterns (text structure) more effectively than magnitude-based attention, resulting in consistent accuracy. All computations are performed on classical CPUs/GPUs with NO quantum hardware or quantum circuits required.

Key limitations include the fully frozen BERT backbone, which restricts domain adaptation. Classical simulation of quantum encoding means gains come from rotation-based math, not real quantum speedup. The model is evaluated only in English. It also depends on observable propagation graphs during inference (ablation score: 88.8% as a sparse-graph lower bound). Future work includes implementing amplitude encoding as a parameterized quantum circuit on NISQ hardware (Noisy Intermediate-Scale Quantum developed by IBM Quantum, Armonk, USA; IBM Quantum backend Heron r3), multilingual evaluation with XLM-R or mBERT, and graph imputation for sparse early-stage propagation. Longer-term

plans involve quantum-inspired (classical approximation) multimodal fusion and online learning to handle evolving misinformation narratives.

Author Contributions: K.K. wrote the original draft, developed the methodology, and conducted all the experiments, which involved conceptualizing the study, designing the experimental framework, implementing the model, generated visualizations and formally analyzed the data. A.V. supervised the project, reviewed the manuscript, identified potential technical errors, provided feedback and corrections, and offered guidance and expertise throughout the experimental phase to ensure the rigor and validity of the research. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported and funded by Research Grant No. SPG/2020/000594 under the SERB POWER grant scheme, Science and Engineering Research Board, Government of India., to Akila Venkatesan, Pondicherry Engineering College, India.

Informed Consent Statement: Not applicable.

Data Availability Statement: The FakeNewsNet and PHEME datasets used in this experiment are publicly available datasets, and they are available in the following links: 1 FakeNewsNet: GitHub—KaiDMML/FakeNewsNet: This is a dataset for misinformation detection research. GitHub. <https://github.com/kaidmml/fakenewsnet> (accessed on 3 May 2026); 2. PHEME: <https://doi.org/10.6084/m9.figshare.6392078>.

Acknowledgments: We express our gratitude to the researchers and organizations that provided public access to the benchmark datasets. We also thank the editorial team and reviewers at MDPI Applied Sciences for their valuable feedback. Additionally, we acknowledge the use of Grammarly for language refinement during manuscript preparation.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

QIE	Quantum-Inspired Encoder
DATT	Dynamic Attention Temporal Transformer
PGAM	Propagation Graph Attention Module
QEMF	Quantum-Entropy Multilayer Perceptron
DTN	Dynamic Temporal Network
TTT	2-level Hierarchical Tree Transformer
BiLSTM	Bidirectional Long Short-Term Memory
BiGRU	Bidirectional Gated Recurrent Unit
FNN	FakeNewsNet
RoBERTa	Robustly Optimized BERT Approach
Focal Loss	Gamma-weighted Cross Entropy
AMP	Automatic Mixed Precision
CLS	Classification Token Embedding
AUC	Area Under the Curve
ROC	Receiver Operating Characteristic
F1	Harmonic Mean of Precision and Recall
TP/FP/TN/FN	True/False Positive/Negative

References

1. Unlu, A.; Truong, S.; Sawhney, N.; Tammi, T. Setting the Misinformation Agenda: Modeling COVID-19 Narratives in Twitter Communities. *New Media Soc.* **2024**, *27*, 3973–3997. [[CrossRef](#)]
2. Singal, K.; Dahiya, Y.; Bharti, D.; Sood, A.; Bansal, P. Misinformation Spread in Social Media. In *Proceedings of the 2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions), ICRITO 2024, Noida, India, 14–15 March 2024*; IEEE: New York, NY, USA, 2024; pp. 1–6.

3. Plikynas, D.; Rizgeliene, I.; Korvel, G. Systematic Review of Fake News, Propaganda, and Disinformation: Examining Authors, Content, and Social Impact Through Machine Learning. *IEEE Access* **2025**, *13*, 17583–17629. [[CrossRef](#)]
4. Ali, Z.S.; Al-Ali, A.; Elsayed, T. Detecting Users Prone to Spread Fake News on Arabic Twitter. In *Proceedings of the 5th Workshop Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection, OSACT 2022—Proceedings at Language Resources and Evaluation Conference, LREC 2022*; European Language Resources Association: Paris, France, 2022; pp. 12–22.
5. Karande, H.; Walambe, R.; Benjamin, V.; Kotecha, K.; Raghu, T.S. Stance Detection with BERT Embeddings for Credibility Analysis of Information on Social Media. *PeerJ Comput. Sci.* **2021**, *7*, e467. [[CrossRef](#)] [[PubMed](#)]
6. Huang, T.; Xu, Z.; Yu, P.; Yi, J.; Xu, X. A Hybrid Transformer Model for Fake News Detection: Leveraging Bayesian Optimization and Bidirectional Recurrent Unit. In *Proceedings of the 2025 8th International Symposium on Big Data and Applied Statistics, ISBDAS 2025*; IEEE: New York, NY, USA, 2025; pp. 696–701.
7. Wu, S.; Deng, Y.; Liu, J.; Luo, X.; Sun, G. Rumor Detection on Social Networks Based on Temporal Tree Transformer. *PLoS ONE* **2025**, *20*, e0320333. [[CrossRef](#)] [[PubMed](#)]
8. Bikku, T.; Thota, S. Quantum-Enhanced Multimodal Fusion for Robust and Accurate Fake News Detection. *Sigma J. Eng. Nat. Sci.* **2025**, *43*, 943–954. [[CrossRef](#)]
9. Suneesh, A.; Palani, B. QCNN-MFND: A Novel Quantum CNN Framework for Multimodal Fake News Detection in Social Media. In *Proceedings of the QuantumNLP: Integrating Quantum Computing with Natural Language Processing*; ACL Anthology; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 44–52.
10. Altıntaş, V. Beyond Classical AI: Detecting Fake News with Hybrid Quantum Neural Networks. *Appl. Sci.* **2025**, *15*, 8300. [[CrossRef](#)]
11. Khalil, M.; Zhang, C.; Ye, Z.; Zhang, P. PegasosQSVM: A Quantum Machine Learning Approach for Accurate Fake News Detection. *Appl. Artif. Intell.* **2025**, *39*, 2457207. [[CrossRef](#)]
12. Pan, Y.; Jiang, H.; Chen, J.; Li, Y.; Zhao, H.; Zhao, L.; Abate, Y.; Wang, Y.; Liu, T. Bridging Classical and Quantum Computing for Next-Generation Language Models. *Proc. AAAI Symp. Ser.* **2025**, *7*, 381–389. [[CrossRef](#)]
13. Chen, X.; Lou, X. Enhancing Text Classification Through Quantum Transfer Learning: A Hybrid Quantum-Classical Approach With Complex Kernel Self-Attention Networks. *IEEE Access* **2025**, *13*, 133882–133890. [[CrossRef](#)]
14. Pal, P.; Das, D. Toward Quantum-Enhanced Natural Language Understanding: Sarcasm and Claim Detection with QLSTM. *Nat. Lang. Process. Gener. AI Era* **2025**, *2025*, 854–859. [[CrossRef](#)]
15. Gao, H.; Zhang, P.; Zhang, J.; Yang, C. QSIM: A Quantum-Inspired Hierarchical Semantic Interaction Model for Text Classification. *Neurocomputing* **2025**, *611*, 128658. [[CrossRef](#)]
16. Shanmugavadeivel, K.; Subramanian, M.; Vishali, K.S.; Priyanka, B.; Naveen Kumar, K. KEC_AI_DATA_DRIFTERS@DravidianLangTech 2025: Fake News Detection in Dravidian Languages. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 173–177.
17. Hu, J.; Zhang, J.; Li, Z. Tracing Truth: Dynamic Temporal Networks for Multi-Modal Fake News Detection. *PeerJ Comput. Sci.* **2025**, *11*, e2998. [[CrossRef](#)] [[PubMed](#)]
18. Peng, X.; Wu, J.; Liu, R.; Xu, K. Rumor Detection on Social Media with Temporal Propagation Structure Optimization. In *Proceedings—International Conference on Computational Linguistics, COLING*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 3865–3878.
19. Plepi, J.; Sakketou, F.; Geiß, H.J.; Flek, L. Temporal Graph Analysis of Misinformation Spreaders in Social Media. In *Proceedings—International Conference on Computational Linguistics, COLING*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022; Volume 29, pp. 89–104.
20. Aktar, S.; Bärtschi, A.; Badawy, A.-H.A.; Eidenben, S. Quantum Graph Transformer for NLP Sentiment Classification. In *2025 IEEE International Conference on Quantum Computing and Engineering (QCE)*; IEEE: New York, NY, USA, 2025; Volume 1.
21. Gruzdeva, A.S.; Iurev, R.N.; Bessmertny, I.A.; Khrennikov, A.Y.; Alodjants, A.P. A Quantum-like Approach to Semantic Text Classification. *Entropy* **2025**, *27*, 767. [[CrossRef](#)] [[PubMed](#)]
22. Aishwarya, C.; Venkatesan, M.; Prabhavathy, P.; Akanksha, D. Applying Multi-Modal Quantum Deep Learning Algorithms for Enhanced Fake News Detection. *Acta Phys. Pol. B* **2025**, *49*, 223–244. [[CrossRef](#)]
23. Bischof, L.; Teodoropol, S.; Fuchslin, R.M.; Stockinger, K. Hybrid Quantum Neural Networks Show Strongly Reduced Need for Free Parameters in Entity Matching. *Sci. Rep.* **2025**, *15*, 4318. [[CrossRef](#)] [[PubMed](#)]
24. Li, Y.C.; Zhang, Y.F.; Xu, R.Q.; Zhou, R.G.; Dong, Y.L. HQRNN-FD: A Hybrid Quantum Recurrent Neural Network for Fraud Detection. *Entropy* **2025**, *27*, 906. [[CrossRef](#)] [[PubMed](#)]
25. Mondal, T.; Dhar, D.; Lahiri, S.; Bandyopadhyay, S. Quantum-Infused Whisper: A Framework for Replacing Classical Components. In *Proceedings of the QuantumNLP: Integrating Quantum Computing with Natural Language Processing*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 1–5.
26. Schuld, M.; Sinayskiy, I.; Petruccione, F. An Introduction to Quantum Machine Learning. *arXiv* **2014**, arXiv:1409.3097. [[CrossRef](#)]

27. Steane, A.M. An Introduction to Spinors. *arXiv* **2013**, arXiv:1312.3824.
28. Tang, E. A Quantum-Inspired Classical Algorithm for Recommendation Systems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*; Association for Computing Machinery: New York, NY, USA, 2019. [[CrossRef](#)]
29. Nakaji, K.; Uno, S.; Suzuki, Y.; Raymond, R.; Onodera, T.; Tanaka, T.; Tezuka, H.; Mitsuda, N.; Yamamoto, N. Approximate Amplitude Encoding in Shallow Parameterized Quantum Circuits and Its Application to Financial Market Indicator. *Phys. Rev. Res.* **2022**, *4*, 023136. [[CrossRef](#)]
30. Young, K.; Scese, M.; Ebneenasir, A. Simulating Quantum Computations on Classical Machines: A Survey. *arXiv* **2023**, arXiv:2311.16505.
31. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30, pp. 6000–6010.
32. Kim, K.G.; Lee, B.T. Self Attention with Temporal Prior: Can We Learn More from Arrow of Time? *arXiv* **2024**, arXiv:2310.18932. [[CrossRef](#)]
33. Kwon, S.; Cha, M.; Jung, K. Rumor Detection over Varying Time Windows. *PLoS ONE* **2017**, *12*, e0168344. [[CrossRef](#)] [[PubMed](#)]
34. Del Vicario, M.; Quattrociocchi, W.; Scala, A.; Zollo, F. Polarization and Fake News: Early Warning of Potential Misinformation Targets. *ACM Trans. Web* **2019**, *13*, 1–22. [[CrossRef](#)]
35. Rout, J.; Mishra, M. Towards Reliable Fake News Detection: Enhanced Attention-Based Transformer Model. *J. Cybersecur. Priv.* **2025**, *5*, 43. [[CrossRef](#)]
36. Chen, J.; Zhang, T.; Yan, Z.; Zheng, Z.; Zhang, W.; Zhang, J. Attention-Based BiLSTM with Positional Embeddings for Fake Review Detection. *J. Big Data* **2025**, *12*, 83. [[CrossRef](#)]
37. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv* **2021**, arXiv:2106.09685.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.