Synergizing Laplacian Positional Encodings and Graph Transformers for Robust Network Intrusion Detection

1st Chengming Liu*
School of Cyber Science and Engineering
Zhengzhou University
Zhengzhou, China
cmliu@zzu.edu.cn

2nd Yongkang Yuan
School of Cyber Science and Engineering
Zhengzhou University
Zhengzhou, China
yyk22332015161@gs.zzu.edu.cn

***Abstract*—Network intrusion detection is critical for securing modern cyber infrastructure. However, traditional graph-based methods often struggle with limited receptive fields and over-smoothing issues. This paper proposes the Dual-stream Graph feature learning Network (Dual-Net), a structural-aware framework designed for robust global relational reasoning. The core of our method lies in the integration of Graph Transformers and Laplacian Positional Encodings (LapPE). The proposed framework comprises three main components: adaptive graph construction, dual-stream feature learning, and structural augmentation. First, an adaptive module is designed to extract high-confidence dependencies from raw traffic using k-NN estimation and semantic pruning. Second, we implement a dual-stream architecture to capture multi-scale features. In this architecture, Graph Transformers are employed to model long-range topological relationships. To enhance this process, LapPE is integrated into the Transformer stream. This encoding effectively captures the structural topology of the network and overcomes the permutation-invariance of standard Transformers. Finally, a dual-dimensional graph data augmentation strategy is implemented to improve model resilience against structural noise. Extensive evaluations on four benchmark datasets (KDD Cup '99, NSL-KDD, UNSW-NB15, and CICIDS2017) demonstrate that Dual-Net outperforms state-of-the-art baselines in both detection accuracy and robustness.**

***Index Terms*—Network Intrusion Detection, Graph Transformer, Laplacian Positional Encodings (LapPE), Dual-stream Learning, Structural Awareness.**

I. Introduction

The rapid evolution of 5G networks and the pervasive integration of Internet of Things (IoT) devices have fundamentally reshaped the digital landscape, leading to an unprecedented surge in network traffic and a corresponding escalation in cybersecurity threats [1]. As modern cyber-attacks become increasingly sophisticated, polymorphic, and persistent, traditional network intrusion detection systems (NIDS) based on static rules or simple statistical heuristics have become insufficient for protecting critical infrastructure [2]. Consequently, the research community has pivoted toward deep learning as a robust alternative for automating the identification of malicious patterns within high-dimensional data [3]. Early deep learning-based approaches, predominantly utilizing recurrent neural networks (RNN) and long short-Term memory (LSTM) networks, were designed to capture the temporal sequences of packet flows [4] [5]. Despite their success in processing flow-based data, these conventional deep learning paradigms often treat network traffic as independent tabular records or flattened sequences, thereby ignoring the rich topological relationships and communication dependencies that exist between interconnected hosts and services.

To address the loss of structural context, graph neural networks (GNN) have recently emerged as a powerful paradigm for NIDS by modeling network communication as graph structures [6]. Models such as graph convolutional networks (GCN) and graph attention networks (GAT) have demonstrated remarkable efficacy in aggregating local neighborhood information to identify structural anomalies and lateral movement within a network [7] [8]. However, traditional GNNs are primarily constrained by their reliance on local message-passing mechanisms, which can lead to the over-smoothing problem when multiple layers are stacked, effectively losing the discriminative global context necessary for detecting complex, distributed attack chains [9]. Furthermore, standard GNN architectures often lack a mechanism to capture long-range dependencies across the entire network graph, making them less effective at identifying coordinate attacks that span distal regions of the network topology.

The emergence of the Transformer architecture, characterized by its global self-attention mechanism [10], offers a promising solution to these limitations. By generalizing Transformers to graphs [11], researchers have begun to develop Graph Transformer models that allow every node to attend to every other node, regardless of their proximity in the graph structure [12] [13]. This global perspective is particularly advantageous for NIDS, where malicious actors may attempt to evade detection by distributing their activities across multiple subnets. Nevertheless, a significant challenge remains: because Transformers are inherently permutation-invariant, they cannot distinguish between different graph structures without the inclusion of structural or positional

*Chengming Liu is the corresponding author.

encodings. Recent advancements have highlighted the importance of Laplacian Positional Encodings (LapPE) in providing the model with a structural awareness of the underlying graph spectrum [14], thereby enabling a more nuanced representation of the network's geometry [15].

In this paper, we propose Dual-Net, a novel dual-stream framework that synergizes the strengths of local spatial aggregation and global relational reasoning for robust intrusion detection. Our architecture integrates a GCN-based stream for capturing fine-grained local connectivity and a Graph Transformer-based stream for modeling global communication dependencies. To further enhance the model's structural awareness, we incorporate LapPE, ensuring that the global attention mechanism is grounded in the actual topology of the network flows. The primary contributions of this work are summarized as follows:

- We propose the Dual-stream Graph feature learning Network, a framework that synergizes GCNs and Graph Transformers. This dual-stream design effectively captures both fine-grained local spatial features and global contextual dependencies, addressing the over-smoothing and limited receptive field issues in traditional models.
- We introduce a structural-aware enhancement mechanism based on LapPE. By incorporating spectral coordinates into the Transformer stream, this mechanism overcomes the permutation-invariance of standard Transformers and significantly improves the representation learning for heterophilous network traffic.
- We conduct extensive evaluations on four benchmark datasets (KDD Cup '99, NSL-KDD, UNSW-NB15, and CICIDS2017). The results demonstrate that Dual-Net consistently achieves superior detection accuracy and robustness compared to state-of-the-art baselines across various performance metrics.

II. Related Work

A. Sequential and Hybrid Models in NIDS

NIDS research has transitioned from basic architectures to complex hybrid systems [2]. To capture temporal dependencies in traffic flows, bidirectional LSTM and recurrent networks are widely utilized for anomaly detection [5] [16]. Recent hybrid models, such as DCNN-BiLSTM [17], Seq2Seq with ConvLSTM subnets [18], and CNN-GRU ensembles [19], enhance detection robustness by combining hierarchical spatial feature extraction with long-term memory capabilities.

B. Graph-based Representation Learning

Limitations in treating traffic as independent samples led to the rise of graph-based learning. Initial GNN models focused on spatial aggregation to identify suspicious connection patterns [6]. Inductive approaches like GraphSAGE generate embeddings for previously unseen IP addresses [20], while advanced embedding techniques map topologies into low-dimensional latent spaces for early and accurate detection [21]. Furthermore, self-supervised systems like Anomal-E [22] and robust interaction-based defenses [23] have unveiled the immense potential of graph features against adversarial evasion attempts.

C. Evolution of Graph Transformers

Graph Transformers overcome GNN locality bottlenecks by generalizing the self-attention mechanism [12]. Research has evolved from simple message passing to global spectral attention paradigms that rethink graph connectivity [15]. Frameworks like GTAE-IDS [24] and IoT-optimized Transformers [25]have successfully enabled real-time anomaly detection. To address traffic heterophily, recent works focus on learning advanced LapPE [14] and robust Transformer-based classification architectures [26]. Dual-Net builds upon these advancements by integrating structural awareness directly into a dual-stream architecture.

III. Methodology

A. Overview of Dual-Net

Dual-Net is a novel hybrid framework designed for accurate and robust network intrusion detection. It integrates data balancing, adaptive graph modeling, and dual-stream feature learning mechanisms to effectively capture local structural patterns, model long-range dependencies, and handle class imbalance in network traffic data. The framework consists of four main components, as shown in Fig. 1: (1) a pre-processing and SMOTE-Tomek [27] based data balancing module, (2) an adaptive graph construction block, (3) a GCN-based local feature extraction module, and (4) a Graph Transformer layer enhanced by LapPE for high-level representation and final attack classification. Finally, the fused representations from both streams are fed into a multi-layer perceptron (MLP) for attack classification.

B. Data Pre-processing and Class Balancing

To ensure the integrity of input features, raw network traffic undergoes a structured pre-processing pipeline. Initially, we perform missing value imputation and convert categorical attributes into numerical vectors via one-hot encoding. All features are then normalized through z-score standardization to eliminate scale-induced bias and ensure uniform feature ranges. To alleviate the inherent class imbalance in NIDS datasets, we implement a hybrid resampling strategy based on SMOTE-Tomek [27]. SMOTE is first utilized to synthesize minority samples for expanding decision boundaries. Subsequently, Tomek Links are employed to remove overlapping noise and clear inter-class boundaries. This balancing procedure is applied strictly to the training set. The validation and testing sets maintain their original distributions to simulate realistic network environments and evaluate the model's performance on rare attack classes.

C. Adaptive Graph Construction

The core of our approach lies in transforming isolated network flow vectors into a structured relational graph $G = (V, E)$. This transformation enables the model to exploit latent

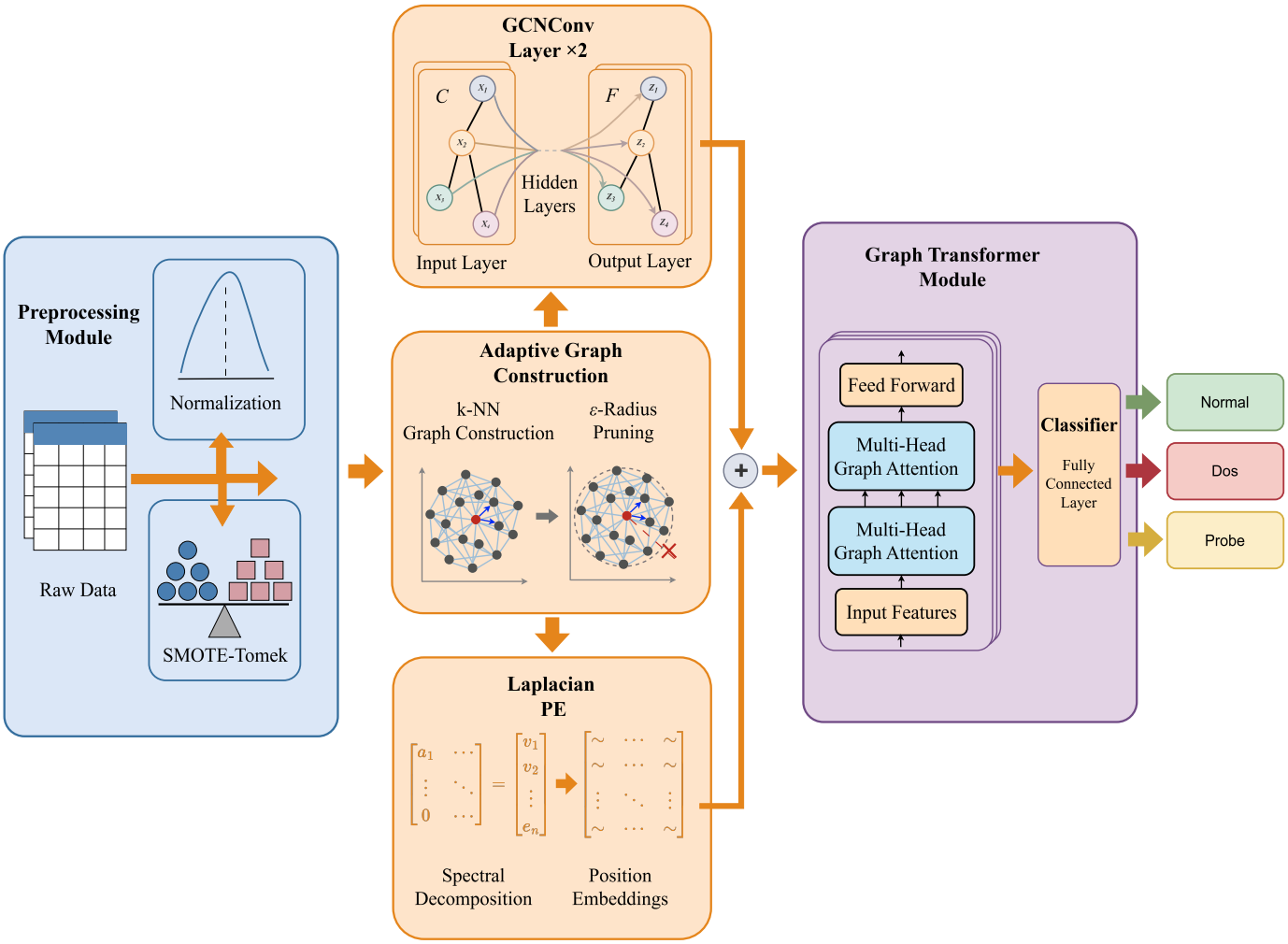


Fig. 1. Overall Architecture of the Proposed Dual-Net Integrating Adaptive Graph Construction and Graph Transformer.

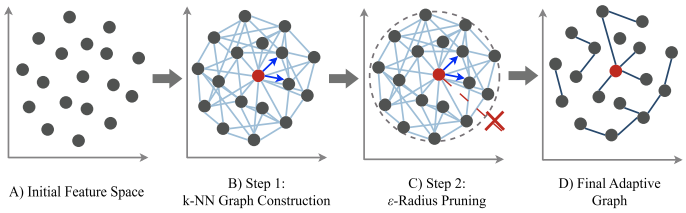


Fig. 2. Process of Adaptive Graph Construction.The nodes (black circles) represent individual network flows. The solid blue lines denote k-NN candidate edges representing latent correlations. The red dashed line with an 'X' indicates an edge pruned by the ϵ-distance threshold. The red node serves as the anchor node for the local topology estimation.

topological dependencies within the feature space. As illustrated in Fig. 2, the adaptive construction process comprises four critical stages.

First, the input feature space consists of N balanced vectors $X = \{x_1, \ldots, x_N\}$, where each $x_i \in \mathbb{R}^d$ represents a single flow. Initially, these samples are treated as independent nodes lacking structural information. Second, we perform k-NN topology estimation to establish local structures. For each anchor node v_i, a set of candidate neighbors $N_k(v_i)$ is identified based on the Euclidean distance $d(x_i, x_j) = \|x_i - x_j\|_2$. Potential correlations are established between v_i and its k closest neighbors to capture similarity-based clusters.

Third, we implement ϵ-radius semantic pruning to filter noise from numerically close but semantically distinct samples. The adjacency matrix A is refined by pruning edges that exceed a distance threshold ϵ:

$$A_{ij} = \begin{cases} 1, & \text{if } v_j \in N_k(v_i) \text{ and } d(x_i, x_j) \leq \epsilon \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Finally, this process generates an optimized, sparse, and semantically rich graph G, providing the structural foundation for subsequent message passing and global attention.

To enhance structural robustness, Dual-Net employs a dual-dimensional graph data augmentation strategy during training. We implement a stochastic perturbation mechanism to prevent over-fitting to specific to pological patterns. For structural augmentation, an edge addition perturbation is performed. We randomly sample a fraction ρ_e of existing edges and

concatenate them into the edge set to form an augmented set $\hat{E} = E \cup \{e \mid e \sim \text{Uniform}(E), \text{count} = \lfloor \rho_e \cdot |E| \rfloor\}$. This mechanism reinforces primary relational paths and ensures invariance to connection density.

Simultaneously, a node feature masking mechanism is integrated to bolster resilience against corrupted data. During each training pass, a subset of feature dimensions $\mathcal{M}$ is randomly masked with a rate ρ_f, setting $\hat{x}_{i,m} = 0$ if $m \in \mathcal{M}$. By compelling the model to extract information from partially occluded vectors, this dual-dimensional augmentation ensures that the resulting embeddings are robust to both structural noise and feature-level perturbations.

For real-time deployment, we adopt an inductive mini-batch strategy. Unseen flows are buffered, and their topologies are dynamically constructed via k-NN and pruning strictly within the batch. This enables efficient structural inference without historical dependencies.

D. Dual-Stream Feature Extraction and Global Modeling

Following adaptive graph construction, the Dual-Net architecture executes a dual-stream feature extraction phase. This stage captures multi-scale structural information by processing local neighborhood patterns and global topological dependencies in parallel.

In the spatial stream, a GCN module processes the refined graph G to encode local geometric structures. The core operation utilizes a localized message-passing mechanism. Each node v_i updates its hidden state $h_i^{(l)}$ by aggregating information from adjacent nodes $\mathcal{N}(i)$ via the following convolution rule:

$$h_i^{(l+1)} = \sigma\left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{1}{\sqrt{\tilde{d}_i \tilde{d}_j}} W^{(l)} h_j^{(l)}\right) \tag{2}$$

where $\tilde{d}$ denotes the degree matrix with self-loops, $W^{(l)}$ is the learnable weight matrix, and σ represents the ReLU activation function. This stream yields the GCN embeddings (H_{GCN}), serving as the foundational spatial representation.

Parallelly, the spectral stream employs LapPE to address the permutation-invariance and limited receptive fields of traditional GNNs. We first construct the normalized graph Laplacian matrix L. By performing eigen-decomposition, we derive a spectral coordinate system defined by eigenvectors U and eigenvalues Λ:

$$L = U \Lambda U^\top, \quad \Phi_i = \text{Linear}([u_{i,1}, u_{i,2}, \ldots, u_{i,k}]) \tag{3}$$

We select the k smallest non-trivial eigenvectors to form the LapPE embeddings Φ_i. These encodings assign unique positional signatures that preserve global structural context, enabling the perception of long-range dependencies invisible to local convolutions.

The Global Modeling Stage synergizes these features using a Graph Transformer. As detailed in Fig. 3, the GCN spatial features H_{GCN} and spectral positional encodings Φ_i are first projected and fused via an additive mechanism to initialize the Transformer input $x_i^{(0)}$:

$$x_i^{(0)} = \text{LayerNorm}(W_{f1} H_{GCN} + W_{f2} \Phi_i) \tag{4}$$

where W_{f1} and W_{f2} are projection matrices balancing local and global information. Subsequently, a Multi-Head Graph Attention mechanism calculates coefficients α_{ij} to weigh relational interactions:

$$\alpha_{ij} = \frac{\exp\left(\frac{(Qx_i)\cdot(Kx_j)^\top}{\sqrt{d_k}}\right)}{\sum_{k \in \mathcal{N}(i)} \exp\left(\frac{(Qx_i)\cdot(Kx_k)^\top}{\sqrt{d_k}}\right)} \tag{5}$$

This mechanism allows nodes to attend to global dependencies across distinct representation subspaces.

Finally, the representation undergoes refinement via a Position-wise Feed-Forward Network (FFN) and residual connections. Following the standard Add & Norm protocol shown in Fig. 3, the layer-wise update is formulated as:

$$\begin{aligned} x'_i &= \text{LayerNorm}(x_i^{(l)} + \text{Attn}(x_i^{(l)})) \\ h_i^{(l+1)} &= \text{LayerNorm}(x'_i + \text{FFN}(x'_i)) \end{aligned} \tag{6}$$

These distilled semantic embeddings integrate localized context with global positioning. A fully-connected layer with Softmax then determines the probability distribution of attack categories.

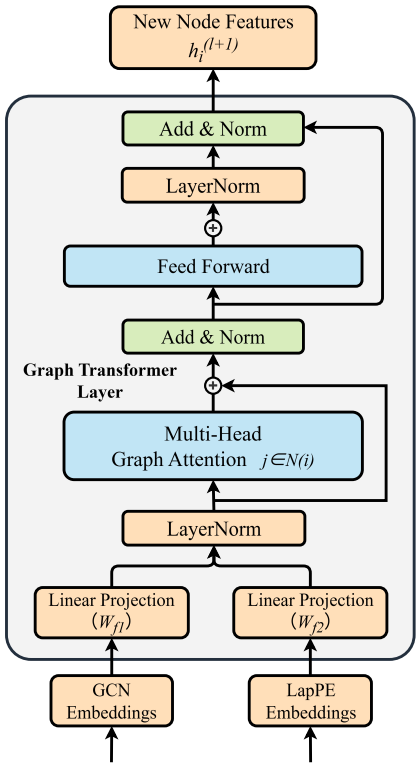


Fig. 3. Detailed Architecture of the Proposed Graph Transformer Layer Integrating GCN and Laplacian Positional Embeddings.

IV. Experiments and Analysis

A. Implementation Settings

Our experiments were conducted using PyTorch and PyTorch Geometric on an NVIDIA RTX 3090 GPU. To ensure robust evaluation, we established a standardized experimental pipeline. The datasets were partitioned into training, validation, and testing sets with a 60:20:20 ratio via stratified sampling to preserve class distributions. For data preparation, input features were normalized using StandardScaler, and the SMOTE-Tomek strategy was applied to the training set to resolve class imbalance.

The Dual-Net architecture was configured to balance complexity and performance. The spatial stream consists of two GCN layers with a hidden dimension of 64. The spectral stream utilizes LapPE with $k = 16$ eigenvectors to encode global structural information. The Graph Transformer layer is designed with 4 attention heads, each having a dimension of 32. To enhance generalization, we applied graph data augmentation during training, setting the edge perturbation rate ρ_e to 0.1 and the feature masking rate ρ_f to 0.2.

The training process was optimized using the Adam algorithm with an initial learning rate of 0.001, adjusted by a cosine annealing scheduler. We adopted the standard Cross-Entropy Loss for classification tasks. To ensure stable convergence and mitigate overfitting, an early stopping mechanism with a patience of 20 epochs was implemented, monitoring validation loss to save the optimal model weights.

B. Dataset Descriptions

We evaluate Dual-Net on four benchmarks: KDD Cup '99 [28], NSL-KDD [28], UNSW-NB15 [29], and CICIDS2017 [30]. While the former serve as classic baselines, we emphasize CICIDS2017 for representing modern, high-dimensional threat landscapes with complex protocols. Regarding task configuration, UNSW-NB15 is set for binary classification, whereas the other three datasets are configured for multi-class scenarios to assess granular detection capabilities. To ensure robust training across these skewed distributions, the SMOTE-Tomek resampling strategy is uniformly applied.

C. Evaluation Metrics

To rigorously evaluate the classification performance of Dual-Net, we employ four standard metrics: Accuracy, Precision, Recall, and F1-score. These metrics are derived from the fundamental counts of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN), which collectively represent the model's correct and incorrect predictions for both attack and benign network samples.

D. Performance Analysis and Comparison

We evaluate the performance of Dual-Net against five representative baselines, including GCN [7], GAT [8], GraphSAGE [20], RF [31], and XGBoost [32]. The experiments encompass four NIDS datasets, with UNSW-NB15 [29] configured for binary classification and others for multi-class scenarios. To comprehensively assess model capabilities, we present comparative results in two distinct settings: Table I details performance on standard datasets without augmentation, while Table II reports results under structural perturbation to validate the efficacy of our graph data augmentation strategy.

On standard benchmarks (Table I), Dual-Net consistently achieves state-of-the-art (SOTA) performance. Traditional models like RF [31] and XGBoost [32] exhibit limited capacity for processing high-dimensional relational traffic. For instance, their F1-scores drop below 0.50 on complex datasets such as NSL-KDD [28]. In contrast, Dual-Net effectively captures latent topological dependencies, establishing a significant performance lead.

The robustness of Dual-Net is further highlighted in the augmented scenarios (Table II). While baseline models suffer noticeable degradation under structural noise, Dual-Net maintains exceptional stability. The CICIDS2017 [30] dataset presents the most significant challenge. In this noisy environment, localized architectures encounter severe bottlenecks; specifically, the F1-score of GCN plummets to 0.6422. Conversely, Dual-Net sustains a superior F1-score of 0.9661. This resilience is attributed to the synergy between GCN-based local aggregation and Graph Transformer's global reasoning, which effectively compensates for structural distortions.

TABLE I
Performance Comparison Of Models Without Augmentation

Dataset	Module	Accuracy	Precision	Recall	F1
KDD Cup '99	**Dual-Net**	**0.9970**	0.9773	**0.9584**	**0.9650**
	GAT	0.9950	0.9802	0.9177	0.9422
	GCN	0.9891	0.8840	0.8975	0.8910
	GraphSAGE	0.9871	**0.9863**	0.6907	0.7728
	RF	0.8921	0.6830	0.5021	0.5419
	XGBoost	0.8901	0.4260	0.4030	0.4089
NSL-KDD	**Dual-Net**	**0.9910**	**0.9742**	**0.9733**	**0.9738**
	GCN	0.9710	0.9595	0.9673	0.9633
	GAT	0.9660	0.9481	0.9663	0.9566
	GraphSAGE	0.9380	0.9394	0.8961	0.9137
	RF	0.5650	0.4884	0.4670	0.4657
	XGBoost	0.5490	0.4566	0.4461	0.4450
CICIDS 2017	**Dual-Net**	**0.9840**	**0.9749**	**0.9633**	**0.9676**
	GAT	0.9660	0.7715	0.9286	0.8293
	GraphSAGE	0.9640	0.9195	0.7778	0.8127
	GCN	0.9600	0.7590	0.9263	0.8197
	RF	0.8194	0.4402	0.4345	0.4357
	XGBoost	0.8074	0.3221	0.4281	0.3390
UNSW -NB15	**Dual-Net**	**0.9925**	**0.9822**	**0.9838**	**0.9830**
	GraphSAGE	0.9920	0.9701	0.9954	0.9823
	GAT	0.9910	0.9666	0.9948	0.9801
	GCN	0.9890	0.9650	0.9869	0.9756
	RF	0.9830	0.7430	0.5867	0.6261
	XGBoost	0.9810	0.6934	0.6146	0.6433

E. Confusion Matrix Analysis

To assess the per-class reliability and diagnostic robustness of Dual-Net, we examine the normalized confusion matrices for NSL-KDD [28] and CICIDS2017 [30] in Fig. 7. These results reflect the model's performance under the proposed graph data augmentation strategy, highlighting its enhanced structural awareness.

TABLE II
PERFORMANCE COMPARISON OF MODELS WITH AUGMENTATION

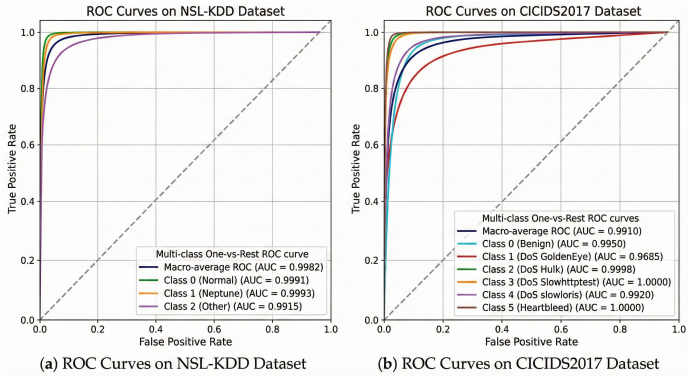
Dataset	Module	Accuracy	Precision	Recall	F1
KDD Cup '99	**Dual-Net**	**0.9980**	**0.9286**	**0.8687**	**0.8811**
	GCN	0.9900	0.9330	0.8498	0.8854
	GAT	0.9861	0.8721	0.8453	0.8543
	GraphSAGE	0.9821	0.9333	0.6575	0.7223
	RF	0.9069	0.5529	0.4931	0.5072
	XGBoost	0.8812	0.3520	0.4075	0.3723
NSL-KDD	**Dual-Net**	**0.9785**	**0.9735**	**0.9704**	**0.9719**
	GAT	0.9620	0.9477	0.9538	0.9506
	GCN	0.9610	0.9429	0.9598	0.9506
	GraphSAGE	0.9260	0.9325	0.8730	0.8952
	RF	0.6520	0.6342	0.5727	0.5859
	XGBoost	0.6520	0.6139	0.5771	0.5862
CICIDS 2017	**Dual-Net**	**0.9830**	**0.9745**	**0.9625**	**0.9661**
	GraphSAGE	0.9560	0.9189	0.7318	0.7712
	GAT	0.9550	0.7284	0.9047	0.7812
	GCN	0.9441	0.5859	0.7520	0.6422
	RF	0.8104	0.4782	0.4508	0.4590
	XGBoost	0.7874	0.3102	0.4233	0.3323
UNSW -NB15	**Dual-Net**	**0.9925**	**0.9762**	**0.9906**	**0.9833**
	GAT	0.9890	0.9705	0.9801	0.9752
	GraphSAGE	0.9860	0.9682	0.9682	0.9682
	RF	0.9820	0.7072	0.5862	0.6204
	XGBoost	0.9750	0.6184	0.6115	0.6149
	GCN	0.9720	0.9716	0.8990	0.9312

On the NSL-KDD dataset (Fig. 7a), Dual-Net achieves a 0.99 detection rate for both Normal traffic (Label 0) and the predominant Neptune attack category (Label 1). Notably, for the challenging minority samples represented by Other Attacks (Label 2), the model maintains a high recall of 0.91. This result validates the efficacy of our SMOTE-Tomek strategy in extracting representative features from sparse training data, ensuring balanced detection capabilities across major and minor attack vectors.

For the complex CICIDS2017 benchmark (Fig. 7b), which focuses on diverse Denial-of-Service (DoS) scenarios, Dual-Net demonstrates exceptional precision. The model achieves near-perfect classification scores ranging from 0.94 to 1.00 for four out of five attack categories, including DoS Hulk, DoS Slowhttptest, DoS slowloris, and Heartbleed. Although DoS GoldenEye (Label 1) exhibits slight confusion with the Benign category, the model still preserves a robust recall of 0.83. These findings confirm that Dual-Net retains high discriminative power and stability, effectively distinguishing between subtle DoS variants and benign traffic in heterogeneous network environments.

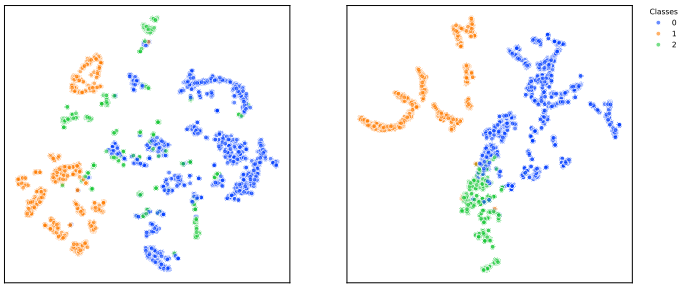
F. ROC-AUC Analysis

To quantify discriminative capability under varying thresholds, we analyze the ROC curves in Fig. 4. The curves exhibit a steep ascent toward the top-left corner, indicating high sensitivity with minimal false alarms. Quantitatively, Dual-Net achieves exceptional macro-average AUCs of 0.9982 on NSL-KDD and 0.9910 on CICIDS2017. Notably, the model demonstrates robust generalization by maintaining a 0.9915 AUC on the heterogeneous "Other" class (Class 2) of NSL-KDD and achieving perfect discrimination (AUC=1.0000) on specific vectors like DoS Slowhttptest (Class 3) in CICIDS2017. These results confirm that the learned embeddings possess strong linear separability, ensuring reliable detection even under stringent operational constraints.



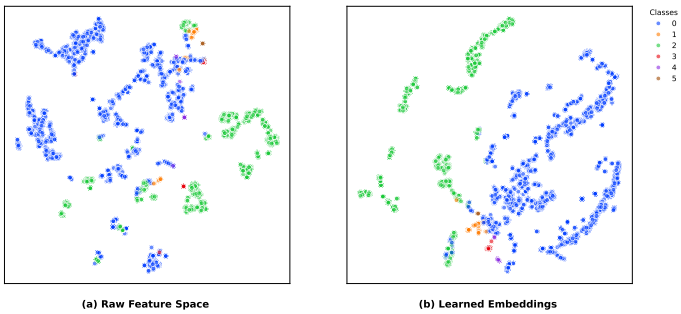
(a) ROC Curves on NSL-KDD Dataset (b) ROC Curves on CICIDS2017 Dataset

Fig. 4. Multi-class ROC analysis of the proposed Dual-Net. (a) ROC curves and specific AUC scores on the NSL-KDD dataset, covering Normal(Class 0), Neptune(Class 1), and Other categories(Class 2). (b) ROC curves on the CICIDS2017 dataset, illustrating performance across six distinct traffic classes. The steep ascent of the curves towards the top-left corner demonstrates the model's high sensitivity and low false positive rate.



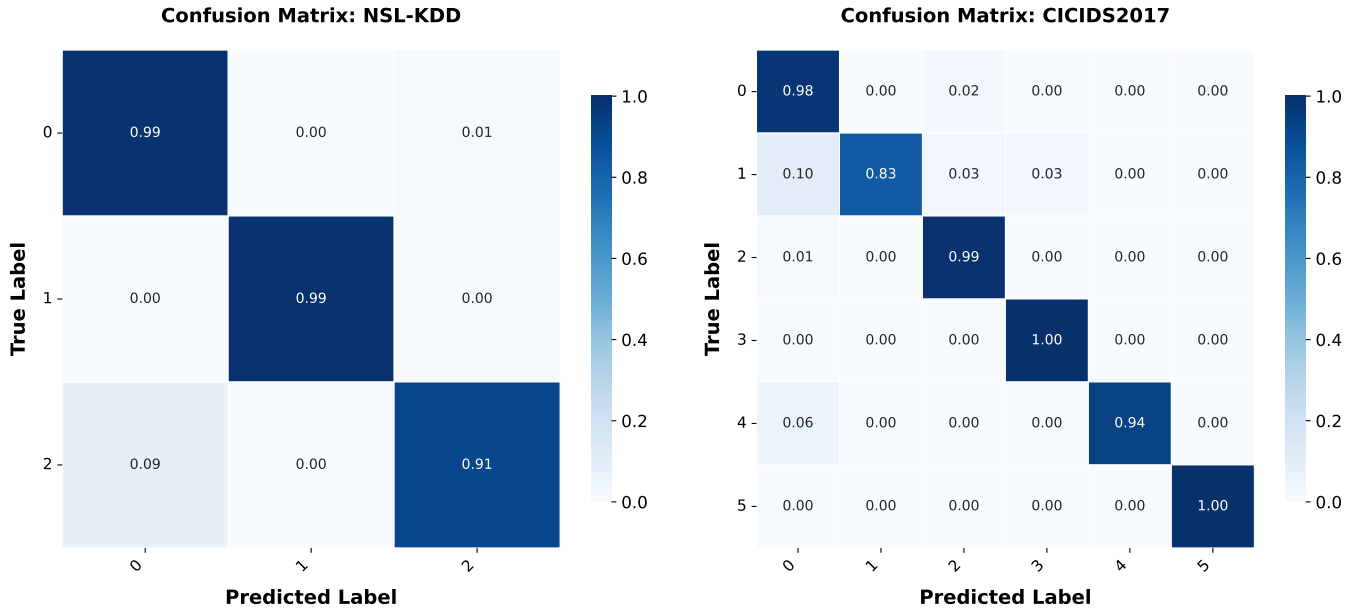
(a) Raw Feature Space (b) Learned Embeddings

Fig. 5. t-SNE visualization of traffic samples on the NSL-KDD dataset: (a) raw feature space, and (b) learned embeddings of Dual-Net.



(a) Raw Feature Space (b) Learned Embeddings

Fig. 6. t-SNE visualization of traffic samples on the CICIDS2017 dataset: (a) raw feature space, and (b) learned embeddings of Dual-Net.



(a) Confusion Matrix on NSL-KDD Dataset

(b) Confusion Matrix on CICIDS2017 Dataset

Fig. 7. Confusion Matrices of the Proposed Model on Different Datasets.

G. Detailed Performance Analysis

To validate the representation learning capability of Dual-Net, we employ t-SNE to visualize the transformation from raw feature spaces to learned embedding spaces. As illustrated in Fig. 5a for the NSL-KDD dataset [28], the raw features of Normal traffic (Label 0) and attack categories (Labels 1-2) exhibit severe overlap. This ambiguity poses significant challenges for linear decision boundaries. Conversely, Fig. 5b demonstrates that Dual-Net maps these samples into distinct, well-separated clusters. The learned embeddings exhibit strong intra-class compactness, validating the model's ability to disentangle complex relationships.

Similarly, the CICIDS2017 dataset [30] presents a highly entangled distribution in Fig. 6a, reflecting the noise inherent in modern attack scenarios (Labels 0-5). After processing via the GCN and Graph Transformer streams, these samples are reorganized into structured manifolds as shown in Fig. 6b. This evolution confirms that Dual-Net effectively extracts discriminative features from noisy data. The clear separation underscores the model's robustness and sensitivity in handling heterogeneous network environments.

H. Ablation Study

To quantify the contribution of each core component in Dual-Net, we conduct a systematic ablation study. We design two variants for comparison: (1) w/o LapPE, which removes the Laplacian Positional Encodings, and (2) w/o Transformer, which eliminates the global reasoning stream. As illustrated in Fig. 8, the full Dual-Net architecture consistently outperforms both variants across all benchmarks, validating the necessity of the proposed dual-stream design. The impact of the global

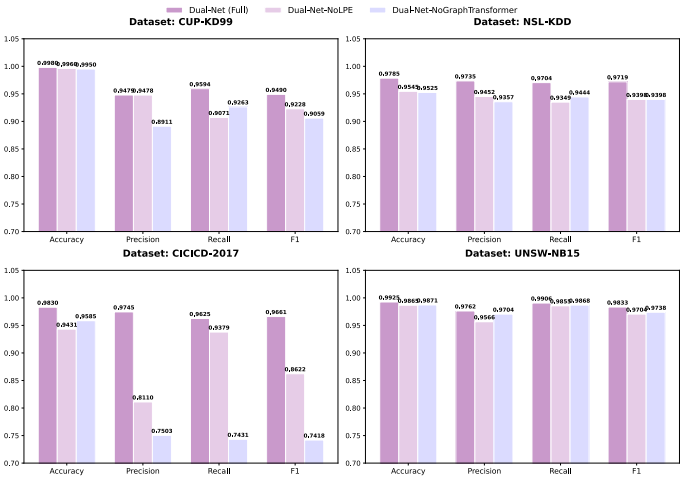


Fig. 8. Ablation study of Dual-Net performance across four benchmark datasets.

stream is most pronounced on the complex CICIDS2017 dataset [30]. Removing the Graph Transformer causes the F1-score to plummet from 0.9661 to 0.7418. This substantial degradation confirms that modeling long-range dependencies is indispensable for detecting sophisticated modern attack patterns. Similarly, the exclusion of LapPE leads to a notable performance drop on NSL-KDD [28], where the F1-score decreases from 0.9719 to 0.9398. This result validates that structural priors encoded by LapPE are essential for distinguishing spatially similar but semantically distinct traffic flows.

V. Discussion

The superior performance of Dual-Net stems from synergizing GCN-based local spatial aggregation with Transformer-driven global relational reasoning, ensuring robust structural awareness against semantic noise. Despite successfully capturing both fine-grained anomalies and distal attack chains, two limitations remain. First, the quadratic complexity of global self-attention requires an inductive mini-batch strategy to maintain near-real-time analysis; future work may adopt linear attention mechanisms like Performer to further alleviate this overhead. Second, the static initialization of our adaptive graph construction currently restricts the dynamic refinement of relational dependencies during backpropagation.

VI. Conclusion

In this paper, we proposed Dual-Net, a structural-aware framework addressing the receptive field limitations of traditional GNNs in network intrusion detection. By synergizing adaptive graph construction, Laplacian Positional Encodings, and a dual-stream mechanism, Dual-Net effectively captures both local features and long-range topological dependencies. Evaluations on four benchmark datasets confirm its superior accuracy and noise robustness compared to state-of-the-art baselines. Future work will explore self-supervised learning and lightweight optimizations for real-time deployment in high-speed networks.

VII. Acknowledgment

This work was supported in part by the the Nature Science Foundation of China (62006210, 62206252), the Key Project of Public Benefit in Henan Province of China (201300210500) and the Key R&D Program of Henan(251111212100).

References

[1] M. Alazab, K. Soman, S. Srinivasan, S. Venkatraman, V. Q. Pham *et al.*, "Deep learning for cyber security applications: A comprehensive survey," *Authorea Preprints*, 2023.

[2] J. Lansky, S. Ali, M. Mohammadi, M. K. Majeed, S. H. T. Karim, S. Rashidi, M. Hosseinzadeh, and A. M. Rahmani, "Deep learning-based intrusion detection systems: a systematic review," *IEEE Access*, vol. 9, pp. 101 574–101 599, 2021.

[3] A. Aldweesh, A. Derhab, and A. Z. Emam, "Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues," *Knowledge-Based Systems*, vol. 189, p. 105124, 2020.

[4] B. Lindemann, B. Maschler, N. Sahlab, and M. Weyrich, "A survey on anomaly detection for technical systems using lstm networks," *Computers in Industry*, vol. 131, p. 103498, 2021.

[5] S. M. Kasongo, "A deep learning technique for intrusion detection system using a recurrent neural networks based framework," *Computer Communications*, vol. 199, pp. 113–125, 2023.

[6] M. Zhong, M. Lin, C. Zhang, and Z. Xu, "A survey on graph neural networks for intrusion detection systems: Methods, trends and challenges," *Computers & Security*, vol. 141, p. 103821, 2024.

[7] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *CoRR*, vol. abs/1609.02907, 2016. [Online]. Available: http://arxiv.org/abs/1609.02907

[8] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[9] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" *arXiv preprint arXiv:1810.00826*, 2018.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[11] V. P. Dwivedi and X. Bresson, "A generalization of transformer networks to graphs," *CoRR*, vol. abs/2012.09699, 2020. [Online]. Available: https://arxiv.org/abs/2012.09699

[12] A. Shehzad, F. Xia, S. Abid, C. Peng, S. Yu, D. Zhang, and K. Verspoor, "Graph transformers: A survey," *arXiv preprint arXiv:2407.09777*, 2024.

[13] S. Yun, M. Jeong, R. Kim, J. Kang, and H. J. Kim, "Graph transformer networks," *Advances in neural information processing systems*, vol. 32, 2019.

[14] M. Ito, J. Zhu, D. Chen, D. Koutra, and J. Wiens, "Learning laplacian positional encodings for heterophilous graphs," *arXiv preprint arXiv:2504.20430*, 2025.

[15] D. Kreuzer, D. Beaini, W. Hamilton, V. Létourneau, and P. Tossou, "Rethinking graph transformers with spectral attention," *Advances in Neural Information Processing Systems*, vol. 34, pp. 21 618–21 629, 2021.

[16] Y. Imrana, Y. Xiang, L. Ali, and Z. Abdul-Rauf, "A bidirectional lstm deep learning approach for intrusion detection," *Expert Systems with Applications*, vol. 185, p. 115524, 2021.

[17] V. Hnamte and J. Hussain, "Dcnnbilstm: An efficient hybrid deep learning-based intrusion detection system," *Telematics and Informatics Reports*, vol. 10, p. 100053, 2023.

[18] S. Hariharan, Y. A. Jerusha, G. Suganeshwari, S. S. Ibrahim, U. Tupakula, and V. Varadharajan, "A hybrid deep learning model for network intrusion detection system using seq2seq and convlstm-subnets," *IEEE Access*, 2025.

[19] B. Cao, C. Li, Y. Song, Y. Qin, and C. Chen, "Network intrusion detection model based on cnn and gru," *Applied Sciences*, vol. 12, no. 9, p. 4184, 2022.

[20] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[21] X. Hu, W. Gao, G. Cheng, R. Li, Y. Zhou, and H. Wu, "Toward early and accurate network intrusion detection using graph embedding," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 5817–5831, 2023.

[22] E. Caville, W. W. Lo, S. Layeghy, and M. Portmann, "Anomal-e: A self-supervised network intrusion detection system based on graph neural networks," *Knowledge-based systems*, vol. 258, p. 110030, 2022.

[23] D. Pujol-Perich, J. Suárez-Varela, A. Cabellos-Aparicio, and P. Barlet-Ros, "Unveiling the potential of graph neural networks for robust intrusion detection," *ACM SIGMETRICS Performance Evaluation Review*, vol. 49, no. 4, pp. 111–117, 2022.

[24] J. Ghadermazi, S. Hore, A. Shah, and N. D. Bastian, "Gtae-ids: Graph transformer-based autoencoder framework for real-time network intrusion detection," *IEEE Transactions on Information Forensics and Security*, 2025.

[25] U. C. Akuthota and L. Bhargava, "Transformer based intrusion detection for iot networks," *IEEE Internet of Things Journal*, 2025.

[26] Z. Wu, H. Zhang, P. Wang, and Z. Sun, "Rtids: A robust transformer-based approach for intrusion detection system," *IEEE Access*, vol. 10, pp. 64 375–64 387, 2022.

[27] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[28] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE symposium on computational intelligence for security and defense applications*. Ieee, 2009, pp. 1–6.

[29] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 military communications and information systems conference (MilCIS)*. IEEE, 2015, pp. 1–6.

[30] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani *et al.*, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." *ICISSp*, vol. 1, no. 2018, pp. 108–116, 2018.

[31] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[32] T. Chen, "Xgboost: A scalable tree boosting system," *Cornell University*, 2016.