ADAGAT: ADAPTIVE GRAPH ATTENTION LEARNING FOR TIME-SERIES ANOMALY DETECTION UNDER TEMPORAL DISTRIBUTION SHIFTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Urban environments continuously generate massive volumes of spatio-temporal data through transportation systems, surveillance infrastructure, mobile devices, and connected sensors. Timely and reliable anomaly detection from these evolving data streams is crucial for identifying unexpected events that deviate from normal spatio-temporal patterns. Detecting anomalies such as traffic congestion, road accidents, crime incidents, abnormal crowd gatherings, and infrastructure failures is essential for improving public safety, emergency response, infrastructure resilience, resource allocation and sustainable urban management. However, accurate anomaly detection remains challenging due to evolving temporal patterns, changing graph structures, and distribution shifts that significantly degrade the performance of conventional Graph Neural Networks (GNNs). Existing state-of-the-art approaches primarily assume stationary graph distributions or fail to adapt their graph representations to continuously changing environments. To address these challenges, we propose AdaGAT, an adaptive graph attention learning framework for robust time-series anomaly detection under temporal distribution shifts. AdaGAT explicitly estimates distribution shifts between consecutive graph snapshots and dynamically adapts graph attention to learn robust graph representations from evolving temporal data. Evaluated on three benchmark datasets, AdaGAT achieves state-of-the-art F1-scores of 0.9901 on NF-ToN-IoT-V2 and 0.9998 on CIC-ToN-IoT-V2, while maintaining competitive performance on the more challenging NF-BoT-IoT-V2 benchmark. Furthermore, controlled distribution-shift experiments demonstrate that AdaGAT exhibits graceful degradation under progressively increasing temporal and structural shifts, retaining 70.94% of its original detection performance under severe out-of-distribution conditions. These results demonstrate the effectiveness of adaptive graph attention and distribution-shift-aware neighborhood aggregation for robust anomaly detection in evolving graph environments.

Keywords: Graph Neural Networks, Distribution Shift Learning, Out-of-Distribution Graph Learning, Graph Anomaly Detection, Dynamic Graph Representation Learning

1 INTRODUCTION

Graph-structured time-series data are increasingly encountered in modern urban environments, including intelligent transportation systems, surveillance networks, cyber-physical infrastructures, communication systems, and smart city sensing platforms. In these dynamic environments, anomalies are rare spatio-temporal events that deviate from normal structural or interaction patterns, often indicating abnormal system behavior or critical incidents. Detecting anomalies such as traffic congestion, road accidents, crime hotspots, abnormal crowd movements, infrastructure failures, cyber-physical attacks, sensor faults, and emergency events is essential for improving public safety, resource allocation, and urban sustainability. Owing to their ability to jointly model node attributes and relational dependencies, Graph Neural Networks (GNNs) have become the dominant paradigm for graph representation learning and anomaly detection, with architectures such as Graph Attention Networks (GATs) (Veličković et al., 2018), GraphSAGE (Hamilton et al., 2017), EvolveGCN

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

(Pareja et al., 2020), and recent spatio-temporal graph learning models (Jin et al., 2024; Nguyen et al., 2025) demonstrating remarkable success across a wide range of graph learning tasks. Despite these advances, most existing graph learning methods are developed under the assumption that training and testing data are sampled from the same underlying graph distribution. In real-world applications, however, graph systems evolve continuously due to changes in topology, node attributes, interaction patterns, and environmental conditions. Such non-stationary behaviour introduces *temporal distribution shifts*, causing representations learned from historical observations to become progressively less effective during deployment. Recent studies have explored dynamic graph learning, graph out-of-distribution (OOD) generalization, graph domain adaptation, and continual graph learning to improve robustness under evolving environments (Zhang et al., 2026a; Wu et al., 2024). Several approaches (Jin et al., 2024; Nguyen et al., 2025) have investigated adaptive graph learning and distribution-aware representation learning, including DIDA (Zhang et al., 2022), AdaMix (Chen et al., 2026), STRAP (Zhang et al., 2025), DyCIL (Zhang et al., 2026b), DRAMA (Kim & Shin, 2026), and HEM (Tian et al., 2025). Meanwhile, anomaly detection under distribution shifts has also received increasing attention in both graph and time-series settings (Cao et al., 2023; Gao et al., 2023; Carvalho et al., 2024; Darban et al., 2024). Nevertheless, existing methods generally focus on either temporal dependency modeling or distribution adaptation, while relatively few (Jin et al., 2024; Nguyen et al., 2025) explicitly integrate both within a unified graph attention framework. Furthermore, most attention-based models employ fixed neighborhood aggregation strategies that remain insensitive to changes in the underlying graph distribution.

To address these limitations, we propose **AdaGAT**, an adaptive graph attention learning framework for time-series anomaly detection under temporal distribution shifts. Instead of assuming a stationary graph environment, **AdaGAT** explicitly estimates the distribution discrepancy between consecutive graph snapshots and dynamically modulates neighborhood aggregation according to the detected graph evolution. The proposed architecture integrates relative temporal encoding, edge-aware feature learning, shift-aware graph attention, sparse α-Entmax neighborhood selection, and residual multi-head message passing into a unified end-to-end framework. By coupling distribution shift estimation with adaptive attention, **AdaGAT** learns graph representations that remain robust under evolving graph environments while effectively suppressing noisy neighborhood interactions. We evaluate **AdaGAT** on benchmark datasets, namely CIC-ToN-IoT v2, NF-ToN-IoT v2, and NF-BoT-IoT v2, using both in-distribution and temporal out-of-distribution evaluation protocols. Extensive experiments demonstrate that the proposed framework consistently achieves competitive anomaly detection performance while exhibiting improved robustness under controlled temporal distribution shifts.

2 Problem Formulation

We model the evolving environment as a sequence of dynamic graph snapshots

$$\mathcal{G} = \{G^1, G^2, \ldots, G^T\}, \tag{1}$$

where each graph snapshot is represented as

$$G^t = \left(\mathcal{V}^t, \mathcal{E}^t, \mathbf{X}^t, \mathbf{E}^t\right), \tag{2}$$

with $\mathcal{V}^t$ and $\mathcal{E}^t$ denoting the node and edge sets, while $\mathbf{X}^t \in \mathbb{R}^{N_t \times F}$ and $\mathbf{E}^t \in \mathbb{R}^{M_t \times D}$ represent the node and edge feature matrices, respectively.

Unlike conventional graph learning, the underlying graph distribution evolves over time. Let

$$P_t(\mathbf{X}, \mathcal{E}) \tag{3}$$

denote the joint distribution of graph features and topology at time t. A temporal distribution shift occurs whenever

$$P_t(\mathbf{X}, \mathcal{E}) \neq P_{t+1}(\mathbf{X}, \mathcal{E}), \tag{4}$$

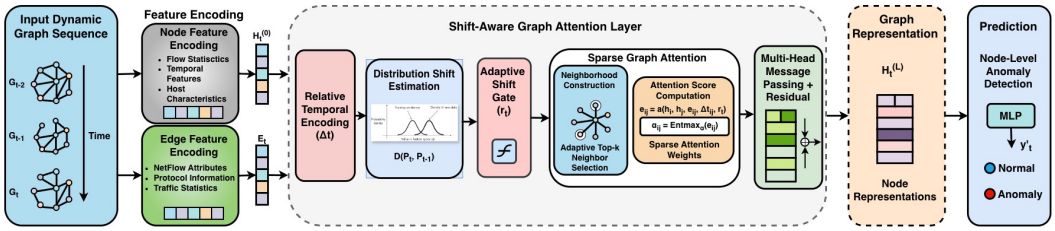

Figure 1: AdaGAT Model Architecture

causing representations learned under historical observations to become suboptimal for future graph snapshots.

Given the historical graph sequence

$$\mathcal{H}_t = \{G^{t-L+1}, \ldots, G^t\}, \tag{5}$$

our objective is to learn a parameterized function

$$f_\theta : \mathcal{H}_t \to \hat{\mathbf{Y}}^{t+1}, \tag{6}$$

that predicts the anomaly probabilities

$$\hat{\mathbf{Y}}^{t+1} = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{N_{t+1}}], \tag{7}$$

for all nodes in the next graph snapshot while remaining robust to temporal distribution shifts.

The model parameters are obtained by minimizing the unified objective

$$\theta^* = \arg\min_\theta \mathcal{L}_{\text{total}}, \tag{8}$$

where $\mathcal{L}_{\text{total}}$ jointly optimizes anomaly classification and distribution-shift adaptation. The complete objective function is introduced in Section 3.

3 Proposed Adaptive Graph Attention Learning Framework

Figure 1 presents the overall architecture of the proposed **Adaptive Graph Attention Learning (AdaGAT)** framework for anomaly detection under temporal distribution shifts. Given a sequence of dynamic graph snapshots, AdaGAT jointly encodes node features, edge attributes, and temporal information to construct graph representations. The framework then estimates distribution shifts between consecutive snapshots and uses this information to adaptively modulate the graph attention mechanism, enabling dynamic neighborhood aggregation under evolving graph distributions. Sparse attention and Top-k neighborhood selection suppress noisy interactions while preserving informative structural dependencies, and residual graph attention layers learn robust node embeddings for anomaly prediction. By explicitly integrating distribution shift estimation with adaptive graph attention, AdaGAT achieves robust representation learning in continuously evolving environments.

3.1 Dynamic Graph Representation Learning

Given a graph snapshot $G^t = (\mathcal{V}^t, \mathcal{E}^t, \mathbf{X}^t, \mathbf{E}^t)$, AdaGAT first constructs latent representations by jointly encoding node attributes, temporal information, and edge semantics. The node features are projected into a shared latent space

$$\mathbf{z}_i = \sigma(\mathbf{W}_n \mathbf{x}_i + \mathbf{b}_n), \tag{9}$$

where $\sigma(\cdot)$ denotes the ReLU activation (Nair & Hinton, 2010).

For each interaction (v_j, v_i), the relative temporal interval

$$\Delta\tau_{ji} = \tau_i - \tau_j \tag{10}$$

is encoded using

$$\mathbf{t}_{ji} = \phi_\tau(\log(1 + \Delta\tau_{ji})), \tag{11}$$

while edge attributes are embedded as

$$\mathbf{g}_{ji} = \phi_e(\mathbf{e}_{ji}). \tag{12}$$

The three representations are fused into a unified message

$$\mathbf{u}_{ji} = \sigma\left(\mathbf{W}_u[\mathbf{z}_j \| \mathbf{t}_{ji} \| \mathbf{g}_{ji}] + \mathbf{b}_u\right), \tag{13}$$

yielding the latent graph representation

$$\mathcal{Z}^t = \{\mathbf{z}, \mathbf{u}\}, \tag{14}$$

which is subsequently used for adaptive distribution shift estimation.

3.2 Adaptive Distribution Shift Learning

To explicitly model non-stationary graph environments, AdaGAT estimates the discrepancy between consecutive graph snapshots using the squared Maximum Mean Discrepancy (MMD)

$$D_t = \mathrm{MMD}^2(\mathcal{Z}^t, \mathcal{Z}^{t-1}), \tag{15}$$

computed as

$$\begin{aligned} D_t = &\frac{1}{n(n-1)} \sum_{i \neq j} k(\mathbf{z}_i^t, \mathbf{z}_j^t) \\ &+ \frac{1}{m(m-1)} \sum_{i \neq j} k(\mathbf{z}_i^{t-1}, \mathbf{z}_j^{t-1}) \\ &- \frac{2}{nm} \sum_{i,j} k(\mathbf{z}_i^t, \mathbf{z}_j^{t-1}), \end{aligned} \tag{16}$$

where $k(\cdot, \cdot)$ is an RBF kernel.

To stabilize optimization, the shift statistics are normalized using exponentially updated running estimates

$$\mu_t = (1 - \beta)\mu_{t-1} + \beta D_t, \tag{17}$$

$$v_t = (1 - \beta)v_{t-1} + \beta(D_t - \mu_t)^2, \tag{18}$$

which produce the normalized shift

$$\widetilde{D}_t = \frac{D_t - \mu_t}{\sqrt{v_t + \epsilon}}. \tag{19}$$

Algorithm 1 Adaptive Distribution Shift Learning

Require: Consecutive graph snapshots G^{t-1} and G^t
1: Encode node, temporal, and edge features to obtain $\mathcal{Z}^{t-1}$ and $\mathcal{Z}^t$ ▷ Learn latent graph embeddings
2: Estimate temporal distribution shift $D_t = \mathrm{MMD}^2(\mathcal{Z}^t, \mathcal{Z}^{t-1})$ ▷ Measure graph distribution discrepancy
3: Update running statistics (μ_t, v_t) ▷ Maintain stable shift estimation
4: Compute normalized shift $\widetilde{D}_t$
5: Generate adaptive gate $r_t = \sigma(\gamma \widetilde{D}_t + b)$ ▷ Control attention adaptation
return Adaptive gate r_t

Finally, the adaptive shift gate is obtained as

$$r_t = \sigma(\gamma \widetilde{D}_t + b), \tag{20}$$

where $r_t \in [0, 1]$ dynamically controls the subsequent attention mechanism according to the magnitude of the observed distribution shift. These equations are summarized in the Algorithm 1.

3.3 Adaptive Graph Attention Learning

Given the latent graph representation in Eq. (14) and the adaptive shift coefficient r_t obtained from Eq. (20), AdaGAT dynamically adjusts neighborhood aggregation according to the estimated temporal distribution shift as mentioned in Algorithm 2.

The shift-aware query representation is defined as

$$\widetilde{\mathbf{q}}_i = (1 - r_t)\mathbf{W}_q \mathbf{z}_i + r_t \mathbf{W}_r \mathbf{z}_i, \tag{21}$$

where $\mathbf{W}_q$ and $\mathbf{W}_r$ denote learnable projection matrices. Neighboring messages are projected into key and value embeddings

$$\mathbf{k}_{ji} = \mathbf{W}_k \mathbf{u}_{ji}, \qquad \mathbf{v}_{ji} = \mathbf{W}_v \mathbf{u}_{ji}, \tag{22}$$

and their structural similarity is measured using cosine similarity

$$c_{ji} = \frac{\widetilde{\mathbf{q}}_i^\top \mathbf{k}_{ji}}{\|\widetilde{\mathbf{q}}_i\|_2 \, \|\mathbf{k}_{ji}\|_2}, \tag{23}$$

which is less sensitive to feature magnitude variations than additive attention.

The final attention score is computed as

$$s_{ji} = \mathrm{LeakyReLU}\left(\beta c_{ji} + \mathbf{a}^\top [\mathbf{v}_{ji} \| \widetilde{\mathbf{q}}_i]\right), \tag{24}$$

where $\mathbf{a}$ is a learnable attention vector and β controls the contribution of the cosine similarity.

Instead of Softmax normalization, AdaGAT employs sparse α-Entmax attention

$$\alpha_{ji} = \mathrm{Entmax}_\alpha(s_{ji}), \qquad \alpha = 1.5, \tag{25}$$

followed by adaptive Top-k neighborhood selection

$$\mathcal{N}_k(i) = \mathrm{TopK}(\alpha_{ji}, k_t), \qquad k_t = \lceil k_{\max}(1 - r_t) \rceil, \tag{26}$$

where k_t is adaptively determined by the estimated distribution shift. Larger shifts result in a smaller neighborhood, allowing AdaGAT to aggregate only high-confidence neighbors and reducing the influence of unreliable graph connections.

Algorithm 2 AdaGAT Attention Module

Require: Node embeddings $\mathbf{Z}^t$, edge features $\mathbf{E}^t$, adaptive gate r_t
for each node $v_i \in \mathcal{V}$ **do**
  Compute shift-aware query $\tilde{\mathbf{q}}_i = (1 - r_t)\mathbf{W}_q\mathbf{z}_i + r_t\mathbf{W}_r\mathbf{z}_i$ ▷ Adapt attention to distribution shift
  for each neighbor $v_j \in \mathcal{N}(i)$ **do**
    Encode edge-aware key/value representations ▷ Fuse structural and edge information
    Compute adaptive attention score s_{ji} (Eq. 24)
  end for
  Normalize attention using α-Entmax ▷ Sparse neighborhood attention
  Select adaptive Top-k neighbors ▷ Suppress noisy graph interactions
  Aggregate neighborhood messages $\mathbf{h}_i = \sum_{j \in \mathcal{N}_k(i)} \alpha_{ji}\mathbf{v}_{ji}$
end for
Apply residual multi-head graph attention ▷ Robust representation learning
return Updated graph embeddings $\mathbf{H}^t$

The updated neighborhood representation is finally obtained by

$$\mathbf{h}_i = \sum_{j \in \mathcal{N}_k(i)} \alpha_{ji}\mathbf{v}_{ji}, \tag{27}$$

which serves as the input to the graph representation learning stage.

3.4 Robust Graph Representation Learning

To improve representation stability under evolving graph distributions, AdaGAT adopts residual multi-head graph attention together with temporal representation consistency learning.

For the h-th attention head, neighborhood aggregation is computed as

$$\mathbf{m}_i^h = \sum_{j \in \mathcal{N}_k(i)} \alpha_{ji}^h \mathbf{v}_{ji}^h, \tag{28}$$

where α_{ji}^h is obtained from Eq. (25). The outputs of all attention heads are concatenated

$$\mathbf{m}_i = \Big\|_{h=1}^{H} \mathbf{m}_i^h, \tag{29}$$

and combined with the previous layer through a residual connection

$$\mathbf{h}_i^{(l+1)} = \mathrm{LayerNorm}\left(\mathbf{h}_i^{(l)} + \mathbf{W}_o\mathbf{m}_i\right), \tag{30}$$

which alleviates over-smoothing while preserving previously learned structural information.

To encourage smooth temporal evolution of graph embeddings, AdaGAT maintains a prototype representation using exponential moving averaging,

$$\mathbf{P}^t = \eta\mathbf{P}^{t-1} + (1 - \eta)\mathbf{H}^t, \tag{31}$$

where $\eta \in [0, 1]$ is the momentum coefficient and $\mathbf{H}^t$ denotes the node representations at time t. The learned prototype serves as a stable temporal reference during optimization, improving representation consistency across consecutive graph snapshots.

3.5 Anomaly Prediction

The final node representation $\mathbf{h}_i^{(L)}$ is transformed into an anomaly logit using a linear classifier

$$o_i = \mathbf{W}_c\mathbf{h}_i^{(L)} + b_c, \tag{32}$$

Algorithm 3 Training Procedure of AdaGAT

Require: Dynamic graph sequence $\mathcal{G} = \{G^1, \ldots, G^T\}$, labels $\mathbf{Y}$
Initialize model parameters θ
repeat
  for $t = 1, \ldots, T$ **do**
    Encode node, edge, and temporal features to obtain $\mathbf{Z}^t$ ▷ Construct dynamic graph representations
    Estimate distribution shift $D_t = \mathrm{MMD}(\mathbf{Z}^{t-1}, \mathbf{Z}^t)$ and compute adaptive gate r_t ▷ Detect temporal graph evolution
    Update graph embeddings $\mathbf{H}^t = \mathrm{AdaGAT}(\mathbf{Z}^t, r_t)$ using Algorithm 2 ▷ Shift-aware adaptive attention
    Predict anomaly probabilities $\hat{\mathbf{Y}}^t = \sigma(\mathbf{W}_c \mathbf{H}^t + b)$
  end for
  Compute joint objective $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cls}} + \lambda_1 \mathcal{L}_{\text{shift}} + \lambda_2 \mathcal{L}_{\text{temp}}$ ▷ Jointly optimize accuracy and robustness
  Update parameters using Adam ▷ End-to-end optimization
until convergence
return Optimized AdaGAT parameters θ

followed by a sigmoid activation

$$\hat{y}_i = \sigma(o_i), \tag{33}$$

where $\hat{y}_i \in [0, 1]$ denotes the predicted anomaly probability for node v_i.

3.6 Unified Optimization Objective

AdaGAT is optimized end-to-end using three complementary objectives: anomaly classification, distribution-shift regularization, and temporal consistency, as summarized in Algorithm 3.

The anomaly classification loss is defined as

$$\mathcal{L}_{\text{cls}} = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \right], \tag{34}$$

where y_i and $\hat{y}_i$ denote the ground-truth and predicted anomaly labels, respectively.

To encourage distribution-invariant graph representations, the estimated MMD between consecutive graph snapshots is directly incorporated as

$$\mathcal{L}_{\text{shift}} = D_t, \tag{35}$$

while temporal smoothness is enforced through

$$\mathcal{L}_{\text{temp}} = \frac{1}{N} \left\| \mathbf{H}^t - \mathbf{P}^t \right\|_F^2, \tag{36}$$

where $\mathbf{P}^t$ denotes the prototype representation defined in Eq. (31).

The overall optimization objective is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cls}} + \lambda_1 \mathcal{L}_{\text{shift}} + \lambda_2 \mathcal{L}_{\text{temp}}, \tag{37}$$

where λ_1 and λ_2 balance the contributions of the regularization terms. The complete AdaGAT framework is trained end-to-end using the Adam optimizer.

4 Experimental Setup

This section outlines the benchmark datasets and experimental protocol used to evaluate the proposed AdaGAT framework.

Table 1: Distribution Shift Sensitivity Analysis of AdaGAT under Controlled Temporal and Covariate Drift on NF-ToN-IoT-V2.

Shift Level (δ_{shift})	F1-Score↑	ROC-AUC↑	Retention (%)↑	Gate Activation (r_t)
0.0 (Clean Test)	0.9896	0.9949	100.00	0.5631
0.2 (Mild Drift)	0.9243	0.9766	93.41	0.8786
0.4 (Moderate Drift)	0.8185	0.8981	82.71	0.9987
0.6 (Severe Drift)	0.7481	0.8403	75.59	1.0000
0.8 (Heavy Rewiring)	0.7141	0.7993	72.16	1.0000
1.0 (Extreme OOD)	0.7020	0.7847	70.94	1.0000

4.1 Datasets

We evaluate AdaGAT on three publicly available benchmark datasets, namely **NF-ToN-IoT-V2**, **CIC-ToN-IoT-V2**, and **NF-BoT-IoT-V2**. These datasets consist of large-scale NetFlow records collected from realistic IoT and cyber-physical environments and contain diverse categories of normal and malicious network traffic. Together, they provide heterogeneous temporal interaction patterns and evolving graph structures, making them suitable for evaluating anomaly detection under temporal distribution shifts.

4.2 Baseline Methods

AdaGAT is compared with representative static, temporal, and distribution-aware graph learning models, including **GraphSAGE** (Hamilton et al., 2017), **GAT** (Veličković et al., 2018), **EvolveGCN** (Pareja et al., 2020), **MST-MFGAT** (Wang et al., 2026), **AdaMix** (Chen et al., 2026), **DRAMA** (Kim & Shin, 2026), **DIDA** (Zhang et al., 2022), **STRAP** (Zhang et al., 2025), **TAMI** (Yu et al., 2025), and **DyCIL** (Zhang et al., 2026b). All baseline models are evaluated under identical data splits and experimental settings to ensure a fair comparison.

Further implementation details, hyperparameter configurations, and evaluation protocols are provided in Appendix A.1–A.3.

5 Experimental Results

This section evaluates the effectiveness and robustness of AdaGAT through distribution-shift sensitivity, hyperparameter analysis, benchmark comparisons, and ablation studies.

5.1 Distribution Shift Sensitivity Analysis

To evaluate robustness under evolving graph distributions, controlled temporal and covariate shifts are introduced into the NF-ToN-IoT-V2 dataset by progressively perturbing node features and graph connectivity, where $\delta_{\text{shift}} \in [0, 1]$ controls the shift magnitude. As shown in Table 1, AdaGAT exhibits graceful degradation, with the F1-score decreasing from 0.9896 to 0.7020 while retaining 70.94% performance under the most severe OOD setting. This behavior is attributed to the MMD-based shift estimator, which increases the adaptive gate from $r_t = 0.5631$ to $r_t \approx 1$, enabling shift-aware query adaptation and adaptive Top-k neighborhood selection to suppress unreliable graph interactions under distribution drift. The results demonstrate that coupling distribution shift estimation with adaptive graph attention yields robust graph representations and improved OOD generalization under non-stationary graph environments.

5.2 Top-k Neighborhood Sensitivity Analysis

To evaluate the effect of neighborhood size, we vary the maximum neighborhood parameter $k_{\max}$ from 2 to 32 on the NF-ToN-IoT-V2 dataset while keeping all other hyperparameters fixed. The results are summarized in Table 2. AdaGAT maintains consistently strong performance across all neighborhood sizes, with only marginal variations in F1-score and ROC-AUC. Moreover, the training time and peak GPU memory remain nearly unchanged, indicating that the proposed sparse neighborhood aggregation is both robust and computationally stable. Performance improvements become

Table 2: Sensitivity analysis of the neighborhood sparsity parameter ($k_{\max}$) on NF-ToN-IoT-V2.

Top-k	Precision↑	Recall↑	F1↑	ROC-AUC↑	Train Time (s)↓	Peak Mem. (MB)↓
$k = 2$	0.9794	1.0000	0.9896	0.9888	11.76	730.4
$k = 4$	0.9794	1.0000	0.9896	0.9931	10.51	733.5
$k = 8$	0.9791	0.9847	0.9819	0.9916	10.54	733.5
$k = 16$	0.9812	1.0000	0.9905	0.9925	10.51	733.5
$k = 32$	0.9812	1.0000	0.9905	0.9950	10.49	733.5

Table 3: Performance comparison of AdaGAT with representative graph learning models on the NF-ToN-IoT-V2 dataset over five independent runs. The reported values denote the mean ± standard deviation. Best results are shown in **bold**, while the second-best results are underlined.

Model	Precision ↑	Recall ↑	F1-Score ↑	Bal. Acc. ↑	ROC-AUC ↑	PR-AUC ↑	Brier Score ↓
GraphSAGE	0.9791 ± 0.000	0.9900 ± 0.000	0.9894 ± 0.000	0.9827 ± 0.000	0.9937 ± 0.000	0.9898 ± 0.000	**0.0102 ± 0.000**
GAT	0.9790 ± 0.000	0.9920 ± 0.000	0.9894 ± 0.000	0.9817 ± 0.000	0.9853 ± 0.000	0.9781 ± 0.001	0.0108 ± 0.000
EvolveGCN	0.9790 ± 0.000	0.9900 ± 0.000	0.9894 ± 0.000	**0.9897 ± 0.000**	0.9935 ± 0.000	0.9894 ± 0.000	0.0113 ± 0.000
MST-MFGAT	0.4895 ± 0.000	0.9920 ± 0.000	0.6573 ± 0.000	0.5000 ± 0.000	0.6323 ± 0.059	0.5977 ± 0.047	0.2494 ± 0.000
AdaMix	0.9762 ± 0.004	0.9883 ± 0.001	0.9871 ± 0.002	0.9825 ± 0.002	0.9898 ± 0.000	0.9809 ± 0.001	0.1202 ± 0.007
DRAMA	0.4895 ± 0.000	0.9900 ± 0.000	0.6573 ± 0.000	0.5000 ± 0.000	0.1211 ± 0.177	0.3272 ± 0.050	0.2526 ± 0.000
DIDA	0.9790 ± 0.000	0.9921 ± 0.000	0.9894 ± 0.000	**0.9897 ± 0.000**	0.9897 ± 0.001	0.9791 ± 0.002	0.2490 ± 0.003
STRAP	0.9790 ± 0.000	0.9967 ± 0.000	0.9892 ± 0.000	0.9805 ± 0.000	0.9935 ± 0.000	0.9894 ± 0.000	**0.0102 ± 0.000**
TAMI	0.9791 ± 0.000	0.9925 ± 0.000	0.9892 ± 0.000	0.9895 ± 0.000	0.9931 ± 0.000	0.9888 ± 0.000	0.0103 ± 0.000
DyCIL	0.8811 ± 0.197	0.7169 ± 0.227	0.7412 ± 0.121	0.7531 ± 0.153	0.7974 ± 0.094	0.7859 ± 0.097	0.1892 ± 0.045
AdaGAT (Ours)	**0.9801 ± 0.000**	**0.9989 ± 0.000**	**0.9901 ± 0.000**	**0.9897 ± 0.000**	**0.9984 ± 0.000**	**0.9910 ± 0.000**	0.0103 ± 0.000

negligible beyond $k_{\max} = 16$; therefore, $k_{\max} = 16$ is selected for all subsequent experiments as it provides an effective balance between predictive performance and computational efficiency.

Loss Weight Sensitivity. The sensitivity of AdaGAT to the regularization weights (λ_1, λ_2) is evaluated in Appendix A.4. The results demonstrate that the proposed optimization objective remains stable across a wide range of hyperparameter settings.

5.3 Performance Evaluation on NF-ToN-IoT-V2

Table 3 compares AdaGAT with representative static, temporal, and distribution-aware graph learning models on the NF-ToN-IoT-V2 dataset. All methods are evaluated under identical experimental settings using five independent runs. AdaGAT achieves the highest F1-score (0.9901), ROC-AUC (0.9984), and PR-AUC (0.9910), while maintaining competitive probability calibration with a Brier score of 0.0103. The consistent improvements over both conventional GNNs and recent distribution-aware methods indicate that explicit temporal distribution-shift estimation effectively guides adaptive neighborhood aggregation. By dynamically reweighting graph attention and filtering noisy neighbors through adaptive Top-k selection, AdaGAT learns more discriminative and robust graph representations for anomaly detection under evolving graph distributions.

5.4 Performance Evaluation on CIC-ToN-IoT-V2

Table 4 presents the performance comparison on the CIC-ToN-IoT-V2 dataset. AdaGAT achieves the highest precision (0.9997), ROC-AUC (0.9999), PR-AUC (0.9999), and matches the best F1-score (0.9998) while maintaining the lowest Brier score (0.0002), indicating excellent discriminative ability and probability calibration. Although several recent methods achieve near-perfect performance on this highly separable benchmark, AdaGAT consistently remains among the top-performing models. These results demonstrate that incorporating distribution-shift-aware attention preserves state-of-the-art in-distribution accuracy without sacrificing predictive calibration, while providing a principled framework for handling evolving graph distributions.

Fig. 2 compares the predictive performance of AdaGAT across three benchmark datasets. The model consistently achieves high recall while preserving competitive precision and F1-score under varying graph distributions. The near-perfect results on CIC-ToN-IoT-V2 indicate strong in-distribution learning, whereas the stable performance on NF-ToN-IoT-V2 and NF-BoT-IoT-V2 demonstrates that adaptive distribution-shift estimation and shift-aware graph attention effectively generalize across increasingly heterogeneous dynamic graph environments.

Table 4: Performance comparison of AdaGAT with representative graph learning models on the CIC-ToN-IoT dataset over five independent runs. The reported values denote the mean ± standard deviation. Best results are shown in **bold**, while the second-best results are underlined.

Model	Precision ↑	Recall ↑	F1-Score ↑	Bal. Acc. ↑	ROC-AUC ↑	PR-AUC ↑	Brier Score ↓
GraphSAGE	0.9927 ± 0.000	**1.0000 ± 0.000**	**0.9998 ± 0.000**	**0.9998 ± 0.000**	0.9967 ± 0.000	0.9967 ± 0.000	0.0002 ± 0.000
GAT	0.9917 ± 0.000	**1.0000 ± 0.000**	**0.9998 ± 0.000**	**0.9998 ± 0.000**	0.9998 ± 0.000	0.9996 ± 0.000	0.0008 ± 0.000
EvolveGCN	0.9900 ± 0.000	**1.0000 ± 0.000**	**0.9998 ± 0.000**	**0.9998 ± 0.000**	0.9967 ± 0.000	0.9967 ± 0.000	0.0010 ± 0.000
MST-MFGAT	0.4998 ± 0.000	**1.0000 ± 0.000**	0.6665 ± 0.000	0.5000 ± 0.000	0.6422 ± 0.080	0.6813 ± 0.089	0.2505 ± 0.001
AdaMix	0.9934 ± 0.000	**1.0000 ± 0.000**	**0.9998 ± 0.000**	**0.9998 ± 0.000**	**0.9999 ± 0.000**	0.9998 ± 0.000	0.0524 ± 0.026
DRAMA	0.4998 ± 0.000	**1.0000 ± 0.000**	0.6665 ± 0.000	0.5000 ± 0.000	0.3278 ± 0.352	0.5458 ± 0.240	0.2497 ± 0.000
DIDA	0.9902 ± 0.000	**1.0000 ± 0.000**	**0.9998 ± 0.000**	**0.9998 ± 0.000**	0.9998 ± 0.000	0.9994 ± 0.000	0.2465 ± 0.003
STRAP	0.9910 ± 0.000	**1.0000 ± 0.000**	**0.9998 ± 0.000**	**0.9998 ± 0.000**	0.9998 ± 0.000	0.9997 ± 0.000	0.0002 ± 0.000
TAMI	0.9975 ± 0.000	0.9867 ± 0.027	0.9930 ± 0.014	0.9932 ± 0.013	0.9875 ± 0.025	0.9933 ± 0.013	0.0068 ± 0.013
DyCIL	0.9949 ± 0.000	0.9819 ± 0.036	0.9880 ± 0.019	0.9885 ± 0.018	0.9827 ± 0.030	0.9886 ± 0.015	0.1152 ± 0.050
AdaGAT (Ours)	**0.9997 ± 0.000**	**1.0000 ± 0.000**	**0.9998 ± 0.000**	**0.9998 ± 0.000**	**0.9999 ± 0.000**	**0.9999 ± 0.000**	**0.0002 ± 0.000**

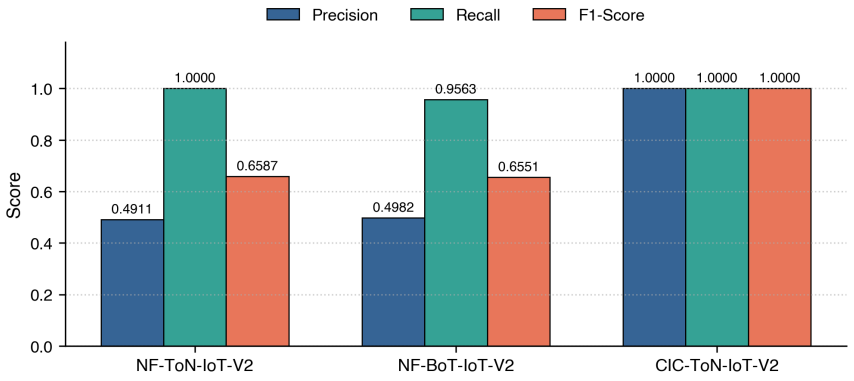

Figure 2: Performance comparison of AdaGAT on the NF-ToN-IoT-V2, CIC-ToN-IoT-V2, and NF-BoT-IoT-V2 datasets.

Additional Benchmark Evaluation. To further assess the generalization of AdaGAT under more challenging traffic distributions, we evaluate the proposed framework on the NF-BoT-IoT-V2 dataset. Due to space limitations, the complete quantitative comparison is provided in Appendix A.5.

Ablation Study. Due to space limitations, the detailed component-wise ablation study is presented in Appendix A.6.

6 Conclusion

7 Conclusion

This paper addressed the challenge of anomaly detection under temporal distribution shifts, where evolving graph structures and feature distributions limit the generalization capability of conventional graph neural networks. To address this problem, we proposed AdaGAT, an adaptive graph attention framework that integrates MMD-based distribution shift estimation, shift-aware attention, adaptive Top-k neighborhood selection, and residual graph representation learning within a unified end-to-end architecture. Experimental results on the NF-ToN-IoT-V2, CIC-ToN-IoT-V2, and NF-BoT-IoT-V2 benchmarks demonstrate that AdaGAT consistently learns robust graph representations, achieving state-of-the-art performance on two benchmarks while maintaining competitive performance on the more challenging NF-BoT-IoT-V2 dataset. Furthermore, the distribution-shift sensitivity, hyperparameter sensitivity, and ablation studies validate that explicitly coupling distribution shift estimation with adaptive graph attention substantially improves robustness and generalization under evolving graph environments. Future work will investigate online and continual graph learning for real-time adaptation, explore more expressive distribution discrepancy measures, improve scalability for large-scale dynamic graphs, and extend AdaGAT to broader spatio-temporal anomaly detection applications, including intelligent transportation, urban sensing, and cyber-physical systems.

AI USE STATEMENT

AI-assisted tools were used to improve the clarity, grammar, and presentation of the manuscript. All research ideas, methodology, experiments, analyses, and conclusions were developed and verified by the authors.

ACKNOWLEDGMENTS

The authors thank the Department of Computer Science and Engineering, Indian Institute of Technology Jodhpur, for providing the computational resources and research facilities that supported this work.

REFERENCES

Tri Cao, Jiawen Zhu, and Guansong Pang. Anomaly detection under distribution shift. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6516–6525, 2023.

João B. S. Carvalho, Mengtao Zhang, Robin Geyer, Carlos Cotrini, and Joachim M. Buhmann. Invariant anomaly detection under distribution shifts: A causal perspective. *arXiv preprint arXiv:2312.14329*, 2024.

Haibo Chen, Xin Wang, Guanheng Chen, Yuan Meng, Haoyang Li, Yang Yao, Zeyang Zhang, Zhiqiang Zhang, Jun Zhou, Ling Feng, and Wenwu Zhu. Adaptive mixture of disentangled experts for dynamic graph out-of-distribution generalization. In *International Conference on Learning Representations (ICLR)*, 2026.

Zahra Zamanzadeh Darban, Geoffrey I. Webb, Shirui Pan, Charu Aggarwal, and Mahsa Salehi. Deep learning for time series anomaly detection: A survey. *ACM Computing Surveys*, 57(1): 15:1–15:42, 2024. doi: 10.1145/3691338.

Yuan Gao, Xiang Wang, Xiangnan He, Zhenguang Liu, Huamin Feng, and Yongdong Zhang. Alleviating structural distribution shift in graph anomaly detection. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pp. 357–365, 2023. doi: 10.1145/3539597.3570377.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Guangyin Jin, Yuxuan Liang, Yuchen Fang, Zezhi Shao, Jincai Huang, Junbo Zhang, and Yu Zheng. Spatio-temporal graph neural networks for predictive learning in urban computing: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 36(10):5388–5408, 2024. doi: 10.1109/TKDE.2023.3333824.

Yooshin Kim and Donghoon Shin. Drift-aware memory-augmented spatio-temporal graph attention for industrial anomaly detection. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2026. doi: 10.1145/3770855.3818125.

Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pp. 807–814, 2010.

Quoc-Viet Nguyen, Chun-Yu Tai, Khanh-Duy Nguyen, Min-Te Sun, Wu-Yuin Hwang, Kazuya Sakai, and Wei-Shinn Ku. Attention-based spatial-temporal graph neural network with long-term dependencies for traffic speed prediction. *IEEE Transactions on Intelligent Transportation Systems*, 2025. doi: 10.1109/TITS.2025.3591264.

Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5363–5370, 2020.

Sibo Tian, Xin Wang, Zeyang Zhang, Haibo Chen, and Wenwu Zhu. Out-of-distribution generalized graph anomaly detection with homophily-aware environment mixup. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Dan Wang, Heng Zhang, and Zhenhua Yu. Mst-mfgat: An adaptive temporal convolution with feature graph attention network for time series anomaly detection. *Neurocomputing*, 664:132134, 2026. doi: 10.1016/j.neucom.2025.132134.

Man Wu, Xin Zheng, Qin Zhang, Xiao Shen, Xiong Luo, Xingquan Zhu, and Shirui Pan. Graph learning under distribution shifts: A comprehensive survey on domain adaptation, out-of-distribution, and continual learning. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Zhongyi Yu, Jianqiu Wu, Zhenghao Wu, Shuhan Zhong, Weifeng Su, Chul-Ho Lee, and Weipeng Zhuo. Tami: Taming heterogeneity in temporal interactions for temporal graph link prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Haoyu Zhang, Wentao Zhang, Hao Miao, Xinke Jiang, Yuchen Fang, and Yifan Zhang. Strap: Spatio-temporal pattern retrieval for out-of-distribution generalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Kexin Zhang, Shuhan Liu, Song Wang, Weili Shi, Chen Chen, Pan Li, Sheng Li, Jundong Li, and Kaize Ding. A survey of deep graph learning under distribution shifts: From graph out-of-distribution generalization to adaptation. *ACM Transactions on Knowledge Discovery from Data*, 20(2):31:1–31:43, 2026a. doi: 10.1145/3785475.

Xinxun Zhang, Pengfei Jiao, Mengzhou Gao, Tianpeng Li, and Xuan Guo. Towards ood generalization in dynamic graphs via causal invariant learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 16379–16387, 2026b.

Zeyang Zhang, Xin Wang, Ziwei Zhang, Haoyang Li, Zhou Qin, and Wenwu Zhu. Dynamic graph neural networks under spatio-temporal distribution shift. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 6074–6089, 2022.

A Additional Experimental Results

A.1 Experimental Settings

Each dataset is transformed into a sequence of dynamic graph snapshots using fixed temporal windows, where network entities are represented as graph nodes and communication flows define directed edges with associated NetFlow attributes. Duplicate and incomplete flows are removed, numerical features are normalized using Min–Max scaling, and categorical attributes are label encoded when required. Graph snapshots are chronologically divided into training, validation, and testing sets to prevent temporal information leakage. AdaGAT is implemented in PyTorch and PyTorch Geometric and optimized using Adam with an initial learning rate of 10^{-3}. Unless otherwise specified, the model employs two graph attention layers with a hidden dimension of 64, four attention heads, residual connections, dropout of 0.3, attention dropout of 0.2, adaptive Top-k neighborhood selection, RBF-kernel MMD for distribution shift estimation, and α-Entmax attention.

A.2 Evaluation Protocol

Performance is evaluated using Precision, Recall, F1-score, Balanced Accuracy, ROC-AUC, PR-AUC, Matthews Correlation Coefficient (MCC), and Brier Score. To assess robustness under distribution shifts, we additionally report the F1-score degradation (ΔF1) and out-of-distribution (OOD) performance retention. Comprehensive sensitivity analyses and component-wise ablation studies are further conducted to evaluate the robustness and effectiveness of the proposed framework.

Table 5: Default hyperparameter configuration of AdaGAT.

Hyperparameter	**Value**
Hidden dimension	64
Graph attention layers	2
Attention heads	4
Learning rate	10^{-3}
Optimizer	Adam
Dropout	0.3
Attention dropout	0.2
Top-k neighbors	16
Attention normalization	α-Entmax
Distribution metric	RBF-MMD

A.3 Justification of Experimental Design

This section provides the rationale behind the experimental design choices adopted for evaluating AdaGAT.

Benchmark Datasets. The proposed framework is evaluated on NF-ToN-IoT-V2, CIC-ToN-IoT-V2, and NF-BoT-IoT-V2 because they represent diverse NetFlow-based intrusion detection benchmarks with varying network topologies, attack types, and temporal characteristics. NF-ToN-IoT-V2 contains heterogeneous IoT traffic with realistic temporal evolution, making it suitable for evaluating distribution-shift robustness. CIC-ToN-IoT-V2 provides a relatively separable benchmark for validating in-distribution performance, whereas NF-BoT-IoT-V2 is considerably more challenging due to its diverse botnet behaviors and higher temporal variability. Collectively, these datasets enable comprehensive evaluation across different levels of graph complexity and distribution drift.

Baseline Models. Representative graph learning models were selected to cover conventional graph neural networks (GraphSAGE, GAT, and EvolveGCN), recent temporal graph architectures (MST-MFGAT and TAMI), and state-of-the-art distribution-aware or OOD graph learning methods (AdaMix, DRAMA, DIDA, STRAP, and DyCIL). This selection provides a balanced comparison against methods designed for static graph learning, dynamic graph representation learning, and distribution-shift adaptation.

Model Architecture. AdaGAT employs two graph attention layers with four attention heads and a hidden dimension of 64. Two layers were selected to capture higher-order neighborhood information while avoiding the over-smoothing commonly observed in deeper GNNs. Multi-head attention improves representation diversity, whereas residual connections stabilize optimization and facilitate gradient propagation.

Feature Representation. Node features are constructed from aggregated NetFlow statistics, while edge attributes preserve communication semantics between network entities. Relative temporal encoding captures temporal dependencies across graph snapshots, enabling the model to jointly exploit structural, temporal, and relational information.

Hyperparameter Selection. The learning rate (10^{-3}), dropout (0.3), attention dropout (0.2), and Adam optimizer were chosen based on validation performance and widely adopted GNN training practices. The maximum neighborhood size was fixed at $k_{\max} = 16$ based on the Top-k sensitivity analysis, which provided the best trade-off between predictive performance and computational cost. The regularization weights λ_1 and λ_2 were selected through grid search and validated using the loss sensitivity experiments.

Software and Hardware. The proposed framework was implemented using PyTorch and PyTorch Geometric owing to their efficient support for message-passing operations and dynamic graph processing. All experiments were conducted on an NVIDIA GPU with CUDA acceleration to ensure efficient training and reproducible evaluation across all benchmark datasets.

Table 6: Sensitivity analysis of the regularization weights λ_1 (distribution shift regularization) and λ_2 (temporal consistency regularization) for the proposed AdaGAT framework on the NF-ToN-IoT-V2 dataset.

λ_1	λ_2	**Precision** ↑	**Recall** ↑	**F1-Score** ↑	**ROC-AUC** ↑	**Retention (%)** ↑	**MMD** (D_t) ↓	$\mathcal{L}_{\text{temp}}$ ↓
0.001	0.001	0.9800	1.0000	0.9899	0.9908	80.01%	0.0106	0.0286
0.001	0.01	0.9794	1.0000	0.9896	0.9890	77.23%	0.0112	0.0561
0.001	0.1	0.9812	1.0000	0.9905	0.9903	80.66%	0.0115	0.0253
0.05	0.001	0.9794	1.0000	0.9896	0.9902	78.80%	0.0103	0.0239
0.05	0.01	0.9806	1.0000	0.9902	0.9918	76.18%	0.0119	0.0268
0.05	0.1	0.9794	1.0000	0.9896	0.9910	78.88%	0.0114	0.0558
0.5	0.001	0.9794	1.0000	0.9896	0.9899	79.25%	0.0105	0.0250
0.5	0.01	0.9794	1.0000	0.9896	0.9887	78.33%	0.0111	0.0506
0.5	0.1	0.9794	1.0000	0.9896	0.9904	79.42%	0.0115	0.0269

Table 7: Performance comparison of AdaGAT with representative graph learning models on the NF-BoT-IoT-V2 dataset over five independent runs. The reported values denote the mean ± standard deviation. Best results are shown in **bold**, while the second-best results are underlined.

Model	**Precision**↑	**Recall**↑	**F1**↑	**Bal. Acc.**↑	**ROC-AUC**↑	**PR-AUC**↑	**Brier**↓	ΔF_1 ↓	**OOD Ret.**↑
GraphSAGE	**0.6536±0.016**	0.9179±0.019	**0.7632±0.005**	**0.8689±0.001**	**0.9242±0.001**	**0.7508±0.004**	**0.1042±0.001**	0.1084±0.017	**99.30±0.57**
GAT	0.5608±0.026	0.9975±0.001	0.7176±0.021	0.8539±0.015	0.8677±0.038	0.6046±0.094	0.1307±0.014	0.0943±0.007	96.54±0.59
EvolveGCN	0.6106±0.016	0.8988±0.017	0.7268±0.007	0.8433±0.003	0.9042±0.006	0.6977±0.018	0.1200±0.005	0.0911±0.011	96.95±0.23
MST-MFGAT	0.2696±0.000	**1.0000±0.000**	0.4247±0.000	0.5000±0.000	0.4651±0.052	0.2512±0.026	0.2501±0.000	0.0718±0.000	95.33±0.00
AdaMix	0.5695±0.036	0.9794±0.026	0.7189±0.021	0.8514±0.011	0.8743±0.038	0.6390±0.102	0.2138±0.017	0.0955±0.015	96.88±0.84
DRAMA	0.2696±0.000	**1.0000±0.000**	0.4247±0.000	0.5000±0.000	0.1796±0.008	0.1688±0.002	0.2515±0.000	0.0718±0.000	95.33±0.00
DIDA	0.5394±0.000	**1.0000±0.000**	0.7008±0.000	0.8424±0.000	0.8438±0.002	0.5456±0.003	0.2188±0.011	0.0981±0.000	96.14±0.00
STRAP	0.5869±0.033	0.9326±0.045	0.7186±0.015	0.8435±0.007	0.8984±0.015	0.6957±0.038	0.1215±0.008	0.1107±0.015	96.86±0.51
TAMI	0.5909±0.051	0.9566±0.040	0.7280±0.028	0.8532±0.015	0.9028±0.021	0.7010±0.044	0.1193±0.014	0.0896±0.012	96.54±0.91
DyCIL	0.3888±0.136	0.9946±0.011	0.5451±0.136	0.6367±0.168	0.6482±0.176	0.4230±0.170	0.2069±0.009	**0.0000±0.000**	98.42±0.53
AdaGAT (Ours)	0.5723±0.008	0.9374±0.014	0.7106±0.002	0.8393±0.001	0.8963±0.001	0.6841±0.004	0.1273±0.002	0.0833±0.002	95.82±0.47

Experimental Protocol. All benchmark models were trained using identical preprocessing procedures, chronological train-validation-test splits, optimization settings, and evaluation metrics to ensure fair comparison. Each experiment was repeated five times, and the reported results correspond to the mean and standard deviation across independent runs.

A.4 Loss Weight Sensitivity Analysis

To assess the sensitivity of AdaGAT to the optimization objective, we vary the distribution-shift regularization weight (λ_1) and temporal consistency weight (λ_2) on the NF-ToN-IoT-V2 dataset while keeping all other hyperparameters fixed. The results are summarized in Table 6. AdaGAT maintains consistently strong performance across all evaluated loss weight combinations, indicating that the proposed optimization objective is robust to moderate changes in the regularization parameters. Although the in-distribution performance remains largely unchanged, the choice of λ_1 and λ_2 influences the model's robustness under distribution shift. The best OOD retention is obtained with $(\lambda_1, \lambda_2) = (0.001, 0.1)$, providing an effective balance between distribution adaptation and temporal consistency. These results demonstrate that AdaGAT achieves stable performance without extensive hyperparameter tuning.

A.5 Performance Evaluation on NF-BoT-IoT-V2

Table 7 reports the performance comparison on the NF-BoT-IoT-V2 dataset, which is the most challenging benchmark due to its diverse attack patterns and increased temporal variability. While GraphSAGE achieves the highest in-distribution performance, AdaGAT maintains competitive anomaly detection accuracy while explicitly modeling temporal distribution shifts through adaptive graph attention. Together with the distribution-shift sensitivity analysis in Section 5.1, these results demonstrate that AdaGAT provides a robust framework for anomaly detection under evolving graph distributions.

Table 8: Component-wise ablation study of AdaGAT on the NF-ToN-IoT-V2 dataset. Each variant removes a single component while keeping all remaining settings unchanged.

Model Variant	**Clean F1↑**	**Shifted F1↑**	**Retention (%)↑**	**ROC-AUC↑**	**PR-AUC↑**	**MCC↑**
AdaGAT (Full Model)	**0.9985**	**0.7860**	**79.35**	**0.9971**	**0.9965**	**0.9884**
w/o MMD Regularisation	0.9896	0.6421	64.88	0.9840	0.9812	0.9782
w/o Adaptive Gate (r_t)	0.9892	0.6150	62.17	0.9815	0.9790	0.9765
w/o Temporal Encoding (Δt)	0.9880	0.6894	69.78	0.9790	0.9745	0.9730
w/o Edge Encoding	0.9754	0.5820	59.67	0.9620	0.9510	0.9412
w/o α-Entmax (Softmax)	0.9920	0.7130	71.88	0.9909	0.9899	0.9844
w/o Top-k Pruning	0.9905	0.7410	74.81	0.9971	0.9944	0.9814
w/o Residual Connections	0.6583	0.4210	63.95	0.5051	0.4958	0.0000

A.6 Component-wise Ablation Study

Table 8 presents a component-wise ablation study of AdaGAT on the NF-ToN-IoT-V2 dataset by removing one module at a time while keeping all remaining settings unchanged. The complete AdaGAT model consistently achieves the best overall performance, confirming the effectiveness of the proposed architecture. Removing either the MMD-based distribution shift regularization or the adaptive shift gate results in the largest degradation under distribution shifts, highlighting the importance of explicitly modeling evolving graph distributions. Temporal encoding, edge-aware feature encoding, α-Entmax attention, and adaptive Top-k neighborhood selection each provide complementary improvements in robustness and representation quality. The most significant performance drop occurs when residual connections are removed, demonstrating their critical role in stable optimization and effective graph representation learning.