AELAD: ADAPTIVE EVIDENTIAL LEARNING FOR ANOMALY DETECTION IN DYNAMIC GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Dynamic graphs are pervasive in real-world scenarios such as financial transactions and social networks, making the detection of anomalous nodes a problem of significant practical importance. However, existing methods predominantly adopt a point estimation paradigm. They lack uncertainty awareness and cannot quantify the reliability of anomaly predictions under sparse histories or highly variable node behaviors. Furthermore, feature-based anomalies and structure-based anomalies are heterogeneous signals whose naive fusion degrades detection performance due to inconsistent scales. To address these challenges, we propose AELAD, a framework for anomaly detection in dynamic graphs based on adaptive evidential learning. Specifically, AELAD incorporates an evidential learning mechanism. It enables the model to output distributional parameters rather than point estimates. Moreover, a Normal-Inverse Gamma (NIG) distribution and a Beta distribution are employed to independently model feature uncertainty and topological uncertainty. Their scores are then adaptively fused via learnable weights. Extensive experiments on six datasets demonstrate that AELAD achieves superior anomaly detection performance compared to state-of-the-art baselines.

1 INTRODUCTION

Anomaly detection is one of the core tasks in graph mining (Qiao et al., 2025). Its goal is to identify a small number of anomalous nodes that deviate from regular patterns among massive normal behaviors (Ai et al., 2026). Unlike static graphs, real-world graph structures tend to evolve continuously over time(Farokhi et al., 2026). This temporal dynamics, on one hand, provides richer contextual information for anomaly detection, and on the other hand, it blurs the boundary between normal and anomalous behavior (Liu et al., 2025). Consequently, anomaly detection in dynamic graphs has emerged as an important research direction.

In the early work of anomaly detection in dynamic graph, researchers primarily scored edges or nodes directly using statistical measures (Chang et al., 2021). SEDANSPOT (Eswaran & Faloutsos, 2018) captures the sparse connectivity of edges via random walks. MIDAS (Bhatia et al., 2020) detects bursty micro-cluster anomalies based on the Count-Min Sketch structure. However, their detection capability relies on predefined statistical assumptions, making it difficult to capture complex non-linear anomaly patterns (Xiao et al., 2026). With the advancement of graph neural networks, embedding-based methods have gradually become mainstream (Wang et al., 2025). SLADE (Lee et al., 2024) learns dynamic node representations in a self-supervised manner and identifies anomalies by detecting representation drift. GeneralDyG (Yang et al., 2025) and DP-DGAD (Zheng et al., 2026) further pursue cross-dataset generalization, enhancing model versatility through TensGNN and dynamic prototype memory modules. Despite the significant progress achieved by these methods in dynamic graph anomaly detection, existing approaches still face the following two challenges:

Challenge 1: Existing dynamic graph anomaly detection methods lack uncertainty awareness and cannot characterize the reliability of anomaly predictions. In real-world dynamic graphs, not all nodes that deviate from normal patterns provide equally reliable anomaly evidence. Take active traders or trending topic nodes as examples. Their behavioral patterns are inherently highly variable, causing their embeddings to drift away from cluster centers. Newly joined nodes, on the other hand, suffer from poor embedding quality due to sparse historical interactions and limited neighborhood structures. As illustrated in Figure 1, both types of nodes may produce high recon-

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.

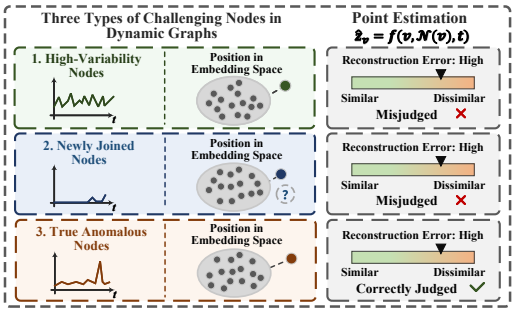


Figure 1: Illustration of three types of challenging nodes in dynamic graphs under the point estimation framework. High-variability nodes and newly joined nodes both produce high reconstruction errors due to their complex behavioral patterns or limited historical interaction information, leading to misjudgment. Only truly anomalous nodes that deviate from normal patterns can be correctly identified in the embedding space.

struction errors under the point estimation framework, even though their predictions are affected by behavioral variability or limited historical information. Existing methods such as SAD (Tian et al., 2023) and DP-DGAD (Zheng et al., 2026) commonly adopt point estimation strategies, outputting only similarity scores or reconstruction errors without providing any measure of prediction confidence. While some efforts such as GEL (Wei et al., 2025) attempt to incorporate uncertainty modeling via evidence fusion, they remain confined to static graphs and cannot capture the continuous temporal evolution of dynamic graphs. Consequently, existing dynamic methods provide no explicit measure of the reliability associated with anomaly predictions under behavioral diversity.

Challenge 2: Feature anomalies and structural anomalies are signals of fundamentally different natures, and the simple concatenation adopted by existing methods weakens overall detection performance. Beyond the limitation of point estimation, a deeper challenge arises when uncertainty is introduced into the detection framework. In dynamic graphs, anomalous behaviors occur in two distinctly different forms. Feature anomalies refer to deviations of a node's own attribute vector from normal patterns, representing a form of deviation in continuous value space. Structural anomalies refer to deviations of a node's connectivity pattern from normal graph topology, representing a form of aberration in discrete topological space. These two types of signals differ fundamentally in scale and statistical properties. Directly concatenating or linearly combining them causes mutual signal interference, ultimately weakening overall detection performance. Existing methods such as SLADE (Lee et al., 2024) and GeneralDyG (Yang et al., 2025) either focus predominantly on structural information modeling or treat feature signals and structural signals as equivalent. This makes it difficult to jointly exploit both types of anomaly evidence, limiting further improvements in detection performance.

To address the aforementioned challenges, we propose **AELAD**, an **A**daptive **E**vidential **L**earning framework for **A**nomaly **D**etection in Dynamic Graphs. Specifically, the model first generates temporal node embeddings through a memory update mechanism and a graph attention network. Building upon this, we construct a graph evidential learning module comprising a feature evidence network and a topological evidence network, which produce two types of uncertainty scores. Finally, the two uncertainty scores are adaptively fused with contrastive learning scores via learnable weights for the anomaly detection task. The main contributions of this paper are as follows:

- We propose AELAD, a graph evidential learning framework for anomaly detection in dynamic graphs, which explicitly models the predictive uncertainty of node embeddings.
- We construct a feature evidence network and a topological evidence network. Accounting for the distinct natures of continuous node features and discrete connectivity structures, we adopt a Normal-Inverse Gamma (NIG) distribution and a Beta distribution for uncertainty modeling, respectively.
- We design an adaptive fusion mechanism based on learnable weights, which dynamically weights the two types of uncertainty scores to avoid the signal interference introduced by naive concatenation.

- We conduct extensive experiments on six datasets. Results demonstrate that AELAD outperforms existing state-of-the-art methods in both anomaly detection accuracy and computational efficiency.

2 Related Work

2.1 Anomaly Detection in Dynamic Graphs

Anomaly detection in dynamic graphs aims to identify nodes or edges that deviate significantly from normal behavioral patterns in evolving networks (Fu et al., 2025). Early methods treat dynamic graphs as discrete snapshot sequences (Xiang & Zhang, 2022), employing GCN- and RNN-based architectures to capture structural dependencies and temporal regularities (Mungari et al., 2026). For instance, AddGraph (Zheng et al., 2019) combines GCN and GRU to model structural dependencies. TADDY (Liu et al., 2021) applies a Transformer model on discrete snapshots to jointly learn static graph patterns and their temporal evolution. However, snapshot-based methods inevitably introduce temporal lag and struggle to capture fine-grained interaction dynamics. To this end, continuous-time methods have been proposed to directly process interaction streams. MIDAS (Bhatia et al., 2020) detects micro-cluster anomalies in edge streams in constant time and memory. In terms of labeled data utilization, semi-supervised methods have been further explored. SAD (Tian et al., 2023) improves detection performance by combining a supervised deviation network with pseudo-label contrastive learning. Recent research has shifted toward improving generalization capability. DP-DGAD (Zheng et al., 2026) introduces dynamic prototypes to simultaneously capture domain-specific and domain-agnostic evolutionary patterns. Despite performing well in their respective settings, these methods universally adopt point estimation strategies.

2.2 Evidential Learning

Evidential learning originates from Dempster-Shafer evidence theory and the subjective logic framework, aiming to provide quantifiable uncertainty estimates for the predictions of deep neural networks (Pandey et al., 2026). Sensoy et al. (Sensoy et al., 2018) first introduced Evidential Deep Learning (EDL) into classification tasks. Building upon EDL, subsequent studies have extended and refined it along multiple dimensions. Regarding distributional expressiveness, F-EDL (Yoon & Kim, 2026) replaces the standard Dirichlet distribution with a flexible Dirichlet to enhance modeling of multimodal and highly ambiguous uncertainty. In terms of application scope, MetaEDL (Neupane et al., 2021) combines evidential learning with meta-learning for uncertainty-aware prediction in cold-start recommendation scenarios. Recently, evidential learning has been further introduced into the field of graph learning. GraphEVER (Ju et al., 2026) applies evidence theory to long-tail graph classification, mitigating the overconfidence problem caused by class imbalance through multi-expert evidence fusion. GEL (Wei et al., 2025) focuses on unsupervised graph anomaly detection, simultaneously quantifying graph uncertainty and reconstruction uncertainty through higher-order evidential distributions. The above works demonstrate the broad applicability and uncertainty modeling potential of evidential learning in the graph domain.

3 Methodology

3.1 Framework of AELAD

As shown in Figure 2, our model learns node embeddings from dynamic graphs, captures feature and topological uncertainties, and fuses multi-source scores to compute node anomaly scores.

3.2 Dynamic Graph Input Module

Given a dynamic graph $\mathcal{G} = \{(u_i, v_i, t_i)\}_{i=1}^{N}$, upon the arrival of each interaction, the model first constructs a bidirectional adjacency list. For node u, its historical neighbor set is sorted in ascending order of timestamps:

$$\mathcal{N}(u, t) = \{(u, v, \tau) \in \mathcal{E} \mid \tau \leq t\}. \tag{1}$$

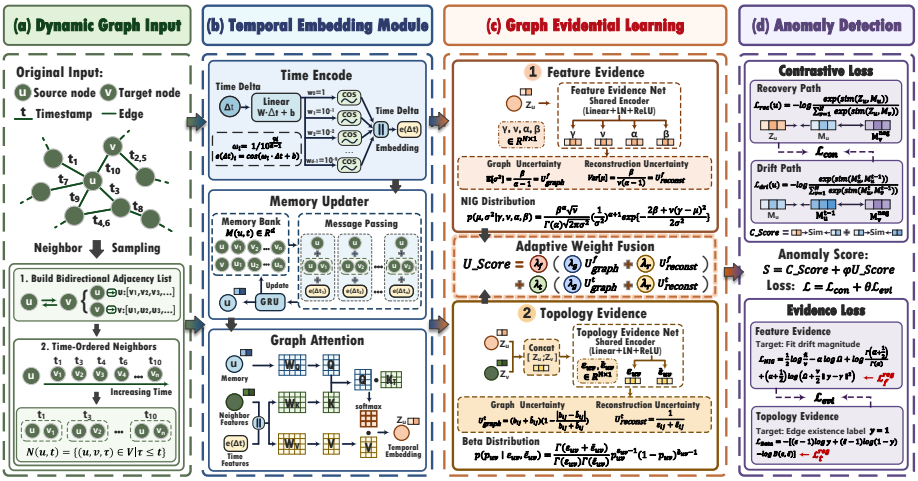


Figure 2: The framework of AELAD. AELAD consists of four modules: (a) Dynamic Graph Input, (b) Temporal Embedding Module, (c) Graph Evidential Learning, and (d) Anomaly Detection.

During the sampling stage, a temporally truncated strategy based on binary search is adopted, retaining only the K most recent neighbors prior to time t.

3.3 Temporal Embedding Module

The temporal embedding module encodes the interaction history of nodes in the dynamic graph into low-dimensional continuous representations. It consists of three components: a time encoder, a memory updater, and a graph attention network.

Time Encoder. The time encoder maps a time interval Δt to a continuous vector representation using a cosine positional encoding:

$$\mathbf{e}(\Delta t)_i = \cos(\omega_i \cdot \Delta t + b_i), \quad \omega_i = \frac{1}{10^{(i-1)/(d-1)}}, \tag{2}$$

where d is the encoding dimension, ω_i is the i-th frequency component, and b_i is a learnable bias.

Memory Updater. The memory updater maintains a node memory bank $\mathbf{M}(u,t) \in \mathbb{R}^d$, recording the historical interaction state of each node up to time t. For an interaction of node u at time t, the raw message is constructed by concatenating the destination node's memory and the time encoding:

$$\mathbf{m}(u,t) = \mathbf{M}(v,t^-) \,\|\, \mathbf{e}(\Delta t). \tag{3}$$

To mitigate aggregation conflicts among multiple messages within the same batch, an attention-weighted aggregation strategy is adopted. Normalized weights are computed for multiple messages of node u within the same batch:

$$w_i = \frac{\exp(a(\mathbf{m}_i))}{\sum_j \exp(a(\mathbf{m}_j))}, \qquad \mathbf{m}(u) = \sum_i w_i \mathbf{m}_i, \tag{4}$$

where $a(\cdot)$ is a single-layer linear mapping. The node memory is then updated via a GRU cell:

$$\mathbf{M}(u,t) = \mathrm{GRU}\big(\mathbf{m}(u),\, \mathbf{M}(u,t^-)\big). \tag{5}$$

Graph Attention Network. Taking node memory and temporal features as input, the graph attention network aggregates temporal neighbor information through a multi-head temporal attention mechanism to generate the temporal embedding $\mathbf{Z}_u \in \mathbb{R}^d$ for node u. For the neighbor set $\mathcal{N}(u,t)$ of node u at time t, the node's temporal feature serves as the query, while the concatenation of neighbor features and temporal features serves as the key-value pairs:

$$\mathbf{H}_{\mathcal{N}} = \big[\mathbf{M}(v_1,t),\, \mathbf{M}(v_2,t),\, \ldots,\, \mathbf{M}(v_K,t)\big] \in \mathbb{R}^{K\times d}, \tag{6}$$

$$\mathbf{Q} = \mathbf{M}(u,t)\mathbf{W}^Q, \quad \mathbf{K} = \big[\mathbf{H}_\mathcal{N} \,\|\, \mathbf{e}(\Delta t_\mathcal{N})\big]\mathbf{W}^K, \quad \mathbf{V} = \big[\mathbf{H}_\mathcal{N} \,\|\, \mathbf{e}(\Delta t_\mathcal{N})\big]\mathbf{W}^V. \tag{7}$$

The weighted aggregated neighbor representation is fused with the node's own memory to obtain the final temporal embedding:

$$\mathbf{Z}_u = \mathrm{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}). \tag{8}$$

3.4 Graph Evidential Learning Module

The graph evidential learning module is the core innovation of AELAD. To address the fundamental difference between feature anomalies and structural anomalies, we design a feature evidence network and a topological evidence network.

Feature Evidence Network. Taking the node temporal embedding $\mathbf{Z}_u$ as input, the feature evidence network outputs the four parameters $(\gamma, \nu, \alpha, \beta)$ of a Normal-Inverse Gamma (NIG) distribution for each node u through a shared encoder (Linear + LN + ReLU). The scalar fitting target is defined as the ℓ_2 norm of the residual between the node embedding and its memory vector:

$$y = \|\mathbf{Z}_u - \mathbf{M}_u\|_2 \in \mathbb{R}^+. \tag{9}$$

This scalar quantifies the magnitude of behavioral drift for node u, and is consistent with the scalar output dimensionality of the NIG network: a larger y indicates a greater deviation of the current embedding from the historical memory state. The NIG distribution is defined as:

$$p\big(\mu, \sigma^2 \mid y, \nu, \alpha, \beta\big) = \frac{\beta^\alpha \sqrt{\nu}}{\Gamma(\alpha)\sqrt{2\pi\sigma^2}} \left(\frac{1}{\sigma^2}\right)^{\alpha+1} \exp\!\left(-\frac{2\beta + \nu(y-\mu)^2}{2\sigma^2}\right). \tag{10}$$

Based on the NIG distribution parameters, the graph uncertainty and reconstruction uncertainty in the node feature space are defined as:

$$u^f_{\mathrm{graph}} = \frac{\beta}{\alpha - 1}, \qquad u^f_{\mathrm{recon}} = \frac{\beta}{\nu(\alpha - 1)}. \tag{11}$$

Topological Evidence Network. Taking the concatenation of source and destination node embeddings $\mathbf{Z}_u \| \mathbf{Z}_v$ as input, the topological evidence network outputs Beta distribution parameters $(\varepsilon^+_{uv}, \varepsilon^-_{uv})$ for each edge (u, v) through a shared encoder, using edge existence as the supervision signal. For each observed edge (u, v), a negative pair is constructed by replacing v with a randomly sampled non-neighbor node. The Beta distribution is defined as:

$$p\big(p_{uv} \mid \varepsilon^+_{uv}, \varepsilon^-_{uv}\big) = \frac{\Gamma(\varepsilon^+_{uv} + \varepsilon^-_{uv})}{\Gamma\big(\varepsilon^+_{uv}\big)\,\Gamma\big(\varepsilon^-_{uv}\big)}\, p_{uv}^{\varepsilon^+_{uv}-1}\,(1 - p_{uv})^{\varepsilon^-_{uv}-1}. \tag{12}$$

Based on the Beta distribution parameters, the graph uncertainty and reconstruction uncertainty in the topological space are defined as:

$$u^t_{\mathrm{recon}} = \frac{1}{\varepsilon^+_{uv} + \varepsilon^-_{uv}}, \qquad u^t_{\mathrm{graph}} = 1 - \frac{|\varepsilon^+_{uv} - \varepsilon^-_{uv}|}{\varepsilon^+_{uv} + \varepsilon^-_{uv}}, \tag{13}$$

Here, u^t_{recon} reflects the inverse total evidence strength: a larger total evidence $S = \varepsilon^+ + \varepsilon^-$ yields a more concentrated Beta distribution and a smaller value of u^t_{recon}. The quantity u^t_{graph} measures the balance between evidence for edge existence and non-existence. When $\varepsilon^+ \approx \varepsilon^-$, the two alternatives receive comparable evidence and u^t_{graph} approaches one, indicating an ambiguous connectivity prediction. When one evidence parameter dominates the other, u^t_{graph} approaches zero, indicating a less ambiguous prediction.

Evidential Loss Functions. The feature evidential loss adopts the negative log marginal likelihood of the NIG distribution:

$$\mathcal{L}_{\mathrm{NIG}} = \frac{1}{2}\log\frac{\pi}{\nu} - \alpha\log\Omega + \left(\alpha + \frac{1}{2}\right)\log\!\left(\Omega + \frac{\nu(y-\gamma)^2}{2}\right) + \log\frac{\Gamma(\alpha)}{\Gamma\big(\alpha + \frac{1}{2}\big)}, \tag{14}$$

where $\Omega = \beta + \frac{\nu(y-\gamma)^2}{2}$. The topological evidential loss adopts the negative log-likelihood of the Beta-Bernoulli predictive distribution:

$$\widehat{p}_{uv} = \frac{\varepsilon^+_{uv}}{\varepsilon^+_{uv} + \varepsilon^-_{uv}}, \qquad \mathcal{L}_{\text{Beta}} = -y_{uv} \log(\widehat{p}_{uv} + \epsilon) - (1 - y_{uv}) \log(1 - \widehat{p}_{uv} + \epsilon), \tag{15}$$

where $y_{uv} = 1$ for an observed edge and $y_{uv} = 0$ for a negatively sampled edge, and ϵ is a small constant for numerical stability.

The total evidential loss is the weighted sum of both terms with regularization:

$$\mathcal{L}_{\text{evl}} = \zeta \cdot \mathcal{L}_{\text{NIG}} + (1 - \zeta) \cdot \mathcal{L}_{\text{Beta}} + \eta \cdot \mathcal{L}^f_{\text{reg}} + (1 - \eta) \cdot \mathcal{L}^t_{\text{reg}}, \tag{16}$$

where $\mathcal{L}^f_{\text{reg}}$ and $\mathcal{L}^t_{\text{reg}}$ are the KL divergence regularization terms for the feature and topological evidence, respectively. They penalize the model for accumulating excessive evidence when predictions are incorrect, preventing spuriously high-confidence outputs on anomalous samples. Here ζ and η are hyperparameters.

3.5 Adaptive Fusion and Anomaly Detection

Contrastive Learning. Contrastive learning constructs node behavioral deviation signals from two perspectives: a recovery path and a drift path. The recovery path captures the consistency between a node's current behavior and its historical state by minimizing the InfoNCE loss between the node embedding $\mathbf{Z}_u$ and its corresponding memory $\mathbf{M}_u$; the drift path captures the smooth evolution of node states by minimizing the InfoNCE loss between memories at adjacent timestamps $\mathbf{M}^t_u$ and $\mathbf{M}^{t+1}_u$. The two losses are defined as:

$$\mathcal{L}_{\text{rec}}(u) = -\log \frac{\exp(\text{sim}(\mathbf{Z}_u, \mathbf{M}_u))}{\sum_j \exp(\text{sim}(\mathbf{Z}_u, \mathbf{M}_j))}, \tag{17}$$

$$\mathcal{L}_{\text{dri}}(u) = -\log \frac{\exp\big(\text{sim}(\mathbf{M}^t_u, \mathbf{M}^{t+1}_u)\big)}{\sum_j \exp\big(\text{sim}(\mathbf{M}^t_u, \mathbf{M}^{t+1}_j)\big)}. \tag{18}$$

The total contrastive loss is $\mathcal{L}_{\text{con}} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{dri}}$. The corresponding contrastive anomaly score is defined as:

$$\text{C_Score}(u) = \frac{2 - S_{\text{rec}}(u_i, t_i) - S_{\text{dri}}(u_i, t_i)}{4}, \tag{19}$$

where S_{rec} and S_{dri} are the recovery and drift similarity scores at inference time:

$$S_{\text{rec}}(u_i, t_i) = \text{sim}\Big(\mathbf{Z}^{(t)}_{u_i}, \mathbf{M}^{(t)}_{u_i}\Big), \quad S_{\text{dri}}(u_i, t_i) = \text{sim}\Big(\mathbf{M}^{(t)}_{u_i}, \mathbf{M}^{(t-1)}_{u_i}\Big). \tag{20}$$

Since cosine similarity lies in $[-1, 1]$, the quantity $2 - S_{\text{rec}} - S_{\text{dri}}$ ranges in $[0, 4]$, and dividing by 4 normalizes the score to $[0, 1]$, where a value closer to 1 indicates stronger anomalousness.

Adaptive Uncertainty Fusion. The four uncertainty components output by the feature evidence network and the topological evidence network are dynamically weighted via learnable parameters to obtain a comprehensive uncertainty score:

$$\text{U_Score}(u) = \lambda_f \Big(\lambda_g \cdot u^f_{\text{graph}} + \lambda_r \cdot u^f_{\text{recon}}\Big) + \lambda_t \big(\lambda_g \cdot u^t_{\text{graph}} + \lambda_r \cdot u^t_{\text{recon}}\big), \tag{21}$$

where λ_f, λ_t, λ_g, λ_r are learnable scalar parameters controlling the relative contributions of the feature and topological evidence, as well as the relative weights of graph uncertainty and reconstruction uncertainty. The weights are initialized as positive values and regularized via weight decay to control their magnitudes during optimization.

Final Anomaly Score and Training Objective. The final anomaly score is obtained by linearly combining the contrastive score and the uncertainty score:

$$S(u) = \text{C_Score}(u) + \varphi \cdot \text{U_Score}(u), \tag{22}$$

where φ is a hyperparameter balancing the two types of signals. Here, U_Score is treated as an uncertainty-related risk signal rather than a direct estimate of anomalousness. It complements the

contrastive deviation score by highlighting predictions made under insufficient or conflicting evidence. The total training loss of the model is:

$$\mathcal{L} = \mathcal{L}_{\text{con}} + \theta \cdot \mathcal{L}_{\text{evl}}, \tag{23}$$

where θ controls the relative weight of the evidential loss in the overall optimization objective. The two objectives are jointly optimized so that the final anomaly score simultaneously accounts for both the degree of behavioral deviation and the predictive confidence.

4 EXPERIMENTS

We conduct extensive experiments to validate the effectiveness of AELAD, centering on four core research questions: 1. How does AELAD perform compared with state-of-the-art methods? 2. Do all components of AELAD contribute to the overall performance? 3. Is the evidential learning framework of AELAD stable under noisy training data? 4. How do key hyperparameters affect model performance? Experiments on complexity analysis are provided in the appendix.

4.1 EXPERIMENTAL SETUP

Datasets: We conduct experiments on six dynamic graph datasets: Wikipedia, Reddit, Bitcoin-alpha, Bitcoin-OTC, Synthetic-Hijack, and Synthetic-New. These datasets cover social networks, financial networks, and synthetic anomaly scenarios. Each dataset is chronologically split into the first 85% for training and the remaining 15% for testing.

Baselines: We compare AELAD with 11 representative anomaly detection methods, including four rule-based methods, SedanSpot (Eswaran & Faloutsos, 2018), MIDAS (Bhatia et al., 2020), F-FADE (Chang et al., 2021), and AnoEdge (Bhatia et al., 2023), and seven representation learning-based methods, JODIE (Kumar et al., 2019), DyRep (Trivedi et al., 2019), TGAT (da Xu et al., 2020), TGN (Rossi et al., 2020), SAD (Tian et al., 2023), SLADE (Lee et al., 2024), and GeneralDyG (Yang et al., 2025).

Implementation Details: Following prior work, we use the Area Under the ROC Curve (AUC) as the primary evaluation metric. AELAD is implemented in PyTorch and trained on an NVIDIA RTX 3060 GPU with 12 GB of memory. All results are averaged over 10 independent runs with different random seeds, and both the mean and standard deviation are reported.

4.2 ANOMALY DETECTION PERFORMANCE (RQ1)

Table 1 summarizes the anomaly detection performance of AELAD and 11 baseline methods on six datasets, where **bold** denotes the best result and underlined denotes the second-best. All results are reported as mean ± standard deviation over 10 independent runs with random initializations.

Table 1: Anomaly detection results (AUC %).

Method	Wikipedia	Reddit	Bitcoin-alpha	Bitcoin-OTC	Synthetic-hijack	Synthetic-new
SedanSpot (ICDM'18)	82.58 ± 1.31	59.06 ± 1.54	68.94 ± 0.78	71.43 ± 0.79	77.43 ± 1.87	78.28 ± 1.49
MIDAS (AAAI'20)	63.07 ± 2.09	60.13 ± 1.27	64.68 ± 0.84	62.43 ± 1.23	81.84 ± 1.22	82.74 ± 0.16
F-FADE (WSDM'21)	45.14 ± 0.18	49.73 ± 0.21	54.14 ± 0.11	50.96 ± 0.21	54.71 ± 0.24	53.94 ± 0.14
AnoEdge (KDD'23)	47.25 ± 0.43	48.28 ± 0.57	62.55 ± 0.47	65.64 ± 0.47	59.67 ± 2.38	61.91 ± 2.37
JODIE (KDD'19)	84.62 ± 0.39	61.45 ± 0.77	73.37 ± 1.21	68.92 ± 0.64	71.76 ± 0.89	73.81 ± 1.18
DyRep (ICLR'19)	84.68 ± 0.63	63.57 ± 0.86	73.42 ± 0.98	70.92 ± 0.74	72.01 ± 0.65	73.28 ± 0.76
TGAT (ICLR'20)	83.27 ± 0.76	65.22 ± 1.38	71.48 ± 0.87	68.52 ± 0.96	81.08 ± 0.84	82.23 ± 0.42
TGN (ICML'20)	87.41 ± 0.47	67.18 ± 0.65	69.87 ± 0.63	75.82 ± 0.43	72.81 ± 1.29	72.45 ± 0.61
SAD (IJCAI'23)	86.34 ± 0.62	68.24 ± 1.25	68.61 ± 2.42	64.62 ± 2.92	86.48 ± 1.08	84.71 ± 1.97
SLADE (KDD'24)	87.73 ± 0.47	72.08 ± 0.49	73.68 ± 0.33	72.74 ± 0.60	82.34 ± 0.25	78.66 ± 0.36
GeneralDyG (AAAI'25)	87.91 ± 0.34	71.38 ± 0.57	75.74 ± 0.51	64.69 ± 0.55	94.16 ± 0.53	89.07 ± 0.75
AELAD (Ours)	**90.83 ± 0.23**	**72.21 ± 0.46**	**80.62 ± 0.37**	**77.58 ± 0.31**	**97.21 ± 0.14**	**93.06 ± 0.28**

As shown in Table 1, AELAD achieves the best AUC on all six datasets. Rule-based methods generally perform poorly because of their predefined statistical assumptions, while general dynamic

graph representation methods provide limited improvements due to the mismatch between their learning objectives and anomaly detection. Although dedicated anomaly detection methods are more competitive, their performance remains unstable across datasets. Compared with the strongest baseline, AELAD improves the AUC by 2.92%, 4.88%, and 1.76% on Wikipedia, Bitcoin-alpha, and Bitcoin-OTC, respectively. It also outperforms GeneralDyG by 3.05% and 3.99% on Synthetic-Hijack and Synthetic-New. These consistent improvements demonstrate that separately modeling feature and topological uncertainty and adaptively fusing the resulting evidence enable AELAD to identify anomalies more effectively across diverse dynamic graph scenarios.

4.3 Ablation Study (RQ2)

To verify the necessity of each core component, we design three ablation variants: **w/o-EL** replaces the evidential learning framework with standard point estimation; **w/o-FE** removes the feature evidence branch; and **w/o-TE** removes the topology evidence branch.

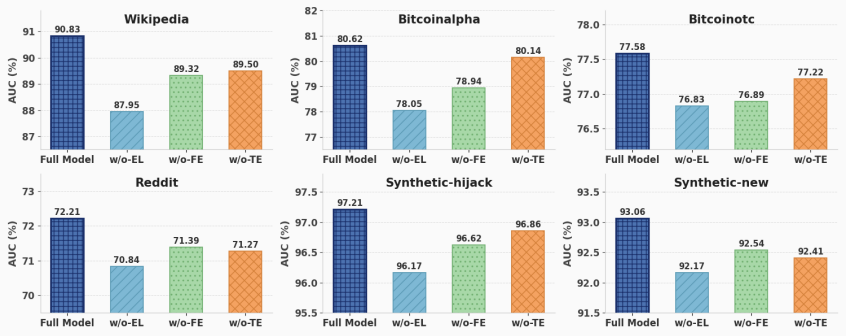


Figure 3: Ablation study results on six datasets.

As shown in Figure 3, the full model consistently achieves the best performance across all six datasets. Removing the evidential learning framework causes the largest degradation, confirming that uncertainty modeling is central to AELAD's effectiveness. Among the two evidence branches, feature evidence contributes more prominently in financial networks (e.g., a 1.68% drop on Bitcoin-alpha after removing w/o-FE), while topology evidence plays a comparably important role in Wikipedia. On synthetic datasets, both branches contribute consistently, reflecting that injected anomalies perturb feature and topological patterns simultaneously. These results validate that each component is indispensable, and their combination enables robust detection across diverse anomaly types.

4.4 Robustness Analysis (RQ3)

To evaluate detection stability under training data contamination, we inject two types of noisy nodes into the training sets of Synthetic-Hijack and Synthetic-New at ratios of 10%, 20%, 30%, and 50%: **Type 1** consists of behavior-variable normal nodes with highly variable interaction patterns, and **Type 2** consists of cold-start nodes with sparse interaction histories. We compare TGN, SLADE, and AELAD against their noise-free baselines.

As shown in Figure 4, AELAD maintains the smallest performance decline across all settings. On Synthetic-Hijack, AUC drops by only 1.1% under Type 1 (50%) noise and 4.0% under Type 2 (50%) noise, both substantially smaller than the degradation observed in TGN and SLADE. TGN is more sensitive to Type 1 noise and struggles to distinguish cold-start noisy nodes from legitimate new nodes. SLADE, while benefiting slightly from Type 1 noise on Synthetic-New, degrades monotonically under Type 2 noise due to its reliance on historical state continuity. In contrast, AELAD explicitly models feature uncertainty via the Normal-Inverse-Gamma distribution and topological uncertainty via the Beta distribution, enabling it to resist both behavior-variable and cold-start interference. These results confirm that the evidential learning framework provides consistent robustness across diverse noise types and injection ratios.

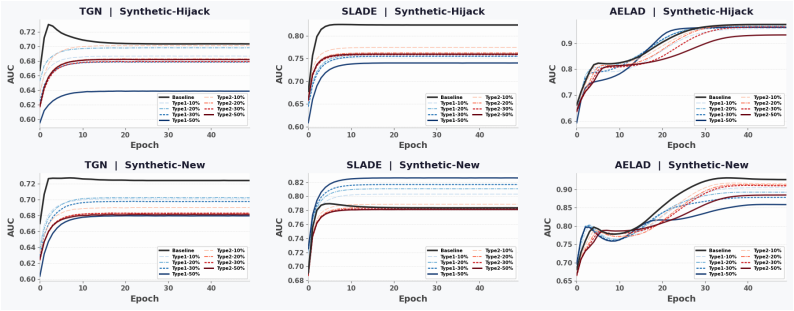


Figure 4: Robustness analysis of TGN, SLADE, and AELAD on Synthetic-Hijack and Synthetic-New under training noise injection. The black solid line denotes the noise-free baseline. Blue and red curves correspond to Type 1 (behavior-variable nodes) and Type 2 (cold-start nodes) at injection ratios of 10%, 20%, 30%, and 50%, respectively.

4.5 Hyperparameter Sensitivity Analysis (RQ4)

AELAD introduces two key hyperparameters: ζ controls the weight between feature evidence loss $\mathcal{L}_{\text{NIG}}$ and topology evidence loss $\mathcal{L}_{\text{Beta}}$, while η balances the two KL regularization terms $\mathcal{L}_{\text{reg}}^{f}$ and $\mathcal{L}_{\text{reg}}^{t}$. We conduct a grid search over $\zeta, \eta \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ on four datasets and present results as heatmaps in Figure 5.

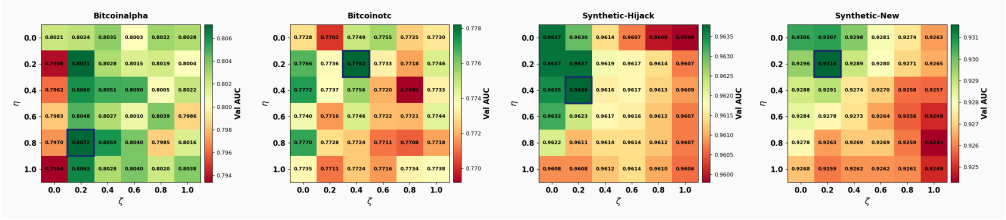


Figure 5: Sensitivity analysis with respect to ζ and η on four datasets. Colors range from red (low AUC) to green (high AUC); the optimal configuration is highlighted with a blue border.

As shown in Figure 5, the best AUC is consistently achieved under small ζ values, with optimal configurations at $(\zeta, \eta) = (0.2, 0.8)$, $(0.4, 0.2)$, $(0.2, 0.4)$, and $(0.2, 0.2)$ on Bitcoin-alpha, Bitcoin-OTC, Synthetic-Hijack, and Synthetic-New, respectively. This indicates that topology evidence loss contributes more to detection performance, while completely discarding feature evidence causes a noticeable AUC drop across all datasets. The model is less sensitive to η, though setting it to an extreme value (0.0 or 1.0) consistently yields slightly lower AUC, confirming that both KL regularization terms jointly suppress spuriously high-confidence predictions. Overall, a small ζ combined with a moderate η achieves optimal or near-optimal performance across all datasets.

5 Conclusion

In this paper, we propose AELAD, an evidential learning framework for anomaly detection in dynamic graphs. Existing methods rely on point estimation and lack uncertainty awareness, making them unable to characterize prediction reliability for behavior-variable users and cold-start newcomers. Moreover, feature and topology anomaly signals are fundamentally different in nature, and naive concatenation causes signal interference. AELAD addresses these challenges by modeling feature-space and topology-space uncertainty through dedicated NIG and Beta distribution networks, each matched to its target signal's statistical nature, and adaptively fusing the two evidence branches with contrastive learning scores via learnable weights. Extensive experiments on six datasets demonstrate that AELAD consistently outperforms eleven competitive baselines, and robustness analysis confirms its stability under training noise. In the future, we plan to extend AELAD to edge-level and subgraph-level anomaly detection in dynamic graphs.

AI USE STATEMENT

In this work, we used generative AI tools (ChatGPT, Claude) for writing assistance, including polishing and paraphrasing manuscript text. We have not used generative AI tools for generating research ideas, producing experimental results, or drawing conclusions, and automated code generation is not applicable to this work. We have reviewed all AI-assisted text to ensure accuracy and consistency with our experimental findings. All AI-assisted writing was verified by at least two authors for correctness and alignment with the original intent. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide the following resources. The proposed AELAD framework, including all model architectures and training procedures, is described in detail in Section 3. Hyperparameter settings and experimental configurations are reported in Section 4. All datasets used in this work are publicly available benchmarks. The source code for AELAD is submitted as anonymous supplementary material and will be made publicly available upon acceptance.

REFERENCES

Guoguo Ai, Hezhe Qiao, Hui Yan, and Guansong Pang. Semi-supervised graph anomaly detection via robust homophily learning. *Advances in Neural Information Processing Systems*, 38:133988–134013, 2026.

Siddharth Bhatia, Bryan Hooi, Minji Yoon, Kijung Shin, and Christos Faloutsos. Midas: Microcluster-based detector of anomalies in edge streams. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 3242–3249, 2020.

Siddharth Bhatia, Mohit Wadhwa, Kenji Kawaguchi, Neil Shah, Philip S Yu, and Bryan Hooi. Sketch-based anomaly detection in streaming graphs. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 93–104, 2023.

Yen-Yu Chang, Pan Li, Rok Sosic, MH Afifi, Marco Schweighauser, and Jure Leskovec. F-fade: Frequency factorization for anomaly detection in edge streams. In *Proceedings of the 14th ACM international conference on web search and data mining*, pp. 589–597, 2021.

da Xu, chuanwei ruan, evren korpeoglu, sushant kumar, and kannan achan. Inductive representation learning on temporal graphs. In *International Conference on Learning Representations (ICLR)*, 2020.

Dhivya Eswaran and Christos Faloutsos. Sedanspot: Detecting anomalies in edge streams. In *2018 IEEE International conference on data mining (ICDM)*, pp. 953–958. IEEE, 2018.

Soheila Farokhi, Xiaojun Qi, and Hamid Karimi. Tawrmac: A novel dynamic graph representation learning method. In *Proceedings of the ACM Web Conference 2026*, pp. 684–695, 2026.

Yun Fu, Che Zhou, Jintang Li, and Liang Chen. Long-term evolutionary patterns matter: Self-supervised anomaly detection on dynamic graphs. *Knowledge-Based Systems*, 311:113049, 2025.

Wei Ju, Siyu Yi, Zhengyang Mao, Yifang Qin, Yifan Wang, Zhiping Xiao, Yiwei Fu, Ziyue Qiao, and Ming Zhang. Long-tailed recognition of evidential experts for graph-level classification. In *Proceedings of the ACM Web Conference 2026*, pp. 1286–1297, 2026.

Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1269–1278, 2019.

Jongha Lee, Sunwoo Kim, and Kijung Shin. Slade: Detecting dynamic anomalies in edge streams without labels via self-supervised learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1506–1517, 2024.

Guanghua Liu, Jia Zhang, Peng Lv, Chenlong Wang, Huan Wang, and Di Wang. Taad: Time-varying adversarial anomaly detection in dynamic graphs. *Information Processing & Management*, 62(1): 103912, 2025.

Yixin Liu, Shirui Pan, Yu Guang Wang, Fei Xiong, Liang Wang, Qingfeng Chen, and Vincent CS Lee. Anomaly detection in dynamic graphs via transformer. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12081–12094, 2021.

Simone Mungari, Albert Bifet, Giuseppe Manco, and Bernhard Pfahringer. Ares: Anomaly recognition model for edge streams. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 1078–1088, 2026.

Krishna Prasad Neupane, Ervine Zheng, and Qi Yu. Metaedl: Meta evidential learning for uncertainty-aware cold-start recommendations. In *2021 IEEE International Conference on Data Mining (ICDM)*, pp. 1258–1263. IEEE, 2021.

Deep Shankar Pandey, Hyomin Choi, and Qi Yu. Generalized regularized evidential deep learning models: Theory and comprehensive evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Hezhe Qiao, Hanghang Tong, Bo An, Irwin King, Charu Aggarwal, and Guansong Pang. Deep graph anomaly detection: A survey and new perspectives. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs. In *ICML 2020 Workshop on Graph Representation Learning*, 2020.

Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31, 2018.

Sheng Tian, Jihai Dong, Jintang Li, Wenlong Zhao, Xiaolong Xu, Baokun Wang, Bowen Song, Changhua Meng, Tianyi Zhang, and Liang Chen. Sad: semi-supervised anomaly detection on dynamic graphs. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 2306–2314, 2023.

Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In *International conference on learning representations*, 2019.

Huan Wang, Junyang Chen, Yirui Wu, Victor CM Leung, and Di Wang. Epm: Evolutionary perception method for anomaly detection in noisy dynamic graphs. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

Chunyu Wei, Wenji Hu, Xingjia Hao, Yunhai Wang, Yueguo Chen, Bing Bai, and Fei Wang. Graph evidential learning for anomaly detection. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 3122–3133, 2025.

Haolong Xiang and Xuyun Zhang. Edge computing empowered anomaly detection framework with dynamic insertion and deletion schemes on data streams. *World Wide Web*, 25(5):2163–2183, 2022.

Yu Xiao, Haolong Xiang, Xiaolong Xu, Lianyong Qi, Xuyun Zhang, Wei Fan, and Wanchun Dou. Cifad: Causal-invariant subspace learning for few-shot anomaly detection on dynamic relational graphs. In *Proceedings of the ACM Web Conference 2026*, pp. 4680–4688, 2026.

Xiao Yang, Xuejiao Zhao, and Zhiqi Shen. A generalizable anomaly detection method in dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 22001–22009, 2025.

Taeseong Yoon and Heeyoung Kim. Uncertainty estimation by flexible evidential deep learning. *Advances in Neural Information Processing Systems*, 38:118601–118641, 2026.

Jialun Zheng, Jie Liu, Jiannong Cao, Xiao Wang, Hanchen Yang, and Yankai Chen. Dp-dgad: A generalist dynamic graph anomaly detector with dynamic prototypes. In *Proceedings of the ACM Web Conference 2026*, pp. 857–868, 2026.

Li Zheng, Zhenpeng Li, Jian Li, Zhao Li, and Jun Gao. Addgraph: anomaly detection in dynamic graph using attention-based temporal gcn. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 4419–4425, 2019.

A Appendix

A.1 Notation Summary

Table 2 summarizes the main mathematical notations used throughout this paper.

Table 2: Summary of main notations.

Notation	Description
$\mathcal{G}$	Dynamic graph
$\mathcal{V}$	Set of nodes
N	Total number of interaction events
$u_i,\ v_i$	Source and destination nodes of the i-th interaction
t_i	Timestamp of the i-th interaction
$\mathbf{x}_v \in \mathbb{R}^d$	Feature vector of node v
d	Feature / embedding dimension
K	Number of sampled temporal neighbors
$\mathbf{M}(u,t) \in \mathbb{R}^d$	Memory vector of node u at time t
$\mathbf{Z}_u \in \mathbb{R}^d$	Temporal embedding of node u
$(\gamma, \nu, \alpha, \beta)$	Output parameters of the NIG distribution
$(\varepsilon^+, \varepsilon^-)$	Output parameters of the Beta distribution
$u^f_{\text{graph}},\ u^f_{\text{recon}}$	Feature-space graph and reconstruction uncertainty
$u^t_{\text{graph}},\ u^t_{\text{recon}}$	Topology-space graph and reconstruction uncertainty
$\lambda_f, \lambda_t, \lambda_g, \lambda_r$	Learnable fusion weights
$s(v,t)$	Final anomaly score of node v at time t
$\varphi,\ \theta$	Hyperparameters balancing loss components
ζ	Hyperparameter balancing NIG and Beta evidence losses
η	Hyperparameter balancing the two KL regularization terms
$S_{\text{rec}}(u,t)$	Recovery similarity score of node u at time t
$S_{\text{dri}}(u,t)$	Drift similarity score of node u at time t
$\text{C_Score}(u)$	Contrastive anomaly score of node u
$\text{U_Score}(u)$	Uncertainty anomaly score of node u

A.2 Dataset Details

A.2.1 Benchmark Datasets

We evaluate AELAD on four widely used benchmark dynamic graph datasets.

Wikipedia: Records user editing behaviors on Wikipedia pages over one month. Each interaction corresponds to a user editing a page, and the associated feature vector encodes the edit content via LIWC features. A user is labeled as anomalous at the corresponding timestamp if they are

subsequently banned following an edit. The anomaly ratio is extremely low (0.14%), making this a highly imbalanced detection task.

Reddit: Comprises posts submitted by users to subreddits over one month. Each interaction represents a user posting to a subreddit, with LIWC-based feature vectors attached. A user's dynamic label reflects whether they were banned after a given post. The anomaly ratio is even lower than Wikipedia (0.05%), posing a greater challenge for detection under severe class imbalance.

Bitcoin-alpha: A trust-weighted directed network on the Bitcoin trading platform, where users rate each other on a scale from -10 to $+10$. Each rating event constitutes a temporal interaction. We construct dynamic anomaly labels based on the cumulative ratings received by each user: users with a sufficiently negative cumulative score are labeled as anomalous at the time of the rating that causes their score to cross the threshold.

Bitcoin-OTC: Another Bitcoin trust network sharing the same rating structure as Bitcoin-alpha, but with a larger scale and a higher anomaly ratio (7.22%). The denser anomaly distribution makes it complementary to the sparse-anomaly settings of Wikipedia and Reddit.

A.2.2 Semi-Synthetic Datasets

We further adopt two semi-synthetic datasets derived from the real-world Email-EU dataset to evaluate the model's ability to detect different types of anomalies.

Synthetic-Hijack: Simulates account hijacking scenarios. Normal nodes are sampled from the Email-EU network. Anomalous nodes are constructed by replacing a normal node's future interactions with those sampled from a different node, simulating the behavioral shift caused by account takeover.

Synthetic-New: Simulates newly emerging malicious nodes. Anomalous nodes are injected as fresh entities with interaction patterns sampled from high-degree nodes in the network, simulating spammers or bots that rapidly establish dense connections upon joining.

Dataset statistics are summarized in Table 3.

Table 3: Statistics of experimental datasets.

Dataset	Nodes	Edges	Feat. Dim	Anomalies	Anomaly Ratio	Anomaly Type
Wikipedia	9,227	157,474	172	217	0.14%	Edit-ban
Reddit	10,984	672,447	172	366	0.05%	Post-ban
Bitcoin-alpha	3,783	24,186	0	874	3.61%	Untrusted user
Bitcoin-OTC	5,881	35,592	0	2,568	7.22%	Untrusted user
Synthetic-Hijack	986	333,200	0	3,290	0.99%	Spammer
Synthetic-New	996	333,200	0	3,290	0.99%	Spammer

A.2.3 Data Splitting and Noise Injection

All datasets are split chronologically: the first 85% of interactions are used for training and the remaining 15% for testing. This strict temporal split prevents any future information from leaking into the training phase.

For the robustness analysis in Section 4, two types of noisy nodes are separately injected into the training sets of Synthetic-Hijack and Synthetic-New at ratios of 10%, 20%, 30%, and 50%: **Type 1 (Behavior-Variable Nodes):** Normal nodes whose interaction patterns exhibit high temporal variability. Their embeddings drift significantly from cluster centers, mimicking the behavior of active traders or trending-topic users. **Type 2 (Cold-Start Nodes):** Newly joined nodes with sparse interaction histories and limited neighborhood structures, resulting in poorly initialized embeddings.

A.3 HYPERPARAMETER SETTINGS

A.3.1 GENERAL HYPERPARAMETERS

Table 4 lists the general hyperparameter settings used across all experiments. Unless otherwise noted, these settings are fixed for all six datasets.

Table 4: General hyperparameter settings of AELAD.

Hyperparameter	**Value**
Batch size	1,000
Learning rate	3×10^{-6}
Memory dimension	256
Message dimension	128
Number of temporal neighbors K	10
Training epochs	50
Learning rate decay factor	0.7
Weight decay	10^{-4}
Dropout probability	0.1
φ (contrastive–uncertainty balance)	0.05
θ (evidential loss weight)	0.0001

A.3.2 DATASET-SPECIFIC OPTIMAL CONFIGURATIONS FOR ζ AND η

The hyperparameters ζ and η control the balance between the NIG and Beta evidence losses, and between the two KL regularization terms, respectively. Table 5 reports the optimal (ζ, η) configurations identified via grid search over $\{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ on four datasets. The sensitivity analysis is presented in Figure 5.

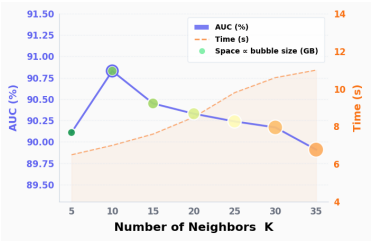
Table 5: Optimal (ζ, η) configurations per dataset.

Dataset	ζ	η
Bitcoin-alpha	0.2	0.8
Bitcoin-OTC	0.4	0.2
Synthetic-Hijack	0.2	0.4
Synthetic-New	0.2	0.2

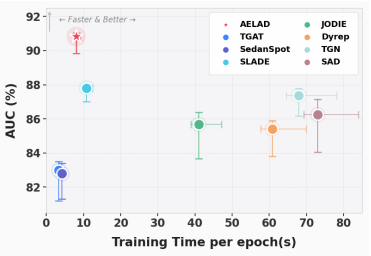
Across all datasets, the optimal ζ is consistently small (≤ 0.4), indicating that the topology evidence loss $\mathcal{L}_{\text{Beta}}$ contributes more to detection performance than the feature evidence loss $\mathcal{L}_{\text{NIG}}$. The model is less sensitive to η, though extreme values (0.0 or 1.0) consistently yield slightly lower AUC, confirming that both KL regularization terms are necessary to suppress spuriously high-confidence predictions.

A.4 COMPLEXITY ANALYSIS (RQ5)

We evaluate the computational efficiency of AELAD on Wikipedia from two perspectives: the effect of temporal neighbor count K on performance and overhead, and an overall efficiency comparison against baselines.



(a) Performance and complexity under varying K.



(b) Per-epoch training time vs. AUC across methods.

Figure 6: Complexity analysis of AELAD on the Wikipedia dataset.

As shown in Figure 6a, AUC peaks at $K = 10$ (90.83%) and decreases monotonically to 89.91% at $K = 35$, suggesting that excessive neighbors introduce redundant noisy interactions that interfere with evidence modeling. Training time grows approximately linearly from 6.5s to 11.0s, while memory overhead remains minimal, confirming that AELAD's space complexity is largely insensitive to K. As shown in Figure 6b, AELAD occupies the upper-left region of the time–AUC plot, achieving the highest AUC with the shortest training time among all compared methods. These results demonstrate that the evidential learning module introduces negligible computational overhead while delivering superior detection accuracy.