Engineering Applications of Artificial Intelligence 179 (2026) 115181

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Research paper

Uncertainty-aware dynamic graph contrastive learning for node classification and anomaly detection

Long Xu, Zhiqiang Pan, Honghui Chen *

National Key Laboratory of Information Systems Engineering Changsha, 410000, China

ARTICLE INFO

Keywords:
Dynamic graph learning
Contrastive learning
Temporal uncertainty
Reliable contrastive pairs
Anomaly detection

ABSTRACT

Dynamic graph contrastive learning methods commonly assume that nodes preserve stable structural roles over time. However, real-world networks often violate this assumption: nodes may change their connectivity patterns, reactivate after long inactivity, or undergo local structural reorganisation. Such heterogeneity not only weakens temporal consistency constraints but also causes volatile nodes to generate unreliable negative signals. This work shows that temporal contrasts vary substantially in reliability and should not be treated uniformly.

We introduce an uncertainty-aware dynamic graph contrastive learning framework driven by temporal contrast uncertainty, which quantifies the reliability of cross-temporal contrasts directly from the graph structure. Nodes with low temporal contrast uncertainty exhibit stable structural patterns, whereas nodes with high temporal contrast uncertainty indicate volatile evolution. Leveraging temporal contrast uncertainty, the proposed framework integrates three mechanisms: (1) adaptive temporal sampling, (2) uncertainty-weighted contrastive objectives, and (3) group-level prototype calibration to correct estimation bias. We theoretically show that temporal contrast uncertainty weighting yields a favourable trade-off with respect to the mutual information lower bound, while prototype calibration provides an upper bound for controlling temporal drift. Extensive experiments on five benchmark datasets for node classification and seven datasets (including two with real-world anomaly labels) for anomaly detection demonstrate that the proposed framework consistently outperforms state-of-the-art methods, with pronounced improvements on datasets containing a high proportion of structurally volatile nodes.

1. Introduction

Real-world graph data evolve continuously. In financial trust networks, a retail trader may rapidly escalate into a high-volume wholesaler, triggering fraud-like structural signatures. In citation networks, a once-obscure paper may suddenly attract citations across diverse research communities. In social platforms, coordinated bot accounts exhibit irregular activation cycles that mimic legitimate users. These dynamics make node classification and anomaly detection particularly challenging, as a node's structural role at one timestamp may be fundamentally incompatible with its role at another (Wu et al., 2020). These processes lead to heterogeneous and asynchronous changes in nodes' structural roles (Feng et al., 2026). Although contrastive learning has become a mainstream paradigm for self-supervised graph representation learning, existing dynamic graph contrastive methods largely adopt a uniform cross-temporal comparison strategy (Ju et al., 2024; Zheng et al., 2025). They implicitly assume that all nodes should maintain consistent temporal representations and that all negative samples provide equally useful contrastive signals.

However, this assumption rarely holds in dynamic environments. Nodes can exhibit vastly different structural evolution patterns: some remain stable over time, whereas others undergo substantial shifts in degree, neighbourhood composition, and role. Consequently, the reliability of temporal comparisons varies significantly. For structurally stable nodes, long-term comparisons offer strong self-supervision, whereas for rapidly evolving nodes, enforcing temporal invariance introduces noise. Moreover, when such nodes are used as negative samples, their inconsistent states may generate misleading signals that hinder representation learning.

As illustrated in Fig. 1, these challenges are evident in real applications. For instance, a previously inactive social media user may become a highly connected content creator, or a local merchant may transform into a cross-border wholesaler, resulting in abrupt changes in neighbourhood density and heterogeneity. Similarly, a once-obscure paper may suddenly gain prominence and attract citations from a diverse research community. Forcing temporal consistency in these cases conflates fundamentally different structural roles and amplifies false

* Corresponding author.
E-mail address: 3468349672@qq.com (H. Chen).

https://doi.org/10.1016/j.engappai.2026.115181
Received 3 December 2025; Received in revised form 28 April 2026; Accepted 19 May 2026
Available online 25 May 2026

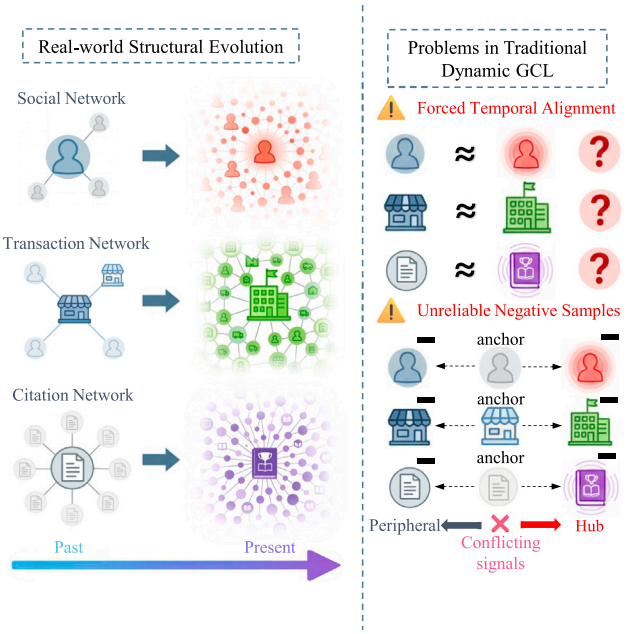


Fig. 1. Heterogeneous structural evolution in real-world networks (left) and fundamental limitations of existing dynamic graph contrastive learning methods for node classification and anomaly detection (right).

negatives (Abdar et al., 2021). Despite their prevalence, such issues remain largely underexplored in current dynamic graph contrastive learning.

Recent studies attempt to improve dynamic contrastive learning through temporal augmentations, recurrent architectures, or attention-based encoders (Sankar et al., 2020; Liu and Liu, 2021; Xu et al., 2024). Yet these methods still employ standard InfoNCE objectives and treat all temporal contrasts uniformly. They lack explicit modelling of contrast reliability and rely on fixed heuristics or encoder-level temporal modelling that is inconsistent with the loss-level invariance assumption. A key open question thus emerges: how can we quantify the credibility of temporal contrasts in an unsupervised manner and adaptively adjust their influence during training?

To address this problem, we propose UDGCL, an uncertainty-aware dynamic graph contrastive learning framework that explicitly incorporates temporal contrast reliability into the learning objective. Central to our approach is the temporal contrast uncertainty(TCU) metric, which is derived from raw graph topology and jointly captures structural entropy and the temporal coefficient of variation to estimate node-level evolution rates. Based on TCU, UDGCL introduces a hierarchical mechanism that operates across sampling, contrastive, and stabilisation stages. In the sampling stage, nodes are assigned adaptive temporal windows: stable nodes use broad windows to leverage long-term consistency, while rapidly evolving nodes use narrow windows to mitigate noise. In the contrastive stage, multi-scale temporal kernels modulate negative sample weights based on uncertainty, down-weighting unreliable or structurally incompatible comparisons. In the stabilisation stage, nodes with similar uncertainty levels are softly grouped and connected to learnable prototypes, providing stable temporal anchors that smooth noise and prevent long-term representation drift.

We further provide theoretical analysis from the perspectives of mutual information estimation and drift control. We show that TCU-guided weighting improves the tightness of the mutual information lower bound by effectively reducing false negatives, and that prototype-level consistency constrains temporal representation drift with guarantees governed by the temperature and group granularity. Extensive experiments on five widely used dynamic graph benchmarks demonstrate that UDGCL achieves consistent and substantial improvements over state-of-the-art baselines in node classification and anomaly detection tasks. Detailed ablation studies and visualisations further validate its effectiveness, scalability, and robustness.

Our main contributions are summarised as follows:

- We reveal an overlooked issue in dynamic graph contrastive learning: nodes exhibit highly uneven temporal comparison reliability, and we introduce TCU as an unsupervised metric to quantify comparison credibility.
- We propose the UDGCL framework, which embeds contrast reliability through adaptive sampling, uncertainty weighting, and prototype calibration. A group-specific multi-scale temporal kernel and a dual objective of node-level weighted contrast and group-level prototype consistency jointly reduce false negatives and enhance temporal semantic stability.
- We provide theoretical insights showing that TCU-based weighting tightens the mutual-information lower bound when the correction of false negatives is beneficial, while prototype-based regularisation explicitly constrains the upper bound of representation drift.
- Experiments on five benchmark datasets for node classification and seven datasets (including two real-world anomaly benchmarks, Wikipedia and MOOC) for anomaly detection demonstrate the effectiveness, scalability, and robustness of our method, with ablation studies and visual analyses verifying the contribution of each component.

2. Related work

2.1. Dynamic graph representation learning

Real-world networks evolve over time through events or snapshots, giving rise to two major paradigms for learning representations that remain stable yet responsive to temporal changes. The first is snapshot-based modelling, which partitions a dynamic graph into discrete time slices and uses graph neural networks to encode structural and attribute information within each snapshot. Temporal dependencies are then captured through mechanisms such as parameter evolution or cross-snapshot attention (Li et al., 2024). Representative approaches include parameter-evolution models (such as EvolveGCN (Pareja et al., 2020)), where encoder weights are updated over time to accommodate structural drift, and attention-based architectures (such as DySAT (Sankar et al., 2020)), which apply self-attention jointly over structural and temporal dimensions to model both intra- and inter-snapshot relationships (Bao et al., 2023). This paradigm integrates well with existing engineering pipelines, enables efficient parallelisation, and scales effectively to large graphs, making it a common choice for tasks such as node classification and link prediction.

The second paradigm, continuous-time modelling, treats interactions as timestamped event streams, emphasising causal order and fine-grained temporal resolution. Typical methods incorporate historical influences through temporal encodings and attention mechanisms (such as TGAT (Xu et al., 2020)), message-based memory modules (such as TGN (Rossi et al., 2020)), or neural point processes (such as DyRep (Trivedi et al., 2019) and HTNE (Zuo et al., 2018)). Other approaches, such as CAW (Wang et al., 2021a), enhance generalisation by preserving local causal structures via anonymised time-bounded walks. While continuous-time models excel in event-level prediction and inductive generalisation, they generally require more memory and computational resources and involve more intricate training and parameter tuning (Cong et al., 2023).

Despite the strengths of both paradigms, existing methods typically adopt a time-invariant objective, lacking mechanisms that adapt to heterogeneous structural evolution across different nodes. This limitation introduces two types of bias: for nodes undergoing substantial

structural change, enforcing strict temporal consistency can introduce noise, whereas for structurally stable nodes, long-term dependencies may be underutilised. More critically, current methods incorporate temporal modelling at the encoder stage but implicitly assume equal reliability for all temporal comparisons at the loss stage, resulting in an encoder–objective mismatch. Wang et al. (2021b) further show that anomalous nodes deviate measurably from their stable historical edge-generation mechanisms, reinforcing the view that structural volatility is a reliable indicator of abnormality. To address this gap, our work explicitly models reliability variations among temporal comparisons at the objective level. We achieve this by adaptively modulating the contribution of each temporal comparison during training, guided by uncertainty metrics derived from the original graph structure.

2.2. Graph contrastive learning

Contrastive learning has emerged as a central paradigm in self-supervised graph representation learning. For static graphs, early approaches improved discriminability by maximising local–global mutual information, as demonstrated by methods such as DGI (Veličković et al., 2019) and InfoGraph (Sun et al., 2020). Subsequent systematic developments focused on multi-view augmentation and invariance, represented by methods such as GraphCL (You et al., 2020) and MVGRL (Hassani and Khasahmadi, 2020), alongside advances that improve stability and efficiency through normalised contrastive objectives or self-guided frameworks, such as GRACE (Zhu et al., 2020), GCA (Zhu et al., 2021), BGRL (Thakoor et al., 2022), and CCA-SSG (Zhang et al., 2021). These approaches collectively established the core workflow of view construction, similarity maximisation, negative sample comparison, and yielded mature empirical practices involving augmentation strength, temperature selection, and projector-head design (Guo et al., 2024b). However, they generally assume stable positive and negative relationships throughout training and draw negative samples from a uniform distribution, overlooking temporal non-stationarity. As a result, directly transferring these methods to dynamic graphs often leads to temporal mismatch.

Extensions of contrastive learning to dynamic graphs typically incorporate temporal factors into positive–negative pair construction and encoder design. One line of work constructs cross-temporal positive pairs (the same node observed at different times) at the snapshot level and applies temporal augmentations to reinforce consistency within local time windows (Sankar et al., 2020; Liu and Liu, 2021). Another line constructs sequence fragments or time-location-sensitive views at the event level, using temporal encodings or temporal-attention-based projection heads to mitigate temporal mismatch (Xu et al., 2020; Rossi et al., 2020). Other approaches combine local and global temporal consistency to suppress long-term drift. While these efforts have collectively advanced the performance of self-supervised pre-training on dynamic graphs, they remain limited in several critical aspects (Zhu et al., 2023). Most notably, they continue to rely on uniform negative-sampling strategies, ignoring the fact that the reliability of temporal comparisons differs across nodes and temporal spans. RustGraph (Guo et al., 2024a) addresses label noise in this setting by jointly encoding structural and temporal dependencies via a variational auto-encoder with snapshot-level contrastive learning, yet still applies uniform temporal weights across nodes of different evolutionary stability. When nodes undergoing substantial structural change are used as negative samples, they produce highly uncertain comparison signals that can mislead the learning process. At the same time, fixed temporal windows fail to account for heterogeneous uncertainty levels, being overly conservative for low-uncertainty nodes and overly aggressive for those with high temporal contrastive uncertainty. HADF (Liu et al., 2025b) further reveals that structural noise from incomplete observations poses a distinct robustness challenge, proposing a hypergraph-based encoder to handle noisy pairwise edges in dynamic graphs. MHisCL (Fu et al., 2025) captures long-term evolutionary patterns via a State Space Model for self-supervised anomaly detection, treating deviation from multi-step historical trajectories as the anomaly signal.

More recently, hybrid self-supervised learning approaches have been explored to incorporate higher-order structural information for graph anomaly detection. For example, HSLGAD (Lv et al., 2026) combines motif-based generative learning with subgraph contrastive learning to capture both topological patterns and attribute-level anomalies, demonstrating that integrating higher-order structures improves anomaly identification. More broadly, graph anomaly detection has evolved along several complementary directions. Ma et al. (2023) provide a comprehensive survey of deep learning-based methods, categorising them into reconstruction-based, GAN-based, and contrastive learning-based families, while identifying key open challenges including the handling of complex topological changes in evolving networks. For dynamic graph anomaly detection, temporal attention mechanisms have proven particularly effective: AddGraph (Zheng et al., 2019) employs an attention-based temporal graph convolutional network that captures evolving structural patterns for anomalous edge detection in streaming dynamic graphs, and Zhao et al. (2026) propose an attention-based framework that jointly models temporal evolution and structural change to track behavioural deviations over time. Beyond node- and edge-level detection, subgraph-level structural anomaly detection has gained increasing attention; Wu et al. (2026) address anomaly subgraph detection across multiple associated attributed networks, highlighting the importance of multi-scale structural analysis where anomalies manifest through coordinated irregularities across groups of nodes. These advances collectively reinforce the necessity of multi-scale, temporally-aware anomaly detection, motivating our UDGCL framework that introduces temporal contrast uncertainty as a principled mechanism for quantifying node-level structural reliability.

To address these limitations, we propose analysing TCU directly from raw graph topology, enabling adaptive adjustment of contrastive strategies along three dimensions: sampling, weighting, and calibration. This formulation explicitly integrates contrast reliability into the optimisation objective.

2.3. Uncertainty-aware learning

Uncertainty estimation is widely used in machine learning for sample weighting, active learning, and robust optimisation. In semi-supervised learning, uncertainty derived from predictive entropy or consistency is commonly adopted for pseudo-label filtering and sample-weight modulation, as seen in methods such as MixMatch (Berthelot et al., 2019) and FixMatch (Sohn et al., 2020). By down-weighting low-confidence samples, these approaches mitigate confirmation bias and improve training stability. In noisy-label learning, uncertainty estimation helps identify potential annotation errors, with methods such as Co-teaching (Han et al., 2018) and DivideMix (Li et al., 2020) enhancing robustness through dynamic loss reweighting or selective sample utilisation. These techniques, however, generally rely on model outputs — such as softmax probabilities or ensemble divergence — to quantify uncertainty. This reliance introduces two fundamental challenges in self-supervised contrastive learning: first, contrastive frameworks lack explicit prediction targets, rendering conventional uncertainty definitions inapplicable. Second, estimating uncertainty from model representations or outputs creates a circular dependency, where uncertain representations are used to assess contrast reliability, which in turn guides representation learning.

Within the graph learning domain, several studies have begun to explore uncertainty-aware representation learning. Some approaches model distributional uncertainty via Bayesian graph neural networks or variational inference, such as Bayesian GCNN (Zhang et al., 2019) and VGAE (Kipf and Welling, 2016), yet these methods primarily capture node- or edge-level prediction confidence rather than the reliability of

sample-pair comparisons. Other work exploits predictive uncertainty for label propagation or sample selection in semi-supervised node classification, but largely overlooks temporal dynamics in evolving graphs. More critically, most graph-based uncertainty methods depend on model outputs or learned representations, lacking mechanisms for estimating uncertainty a priori from the raw graph structure (Huang et al., 2025; Rajaoarisoa et al., 2024). Recent dynamic graph studies have begun to explicitly incorporate uncertainty into representation learning. For example, ConUMIP (Zhang and Jiang, 2024) applies uncertainty-masked mix-up to improve continuous-time dynamic graph embeddings, HCUAN (Bai et al., 2025) introduces context-aware uncertainty to adaptively weight negative samples in heterogeneous dynamic graphs, and IB-D2GAT (Wang et al., 2026) models uncertainty-aware invariant and variant spatio-temporal patterns through disentangled dynamic graph attention and information bottleneck regularisation for out-of-distribution generalisation.

To address these limitations, we introduce TCU, which offers a structured, topology-driven estimation of sample-pair reliability for contrastive learning. TCU is derived entirely from the dynamic degree patterns of the graph, making it available prior to representation learning and thereby eliminating circular dependencies. Moreover, TCU is not limited to sample weighting, and it integrates into the entire contrastive pipeline through adaptive sampling and group-level prototype calibration, forming a synergistic denoising mechanism that embeds uncertainty awareness across all stages of dynamic graph contrastive learning.

3. Preliminaries

3.1. Notations and definitions

Given the evolution of a dynamic graph in the continuous time domain, we discretise it into T temporal snapshots $\{\mathbf{G}^{(t)}\}_{t=1}^{T}$. Each snapshot $\mathbf{G}^{(t)} = (\mathbf{V}, \mathbf{E}^{(t)}, \mathbf{X}^{(t)})$ consists of a node set $\mathbf{V}$ ($|\mathbf{V}| = N$), a set of edges $\mathbf{E}^{(t)} \subseteq \mathbf{V} \times \mathbf{V}$ at time slice t, and a node feature matrix $\mathbf{X}^{(t)} \in \mathbb{R}^{N \times d_f}$. The degree of node i at time t is denoted as $d_i^{(t)}$, and its neighbourhood set is $\mathcal{N}_i^{(t)}$. The goal of dynamic graph representation learning is to develop an encoder $f_\theta : (\mathbf{G}^{(t)}, i) \mapsto \mathbf{z}_i^{(t)} \in \mathbb{R}^d$ that maps the local structure and historical context of node i at time t into a low-dimensional representation vector. This representation should capture the node's structural role, semantic attributes, and evolutionary patterns, which are critical for downstream tasks such as node classification, link prediction, and anomaly detection.

Traditional dynamic graph contrastive learning methods generally treat the temporal dimension as a natural source of positive samples. The optimisation aims to maximise the mutual information between the representations of the same node at different time steps (Belghazi et al., 2018):

$$\max_{\theta} \sum_{i \in \mathbf{V}} \sum_{t, t' \in \mathcal{T}} I(\mathbf{Z}_i^{(t)}; \mathbf{Z}_i^{(t')}) \tag{1}$$

where $I(\cdot;\cdot)$ represents mutual information, $\mathcal{T}$ is the sampled time set, and $\mathbf{Z}_i^{(t)}$ is the random variable representing node i at time t. This objective implicitly assumes three equally weighted conditions: (1) consistent structural behaviour across time for all nodes, (2) uniform contrast strength for all negative samples, and (3) uniform effects of temporal intervals for all nodes. However, as noted in the introduction, these assumptions fail in the case of dynamically evolving heterogeneous graphs. Specifically, high-uncertainty nodes introduce noise when compared across time, unstable nodes generate unreliable contrastive signals as negative samples, and uniform temporal processing strategies fail to account for varying levels of temporal contrastive uncertainty. These limitations motivate the need for a reliability-aware reformulation of the contrastive objective, which we address in Section 4.

4. Approach

UDGCL is an end-to-end, uncertainty-aware dynamic graph contrastive learning framework that enhances the quality of the contrastive learning signal by explicitly modelling temporal contrastive uncertainty. As shown in Fig. 2, our model architecture consists of four synergistic modules. First, it analyses TCU for each node based on the degree sequence of the original graph, quantifying its reliability over time. Next, the model dynamically allocates temporal sampling windows according to TCU: nodes with low TCU use large windows to capture long-term consistency, while nodes with high TCU use smaller windows to reduce noise accumulation. TCU also modulates the negative sample weights in the node-level InfoNCE loss (Oord et al., 2018), diminishing the impact of negative samples with high TCU. Finally, cross-group attention is employed to aggregate nodes with similar TCU, softly assigning them to learnable prototypes. This process effectively smooths multi-node noise and enables adaptive calibration. The overall training workflow includes the following steps: calculating TCU and performing K-means clustering, adaptively sampling time points based on TCU, obtaining node representations via the GNN encoder, and jointly optimising node-level weighted contrast loss and group-level prototype contrast loss.

To formally incorporate contrast reliability, we introduce a temporal contrast uncertainty metric $\mathrm{TCU}_i \in [0, 1]$ for each node i, which is unsupervised and estimated from the degree dynamics $\{d_i^{(t)}\}_{t=1}^{T}$ of the original graph. Using TCU, we reformulate the optimisation objective in a reliability-weighted form:

$$\max_{\theta} \sum_{i \in \mathbf{V}} \sum_{t, t' \in \mathcal{T}} \rho(i, t, t') \cdot I(\mathbf{Z}_i^{(t)}; \mathbf{Z}_i^{(t')}) \tag{2}$$

where $\rho(i, t, t')$ denotes the comparison reliability weight for node i at the time pair (t, t'), determined by TCU, temporal interval, and multi-scale temporal dependencies. Comparisons involving nodes with high TCU (indicating structural volatility) are downweighted, while those involving nodes with low TCU (indicating structural stability) are preserved or even amplified. Furthermore, when comparing negative sample pairs (i, j), the weight $w(i, j, t, t')$ considers both the TCU of node i and that of the negative sample j. If node j exhibits structural instability, its contribution to the comparison is automatically reduced.

4.1. Temporal contrastive uncertainty

Existing dynamic graph contrastive learning methods assign equal weight to cross-temporal comparisons across all nodes, overlooking the heterogeneous uncertainty levels inherent in temporal contrasts. To illustrate, consider the degree sequences of two nodes within the time window $\mathcal{T} = \{1, 2, \ldots, 10\}$: node i exhibits a degree sequence $[18, 20, 19, 21, 20, 19, 20, 21, 19, 20]$, which remains stable around 20, while node j exhibits a strongly oscillating degree sequence $[5, 3, 45, 50, 6, 4, 48, 52, 5, 47]$. For node i, the similarity between the cross-temporal comparisons $(\mathbf{z}_i^{(1)}, \mathbf{z}_i^{(10)})$ reflects genuine structural consistency, making it reasonable to enforce proximity in their representations. However, for node j, time 1 and time 10 represent entirely distinct structural roles. Forcing proximity between these time points not only fails to learn discriminative features but also obscures the boundaries of structural transitions. More critically, when node j serves as a negative sample for other nodes, the uncertainty in its representation propagates throughout the contrastive learning process.

To address this, we propose TCU to directly quantify contrast reliability from degree sequence statistics. For node i within the time window $\mathcal{T} = \{t_1, \ldots, t_T\}$, TCU is defined as:

$$\mathrm{TCU}_i(\mathcal{T}) = \alpha \cdot H^i_{\mathrm{struct}}(\mathcal{T}) + \beta \cdot H^i_{\mathrm{temp}}(\mathcal{T}) \tag{3}$$

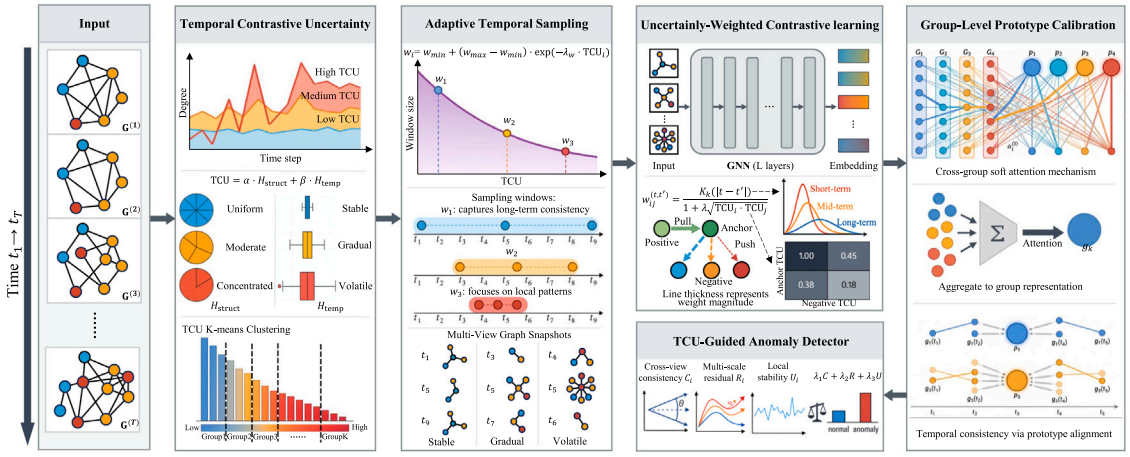


Fig. 2. Overview of the UDGCL framework, which consists of four modules: (1) TCU Computation, (2) Adaptive Temporal Sampling, (3) Uncertainty-Weighted Contrastive Learning, and (4) Group-Level Prototype Calibration. In the prototype calibration module, nodes are first softly assigned to prototypes via cross-group attention, then aggregated into group representations via attention-weighted averaging, and finally aligned with their corresponding prototypes through a group-level contrastive loss to correct TCU estimation noise and stabilise temporal representations.

where $\alpha, \beta \geq 0$ and $\alpha + \beta = 1$ are weighting hyperparameters. The structural entropy H^i_{struct} quantifies the temporal dynamics of the degree distribution based on Shannon entropy:

$$p^i_t = \frac{d^{(t)}_i}{\sum_{t' \in \mathcal{T}} d^{(t')}_i} \tag{4}$$

$$H^i_{\text{struct}}(\mathcal{T}) = -\sum_{t \in \mathcal{T}} p^i_t \log p^i_t \tag{5}$$

where $d^{(t)}_i$ denotes the degree of node i at time t, p^i_t represents the normalised distribution of degrees, and H^i_{struct} ranges from $[0, \log T]$. Structural entropy captures the uniformity of the degree distribution: higher entropy indicates highly dynamic structures, where degrees are more evenly distributed over time, while lower entropy signifies sparse activity concentrated at fewer time points.

The temporal variance coefficient H^i_{temp} quantifies the normalised degree fluctuation intensity:

$$H^i_{\text{temp}}(\mathcal{T}) = \frac{\sigma(\{d^{(t)}_i\}_{t \in \mathcal{T}})}{\mu(\{d^{(t)}_i\}_{t \in \mathcal{T}}) + \epsilon} \tag{6}$$

where $\sigma(\cdot)$ and $\mu(\cdot)$ denote the standard deviation and mean, respectively, and $\epsilon = 10^{-6}$ prevents division by zero. The coefficient of variation captures the relative amplitude of degree fluctuations, independent of their absolute magnitude. A high coefficient of variation indicates highly unpredictable behaviour and a low signal-to-noise ratio when comparing across time.

TCU is computed entirely from degree sequences and can be obtained prior to learning, thus avoiding circular dependencies. The TCU values of different nodes reflect their individual evolutionary characteristics. In experimental datasets, the TCU distribution typically exhibits a long-tail pattern: most nodes concentrate in the low-to-medium TCU range, indicating relatively stable evolution, while a minority exhibit higher TCU values, often corresponding to anomalous behaviour or identity transitions. The two components of TCU capture complementary temporal patterns: H_{struct} is sensitive to persistent dynamics, while H_{temp} is sensitive to intermittent mutations.

4.2. Adaptive temporal sampling

Traditional methods employ fixed time-sampling strategies for all nodes, making them unable to adapt to the heterogeneous temporal contrastive uncertainty across nodes (Liu et al., 2025a). Stable nodes (with low TCU) maintain similar topological roles across different time points, which makes comparisons over extended windows reliable. Therefore, large windows are assigned to fully leverage temporally consistent signals. In contrast, volatile nodes (with high TCU) exhibit significant variations in topological roles over time, and large windows may introduce semantically inconsistent pseudo-pairs. Consequently, smaller windows are assigned to minimise unreliable comparisons.

To address this, we propose a continuous, node-adaptive window allocation mechanism. For a node i, its temporal sampling window size is defined as:

$$w_i = w_{\min} + (w_{\max} - w_{\min}) \cdot \exp(-\lambda_w \cdot \text{TCU}_i) \tag{7}$$

where $w_{\min} > 0$ denotes the lower bound of the window, $w_{\max} > w_{\min}$ denotes the upper bound, and $\lambda_w > 0$ controls the sensitivity of the window size to TCU. Taking the partial derivative of the window size with respect to TCU yields:

$$\frac{\partial w_i}{\partial \text{TCU}_i} = -(w_{\max} - w_{\min})\lambda_w \exp(-\lambda_w \cdot \text{TCU}_i) < 0 \tag{8}$$

This ensures that higher TCU values correspond to smaller windows. Specifically, as $\text{TCU}_i \to 0$, $w_i \to w_{\max}$, and as $\text{TCU}_i \to \infty$, $w_i \to w_{\min}$. The exponential function guarantees continuous variation in window size, avoiding the instability inherent in threshold-based methods at boundary points.

Given a window size w_i, we sample A time points within the global time range $[t_{\min}, t_{\max}]$ to form the multi-view set $\mathcal{T}_i = \{t^i_1, \ldots, t^i_A\}$:

$$t^i_a \sim \text{Uniform}\left(\max\{t_{\min}, t_{\text{ref}} - w_i/2\}, \min\{t_{\max}, t_{\text{ref}} + w_i/2\}\right) \tag{9}$$

where $t_{\text{ref}} = (t_{\min} + t_{\max})/2$ denotes the reference time. In discrete-time snapshots, we divide the effective range into A equal intervals, randomly sample one time point from each interval, and perform deduplication.

4.3. Uncertainty-weighted contrastive learning

The standard InfoNCE loss assigns identical weights to all negative samples, ignoring the uncertainty associated with both anchors and negative samples. To address this limitation, we introduce an uncertainty-aware weight for each anchor–negative pair (i, t) and (j, t'):

$$w^{(t,t')}_{ij} = \frac{K_k(|t - t'|)}{1 + \lambda\sqrt{\text{TCU}_i \cdot \text{TCU}_j}} \tag{10}$$

Here, $K_k(\Delta t)$ denotes the multi-scale temporal kernel function corresponding to the TCU group k to which node i belongs, $\lambda > 0$ controls the

strength of the uncertainty penalty, and $\sqrt{\mathrm{TCU}_i \cdot \mathrm{TCU}_j}$ is the geometric mean of the two nodes' uncertainties. The numerator $K_k(\Delta t)$ provides a base weight based on the temporal interval Δt; intuitively, larger time intervals indicate greater divergence in node representations and thus reduced reliability. The denominator incorporates uncertainty from both nodes: the geometric mean ensures that high uncertainty from either party appropriately downweights the pair while preserving symmetry and smoothness.

In practice, we normalise all negative sample weights for each anchor (i,t) as $\tilde{w}_{ij}^{(t,t')} = w_{ij}^{(t,t')} / \sum_{j' \in \mathcal{N}} w_{ij'}^{(t,t')}$ to maintain numerical stability. The weighted InfoNCE loss is then defined as:

$$\mathcal{L}_{\text{local}} = -\mathbb{E}_{(i,t,t') \sim D} \left[\log \frac{\exp(s(\mathbf{z}_i^{(t)}, \mathbf{z}_i^{(t')})/\tau)}{\exp(s(\mathbf{z}_i^{(t)}, \mathbf{z}_i^{(t')})/\tau) + \sum_{j \in \mathcal{N}_i} \tilde{w}_{ij}^{(t,t')} \exp(s(\mathbf{z}_i^{(t)}, \mathbf{z}_j^{(t')})/\tau)} \right] \tag{11}$$

where $\mathbf{z}_i^{(t)} \in \mathbb{R}^d$ is the L_2-normalised representation of node i at time t, $s(\cdot,\cdot)$ denotes the cosine similarity, τ is the temperature parameter, and $\mathcal{N}_i$ is the negative sample set for node i.

The quality of comparison varies across nodes with differing contrast uncertainties at different time intervals. To capture these variations, we apply K-means clustering to partition nodes into K groups, $\{C_1, \ldots, C_K\} = \arg\min_{\{C_k\}} \sum_{k=1}^{K} \sum_{i \in C_k} (\mathrm{TCU}_i - \mu_k)^2$, where μ_k denotes the TCU centroid of the kth cluster. For each cluster, we then learn a multiscale Gaussian kernel mixture:

$$K_k(\Delta t) = \sum_{s=1}^{S} \alpha_{k,s} \cdot \exp\left(-\frac{(\Delta t)^2}{2\sigma_{k,s}^2} \right) \tag{12}$$

where $S = 3$ is the number of temporal scales, $\alpha_{k,s} \geq 0$ and $\sum_{s=1}^{S} \alpha_{k,s} = 1$ represent the mixture weights for scale s. The weights are parameterised via a softmax function, $\alpha_{k,s} = \exp(\theta_{k,s}^{\alpha}) / \sum_{s'} \exp(\theta_{k,s'}^{\alpha})$, and the bandwidths $\sigma_{k,s} > 0$ are ensured to remain positive through logarithmic parameterisation $\sigma_{k,s} = \exp(\theta_{k,s}^{\sigma})$. Each Gaussian kernel is a positive-definite function, and their non-negative linear combination preserves positive definiteness, ensuring the well-posedness of the resulting kernel matrix. This kernel mixture allows the model to automatically learn each TCU group's preference for different temporal scales. For example, groups with low TCU values may learn weight distributions favouring longer temporal horizons, while groups with high TCU values may place greater emphasis on shorter horizons.

4.4. Group-level prototype calibration

Statistical estimation of TCU from degree sequences inevitably introduces noise, arising mainly from three sources: boundary noise (high randomness in node assignments near TCU cluster boundaries), sparsity noise (high variance caused by sparse degree sequences), and local noise (degrees reflecting only first-order neighbourhood information). To mitigate these issues, we propose a group-level prototype calibration mechanism that smooths noise through multi-node aggregation and soft assignment. For each group $k \in \{1, \ldots, K\}$, we introduce a learnable unit-norm prototype $\mathbf{p}_k \in \mathbb{R}^d$. The group-level representation $\mathbf{g}_k^{(t)}$ at time t is obtained via attention-weighted aggregation of the representations of all nodes in the group:

$$\boldsymbol{\alpha}_i^{(t)} = \frac{\exp(\gamma \cdot s(\mathbf{z}_i^{(t)}, \mathbf{p}_k))}{\sum_{k'=1}^{K} \exp(\gamma \cdot s(\mathbf{z}_i^{(t)}, \mathbf{p}_{k'}))} \tag{13}$$

$$\mathbf{g}_k^{(t)} = \frac{\sum_{i \in C_k} \boldsymbol{\alpha}_i^{(t)} \mathbf{z}_i^{(t)}}{\left\| \sum_{i \in C_k} \boldsymbol{\alpha}_i^{(t)} \mathbf{z}_i^{(t)} \right\|_2} \tag{14}$$

where $s(\mathbf{z}_i^{(t)}, \mathbf{p}_k) = \mathbf{z}_i^{(t)\top} \mathbf{p}_k$ denotes cosine similarity and γ is the attention temperature. A key feature of this design is that the attention denominator normalises over all K prototypes rather than only the current group's prototype. This allows a node $i \in C_k$ to be softly reassigned to another group k' if its current representation $\mathbf{z}_i^{(t)}$ is more aligned with prototype $\mathbf{p}_{k'}$. For example, a node initially assigned to a low-TCU (stable) group may be reweighted toward a high-uncertainty group during training if its neighbours introduce instability into its representation. Meanwhile, weighted averaging across many nodes naturally reduces the impact of individual TCU estimation errors: if group k contains n_k nodes and most estimates are correct, the aggregated group representation is dominated by the accurately estimated members.

The group-level contrastive loss is defined as:

$$\begin{aligned} \mathcal{L}_{\text{global}} = -\frac{1}{2K|\mathcal{T}|^2} \sum_{k=1}^{K} \sum_{t,t' \in \mathcal{T}} \Bigg[& \log \frac{\exp(s(\mathbf{g}_k^{(t)}, \mathbf{p}_k)/\tau)}{\sum_{k'=1}^{K} \exp(s(\mathbf{g}_k^{(t)}, \mathbf{p}_{k'})/\tau)} \\ & + \log \frac{\exp(s(\mathbf{g}_k^{(t')}, \mathbf{p}_k)/\tau)}{\sum_{k'=1}^{K} \exp(s(\mathbf{g}_k^{(t')}, \mathbf{p}_{k'})/\tau)} \Bigg] \end{aligned} \tag{15}$$

This loss encourages group representations across different time steps to align with the same prototype, thereby reinforcing temporal consistency within each group. Simultaneously, it pushes group representations toward different prototypes across groups, enhancing inter-group separability. Notably, the prototype $\mathbf{p}_k$ does not represent the semantic centre of nodes in group k, but instead, it captures a representational pattern characteristic of the corresponding uncertainty level, learned in an end-to-end manner.

To improve training stability, we introduce two additional regularisation terms. First, kernel sparsity regularisation is applied to prevent scale collapse and bandwidth degradation in the learned kernel mixtures:

$$\mathcal{L}_{\text{kernel}} = \sum_{k=1}^{K} \left(\|\boldsymbol{\alpha}_k\|_2^2 + \|\log \boldsymbol{\sigma}_k\|_2^2 \right) \tag{16}$$

This term penalises excessively diffuse or overly concentrated kernel weights and ensures that the logarithmic bandwidth parameters remain well behaved. Second, we introduce TCU consistency regularisation:

$$\mathcal{L}_{\text{cons}} = \frac{2}{|\mathcal{T}|(|\mathcal{T}|-1)} \sum_{t<t'} \left\| \mathrm{TCU}^{(t)} - \mathrm{TCU}^{(t')} \right\|_2^2 \tag{17}$$

which encourages TCU estimates to remain consistent across different temporal sampling windows. By constraining fluctuations in TCU values, this term stabilises the downstream components that rely on TCU-driven adaptations.

The overall training objective integrates these components into the final loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{local}} + \lambda_{\text{global}} \mathcal{L}_{\text{global}} + \lambda_{\text{kernel}} \mathcal{L}_{\text{kernel}} + \lambda_{\text{cons}} \mathcal{L}_{\text{cons}} \tag{18}$$

Algorithm 1 summarises the complete UDGCL pipeline, which consists of four tightly coupled phases. Phase 1 computes TCU from degree dynamics and clusters nodes into groups with similar uncertainty levels. During training, Phase 2 adaptively samples temporal views according to each node's TCU, assigning larger windows to stable nodes and smaller windows to volatile ones. Phase 3 performs uncertainty-weighted node-level contrastive learning, where both positive and negative pairs are modulated by TCU values and the learned multi-scale temporal kernels to downweight unreliable contrasts. Phase 4 further smooths TCU estimation noise through group-level prototype alignment with soft cross-group attention. Together, these phases form a coherent framework in which TCU guides adaptive sampling and contrast weighting, while prototype calibration refines group assignments and stabilises representations, enabling the learned embeddings to capture structural roles, semantic attributes, and temporal consistency useful for downstream tasks.

4.5. TCU-guided anomaly detector

During the training phase of UDGCL, the model captures dynamic structural patterns of nodes across multiple temporal scales through multi-scale contrastive learning enabled by TCU. In the inference stage,

Algorithm 1 UDGCL Training Framework

Require: Dynamic graphs $\{\mathbf{G}^{(t)}\}_{t=1}^{T}$, hyperparameters K, A
Ensure: Node representations $\{\mathbf{z}_i^{(t)}\}$
Phase 1: TCU & Grouping
for $i \in \mathbf{V}$ **do**
  $\mathrm{TCU}_i \leftarrow \alpha H^i_{\mathrm{struct}} + \beta H^i_{\mathrm{temp}}$
end for
$\{C_k\}_{k=1}^{K} \leftarrow$ K-means($\{\mathrm{TCU}_i\}$)
Initialize prototypes $\{\mathbf{p}_k\}$ and kernels $\{K_k\}$
Phase 2-4: Training Loop
while not converged **do**
  for $i \in \mathbf{V}$ **do** ▷ Adaptive sampling
    $w_i \leftarrow w_{\min} + (w_{\max} - w_{\min})e^{-\lambda_w \mathrm{TCU}_i}$
    Sample $\mathcal{T}_i$ within window w_i
    $\mathbf{z}_i^{(t)} \leftarrow g_\phi(f_\theta(\mathbf{G}^{(t)}, i)), \forall t \in \mathcal{T}_i$
  end for
  Compute $w_{ij}^{(t,t')} \leftarrow \frac{K_k(\Delta t)}{1+\lambda\sqrt{\mathrm{TCU}_i \mathrm{TCU}_j}}$
  $\mathcal{L}_{\mathrm{local}} \leftarrow$ weighted InfoNCE loss
  for $k = 1, \ldots, K$ **do** ▷ Prototype calibration
    $\mathbf{g}_k^{(t)} \leftarrow \mathrm{Aggregate}(\{\mathbf{z}_i^{(t)}\}_{i \in C_k})$
  end for
  Compute $\mathcal{L}_{\mathrm{global}}, \mathcal{L}_{\mathrm{kernel}}, \mathcal{L}_{\mathrm{cons}}$
  Update parameters via $\nabla \mathcal{L}_{\mathrm{total}}$
end whilereturn $\{\mathbf{z}_i^{(t)}\}$

we leverage the learned TCU distribution and its associated representational patterns to design an unsupervised anomaly detector that assigns an anomaly score to each node. The core intuition is that normal nodes should maintain stable TCU and produce consistent representations across temporal views that conform to the multi-scale prior of their TCU group. In contrast, anomalous nodes typically exhibit irregular TCU fluctuations, pronounced representational shifts across views, or substantial deviations from the learned group prior.

We construct the anomaly score from three complementary dimensions. First, we freeze the trained encoder f_θ and projection head g_ϕ. For each node i, we obtain a set of normalised representations $\mathbf{z}_i^{(\mathbf{v}_1)}, \ldots, \mathbf{z}_i^{(\mathbf{v}_V)}$ from the views $\mathbf{v}_1, \ldots, \mathbf{v}_V$ sampled across V distinct time windows. View sampling follows the same adaptive strategy used during training, in which time window sizes are determined by the node's TCU value. We employ cosine dissimilarity, $D(x, y) = 1 - \langle x, y \rangle$, as the fundamental metric for measuring representation divergence.

The first metric, cross-view consistency, quantifies the extent to which a node's representations vary across different temporal scales. For each view pair $(\mathbf{v}_p, \mathbf{v}_q)$, their temporal scale difference is defined as $\Delta(p, q) = |h_p - h_q|$, where h_p denotes the time window size of view $\mathbf{v}_p$. Since views at similar scales are expected to yield similar representations, we assign higher weights to pairs with smaller scale differences using $w(p, q) = \exp(-\Delta(p, q)/\tau_d)$, where $\tau_d > 0$ controls the decay rate. The weighted consistency deviation for node i is then given by

$$C_i = \frac{\sum_{p<q} w(p,q) \cdot D(\mathbf{z}_i^{(\mathbf{v}_p)}, \mathbf{z}_i^{(\mathbf{v}_q)})}{\sum_{p<q} w(p,q)} \tag{19}$$

A normal node should produce highly similar representations across comparable temporal scales, leading to a small C_i. In contrast, anomalous nodes — due to sudden structural disruptions or behavioural changes — tend to show larger representational fluctuations, resulting in an elevated C_i.

The second metric, multi-scale prior residual, evaluates the deviation from the group-level expected similarity patterns captured during training. Each TCU group k is associated with a multi-scale kernel $K_k(\cdot)$ that encodes the expected similarity prior. For a scale difference $\Delta(p, q)$, the expected similarity is $\mu(\Delta(p, q)) = K_k(\Delta(p, q))$. The prior residual is defined as

$$R_i = \frac{2}{V(V-1)} \sum_{p<q} \left| \langle \mathbf{z}_i^{(\mathbf{v}_p)}, \mathbf{z}_i^{(\mathbf{v}_q)} \rangle - \mu(\Delta(p, q)) \right| \tag{20}$$

This metric measures how much a node's cross-view similarities deviate from the multi-scale prior learned by its TCU group. Its key strength is providing a population-level benchmark. If a node exhibits temporal patterns that diverge substantially from others in its TCU group, regardless of whether its TCU is low, R_i will increase significantly. Conversely, even nodes with high TCU may yield small R_i when their temporal behaviour aligns with the group's learned multi-scale prior, thus avoiding misclassifying nodes with high but expected uncertainty as anomalies.

Local context stability assesses the consistency of a node's relationships with its neighbourhood. Anomalies often manifest as abrupt changes in local structure, such as a fraudulent node suddenly connecting to numerous anomalous accounts or a dormant node being activated, altering its neighbourhood pattern. For each view v, we aggregate the neighbourhood representation using the adjacency matrix $\mathbf{A}^{(v)}$ to obtain the context vector:

$$\mathbf{m}_i^{(v)} = \begin{cases} \mathrm{Normalize}\left(\sum_{j \in \mathcal{N}_i^{(v)}} \mathbf{A}_{ij}^{(v)} \mathbf{z}_j^{(v)}\right) & \text{if } |\mathcal{N}_i^{(v)}| > 0 \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

Here, $\mathcal{N}_i^{(v)} = \{j : \mathbf{A}_{ij}^{(v)} > 0\}$ denotes the neighbourhood set of node i in view v, and $A_{ij}^{(v)}$ represents the adjacency weight. The node-context alignment degree is computed as $a_i^{(v)} = \langle \mathbf{z}_i^{(v)}, \mathbf{m}_i^{(v)} \rangle$, which measures the similarity between the node's representation and the aggregated representation of its neighbourhood. The variance of alignment degrees across views is given by:

$$U_i = \frac{1}{V} \sum_{v=1}^{V} (a_i^{(v)} - \bar{a}_i)^2, \quad \bar{a}_i = \frac{1}{V} \sum_{v=1}^{V} a_i^{(v)} \tag{22}$$

U_i captures fluctuations in a node's relationship with its local structure across different temporal views. Normal nodes typically maintain stable relationships with their neighbourhood, exhibiting smooth variations in alignment across views. In contrast, anomalous nodes experience abrupt changes in their neighbourhood relationships or structural disruptions, resulting in sharp fluctuations in alignment and consequently an increased U_i.

To eliminate dimensional effects and make the three metrics comparable, we standardise C_i, R_i, and U_i using z-score normalisation across all nodes:

$$\hat{C}_i = \frac{C_i - \mu_C}{\sigma_C}, \quad \hat{R}_i = \frac{R_i - \mu_R}{\sigma_R}, \quad \hat{U}_i = \frac{U_i - \mu_U}{\sigma_U} \tag{23}$$

Here, μ_C, σ_C denote the mean and standard deviation of the respective metrics across all nodes. The final anomaly score is computed by linearly combining these standardised metrics:

$$S_i = \lambda_1 \hat{C}_i + \lambda_2 \hat{R}_i + \lambda_3 \hat{U}_i \tag{24}$$

where $\lambda_1, \lambda_2, \lambda_3 \geq 0$ are weighting hyperparameters satisfying $\lambda_1 + \lambda_2 + \lambda_3 = 1$, ensuring the interpretability of the score.

4.6. Theoretical analysis

We provide theoretical guarantees for the core mechanism of UDGCL from two complementary perspectives: mutual information estimation and drift-control characterisation.

4.6.1. Theorem 1: Uncertainty-weighted contrastive bound

Let p_j denote the true negative-sample distribution of node j, and q_j denote the sampling distribution. Then the mutual information lower

bound for the weighted InfoNCE loss $\mathcal{L}_{\text{local}}$ is given by:

$$I(\mathbf{Z}_i^{(t)};\mathbf{Z}_i^{(t')}) \geq \log N - \mathcal{L}_{\text{local}} - \mathbb{E}_{j\sim q}\left[\frac{p_j}{q_j}\cdot \Delta_{ij}\right] \tag{25}$$

where $\Delta_{ij} = |w_{ij}^{(t,t')} - 1|$ represents the importance-sampling correction term.

Proof. We begin with the mutual information lower bound of the standard InfoNCE objective:

$$I(\mathbf{Z}_i^{(t)};\mathbf{Z}_i^{(t')}) \geq \log N - \mathcal{L}_{\text{InfoNCE}} \tag{26}$$

After introducing TCU-weighted sampling, importance sampling theory implies that an additional correction term must be incorporated:

$$\mathbb{E}_{j\sim q}\left[\frac{p_j}{q_j}\cdot |w_{ij}^{(t,t')} - 1|\right] \tag{27}$$

For contrast pairs that exhibit high representational uncertainty, the reliability of their contrastive signals becomes limited. In particular, when either the anchor node i or the negative sample j experiences substantial structural evolution, the stability of their cross-temporal representations degrades accordingly. Contrast pairs formed from such nodes tend to introduce gradient signals with large variance. To mitigate this issue, the weighting mechanism applies the geometric mean $\sqrt{\text{TCU}_i \cdot \text{TCU}_j}$, which adaptively suppresses the contribution of any pair whenever one of the TCU values is significantly elevated. Let the minimum TCU of at least one node in pseudo-negative pairs be $\delta > 0$, and let these pairs constitute a proportion ϵ. Under these assumptions, the correction term admits the following upper bound:

$$\mathbb{E}_{j\sim q}\left[\frac{p_j}{q_j}\cdot |w_{ij}^{(t,t')} - 1|\right] \leq \epsilon \cdot \frac{1}{1+\lambda\delta} \tag{28}$$

When λ or δ is sufficiently large, the upper bound approaches zero, effectively suppressing the influence of pseudo-negatives. Conversely, for reliable negative samples, where both nodes have low TCU values, $w_{ij}^{(t,t')} \approx 1$, and thus the original contrast strength is preserved.

4.6.2. Theorem 2: Prototype-induced drift control

By optimising the group-level comparison loss $\mathcal{L}_{\text{global}}$, the cross-temporal representation drift of the kth group admits the following upper bound:

$$\mathbb{E}_{t,t'\in\mathcal{T}}\left[\|\mathbf{g}_k^{(t)} - \mathbf{g}_k^{(t')}\|_2^2\right] \leq \frac{4\tau\mathcal{L}_{\text{global}}}{K} + \frac{2}{n_k}\sum_{i\in C_k}\text{TCU}_i \tag{29}$$

where $n_k = |C_k|$ denotes the number of nodes in group k.

Proof. When $\mathcal{L}_{\text{global}}$ is minimised, the properties of the cross-entropy objective imply that the cosine similarity between each group representation and its corresponding prototype satisfies

$$s(\mathbf{g}_k^{(t)}, \mathbf{p}_k) \geq 1 - \frac{\tau\mathcal{L}_{\text{global}}}{K} \tag{30}$$

Using the relationship between cosine similarity and squared Euclidean distance, $s(x,y) = 1 - 1/2\|x-y\|_2^2$, we obtain $\|\mathbf{g}_k^{(t)} - \mathbf{p}_k\|_2^2 \leq 2\tau\mathcal{L}_{\text{global}}/K$. Applying the triangle inequality gives

$$\|\mathbf{g}_k^{(t)} - \mathbf{g}_k^{(t')}\|_2 \leq \|\mathbf{g}_k^{(t)} - \mathbf{p}_k\|_2 + \|\mathbf{p}_k - \mathbf{g}_k^{(t')}\|_2 \leq 2\sqrt{\frac{2\tau\mathcal{L}_{\text{global}}}{K}} \tag{31}$$

Squaring both sides yields the first term $4\tau\mathcal{L}_{\text{global}}/K$. The second term reflects the influence of TCU values within the group. Under attention-weighted aggregation, the variance of the group representation satisfies

$$\text{Var}(\mathbf{g}_k^{(t)}) \leq \frac{1}{n_k}\sum_{i\in C_k}\text{Var}(\mathbf{z}_i^{(t)}). \tag{32}$$

By definition, $\text{Var}(\mathbf{z}_i^{(t)}) \propto \text{TCU}_i$. Consequently, the cross-temporal variance contribution of group k is given by $2/n_k \sum_{i\in C_k}\text{TCU}_i$.

4.7. Complexity analysis

The training process of UDGCL consists of four stages: TCU computation $O(NT)$, K-means clustering $O(NKI_{\text{km}})$, graph encoding $O(NAL|E|d)$, and contrastive learning $O(BA^2|N_i|d + NKd + K^2A^2d)$. Here, N denotes the number of nodes, T the number of temporal snapshots, K the number of clusters, A the number of sampled views, L the number of GNN layers, $|E|$ the number of edges, d the embedding dimension, B the batch size, $|N_i|$ the number of negative samples, and I_{km} the number of K-means iterations. Since T, K, A, and L are small constants with $K \ll N$, the dominant computational terms arise from graph encoding $O(NAL|E|d)$ and node-level comparison $O(BA^2|N_i|d)$. Both TCU computation and K-means clustering are executed only once at the beginning of training, making their amortised cost negligible. Moreover, the adaptive sampling mechanism reduces the effective number of temporal views from T to A (with $A \ll T$), ensuring that the model's complexity remains insensitive to the temporal span.

Regarding spatial complexity, UDGCL stores node representations $O(NAd)$, TCU values and group labels $O(N)$, group prototypes $O(Kd)$, kernel parameters $O(KS)$, GNN parameters $O(Ld^2)$, and the graph structure $O(|E|)$. The total spatial cost is therefore $O(NAd + Ld^2 + |E|)$, dominated by $O(NAd)$ and $O(|E|)$. Compared with approaches that require retaining the full time series, UDGCL substantially reduces memory consumption through adaptive sampling. All model components can be parallelised across the node dimension, making the framework highly suitable for GPU acceleration.

5. Experiments

We structure the experimental evaluation around four research questions. First, we assess whether UDGCL can outperform state-of-the-art baselines in unsupervised dynamic graph representation learning (Section 6.1). Second, we examine the contribution of each key module through ablation studies (Section 6.2). Third, we evaluate how effectively UDGCL handles nodes with different TCU levels, particularly those exhibiting extremely high or extremely low uncertainty (Section 6.3). Fourth, we analyse the sensitivity of the model's performance to variations in hyperparameter settings (Section 6.4).

5.1. Datasets

We evaluate UDGCL on seven publicly available real-world datasets. Five datasets (Bitcoinotc, BITotc, TAX51, DBLP, and Reddit) are used for both node classification and anomaly detection with injected anomaly labels. Two additional datasets (Wikipedia and MOOC) carry real-world ground-truth anomaly labels and are used exclusively for anomaly detection evaluation.

- **Bitcoinotc** (Kumar et al., 2016) is a signed and directed trust network in which nodes correspond to platform users and edges represent trust ratings exchanged among them. The dataset spans the period from November 2011 to January 2014.
- **BITotc** (Kumar et al., 2016) is a cryptocurrency trading trust network similar in nature to Bitcoinotc but differs in its collection period and preprocessing rules. Nodes denote user accounts, and edges capture trust scores derived from historical trading activities.
- **TAX51** contains tax-haven entities and corporate tax-avoidance structures. Nodes represent legal entities, individuals, or financial institutions, while edges encode equity links, control relationships, or capital flows involved in tax-planning processes.
- **DBLP** (Tang et al., 2008) is a scholarly network covering the computer science research community. Nodes represent authors, publications, or venues, and edges indicate authorship links or collaboration relationships.

Table 1
Statistics of the experimental datasets.

Datasets	Nodes	Temporal edges	Classes	Anomaly Labels
DBLP	25,387	185,480	10	Injected
Bitcoinotc	5881	35,592	3	Injected
BITotc	4863	28,473	7	Injected
TAX51	132,524	467,279	51	Injected
Reddit	898,194	2,575,464	3	Injected
Wikipedia	9227	157,474	–	Real-world
MOOC	7074	411,749	–	Real-world

- **Reddit** (Kumar et al., 2018) is a large-scale social media interaction network. Nodes correspond to users or posts, and edges reflect commenting, replying, or other forms of interaction.
- **Wikipedia** (Kumar et al., 2019) is a temporal interaction network from Wikipedia editing activities. Nodes represent editors and pages, and edges denote edit events with LIWC features. Anomalous nodes are editors banned for vandalism or policy violations.
- **MOOC** (Kumar et al., 2019) is a temporal interaction network from online course activity logs. Nodes represent students and course units, and edges encode learning actions. Anomalous nodes are students who dropped out or showed abnormal engagement patterns.

Table 1 summarises the key statistics of all datasets. Each dataset is randomly partitioned into training, validation, and test sets with a split ratio of $0.8/0.1/0.1$ to ensure efficient and fair model training. Wikipedia and MOOC retain their original chronological splits to respect temporal ordering, following the standard evaluation protocol adopted in prior dynamic graph anomaly detection work (Tian et al., 2023)

In anomaly detection tasks, we use the same datasets as those employed for node classification. Since the original dynamic graph node-classification datasets do not contain explicit anomaly labels, we follow the standard protocol in prior work and manually inject two types of sparse node-level anomalies — structural and attribute-based — into each dataset. To ensure that the injected anomalies are realistic and consistent with common anomaly patterns, we reference established characterisations of structural irregularities. The injection procedure is described as follows.

We first construct several non-overlapping subgroups from nodes that naturally exhibit weak connectivity. Each subgroup contains m nodes, where $m \in [6, 10]$ is uniformly sampled. For each group, a single timestamp t^* is independently drawn from the global timeline. At t^*, edges are added between nodes within the group with probability $p_c \in [0.9, 1)$, forming a nearly complete dense subgraph, while all other timestamps remain unchanged. All nodes in the group are labelled as structurally anomalous at t^*. To avoid introducing excessive global perturbations, we limit the number of newly added edges per node to no more than twice its original degree and discard node pairs with abnormally high historical common-neighbour overlap.

We randomly sample a target node v_i and select a short time window W of length $L \in \{1, 2, 3\}$ to induce a sudden feature drift. In the standardised feature space, we identify candidate donor nodes for v_i by selecting those with the largest Euclidean distances from either the full node set or a random subset. The top $k = 50$ nodes are retained as candidates, and the farthest node v_j is chosen as the donor. Within window W, the features of v_i are updated as $\mathbf{x}_{v_i} = \alpha \mathbf{x}_{v_j} + (1 - \alpha)\mathbf{x}_{v_i}$ where $\alpha \in [0.85, 0.95]$ is randomly sampled, and minor Gaussian noise proportional to the feature standard deviation may be added to ensure sufficient perturbation. Outside W, the original features of v_i are preserved, and the node is labelled as an attribute anomaly within W. To maintain compatibility with static graph baselines, we apply temporal averaging to each node's post-injection feature sequence to form a single aggregated feature vector.

The number of injected anomalies is adapted to the scale of each dataset. Structurally anomalous nodes constitute approximately 0.2%-0.8% of all nodes, whereas attribute-anomalous nodes account for roughly 0.5%-1.5%. For small, medium, and large graphs, this corresponds to approximately $3 \times e^2$-$8 \times e^2$, $8 \times e^2$-$2 \times e^3$ and $3 \times e^3$-$8 \times e^3$ anomalous nodes, respectively. All injections are executed with fixed random seeds to ensure reproducibility.

5.2. Baseline summary

For node classification tasks, we select representative baseline methods spanning supervised learning, unsupervised dynamic representation learning, and contrastive learning on both static and dynamic graphs. Among classic supervised baselines, GCN (Kipf and Welling, 2017), GAT (Veličković et al., 2018), and GraphSAGE (Hamilton et al., 2017) are included. GCN integrates node and neighbourhood features through spectral graph convolutions. GAT enhances representational expressiveness by assigning adaptive attention weights to neighbours. GraphSAGE generates generalisable node embeddings using differentiable neighbour-sampling-based aggregation functions. For unsupervised methods on dynamic graphs, TGAT (Xu et al., 2020) integrates temporal encoding with attention mechanisms to model continuous-time events. DySAT (Sankar et al., 2020) applies self-attention across structural and temporal dimensions to capture evolving dependencies. MNCI (Liu and Liu, 2021) improves dynamic node representations by enforcing temporal consistency through contrastive learning. For contrastive learning on static graphs, we incorporate DGI (Veličković et al., 2019), GRACE (Zhu et al., 2020), MVGRL (Hassani and Khasahmadi, 2020), CCA-SSG (Zhang et al., 2021), and SpeGCL (Shou et al., 2025). DGI maximises local–global mutual information. GRACE performs multi-graph augmentation to strengthen cross-view consistency. MVGRL aligns representations across multiple structural views. CCA-SSG conducts multi-view self-supervised alignment via canonical correlation analysis. SpeGCL exploits Fourier-domain representations and negative-sample-driven contrastive learning. For dynamic graph contrastive learning, we include DySubC (Chen et al., 2023), CLDG++(Xu et al., 2024), and LDGC (Zhu et al., 2025). DySubC compares temporal subgraphs by modelling timestamps and node activity. CLDG++ introduces time-shift invariance through graph diffusion and multi-scale local–global comparison. LDGC enhances robustness and generalisation by mitigating overfitting and noise in highly active nodes.

For graph anomaly detection, we evaluate UDGCL against a comprehensive set of representative baselines. AEGIS (Suh et al., 2003) employs adversarial graph differentiation and anomaly-aware GNN layers. CoLA (Liu et al., 2021a) combines node–neighbourhood substructure contrastive learning with statistical estimation and supports efficient batch-wise training. ANEMONE (Jin et al., 2021) derives anomaly scores through statistical estimation over multi-scale patch–context comparisons. GRADATE (Duan et al., 2023) performs multi-view and multi-scale subgraph–subgraph comparison while jointly leveraging node–subgraph and node–node interactions for improved robustness. NetWalk (Yu et al., 2018) learns dynamic graph representations via cluster embedding and autoencoders, enabling real-time detection through reservoir sampling and incremental clustering. TADDY (Liu et al., 2021b) captures coupled spatio-temporal patterns using structural and temporal role encoding within a dynamic graph Transformer, supporting attribute-free scenarios. SAD (Tian et al., 2023) adopts semi-supervised learning with temporal memory banks and pseudo-label comparisons. SLADE (Lee et al., 2024) minimises representation drift over edge flows via self-supervision, extracting long-term patterns from short-term interactions while supporting $O(1)$ edge-level updates. CLDG++ constructs anomaly indicators through cross-temporal consistency using graph diffusion and multi-scale comparisons under temporal shift invariance. General-DyG (Yang et al.,

Table 2
The evaluation accuracy and Weighted-F1 results of the **node classification** task.

Method	DBLP		Bitcoinotc		BITotc		TAX51		Reddit	
	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)
GCN	$71.28_{\pm 0.35}$	$70.96_{\pm 0.41}$	$54.54_{\pm 1.83}$	$53.17_{\pm 1.96}$	$59.18_{\pm 2.12}$	$50.45_{\pm 2.37}$	$40.37_{\pm 1.52}$	$33.58_{\pm 1.73}$	$67.82_{\pm 0.68}$	$63.21_{\pm 0.79}$
GAT	$69.93_{\pm 0.58}$	$68.74_{\pm 0.64}$	$52.38_{\pm 2.15}$	$49.63_{\pm 2.41}$	$62.46_{\pm 1.87}$	$47.29_{\pm 2.93}$	$38.76_{\pm 1.89}$	$31.45_{\pm 2.08}$	$69.17_{\pm 0.81}$	$56.92_{\pm 1.13}$
GraphSAGE	$72.29_{\pm 0.29}$	$71.63_{\pm 0.33}$	$57.23_{\pm 1.42}$	$55.84_{\pm 1.61}$	$62.60_{\pm 1.65}$	$57.19_{\pm 1.82}$	$40.74_{\pm 1.18}$	$34.26_{\pm 1.39}$	$71.38_{\pm 0.41}$	$\mathbf{65.47}_{\pm 0.56}$
TGAT	$57.41_{\pm 1.15}$	$52.68_{\pm 1.34}$	$58.49_{\pm 2.47}$	$54.72_{\pm 2.83}$	$62.45_{\pm 2.89}$	$48.56_{\pm 3.17}$	$34.45_{\pm 2.35}$	$27.91_{\pm 2.64}$	$65.53_{\pm 1.28}$	$57.34_{\pm 1.59}$
DySAT	$54.05_{\pm 1.47}$	$51.82_{\pm 1.73}$	$51.25_{\pm 3.12}$	$47.93_{\pm 3.52}$	$61.94_{\pm 2.73}$	$49.27_{\pm 3.46}$	$23.77_{\pm 2.84}$	$21.49_{\pm 3.05}$	–	-
MNCI	$66.98_{\pm 0.62}$	$65.83_{\pm 0.74}$	$55.06_{\pm 1.68}$	$53.94_{\pm 1.79}$	$63.42_{\pm 1.94}$	$53.76_{\pm 2.18}$	$39.06_{\pm 1.57}$	$31.64_{\pm 1.86}$	$70.93_{\pm 0.59}$	$64.38_{\pm 0.77}$
DGI	$69.90_{\pm 0.67}$	$68.52_{\pm 0.71}$	$56.60_{\pm 1.79}$	$54.68_{\pm 2.03}$	$61.59_{\pm 2.18}$	$50.82_{\pm 2.49}$	$39.13_{\pm 1.64}$	$32.07_{\pm 1.91}$	$70.48_{\pm 0.71}$	$59.76_{\pm 0.98}$
GRACE	$71.20_{\pm 0.53}$	$70.19_{\pm 0.61}$	$54.45_{\pm 2.08}$	$52.73_{\pm 2.26}$	$62.41_{\pm 2.15}$	$46.84_{\pm 2.79}$	–	–	–	–
MVGRL	$71.25_{\pm 0.48}$	$70.37_{\pm 0.56}$	$55.25_{\pm 1.92}$	$53.67_{\pm 2.07}$	$59.55_{\pm 2.24}$	$51.93_{\pm 2.56}$	–	–	–	–
CCA-SSG	$68.21_{\pm 0.79}$	$67.03_{\pm 0.87}$	$56.41_{\pm 1.73}$	$53.86_{\pm 1.98}$	$63.39_{\pm 2.05}$	$50.72_{\pm 2.31}$	$36.98_{\pm 1.76}$	$28.94_{\pm 1.97}$	$69.42_{\pm 0.86}$	$57.65_{\pm 1.19}$
SpeGCL	$71.79_{\pm 0.42}$	$70.86_{\pm 0.49}$	$58.17_{\pm 1.35}$	$56.24_{\pm 1.58}$	$63.39_{\pm 1.72}$	$52.31_{\pm 2.06}$	$40.28_{\pm 1.28}$	$33.17_{\pm 1.54}$	$70.79_{\pm 0.49}$	$60.89_{\pm 0.73}$
DySubC	$72.42_{\pm 0.31}$	$71.56_{\pm 0.38}$	$59.24_{\pm 1.18}$	$57.68_{\pm 1.29}$	$64.79_{\pm 1.53}$	$54.13_{\pm 1.81}$	$40.74_{\pm 1.05}$	$33.62_{\pm 1.27}$	$71.35_{\pm 0.38}$	$61.58_{\pm 0.54}$
LDGC	$71.23_{\pm 0.44}$	$70.41_{\pm 0.53}$	$58.51_{\pm 1.26}$	$56.89_{\pm 1.45}$	$64.03_{\pm 1.61}$	$53.24_{\pm 1.88}$	$39.70_{\pm 1.15}$	$32.46_{\pm 1.38}$	$71.54_{\pm 0.52}$	$61.73_{\pm 0.69}$
CLDG++	$72.90_{\pm 0.27}$	$72.14_{\pm 0.32}$	$59.82_{\pm 1.08}$	$58.31_{\pm 1.24}$	$65.30_{\pm 1.45}$	$54.91_{\pm 1.71}$	$41.00_{\pm 0.95}$	$33.81_{\pm 1.16}$	$71.64_{\pm 0.34}$	$62.09_{\pm 0.51}$
UDGCL	$\mathbf{74.69}_{\pm 0.21}$	$\mathbf{73.89}_{\pm 0.26}$	$\mathbf{61.10}_{\pm 0.94}$	$\mathbf{59.45}_{\pm 1.12}$	$\mathbf{67.16}_{\pm 1.28}$	$\mathbf{55.73}_{\pm 1.46}$	$\mathbf{42.47}_{\pm 0.83}$	$\mathbf{35.19}_{\pm 0.96}$	$\mathbf{73.12}_{\pm 0.28}$	$63.54_{\pm 0.42}$

Models that do not give results because no code is given or there is not enough memory.

2025) samples temporally self-centred subgraphs and sequentially extracts structural and temporal features to balance diversity, dynamic modelling, and computational cost. Finally, AnomalyGFM (Qiao et al., 2025), a foundation model for graph anomaly detection, aligns data-agnostic normal and anomalous prototypes with neighbourhood residuals, supporting zero-shot detection and enabling few-shot prompt-based adaptation.

5.3. Experimental setup and evaluation metrics

To evaluate model performance, we adopt accuracy and weighted F1 score as the primary metrics for node classification. During linear evaluation, the encoder is kept fixed while a single-layer logistic regression classifier is trained. The optimal model is selected through early stopping on the validation set, and results are reported on the test set. For graph anomaly detection, we use AUC–ROC as the main evaluation metric. For the five datasets with injected anomalies, multiple views are generated following the same temporal ordering as in training. For Wikipedia and MOOC, which carry real-world ground-truth anomaly labels, the encoder is evaluated on their original chronological test splits without any label injection. In all anomaly detection settings, the encoder is frozen to compute three component-wise anomaly scores, which are then linearly combined to produce the final score. For both task types, we run five independent trials with different random seeds and report the mean and variance to reduce the impact of randomness. All baseline methods follow the original implementation and hyperparameter settings in their respective papers to ensure fair comparison.

A unified default configuration is used for most datasets (e.g., Bitcoin-Alpha, Bitcoin-OTC, DBLP). The encoder adopts a two-layer GraphConv architecture with a hidden dimension of 128 and an output dimension of 64. Neighbour sampling uses a fanout of $[20, 20]$. The Adam optimiser is used with a learning rate of $2 \times e^{-3}$, weight decay of $5 \times e^{-4}$, a training batch size of 256, and 400 total training epochs. The adaptive time-window range is configured as $w_{\min} = 3$ and $w_{\max} = 10$. The InfoNCE temperature is $\tau = 0.1$, and the attention-temperature parameter is $\gamma = 5.0$. The batch sizes for inference are 4096 for node classification and 1024 for anomaly detection. For the larger Reddit dataset, to balance expressive power with memory constraints, we increase the hidden and output dimensions to 256 and 128, respectively. The fanout is reduced to $[15, 10]$, the learning rate lowered to e^{-3}, and the batch size set to 128, with all other hyperparameters unchanged. For the more hierarchical Yelp dataset, the encoder depth is increased to three layers with a fanout of $[20, 20, 10]$, and the number of training epochs is extended to 500. All other settings follow the default configuration. Hyperparameters central to our methodology are not discussed here, and their sensitivity and selection rationale will be presented in the Hyperparameter Analysis section. All experiments are conducted on a single NVIDIA RTX 3090Ti GPU (24 GB) using PyTorch and DGL.

6. Results and discussion

6.1. Overall performance

6.1.1. Node classification

We present the node classification results in Table 2, evaluating all methods using Accuracy and Weighted F1 scores across five datasets. Anomaly detection results are reported in Table 3, covering five injected-anomaly datasets as well as two real-world anomaly benchmarks, Wikipedia and MOOC.

Among supervised methods, GraphSAGE delivers competitive performance, particularly on DBLP and Reddit, thanks to its neighbour sampling mechanism, which facilitates scalable training and effective aggregation of local structural information. GAT, in contrast, shows relatively weaker performance, likely due to overfitting from its large number of trainable parameters, especially when label supervision is limited. All supervised methods struggle on TAX51, where pure structural aggregation without temporal modelling fails to capture the complex temporal patterns in tax avoidance networks.

Dynamic graph methods without contrastive learning yield mixed results across datasets. TGAT achieves moderate but inconsistent performance, suggesting that temporal encoding alone cannot effectively address temporal uncertainty when nodes exhibit varying degrees of structural stability across time. DySAT performs poorly on most datasets and fails to run on Reddit due to memory constraints, while MNCI shows improvement through temporal consistency constraints but still struggles when nodes exhibit highly heterogeneous temporal contrastive reliability.

Static contrastive methods show moderate success, with SpeGCL performing the best in this category through its spectral augmentation strategy that preserves essential graph properties. However, these methods treat temporal graphs as static snapshot aggregates, losing fine-grained temporal dependency information. This limitation is especially evident on Bitcoinotc and TAX51, where static methods show suboptimal accuracy due to high temporal uncertainty.

Among temporal contrastive approaches, CLDG++ achieves state-of-the-art performance in most cases by explicitly modelling time-shift invariance via graph diffusion and multi-scale local–global contrast. DySubC and LDGC also show improvements by incorporating temporal subgraph contrast and temporal consistency constraints, respectively. However, these methods adopt uniform temporal weighting schemes

Table 3
AUC–ROC scores (%) of the **anomaly detection** task.

Method	Injected-anomaly datasets					Real-anomaly datasets	
	DBLP	Bitcoinotc	BITotc	TAX51	Reddit	Wikipedia	MOOC
AEGIS	$59.18_{\pm 3.17}$	$50.97_{\pm 6.15}$	$53.17_{\pm 0.57}$	$54.52_{\pm 1.67}$	–	$61.47_{\pm 4.53}$	$55.26_{\pm 2.63}$
CoLA	$71.06_{\pm 0.24}$	$79.44_{\pm 1.46}$	$81.80_{\pm 0.31}$	$77.29_{\pm 0.12}$	–	$74.28_{\pm 0.45}$	$65.94_{\pm 0.78}$
ANEMONE	$70.69_{\pm 1.00}$	$79.28_{\pm 0.56}$	$80.09_{\pm 0.93}$	$76.61_{\pm 0.23}$	–	$75.13_{\pm 1.08}$	$67.48_{\pm 1.15}$
GRADATE	$69.53_{\pm 0.10}$	$74.14_{\pm 0.32}$	$73.12_{\pm 0.42}$	–	–	$72.47_{\pm 0.35}$	$64.32_{\pm 0.58}$
NetWalk	$54.42_{\pm 0.33}$	$56.13_{\pm 0.71}$	$58.46_{\pm 0.44}$	–	–	$57.81_{\pm 0.62}$	$60.18_{\pm 0.48}$
TADDY	$50.93_{\pm 0.59}$	$50.58_{\pm 0.33}$	$51.48_{\pm 0.38}$	–	–	$82.74_{\pm 1.32}$	$69.14_{\pm 0.87}$
SAD	$78.24_{\pm 0.31}$	$75.67_{\pm 0.28}$	$71.29_{\pm 0.46}$	$61.70_{\pm 0.79}$	$72.19_{\pm 0.60}$	$87.31_{\pm 0.38}$	$72.53_{\pm 1.18}$
SLADE	$82.15_{\pm 0.68}$	$77.18_{\pm 0.27}$	$75.80_{\pm 0.19}$	$71.24_{\pm 0.58}$	$75.08_{\pm 0.50}$	$84.19_{\pm 0.71}$	$68.91_{\pm 0.94}$
CLDG++	$86.41_{\pm 0.32}$	$81.97_{\pm 1.08}$	$82.92_{\pm 0.54}$	$81.02_{\pm 0.26}$	$72.77_{\pm 0.08}$	$85.48_{\pm 0.67}$	$70.23_{\pm 0.45}$
GeneralDyG	$85.23_{\pm 0.74}$	$88.47_{\pm 0.65}$	$86.32_{\pm 0.58}$	$82.56_{\pm 0.48}$	$81.38_{\pm 0.72}$	$87.06_{\pm 0.83}$	$71.84_{\pm 1.12}$
AnomalyGFM	$84.68_{\pm 0.89}$	$83.25_{\pm 0.76}$	$81.74_{\pm 0.82}$	$80.15_{\pm 0.69}$	$79.42_{\pm 0.85}$	$85.34_{\pm 1.04}$	$71.67_{\pm 0.88}$
UDGCL	$\mathbf{89.82}_{\pm 0.28}$	$\mathbf{92.04}_{\pm 0.35}$	$\mathbf{87.86}_{\pm 0.42}$	$\mathbf{86.80}_{\pm 0.31}$	$\mathbf{86.25}_{\pm 0.47}$	$\mathbf{90.47}_{\pm 0.33}$	$\mathbf{75.93}_{\pm 0.68}$

Models that do not give results because no code is given or there is not enough memory.

without adapting to node-level temporal contrastive uncertainty. They implicitly assume equal reliability across all temporal contrasts. Fast-evolving nodes with high TCU may inject unreliable contrastive signals, while their temporal pairs are incorrectly enforced to be similar despite representing fundamentally different structural roles at different timestamps.

UDGCL achieves the best performance in almost all cases across both metrics and datasets, consistently outperforming the strongest baseline, CLDG++, with noticeable margins. The improvements are especially significant on datasets exhibiting higher TCU heterogeneity, where TCU-guided adaptive sampling prevents noisy alignments for high-uncertainty nodes while maintaining long-term consistency for low-uncertainty nodes. Additionally, uncertainty-weighted negative sampling effectively suppresses unreliable contrastive pairs, which could otherwise mislead the learning process. On Reddit, UDGCL achieves the highest accuracy but shows comparable F1 performance to GraphSAGE, likely due to Reddit's severe class imbalance, where supervised methods can directly optimise class-specific boundaries using labelled data.

6.1.2. Anomaly detection

Table 3 presents AUC–ROC scores for anomaly detection, where UDGCL demonstrates significant improvements over all baselines across every dataset.

Classical methods such as AEGIS, NetWalk, and TADDY perform poorly, achieving near-random results on most datasets. These methods rely on reconstruction or random-walk strategies, which lack explicit temporal modelling and fail to distinguish between legitimate temporal structural changes and anomalous behaviour. As a result, they do not capture deviations from expected temporal patterns under varying uncertainty levels.

Static contrastive methods show moderate improvements. CoLA and ANEMONE perform better by leveraging node-neighbourhood substructure contrast and multi-scale patch-context contrast, respectively, while GRADATE employs multi-view subgraph contrast. However, treating temporal graphs as static snapshots misses crucial temporal dependency signals and the variations in cross-time reliability essential for anomaly detection.

SAD utilises pseudo-labelling and memory banks in a semi-supervised setting, showing reasonable performance on some datasets, but its performance drops notably on others. SLADE, with its self-supervised drift minimisation in a streaming paradigm, demonstrates improved stability by explicitly modelling representation drift. However, SLADE's uniform drift penalty fails to account for node-specific temporal uncertainty and varying temporal contrastive reliability across nodes.

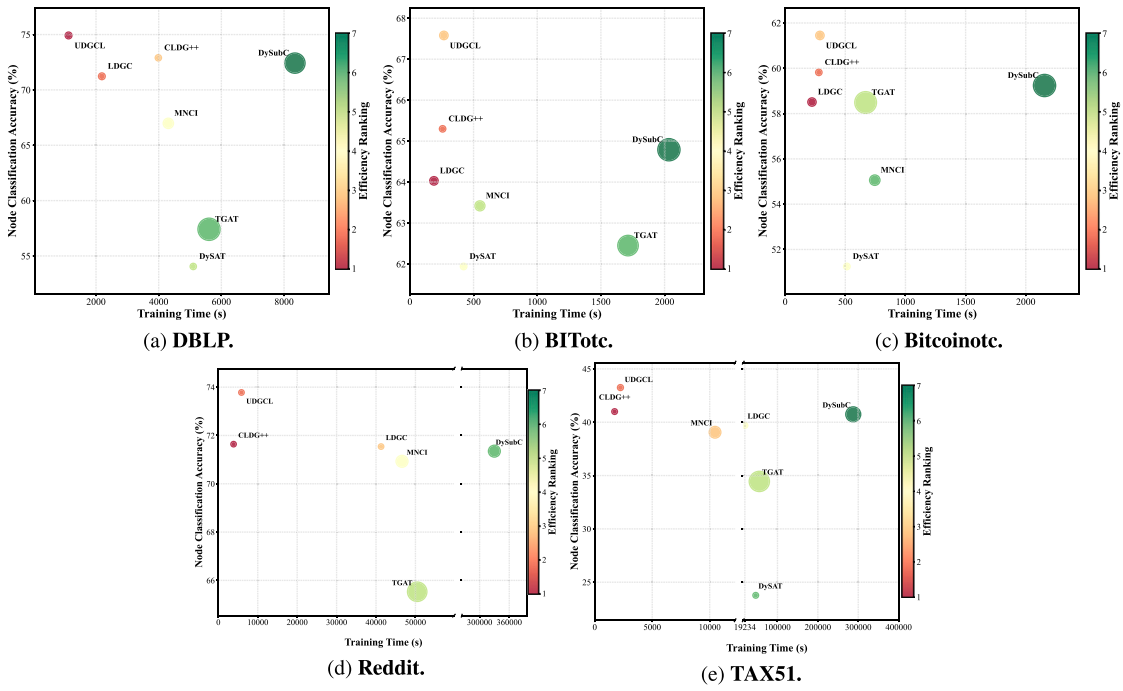
Recent strong baselines show marked improvements. CLDG++ achieves good performance with time-shift invariance and diffusion-based multi-scale contrast, particularly on DBLP and Bitcoinotc. GeneralDyG provides the strongest baseline performance on most datasets by extracting structural and temporal features from ego-subgraphs. AnomalyGFM, as a foundational model, shows competitive results through prototype alignment. However, the uniform time-shift invariance assumption in CLDG++ fails when nodes exhibit asynchronous temporal dynamics with varying reliability. GeneralDyG's fixed sequential feature extraction lacks adaptability to temporal contrastive uncertainty, and AnomalyGFM's data-agnostic prototypes cannot capture dataset-specific temporal dynamics and node-level uncertainty heterogeneity.

UDGCL consistently outperforms all baselines across five datasets, with substantial improvements over the strongest baseline, GeneralDyG. These improvements are especially pronounced on datasets exhibiting high structural volatility and temporal uncertainty heterogeneity. The temporal contrastive uncertainty estimation effectively captures node-specific temporal reliability by combining structural entropy and the temporal variation coefficient. The multi-scale temporal kernels learned for different TCU groups enable adaptive modelling of temporal similarity across diverse uncertainty patterns. Additionally, the uncertainty-weighted negative reweighting mechanism prevents the misclassification of high-uncertainty normal nodes as hard negatives, fostering a more discriminative and reliability-aware embedding space where anomalies are distinguished not by structural volatility alone, but by deviation from expected temporal contrastive patterns.

To further examine whether the observed gains extend beyond injected settings, we also evaluate UDGCL on Wikipedia and MOOC, where anomaly labels are derived from real-world platform records. UDGCL still achieves the best results on both datasets, showing that its advantage is not confined to artificially constructed anomalies. Compared with injected benchmarks, these two datasets provide a more realistic test of temporal irregularity. The results suggest that methods based on uniform temporal assumptions are less reliable in this setting, whereas UDGCL remains robust. This trend is more evident on Wikipedia, where anomalous behaviour is often accompanied by clearer structural changes. SAD also performs relatively well, which is consistent with its semi-supervised setting, but UDGCL still shows stronger overall robustness. On MOOC, the gain is more moderate, indicating that the anomalies are not driven purely by abrupt structural variation. Overall, these results further support the effectiveness of UDGCL on real-world anomaly detection benchmarks.

6.1.3. Efficiency comparison

We compare the training efficiency of UDGCL against dynamic-graph baselines using bubble plots in Fig. 3, where bubble size represents the parameter count and colour indicates the Acc/Time efficiency



(a) **DBLP.** (b) **BITotc.** (c) **Bitcoinotc.**

(d) **Reddit.** (e) **TAX51.**

Fig. 3. Training efficiency comparison of UDGCL and the dynamic graph methods.

Table 4
Ablation Study on **node classification** task.

Method	DBLP		Bitcoinotc		BITotc		TAX51		Reddit	
	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)
UDGCL	**74.69**$_{\pm 0.21}$	**73.89**$_{\pm 0.26}$	**61.10**$_{\pm 0.94}$	**59.45**$_{\pm 1.12}$	**67.16**$_{\pm 1.28}$	**55.73**$_{\pm 1.46}$	**42.47**$_{\pm 0.83}$	**35.19**$_{\pm 0.96}$	**73.12**$_{\pm 0.28}$	**63.54**$_{\pm 0.42}$
w/o AS	73.82$_{\pm 0.29}$	73.01$_{\pm 0.33}$	58.47$_{\pm 1.16}$	56.79$_{\pm 1.31}$	65.92$_{\pm 1.41}$	54.23$_{\pm 1.57}$	38.74$_{\pm 0.97}$	31.52$_{\pm 1.09}$	71.63$_{\pm 0.35}$	62.18$_{\pm 0.49}$
w/o UW	72.86$_{\pm 0.36}$	71.98$_{\pm 0.41}$	56.73$_{\pm 1.34}$	54.96$_{\pm 1.49}$	64.51$_{\pm 1.53}$	52.71$_{\pm 1.71}$	39.18$_{\pm 1.02}$	32.07$_{\pm 1.14}$	70.78$_{\pm 0.41}$	61.32$_{\pm 0.56}$
w/o GP	74.19$_{\pm 0.25}$	73.34$_{\pm 0.28}$	60.18$_{\pm 1.01}$	58.57$_{\pm 1.16}$	66.47$_{\pm 1.33}$	54.91$_{\pm 1.48}$	41.59$_{\pm 0.87}$	34.31$_{\pm 0.99}$	72.23$_{\pm 0.32}$	62.61$_{\pm 0.46}$
w/o MK	74.27$_{\pm 0.24}$	73.42$_{\pm 0.29}$	60.34$_{\pm 0.98}$	58.71$_{\pm 1.13}$	66.69$_{\pm 1.36}$	55.14$_{\pm 1.51}$	41.78$_{\pm 0.85}$	34.52$_{\pm 0.97}$	72.08$_{\pm 0.33}$	62.43$_{\pm 0.48}$

rank. All methods are trained under the same protocol and hardware setup.

UDGCL consistently achieves superior speed–accuracy trade-offs, ranking first or second by the Acc/Time efficiency metric while delivering the highest accuracy. It trains significantly faster than attention-heavy architectures (e.g., TGAT, DySAT) and memory-intensive subgraph methods (e.g., MNCI, DySubC), while maintaining competitive speed relative to lightweight contrastive baselines (e.g., CLDG++, LDGC). In large-scale settings (e.g., TAX51, Reddit), the adaptive temporal sampling proves particularly beneficial, as it dynamically adjusts window sizes based on TCU with $A \ll T$ for high-TCU nodes and avoids the quadratic complexity explosion faced by fixed-window methods as T increases. In terms of parameters, UDGCL remains lightweight, with modest bubble sizes across datasets. This is because the group-level components scale with small constants rather than the node count.

6.2. Ablation studies

To systematically assess the contribution of each component in UDGCL, we compare the full model with four degraded variants: (1) w/o AS, replacing adaptive temporal sampling with fixed uniform windows; (2) w/o UW, removing uncertainty weighting and treating all negative samples equally; (3) w/o GP, discarding group-level prototype calibration; (4) w/o MK, replacing multi-scale temporal kernels with a single fixed decay function. Experimental results are reported in Table 4 for node classification and Fig. 4 for anomaly detection.

UDGCL consistently outperforms all variants across both tasks, with anomaly detection showing roughly double the sensitivity to component removal compared to node classification. This elevated sensitivity is expected given the sparsity of anomalies—false negatives in contrastive learning distort decision boundaries far more severely in anomaly detection scenarios.

Uncertainty weighting (UW) emerges as the most critical component. Removing UW leads to substantial degradation on datasets with heterogeneous TCU distributions such as TAX51 and Bitcoinotc, confirming that high TCU amplifies the impact of false negatives: nodes with different temporal contrastive reliability levels are incorrectly treated as negatives despite occupying distinct structural roles at different timestamps. Even in Bitcoinotc, where most nodes have similar TCU values, the small minority of high-TCU nodes forms unreliable negative samples that disproportionately corrupt the learned representations when not properly downweighted.

The adaptive sampling (AS) mechanism provides the second-largest contribution, with its effectiveness directly tied to TCU heterogeneity. On TAX51, where nodes are well distributed across TCU regimes, uniform sampling fails to capture heterogeneous temporal reliability, resulting in significant performance degradation. On Reddit, where TCU is relatively uniform, the impact is less pronounced. However, on Bitcoinotc, despite its concentrated TCU distribution, AS remains essential because the small high-TCU minority contains critical unstable patterns that fixed windows misalign. The importance of AS is further amplified in anomaly detection — particularly on TAX51, where its

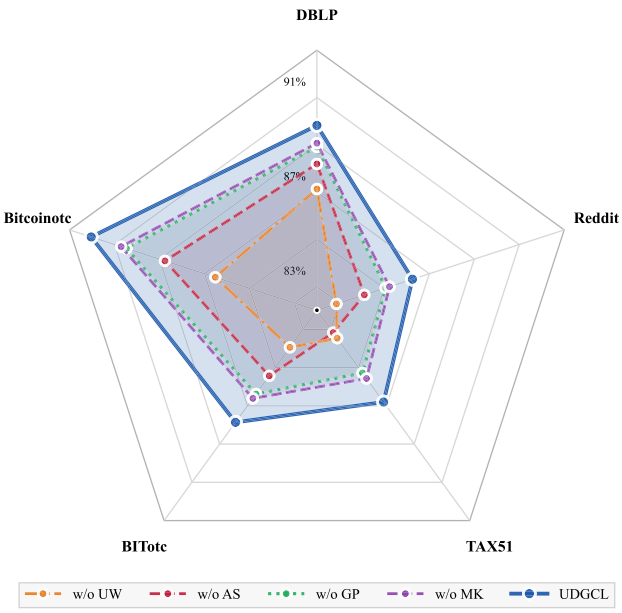


Fig. 4. Ablation Study on Anomaly Detection.

contribution rivals that of UW — reflecting the tendency of anomalies to exhibit high TCU and require adaptive temporal alignment.

Group-level prototype calibration (GP) provides moderate but consistent improvements. On DBLP, where TCU variance is high, GP reduces boundary noise across groups via cross-group attention. On Reddit, despite a homogeneous TCU distribution, the large node count offers sufficient statistics for stable aggregation. The effect of GP is more pronounced in anomaly detection, as anomalies often lie near TCU group boundaries with ambiguous assignments.

The multi-scale kernel (MK) component yields the smallest overall contribution, with its impact inversely correlated with TCU variance. On DBLP, which has a wide TCU spread, removing MK causes only minor degradation, as uncertainty heterogeneity itself captures multiple temporal scales. Conversely, on Reddit and Bitcoinotc, where TCU values are highly concentrated, MK becomes the primary mechanism for modelling within-group temporal complexity. Its contribution is consistently amplified in anomaly detection, suggesting that anomalies often exhibit cross-scale inconsistencies requiring multi-scale modelling.

A cross-dataset analysis reveals that component sensitivities are strongly aligned with TCU distribution characteristics. Although TAX51 and Bitcoinotc represent opposite extremes of TCU heterogeneity, both exhibit substantial degradation when AS or UW is removed, albeit for different reasons. On TAX51, adaptive sampling is needed to accommodate diverse temporal reliability levels, whereas on Bitcoinotc, a small high-TCU minority generates unreliable temporal contrast signals that fixed windows cannot properly handle. These findings indicate that AS and UW address complementary failure modes: AS provides proper temporal alignment across uncertainty regimes, while UW prevents unreliable negatives from corrupting the learned representations. The hierarchical design of UDGCL — where TCU first guides sampling and weighting, followed by GP and MK refining group-level and scale-specific patterns — proves effective across datasets with markedly different temporal uncertainty profiles.

6.3. Performance stratification by TCU

To directly validate the hypothesis that temporal contrastive reliability varies heterogeneously across nodes, we conduct a fine-grained stratified performance analysis by partitioning nodes into five disjoint TCU groups based on quintile boundaries: Very Low, Low, Medium, High, and Very High. For each group, we freeze the trained encoder and evaluate node classification accuracy exclusively on test nodes within that TCU range. This stratification isolates the model's capability to handle nodes with fundamentally different temporal contrastive reliability characteristics, from stable nodes for which naive temporal alignment suffices, to high-uncertainty nodes for which blind consistency enforcement introduces noise.

Fig. 5 presents per-group performance curves together with sample-distribution histograms overlaid as background bars. The results reveal several key patterns that directly support UDGCL's architectural design. Notably, UDGCL does not uniformly dominate across all TCU groups, a behaviour that paradoxically reinforces our claims. In the Very Low TCU regime—where nodes maintain stable structural patterns over time, UDGCL performs comparably to, or slightly below, simpler baselines such as MNCI and CLDG++. This finding is theoretically expected: for temporally consistent nodes, aggressive temporal alignment is inherently valid, and lightweight methods without uncertainty modelling already perform well. UDGCL's conservative weighting strategy intentionally allocates less emphasis to such nodes, accepting minor losses on easy cases to avoid catastrophic failures on more challenging ones.

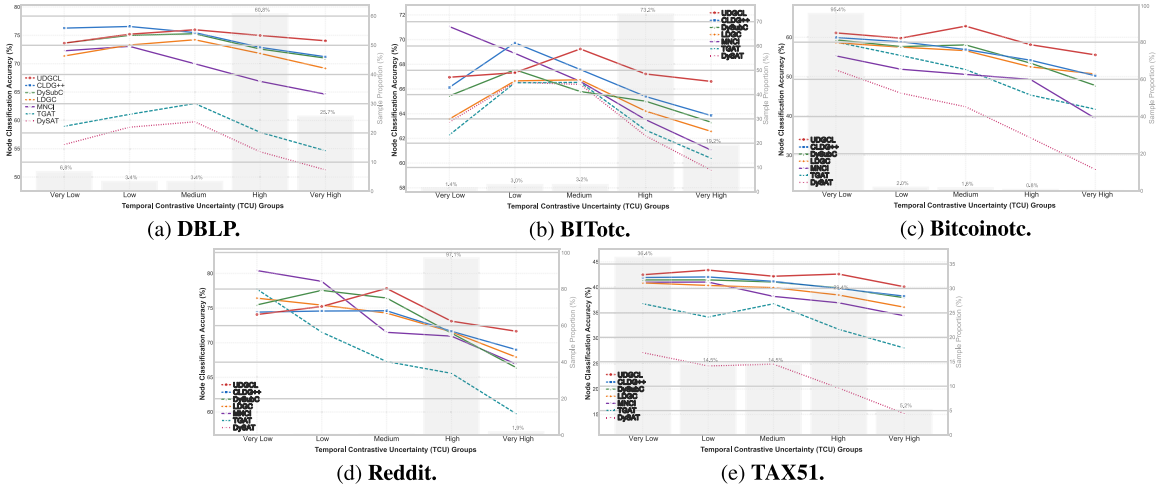
Clear divergence emerges in the Medium to High TCU regimes, where adaptive mechanisms directly address temporal contrastive uncertainty. As TCU increases, performance gaps widen substantially, with UDGCL maintaining stable accuracy while baselines show systematic degradation. This stratification confirms that UDGCL's improvements concentrate precisely where the problem is hardest. Traditional methods collapse in high-uncertainty regimes, often suffering performance drops more than twice those observed for UDGCL, because their implicit uniform-reliability assumption forces alignment of representations for nodes undergoing genuine temporal changes. As a result, these methods tend to suppress true temporal variation signals and produce overly homogenised embeddings that fail on high-uncertainty subpopulations. In contrast, UDGCL's robustness stems from its uncertainty-weighted contrastive loss and adaptive sampling, which jointly prevent noisy temporal pairs from contaminating the embedding space.

The relative importance of UDGCL's components varies with the dataset-specific TCU distribution. On highly concentrated datasets, where Very Low TCU nodes constitute more than 95% of the population, UDGCL's overall improvement is driven primarily by preserving performance on the sparse high-TCU tail, rather than enhancing the majority group. While performance in the Very Low group remains nearly identical to that of baselines, dramatic preservation occurs in the Very High group, where competing methods experience catastrophic failure. On high-uncertainty dominant datasets — where most nodes exhibit high TCU — adaptive sampling becomes critical for the majority population. Group-level prototype calibration further stabilises representations by aggregating signals across similar-uncertainty nodes, producing consistent gains across the Medium, High, and Very High groups. On datasets with balanced TCU distributions, UDGCL demonstrates broad adaptability, achieving improvements across most groups and suggesting that the learned multi-scale temporal kernels capture regime-specific dependencies when sufficient samples are available.

Small-sample TCU groups naturally exhibit higher variance due to limited statistical support. Importantly, UDGCL maintains lower standard deviation than baselines in the high-TCU groups, demonstrating that uncertainty quantification not only improves mean accuracy but also enhances prediction reliability. This increased stability is especially valuable because high-TCU nodes often correspond to structurally anomalous or temporally unstable entities for which robustness is crucial.

6.4. Hyperparameter sensitivity

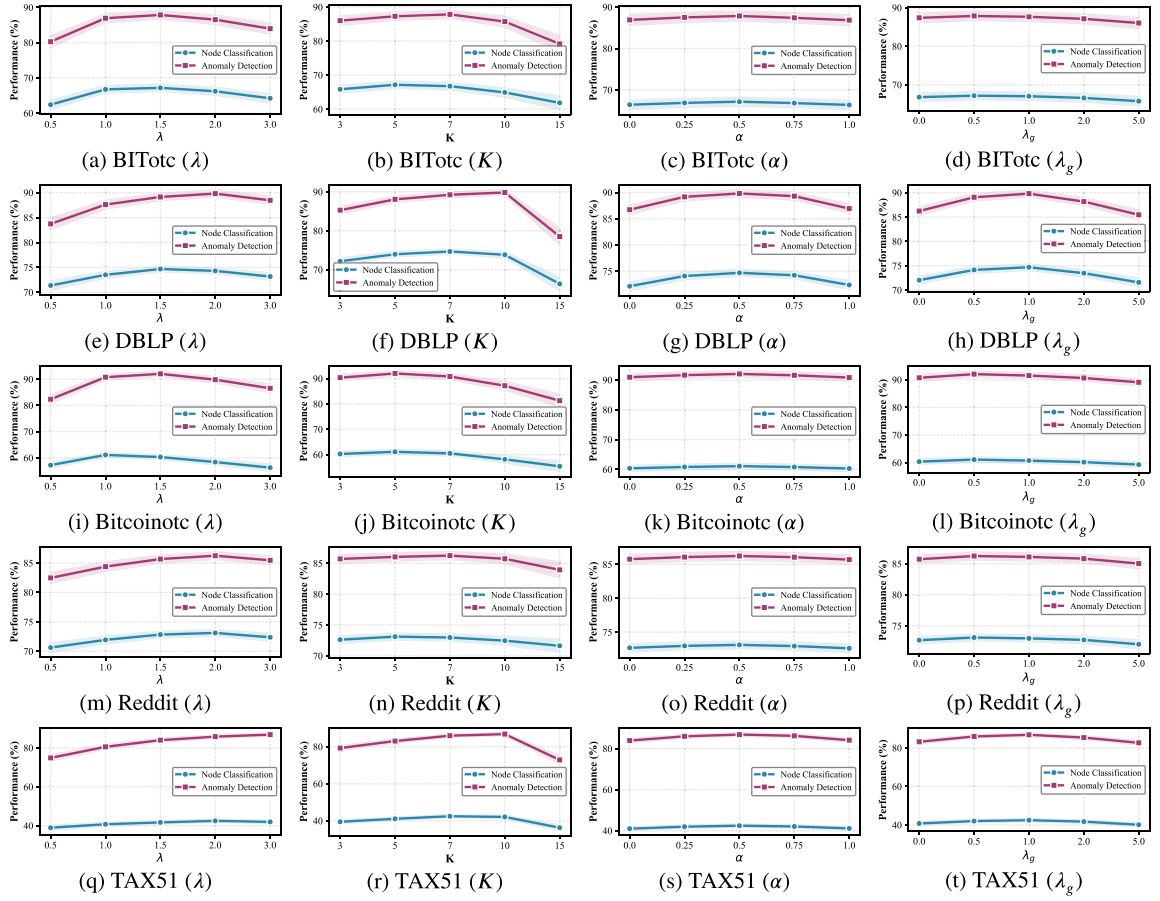
To evaluate the sensitivity of UDGCL to hyperparameter choices, we systematically examine the effects of four key parameters: λ: uncer-



(a) **DBLP.** (b) **BITotc.** (c) **Bitcoinotc.**

(d) **Reddit.** (e) **TAX51.**

Fig. 5. Node classification accuracy across TCU groups on five datasets (using only dynamic graph methods).



(a) BITotc (λ) (b) BITotc (K) (c) BITotc (α) (d) BITotc (λ_g)

(e) DBLP (λ) (f) DBLP (K) (g) DBLP (α) (h) DBLP (λ_g)

(i) Bitcoinotc (λ) (j) Bitcoinotc (K) (k) Bitcoinotc (α) (l) Bitcoinotc (λ_g)

(m) Reddit (λ) (n) Reddit (K) (o) Reddit (α) (p) Reddit (λ_g)

(q) TAX51 (λ) (r) TAX51 (K) (s) TAX51 (α) (t) TAX51 (λ_g)

Fig. 6. Hyperparameter sensitivity analysis on five datasets. From left to right, the columns correspond to λ, K, α, and λ_g.

tainty penalty strength in the contrastive weighting module, K: number of TCU groups, α: weight of structural entropy in TCU computation, and λ_g: weight of the group-level prototype loss. We vary these parameters over the following ranges: $\lambda \in \{0.5, 1.0, 1.5, 2.0, 3.0\}$, $K \in \{3, 5, 7, 10, 15\}$, $\alpha \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$, $\lambda_g \in \{0.0, 0.5, 1.0, 2.0, 5.0\}$. During the tuning of each parameter, all others are kept at their default settings: $\lambda = 1.5$, $K = 7$, $\alpha = 0.5$, and $\lambda_g = 1.0$. The resulting performance curves, reported in terms of Accuracy (Acc) and AUC, are shown in Fig. 6. The joint sensitivity of the anomaly score weights $\lambda_1, \lambda_2, \lambda_3$ involves a three-way interaction that cannot be captured by univariate curves, and it is therefore analysed separately via a full simplex grid search in Section 6.4.5.

6.4.1. *Hyperparameter λ of uncertainty penalty*

The uncertainty penalty coefficient λ exhibits a clear sensitivity pattern across our experimental datasets. Performance peaks at moderate λ values, where the model effectively downweights negative samples from high-TCU nodes without discarding all contrastive signals. Setting

λ too low fails to suppress noisy gradients from structurally unstable nodes, while excessive values indiscriminately penalise even reliable negative pairs, thereby removing valuable contrastive information from nodes experiencing legitimate temporal evolution. Notably, the trust networks (Bitcoinotc and BITotc) demonstrate heightened sensitivity to λ compared to DBLP and Reddit. We attribute this to the bimodal TCU distributions in trust datasets, where a substantial fraction of nodes exhibit either extremely high or extremely low uncertainty—making the selective weighting mechanism critical for separating signal from noise in temporal contrasts.

6.4.2. Hyperparameter K of TCU group number

From the results, we observe that as K increases, the performance of UDGCL on all datasets initially improves and then declines sharply. This pattern reflects the role of K in controlling the granularity of TCU-based grouping: when K is too small, nodes with heterogeneous temporal contrastive uncertainty are forced into overly coarse partitions, leading to incompatible temporal kernels being averaged together, and when K is too large, excessive fragmentation produces small groups that are vulnerable to noise and overfitting due to insufficient samples per group. The sharp performance degradation at large K values across all datasets further confirms that over-partitioning is generally more harmful than under-partitioning. Moreover, the optimal choice of K varies across datasets. Larger and more heterogeneous networks such as DBLP and TAX51 benefit from finer-grained partitioning to accommodate their diverse TCU distributions. In contrast, smaller datasets like Bitcoinotc and BITotc achieve the best results with moderate group numbers that balance the need to capture uncertainty heterogeneity with the need to maintain statistical stability.

6.4.3. Hyperparameter α of structural entropy weight

Parameter α governs the relative contribution of structural entropy versus temporal variation coefficient in quantifying TCU. The consistent peak at intermediate α values across datasets validates our design choice to fuse both components. Pure reliance on variation coefficient captures short-term volatility but misses persistent distributional drift. Conversely, entropy-only measures detect long-term structural shifts but remain insensitive to sudden anomalies in node degrees. The synergistic combination allows UDGCL to identify both gradual evolution and abrupt changes in local topology. Importantly, the relatively flat performance plateau around optimal α indicates robustness to this hyperparameter—precise tuning is unnecessary as long as both entropy and variation meaningfully inform the uncertainty estimate.

6.4.4. Hyperparameter λ_g of group-level loss weight

Group-level regularisation strength λ_g controls the degree to which nodes leverage their TCU group prototypes for representation refinement. At low λ_g, nodes rely solely on noisy individual TCU estimates derived from degree sequences, a particularly problematic scenario for sparsely-connected nodes or those near group boundaries. Introducing moderate λ_g activates soft cross-group attention, enabling nodes to calibrate their embeddings against learned prototypes rather than raw statistics, which effectively smooths estimation errors. However, excessive λ_g imposes over-regularisation that collapses representations toward group centroids, forcing nodes with similar TCU but distinct local contexts into identical embeddings. This discriminative loss is most severe in datasets with high intra-group variance (e.g., TAX51), where aggressive prototype alignment undermines the model's capacity to capture the nuanced uncertainty patterns that TCU aims to characterise. The optimal λ_g thus strikes a balance: sufficient regularisation to denoise TCU estimates without sacrificing the representational diversity needed for fine-grained temporal dynamics modelling.

6.4.5. Hyperparameter $\lambda_1, \lambda_2, \lambda_3$ of anomaly score weights

The anomaly score $S_i = \lambda_1 \hat{C}_i + \lambda_2 \hat{R}_i + \lambda_3 \hat{U}_i$ is governed by three non-negative weights subject to $\lambda_1 + \lambda_2 + \lambda_3 = 1$. To characterise how performance varies across the full weight space, we enumerate all weight triplets on the probability simplex with a uniform step size of 0.125, yielding 45 distinct configurations. AUC–ROC is evaluated on all five datasets for each configuration, and a continuous sensitivity landscape is obtained via linear triangular interpolation over the Cartesian projection of the simplex. The results are visualised as ternary contour heatmaps in Fig. 7.

The landscapes reveal two consistent patterns across datasets. First, configurations concentrated near any single vertex — corresponding to exclusive reliance on one metric — yield the lowest performance, confirming that $\hat{C}_i$, $\hat{R}_i$, and $\hat{U}_i$ capture complementary anomaly signals that are not individually sufficient. Second, the high-performance region forms a broad interior plateau rather than a sharply localised peak, indicating that the model is robust to moderate weight variations. The optimal configuration shifts modestly across datasets, reflecting differences in dominant anomaly types: datasets where anomalies primarily manifest as representational drift benefit from higher λ_2, whereas those with stronger local structural disruptions favour elevated λ_3.

6.5. TCU robustness to degree-preserving neighbourhood substitution

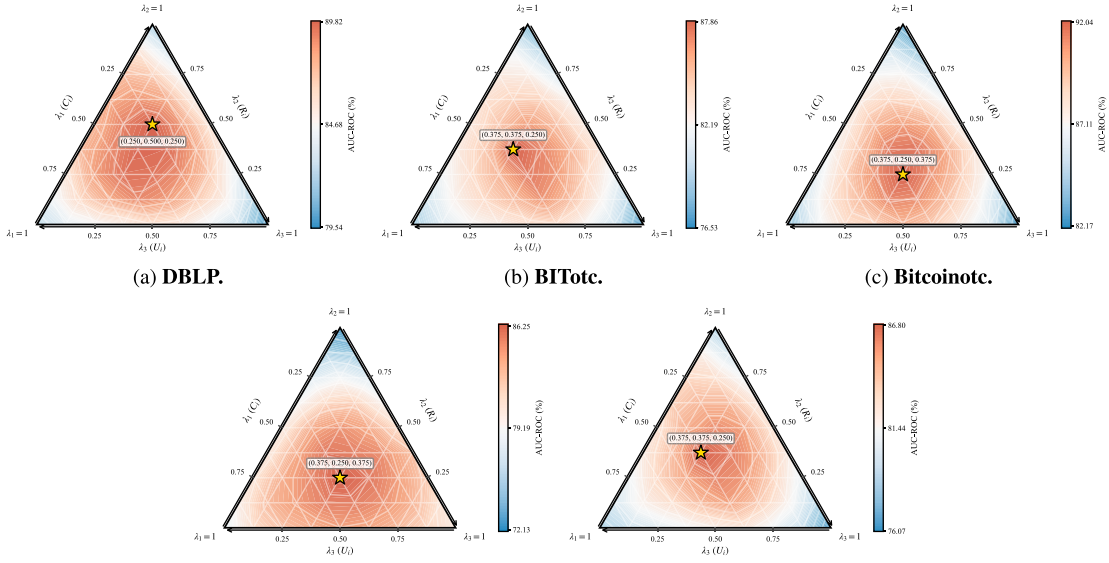
TCU is computed solely from degree sequences, so neighbourhood substitutions that preserve node degrees leave TCU values unchanged even when local semantic context shifts. This is intentional: degree-based TCU provides stable, pre-computed reliability estimates that are insensitive to snapshot-level neighbourhood noise. The remaining question is whether the model stays robust when such semantic shifts occur without a corresponding TCU signal.

Table 5 shows that UDGCL suffers accuracy drops of only 0.52–0.66 percentage points under perturbation, whereas removing group-level prototype calibration (w/o GP) roughly doubles the degradation (1.08–1.47 points). This confirms that prototype calibration compensates for residual semantic drift that degree dynamics cannot capture: when neighbourhood context shifts without changing TCU, the learned group prototypes anchor perturbed representations back toward the group's expected pattern.

7. Conclusion

In this paper, we proposed UDGCL, an uncertainty-aware dynamic graph contrastive learning framework for node classification and anomaly detection on temporal networks. Central to our approach is the notion of TCU, which quantifies the reliability of cross-time contrasts based on degree dynamics, revealing that temporal pairs should not be treated with uniform trust. Building on this insight, UDGCL introduces an adaptive temporal sampling strategy that allocates time windows according to node-level TCU, and employs uncertainty-weighted contrastive learning with multi-scale temporal kernels to suppress unreliable negative samples, and incorporates group-level prototype calibration to mitigate TCU estimation noise through soft cross-group attention. Extensive experiments on five public dynamic graph datasets for node classification, and seven datasets (including two with real-world anomaly labels) for anomaly detection, demonstrate that UDGCL consistently outperforms strong baselines in terms of Accuracy, F1-score, and AUC–ROC.

For future work, we aim to extend TCU estimation beyond degree dynamics by incorporating higher-order structural descriptors such as clustering coefficients and motif patterns to capture richer evolutionary behaviour. We are also interested in developing online or incremental variants of UDGCL for streaming graphs, where TCU estimates and prototypes are updated in real time as new snapshots arrive, enabling efficient and scalable deployment in production systems.



(a) **DBLP.** (b) **BITotc.** (c) **Bitcoinotc.**

(d) **Reddit.** (e) **TAX51.**

Fig. 7. Sensitivity analysis of anomaly score weights $(\lambda_1, \lambda_2, \lambda_3)$ on five datasets. The star marker denotes the dataset-specific optimal configuration.

Table 5
Node classification performance under degree-preserving neighbourhood substitution (5% of nodes in one snapshot perturbed; unperturbed results in parentheses).

Method	Bitcoinotc		DBLP		BITotc	
	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)
UDGCL	$60.44_{\pm 1.07}$	$58.68_{\pm 1.25}$	$74.17_{\pm 0.30}$	$73.31_{\pm 0.34}$	$66.58_{\pm 1.33}$	$55.11_{\pm 1.54}$
(unperturbed)	(61.10)	(59.45)	(74.69)	(73.89)	(67.16)	(55.73)
w/o GP	$58.71_{\pm 1.18}$	$56.93_{\pm 1.34}$	$73.11_{\pm 0.36}$	$72.20_{\pm 0.41}$	$65.03_{\pm 1.45}$	$53.45_{\pm 1.65}$
(unperturbed)	(60.18)	(58.57)	(74.19)	(73.34)	(66.47)	(54.91)

CRediT authorship contribution statement

Long Xu: Visualization, Software, Methodology, Conceptualization. **Zhiqiang Pan:** Writing – review & editing, Writing – original draft, Investigation, Data curation. **Honghui Chen:** Validation, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data used in this study are publicly available.

References

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U.R., et al., 2021. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. Inf. Fusion 76, 243–297.

Bai, W., Qiu, L., Zhao, W., 2025. Dynamic heterogeneous graph representation based on adaptive negative sample mining via high-fidelity instances and context-aware uncertainty learning. Expert Syst. Appl. 278, 127291.

Bao, Y., Huang, J., Shen, Q., Cao, Y., Ding, W., Shi, Z., Shi, Q., 2023. Spatial–temporal complex graph convolution network for traffic flow prediction. Eng. Appl. Artif. Intell. 121, 106044.

Belghazi, M.I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., Hjelm, D., 2018. Mutual information neural estimation. In: Proc. Int. Conf. Mach. Learn., vol. 80, pp. 531–540.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.A., 2019. Mixmatch: A holistic approach to semi-supervised learning. Adv. Neural Inf. Process. Syst. 32, 5050–5060.

Chen, K.-J., Liu, L., Jiang, L., Chen, J., 2023. Self-supervised dynamic graph representation learning via temporal subgraph contrast. Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. 18 (1), 1–20.

Cong, W., Zhang, S., Kang, J., Yuan, B., Wu, H., Zhou, X., Tong, H., Mahdavi, M., 2023. Do we really need complicated model architectures for temporal networks? In: Proc. Int. Conf. Learn. Represent.. pp. 29620–29646.

Duan, J., Wang, S., Zhang, P., Zhu, E., Hu, J., Jin, H., Liu, Y., Dong, Z., 2023. Graph anomaly detection via multi-scale contrastive learning networks with augmented view. In: Proc. AAAI Conf. Artif. Intell., vol. 37, (6), pp. 7459–7467.

Feng, Z., Wang, R., Wang, T., Song, M., Wu, S., He, S., 2026. A comprehensive survey of dynamic graph neural networks: Models, frameworks, benchmarks, experiments and challenges. IEEE Trans. Knowl. Data Eng. 38 (1), 26–46.

Fu, Y., Zhou, C., Li, J., Chen, L., 2025. Long-term evolutionary patterns matter: Self-supervised anomaly detection on dynamic graphs. Knowl.-Based Syst. 311, 113049.

Guo, J., Tang, S., Li, J., Pan, K., Wu, L., 2024a. RustGraph: Robust anomaly detection in dynamic graphs by jointly learning structural-temporal dependency. IEEE Trans. Knowl. Data Eng. 36 (7), 3472–3485.

Guo, Q., Yang, X., Zhang, F., Xu, T., 2024b. Perturbation-augmented graph convolutional networks: A graph contrastive learning architecture for effective node classification tasks. Eng. Appl. Artif. Intell. 129, 107616.

Hamilton, W., Ying, Z., Leskovec, J., 2017. Inductive representation learning on large graphs. Adv. Neural Inf. Process. Syst. 30.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., Sugiyama, M., 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. Adv. Neural Inf. Process. Syst. 31, 8535–8545.

Hassani, K., Khasahmadi, A.H., 2020. Contrastive multi-view representation learning on graphs. In: Proc. Int. Conf. Mach. Learn.. pp. 4116–4126.

Huang, L., Ruan, S., Decazes, P., Denœux, T., 2025. Deep evidential fusion with uncertainty quantification and reliability learning for multimodal medical image segmentation. Inf. Fusion 113, 102648.

Jin, M., Liu, Y., Zheng, Y., Chi, L., Li, Y.-F., Pan, S., 2021. Anemone: Graph anomaly detection with multi-scale contrastive learning. In: Proc. ACM Int. Conf. Inf. Knowl. Manag.. pp. 3122–3126.

Ju, W., Fang, Z., Gu, Y., Liu, Z., Long, Q., Qiao, Z., Qin, Y., Shen, J., Sun, F., Xiao, Z., et al., 2024. A comprehensive survey on deep graph representation learning. Neural Netw. 173, 106207.

Kipf, T.N., Welling, M., 2016. Variational graph auto-encoders. In: Adv. Neural Inf. Process. Syst. Worksh. Bayesian Deep Learning.

Kipf, T.N., Welling, M., 2017. Semi-supervised classification with graph convolutional networks. In: Proc. Int. Conf. Learn. Represent.. pp. 2713–2726.

Kumar, S., Hamilton, W.L., Leskovec, J., Jurafsky, D., 2018. Community interaction and conflict on the web. In: Proc. Web Conf.. pp. 933–943.

Kumar, S., Spezzano, F., Subrahmanian, V.S., Faloutsos, C., 2016. Edge weight prediction in weighted signed networks. In: Proc. IEEE Int. Conf. Data Min.. pp. 221–230.

Kumar, S., Zhang, X., Leskovec, J., 2019. Predicting dynamic embedding trajectory in temporal interaction networks. In: Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.. pp. 1269–1278.

Lee, J., Kim, S., Shin, K., 2024. Slade: Detecting dynamic anomalies in edge streams without labels via self-supervised learning. In: Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.. pp. 1506–1517.

Li, Z., Gao, Z., Zhang, G., Liu, J., Xu, L., 2024. Dynamic personalized graph neural network with linear complexity for multivariate time series forecasting. Eng. Appl. Artif. Intell. 127, 107291.

Li, J., Socher, R., Hoi, S.C.H., 2020. DivideMix: Learning with noisy labels as semi-supervised learning. In: Proc. Int. Conf. Learn. Represent.. pp. 8923–8936.

Liu, Y., Li, Z., Pan, S., Gong, C., Zhou, C., Karypis, G., 2021a. Anomaly detection on attributed networks via contrastive self-supervised learning. IEEE Trans. Neural Netw. Learn. Syst. 33 (6), 2378–2392.

Liu, M., Liu, Y., 2021. Inductive representation learning in temporal networks via mining neighborhood and community influences. In: Proc. ACM SIGIR Int. Conf. Res. Dev. Inf. Retr.. pp. 2202–2206.

Liu, M., Liu, Y., Ren, Q., Han, M., 2025a. Rethinking multi-level information fusion in temporal graphs: Pre-training then distilling for better embedding. Inf. Fusion 121, 103127.

Liu, Y., Pan, S., Wang, Y.G., Xiong, F., Wang, L., Chen, Q., Lee, V.C., 2021b. Anomaly detection in dynamic graphs via transformer. IEEE Trans. Knowl. Data Eng. 35 (12), 12081–12094.

Liu, G., Wang, C., Gong, Z., Zhang, J., Tang, S., Wang, H., 2025b. Hypergraph-driven anomaly detection in dynamic noisy graphs. IEEE Trans. Inf. Forensics Secur. 20, 9848–9863.

Lv, T., Wang, X., Lai, S., Xie, T., Zheng, X., Quan, D., Qi, Y., Zhou, F., 2026. Hybrid self-supervised learning based on higher-order structures for graph anomaly detection. Neurocomputing 663, 132040.

Ma, X., Wu, J., Xue, S., Yang, J., Zhou, C., Sheng, Q.Z., Xiong, H., Akoglu, L., 2023. A comprehensive survey on graph anomaly detection with deep learning. IEEE Trans. Knowl. Data Eng. 35 (12), 12012–12038.

Oord, A.v.d., Li, Y., Vinyals, O., 2018. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748.

Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T., Leiserson, C., 2020. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. vol. 34, (04), pp. 5363–5370.

Qiao, H., Niu, C., Chen, L., Pang, G., 2025. AnomalyGFM: Graph foundation model for zero/few-shot anomaly detection. In: Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.. pp. 2326–2337.

Rajaoarisoa, L., Kuk, M., Bobek, S., Sayed-Mouchaweh, M., 2024. Hybrid and co-learning approach for anomalies prediction and explanation of wind turbine systems. Eng. Appl. Artif. Intell. 133, 108046.

Rossi, E., Chamberlain, B., Frasca, F., Eynard, D., Monti, F., Bronstein, M., 2020. Temporal graph networks for deep learning on dynamic graphs. In: Proc. ICML Worksh. Graph Representation Learning and beyond.

Sankar, A., Wu, Y., Gou, L., Zhang, W., Yang, H., 2020. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In: Proc. Int. Conf. Web Search Data Min.. pp. 519–527.

Shou, Y., Cao, X., Meng, D., 2025. Spegcl: Self-supervised graph spectrum contrastive learning without positive samples. IEEE Trans. Neural Netw. Learn. Syst. 36 (11), 19546–19559.

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C.A., Cubuk, E.D., Kurakin, A., Li, C.-L., 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. Adv. Neural Inf. Process. Syst. 33, 596–608.

Suh, G.E., Clarke, D., Gassend, B., Van Dijk, M., Devadas, S., 2003. AEGIS: Architecture for tamper-evident and tamper-resistant processing. In: Proc. Int. Conf. Supercomput.. pp. 357–368.

Sun, F.-Y., Hoffman, J., Verma, V., Tang, J., 2020. InfoGraph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In: Proc. Int. Conf. Learn. Represent.. pp. 12313–12328.

Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z., 2008. ArnetMiner: Extraction and mining of academic social networks. In: Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.. pp. 990–998.

Thakoor, S., Tallec, C., Azar, M.G., Azabou, M., Dyer, E.L., Munos, R., Veličković, P., Valko, M., 2022. Large-scale representation learning on graphs via bootstrapping. In: Proc. Int. Conf. Learn. Represent.. pp. 19932–19952.

Tian, S., Dong, J., Li, J., Zhao, W., Xu, X., Wang, B., Song, B., Meng, C., Zhang, T., Chen, L., 2023. SAD: Semi-supervised anomaly detection on dynamic graphs. In: Proc. Int. Joint Conf. Artif. Intell.. pp. 2306–2314.

Trivedi, R., Farajtabar, M., Biswal, P., Zha, H., 2019. Dyrep: Learning representations over dynamic graphs. In: Proc. Int. Conf. Learn. Represent.. pp. 274–298.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y., 2018. Graph attention networks. In: Proc. Int. Conf. Learn. Represent.. pp. 2920–2931.

Veličković, P., Fedus, W., Hamilton, W.L., Liò, P., Bengio, Y., Hjelm, R.D., 2019. Deep graph infomax. In: Proc. Int. Conf. Learn. Represent.. pp. 6633–6649.

Wang, Y., Chang, Y.-Y., Liu, Y., Leskovec, J., Li, P., 2021a. Inductive representation learning in temporal networks via causal anonymous walks. In: Proc. Int. Conf. Learn. Represent.. pp. 7208–7228.

Wang, X., Li, H., Zhang, Z., Chen, H., Xiao, T., Li, K., Zhu, W., 2026. Uncertainty-aware disentangled dynamic graph attention network for out-of-distribution generalization. IEEE Trans. Pattern Anal. Mach. Intell. 48 (3), 2200–2220.

Wang, H., Qiao, C., Guo, X., Fang, L., Sha, Y., Gong, Z., 2021b. Identifying and evaluating anomalous structural change-based nodes in generalized dynamic social networks. ACM Trans. Web 15 (4), 19:1–19:22.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S., 2020. A comprehensive survey on graph neural networks. IEEE Trans. Neural Netw. Learn. Syst. 32 (1), 4–24.

Wu, N., Sun, Y., Zhao, Y., Wang, W., Liu, X., Li, J., 2026. Anomaly subgraph detection on multiple associated attributed networks. IEEE Trans. Neural Netw. Learn. Syst. 1–15.

Xu, Y., Peng, Z., Shi, B., Hua, X., Dong, B., 2024. Learning dynamic graph representations through timespan view contrasts. Neural Netw. 176, 106384.

Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., Achan, K., 2020. Inductive representation learning on temporal graphs. In: Proc. Int. Conf. Learn. Represent.. pp. 4379–4397.

Yang, X., Zhao, X., Shen, Z., 2025. A generalizable anomaly detection method in dynamic graphs. In: Proc. AAAI Conf. Artif. Intell., vol. 39, (20), pp. 22001–22009.

You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., Shen, Y., 2020. Graph contrastive learning with augmentations. Adv. Neural Inf. Process. Syst. 33, 5812–5823.

Yu, W., Cheng, W., Aggarwal, C.C., Zhang, K., Chen, H., Wang, W., 2018. Netwalk: A flexible deep embedding approach for anomaly detection in dynamic networks. In: Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.. pp. 2672–2681.

Zhang, H., Jiang, X., 2024. ConUMIP: Continuous-time dynamic graph learning via uncertainty masked mix-up on representation space. Knowl.-Based Syst. 306, 112748.

Zhang, Y., Pal, S., Coates, M., Ustebay, D., 2019. Bayesian graph convolutional neural networks for semi-supervised classification. In: Proc. AAAI Conf. Artif. Intell., vol. 33, (01), pp. 5829–5836.

Zhang, H., Wu, Q., Yan, J., Wipf, D., Yu, P.S., 2021. From canonical correlation analysis to self-supervised graph neural networks. Adv. Neural Inf. Process. Syst. 34, 76–89.

Zhao, Y., Wang, W., Wu, N., Liu, J., Sun, Y., Yu, W., Zu, Q., 2026. Attention-based anomaly detection in dynamic network. Big Data Min. Anal. 9 (1), 70–86.

Zheng, L., Li, Z., Li, J., Li, Z., Gao, J., 2019. AddGraph: Anomaly detection in dynamic graph using attention-based temporal GCN. In: Proc. Int. Joint Conf. Artif. Intell.. pp. 4419–4425.

Zheng, Y., Yi, L., Wei, Z., 2025. A survey of dynamic graph neural networks. Front. Comput. Sci. 19 (6), 196323.

Zhu, M., Qiu, L., Zhao, W., 2025. Dynamic graph contrastive learning based on learnable view generators. Knowl.-Based Syst. 113845.

Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., Wang, L., 2020. Deep graph contrastive representation learning. In: Proc. ICML Worksh. Graph Representation Learning and beyond.

Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., Wang, L., 2021. Graph contrastive learning with adaptive augmentation. In: Proc. Web Conf.. pp. 2069–2080.

Zhu, J., Zeng, W., Zhang, J., Tang, J., Zhao, X., 2023. Cross-view graph contrastive learning with hypergraph. Inf. Fusion 99, 101867.

Zuo, Y., Liu, G., Lin, H., Guo, J., Hu, X., Wu, J., 2018. Embedding temporal network via neighborhood formation. In: Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.. pp. 2857–2866.