Adaptive Context Evolution for Zero-Fine-Tuning Generalist Graph Anomaly Detection

Haoran Dan[1], Chen Chen[2], Ruishuang Sun[2], Yinhong Li[1], Wei He[3],
Zinwei Yan[2], Xiaoyi Lv[2], ChangYong Liu[4], and Enguang Zuo[5,6,*],
[1]School of Computer Science and Technology, Xinjiang University, Urumqi, China
[2]School of Software, Xinjiang University, Urumqi, China
[3]Xinjiang Jiuding Testing Technology Co.LTD, Urumqi, China
[4]Xinjiang Academy of Agricultural and Reclamation Sciences, Urumqi, China
[5]School of Intelligence Science and Technology, Xinjiang University, Urumqi, China
[6]Department of Electronic Engineering, Tsinghua University, BeiJing, China
{dhr999307, chen_chen, 107552304975, liyinhong}@stu.xju.edu.cn
{m15503819172, zhou2645067016}@163.com, {yzw, lvxiaoyi, zeg}@xju.edu.cn

***Abstract*—Graph Anomaly Detection (GAD) aims to identify anomalous nodes whose behavioral patterns or structural relationships deviate from the majority, a task that has attracted significant research attention. However, constructing a universal GAD framework capable of adapting to diverse real-world applications faces the following technical bottlenecks: on one hand, existing models struggle to achieve a balance between capturing complex long-range dependencies and maintaining high computational efficiency; on the other hand, mainstream methods are constrained by "generalization silos," lacking the cross-domain capability to handle data distribution shifts. To alleviate these issues, we propose AceGAD, a universal graph anomaly detection method. Specifically, AceGAD consists of two core components: first, to address the dilemma of reconciling long-range dependencies with computational efficiency, we designed the Linear Graph State-Space Encoder (LG-SSE), which reformulates graph propagation as a continuous linear state evolution process, achieving precise memory of global context while maintaining linear time complexity; second, to break through the limitations of generalization silos, we proposed the Context-Aware Metric Projector (CAM-Pro), which enables the model to adapt to new environments using only a few normal samples by constructing an adaptive metric space, achieving Zero-Fine-Tuning inference on new graphs. Extensive experiments conducted on eight public benchmark datasets across different domains demonstrate that AceGAD achieves superior performance, efficiency, and universality.**

***Index Terms*—Graph Anomaly Detection; Linear Graph State-Space Encoder; Context-Aware Metric Projector; Zero-Fine-Tuning**

I. Introduction

As a powerful data structure capable of abstracting entities and their complex interactions, graphs have been widely adopted to represent complex systems, such as citation networks, social networks, and financial transaction networks. Within these complex topological structures, mining anomalous nodes that deviate from normative patterns from massive data is critical for maintaining system security and risk management [1], [2].

*Corresponding author.

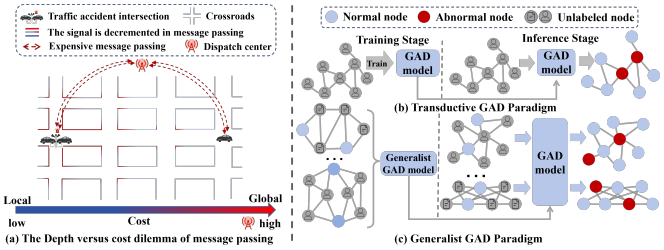


Fig. 1. (a) Illustration of the depth-cost dilemma in traffic incident propagation. (b) The Transductive GAD Paradigm. (c) The Generalist GAD Paradigm.

Despite the excellent detection performance of existing GAD methods, the field still faces two major challenges that limit their application in real-world scenarios. The first challenge is the Depth-Efficiency Dilemma, which refers to the inherent conflict between capturing long-range dependencies and maintaining computational efficiency. Complex anomaly patterns often do not exist solely locally but are latent in the non-linear interactions between nodes and their global neighbors. We use the propagation of traffic incidents to illustrate real-world message passing, as shown in Fig.1(a). When congestion signals caused by accidents at intersections gradually diminish across multiple junctions, the dispatch center transmits these signals with high fidelity but at a higher cost. This analogy reveals the status quo of the GAD field: traditional Graph Neural Networks (GNNs) are limited by finite receptive fields and struggle to capture deep dependencies by stacking layers; Although the global Transformer family of models can capture long-distance dependencies, their high computational resource requirements limit their practicability on large-scale graphs.

However, adapting GAD to multiple scenarios faces a second dilemma: the Generalization Silo, referring to the lack of model generality and generalization capability. Current mainstream GAD methods mostly follow a "one model, one dataset" training paradigm, as shown in Fig.1(b). This paradigm results from traditional methods fitting domain-

specific static distributions, relying on fixed decision boundaries, lacking mechanisms to dynamically adapt to new environments, and the inherent heterogeneity of graph data. Consequently, this forces models to be trained from scratch or fine-tuned whenever facing a new scenario. In contrast, an ideal generalist GAD method, as depicted in Fig.1(c), should be capable of learning general patterns across diverse source domains and achieving instant inference on any arbitrary new graph without fine-tuning.

To address these issues, this paper proposes AceGAD, which is a zero-fine-tuning universal graph anomaly detection framework based on adaptive context evolution. AceGAD achieves an effective unification of depth, efficiency, and generality through a novel information processing perspective. First, to resolve the dilemma of long-range dependencies and computational efficiency, we design the LG-SSE. This module breaks away from the traditional paradigm of stacking graph convolution layers by reconstructing information propagation on graphs as a continuous linear state-space evolution process. This mechanism endows the model with the long-term memory capabilities of State Space Models (SSMs), enabling it to precisely capture long-range dependencies from the global background to local entities while maintaining linear time complexity. Thereby, it utilizes the macro background to effectively contrast micro anomalies, alleviating the "Depth-Efficiency" dilemma and providing high-quality deep node representations for the subsequent module. Second, to tackle the Generalization Silo and label scarcity, we construct the CAM-Pro. This module discards dataset-specific fixed classification boundaries in favor of establishing an adaptive metric space. In a Few-Shot setting, it utilizes a self-queried attention mechanism to dynamically synthesize a Normalcy Anchor adapted to the current graph environment from a small number of normal samples. This mechanism essentially establishes a dynamic reference frame for each new environment, allowing AceGAD to adapt to brand-new graph data without any parameter updates, thus realizing efficient Zero-Fine-Tuning inference. The main contributions of this paper are summarized as follows:

- We propose AceGAD, which pioneers the integration of linear state-space evolution into generalist graph anomaly detection. With the design of LG-SSE, we successfully resolve the Depth-Efficiency Dilemma, achieving precise memory of the global context while maintaining linear complexity.
- We establish a Context-Aware Metric Projection paradigm and design the CAM-Pro to realize Zero-Fine-Tuning inference. This mechanism effectively breaks "the Generalization Silo" effect, establishing a new paradigm for generalist GAD characterized as Train Once, Apply Everywhere.
- We conduct extensive experiments on multiple public datasets across different domains. The results demonstrate that our proposed AceGAD consistently outperforms various state-of-the-art baseline models in anomaly detection performance (ACM $\uparrow$ 6.72%, cs $\uparrow$5.46%, tfinance $\uparrow$12.24%).

II. Related Work

Graph Anomaly Detection: GAD aims to identify nodes that exhibit significant anomalies in either node attributes or graph structures. Existing supervised GAD methods [3] focus on fitting specific decision boundaries by designing sophisticated graph convolutional operators, but they rely heavily on large-scale labeled data. Unsupervised GAD methods capture normal patterns through diverse learning techniques in a label-free setting [4], [5], including reconstruction-based methods [1], contrastive learning-based methods [4], and other auxiliary objectives [5], [6]. Although the aforementioned methods perform excellently on specific datasets, the mainstream paradigm typically follows a "one model, one dataset" training approach. Recently, to break through "generalization silos," researchers have begun exploring universal graph anomaly detection. Early ARC [7] leveraged contextual learning, enabling a single model to adapt and detect anomalies in new graphs in real time during inference with only a small number of normal samples.

State Space Models on Graphs: Recently, SSMs, have emerged as an efficient sequence modeling paradigm. With their linear time complexity $O(N)$ and robust long-range memory capabilities, they are regarded as potent competitors to Transformers. Consequently, researchers have begun extending SSMs to graph-structured data, termed Graph Mamba. For instance, [8] investigated the advantages of SSMs on temporal graphs from first principles; [9] proposed Graph Generated State Space Models (GG-SSMs) to handle non-local dependencies; and [10] further explored SSM extensions on directed graphs. some works have attempted to unify message passing with SSM mechanisms [11], and have even validated the effectiveness of Graph Mamba in complex applications such as medical image segmentation [12].

Graph In-Context Learning: To address the challenges of few-shot learning and zero-fine-tuning, In-Context Learning (ICL) has emerged as a prominent research hotspot. ICL allows models to instantly adapt to new tasks solely through contextual examples without updating model parameters. Although Large Language Models (LLMs) have demonstrated powerful Graph ICL capabilities [13], their prohibitive inference costs and massive memory footprint make them difficult to deploy in practical applications. Consequently, researchers have begun exploring efficient graph in-context learning mechanisms. For instance, [14] provides a theoretical analysis of ICL on graphs, while [15] proposes a pseudo-labeling framework based on graph structures to optimize contextual prompts.

III. Method

This section introduces AceGAD, a zero-fine-tuning universal graph anomaly detection framework based on adaptive context evolution (overall architecture shown in Fig.2). The framework comprises three core components: First, the data

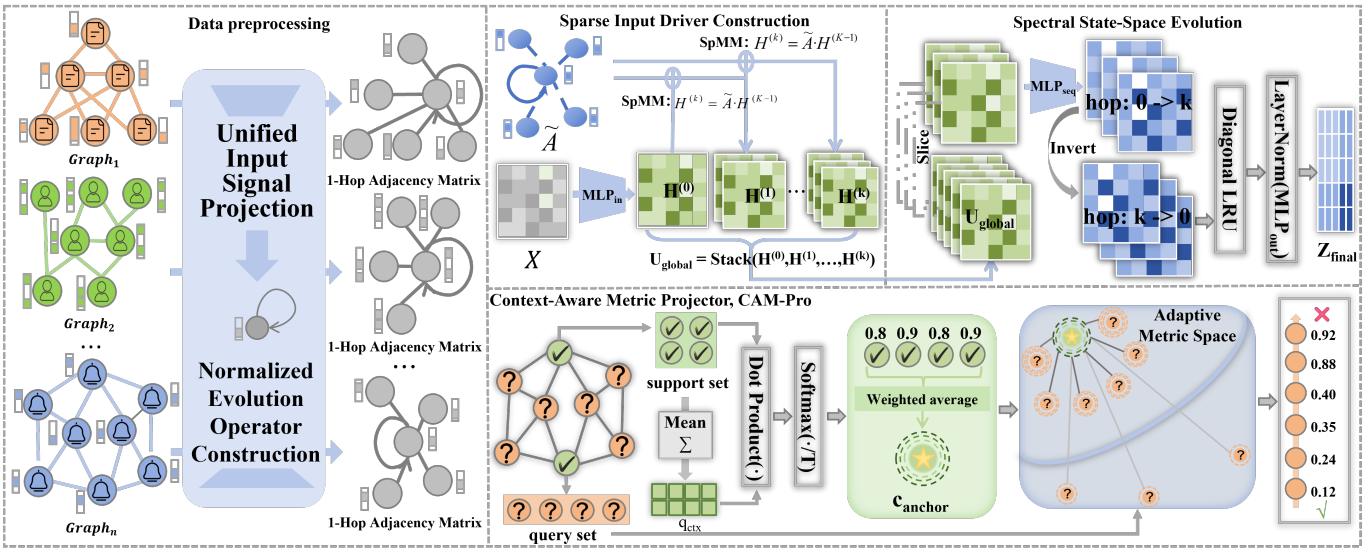


Fig. 2. Overall workflow of the AceGAD model, including the data preprocessing module, LG-SSE and CAM-Pro.

preprocessing module standardizes raw graph data into unified input signals and normalized evolution operators; Second, LG-SSE efficiently captures deep representations containing global context through continuous state evolution in the spectral domain with linear complexity, resolving the trade-off between long-range dependencies and efficiency. Finally, CAM-Pro constructs an adaptive metric space by dynamically synthesizing "normal anchors" tailored to the current environment, enabling zero-fine-tuning precise inference on new graphs.

A. Data preprocessing

Unified Input Signal Projection: We introduce a projection function Φ to serve as a non-parametric Channel Adapter. This adapter employs a statistical transformation strategy to map node features into a unified latent signal space, a process defined as:

$$\mathbf{X}_{aligned} = \Phi(\mathbf{X}_{raw}, d) \tag{1}$$

where $\mathbf{X}_{raw} \in \mathbb{R}^{N \times d_{raw}}$ represents the raw input signal matrix, d denotes the target dimension, and Φ adaptively executes strategies based on the relationship between the raw dimension d_{raw} and the target dimension d.

Normalized Evolution Operator Construction: First, we introduce self-loops to the original adjacency matrix A, yielding $\tilde{A} = A + I$. This ensures that nodes can achieve State Retention during discrete state evolution steps. Subsequently, the normalized operator is calculated as:

$$\hat{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} \tag{2}$$

where $\tilde{D}$ is the degree matrix of $\tilde{A}$. This Spectral Normalization constrains the eigenvalues of the operator within the interval $[-1, 1]$, theoretically guaranteeing the convergence and stability of the subsequent linear state evolution process.

B. Linear Graph State-Space Encoder

Sparse Input Driver Construction: First, a linear projection layer MLP_{in} is used to map the unified input signal $\mathbf{X}_{aligned}$ into the hidden space, initializing the 0-th order state $\mathbf{H}^{(0)}$:

$$\mathbf{H}^{(0)} = \text{MLP}_{in}(\mathbf{X}_{aligned}) \tag{3}$$

Subsequently, the system executes max_dist steps of explicit spatial diffusion to generate a series of state snapshots:

$$\mathbf{H}^{(k)} = \hat{A} \cdot \mathbf{H}^{(k-1)}, \quad k = 1, 2, ..., max\_dist \tag{4}$$

where $\mathbf{H}^{(k)}$ represents the physical mapping of k-th order neighborhood information within the system. Once generated, all state snapshots are stacked (Stack); this process involves no non-linear activations and is essentially constructing a multi-scale structural driving tensor $\mathbf{U}_{global}$, which stores the complete information path for each node from itself to distant neighbors:

$$\mathbf{U}_{global} = \text{Stack}(\mathbf{H}^{(0)}, \mathbf{H}^{(1)}, ..., \mathbf{H}^{(max\_dist)}) \in \mathbb{R}^{N \times (max\_dist+1) \times D_h} \tag{5}$$

max_dist defines the maximum evolution horizon of the system's lookback, and $\mathbf{U}_{global}$ defines the pre-computed global feature sequence library.

Reverse-Causal Driver Organization: During the batch training phase, we extract the driver sequences for the nodes in the current batch from U_{global}. To implement the concept of context-awareness, we redefine the direction of information integration. We perform Reverse Organization on the extracted sequences along the spatial dimension. This operation essentially restructures the spatial propagation sequence by establishing an aggregation order that points from max_dist hops to 0 hops:

$$\mathbf{U}_{batch} = \text{Flip}(\text{MLP}_{seq}(\mathbf{U}_{global}[\text{indices}]), \text{dim} = 1) \tag{6}$$

where $\mathbf{U}_{global}[\text{indices}]$ represents the sub-sequences sliced from the global feature sequence library, MLP_{seq} further refines the features of the sequence, and $\mathbf{U}_{batch}$ is the constructed reverse driver sequence.

Spectral State-Space Evolution: Once the reverse driver sequence $\mathbf{U}_{batch}$ is constructed, it is fed into the core evolution unit—the Spectral State-Space Evolver. We employ the Diagonal Linear Recurrent Unit (Diagonal LRU) as the evolution kernel. This component is essentially a discretized linear state-space equation:

$$\mathbf{h}_t = \Lambda \odot \mathbf{h}_{t-1} + \mathbf{B}\mathbf{u}_t \tag{7}$$

where $\mathbf{u}_t$ is the input from the driver sequence $\mathbf{U}_{batch}$ at step t; $\mathbf{h}_t \in \mathbb{C}^{D_h}$ is the Complex Hidden State; and $\Lambda \in \mathbb{C}^{D_h}$ is the diagonal state transition matrix. We parameterize Λ in the form

$\lambda_j = \gamma_j e^{i\theta_j}$. In this spectral parameterization, the magnitude γ precisely controls the memory decay of long-range dependencies, while the phase θ captures the structural oscillation of graph signals; $\odot$ denotes element-wise multiplication.

As the evolution step t proceeds from 1 to max_dist, the hidden state $\mathbf{h}_{\text{max\_dist}}$ completes a full state integration from the global context to the local entity. To transform this set of complex states—which contains deep dynamical information—into a real-valued representation suitable for downstream anomaly detection tasks, we introduce a State Readout and Projection mechanism. The final deep node representation $\mathbf{z}_{final}$ is calculated as follows:

$$\mathbf{z}_{final} = \text{LayerNorm}(\text{MLP}_{out}(\text{Real}(\mathbf{C}\mathbf{h}_{max\_dist}))) \tag{8}$$

where $\mathbf{C} \in \mathbb{C}^{D_h \times D_h}$ is the Output Mixing Matrix; the $\text{Real}(\cdot)$ operation performs Domain Transformation, discarding imaginary components to map signals from the abstract complex spectral domain back to the physical real-valued feature space. MLP_{out} is a non-linear output projection layer designed to enhance the expressive power of the linear state-space model and adapt to complex anomaly detection boundaries. LayerNorm is used to normalize the distribution of the final embedding, ensuring input stability for the CAM-Pro.

C. Context-Aware Metric Projector

Dynamic Normalcy Anchor Synthesis: First, we extract the context query vector $\mathbf{q}_{ctx}$ for the support set, defined as the centroid of the support sample representations:

$$\mathbf{q}_{ctx} = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{z}_i \in \mathcal{S}} \mathbf{z}_i \tag{9}$$

where $\mathcal{S} = \{\mathbf{z}_1, \ldots, \mathbf{z}_K\}$ represents the support set containing K known normal samples (i.e., the number of shots), and $\mathbf{z}_i$ is the deep node representation generated by the LG-SSE encoder. Subsequently, we calculate the consistency score between each support sample $\mathbf{z}_i$ and the context query vector $\mathbf{q}_{ctx}$. This process evaluates the representativeness of each sample regarding the normal pattern:

$$e_i = \mathbf{z}_i^T \cdot \mathbf{q}_{ctx} \tag{10}$$

$$\alpha_i = \frac{\exp(e_i/\tau)}{\sum_{j=1}^{|\mathcal{S}|} \exp(e_j/\tau)} \tag{11}$$

where e_i represents the unnormalized attention logits, measuring the directional alignment between $\mathbf{z}_i$ and the centroid; $\tau \in (0, 1]$ is a learnable Temperature Parameter used to adjust the sharpness of the attention distribution. The final normalcy anchor $\mathbf{c}_{anchor}$ is generated through weighted aggregation, incorporating a learnable scaling factor γ to optimize its magnitude within the metric space:

$$\mathbf{c}_{anchor} = \gamma \sum_{i=1}^{|\mathcal{S}|} \alpha_i \cdot \mathbf{z}_i \tag{12}$$

Here, γ is a learnable scalar scaling parameter that allows the model to adaptively adjust the norm of the anchor vector to obtain a superior geometric representation.

Adaptive Metric Space Construction: To endow the generated metric space with superior discriminability, we design a Geometric Constraint Objective to guide the topological evolution of the space. This objective consists of four components.

Positive Attraction Loss $\mathcal{L}_{pos}$: This component minimizes the squared Euclidean distance between normal query nodes and the normal anchor $\mathbf{c}_{anchor}$, encouraging all normal samples to cluster tightly around the normal anchor within the embedding space.

$$\mathcal{L}_{pos} = \mathbb{E}_{(z_q, y_q)|y_q=0}[\|z_q - \mathbf{c}_{anchor}\|_2^2] \tag{13}$$

Negative Repulsion Loss $\mathcal{L}_{neg}$: We employ a squared hinge loss to penalize anomalous query samples. Penalties are incurred only when anomalous samples intrude into the safety radius $M_{adaptive}$ of the normal anchor $\mathbf{c}_{anchor}$, thereby effectively repelling these anomalous samples from the normal cluster.

$$\mathcal{L}_{neg} = \mathbb{E}_{(z_q, y_q)|y_q=1}[\max(0, M_{adaptive} - \|z_q - \mathbf{c}_{anchor}\|_2)^2] \tag{14}$$

Here, $M_{adaptive} = M_{base} + \mu_{\mathcal{S}} + \sigma_{\mathcal{S}}$ represents a Context-Aware Dynamic Margin. M_{base} is a base margin hyperparameter, while $\mu_{\mathcal{S}}$ and $\sigma_{\mathcal{S}}$ denote the mean and standard deviation, respectively, of the distances from samples within the support set to the normal anchor $\mathbf{c}_{anchor}$. This design enables the safety radius to undergo Adaptive Breathing based on the compactness of the current normal cluster, significantly enhancing the model's adaptability to normal clusters of varying morphological shapes.

Intra-Class Compactness Loss $\mathcal{L}_{com}$: Serving as a regularization term, this component directly constrains the distances between samples within the support set and the normal anchor. This prevents the normal cluster from becoming effectively sparse and enhances intra-class cohesion.

$$\mathcal{L}_{com} = \mathbb{E}_{\mathbf{z}_i \in \mathcal{S}}[\|\mathbf{z}_i - \mathbf{c}_{anchor}\|_2^2] \tag{15}$$

Anchor Regularization Loss $\mathcal{L}_{reg}$: This term constrains the norm of the anchor vector to prevent numerical drift.

$$\mathcal{L}_{reg} = \|\mathbf{c}_{anchor}\|_2 \tag{16}$$

The Total Loss is formulated as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{pos} + w_{neg}\mathcal{L}_{neg} + w_{com}\mathcal{L}_{com} + w_{reg}\mathcal{L}_{reg} \tag{17}$$

where $w_{neg}, w_{com}, w_{reg}$ are hyperparameters used to balance the intensity of each geometric constraint. By minimizing this objective, CAM-Pro is essentially learning a Universal Metric Law—specifically, how to map complex graph structural discrepancies into standardized geometric distance deviations.

Zero-Fine-Tuning Anomaly Metric: During the inference phase, AceGAD demonstrates its Zero-Fine-Tuning capability as a universal framework when encountering an entirely new graph dataset G. First, the normalcy anchor $\mathbf{c}_{anchor}$ specific to graph G is generated instantaneously using the support set. Subsequently, the Geometric Drift Distance of the query node $\mathbf{z}_{query}$ relative to this anchor is calculated directly as the anomaly score:

$$Score(q) = \|\mathbf{z}_{query} - \mathbf{c}_{anchor}\|_2^2 \tag{18}$$

This distance directly quantifies the degree to which a node deviates from the canonical patterns of its current environment.

D. Complexity Analysis

Assume that the total number of nodes in graph G is N, and the total number of edges is $|\mathcal{E}|$. Let D denote the channel dimension of the latent signal space, and L (corresponding to max_dist) denote the maximum historical evolution horizon. In batch processing, the batch size is set to b, and the support set size is K.

The computational complexity of AceGAD primarily consists of two phases: a one-time sparse input driver construction phase and a batch-based online state evolution phase. Under

TABLE I
ANOMALY DETECTION PERFORMANCE IN TERMS OF AUROC (IN PERCENT, MEAN±STD). HIGHLIGHTED ARE THE RESULTS RANKED **FIRST**, **SECOND**.

Method	Cora	Citeseer	ACM	BlogCatalog	Weibo	Amazon	cs	tfinance
Supervised-Pre-Train Only								
GCN	59.64±8.30	60.27±8.11	60.49±9.65	56.19±6.39	76.64±17.69	46.63±3.47	66.64±7.69	61.43±8.62
GAT	50.06±2.65	51.59±3.49	48.79±2.73	50.40±2.80	53.06±7.48	50.52±17.22	56.16±6.52	51.35±2.75
BGNN	42.45±11.57	42.32±11.82	44.00±13.69	47.67±8.52	32.75±35.35	52.26±3.31	45.35±15.37	41.23±11.64
BWGNN	54.06±3.27	52.61±2.88	67.59±0.70	56.34±1.21	53.38±1.61	55.26±16.95	58.32±1.67	62.51±1.73
GHRN	59.89±6.57	56.04±9.19	55.65±6.37	57.64±3.48	51.87±14.18	49.48±17.13	54.82±8.25	55.33±3.54
Unsupervised - Pre-Train Only								
DOMINANT	66.53±1.15	69.47±2.02	70.08±2.34	74.25±0.65	**92.88±0.32**	48.94±2.69	72.88±1.34	71.32±4.37
CoLA	63.29±8.88	62.84±9.52	66.85±4.43	50.04±3.25	16.27±5.64	47.40±7.97	56.21±4.48	63.25±3.63
HCM-A	54.28±4.73	48.12±6.80	53.70±4.64	55.31±0.57	65.52±12.58	43.99±0.72	61.42±6.53	53.70±4.64
TAM	62.02±2.39	72.27±0.83	74.43±1.59	49.86±0.73	71.54±0.18	56.06±2.19	77.24±2.28	67.41±2.54
Unsupervised - Pre-Train & Fine-Tune								
DOMINANT	72.23±0.34	74.69±0.32	74.34±0.12	**74.61±0.04**	92.21±0.10	59.06±2.80	79.23±1.12	74.32±1.28
CoLA	67.62±4.26	70.75±3.42	69.11±0.67	62.49±3.38	31.55±6.02	52.51±6.66	69.76±3.34	69.17±2.57
HCM-A	56.45±4.93	55.54±4.07	57.69±3.59	55.10±0.29	71.89±2.79	42.20±0.55	74.69±2.29	67.69±3.54
TAM	62.56±2.10	76.54±1.33	**86.29±1.57**	57.69±0.88	71.73±0.16	57.13±1.59	73.53±2.36	73.29±5.17
Generalist								
ARC	**85.33±0.44**	**90.64±0.32**	79.27±0.17	**74.81±0.28**	89.24±0.35	**69.77±3.88**	**82.44±0.48**	**75.60±3.01**
Ours								
AceGAD	**85.93±0.21**	**90.08±0.13**	**93.01±1.20**	74.32±1.34	**94.72±1.79**	**70.93±7.63**	**87.90±4.00**	**87.84±2.35**

large-scale sparse graph conditions ($|\mathcal{E}| \gg N$), the construction phase is dominated by L sparse matrix multiplications, with a complexity of approximately $O(L \cdot |\mathcal{E}| \cdot D)$. In the online inference phase, the linear properties of the spectral state-space evolution significantly reduce the computational overhead. Consequently, the overall complexity of batch processing is dominated by sequence transformations, amounting to approximately $O((b+K) \cdot L \cdot D^2)$. This architecture achieves linear scalability with respect to the propagation depth L. Combined with the zero-fine-tuning mechanism, it substantially reduces the computational costs for capturing long-range dependencies and enabling cross-domain deployment.

IV. EXPERIMENTS AND RESULT ANALYSIS

A. Experimental Setup

Datasets. To train and evaluate the universal GAD model, we trained baseline models and the AceGAD model on one set of graph datasets and conducted testing on a separate set of graph datasets. To ensure a comprehensive evaluation, the selected datasets span multiple domains, including citation networks, social networks, and e-commerce co-review networks. Each domain's dataset contains either injected anomalies or real-world anomalies [4], [6]. Inspired by [16], our training set is composed of the largest-scale dataset within each domain, while the remaining datasets are reserved as the test set. The training set T_{train} includes PubMed, Flickr, Questions, and YelpChi. The test set T_{test} includes Cora, Citeseer, ACM, BlogCatalog, Weibo, Amazon, cs, and tfinance.

Baselines. To comprehensively evaluate the performance of AceGAD, we compare it against a series of representative supervised and unsupervised GAD methods. Supervised methods include traditional GNN approaches, such as GCN [17] and GAT [18], as well as three methods specifically designed for GAD: BGNN [19], BWGNN [2], and GHRN [3]. Unsupervised methods encompass four different design paradigms: the generative method DOMINANT [1], the contrastive learning method CoLA [4], the hop-count prediction method HCM-A [5], and the affinity-based method TAM [6]. Few-shot methods include the in-context learning-based approach ARC [7].

Evaluation Metrics and Implementation. Evaluation Metrics and Implementation. We employ AUROC as the evaluation metric [6], reporting the mean ± standard deviation across five independent experiments. AceGAD is trained on four datasets and evaluated on eight test sets. Baseline comparisons are defined as follows: Supervised models are evaluated only in the "pre-training only" mode. Unsupervised models additionally include a "pre-training & fine-tuning" mode, which allows leveraging self-supervised objectives for fine-tuning on test images. To validate generalizability, all models align feature spaces via projection layers and employ a fixed set of hyperparameters determined by random search across all test sets, without dataset-specific parameter tuning.

B. Performance Comparison

Across test sets spanning eight distinct domains, AceGAD demonstrates outstanding performance (Table I). It achieves state-of-the-art results on six datasets, significantly outperforming advanced baselines—for instance, improving by 6.72%, 5.46%, and 12.24% on ACM, cs, and tfinance, respectively. In contrast, traditional baselines exhibited negative transfer under "pre-training only" conditions. Even after fine-tuning, their performance remained inferior to AceGAD, which requires no parameter updates. By abandoning the inefficient "pre-training + fine-tuning" approach, AceGAD employs a context-aware metric projection paradigm to synthesize valid anchors on-the-fly. This achieves detection performance

TABLE II
AUROC OF ACEGAD AND ITS VARIANTS.

Variant	Cora	Citeseer	ACM	BlogCatalog	Weibo	Amazon	cs	tfinance
AceGAD	**85.93**	**90.08**	**93.01**	**74.12**	**94.72**	**70.93**	**87.9**	**87.84**
AceGAD w/o Sequential Processing	81.33	82.15	80.64	63.28	93.84	53.31	72.94	75.64
AceGAD w/o Flip	83.57	85.47	92.39	71.33	93.96	60.09	85.47	81.28
AceGAD w/o Attention	84.21	88.69	92.94	71.93	92.14	68.94	87.31	85.37

surpassing finely-tuned baselines with zero fine-tuning and minimal deployment overhead.

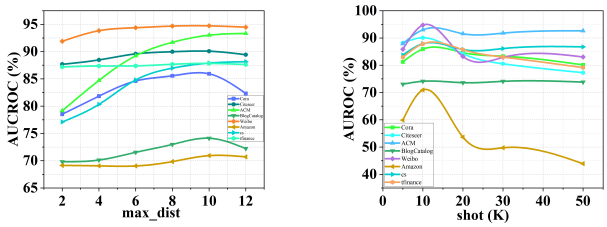
C. Ablation Study

We conducted ablation experiments to thoroughly analyze the contributions of three core components—spectral domain state space evolver, reverse-order causal reordering, and self-querying dot-product attention mechanism—across eight datasets. Detailed experimental results are presented in Table II. As expected, the integrated model achieved the best overall performance. Notably, the spectral domain state space evolver contributed most significantly to performance gains, achieving AUROC improvements of 12.37%, 14.96%, and 12.2% on the ACM, cs, and tfinance datasets, respectively. This demonstrates the necessity of the evolutionary mechanism within the linear graph state space encoder, which is crucial for anomaly detection accuracy. Without reverse-order causal reordering, forward-order inputs cause idiosyncratic information at target nodes to be diluted by background noise during long-distance evolution to the endpoint. The Query Dot Product Attention mechanism dynamically suppresses weights of marginal samples while focusing on high-confidence core samples, enabling the construction of a more robust and compact metric benchmark.

D. Parameter Sensitivity Analysis

Validity of Maximum Evolution Horizon (max_dist): To validate the effectiveness of LG-SSE in capturing long-range dependencies, we analyzed the impact of varying the maximum evolutionary range max_dist from 2 to 12 (experimental results shown in Fig.3(a)). Results show that model performance significantly improves as max_dist increases to moderate ranges, confirming the universal importance of long-range dependencies for identifying complex anomaly patterns. However, when $max\_dist > 10$, performance tends toward saturation or even slight decline due to introduced noise. Experiments indicate an optimal depth range that maximizes signal-to-noise ratio (e.g., [8,10]).

Validity of Shot Number (K): To assess AceGAD's sensitivity to the support set size K, we analyzed performance variations as K ranged from 5 to 50 (Fig.3(b)). Results demonstrate that near-optimal performance is achieved with only a minimal number of samples ($K = 10$), strongly validating CAM-Pro's capability to efficiently extract normal patterns. Notably, performance tends to plateau or even decline when $K > 10$. This reveals that excessively large support sets introduce boundary sample noise, causing synthetic normal anchors to drift and blurring the fine boundaries with anomalies.



(a) (b)

Fig. 3. (a) Performance of different max_dist. (b) Validity of Shot Number.

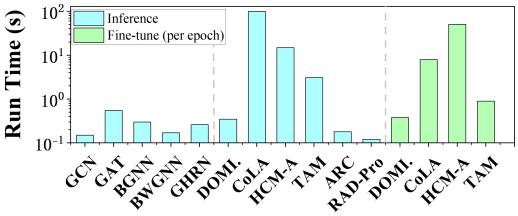


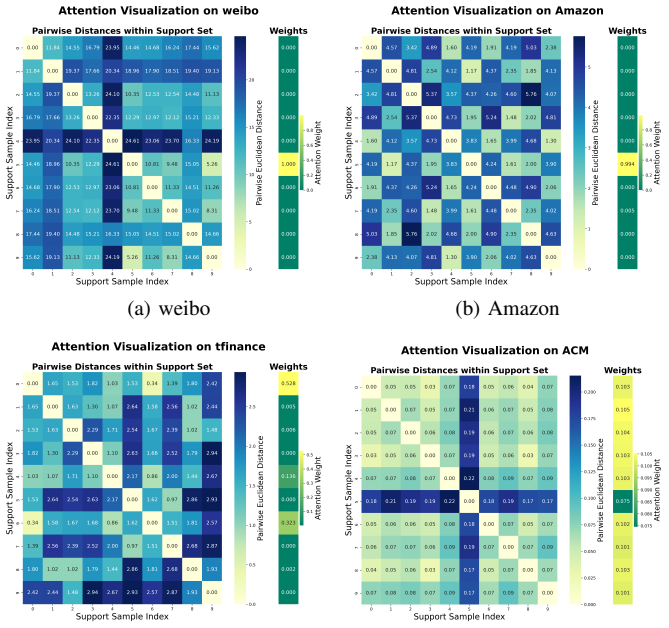
Fig. 4. Time Comparison.

E. Efficiency Analysis

To evaluate AceGAD's operational efficiency, we compared inference and fine-tuning times on the ACM dataset. As shown in Fig.4, first, thanks to the linear time complexity of the linear graph state space encoder, the model achieves extremely fast inference speeds. It not only demonstrates performance comparable to the fastest GCNs but also significantly outperforms unsupervised methods in terms of efficiency. Furthermore, dataset-specific fine-tuning consumes more time compared to inference. The advantage of our AceGAD model lies in completely eliminating the need for time-consuming fine-tuning on new graphs.

F. Visualization

To gain deeper insights into the underlying workings and interpretability of our proposed self-query dot-product attention mechanism, we visualized its decision-making process across multiple benchmark datasets. As shown in Fig.5, this mechanism employs an adaptive weighting strategy that dynamically switches between various approaches based on the geometric distribution of support set samples in the embedding space. We observed the following findings: In the Weibo and Amazon datasets, where support set samples exhibit high dispersion, this mechanism adopts a single-point dominance strategy, assigning nearly all weights to a relatively central anchor point to counteract noise interference from peripheral points. In the tfinance dataset, which exhibits clear core clusters, it switches to a core-subset consensus strategy, primarily weighting interactions among intra-cluster members; In the ACM dataset, where the support set samples are inherently highly homogeneous, the model adaptively converges to an optimal average-weighting strategy, as it determines that fine-grained weight differentiation is unnecessary in this scenario.



(a) weibo (b) Amazon

(c) tfinance (d) ACM

Fig. 5. Visualization of the Attention Mechanism Generating Prototypes.

V. Conclusions

This paper addresses the cutting-edge challenge of Generalist GAD by proposing a novel "train once, use everywhere" solution. AceGAD centers on two core innovations: First, we designed LG-SSE, which reframes long-range dependency capture in the generalist GAD setting as a continuous state evolution process within the spectral domain. This achieves precise memory of global context while maintaining linear time complexity, effectively resolving the "depth-efficiency" dilemma. Second, we establish CAM-Pro, which dynamically synthesizes "normal anchor points" through a self-querying dot-product attention mechanism, enabling a metric-learning-based zero-fine-tuning inference paradigm. Extensive experiments across real-world datasets demonstrate AceGAD's superior performance, computational efficiency, and versatility.

Future research will explore adaptive clustering mechanisms to automatically identify normal substructures on the support set and generate multiple anchor points, thereby enhancing the model's robustness in complex scenarios.

VI. Acknowledgments

This work was supported by the Key Scientific and Technological Research Program of Xinjiang Production and Construction Corps (2023AB062), the Youth Science Fund of the Natural Science Foundation of Xinjiang Uygur Autonomous Region (2025D01C298), the Xinjiang Uygur Autonomous Region "Tianchi Talents" Introduction Program and The Key Research and Development Program of Xinjiang Uygur Autonomous Region (2024B03028) .

References

[1] Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu, "Deep anomaly detection on attributed networks," in *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM, 2019, pp. 594–602.

[2] Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li, "Rethinking graph neural networks for anomaly detection," in *International Conference on Machine Learning*. PMLR, 2022, pp. 21076–21089.

[3] Yuan Gao, Xiang Wang, Xiangnan He, Zhenguang Liu, Huamin Feng, and Yongdong Zhang, "Addressing heterophily in graph anomaly detection: A perspective of graph spectrum," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 1528–1538.

[4] Yixin Liu, Zhao Li, Shirui Pan, Chen Gong, Chuan Zhou, and George Karypis, "Anomaly detection on attributed networks via contrastive self-supervised learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 6, pp. 2378–2392, 2021.

[5] Tianjin Huang, Yulong Pei, Vlado Menkovski, and Mykola Pechenizkiy, "Hop-count based self-supervised anomaly detection on attributed networks," in *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*. Springer, 2022, pp. 225–241.

[6] Hezhe Qiao and Guansong Pang, "Truncated affinity maximization: One-class homophily modeling for graph anomaly detection," *Advances in Neural Information Processing Systems*, vol. 36, pp. 49490–49512, 2023.

[7] Yixin Liu, Shiyuan Li, Yu Zheng, Qingfeng Chen, Chengqi Zhang, and Shirui Pan, "ARC: A generalist graph anomaly detector with in-context learning," *Advances in Neural Information Processing Systems*, vol. 37, pp. 50772–50804, 2024.

[8] Jintang Li, Ruofan Wu, Xinzhou Jin, et al., "State space models on temporal graphs: A first-principles study," in *Proceedings of the 38th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025, pp. 127030–127058.

[9] Nikola Zubic and Davide Scaramuzza, "GG-SSMs: Graph-generating state space models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 28863–28873.

[10] Junzhi She, Xunkai Li, Rong-Hua Li, and Guoren Wang, "State space models over directed graphs," *arXiv preprint arXiv:2509.13735*, 2025.

[11] Andrea Ceni, Alessio Gravina, Claudio Gallicchio, Davide Bacciu, Carola-Bibiane Schonlieb, and Moshe Eliasof, "Message-passing state-space models: Improving graph learning with modern sequence modeling," *arXiv preprint arXiv:2505.18728*, 2025.

[12] Yueyue Zhu, Haolin Lv, Geng Chen, Zhonghao Zhang, Haotian Jiang, and Yong Xia, "Hybrid Graph Mamba: Unlocking Non-Euclidean potential for accurate polyp segmentation," in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2025, pp. 277–286.

[13] Zhengyu Hu, Yichuan Li, Zhengyu Chen, Jingang Wang, Han Liu, Kyumin Lee, and Kaize Ding, "Let's ask GNN: Empowering large language model for graph in-context learning," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, Miami, Florida, USA, 2024, pp. 1396–1409, Association for Computational Linguistics.

[14] Core Francisco Park, Andrew Lee, Ekdeep Singh Lubana, Yongyi Yang, Maya Okawa, Kento Nishi, Martin Wattenberg, and Hidenori Tanaka, "ICLR: In-context learning of representations," *arXiv preprint arXiv:2501.00070*, 2024.

[15] Rui Lv, Zaixi Zhang, Kai Zhang, Qi Liu, Weibo Gao, Jiawei Liu, Jiaxian Yan, Linan Yue, and Fangzhou Yao, "GraphPrompter: Multi-stage adaptive prompt optimization for graph in-context learning," in *Proceedings of the 2025 IEEE 41st International Conference on Data Engineering (ICDE)*. IEEE Computer Society, 2025, pp. 3917–3930.

[16] Kaiwen Dong, Haitao Mao, Zhichun Guo, and Nitesh V Chawla, "Universal link predictor by in-context learning on graphs," *arXiv preprint arXiv:2402.07738*, 2024.

[17] Thomas N. Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

[18] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio, "Graph Attention Networks," in *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018.

[19] Sergei Ivanov and Liudmila Prokhorenkova, "Boost then convolve: Gradient boosting meets graph neural networks," in *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.