TEMPORALLY CONTEXTUALIZED SUPERVISION FOR DYNAMIC GRAPH ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Dynamic graph anomaly detection aims to identify abnormal behaviors in evolving interaction streams. Existing methods often rely on point-level evidence scores, such as drift scores that capture abnormal temporal deviations in interaction patterns or node representations, and reconstruction scores that measure reconstruction inconsistency with historical graph states. Although such point-level evidence scores can characterize abnormality-related signals, using them as isolated training signals assumes that the evidence for an anomaly is temporally aligned with the observed label at the same interaction. In this work, we show that this assumption does not always hold in dynamic graphs: the evidence associated with an annotated anomaly may emerge before the labeled interaction, persist around it, or extend beyond it, leading to an evidence–label temporal mismatch. To address this issue, we propose TECOS (**Te**mporally **Co**ntextualized **S**upervision), a temporally contextualized supervision framework for dynamic graph anomaly detection. Instead of applying supervision to isolated point-level evidence, TECOS uses short-range source-wise history together with source- and destination-side representations to learn a bounded correction to a reconstruction–drift base score, better aligning supervision with temporally misaligned anomaly evidence. We further provide empirical and theoretical analyses to characterize when point-level learning remains sufficient and when temporal context becomes beneficial. Experiments on seven temporal interaction datasets show that TECOS achieves the best overall performance across AUC and AP, demonstrating the effectiveness of temporally contextualized supervision.

1 INTRODUCTION

Dynamic graph anomaly detection (DGAD) aims to identify abnormal behavior in evolving interaction streams, with applications in fraud detection, network security, and social platforms (Ekle & Eberle, 2024; Ma et al., 2021).

Static graph anomaly detection commonly detects anomalous nodes by modeling structural and attribute deviations on a fixed graph, for example through graph autoencoding or reconstruction-based objectives (Li et al., 2017; Ding et al., 2019; Zhou & Paffenroth, 2017). DGAD instead requires jointly modeling relational dependencies and their temporal evolution, making it essential to capture not only whether an interaction is anomalous but also how anomalous behaviors emerge and evolve over time (Xu et al., 2020; Kazemi et al., 2020; Skarding et al., 2021). This temporal modeling requirement makes DGAD challenging, since anomalous evidence may span neighboring interactions rather than a single labeled event.

Existing dynamic graph representation learning methods capture interaction history through evolving node memories or aggregation over previous events (Trivedi et al., 2019; Kumar et al., 2019; Rossi et al., 2020; Xu et al., 2020), and use such representations to characterize departures from expected behavior. For example, SLADE learns reconstruction and temporal-consistency signals through self-supervised objectives, producing event-level anomaly evidence without anomaly labels (Lee et al., 2024). These point-level scores, typically derived from reconstruction inconsistency or representation drift, provide useful local evidence. However, directly applying supervision or evaluation to isolated evidence points implicitly assumes that the evidence is temporally aligned with the label at the same

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

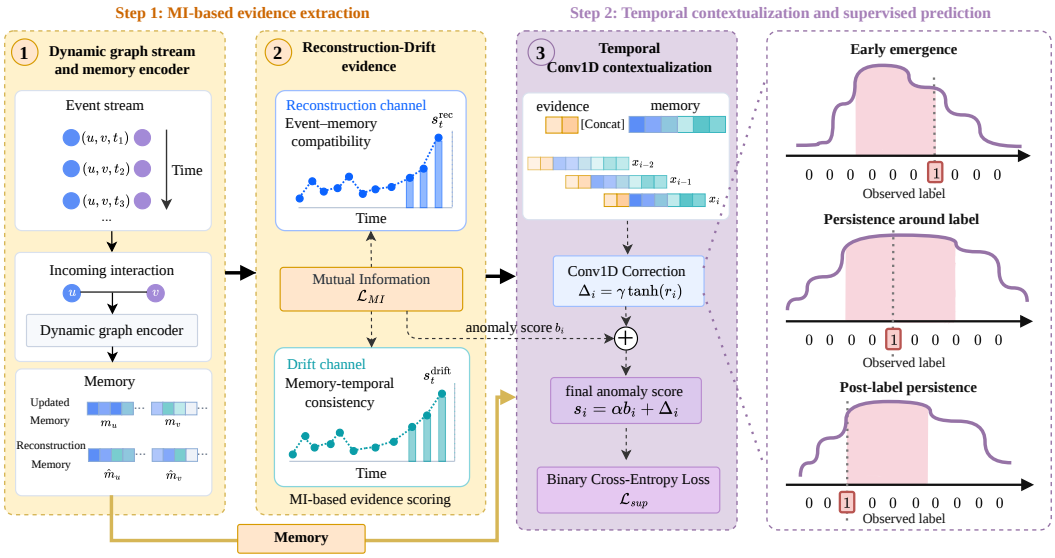


Figure 1: Framework of TECOS. Stage 1 extracts reconstruction–drift MI-based evidence with a memory-based dynamic graph encoder. Stage 2 forms source-wise temporal sequences from evidence and memory representations and applies Conv1D contextualization to predict a bounded correction to the anomaly score. The right panel illustrates evidence–label temporal mismatch patterns that motivate this design: early emergence, persistence around the label, and post-label persistence.

interaction. This assumption can be fragile when anomalous behavior unfolds over a local temporal region rather than manifesting only at the labeled interaction.

We identify this issue as *evidence–label temporal mismatch*. As illustrated in Figure 1, anomaly evidence may emerge before the labeled anomalous interaction, persist around it, or extend beyond it. Appendix B presents representative source-wise evidence trajectories illustrating these temporal patterns. This evidence–label temporal mismatch suggests that anomalous behavior in dynamic graphs is often better viewed as a temporally extended process rather than an isolated interaction. Direct point-level supervision can induce noisy or misleading training signals: evidence near anomaly onset may be penalized, early evidence before the label appears may be ignored, or late evidence may be reinforced after the abnormal behavior has faded. This perspective is consistent with observations in delayed-feedback learning and weakly supervised temporal anomaly detection, where supervision can be delayed, sparse, or temporally imprecise (Chapelle, 2014; Sultani et al., 2018).

We propose TECOS (**Te**mporally **Co**ntextualized **S**upervision), a two-stage framework based on *temporally contextualized residual supervision*. In Stage 1, a memory-based encoder produces complementary reconstruction and drift evidence, which are combined into a fixed base anomaly score (Goyal et al., 2018; Yu et al., 2018; Liu et al., 2021; Lee et al., 2024; Xu et al., 2020). In Stage 2, TECOS learns a bounded correction from the current event and recent interactions of the same source node, using both the evidence scores and representations of the interacting nodes. The base score provides an anomaly-specific reference, while the residual uses already observed temporal context to reduce sensitivity to temporally misaligned labels. Experiments on seven temporal interaction datasets show that TECOS achieves the best overall performance across AUC and AP among the evaluated baselines. We summarize our main contributions as follows:

Evidence–label temporal mismatch. We identify evidence–label mismatch in DGAD: anomaly evidence may precede, persist around, or outlast observed labels.

Temporally contextualized supervision. We propose TECOS, which refines reconstruction–drift evidence through short-range source-wise temporal context and applies supervision to a bounded residual correction of the base anomaly score.

Extensive experiments. Experiments on seven datasets show that TECOS achieves the best overall performance across AUC and AP among representative baselines.

2 Related Work and Preliminaries

2.1 Related Work

Dynamic Graph Anomaly Detection. Graph anomaly detection aims to identify abnormal nodes, edges, subgraphs, or graphs that deviate from the dominant patterns of graph-structured data (Akoglu et al., 2015; Ma et al., 2021; Qiao et al., 2025). On static graphs, anomalies are typically characterized by structural or attribute deviations on a fixed topology (Akoglu et al., 2010; Li et al., 2017; Ding et al., 2019; Zhou & Paffenroth, 2017), which cannot capture how abnormal behavior unfolds over time. Dynamic graph anomaly detection (DGAD) instead operates on evolving graph states, treating anomalies as unusual changes in interaction patterns rather than static outliers (Ekle & Eberle, 2024; Kazemi et al., 2020; Skarding et al., 2021). Within this broader setting, a line of work targets graph streams, edge streams, streaming heterogeneous graphs, and tensor streams, and detects anomalies through streaming statistics, microcluster-based summaries, or dense-substructure and subtensor mining (Aggarwal et al., 2011; Manzoor et al., 2016; Shin et al., 2017; Eswaran & Faloutsos, 2018). More recent studies build anomaly scores based on learned dynamic representations, via reconstruction and temporal-consistency modeling, self-supervised pretext tasks, or few-shot LLM-based reasoning (Goyal et al., 2018; Zheng et al., 2019; Cai et al., 2021; Yang et al., 2025; Liu et al., 2024). Despite their different techniques, these methods share a common focus: extracting anomaly evidence from evolving graphs. The complementary question of how this evidence should be temporally organized for supervision has received comparatively little attention.

Supervision under Temporal Label Misalignment. When anomaly labels are available, supervised or label-aided DGAD methods can directly anchor learned representations to anomaly semantics (Zheng et al., 2019; Tian et al., 2023). Temporal graph representation learners such as DyRep (Trivedi et al., 2019), JODIE (Kumar et al., 2019), and TGAT (Xu et al., 2020) can serve as dynamic encoders for downstream supervised classifiers. Implicit in these formulations is a temporal assumption that each observed label is aligned, at the individual-interaction level, with the graph instance to which supervision is applied. However, this assumption is fragile. In dynamic interaction graphs, suspicious behavior is rarely concentrated at a single timestamp: anomaly-related evidence may emerge before the annotated interaction, persist around it, or remain after the label disappears. Analogous misalignments have long been recognized in time-series and video anomaly detection, where they have motivated weakly supervised, segment-level, multiple-instance, and delay-aware objectives that decouple supervision from exact point-wise timing (Dietterich et al., 1997; Wang et al., 2017; Ilse et al., 2018; Chapelle, 2014; Sultani et al., 2018). Yet in DGAD, supervision is still typically imposed point by point, leaving this evidence–label temporal mismatch largely unaddressed. Our work targets this gap. We organize source-side evidence into temporal sequences and apply supervision after local temporal contextualization, thereby incorporating information from neighboring interactions.

2.2 Preliminaries

Problem setup. We consider a continuous-time dynamic graph represented as a chronologically ordered interaction stream $\mathcal{G} = \{e_i = (u_i, v_i, t_i)\}_{i=1}^{|\mathcal{E}|}$, where $u_i, v_i \in \mathcal{V}$ denote the source and destination nodes and $t_1 \leq \cdots \leq t_{|\mathcal{E}|}$. Each interaction has an observed source-side label $y_i \in \{0, 1\}$. Our task is to predict whether the source node u_i is anomalous at the time of interaction e_i. Frequently used notations are summarized in Appendix D.1.

Point-level anomaly evidence. A memory-based dynamic graph encoder maintains a current memory $\mathbf{m}_{u_i,i}$ and a temporal reference memory $\mathbf{m}'_{u_i,i}$ for the source node, and reconstructs an expected memory $\widehat{\mathbf{m}}_{u_i,i}$ from its interaction history (Yu et al., 2018; Liu et al., 2021; Lee et al., 2024). Reconstruction and drift evidence can be generally written as

$$c_i^{\text{rec}} = f_{\text{rec}}(\widehat{\mathbf{m}}_{u_i,i}, \mathbf{m}_{u_i,i}), \qquad c_i^{\text{drift}} = f_{\text{drift}}(\mathbf{m}'_{u_i,i}, \mathbf{m}_{u_i,i}). \tag{1}$$

These two event-level scores form the point-wise evidence used by TeCoS. The detailed memory encoder is provided in Appendix D.2.

3 THEORETICAL MOTIVATION

We analyze why point-wise supervision can be unreliable under temporal evidence–label mismatch and when local temporal context can help. Let $\mathbf{z}_t = [c_t^{\mathrm{rec}}, c_t^{\mathrm{drift}}]$ denote the reconstruction–drift evidence at source-side interaction position t, and let $A_t \in \{0, 1\}$ denote the corresponding latent anomaly state. We consider a representative delayed-label model $\widetilde{A}_t = A_{t-\Delta}$, where $\Delta > 0$.

Boundary fragility of point-wise supervision. Define the mismatched positions as $\mathcal{B}_\Delta = \{t : A_t \neq \widetilde{A}_t\}$. For BCEWithLogits, the gradient with respect to a logit u is $\partial_u \ell_{\mathrm{BCE}}(u, y) = \sigma(u) - y$.

Proposition 3.1 (Boundary fragility). *For every $t \in \mathcal{B}_\Delta$ and any finite logit u_t,*

$$\big(\sigma(u_t) - \widetilde{A}_t\big)\big(\sigma(u_t) - A_t\big) < 0.$$

Thus, at mismatched anomaly boundaries, the observed label induces a gradient with the opposite sign to that induced by the latent state. Point-wise supervision may therefore suppress emerging anomaly evidence or reinforce evidence after the underlying anomaly has ended.

Risk advantage of temporal context. Let

$$\mathbf{C}_t^{(k)} = (\mathbf{z}_{t-k}, \ldots, \mathbf{z}_{t-1}, \mathbf{z}_t) = (\mathbf{z}_t, \mathbf{N}_t^{(k)})$$

denote a historical context of length $k + 1$, where $\mathbf{N}_t^{(k)}$ contains the preceding evidence. For an input $\mathbf{V}$, define the Bayes-optimal BCE risk as

$$R^*(\mathbf{V}) = \inf_f \mathbb{E}[\ell_{\mathrm{BCE}}(f(\mathbf{V}), \widetilde{A}_t)].$$

Proposition 3.2 (Risk advantage of temporal context). *Under the Bayes-optimal log-loss predictor,*

$$R^*(\mathbf{z}_t) - R^*(\mathbf{C}_t^{(k)}) = I(\widetilde{A}_t; \mathbf{N}_t^{(k)} \mid \mathbf{z}_t) \geq 0.$$

The inequality is strict whenever the preceding evidence provides additional information about the observed label. In particular, if $k \geq \Delta$ and $I(\widetilde{A}_t; \mathbf{z}_{t-\Delta} \mid \mathbf{z}_t) > 0$, then $R^*(\mathbf{C}_t^{(k)}) < R^*(\mathbf{z}_t)$.

Because the context contains the current evidence, a contextual predictor can recover a point-wise rule by ignoring preceding inputs. It can achieve a strictly lower optimal risk when the context additionally contains informative delay-aligned evidence.

A linear illustration of lag recovery. Consider standardized, temporally uncorrelated scalar evidence x_t and a surrogate target $r_t = \beta x_{t-\Delta} + \varepsilon_t$, where ε_t is uncorrelated with the evidence window. For $k \geq \Delta$, the population squared-loss predictor $\widehat{r}_t = \sum_{j=-k}^{0} w_j x_{t+j}$ has the unique solution

$$w_j^* = \begin{cases} \beta, & j = -\Delta, \\ 0, & j \neq -\Delta. \end{cases}$$

Thus, when its receptive field covers the lag, a learnable temporal filter can assign its weight to the delay-aligned evidence position. Full proofs and derivations are provided in Appendix E.

4 METHODOLOGY

The analysis in Section 3 shows that point-wise supervision can be sensitive to temporal misalignment between observed labels and the underlying anomaly evidence. When labels are delayed or temporally imprecise, supervising isolated event scores may provide gradients that are inconsistent with the local evolution of anomaly evidence (Propositions 3.1 and 3.2).

Guided by this observation, TECOS combines event-level anomaly evidence with local temporal context to refine anomaly scoring. The framework consists of two stages. Stage 1 extracts reconstruction–drift evidence and endpoint representations, and produces an initial anomaly score for each event. Stage 2 contextualizes these event-level features with a lightweight Conv1D network and learns an additive correction to the Stage-1 score. Figure 1 provides an overview.

4.1 Evidence and Endpoint Representations

Stage 1 characterizes each interaction using reconstruction–drift evidence and the underlying memory representations. We instantiate the reconstruction and drift evidence using an MI-based contrastive operator (Bachman et al., 2019; Poole et al., 2019; Belghazi et al., 2018). Both types of evidence are relational: reconstruction measures the consistency between a reconstructed memory and the current memory, whereas drift measures the consistency between consecutive memory states.

For a source node u_i, let $\mathbf{m}_{u_i,i}$, $\mathbf{m}'_{u_i,i}$, and $\widehat{\mathbf{m}}_{u_i,i}$ denote its current memory, temporal reference memory, and neighborhood-based reconstructed memory, respectively. Given a matched pair $(\mathbf{c}, \mathbf{m})$ and a set of mismatched negative memories $\mathcal{N}_i^-$, we use cosine similarity

$$f_{\cos}(\mathbf{c}, \mathbf{m}) = \frac{\mathbf{c}^\top \mathbf{m}}{\|\mathbf{c}\|_2 \|\mathbf{m}\|_2} \tag{2}$$

to define the contrastive dependence score

$$\hat{I}\left(\mathbf{c}, \mathbf{m}; \mathcal{N}_i^-\right) = f_{\cos}(\mathbf{c}, \mathbf{m}) - \log \frac{1}{|\mathcal{N}_i^-|} \sum_{\mathbf{n} \in \mathcal{N}_i^-} \exp\left(f_{\cos}(\mathbf{c}, \mathbf{n})\right). \tag{3}$$

A larger value indicates stronger compatibility of the matched memory pair relative to mismatched memories.

The source-side reconstruction and drift evidence are then

$$\begin{aligned} c_i^{\text{rec}} &= \hat{I}(\widehat{\mathbf{m}}_{u_i,i}, \mathbf{m}_{u_i,i}; \mathcal{N}_i^-), \\ c_i^{\text{drift}} &= \hat{I}(\mathbf{m}_{u_i,i}, \mathbf{m}'_{u_i,i}; \mathcal{N}_i^-). \end{aligned} \tag{4}$$

The reconstruction score captures the consistency between reconstructed and current memories, while the drift score captures temporal consistency across memory states. Appendix A.5 further examines the complementary contributions of the reconstruction and drift evidence.

For each interaction $e_i = (u_i, v_i, t_i)$, Stage 2 receives the two scalar evidence scores together with the source and destination endpoint representations:

$$\mathbf{x}_i = \left[c_i^{\text{rec}}, c_i^{\text{drift}}, \mathbf{m}_{u_i,i}^\top, \widehat{\mathbf{m}}_{u_i,i}^\top, \mathbf{m}_{v_i,i}^\top, \widehat{\mathbf{m}}_{v_i,i}^\top\right]^\top. \tag{5}$$

The scalar scores summarize reconstruction consistency and temporal drift, while the endpoint representations retain complementary information about the interacting nodes. Stage 1 also produces an initial anomaly score b_i from the reconstruction–drift evidence, which forms the direct scoring branch in Stage 2. Further details of the memory encoder, evidence objective, and base-score computation are provided in Appendix A.

4.2 Temporal Contextualization and Score Correction

Temporal misalignment is typically local: anomaly-related evidence may emerge or persist across interactions near the observed label. Stage 2 therefore uses recent event features to contextualize the current interaction before correcting its anomaly score.

Temporal context. For each event i, let $\pi_0(i) = i$ and let $\pi_\ell(i)$ denote the ℓ-th most recent event preceding i with source node u_i, for $\ell \in \{1, 2\}$. We construct

$$\mathbf{X}_i = \left[\mathbf{x}_{\pi_2(i)}, \mathbf{x}_{\pi_1(i)}, \mathbf{x}_i\right], \tag{6}$$

where each event feature contains both reconstruction–drift evidence and endpoint representations. This allows the temporal module to capture local changes in both anomaly evidence and the graph representations from which that evidence is obtained.

Each feature dimension is standardized using training-set statistics. A shared projection maps the event features into a common hidden space:

$$\mathbf{h}_{i,\ell} = \mathbf{W}_p \overline{\mathbf{x}}_{\pi_\ell(i)} + \mathbf{b}_p, \qquad \ell \in \{0, 1, 2\}, \tag{7}$$

where $\overline{\mathbf{x}}$ denotes the standardized event feature. Missing historical positions are masked and contribute zero features to the temporal module.

Conv1D correction. The three projected event features are arranged chronologically and processed jointly by a one-dimensional convolution with kernel size three:

$$\mathbf{g}_i = \mathrm{ReLU}\left(\mathbf{b}_c + \sum_{\ell=0}^{2} \mathbf{W}_\ell \mathbf{h}_{i,\ell}\right). \tag{8}$$

The position-specific convolution weights distinguish the current interaction from its recent context, while the same parameters are shared across event windows. A lightweight correction head maps the contextual representation to a scalar:

$$r_i = \mathbf{w}_o^\top \mathrm{Dropout}(\mathbf{g}_i) + b_o. \tag{9}$$

The resulting temporal correction is

$$\Delta_i = \gamma \tanh(r_i), \tag{10}$$

where $\gamma \geq 0$ controls its magnitude. The final anomaly score combines the Stage-1 score and the temporal correction:

$$s_i = \alpha b_i + \Delta_i = \alpha b_i + \gamma \tanh(r_i), \tag{11}$$

where $\alpha \geq 0$ controls the contribution of the Stage-1 score. The Stage-1 branch captures event-level anomaly evidence, while the Conv1D correction incorporates endpoint information and its local temporal evolution.

We initialize the correction output layer with zero weights and bias, giving $\Delta_i = 0$ at initialization. The correction network is trained with weighted binary cross entropy using the observed event labels. At inference, s_i is used as the final anomaly score. Additional implementation and training details are provided in Appendix A.

5 Experiments

We conduct comprehensive experiments to answer the following research questions: (1) How does TECoS compare with representative methods for dynamic graph anomaly detection? (2) How does temporal contextualization contribute to Stage-2 score correction? (3) Is temporal score correction robust to controlled evidence–label misalignment? (4) How sensitive is TECoS to the amount of temporal context? (5) How do key Stage-2 components contribute to performance?

5.1 Experimental Settings

Datasets. We evaluate TECoS on seven temporal interaction datasets: Wikipedia, Reddit, and MOOC (Kumar et al., 2019); BitcoinAlpha and BitcoinOTC (Kumar et al., 2016); EU-Core (Paranjape et al., 2017); and Email-DNC (Rossi & Ahmed, 2015). We use the same event records, anomaly labels, and chronological data partitions for all methods. Each dataset is split chronologically into 70%/15%/15% for training, validation, and testing, with equal timestamps kept in the same partition. Dataset statistics, preprocessing, and anomaly-label construction are provided in Appendix C.1.

Baselines. We compare TECoS with six representative baselines: SAD (Tian et al., 2023), SLADE (Lee et al., 2024), TGAT (Xu et al., 2020), TGN (Rossi et al., 2020), JODIE (Kumar et al., 2019), and DyRep (Trivedi et al., 2019). These methods cover supervised, self-supervised, and pretrain-then-supervise anomaly detection settings. SLADE and TGAT use reconstruction- or drift-based anomaly scoring without an additional supervised anomaly classifier, whereas SAD and the evaluated TGN/JODIE/DyRep pipelines use anomaly labels during training. Implementation and configuration details are provided in Appendix A.4.

Training and evaluation. The main TECoS model uses two preceding same-source events for temporal contextualization across all datasets. Checkpoints are selected by validation AP, with validation AUC and then the earlier epoch used to break ties. Each experiment is repeated with three random seeds. We report AUC and AP as the primary ranking metrics, using the mean and population standard deviation across seeds. Detailed training settings and model-selection procedures are provided in Appendix A.

Table 1: Main event-level anomaly detection results. AUC and AP are reported as mean±std (%). Higher is better, and the best mean result for each dataset is shown in bold. The superscript sup denotes variants of temporal graph representation models equipped with a supervised anomaly classifier trained on frozen pretrained representations.

Method	Wikipedia	Reddit	BitcoinAlpha	BitcoinOTC	MOOC	EU-Core	Email-DNC
AUC							
SAD	86.03 ± 0.51	68.85 ± 0.70	69.52 ± 3.59	67.52 ± 5.24	63.32 ± 0.65	87.96 ± 0.14	94.20 ± 0.05
SLADE	84.90 ± 0.49	69.91 ± 0.36	76.54 ± 0.57	75.32 ± 0.03	59.87 ± 0.46	80.19 ± 0.26	90.55 ± 0.40
TGAT	83.85 ± 0.89	$\mathbf{73.53 \pm 0.47}$	75.07 ± 0.40	76.21 ± 0.32	63.75 ± 0.26	82.13 ± 0.22	91.43 ± 0.35
TGN$^{\text{sup}}$	84.76 ± 0.46	61.46 ± 0.59	59.11 ± 0.87	66.58 ± 4.84	66.65 ± 1.79	52.56 ± 1.91	50.33 ± 2.21
JODIE$^{\text{sup}}$	86.99 ± 0.49	60.84 ± 4.19	72.35 ± 0.82	66.73 ± 0.56	58.30 ± 2.30	68.61 ± 13.54	56.48 ± 6.09
DyRep$^{\text{sup}}$	85.80 ± 0.74	60.75 ± 2.82	68.38 ± 3.70	66.23 ± 1.41	61.14 ± 1.24	50.27 ± 0.52	50.65 ± 1.62
TECOS	$\mathbf{87.66 \pm 0.93}$	73.22 ± 1.36	$\mathbf{78.24 \pm 0.03}$	$\mathbf{76.72 \pm 0.34}$	$\mathbf{70.69 \pm 0.55}$	$\mathbf{92.27 \pm 0.22}$	$\mathbf{98.31 \pm 0.04}$
AP							
SAD	$\mathbf{1.76 \pm 0.03}$	0.24 ± 0.01	12.27 ± 1.80	17.32 ± 1.92	2.57 ± 0.10	42.57 ± 0.56	50.18 ± 0.46
SLADE	1.24 ± 0.01	0.22 ± 0.01	15.02 ± 0.07	19.70 ± 0.01	1.64 ± 0.02	28.18 ± 0.78	43.50 ± 0.76
TGAT	1.56 ± 0.01	0.24 ± 0.02	14.97 ± 0.07	20.26 ± 0.13	2.00 ± 0.17	43.78 ± 1.45	45.19 ± 0.55
TGN$^{\text{sup}}$	1.45 ± 0.09	0.17 ± 0.01	8.43 ± 0.21	16.22 ± 1.62	2.14 ± 0.21	10.53 ± 0.38	10.00 ± 0.78
JODIE$^{\text{sup}}$	1.59 ± 0.02	0.19 ± 0.04	13.57 ± 0.79	15.06 ± 0.68	1.55 ± 0.09	26.15 ± 11.32	12.63 ± 1.64
DyRep$^{\text{sup}}$	1.44 ± 0.10	0.16 ± 0.02	11.18 ± 1.22	14.98 ± 0.74	1.98 ± 0.09	10.10 ± 0.15	10.94 ± 0.54
TECOS	1.70 ± 0.21	$\mathbf{0.31 \pm 0.08}$	$\mathbf{15.69 \pm 0.24}$	$\mathbf{22.34 \pm 0.83}$	$\mathbf{2.79 \pm 0.01}$	$\mathbf{53.08 \pm 3.97}$	$\mathbf{87.13 \pm 0.21}$

Table 2: Effect of temporal contextualization under two fixed history lengths. AP is reported as mean±std (%). Higher is better.

Variant	Wikipedia	Reddit	BitcoinAlpha	BitcoinOTC	MOOC	EU-Core	Email-DNC
Current-only	1.45 ± 0.32	0.35 ± 0.10	14.90 ± 0.43	21.65 ± 0.17	2.69 ± 0.05	46.66 ± 3.91	84.21 ± 1.17
$H = 2$							
Mean pooling	$\mathbf{1.82 \pm 0.35}$	$\mathbf{0.40 \pm 0.01}$	14.57 ± 0.71	22.00 ± 0.76	$\mathbf{2.84 \pm 0.15}$	50.81 ± 4.10	71.55 ± 0.43
Conv1D	1.70 ± 0.21	0.31 ± 0.08	$\mathbf{15.69 \pm 0.24}$	$\mathbf{22.34 \pm 0.83}$	2.79 ± 0.01	$\mathbf{53.08 \pm 3.97}$	$\mathbf{87.13 \pm 0.21}$
$H = 3$							
Mean pooling	$\mathbf{2.32 \pm 0.54}$	0.38 ± 0.04	14.32 ± 0.66	21.14 ± 0.64	2.94 ± 0.07	49.27 ± 3.34	70.79 ± 0.52
Conv1D	2.05 ± 0.37	$\mathbf{0.39 \pm 0.05}$	$\mathbf{15.41 \pm 0.39}$	$\mathbf{21.67 \pm 0.34}$	$\mathbf{3.06 \pm 0.03}$	$\mathbf{52.37 \pm 3.26}$	$\mathbf{87.24 \pm 0.36}$

5.2 Overall Anomaly Detection Performance

Table 1 reports the main event-level anomaly detection results. TECOS achieves the highest mean AUC and AP on six of the seven datasets, showing consistent improvements across diverse temporal interaction graphs. The gains are particularly pronounced on EU-Core and Email-DNC, where AP improves from the strongest baseline results of 43.78% and 50.18% to 53.08% and 87.13%, respectively. On BitcoinAlpha and BitcoinOTC, TECOS also improves both AUC and AP over the strongest baseline results.

The remaining differences are comparatively small. TGAT achieves a slightly higher AUC on Reddit (73.53% vs. 73.22%), while SAD achieves a slightly higher AP on Wikipedia (1.76% vs. 1.70%). Overall, these results show that temporal score correction consistently improves event-level anomaly ranking across diverse dynamic graphs. We further compare TECOS with representative temporal graph models, TGN and DyGFormer, in Appendix G.1.

5.3 Effect of Temporal Contextualization

We next examine how temporal context and its aggregation contribute to Stage-2 score correction. We keep the Stage-1 features, score composition, and training protocol unchanged, and compare three variants. *Current-only* uses only the current event feature and removes historical context from the correction branch. *Mean pooling* aggregates the current and historical event features without modeling their temporal positions. *Conv1D* learns position-sensitive aggregation over the same window. We evaluate the two temporal variants with history lengths $H = 2$ and $H = 3$.

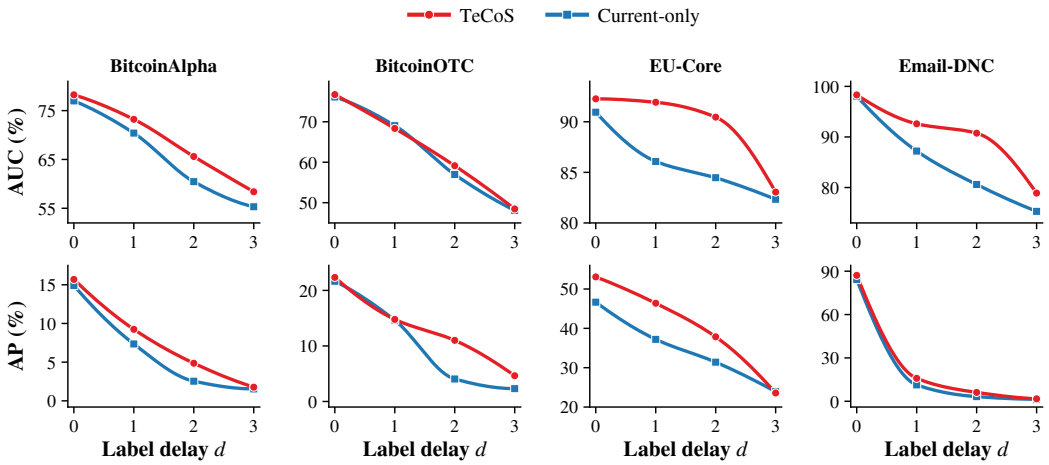


Figure 2: Performance under controlled label delays. The top and bottom rows report AUC and AP, respectively. Markers show three-seed means; shape-preserving curves serve only as visual guides.

Table 2 reports AP for the controlled comparison. With $H = 2$, Conv1D improves over Current-only on six of the seven datasets, with particularly clear gains on EU-Core and Email-DNC. With $H = 3$, Conv1D improves over Current-only on all seven datasets. These results show that recent interactions provide useful information beyond the evidence and representations of the current event alone.

The Mean pooling comparison examines how temporal context is aggregated. Conv1D achieves higher AP on four datasets at $H = 2$ and six datasets at $H = 3$. The gains are pronounced on EU-Core and Email-DNC, indicating that preserving temporal position can be important when anomaly-related patterns evolve across nearby interactions. Mean pooling remains competitive on Wikipedia and several shorter-context settings, showing that the benefit of position-sensitive aggregation varies across datasets. Appendix G.2 compares TeCoS with generic window aggregation strategies.

5.4 Label-Delay Analysis

We evaluate controlled label delays on BitcoinAlpha, BitcoinOTC, EU-Core, and Email-DNC. For $d \in \{0, 1, 2, 3\}$, positive labels are shifted forward by d same-source interactions, independently within each training, validation, and test partition. Out-of-range labels are discarded, while all interaction rows are retained. We compare TeCoS with *Current-only*, which masks the two historical positions after projection. Both predictors share the frozen encoder features, Stage-2 architecture, and training settings. For each nonzero delay, Stage 2 is retrained on transformed training labels and selected by transformed validation AP.

Figure 2 shows that TeCoS outperforms Current-only in mean AUC and AP in 11 of the 12 nonzero-delay conditions for each metric. The exceptions are BitcoinOTC AUC at $d = 1$ and EU-Core AP at $d = 3$. These results support the usefulness of neighboring event features for predicting delayed labels. Because label shifts alter anomaly prevalence, we emphasize comparisons within each delay condition. Appendix F provides detailed protocols, standard deviations, label counts, and an additional comparison with mean pooling.

5.5 Sensitivity to History Length

We examine how temporal context length affects score correction by varying the number of preceding events $H \in \{1, 2, 3, 4, 5\}$. The current event is included in every window, and the Conv1D kernel spans the complete temporal context. All other model and training settings remain unchanged.

Table 3 shows that the effect of history length is dataset dependent and generally non-monotonic. The two Bitcoin datasets achieve their highest AP with $H = 2$, while Wikipedia and MOOC favor $H = 3$. Shorter context performs best on Reddit and EU-Core, whereas Email-DNC continues to benefit

Table 3: Sensitivity to history length H. AP is reported as mean±std (%). Higher is better, and the best mean result for each dataset is shown in bold.

Dataset	$H = 1$	$H = 2$	$H = 3$	$H = 4$	$H = 5$
Wikipedia	1.52 ± 0.11	1.70 ± 0.21	$\mathbf{2.05 \pm 0.37}$	1.77 ± 0.38	1.91 ± 0.08
Reddit	$\mathbf{0.45 \pm 0.17}$	0.31 ± 0.08	0.39 ± 0.05	0.32 ± 0.05	0.33 ± 0.06
BitcoinAlpha	15.17 ± 0.31	$\mathbf{15.69 \pm 0.24}$	15.41 ± 0.39	15.07 ± 0.24	14.97 ± 0.24
BitcoinOTC	20.76 ± 0.74	$\mathbf{22.34 \pm 0.83}$	21.67 ± 0.34	21.63 ± 0.81	21.43 ± 0.62
MOOC	2.80 ± 0.05	2.79 ± 0.01	$\mathbf{3.06 \pm 0.03}$	3.02 ± 0.03	3.00 ± 0.04
EU-Core	$\mathbf{53.88 \pm 3.67}$	53.08 ± 3.97	52.37 ± 3.26	52.84 ± 4.15	53.16 ± 3.31
Email-DNC	85.97 ± 1.10	87.13 ± 0.21	87.24 ± 0.36	88.30 ± 0.33	$\mathbf{89.31 \pm 0.19}$

Table 4: Ablation of endpoint representations and the direct base-score branch. AP is reported as mean±std (%) over three seeds.

Dataset	Base-only	Evidence-only	Without base	Full
Wikipedia	1.20 ± 0.00	1.41 ± 0.16	0.63 ± 0.04	$\mathbf{1.70 \pm 0.21}$
Reddit	0.25 ± 0.03	0.26 ± 0.04	0.15 ± 0.05	$\mathbf{0.31 \pm 0.08}$
BitcoinAlpha	14.84 ± 0.07	15.28 ± 0.33	11.77 ± 0.17	$\mathbf{15.69 \pm 0.24}$
BitcoinOTC	19.99 ± 0.05	19.99 ± 0.05	20.57 ± 0.53	$\mathbf{22.34 \pm 0.83}$
MOOC	1.94 ± 0.05	2.38 ± 0.04	2.72 ± 0.05	$\mathbf{2.79 \pm 0.01}$
EU-Core	15.34 ± 1.30	22.34 ± 2.38	$\mathbf{60.07 \pm 3.46}$	53.08 ± 3.97
Email-DNC	45.86 ± 0.70	48.47 ± 0.68	85.01 ± 0.58	$\mathbf{87.13 \pm 0.21}$

from longer history and reaches its highest AP at $H = 5$. These results indicate that anomaly-related temporal patterns occur at different local scales across datasets, and that increasing the amount of history does not uniformly improve detection.

5.6 Ablation Study

We ablate two components of Stage 2: endpoint representations and the base-score branch. *Base-only* removes the learned correction; *Evidence-only* excludes endpoint representations from the correction input; *Without base* removes the direct base-score branch; and *Full* denotes the complete model. All learned variants share frozen Stage-1 features, temporal context, architecture, and training protocol.

As shown in Table 4, Full outperforms Base-only and Evidence-only on all seven datasets, showing that both learned correction and endpoint representations contribute beyond the base anomaly score and scalar evidence. Removing the base-score branch reduces AP on six datasets but improves EU-Core, indicating that the base score is generally useful while its contribution is dataset dependent.

6 Conclusion

We presented TeCoS, a framework that combines event-level anomaly evidence with local temporal context through lightweight score correction. Experiments on seven datasets demonstrate that recent interactions provide useful information beyond the current event, improving anomaly ranking across diverse dynamic graphs. Controlled label-delay experiments further show that temporal context often improves prediction under delayed supervision. We will further investigate adaptive temporal context selection to accommodate the varying benefits across datasets and history lengths. We also plan to extend the framework to more diverse annotation-delay mechanisms.

AI USE STATEMENT

We used generative AI tools solely for language polishing and improving the clarity of the manuscript. All AI-assisted edits were reviewed by the authors.

ETHICS STATEMENT

This study uses public datasets without new participant recruitment or private-data collection. Anomaly labels do not establish individual misconduct. Practical use should account for privacy, bias, and false positives, with appropriate human oversight.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the model architecture, training procedure, and dataset construction and preprocessing in the main paper and appendix. We also document chronological data splits, hyperparameter settings, checkpoint selection, and evaluation protocols, and report results over three random seeds. The project repository provides implementation and instructions for running the training and evaluation pipeline: `https://anonymous.4open.science/r/TECOS-37BB/`.

REFERENCES

Charu C Aggarwal, Yuchen Zhao, and S Yu Philip. Outlier detection in graph streams. In *2011 IEEE 27th international conference on data engineering*, pp. 399–409. IEEE, 2011.

Leman Akoglu, Mary McGlohon, and Christos Faloutsos. Oddball: Spotting anomalies in weighted graphs. In *Pacific-Asia conference on knowledge discovery and data mining*, pp. 410–421. Springer, 2010.

Leman Akoglu, Hanghang Tong, and Danai Koutra. Graph based anomaly detection and description: a survey. *Data mining and knowledge discovery*, 29(3):626–688, 2015.

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, 32, 2019.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International conference on machine learning*, pp. 531–540. PMLR, 2018.

Lei Cai, Zhengzhang Chen, Chen Luo, Jiaping Gui, Jingchao Ni, Ding Li, and Haifeng Chen. Structural temporal graph neural networks for anomaly detection in dynamic graphs. In *Proceedings of the 30th ACM international conference on Information & Knowledge Management*, pp. 3747–3756, 2021.

Olivier Chapelle. Modeling delayed feedback in display advertising. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1097–1105, 2014.

Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1-2):31–71, 1997.

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*, pp. 594–602. SIAM, 2019.

Ocheme Anthony Ekle and William Eberle. Anomaly detection in dynamic graphs: A comprehensive survey. *ACM Transactions on Knowledge Discovery from Data*, 18(8):1–44, 2024.

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

Palash Goyal, Nitin Kamra, Xinran He, and Yan Liu. Dyngem: Deep embedding method for dynamic graphs. *arXiv preprint arXiv:1805.11273*, 2018.

Fengrui Hua, Yiyan Qi, Zikai Wei, Yuxing Tian, Chengjin Xu, Xiaojun Wu, Jia Li, and Jian Guo. Bag: Benchmarking anomaly detection on dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 14892–14900, 2026. doi: 10.1609/aaai.v40i17.38510.

Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pp. 2127–2136. PMLR, 2018.

Seyed Mehran Kazemi, Rishab Goel, Kshitij Jain, Ivan Kobyzev, Akshay Sethi, Peter Forsyth, and Pascal Poupart. Representation learning for dynamic graphs: A survey. *Journal of Machine Learning Research*, 21(70):1–73, 2020.

Srijan Kumar, Francesca Spezzano, VS Subrahmanian, and Christos Faloutsos. Edge weight prediction in weighted signed networks. In *2016 IEEE 16th international conference on data mining (ICDM)*, pp. 221–230. IEEE, 2016.

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

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

Jundong Li, Harsh Dani, Xia Hu, and Huan Liu. Radar: Residual analysis for anomaly detection in attributed networks. In *Ijcai*, volume 17, pp. 2152–2158, 2017.

Shuo Liu, Di Yao, Lanting Fang, Zhetao Li, Wenbin Li, Kaiyu Feng, Xiaowen Ji, and Jingping Bi. Anomalyllm: Few-shot anomaly edge detection for dynamic graphs using large language models. In *2024 IEEE International Conference on Data Mining (ICDM)*, pp. 785–790. IEEE, 2024.

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

Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z Sheng, Hui Xiong, and Leman Akoglu. A comprehensive survey on graph anomaly detection with deep learning. *IEEE transactions on knowledge and data engineering*, 35(12):12012–12038, 2021.

Emaad Manzoor, Sadegh M Milajerdi, and Leman Akoglu. Fast memory-efficient anomaly detection in streaming heterogeneous graphs. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1035–1044, 2016.

Ashwin Paranjape, Austin R. Benson, and Jure Leskovec. Motifs in temporal networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pp. 601–610, 2017. doi: 10.1145/3018661.3018731.

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International conference on machine learning*, pp. 5171–5180. PMLR, 2019.

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

Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs. *arXiv preprint arXiv:2006.10637*, 2020.

Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015. doi: 10.1609/aaai.v29i1.9277.

Kijung Shin, Bryan Hooi, Jisu Kim, and Christos Faloutsos. Densealert: Incremental dense-subtensor detection in tensor streams. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1057–1066, 2017.

Joakim Skarding, Bogdan Gabrys, and Katarzyna Musial. Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey. *iEEE Access*, 9:79143–79168, 2021.

Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6479–6488, 2018.

Sheng Tian, Jihai Dong, Jintang Li, Wenlong Zhao, Xiaolong Xu, Bowen Song, Changhua Meng, Tianyi Zhang, Liang Chen, et al. Sad: Semi-supervised anomaly detection on dynamic graphs. *arXiv preprint arXiv:2305.13573*, 2023.

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

Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 4325–4334, 2017.

Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. *arXiv preprint arXiv:2002.07962*, 2020.

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

Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. Towards better dynamic graph learning: New architecture and unified library. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Wenchao Yu, Wei Cheng, Charu C Aggarwal, Kai Zhang, Haifeng Chen, and Wei Wang. Netwalk: A flexible deep embedding approach for anomaly detection in dynamic networks. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2672–2681, 2018.

Li Zheng, Zhenpeng Li, Jian Li, Zhao Li, and Jun Gao. Addgraph: Anomaly detection in dynamic graph using attention-based temporal gcn. In *IJCAI*, volume 3, pp. 7, 2019.

Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 665–674, 2017.

Table 5: Event counts with anomaly counts in parentheses for the shared chronological partitions.

Dataset	Train	Validation	Test
Wikipedia	110,232 (156)	23,621 (17)	23,621 (44)
Reddit	470,713 (190)	100,867 (82)	100,867 (94)
BitcoinAlpha	16,940 (521)	3,628 (123)	3,618 (230)
BitcoinOTC	24,914 (1,123)	5,339 (975)	5,339 (470)
MOOC	288,224 (2,977)	61,762 (510)	61,763 (579)
EU-Core	232,634 (23,240)	49,850 (5,008)	49,850 (4,985)
Email-DNC	27,492 (2,744)	5,882 (593)	5,890 (589)

Table 6: Stage-1 training settings. Batches/update is the number of encoder batches in a gradient segment, not a Stage-2 prediction unit.

Dataset	Batch size	Batches/update	Source rec. weight	Destination rec. weight
Wikipedia	100	1	0.2	0.2
Reddit	100	1	0.1	1.0
BitcoinAlpha	100	1	0.2	0.2
BitcoinOTC	100	1	0.2	0.2
MOOC	300	8	10.0	10.0
EU-Core	100	1	0.08	0.08
Email-DNC	100	1	0.08	0.08

A Experimental Details

A.1 Encoder and Correction-Network Settings

Stage-1 encoder. Stage 1 uses a memory encoder with 256-dimensional memory, a 128-dimensional message representation, an MLP message function, a GRU memory updater, two temporal-attention heads, and 20 sampled historical neighbors. It is trained for ten epochs using Adam, with an initial learning rate of 3×10^{-6}, per-epoch decay of 0.8, weight decay of 10^{-4}, dropout of 0.1, and gradient-norm clipping at 1. Negative memories are sampled from training nodes. The source and destination reconstruction coefficients and batch settings are given in Table 6. Reconstruction coefficients do not weight the drift terms. A chronological replay with frozen parameters produces the stored evidence and memory features.

Base anomaly score. The reconstruction and drift scores measure consistency, with larger values indicating stronger compatibility between memory states. We combine their unstandardized values into the base anomaly score

$$b_i = 1 - \mathrm{clip}_{[0,1]} \left(\frac{c_i^{\mathrm{rec}} + c_i^{\mathrm{drift}}}{M_{\mathrm{rec}} + M_{\mathrm{drift}}} \right), \tag{12}$$

where

$$M_q = \max \left\{ Q_{0.95} \left(\{ c_j^q : j \in \mathcal{D}_{\mathrm{tr}} \} \right), 10^{-8} \right\}, \qquad q \in \{\mathrm{rec}, \mathrm{drift}\}. \tag{13}$$

Here, $Q_{0.95}$ denotes the 95th percentile over training events. The scales remain fixed during validation and testing. The resulting score satisfies $b_i \in [0, 1]$, with larger values indicating greater abnormality, and supplies the direct scoring branch in Stage 2.

Stage-2 correction network. A shared $1026 \to 64$ projection precedes a single $64 \to 64$ convolution with kernel size three. ReLU and dropout of 0.1 are followed by a $64 \to 1$ output layer whose weights and bias are initialized to zero. Adam uses a batch size of 1024, weight decay of 10^{-4}, and gradient-norm clipping at 1. Table 7 supplies the remaining training settings. Standardization means and scales are fitted on training events; dimensions with standard deviation below 10^{-6} use scale one. Padded positions are set to zero after projection.

Table 7: Fixed main-model configurations. C is the positive-class weight cap. Every row uses the full input, frozen Stage-1 parameters, two past events, and weighted BCE without a ranking term.

Dataset	α	γ	Learning rate	Epochs	Decay	C
Wikipedia	1.0	0.05	0.0003	10	0.8	700
Reddit	1.0	0.1	0.0005	10	0.8	300
BitcoinAlpha	1.0	0.1	0.0005	10	0.8	300
BitcoinOTC	2.0	0.1	0.0005	10	0.8	300
MOOC	0.25	1.0	0.00025	30	0.95	10
EU-Core	1.0	1.0	0.0005	100	0.98	1
Email-DNC	1.0	1.0	0.0001	30	0.95	300

Table 8: Computational efficiency of TECOS and SAD. Training time is measured over ten epochs, latency is reported per event, and memory denotes peak GPU memory. Lower values are better; the lower value in each comparison is marked in bold.

Dataset	Method	Time		Peak memory (MiB)	
		Train (s)	Latency (ms/event)	Train	Inference
Wikipedia	TECOS	**471.16**	**0.0892**	1,937.2	**292.1**
	SAD	592.27	0.1635	**679.7**	317.1
Reddit	TECOS	**2,361.06**	**0.0885**	2,712.4	743.0
	SAD	2,712.58	0.1703	**870.2**	**389.1**
BitcoinAlpha	TECOS	**67.03**	**0.0992**	**897.5**	**115.3**
	SAD	93.44	0.3950	1,185.4	513.0
BitcoinOTC	TECOS	**96.09**	**0.1006**	**1,238.6**	**139.5**
	SAD	133.26	0.3170	1,610.9	502.2

A.2 MACHINES

All experiments were conducted on a Linux machine with NVIDIA GeForce RTX 5090 GPUs (32 GB VRAM each), using CUDA 13.0.

A.3 COMPUTATIONAL EFFICIENCY

We compare the computational efficiency of TECOS and SAD (Tian et al., 2023) on an NVIDIA GeForce RTX 5090 with a batch size of 100. Training time covers ten epochs and excludes data loading, graph construction, and model initialization. Inference latency covers the complete prediction pipeline, including chronological evidence extraction and temporal prediction.

Table 8 shows that TECOS requires less training time and achieves lower inference latency than SAD on all four datasets. TECOS also uses less peak inference memory on three datasets, while its training-memory usage is dataset-dependent. Across all settings, the peak training-memory consumption of TECOS remains below 2.8 GiB.

A.4 DETAILS OF BASELINES

All baselines use the same event rows, anomaly labels, and chronological 70/15/15 train/validation/test assignments as TECOS, with equal timestamps kept in the same partition. Model parameters and fitted preprocessing statistics use the training partition. Checkpoints are selected by validation AP, with validation AUC and then the earlier epoch breaking ties. Each method produces one source-side anomaly score for every test interaction, without filtering events by history availability. We report AUC and AP as mean$\pm$population standard deviation over seeds 0, 1, and 2.

For TGN (Rossi et al., 2020), JODIE (Kumar et al., 2019), and DyRep (Trivedi et al., 2019), we pretrain the temporal encoder on training interactions using a link-prediction objective, then freeze its parameters and train an anomaly-classification head using training anomaly labels. The classifier's sigmoid output serves as the event-level anomaly score. We denote these supervised adaptations by TGN^{sup}, $\text{JODIE}^{\text{sup}}$, and $\text{DyRep}^{\text{sup}}$ in the main results table.

Table 9: SLADE component ablation, reported as AUC (mean±std, %).

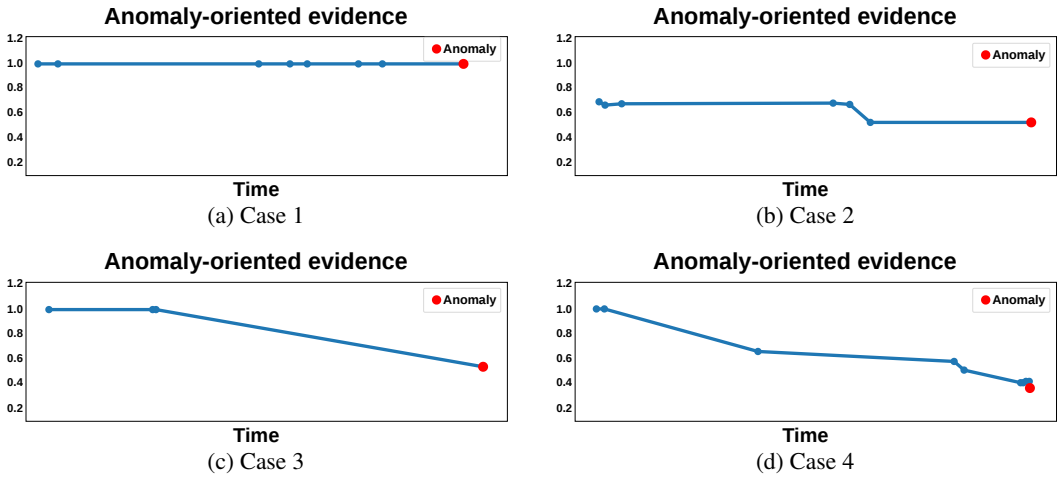
Variant	Wikipedia	Reddit	BitcoinAlpha	BitcoinOTC
Drift only	85.69±1.17	48.89±1.53	**76.38±0.48**	73.87±0.23
Reconstruction only	87.57±0.35	61.77±0.12	74.84±0.40	**75.80±0.26**
Reconstruction + drift	**87.75±0.68**	**72.19±0.60**	76.32±0.28	**75.80±0.19**

A.5 Reconstruction–Drift Evidence Ablation

Since our evidence module follows SLADE (Lee et al., 2024), we summarize its component ablation in Table 9. In SLADE, $(\mathcal{L}_c, \mathrm{sc}_c)$ corresponds to drift-style temporal evidence, and $(\mathcal{L}_g, \mathrm{sc}_g)$ corresponds to reconstruction-style evidence. Using both channels achieves the best overall performance, supporting our use of joint reconstruction–drift evidence.

B Evidence Trajectory Examples

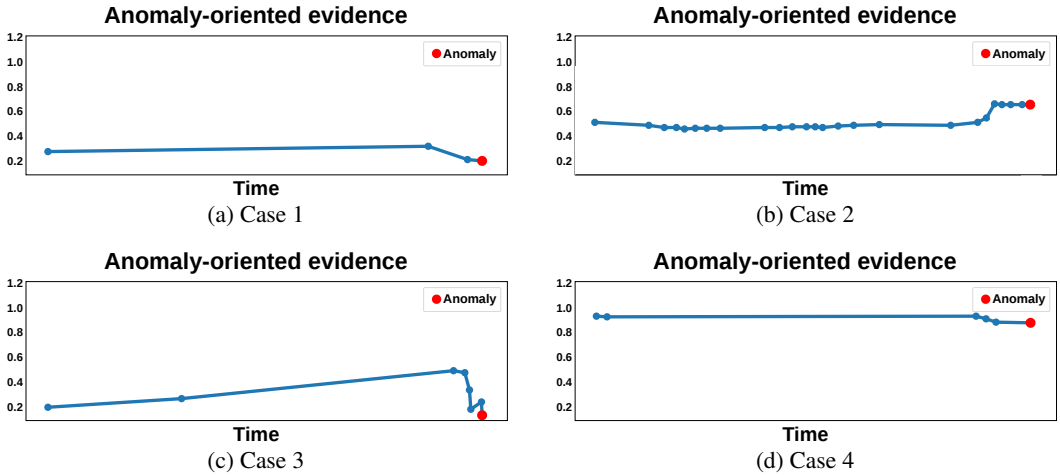
Figures 3–6 illustrate source-wise reconstruction–drift evidence trajectories and observed anomaly labels on Wikipedia, Reddit, BitcoinAlpha, and BitcoinOTC. Blue curves show normalized MI-based evidence, oriented so that larger values indicate stronger anomaly evidence; red markers indicate observed anomaly labels. These transformations are for visualization only: TeCoS uses the raw MI-based reconstruction and drift scores as inputs to its temporal predictor.



(a) Case 1 (b) Case 2

(c) Case 3 (d) Case 4

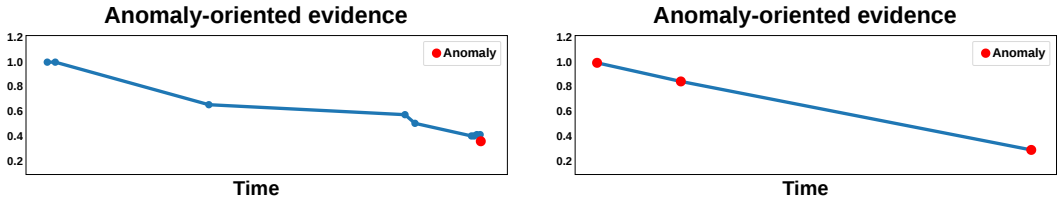
Figure 3: Representative source-wise anomaly-oriented evidence trajectories from Wikipedia.



(a) Case 1 (b) Case 2

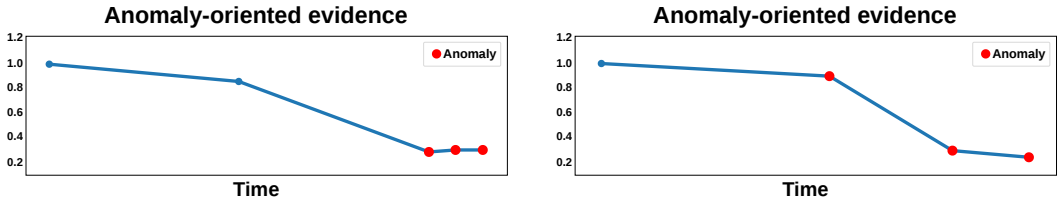
(c) Case 3 (d) Case 4

Figure 4: Representative source-wise anomaly-oriented evidence trajectories from Reddit.



(a) Case 1

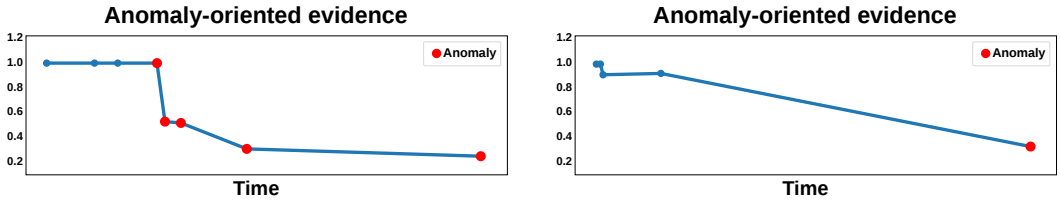
(b) Case 2



(c) Case 3

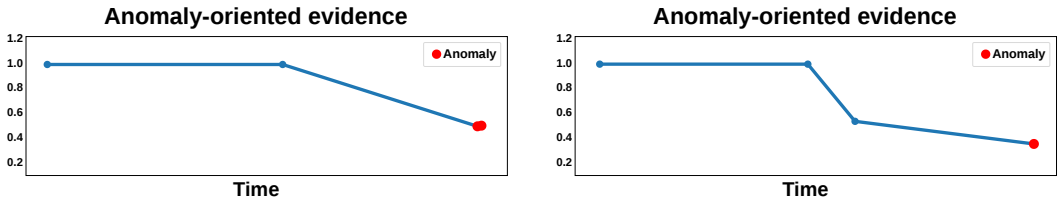
(d) Case 4

Figure 5: Representative source-wise anomaly-oriented evidence trajectories from BitcoinAlpha.



(a) Case 1

(b) Case 2



(c) Case 3

(d) Case 4

Figure 6: Representative source-wise anomaly-oriented evidence trajectories from BitcoinOTC.

C Datasets

C.1 Dataset Details

We evaluate TeCoS on seven temporal interaction datasets: Wikipedia, Reddit, MOOC, BitcoinAlpha, BitcoinOTC, EU-Core, and Email-DNC. These datasets cover social, educational, financial, and communication networks and contain different forms of anomaly supervision. Table 10 summarizes their basic statistics.

Wikipedia, Reddit, and MOOC (Kumar et al., 2019). Wikipedia and Reddit are temporal bipartite interaction networks between users and content: Wikipedia records user–page edits, whereas Reddit records user–subreddit posts. Their event-linked labels indicate whether the source user was banned after the corresponding interaction (Lee et al., 2024). MOOC is a temporal bipartite network containing interactions between students and online course activities. A positive MOOC label indicates that the student dropped out after the corresponding action, i.e., that the action is the student's final recorded action. In our event-level evaluation, interactions carrying positive labels are treated as anomalous.

BitcoinAlpha and BitcoinOTC (Kumar et al., 2016). BitcoinAlpha and BitcoinOTC are temporal signed networks from Bitcoin trading platforms. Each original interaction is a time-stamped rating from one user to another, ranging from -10 (complete distrust) to $+10$ (complete trust). Following

Table 10: Statistics of the seven temporal interaction datasets. The anomaly count refers to the number of source-side events labeled as anomalous, and the anomaly ratio is computed over all evaluated events.

Dataset	# Nodes	# Edges	# Anomalies	Anomaly ratio
Wikipedia	9,227	157,474	217	0.14%
Reddit	10,984	672,447	366	0.05%
MOOC	7,144	411,749	4,066	0.99%
BitcoinAlpha	3,783	24,186	874	3.61%
BitcoinOTC	5,881	35,592	2,568	7.22%
EU-Core	986	332,334	33,233	10.00%
Email-DNC	1,891	39,264	3,926	10.00%

the proxy-label construction used by SLADE (Lee et al., 2024), we first assign each user an overall state based on the sum of trust ratings received over the complete interaction stream. A user whose aggregate received rating is negative is considered unreliable. For such users, interactions in which they receive negative ratings are assigned positive dynamic labels, whereas interactions carrying positive ratings remain normal. Users with nonnegative aggregate received ratings remain normal throughout the stream. Because these labels originally describe the destination receiving the rating, we follow SLADE and reverse the processed edge direction so that the labeled user becomes the source-side actor being evaluated.

EU-Core and Email-DNC. EU-Core (Paranjape et al., 2017) and Email-DNC (Rossi & Ahmed, 2015) are temporal communication networks containing time-stamped emails between participants. We adapt the BAG construction (Hua et al., 2026), based on the anomaly-injection strategy of TADDY (Liu et al., 2021). Candidate anomalies are directed non-edges whose endpoints belong to different spectral clusters. Selected normal interactions are replaced by these candidates while retaining their timestamps and the total number of interactions. We use an overall anomaly ratio of 10% and split each dataset chronologically into training, validation, and test sets with a 70%/15%/15% ratio. The resulting datasets and split assignments are fixed and shared across all methods and random seeds.

Preprocessing and splits. We store every dataset as a chronologically ordered, with reindexed node identifiers. For bipartite datasets, the source and destination node spaces are kept disjoint. MOOC retains its four-dimensional raw edge features, whereas all-zero node and edge features are provided when raw features are unavailable.

We use a chronological 70%/15%/15% split for training, validation, and testing. The cutoffs are determined by the 0.70 and 0.85 quantiles of interaction timestamps: events at or before the first cutoff form the training set, events after the first cutoff and at or before the second form the validation set, and the remaining events form the test set. Equal timestamps are kept in the same partition. All methods use the same fixed interaction files and split assignments. Checkpoints are selected by validation AP, with validation AUC and then the earlier epoch used to break ties. The actual event and anomaly counts are reported in Table 5.

D Additional Details

D.1 Notation

Table 11 summarizes the frequently used notation in our two-stage framework. We focus on symbols used in evidence extraction, temporal contextualization, and score correction, while omitting analysis-only symbols local to §3.

D.2 Node Memory Embedding

Following SLADE (Lee et al., 2024), the upstream encoder maintains a memory vector $\mathbf{m}_v \in \mathbb{R}^{d_m}$ for each node v, initialized as zero and updated chronologically. For the i-th interaction $e_i = (u_i, v_i, t_i)$, $\mathbf{m}'_v$ and $\mathbf{m}_v$ denote the memory of v before and after the current update, respectively.

Table 11: Frequently used notation in the two-stage framework.

Stage 1: evidence and representation extraction		**Stage 2: temporal contextualization and correction**	
Notation	Description	Notation	Description
$e_i = (u_i, v_i, t_i)$	Interaction i.	$\pi_\ell(i)$	ℓ-th same-source predecessor.
$\mathbf{m}_{u_i,i}$	Current source memory.	$\mathbf{X}_i$	Local temporal window.
$\mathbf{m}'_{u_i,i}$	Source reference memory.	$\overline{\mathbf{x}}_i$	Standardized event feature.
$\widehat{\mathbf{m}}_{u_i,i}$	Reconstructed source memory.	$\mathbf{h}_{i,\ell}$	Projected event feature.
$\mathbf{m}_{v_i,i}$	Current destination memory.	$\mathbf{g}_i$	Contextual representation.
$\widehat{\mathbf{m}}_{v_i,i}$	Reconstructed destination memory.	r_i	Raw correction score.
$\mathcal{N}_i^-$	Negative memory set.	Δ_i	Bounded score correction.
$f_{\cos}$	Cosine similarity.	γ	Correction magnitude.
$\widehat{I}$	Contrastive dependence score.	s_i	Final anomaly score.
c_i^{rec}	Reconstruction evidence.	α	Stage-1 score weight.
c_i^{drift}	Temporal drift evidence.		
$\mathbf{x}_i$	Event feature.		
b_i	Initial anomaly score.		

To construct an interaction-conditioned reconstructed memory for node v, the encoder retrieves its recent temporal neighborhood $\mathcal{N}(v, t_i)$. For each neighbor $w \in \mathcal{N}(v, t_i)$ with interaction time $t_w < t_i$, the neighbor memory is paired with a relative time encoding $\Phi(t_i - t_w)$. A temporal attention module aggregates these time-aware neighbor states:

$$\hat{\mathbf{m}}_v = \mathrm{Attn}_\theta\left(\Phi(0), \{[\mathbf{m}'_w; \Phi(t_i - t_w)] : w \in \mathcal{N}(v, t_i)\}\right), \qquad \hat{\mathbf{m}}_v \in \mathbb{R}^{d_m}. \tag{14}$$

Here $\Phi(0)$ is the query time representation, and the neighbor memories with relative time encodings serve as keys and values. The resulting $\hat{\mathbf{m}}_v$ is used as the reconstructed memory.

The memory update is message-based. Each interaction is first transformed into a raw message $\tilde{\mathbf{r}}_v = [\mathbf{m}'; \Phi(t_i - t'_v)]$ by concatenating a previous memory state used for message construction with a time encoding of the elapsed time since the node's last interaction.

Here, t'_v denotes the previous interaction time of node v. The raw message is mapped by an MLP and used to update the node memory with a recurrent updater:

$$\mathbf{r}_v = \mathrm{MLP}\left(\tilde{\mathbf{r}}_v\right), \qquad \mathbf{m}_v = \mathrm{GRU}\left(\mathbf{r}_v, \mathbf{m}'_v\right), \qquad \mathbf{m}_v \in \mathbb{R}^{d_m}. \tag{15}$$

For each interaction, the same mechanism is applied to both endpoints. The downstream temporal predictor uses the source-side reconstruction–drift evidence together with the current and reconstructed memory representations of both interaction endpoints, as defined in Eq. (5).

E Robustness Under Delayed Labels

E.1 Setup and Scope

This appendix provides the complete proofs of Propositions 3.1 and 3.2, together with the derivation of the linear lag-recovery illustration in Section 3. The purpose of the analysis is to explain why applying supervised BCE after local temporal contextualization can be more robust than applying it directly to isolated reconstruction–drift evidence.

We follow the notation in Section 3. Let $\mathbf{z}_t = [c_t^{\mathrm{rec}}, c_t^{\mathrm{drift}}] \in \mathbb{R}^2$ denote the reconstruction–drift evidence at source-side interaction position t, and let $A_t \in \{0, 1\}$ denote the corresponding latent anomaly state. We consider the representative fixed-delay model

$$\widetilde{A}_t = A_{t-\Delta}, \qquad \Delta \in \mathbb{N}_{>0}, \tag{16}$$

where $\widetilde{A}_t$ is the observed label at position t. The set of positions at which the current latent state and the observed label disagree is

$$\mathcal{B}_\Delta = \{t : A_t \neq \widetilde{A}_t\}. \tag{17}$$

For a history length $k \geq 0$, define the historical context as

$$\mathbf{C}_t^{(k)} = (\mathbf{z}_{t-k}, \ldots, \mathbf{z}_{t-1}, \mathbf{z}_t) = (\mathbf{z}_t, \mathbf{N}_t^{(k)}), \tag{18}$$

where $\mathbf{N}_t^{(k)}$ collects the preceding evidence variables in the context. This historical context abstracts the reconstruction–drift evidence component available within the causal temporal receptive field of Stage 2 in Section 4.2. At the beginning of a source-wise interaction history, fewer preceding events may be available; statements that require $\mathbf{z}_{t-\Delta}$ therefore apply when the delay-aligned evidence lies within the available historical context.

For any feature variable $\mathbf{V}$, define the Bayes-optimal BCE risk for predicting $\widetilde{A}_t$ as

$$R^*(\mathbf{V}) = \inf_f \mathbb{E}\Big[\ell_{\mathrm{BCE}}(f(\mathbf{V}), \widetilde{A}_t)\Big], \tag{19}$$

where f ranges over measurable logit predictors and the expectation is taken with respect to the joint population distribution of source-wise interaction positions.

The fixed-delay model is an idealized mechanism model rather than a claim that all real annotation delays are constant. Likewise, the Bayes-risk result below characterizes the information available to unrestricted population-optimal predictors; it does not guarantee that a finite-sample Conv1D trained by SGD will attain the Bayes risk or recover an exact delay.

E.2 Proof of Boundary Fragility

Proof of Proposition 3.1. For BCEWithLogits, the loss for a logit $u \in \mathbb{R}$ and a binary target $y \in \{0, 1\}$ is

$$\ell_{\mathrm{BCE}}(u, y) = \log(1 + \exp(u)) - yu, \tag{20}$$

and its logit derivative is

$$\partial_u \ell_{\mathrm{BCE}}(u, y) = \sigma(u) - y. \tag{21}$$

For every $t \in \mathcal{B}_\Delta$, the two binary variables A_t and $\widetilde{A}_t$ differ. Consequently,

$$\{A_t, \widetilde{A}_t\} = \{0, 1\}.$$

For any finite logit u_t, we have $0 < \sigma(u_t) < 1$. Irrespective of which of A_t and $\widetilde{A}_t$ equals one, it follows that

$$\begin{aligned} &\big(\sigma(u_t) - \widetilde{A}_t\big)\big(\sigma(u_t) - A_t\big) \\ &\quad = \sigma(u_t)\big(\sigma(u_t) - 1\big) < 0. \end{aligned} \tag{22}$$

Therefore, the gradients induced by the observed delayed label and by the current latent anomaly state have opposite signs at every mismatched position. This proves the proposition. □

Explicit boundary bands for a single episode. The mismatch set can be characterized explicitly for a single anomalous episode. Let

$$\mathcal{A} = [\tau, \tau + L_A - 1]$$

with $A_t = 1$ for $t \in \mathcal{A}$ and $A_t = 0$ otherwise. Under (16), the observed label is positive on the shifted interval

$$\mathcal{A} + \Delta = [\tau + \Delta, \tau + \Delta + L_A - 1].$$

Hence, the mismatch set is the symmetric difference

$$\mathcal{B}_\Delta = \mathcal{A} \triangle (\mathcal{A} + \Delta). \tag{23}$$

When $0 < \Delta < L_A$, the two intervals overlap and

$$\mathcal{B}_\Delta = [\tau, \tau + \Delta - 1] \cup [\tau + L_A, \tau + L_A + \Delta - 1]. \tag{24}$$

On the onset band,

$$A_t = 1, \qquad \widetilde{A}_t = 0,$$

because the latent episode has started but $t - \Delta < \tau$. On the offset band,

$$A_t = 0, \qquad \widetilde{A}_t = 1,$$

because the latent episode has ended while $t - \Delta \in \mathcal{A}$. When $\Delta \geq L_A$, the two intervals are disjoint and

$$\mathcal{B}_\Delta = \mathcal{A} \cup (\mathcal{A} + \Delta).$$

In both cases, Proposition 3.1 applies to every position in $\mathcal{B}_\Delta$.

E.3 Proof of the Bayes-Risk Advantage

We first establish the standard relationship between Bayes-optimal BCE risk and conditional entropy.

Lemma E.1 (Bayes-optimal BCE risk)**.** *Let $Y \in \{0, 1\}$ be a binary target and let Z be any feature variable. For the BCEWithLogits loss*

$$\ell_{\mathrm{BCE}}(r, Y) = \mathrm{softplus}(r) - Yr,$$

the infimum of the population risk over all measurable logit predictors $r = f(Z)$ is

$$\inf_f \mathbb{E}[\ell_{\mathrm{BCE}}(f(Z), Y)] = H(Y \mid Z), \tag{25}$$

where entropy is measured in nats.

Proof. Fix a feature value $Z = z$, and let

$$q(z) = \mathbb{P}(Y = 1 \mid Z = z).$$

For a fixed logit r, the conditional expected loss is

$$\begin{aligned} \mathbb{E}[\ell_{\mathrm{BCE}}(r, Y) \mid Z = z] &= q(z)\ell_{\mathrm{BCE}}(r, 1) + (1 - q(z))\ell_{\mathrm{BCE}}(r, 0) \\ &= \mathrm{softplus}(r) - q(z)r. \end{aligned} \tag{26}$$

Let $p = \sigma(r)$. Using

$$r = \log \frac{p}{1 - p},$$

the conditional expected BCE can equivalently be written as

$$\phi(p) = -q(z) \log p - (1 - q(z)) \log(1 - p). \tag{27}$$

For $q(z) \in (0, 1)$, differentiation gives

$$\phi'(p) = -\frac{q(z)}{p} + \frac{1 - q(z)}{1 - p},$$

and $\phi'(p) = 0$ if and only if $p = q(z)$. Moreover,

$$\phi''(p) = \frac{q(z)}{p^2} + \frac{1 - q(z)}{(1 - p)^2} > 0,$$

so this minimizer is unique. For $q(z) = 0$ or $q(z) = 1$, the same infimum is obtained in the limit as p approaches $q(z)$. Substitution yields

$$\begin{aligned} \inf_p \phi(p) &= -q(z) \log q(z) - (1 - q(z)) \log(1 - q(z)) \\ &= h(q(z)), \end{aligned} \tag{28}$$

where $h(q) = -q \log q - (1 - q) \log(1 - q)$. Taking the expectation over Z gives

$$\inf_f \mathbb{E}[\ell_{\mathrm{BCE}}(f(Z), Y)] = \mathbb{E}_Z[h(\mathbb{P}(Y = 1 \mid Z))] = H(Y \mid Z),$$

which proves the lemma. □

Proof of Proposition 3.2. Applying Lemma E.1 with $Y = \widetilde{A}_t$ gives

$$R^*(\mathbf{z}_t) = H(\widetilde{A}_t \mid \mathbf{z}_t) \tag{29}$$

and

$$R^*(\mathbf{C}_t^{(k)}) = H(\widetilde{A}_t \mid \mathbf{C}_t^{(k)}). \tag{30}$$

Because $\mathbf{C}_t^{(k)} = (\mathbf{z}_t, \mathbf{N}_t^{(k)})$, subtracting (30) from (29) yields

$$\begin{aligned} &R^*(\mathbf{z}_t) - R^*(\mathbf{C}_t^{(k)}) \\ &\quad = H(\widetilde{A}_t \mid \mathbf{z}_t) - H(\widetilde{A}_t \mid \mathbf{z}_t, \mathbf{N}_t^{(k)}) \\ &\quad = I(\widetilde{A}_t; \mathbf{N}_t^{(k)} \mid \mathbf{z}_t) \geq 0. \end{aligned} \tag{31}$$

The last inequality follows from the non-negativity of conditional mutual information. It is strict precisely when

$$I(\widetilde{A}_t; \mathbf{N}_t^{(k)} \mid \mathbf{z}_t) > 0, \tag{32}$$

that is, when the preceding evidence contains information about the observed label that is unavailable from the current evidence alone.

We next establish the stated delay-aligned sufficient condition. If $k \geq \Delta$ and the delay-aligned position lies in the valid receptive field, then $\mathbf{z}_{t-\Delta}$ is one component of $\mathbf{N}_t^{(k)}$. Let $\mathbf{O}_t$ collect the remaining preceding evidence variables, so that, up to a permutation of components,

$$\mathbf{N}_t^{(k)} = (\mathbf{z}_{t-\Delta}, \mathbf{O}_t).$$

The chain rule for conditional mutual information gives

$$\begin{aligned} I(\widetilde{A}_t; \mathbf{N}_t^{(k)} \mid \mathbf{z}_t) &= I(\widetilde{A}_t; \mathbf{z}_{t-\Delta}, \mathbf{O}_t \mid \mathbf{z}_t) \\ &= I(\widetilde{A}_t; \mathbf{z}_{t-\Delta} \mid \mathbf{z}_t) \\ &\quad + I(\widetilde{A}_t; \mathbf{O}_t \mid \mathbf{z}_t, \mathbf{z}_{t-\Delta}) \\ &\geq I(\widetilde{A}_t; \mathbf{z}_{t-\Delta} \mid \mathbf{z}_t). \end{aligned} \tag{33}$$

Therefore, if

$$I(\widetilde{A}_t; \mathbf{z}_{t-\Delta} \mid \mathbf{z}_t) > 0,$$

then

$$R^*(\mathbf{z}_t) - R^*(\mathbf{C}_t^{(k)}) > 0,$$

or equivalently,

$$R^*(\mathbf{C}_t^{(k)}) < R^*(\mathbf{z}_t).$$

This proves both the general risk identity and the delay-aligned strict improvement condition. □

Equivalent function-class interpretation. The non-strict inequality also follows directly from hypothesis-class containment. Since $\mathbf{C}_t^{(k)}$ contains $\mathbf{z}_t$, there exists a projection π such that

$$\pi(\mathbf{C}_t^{(k)}) = \mathbf{z}_t.$$

For every point-wise predictor $g(\mathbf{z}_t)$, define

$$f_g(\mathbf{C}_t^{(k)}) = g(\pi(\mathbf{C}_t^{(k)})).$$

Then $f_g(\mathbf{C}_t^{(k)}) = g(\mathbf{z}_t)$. Thus, a sufficiently rich context-aware predictor can reproduce every point-wise rule by ignoring its preceding inputs, implying

$$R^*(\mathbf{C}_t^{(k)}) \leq R^*(\mathbf{z}_t).$$

Equation (31) further identifies exactly when the additional context makes this inequality strict.

E.4 Derivation of the Linear Lag-Recovery Example

We now derive the linear illustration stated in Section 3. Let $x_t \in \mathbb{R}$ be a scalar projection of the evidence $\mathbf{z}_t$, and suppose that the evidence sequence is zero-mean, standardized, and temporally uncorrelated:

$$\mathbb{E}[x_t] = 0, \qquad \mathbb{E}[x_t x_s] = \mathbf{1}[t = s]. \tag{34}$$

Consider the real-valued supervision surrogate

$$r_t = \beta x_{t-\Delta} + \varepsilon_t, \qquad \beta \neq 0, \tag{35}$$

where

$$\mathbb{E}[\varepsilon_t] = 0, \qquad \mathbb{E}[\varepsilon_t^2] < \infty, \qquad \mathbb{E}[\varepsilon_t x_{t+j}] = 0 \quad \text{for all } j \in \{-k, \dots, 0\}. \tag{36}$$

For integer $k \geq \Delta$, consider the linear temporal predictor over the historical window

$$\widehat{r}_t = \sum_{j=-k}^{0} w_j x_{t+j}. \tag{37}$$

Its population squared loss is

$$\begin{aligned} \mathcal{L}_{\text{lin}}(\mathbf{w}) &= \mathbb{E}\big[(r_t - \widehat{r}_t)^2\big] \\ &= \mathbb{E}\left[\left(\beta x_{t-\Delta} + \varepsilon_t - \sum_{j=-k}^{0} w_j x_{t+j}\right)^2\right]. \end{aligned} \tag{38}$$

Separating the coefficient of the delay-aligned evidence gives

$$\mathcal{L}_{\text{lin}}(\mathbf{w}) = \mathbb{E}\left[\left((\beta - w_{-\Delta}) x_{t-\Delta} - \sum_{\substack{j=-k \\ j \neq -\Delta}}^{0} w_j x_{t+j} + \varepsilon_t\right)^2\right]. \tag{39}$$

By (34) and (36), all cross terms vanish. Therefore,

$$\mathcal{L}_{\text{lin}}(\mathbf{w}) = (\beta - w_{-\Delta})^2 + \sum_{\substack{j=-k \\ j \neq -\Delta}}^{0} w_j^2 + \mathbb{E}[\varepsilon_t^2]. \tag{40}$$

The last term does not depend on $\mathbf{w}$, and every other term is non-negative. Hence, the unique population minimizer is

$$w_j^* = \begin{cases} \beta, & j = -\Delta, \\ 0, & j \neq -\Delta. \end{cases} \tag{41}$$

Thus, when the historical receptive field covers the evidence–supervision lag, population squared-loss minimization places its nonzero weight on the delay-aligned evidence position.

This result complements Proposition 3.2. The proposition is a general Bayes-risk result for BCE, whereas the present example uses a simplified linear model and squared loss. Its role is to provide a constructive illustration of how a learnable temporal filter can emphasize a delay-aligned evidence position using preceding context. It does not imply that a finite-sample nonlinear predictor is guaranteed to recover an exact lag.

E.5 Takeaway

The analysis establishes three complementary conclusions.

Boundary fragility of isolated supervision. At every position where the delayed observed label differs from the current latent anomaly state, BCEWithLogits produces a logit gradient whose sign is opposite to that induced by the latent state. For a delayed anomalous episode, these disagreements occur in explicit onset and offset boundary regions.

Bayes-risk advantage of temporal context. The exact reduction in the Bayes-optimal BCE risk obtained by replacing isolated evidence with a historical context is

$$R^*(\mathbf{z}_t) - R^*(\mathbf{C}_t^{(k)}) = I(\widetilde{A}_t; \mathbf{N}_t^{(k)} \mid \mathbf{z}_t) \geq 0.$$

The reduction is strict when the preceding evidence contains information about the observed label that is unavailable from $\mathbf{z}_t$. A valid delay-aligned evidence position provides one sufficient route to this strict improvement.

Constructive illustration of lag recovery. Under the simplified linear model, when the historical receptive field covers the lag, the unique population squared-loss solution places its nonzero weight at the delay-aligned offset. This illustrates how a learnable temporal filter can emphasize temporally aligned evidence.

Together, these results motivate applying supervision after local temporal contextualization of reconstruction–drift evidence. They explain why temporal context can help under evidence–label misalignment without changing the event-level prediction target, while avoiding a claim that finite-sample training is guaranteed to recover an exact annotation delay.

F Controlled Label-Delay Experiments

This appendix supplements Section 5.4 with detailed protocols, complete numerical results, a mean-pooling baseline, and label counts under uniform artificial delays.

F.1 Experimental Protocol

Data and temporal scope. We evaluate uniform delays on BitcoinAlpha, BitcoinOTC, EU-Core, and Email-DNC. All experiments use the chronological 70/15/15 train/validation/test partitions from the main evaluation. Labels are transformed independently within each source-wise partition and never cross source or partition boundaries. All interaction rows are retained. Prediction windows contain only the current and preceding same-source events; historical features may cross partition boundaries, but labels are never prediction inputs.

Uniform delays. Let $y_{u,j}^{(p)}$ denote the original label at position j in source u's sequence within partition p, with length $n_u^{(p)}$. For $d \in \{0, 1, 2, 3\}$, the transformed label at position k is

$$\widetilde{y}_{u,k}^{(p)} = \begin{cases} y_{u,k-d}^{(p)}, & 1 \leq k - d \leq n_u^{(p)}, \\ 0, & \text{otherwise.} \end{cases} \tag{42}$$

Thus, positive labels move forward by d same-source interactions, and those shifted beyond the sequence boundary are discarded. Setting $d = 0$ preserves the original labels.

Compared predictors. All predictors use the same full event features: the two scalar evidence scores and the source- and destination-side memory representations. They also share the base-score branch and bounded correction formulation.

Current-only retains the three-position convolutional architecture but masks both historical positions after projection. It shares the initialization and nominal parameter count of TeCoS ($H = 2$), with 78,145 trainable parameters.

Mean pooling averages the valid projected features of the current event and its two predecessors, followed by a learned linear transformation, ReLU, dropout, and the scalar output head.

TeCoS *($H = 2$)* applies position-sensitive convolution to the current event and its two predecessors.

Current-only still receives history-aware encoder representations; the comparison isolates the incremental contribution of neighboring event features in Stage 2.

Training and model selection. We freeze the encoder and event features and retrain Stage 2 for each nonzero delay using transformed training labels. Training follows Appendix A, with label-dependent quantities recomputed from the transformed training labels and checkpoints selected

Table 12: AUC under uniform label delays (mean±population standard deviation, %). Both contextual predictors use $H = 2$. All methods within a condition use identical transformed labels.

Dataset	Delay	Current-only	Mean pooling	TECOS
BitcoinAlpha	$d = 0$	76.99 ± 0.13	77.54 ± 0.24	78.24 ± 0.03
	$d = 1$	70.40 ± 0.50	71.75 ± 0.27	73.22 ± 0.47
	$d = 2$	60.45 ± 0.74	64.00 ± 0.42	65.60 ± 0.79
	$d = 3$	55.31 ± 0.69	58.60 ± 0.85	58.38 ± 1.13
BitcoinOTC	$d = 0$	76.13 ± 0.09	76.76 ± 0.35	76.72 ± 0.34
	$d = 1$	69.05 ± 0.57	69.03 ± 0.59	68.36 ± 0.35
	$d = 2$	56.97 ± 0.90	58.88 ± 1.21	59.14 ± 0.59
	$d = 3$	48.10 ± 0.67	48.50 ± 0.64	48.46 ± 0.64
EU-Core	$d = 0$	90.95 ± 0.26	89.52 ± 0.38	92.27 ± 0.22
	$d = 1$	86.08 ± 0.66	89.02 ± 0.40	91.93 ± 0.27
	$d = 2$	84.47 ± 0.29	87.67 ± 0.82	90.46 ± 0.25
	$d = 3$	82.34 ± 0.11	82.72 ± 0.58	83.05 ± 0.40
Email-DNC	$d = 0$	98.07 ± 0.09	96.07 ± 0.20	98.31 ± 0.04
	$d = 1$	87.21 ± 0.43	90.08 ± 0.17	92.60 ± 0.15
	$d = 2$	80.60 ± 1.45	86.04 ± 0.55	90.75 ± 0.41
	$d = 3$	75.29 ± 2.18	77.51 ± 0.78	78.90 ± 1.52

Table 13: AP under uniform label delays (mean±population standard deviation, %). Both contextual predictors use $H = 2$. Anomaly counts vary across delay conditions; see Table 14.

Dataset	Delay	Current-only	Mean pooling	TECOS
BitcoinAlpha	$d = 0$	14.90 ± 0.43	14.57 ± 0.71	15.69 ± 0.24
	$d = 1$	7.35 ± 0.06	7.03 ± 0.21	9.24 ± 0.26
	$d = 2$	2.54 ± 0.03	3.04 ± 0.09	4.86 ± 0.66
	$d = 3$	1.54 ± 0.02	1.70 ± 0.02	1.77 ± 0.06
BitcoinOTC	$d = 0$	21.65 ± 0.17	22.00 ± 0.76	22.34 ± 0.83
	$d = 1$	14.65 ± 0.38	11.53 ± 0.36	14.78 ± 0.38
	$d = 2$	4.05 ± 0.18	6.97 ± 0.29	11.01 ± 1.05
	$d = 3$	2.31 ± 0.68	3.17 ± 0.21	4.64 ± 0.77
EU-Core	$d = 0$	46.66 ± 3.91	50.81 ± 4.10	53.08 ± 3.97
	$d = 1$	37.21 ± 6.15	39.61 ± 4.44	46.39 ± 3.57
	$d = 2$	31.41 ± 1.16	32.44 ± 4.89	37.85 ± 1.79
	$d = 3$	23.91 ± 0.13	23.81 ± 2.94	23.57 ± 1.51
Email-DNC	$d = 0$	84.21 ± 1.17	71.55 ± 0.43	87.13 ± 0.21
	$d = 1$	11.33 ± 0.64	13.15 ± 0.31	15.87 ± 0.30
	$d = 2$	3.19 ± 0.32	4.33 ± 0.18	6.13 ± 0.22
	$d = 3$	1.38 ± 0.16	1.51 ± 0.09	1.68 ± 0.19

by transformed validation AP. We report mean±std over three seeds; zero-delay results reuse the original-label runs.

F.2 RESULTS UNDER UNIFORM DELAYS

Tables 12 and 13 report the complete results. Across the 12 nonzero-delay conditions, TECOS achieves higher mean AUC and AP than Current-only in 11 conditions for each metric. The exceptions are BitcoinOTC AUC at $d = 1$ and EU-Core AP at $d = 3$. Compared with Mean pooling, TECOS achieves higher mean AUC in nine conditions and higher mean AP in 11 conditions. These results support the usefulness of local temporal context and position-sensitive aggregation under the evaluated uniform delays, with dataset-dependent benefits.

The $H = 2$ correction window directly accesses only two predecessors. Improvements at $d = 3$ therefore do not imply direct access to the delay-aligned event through this window.

Table 14: Positive-label counts under uniform delays. Train, Val., and Test denote retained positives; Boundary denotes test-set positives lost at sequence boundaries. All interaction rows are retained.

Dataset	Delay	Train	Val.	Test	Boundary
BitcoinAlpha	$d = 0$	521	123	230	0
	$d = 1$	388	73	128	102
	$d = 2$	304	53	75	155
	$d = 3$	247	41	53	177
BitcoinOTC	$d = 0$	1,123	975	470	0
	$d = 1$	760	724	225	245
	$d = 2$	546	585	136	334
	$d = 3$	414	492	91	379
EU-Core	$d = 0$	23,240	5,008	4,985	0
	$d = 1$	22,679	4,473	4,458	527
	$d = 2$	22,184	3,983	3,978	1,007
	$d = 3$	21,676	3,530	3,540	1,445
Email-DNC	$d = 0$	2,744	593	589	0
	$d = 1$	1,636	144	159	430
	$d = 2$	966	67	65	524
	$d = 3$	592	44	34	555

F.3 Label Counts and Interpretation

Table 14 reports retained positive labels in each partition and test-set boundary losses. All interaction rows are retained, but positives shifted beyond source-wise partition boundaries are discarded. For example, Email-DNC retains only 34 of its 589 original test positives at $d = 3$.

Label shifts therefore change both the target labels and anomaly prevalence. We emphasize comparisons within the same delay condition, since declining AP across delays cannot be attributed solely to temporal misalignment.

G Additional Experimental Results

G.1 Temporal Graph Models

We additionally include TGN (Rossi et al., 2020) and DyGFormer (Yu et al., 2023) as representative temporal graph models. Both models incorporate historical interactions when constructing the representation of the current event, but use different temporal modeling mechanisms: TGN maintains node memories, whereas DyGFormer models historical interactions through attention. For anomaly detection, we attach a classification head to the source-side representation of the current event and optimize the model using the same anomaly labels, training schedule, and three random seeds as TeCoS.

Table 15: Additional comparison with temporal graph models. We report AUC as mean±std (%); higher is better. Bold entries indicate the best result on each dataset.

Method	Wikipedia	Reddit	BitcoinAlpha	BitcoinOTC
TGN	85.44±0.14	55.06±2.04	52.18±3.13	54.17±2.22
DyGFormer	84.69±0.05	62.40±0.35	70.35±0.89	66.13±2.67
TeCoS	**87.66±0.93**	**73.22±1.36**	**78.24±0.03**	**76.72±0.34**

Table 15 shows that TeCoS achieves higher AUC than TGN and DyGFormer on all four datasets. Compared with the stronger of the two temporal graph models on each dataset, TeCoS improves AUC by 2.22, 10.82, 7.89, and 10.59 points on Wikipedia, Reddit, BitcoinAlpha, and BitcoinOTC, respectively. These results provide an additional comparison with temporal models that directly incorporate interaction history into event representations and further support the effectiveness of contextualizing anomaly-specific evidence.

G.2 GENERIC WINDOW AGGREGATION

We compare TECOS with two generic strategies for aggregating evidence within overlapping local windows. Both alternatives construct three-event windows with stride one. Max-window assigns each window the maximum point-wise anomaly score, while Softmax-window applies a shared MLP to each evidence position and combines the resulting representations using softmax-normalized weights. Without positional encoding, these aggregation rules do not distinguish between different orderings of the same evidence values. By contrast, TECOS applies position-sensitive Conv1D filters to the current event and its recent same-source history.

Table 16: Comparison with generic overlapping-window aggregation. AUC is reported as mean±std (%) over three random seeds. Bold values indicate the best result on each dataset.

Method	Wikipedia	Reddit	BitcoinAlpha	BitcoinOTC
Max-window	81.49±0.61	57.26±1.58	74.89±0.79	73.67±0.39
Softmax-window	82.44±0.62	43.89±2.33	77.14±0.74	75.30±0.46
TECOS	**87.66±0.93**	**73.22±1.36**	**78.24±0.03**	**76.72±0.34**

As shown in Table 16, TECOS consistently outperforms both generic window-aggregation strategies across all four datasets. Compared with the stronger aggregation baseline on each dataset, TECOS improves AUC by 5.22, 15.96, 1.10, and 1.42 points on Wikipedia, Reddit, BitcoinAlpha, and BitcoinOTC, respectively. The gains are particularly pronounced on Wikipedia and Reddit, indicating that simply pooling evidence within a local window is insufficient on these datasets. These results support the use of position-sensitive temporal contextualization rather than order-invariant aggregation of neighboring evidence.