Spatio-Temporal Directed Graph Learning for Account Takeover Fraud Detection

Mohsen Nayebi Kerdabadi
mohsen.nayebikerdabadi@capitalone.com

William Andrew Byron
drew.byron@capitalone.com

Xin Sun
xin.sun@capitalone.com

Amirfarrokh Iranitalab*
amirfarrokh.iranitalab@capitalone.com

AI Foundations, Capital One, McLean, Virginia, United States

Abstract

Account Takeover (ATO) fraud poses a significant challenge in consumer banking, requiring high recall under strict latency while minimizing friction for legitimate users. Production systems typically rely on tabular gradient-boosted decision trees (e.g., XGBoost) that score sessions independently, overlooking the relational and temporal structure of online activity that characterizes coordinated attacks and "fraud rings." We introduce **ATLAS** (**A**ccount **T**akeover **L**earning **A**cross **S**patio-Temporal Directed Graph), a framework that reformulates ATO detection as spatio-temporal node classification on a time-respecting directed session graph. ATLAS links entities via shared identifiers (account, device, IP) and regulates connectivity with time-window and recency constraints, enabling causal, time-respecting message passing and latency–aware label propagation that uses only labels available at scoring time, non-anticipative and leakage-free. We operationalize ATLAS with inductive GraphSAGE variants trained via neighbor sampling, at scale on a sessions graph with 100M+ nodes and ~1B edges. On a high-risk digital product at Capital One, ATLAS delivers +6.38% AUC and $> 50\%$ reduction in customer friction, improving fraud capture while reducing user friction.

1 Introduction

Account Takeover (ATO) fraud occurs when an adversary gains unauthorized access to a legitimate customer account via credential stuffing, phishing, device spoofing, or related tactics, and initiates high-risk transactions (HRTs). HRTs are monetizable actions (e.g., funds transfers) that are prime targets for fraudsters. In consumer banking, ATO drives both direct financial loss and customer-experience degradation. Therefore, effective defenses must increase fraud capture while minimizing friction for legitimate users.

A naïve approach is to impose friction on every HRT in every online session (e.g., additional verification). While simple, blanket friction creates a substantial customer burden, raising abandonment and complaints, and can erode trust and retention. We instead adopt a risk-based approach: train a model

*Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: New Perspectives in Advancing Graph Machine Learning.

to produce a real-time risk score per HRT online session and apply additional friction only to the high-risk subset. This reduces overall friction while maintaining high fraud recall, yielding a more favorable capture–friction trade-off.

Practically, ATO detection in banking faces extreme class imbalance, rapidly evolving attacker behavior (concept drift), and a strict online latency budget (e.g., < 250 ms). In this setting, production systems have long relied on tabular gradient-boosted decision trees, XGBoost [Chen and Guestrin, 2016], that score each session independently from engineered numerical features. Despite extensive trials with deep architectures (fully connected neural networks [Goodfellow et al., 2016], RNNs [Hochreiter and Schmidhuber, 1997], Transformers [Vaswani et al., 2017]), none consistently surpassed the boosted baseline under comparable latency and reliability constraints. Consequently, most gains have historically come from feature-engineering improvements atop XGBoost, which remains the dominant production solution.

Although XGBoost has established itself as the dominant in-production solution for tabular data [Shwartz-Ziv and Armon, 2022], it scores each session in isolation, effectively assuming independent and identically distributed (i.i.d.) observations. This flat, per-row view ignores the relational structure (entity linkages via account/device/IP) and temporal structure (causal ordering/recency) that characterize coordinated campaigns ("fraud rings"). As a result, the model cannot transfer risk across connected sessions and discards high-signal cues such as historical neighbor labels and temporal dependencies. Crucially, these shortcomings are intrinsic to the tabular formulation: collapsing sessions into independent rows prevents any per-session learner (XGBoost, MLP, or per-row Transformer) from representing time-respecting edges, path-based evidence sharing, or non-anticipative (serve-time) past fraud label availability; capturing these signals requires an explicit spatio-temporal structure representation.

To close this gap, we introduce **ATLAS** (**A**ccount **T**akeover **L**earning **A**cross **S**patio-Temporal Directed Graph), which reformulates ATO detection as spatio-temporal node classification on a time-respecting directed session graph. This framing enables time-respecting message passing across connected sessions and lag-aware (partially observed) label propagation. More specifically, our contributions are:

- We reformulate ATO as spatio-temporal graph learning with node classification over sessions on a time-respecting directed acyclic graph, enabling causal, time-respecting message passing and lag-aware label propagation that incorporates only serve-time-available historical evidence.
- We construct a directed temporal graph with strict causal ordering (past-to-future); link entities via shared identifiers (account, device, IP); and regulate connectivity with designed time-window and recency constraints. We operationalize this with an inductive GraphSAGE Hamilton et al. [2017] encoder trained via neighbor sampling, ensuring serve-time consistency and latency compliance.
- We conduct extensive experiments on a high-risk digital product dataset at Capital One financial institution (100M+ nodes, ~1B edges), demonstrating the effectiveness of our approach: +6.38% AUC and $> 50\%$ reduction in customer friction, improving fraud capture while lowering friction.

2 Method

We present ATLAS, a spatio-temporal directed-graph approach to ATO with three parts: (i) a time-respecting session graph with causal edges and connectivity controlled by window T and recency cap K (Section 2.1), (ii) serve-time–consistent (non-anticipative) label aggregation that appends lagged neighbor-label features (Section 2.2), and (iii) an inductive GraphSAGE encoder with mini-batch neighbor sampling and relational/time-aware/attention variants (Section 2.3). This yields leakage-free features and latency-compliant inference.

2.1 Graph Formulation Strategy

We model ATO as spatio-temporal node classification on a directed session graph $G = (V, E)$ depicted in Figure 1. Each node $v \in V$ is a high-risk transaction (HRT) session scored at serve time; edges $e \in E$ encode temporal, identifier-based links from prior sessions.

Nodes. Each node $v \in V$ is uniquely keyed by (account_id, device_id, ip_address, timestamp) and carries a feature vector $\mathbf{x}_v \in \mathbb{R}^d$ (curated tabular features) and a binary label $y_v \in \{0, 1\}$ (1 = fraud,

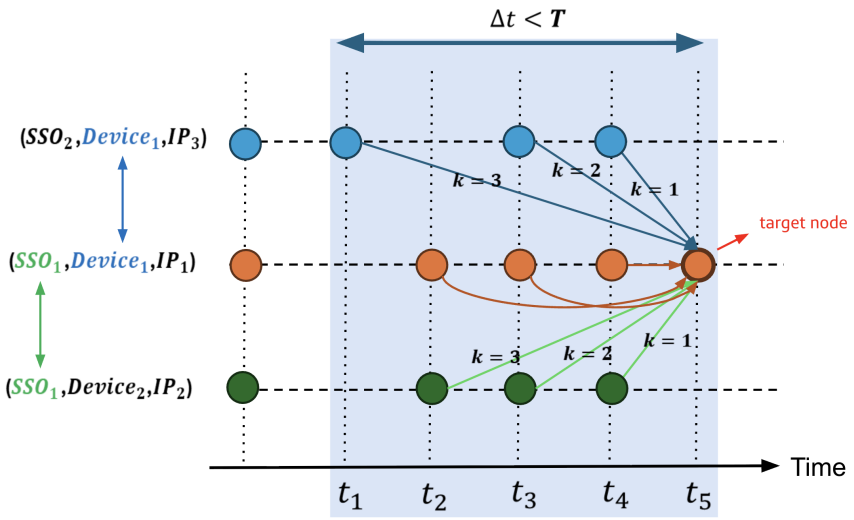

Figure 1: **ATLAS graph formulation.** Nodes are HRT sessions keyed by (account_id, device_id, ip_address, timestamp). Directed edges point past→future between sessions sharing an identifier, restricted by time window T and per-identifier recency cap K. Edge types correspond to the linking identifier (account/device/IP).

0 = non-fraud). The model outputs a risk score $s_v = \Pr\big(y_v = 1 \mid G_{\prec v},\, X_{\preceq v}\big) \in [0,1]$, where $G_{\prec v}$ is the subgraph of past neighbors of v (strictly earlier sessions) and $X_{\preceq v} = \{\mathbf{x}_u : u \in V,\, t_u \le t_v\}$, ensuring non-anticipative, leakage-free scoring.

Edges. For sessions u and v with timestamps $t_u < t_v$, we add a directed edge $(u \to v)$ if they share an identifier $m \in \mathcal{M} = \{\text{account\_id}, \text{device\_id}, \text{ip\_address}\}$. Edges are typed by m, yielding $E = \bigcup_{m\in\mathcal{M}} E_m$. Because $t_u < t_v$, the graph is time-respecting (acyclic).

Graph Connectivity Regulation. To ensure non-anticipative connectivity and control degree/latency, we enforce two connectivity constraints on graph edges :

- *Time-window* T: include $(u \to v)$ only if $0 < t_v - t_u \le T$.
- *Recency cap* K: for each v and type m, keep at most the K most recent predecessors in E_m.

These constraints (i) preserve causal ordering, (ii) prioritize informative recent history, and (iii) cap neighborhood size for stable sampling and serve-time latency.

Resulting Structure. The graph exposes cross-session patterns (e.g., coordinated "fraud rings") and supports causal, time-respecting message passing and serve-time–consistent lagged-label features (Section 2.2), as well different GraphSAGE variants with neighbor sampling at scale (Section . 2.3).

2.2 Lag-aware Label Propagation

To incorporate historical evidence without leakage, we augment each session node with labels from past, connected sessions whose ground truth is already known at the target's serve time. Let v denote the target session with serve time t_v. For any earlier session u with timestamp $t_u < t_v$, the binary label $y_u \in \{0,1\}$ becomes available at adjudication time τ_u. We therefore restrict propagation to labels that would be known at serve time by enforcing $\tau_u \le t_v$.

Starting from the directed session graph $G = (V, E)$ (Section 2.1), we collect the in-neighborhood of v within a finite history window $T > 0$:

$$\mathcal{N}_T^-(v) \;=\; \big\{\, u \;:\; (u \to v) \in E,\; 0 < t_v - t_u \le T \,\big\}. \tag{1}$$

To control degree and latency, we keep only the K most recent predecessors:

$$\mathcal{R}(v) \;=\; \underset{u \in \mathcal{N}_T^-(v)}{\text{TopK}}\,\big(t_u\big), \tag{2}$$

so that $|\mathcal{R}(v)| \le K$. We then apply the delayed-label filter to mirror what is available online:

$$\mathcal{A}(v) \;=\; \big\{\, u \in \mathcal{R}(v) \;:\; \tau_u \le t_v \,\big\}. \tag{3}$$

Here, T (e.g., days) bounds how far back we look, K caps neighborhood size for stable sampling and low latency, and the condition $\tau_u \le t_v$ ensures causal, serve-time–correct neighbor-label features.

From the available set $\mathcal{A}(v)$ we compute simple aggregates:

$$n_v^{\text{lab}} = \big|\mathcal{A}(v)\big| \qquad \text{(count of neighbors with known labels)} \tag{4}$$

$$n_v^{\text{fraud}} = \sum_{u \in \mathcal{A}(v)} y_u \qquad \text{(count of known-fraud neighbors)} \tag{5}$$

$$r_v = \frac{n_v^{\text{fraud}}}{\max\!\big(1,\, n_v^{\text{lab}}\big)} \qquad \text{(empirical fraud rate among known labels)} \tag{6}$$

$$a_v = \mathbb{1}\big[n_v^{\text{fraud}} \ge 1\big] \qquad \text{(any known fraud upstream)} \tag{7}$$

We then form a label-propagation feature vector

$$\ell_v \;=\; \big[\, n_v^{\text{lab}},\; n_v^{\text{fraud}},\; r_v,\; a_v \,\big] \tag{8}$$

and append it to the node input used by the encoder:

$$\mathbf{h}_v^{(0)} \;=\; \big[\, \mathbf{x}_v \,;\, \ell_v \,\big], \tag{9}$$

where $\mathbf{x}_v$ are the curated tabular features. During training, we apply the same T, K, and $\tau_u \le t_v$ rules to avoid training–serving skew. If no labels are available ($n_v^{\text{lab}} = 0$), we set $r_v = 0$ and $a_v = 0$. This design preserves strict causal ordering, aligns with delayed supervision in production, and provides the model with lightweight, high-signal context from truly known historical outcomes.

2.3 GNN Architecture

Our encoder is based on GraphSAGE [Hamilton et al., 2017], chosen for its inductive capability and support for mini-batch neighbor sampling, enabling training at our large scale. Each node begins with $\mathbf{h}_v^{(0)} = \big[\, \mathbf{x}_v \,;\, \ell_v \,\big]$ where $\mathbf{x}_v$ are curated tabular features and ℓ_v are lagged label features (Section 2.2).

Homogeneous GraphSAGE. We construct a time-respecting h-hop ego-graph around v using a neighbor sampler that enforces the same T and K constraints (Section 2.1). Let the per-hop fanouts be $\mathbf{f} = (f_1, \ldots, f_h)$ (at most f_k past neighbors sampled at hop k; when constant, $f_k \equiv f$ for all k). For layer $k = 1, \ldots, L$ (with $L \le h$), let $\mathcal{S}^{(k)}(v) \subseteq \mathcal{N}_T^-(v)$ denote the sampled in-neighbors used at layer k. A GraphSAGE block aggregates neighbor states and updates the target:

$$\mathbf{m}_v^{(k)} = \mathrm{AGG}^{(k)}\Big(\big\{\mathbf{h}_u^{(k-1)} : u \in \mathcal{S}^{(k)}(v)\big\}\Big), \tag{10}$$

$$\mathbf{h}_v^{(k)} = \sigma\Big(W^{(k)}\big[\mathbf{h}_v^{(k-1)} \,;\, \mathbf{m}_v^{(k)}\big] + \mathbf{b}^{(k)}\Big), \tag{11}$$

optionally followed by ℓ_2-normalization and dropout. We use *mean* as the aggregator $\mathrm{AGG}^{(k)}$.

Relational GraphSAGE. As edges are typed by identifier (account/device/IP), we use a relational variant that aggregates per type and then fuses the results. Let $\mathcal{M}$ be the set of relation types and let $\mathcal{S}_m^{(k)}(v) \subseteq \mathcal{N}_{T,m}^-(v)$ denote the sampled in-neighbors of type m used at layer k. We compute

$$\mathbf{m}_v^{(k)} \;=\; \sum_{m \in \mathcal{M}} \Phi_m^{(k)}\Big(\mathrm{AGG}_m^{(k)}\Big(\big\{\mathbf{h}_u^{(k-1)} \;:\; u \in \mathcal{S}_m^{(k)}(v)\big\}\Big)\Big), \tag{12}$$

where $\mathrm{AGG}_m^{(k)}$ is a type-specific aggregator (we use *mean*) and $\Phi_m^{(k)}$ is a learnable type-specific transform (e.g., a linear map or gate). The node update then follows equation 11.

Time-aware / attention variant. To encode recency and relation importance, we incorporate simple edge features (e.g., binned $\Delta t = t_v - t_u$ and edge-type embeddings) either by concatenation into the neighbor vector or via an attention aggregator:

$$\alpha_{vu}^{(k)} = \mathrm{softmax}_{u \in \mathcal{S}^{(k)}(v)} \Big(\mathbf{a}^\top \big[W_{\mathrm{qry}} \mathbf{h}_v^{(k-1)} \,;\, W_{\mathrm{key}} \mathbf{h}_u^{(k-1)} \,;\, \mathbf{e}_{uv} \big] \Big), \tag{13}$$

$$\mathbf{m}_v^{(k)} = \sum_{u \in \mathcal{S}^{(k)}(v)} \alpha_{vu}^{(k)} \, W_{\mathrm{val}} \mathbf{h}_u^{(k-1)}, \tag{14}$$

where $\mathcal{S}^{(k)}(v) \subseteq \mathcal{N}_T^-(v)$ is the sampled in-neighborhood at layer k, $\mathbf{e}_{uv}$ encodes the (binned) time gap and relation type for edge $(u \to v)$, and $W_{\mathrm{qry}}, W_{\mathrm{key}}, W_{\mathrm{val}}$ and $\mathbf{a}$ are learnable parameters. Optionally, a multi-head variant can replace (13)–(14) with head-wise projections and concatenation.

Neighbor sampling and depth. We train with mini-batches of seed nodes and per-layer fanouts $(f_1, \ldots, f_L)$, sampling $\mathcal{S}^{(k)}(v) \subseteq \mathcal{N}_T^-(v)$ under the same (T, K) constraints used at serve time to avoid train–serve skew. In practice, shallow depth ($L \in \{2, 3\}$) with moderate fanouts provides a good accuracy–latency trade-off.

Output and loss. The final embedding $\mathbf{h}_v^{(L)}$ is passed to a logistic head

$$s_v = \sigma(\mathbf{w}^\top \mathbf{h}_v^{(L)} + b). \tag{15}$$

We optimize a weighted binary cross-entropy to address class imbalance. Decision thresholds are calibrated to the target friction envelope.

3 Results

We compare our GNNs to the production XGBoost baseline and ablate graph hyperparameters. Beyond a simple homogeneous GraphSAGE, added architectural complexity yields little benefit; most gains come from the graph formulation and serve-time–consistent lagged labels. Prior trials with FNNs and tabular Transformers matched XGBoost under the same features/latency, reinforcing that improvements stem from exploiting graph structure rather than deeper per-row models.

3.1 Dataset

The dataset comprises tens of millions of sessions with a very low ATO base rate (extreme class imbalance). To evaluate generalization to future traffic, we use a chronological split: 8 months for training, 2 months for validation, and 5 months for testing (no overlap). All numerical features are standardized using statistics computed on the training set only. We assemble features, labels, and edge indices into PyTorch Geometric (PyG) data objects and employ PyG's NeighborLoader for efficient, out-of-core neighborhood sampling. This dynamic loading is essential for a continuously growing graph. Owing to data sensitivity and confidentiality, we do not report descriptive statistics. The corpus contains two major segments from a digital product at Capital One; due to confidentiality, we anonymize them as *Segment 1* and *Segment 2*.

3.2 Performance Comparison

The GNN consistently outperforms XGBoost for both segments. As shown in Table 1, the GNN achieves an overall ROC AUC of 82.27 (vs. 79.83 for XGBoost), a *+3.06%* relative improvement. By segment, gains are *+3.43%* (Segment 1) and *+1.66%* (Segment 2). The strongest results come from homogeneous GraphSAGE *with* label propagation, yielding an overall ROC AUC of 84.46 and a *+5.8%* relative improvement over XGBoost.

3.3 Hyperparameter Analysis (K and T)

We study the recency cap K and time window T. As illustrated in Figure 2, increasing K from 1 to 10 steadily improves ROC AUC, indicating that incorporating more *recent* historical sessions is beneficial. Likewise, extending T from 1 to 120 days yields consistent gains, underscoring the value of a longer temporal context for detecting coordinated activity.

Table 1: Performance comparison of XGBoost (XGB) vs. GNN, with and without label propagation. (ROC AUC reported as percentages; improvements are relative to XGB.)

Model	AUC Overall	AUC Segment 1	AUC Segment 2
XGB	79.83	78.88	82.45
GNN	82.27	81.59	83.82
Improvement (vs. XGB)	*+3.06%*	*+3.43%*	*+1.66%*
GNN + Label Propagation	84.46	83.92	85.45
Improvement (vs. XGB)	*+5.8%*	*+6.38%*	*+3.63%*

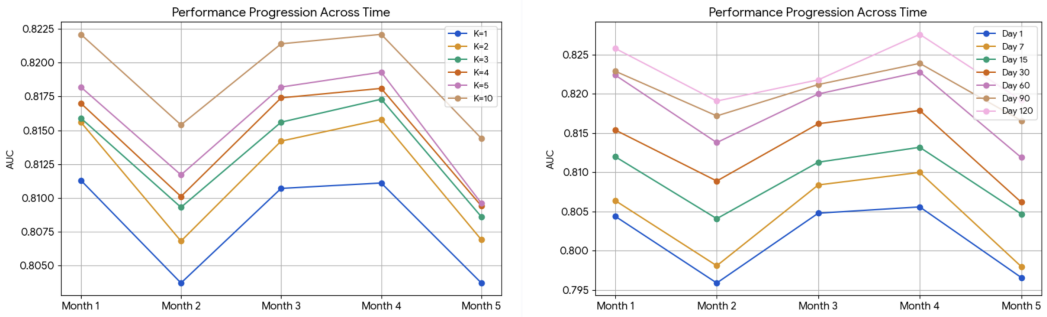

Figure 2: Effect of K (left) and T (right) on ROC AUC for Segment 1; similar trends hold for Segment 2.

4 Related Work

ATO detection in industry commonly relies on tabular gradient-boosted trees such as XGBoost [Chen and Guestrin, 2016]; tree ensembles often remain strong on structured data [Shwartz-Ziv and Armon, 2022], but they struggle to capture cross-session relational patterns.

Foundational GNNs include GCNs for node classification [Kipf and Welling, 2017], inductive GraphSAGE with neighbor sampling [Hamilton et al., 2017], and attention-based GAT [Veličković et al., 2018, Kerdabadi et al., 2025, Hadizadeh Moghaddam et al., 2025]. For evolving interactions, temporal models such as Temporal Graph Networks (TGN) [Rossi et al., 2020], TGAT [Xu et al., 2020], and DySAT [Sankar et al., 2020] incorporate time into message passing. Our work differs by enforcing a time-respecting DAG and serve-time constraints to achieve non-anticipative inference under strict latency at enterprise scale.

Recent graph-based approaches to transactional fraud explicitly model accounts, devices, and transactions as graphs with temporal or entity-sharing edges. Semi-supervised credit-card fraud detection via attribute-driven graph representations treats users/transactions as nodes and propagates attribute signals [Xiang et al., 2023]. For edge-level scoring, FraudGT applies a graph transformer with edge-aware attention and message gating [Lin et al., 2024]. Under sparse labels, Barely Supervised learning introduces structure-aware contrastive objectives [Yu et al., 2024]. On heterogeneous graphs, DGA-GNN combats noisy neighborhoods via dynamic grouping [Duan et al., 2024]. Low-homophily settings motivate label-aware aggregation and transformer encoders [Wang et al., 2023].

5 Conclusion

We presented ATLAS, a spatio-temporal graph framework for ATO detection that operates on a time-respecting directed session graph with connectivity regulated by a time window and recency cap. By combining serve-time-consistent lagged label aggregation with inductive GraphSAGE variants and neighbor sampling, ATLAS scales to 100M+ nodes and ~1B edges while remaining latency compliant. On a high-risk digital product, it yields +6.38% AUC and $>50\%$ reduction in customer friction, improving fraud capture and user experience. Due to privacy and regulatory constraints at Capital One, we cannot release data or detailed dataset statistics, and our evaluation is limited to anonymized segments of an online digital product.

References

T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, 2016.

M. Duan, T. Zheng, Y. Gao, G. Wang, Z. Feng, and X. Wang. Dga-gnn: Dynamic grouping aggregation gnn for fraud detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 11820–11828, 2024.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

A. Hadizadeh Moghaddam, M. Nayebi Kerdabadi, B. Liu, M. Liu, and Z. Yao. Discovering time-aware hidden dependencies with personalized graphical structure in electronic health records. *ACM Transactions on Knowledge Discovery from Data*, 19(2):1–21, 2025.

W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

M. N. Kerdabadi, A. H. Moghaddam, D. Wang, and Z. Yao. Multi-ontology integration with dual-axis propagation for medical concept representation. *arXiv preprint arXiv:2508.21320*, 2025.

T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

J. Lin, X. Guo, Y. Zhu, S. Mitchell, E. Altman, and J. Shun. Fraudgt: A simple, effective, and efficient graph transformer for financial fraud detection. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pages 292–300, 2024.

E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein. Temporal graph networks for deep learning on dynamic graphs. *arXiv preprint arXiv:2006.10637*, 2020.

A. Sankar, Y. Wu, L. Gou, W. Zhang, and H. Yang. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In *Proceedings of the 13th international conference on web search and data mining*, pages 519–527, 2020.

R. Shwartz-Ziv and A. Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Y. Wang, J. Zhang, Z. Huang, W. Li, S. Feng, Z. Ma, Y. Sun, D. Yu, F. Dong, J. Jin, et al. Label information enhanced fraud detection against low homophily in graphs. In *Proceedings of the ACM Web Conference 2023*, pages 406–416, 2023.

S. Xiang, M. Zhu, D. Cheng, E. Li, R. Zhao, Y. Ouyang, L. Chen, and Y. Zheng. Semi-supervised credit card fraud detection via attribute-driven graph representation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 14557–14565, 2023.

D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, and K. Achan. Inductive representation learning on temporal graphs. *arXiv preprint arXiv:2002.07962*, 2020.

H. Yu, Z. Liu, and X. Luo. Barely supervised learning for graph-based fraud detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 16548–16557, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer:[Yes]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We specify the graph formulation (node keys, edge types, time-respecting directionality), the connectivity rules (time window T, recency cap K), the lag-aware label-aggregation procedure (sets $\mathcal{N}^-(v), \mathcal{R}(v), \mathcal{A}(v)$ and feature vector ℓ_v), the encoder family (GraphSAGE variants), neighbor-sampling scheme (hops and per-hop fanouts), loss (weighted BCE), threshold calibration procedure (friction envelope), evaluation protocol (chronological 8/2/5-month split, train-only standardization, segment-wise reporting), and ablations over T and K. These details are sufficient for independent groups to reproduce the method and claims on comparable datasets.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility.

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The production dataset consists of sensitive banking sessions subject to privacy and regulatory constraints, so we cannot release data or detailed statistics. For the same reason, we do not release the full training code tied to internal data pipelines.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: we did not report it because the variation observed were very insignificant across different seeds.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: We used AWS servers, but more detailes are confidential

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: [NA]

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: We do not release data or trained models due to institutional privacy and regulatory constraints; therefore, release-specific safeguards are not applicable.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: No new public assets (datasets, models, or code) are released

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The work does not involve crowdsourcing or human-subject experiments.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:No human-subject studies were conducted; all analyses used de-identified operational data under institutional governance.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLMs are not part of the core methodology; any language tooling was limited to writing/editing and does not affect scientific results.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.