MULTI-IDENTITY TRANSACTION GRAPH PRETRAINING FOR FRAUD DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Real-world payment risk modeling is constrained by scarce fraud labels and fragmented behavioral context across multiple payment identities. A physical transaction may appear simultaneously in multiple behavioral histories defined by different identities, while each identity induces distinct temporal dependencies. Existing sequence models typically encode these histories independently, duplicating the same physical event across trajectories and leaving cross-identity dependencies implicit. We propose Multi-Identity Transaction Graph Pretraining (MITraG), a self-supervised framework that organizes overlapping identity-conditioned transaction histories as a shared-event, multi-relational temporal graph. Each transaction is represented once as a node, while edges preserve identity-specific temporal relations and irregular time gaps. MITraG further uses Masked Transaction Node Reconstruction (MTNR) to learn contextual transaction representations from unlabeled histories before supervised adaptation to fraud detection. On real-world payment data under a strict out-of-time evaluation protocol, MITraG consistently outperforms transaction-level, sequence-pretraining, and temporal-graph baselines. These results suggest that modeling transactions as shared events connected by identity-specific temporal relations provides a general basis for self-supervised payment behavior representation learning.

1 INTRODUCTION

Real-world payment risk modeling faces two related challenges. Confirmed fraud labels are scarce relative to the volume of observed transactions, while the behavioral context needed for risk assessment is distributed across multiple payment identities. A transaction that appears ordinary in isolation may become suspicious in the context of recent behavior, yet learning such context directly from fraud supervision is difficult when positive labels are rare. Effective risk modeling therefore requires behavioral representations that can be learned from largely unlabeled transaction histories while preserving evidence distributed across different identity-conditioned views.

These views are not independent. A physical transaction may appear in multiple histories, with each identity defining distinct temporal dependencies. Sequence-based approaches typically encode these histories separately, causing the same physical event to be represented multiple times while leaving the points at which the histories overlap implicit. The problem therefore goes beyond how multiple sequence representations are fused; it also concerns how overlapping behavioral histories are represented in the first place.

Existing self-supervised transaction models learn from unlabeled event sequences Padhi et al. (2021); Babaev et al. (2022); Chen et al. (2022), while temporal graph methods capture relational and temporal dependencies among transactions Lu et al. (2022); Xiang et al. (2023). These two lines of work address complementary aspects of the problem but rely on different structural assumptions. Sequence pretraining typically treats an identity-conditioned trajectory as the basic representation unit, whereas transaction-graph methods usually operate on a predefined relational topology. Neither explicitly represents overlap among multiple identity-conditioned histories as part of the behavioral structure.

We propose Multi-Identity Transaction Graph Pretraining (MITraG), a self-supervised framework for learning from overlapping identity-conditioned transaction histories. Instead of maintaining separate representations of the same physical transaction across different histories, MITraG represents

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.

each transaction once as a shared event, while identity-specific temporal relations connect these events into a transaction-centric, multi-relational temporal graph. Masked Transaction Node Reconstruction (MTNR) learns contextual transaction representations from unlabeled histories, and Target & Graph Interaction Fusion (TGIF) adapts the learned representation for downstream fraud detection. Our main experiments use account- and card-conditioned histories, while the formulation also accommodates additional identity views.

We evaluate MITraG on industrial payment data under a strict out-of-time (OOT) protocol. Averaged over three OOT partitions, MITraG achieves a ROC-AUC of 0.8174 and Recall@1% of 0.1360, outperforming the evaluated transaction-level, sequence-pretraining, and temporal-graph baselines. Approximately 79% of the evaluated samples contain cross-identity historical overlap, and the benefit of shared-transaction modeling is concentrated in samples where such overlap exists. Extending the same formulation with an additional identity view further improves OOT performance. Together, these results support a shared-event formulation of multi-identity transaction behavior, self-supervised learning of this representation through MTNR, and its effectiveness under temporal distribution shift.

2 Related Work

2.1 Transaction Sequence Representation and Pretraining

Transaction representation learning has evolved from field-aware encoding to self-supervised event-sequence modeling. The TabFormer framework includes Tabular BERT (TabBERT), a representation-learning model that hierarchically encodes heterogeneous transaction fields and transaction sequences through masked pretraining Padhi et al. (2021), while CoLES learns sequence representations by contrasting subsequences from the same history Babaev et al. (2022). CASPR applies masked entity recovery to customer activity sequences and transfers the learned representations across enterprise tasks Chen et al. (2022).

Later work extends transaction pretraining to fraud detection and large-scale behavioral modeling. UB-PTM introduces self-supervised objectives at multiple behavioral granularities Liu et al. (2022), SAGE models structured e-commerce behavior sequences Wang et al. (2023), and the Foundation Purchasing Model demonstrates large-scale generative pretraining and cross-domain transfer Skalski et al. (2023). Recent industrial systems also incorporate heterogeneous attributes, periodic patterns, payment-network signals, and multi-source banking events Braithwaite et al. (2025); Li et al. (2025); Yeh et al. (2026); Ostroukhov et al. (2026). These studies show that behavioral representations can be effectively pretrained from unlabeled financial activity.

Despite richer objectives and inputs, these methods remain largely sequence-centric: overlapping identity-conditioned histories are encoded as separate trajectories, leaving shared physical events implicit. MITraG instead treats these shared events as part of the representation itself, moving from independent sequences to an overlapping multi-identity temporal topology.

2.2 Temporal and Multi-Relational Graph Learning for Fraud Detection

Graph-based fraud detection uses relational information beyond local transaction features. Early methods learn representations from financial interaction networks Khazane et al. (2019), while multi-relational models such as CARE-GNN handle relation-dependent and potentially deceptive neighborhoods Dou et al. (2020). Industrial systems such as xFraud further show that heterogeneous transaction graphs can support scalable risk prediction Rao et al. (2022). These methods demonstrate the value of relational structure, but typically rely on predefined global networks and fraud-specific supervision.

Temporal graph methods further control how information propagates over these structures. BRIGHT uses directed graph construction to separate historical propagation from real-time inference Lu et al. (2022), while GTAN represents transactions as temporal graph nodes and applies gated temporal attention Xiang et al. (2023). DCL-GFD introduces node–subgraph contrastive learning for heterophilic and highly imbalanced fraud graphs Yu et al. (2025). Other approaches combine sequence and graph modeling, as in TRUST Y et al. (2026), or pretrain representations on an existing transaction graph, as in IGT4ETH Liu et al. (2026). In these methods, however, the graph topology

is generally predefined rather than constructed from overlap among multiple identity-conditioned histories.

MITraG differs primarily in how the graph topology is obtained. Rather than starting from a predefined network, it derives the graph from overlap among identity-conditioned transaction histories and pretrains on the resulting shared-event temporal structure. Shared transactions connect identity-specific temporal paths while preserving their relation semantics.

3 Problem Formulation and Method Overview

For a target transaction i, let $\mathcal{R}_i$ denote the set of available identity views and

$$\mathcal{S}_i = \{S_i^r : r \in \mathcal{R}_i\}$$

the corresponding transaction histories. Each S_i^r is chronologically ordered and ends at i. An occurrence $s_j^r \in S_i^r$ contains the transaction attributes available at decision time, its timestamp, and a physical-transaction identifier used only to align occurrences across histories. The histories need not be disjoint: occurrences in different views may correspond to the same underlying transaction.

Our goal is to learn a transaction-context representation that preserves both this cross-identity overlap and the temporal dependencies induced by individual identities. We therefore construct

$$G_i = \mathcal{B}(\mathcal{S}_i), \qquad g_i = \mathrm{Enc}_\theta(G_i),$$

where $\mathcal{B}$ maps occurrences of the same physical transaction to a shared node while retaining the temporal relations contributed by each identity view. Thus, physical transactions form the representation units, whereas identities determine how these events are connected over time.

Because confirmed fraud labels are sparse relative to the available transaction histories, the graph encoder is first pretrained without fraud supervision. MITraG uses Masked Transaction Node Reconstruction (MTNR) to recover masked transaction attributes from the remaining transaction content and multi-identity graph context. The resulting parameters initialize downstream fraud detection. Let x_i denote the engineered features of the target transaction and $y_i \in \{0, 1\}$ its fraud label. During supervised fine-tuning,

$$\hat{y}_i = h_\psi(x_i, \mathrm{Enc}_\theta(G_i)),$$

where the graph encoder and downstream prediction components are optimized jointly.

Figure 1 provides an overview of the complete MITraG pipeline. The following section details the multi-identity graph construction, transaction encoding, temporal graph encoder, MTNR pretraining, and Target & Graph Interaction Fusion (TGIF) used for downstream prediction. Our main experiments instantiate $\mathcal{R}_i$ with account and card views, while the same formulation can incorporate additional identity views without changing the underlying representation principle.

4 Method

4.1 Multi-Identity Transaction Graph Construction

For clarity, we describe the account–card instantiation used in our main experiments. The construction follows the general formulation in Section 3 and extends to additional identity views through the corresponding identity-specific temporal relations. For each target transaction i, we construct a local graph from its account-conditioned history S_i^a and card-conditioned history S_i^c, each comprising up to L most recent transactions ending at i. An occurrence in identity view $r \in \{a, c\}$ is denoted by

$$s_j^r = \left(o_j^r, t_j^r, \mathbf{a}_j^r\right), \tag{1}$$

where o_j^r is the physical-transaction identifier, t_j^r is the timestamp, and $\mathbf{a}_j^r$ contains the transaction features. The identifier is used only for occurrence alignment and is never exposed as a learnable input.

The key construction is to preserve overlap between the two identity-conditioned histories rather than encode them as independent sequences. Let π perform this alignment by mapping occurrences of the same physical transaction to a single transaction node. We define

$$\begin{aligned} V_i^a &= \{\pi(s) : s \in S_i^a\}, \quad V_i^c = \{\pi(s) : s \in S_i^c\}, \\ V_i &= V_i^a \cup V_i^c. \end{aligned} \tag{2}$$

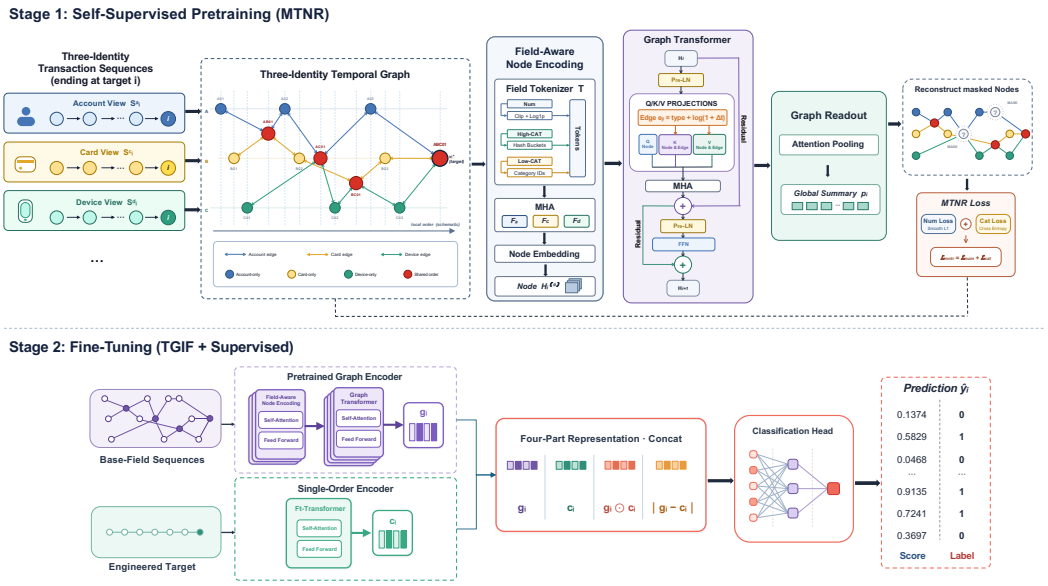

Figure 1: Overview of the MITraG framework.

with the shared-node set

$$V_i^{\text{shared}} = V_i^a \cap V_i^c. \tag{3}$$

Thus, each $v \in V_i^{\text{shared}}$ represents one physical transaction observed in both histories and forms a structural junction between their temporal paths. Shared nodes may occur throughout the historical context rather than only at the target transaction, allowing the graph to explicitly represent where account- and card-conditioned behavior overlaps.

For temporally consecutive occurrences s_j^r and s_k^r in view r, we add a directed temporal edge

$$\begin{aligned} e_{jk}^r &= \left(\pi(s_j^r), \pi(s_k^r), r, \Delta t_{jk}\right), \\ \Delta t_{jk} &= t_k^r - t_j^r \geq 0. \end{aligned} \tag{4}$$

Account- and card-induced edges retain distinct relation types; if the same node pair is consecutive in both histories, the two relations are preserved as parallel edges. We additionally introduce a reverse edge type for each forward relation type and a self-loop type for message passing, yielding five computation relation types: account-forward, card-forward, account-reverse, card-reverse, and self-loop. Reverse edges inherit the time-gap magnitude of their corresponding forward edges, while self-loops use zero time gap. The target transaction is designated as the anchor node $v_i^\star$ and forms the terminal node of each available forward identity path. Since all transactions in G_i occur no later than the target transaction, the added reverse computation edges do not introduce post-decision information.

4.2 Field-Aware Transaction Node Encoding

Each transaction occurrence contains d_n numeric and d_c categorical transaction features, distinct from the engineered target-transaction features $\mathbf{x}_i$. A shared field tokenizer T maps these heterogeneous features to typed token embeddings, after which view-specific field Transformers encode intra-transaction feature interactions:

$$\mathbf{z}_v^r = W_r F_r(T(\mathbf{a}_v^r))_{\text{CLS}}, \qquad r \in \{a, c\}. \tag{5}$$

The tokenizer is shared across the account and card views to maintain a common field-embedding space, while F_a and F_c and their projection layers are separately parameterized before node-level fusion.

Because a transaction node may occur in either or both identity-conditioned histories, let $m_v^r \in \{0, 1\}$ indicate whether node v is present in view r. We initialize each unique transaction node as

$$\mathbf{h}_v^{(0)} = \mathrm{MLP}_{\mathrm{node}}([m_v^a \mathbf{z}_v^a;\, m_v^c \mathbf{z}_v^c;\, \mathbf{e}_{\mathrm{pres}}(m_v^a, m_v^c)])\,, \tag{6}$$

where absent views contribute zero vectors and $\mathbf{e}_{\mathrm{pres}}$ is a learned embedding of view availability. Thus, occurrences of a shared physical transaction contribute to a single node state while retaining account- and card-specific field transformations. The resulting $\mathbf{h}_v^{(0)}$ is then contextualized across transactions by the multi-relational temporal graph encoder.

4.3 Multi-Relational Temporal Graph Encoding

MITraG propagates transaction context only along the graph edges defined in Section 4.1. For a computation edge $e = (u, v, r_e, \Delta t_e)$, we encode its relation type and irregular time interval as

$$\mathbf{e}_e = E_{\mathrm{type}}(r_e) + \mathrm{MLP}_t(\log(1 + \Delta t_e))\,. \tag{7}$$

At graph layer ℓ, edge-conditioned multi-head attention operates over the incoming edge instances of each node. Let $\mathcal{E}_i^-(v)$ denote the incoming edges of v. Omitting head indices for clarity,

$$\begin{aligned}
\mathbf{q}_v &= W_Q \bar{\mathbf{h}}_v^{(\ell)}, \\
\mathbf{k}_e &= W_K \bar{\mathbf{h}}_u^{(\ell)} + W_K^e \mathbf{e}_e, \\
\boldsymbol{\nu}_e &= W_V \bar{\mathbf{h}}_u^{(\ell)} + W_V^e \mathbf{e}_e, \\
\alpha_e &= \frac{\exp\big(\mathbf{q}_v^\top \mathbf{k}_e / \sqrt{d_h}\big)}{\sum_{e' \in \mathcal{E}_i^-(v)} \exp\big(\mathbf{q}_v^\top \mathbf{k}_{e'} / \sqrt{d_h}\big)}, \\
\mathbf{m}_v^{(\ell)} &= \sum_{e \in \mathcal{E}_i^-(v)} \alpha_e \boldsymbol{\nu}_e,
\end{aligned} \tag{8}$$

where $\bar{\mathbf{h}}_v^{(\ell)} = \mathrm{LN}(\mathbf{h}_v^{(\ell)})$. Defining attention over edge instances preserves parallel account- and card-induced relations between the same pair of transaction nodes. The attention and feed-forward outputs are incorporated through scaled residual connections:

$$\begin{aligned}
\widetilde{\mathbf{h}}_v^{(\ell)} &= \mathbf{h}_v^{(\ell)} + \gamma_a W_O \mathbf{m}_v^{(\ell)}, \\
\mathbf{h}_v^{(\ell+1)} &= \widetilde{\mathbf{h}}_v^{(\ell)} + \gamma_f \,\mathrm{FFN}\Big(\mathrm{LN}(\widetilde{\mathbf{h}}_v^{(\ell)})\Big)\,.
\end{aligned} \tag{9}$$

Shared transaction nodes provide the interaction points between identity-conditioned histories. At any $v \in V_i^{\mathrm{shared}}$, messages induced by account- and card-specific relations are aggregated into the same hidden state, allowing evidence from the two temporal paths to interact at the corresponding physical transaction. This jointly contextualized state can subsequently propagate along either path. Each layer communicates only through explicit one-hop graph edges; stacking K layers therefore expands the receptive field to at most K hops without introducing dense all-pairs attention.

After K layers, we combine the contextualized target-anchor state with an attention-pooled graph summary:

$$\begin{aligned}
\beta_v &= \frac{\exp\Big(\mathbf{w}^\top \tanh(W_p \mathbf{h}_v^{(K)})\Big)}{\sum_{u \in V_i} \exp\Big(\mathbf{w}^\top \tanh(W_p \mathbf{h}_u^{(K)})\Big)}, \\
\mathbf{p}_i &= \sum_{v \in V_i} \beta_v \mathbf{h}_v^{(K)}, \\
\mathbf{g}_i &= \mathrm{MLP}_g\Big([\mathbf{h}_{v_i^\star}^{(K)}; \mathbf{p}_i]\Big)\,.
\end{aligned} \tag{10}$$

Here, $\mathbf{h}_{v_i^\star}^{(K)}$ captures the target transaction after message passing that incorporates relation types and time gaps, while $\mathbf{p}_i$ summarizes complementary context from the local graph. Their combination $\mathbf{g}_i$ is the graph-contextualized target representation used by MTNR and downstream adaptation.

4.4 MASKED TRANSACTION NODE RECONSTRUCTION

MITraG is pretrained with Masked Transaction Node Reconstruction (MTNR) using unlabeled transaction histories. For each graph, every unique transaction node is independently selected with probability ρ; if no node is selected, one node is sampled uniformly. Let

$$\begin{aligned} m_v &\sim \mathrm{Bernoulli}(\rho), \\ \mathcal{M}_i &= \{v \in V_i : m_v = 1\}. \end{aligned} \tag{11}$$

For a selected node, MTNR masks its numeric and categorical transaction features while preserving the node itself, the graph topology, relation types, and time gaps. The same node-level mask is applied across the account and card views:

$$\widetilde{\mathbf{a}}_v^r = \mathrm{Mask}(\mathbf{a}_v^r; m_v), \qquad r \in \{a, c\}. \tag{12}$$

Consequently, when a shared transaction is masked, its occurrences in both identity views are hidden simultaneously. This prevents reconstruction by directly copying the same physical transaction from the other view and requires the encoder to exploit the remaining transaction semantics and multi-identity graph context. The target node follows the same masking rule as every other transaction node.

For each masked occurrence, a view-specific decoder reconstructs the original transaction features from the contextualized node state and graph representation,

$$\widehat{\mathbf{a}}_v^r = D_r\Big([\mathbf{h}_v^{(K)}; \mathbf{g}_i]\Big), \qquad r \in \{a, c\}, \tag{13}$$

with feature-specific output heads for the reconstructed attributes. Numeric features are optimized with Smooth L1 loss, while selected low-cardinality categorical features use cross-entropy. Let Ω_n^r and Ω_c^r denote the corresponding masked numeric and categorical targets in view r. We define

$$\begin{aligned} \mathcal{L}_n &= \frac{1}{\sum_{r\in\{a,c\}} |\Omega_n^r|} \sum_{r\in\{a,c\}} \sum_{(v,k)\in\Omega_n^r} \mathrm{SmoothL1}\big(\widehat{\mathbf{a}}_{v,k}^r, \mathbf{a}_{v,k}^r\big), \\ \mathcal{L}_c &= \frac{1}{\sum_{r\in\{a,c\}} |\Omega_c^r|} \sum_{r\in\{a,c\}} \sum_{(v,k)\in\Omega_c^r} \mathrm{CE}\big(\widehat{\mathbf{a}}_{v,k}^r, \mathbf{a}_{v,k}^r\big), \\ \mathcal{L}_{\mathrm{MTNR}} &= \mathcal{L}_n + \mathcal{L}_c. \end{aligned} \tag{14}$$

High-cardinality categorical features retained as encoder inputs are excluded from reconstruction, as exact prediction over large long-tailed vocabularies can encourage identifier memorization and impose substantial output cost. By reconstructing masked transaction attributes from the surrounding shared-event topology, MTNR learns contextual representations of transaction nodes without fraud supervision. After pretraining, the reconstruction decoders are discarded and the graph encoder is transferred to downstream fraud detection.

4.5 TARGET & GRAPH INTERACTION FUSION AND FINE-TUNING

For downstream fraud detection, we combine the pretrained graph-contextualized target representation with the manually engineered features of the target transaction. A target-transaction encoder E_x processes $\mathbf{x}_i$, while the graph encoder produces $\mathbf{g}_i$ from G_i. The two representations are projected to a common dimension:

$$\mathbf{c}_i = P_x E_x(\mathbf{x}_i), \qquad \widetilde{\mathbf{g}}_i = P_g \mathbf{g}_i. \tag{15}$$

Target & Graph Interaction Fusion (TGIF) augments direct concatenation with element-wise interaction and discrepancy terms:

$$\mathbf{u}_i = [\mathbf{c}_i; \widetilde{\mathbf{g}}_i; \mathbf{c}_i \odot \widetilde{\mathbf{g}}_i; |\mathbf{c}_i - \widetilde{\mathbf{g}}_i|]. \tag{16}$$

Here, the element-wise product captures cross-branch interactions, while the absolute difference represents discrepancies between engineered target evidence and its graph-contextualized representation. The fused representation is normalized and passed to the fraud-classification head:

$$\begin{aligned} \widehat{\mathbf{y}}_i &= \mathrm{MLP}_{\mathrm{cls}}(\mathrm{LN}(\mathbf{u}_i)), \\ \mathcal{L}_{\mathrm{sup}} &= \mathrm{CE}(\widehat{\mathbf{y}}_i, y_i). \end{aligned} \tag{17}$$

During supervised fine-tuning, the pretrained graph encoder, target-transaction encoder, projection layers, and classification head are optimized jointly under $\mathcal{L}_{\mathrm{sup}}$, while the MTNR reconstruction decoders are discarded.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Dataset and temporal protocol. We evaluate MITraG on a large-scale industrial payment dataset containing tens of millions of transactions, with a fraud rate below 1%. Graph pretraining uses July–October 2025 for optimization and November 2025 for checkpoint selection. Supervised training uses July–November 2025, validation uses December 2025, and the three OOT test partitions cover January, February, and March 2026. All checkpoints are selected using validation data before OOT evaluation. Sample and physical-transaction identifiers are disjoint across supervised training, validation, and test partitions, while account and card identities may recur over time. Additional details on data splits and leakage control are provided in Appendix A.

Baselines and metrics. We compare MITraG with transaction-level, sequence-pretraining, and temporal-graph baselines. FT-Transformer Gorishniy et al. (2021) and LightGBM Ke et al. (2017) use only engineered features of the target transaction. In our dual-sequence adaptation of TabBERT Padhi et al. (2021), separate encoders process the account- and card-conditioned histories through masked sequence pretraining, and their representations are combined with the target-transaction representation only during downstream fraud detection. GTAN Xiang et al. (2023) serves as the temporal-graph baseline and does not use historical or target fraud labels as node inputs. All neural models follow the same temporal protocol and select checkpoints using validation data only. Unless otherwise stated, all reported model-performance values are means over five independent training seeds. For each seed, the final test score is defined as the unweighted mean across the three OOT partitions.

We report ROC-AUC and Recall@1% as the primary metrics. PR-AUC, Recall@10%, Recall-CAP, Top-10 Area, partition-specific Recall@1%, and statistical tests are reported in Appendix C.

Implementation details. Each identity-conditioned history contains at most $L = 35$ transactions. The graph encoder uses $K = 4$ layers, and the MTNR mask ratio is $\rho = 0.15$. Both pretraining and fine-tuning use AdamW, with pretraining performed on one NVIDIA B300. All ablations follow the same fine-tuning and model-selection protocol as MITraG; complete settings are provided in Appendix B.

5.2 MAIN RESULTS

Table 1: Results on the three OOT test partitions. TabBERT denotes the dual-sequence adaptation described in Appendix B.3.

	ROC-AUC ↑				Recall@1% ↑
Model	Jan.	Feb.	Mar.	Avg.	Avg.
FT-Transformer Gorishniy et al. (2021)	0.7712	0.7761	0.7538	0.7670	0.1115
LightGBM Ke et al. (2017)	0.7827	0.7908	0.8071	0.7935	0.1119
TabBERT Padhi et al. (2021)	0.7747	0.7683	0.7438	0.7623	0.0824
GTAN Xiang et al. (2023)	0.7887	0.7818	0.7866	0.7857	0.1207
MITraG	**0.8200**	**0.8162**	**0.8161**	**0.8174**	**0.1360**

Table 1 shows that MITraG achieves the highest ROC-AUC in each OOT partition and on average, while also obtaining the highest mean Recall@1%. The dual-sequence TabBERT adaptation provides the closest sequence-pretraining comparison: it uses the same two identity-conditioned histories but encodes them as independent trajectories before downstream fusion. MITraG improves mean ROC-AUC from 0.7623 to 0.8174, supporting the benefit of representing overlapping histories through shared transaction events rather than treating them as separate sequences. MITraG also outperforms the temporal-graph baseline GTAN by 0.0317 mean ROC-AUC, indicating that deriving the graph structure from overlapping behavioral histories provides additional value beyond the evaluated conventional temporal-graph formulation.

5.3 Ablation Studies

Table 2: Ablation results on the three OOT test partitions.

	ROC-AUC ↑				Recall@1% ↑
Variant	Jan.	Feb.	Mar.	Avg.	Avg.
MITraG	**0.8200**	**0.8162**	**0.8161**	**0.8174**	**0.1360**
Matched Target-Only	*0.8159*	*0.8146*	*0.7779*	*0.8028*	*0.1217*
w/o MTNR Pretraining	0.8088	0.8033	0.7964	0.8028	0.1205
w/o Shared-Transaction Merging	0.8115	0.8079	0.7980	0.8058	0.1219
w/o Relation Types	0.8097	0.8050	0.7967	0.8038	0.1121
w/o Time-Gap Encoding	0.8158	0.8111	0.8031	0.8100	0.1227
Account-Conditioned Graph Only	0.8038	0.7977	0.7829	0.7948	0.1105
Card-Conditioned Graph Only	0.8067	0.8033	0.8002	0.8034	0.1146
Concatenation-Only Fusion	0.8104	0.8065	0.8008	0.8059	0.1225

Table 2 evaluates the main design choices in MITraG. Compared with Matched Target-Only, which retains the same target inputs and downstream architecture but removes historical transactions, MITraG improves mean ROC-AUC by 0.0146, confirming the value of behavioral context beyond the target transaction itself.

Restricting the graph to a single identity view or removing shared-transaction merging, relation types, or time-gap encoding consistently lowers ROC-AUC. Together, these results support the central representation assumption of MITraG: multiple identity-conditioned histories provide complementary behavioral context, while their shared events, identity-specific temporal relations, and irregular timing need to be preserved explicitly. Removing MTNR further reduces performance, showing that this structured context can be learned effectively from unlabeled transaction histories before fraud supervision. Replacing TGIF with direct concatenation also degrades performance, supporting the proposed downstream adaptation mechanism.

5.4 Multi-Identity Analysis

Cross-identity overlap is common in the evaluated payment data. Approximately 79% of the OOT samples contain at least one shared historical transaction between the account- and card-conditioned histories, excluding the target transaction. This indicates that overlap among identity-conditioned histories is a prevalent property of the underlying payment behavior rather than a rare structural case.

We further examine whether the benefit of shared-transaction modeling is associated with this overlap. Compared with the variant without shared-transaction merging, MITraG shows positive gains primarily in samples with cross-identity overlap, with the improvement generally increasing as the overlap ratio grows. Full overlap-conditioned results are reported in Appendix C. This pattern provides direct evidence that the performance gain is linked to the shared-event structure that MITraG is designed to represent.

We additionally stratify performance by history length and identity-view availability. MITraG outperforms FT-Transformer across all evaluated strata, including both single-view and dual-view cases. Cold-start results, additional sequence controls, and complete subgroup statistics are reported in Appendix C.

Finally, we examine whether the same representation extends beyond the account–card setting by introducing a device-conditioned history as an additional identity view. Under the same graph construction principle, the three-view model improves mean ROC-AUC from 0.8174 to 0.8329 and mean Recall@1% from 0.1360 to 0.1579 across the three OOT partitions. These results provide empirical evidence that the proposed representation is not tied to a fixed pair of identity-conditioned histories. Full partition-specific results are provided in Appendix C.6.

6 Discussion, Limitations, and Conclusion

Our results support representing identity-conditioned histories as overlapping views of shared physical events rather than independent transaction trajectories. The comparison with dual-sequence TabBERT, together with the shared-transaction and overlap-conditioned analyses, shows that the observed gains are associated with explicitly preserving cross-identity overlap rather than simply providing multiple histories. The single-view, relation, and time-gap ablations further support the role of complementary identity context and temporal structure, while the additional identity view suggests that the formulation is not tied to a fixed account–card pair.

MTNR improves OOT performance over training without self-supervised pretraining, supporting transaction-context learning from unlabeled behavioral histories before fraud supervision. Our evaluation is limited to one industrial payment domain and one downstream fraud-detection task, and does not directly measure label efficiency or cross-task and cross-domain transfer. Broader identity types, larger numbers of views, and histories beyond 35 transactions also remain unexplored. Overall, MITraG achieves a mean ROC-AUC of 0.8174 and Recall@1% of 0.1360 under strict OOT evaluation, supporting shared-event modeling as an effective basis for self-supervised payment behavior representation learning.

AI Use Statement

ChatGPT was used to assist with language editing, manuscript organization, LaTeX formatting, experimental log organization, and debugging of implementation issues. The research ideas, methodological design, and experimental protocol were developed by the authors. All AI-assisted content and debugging suggestions were reviewed and verified by the authors, and all reported results were checked against the original experiments and logs. The authors take full responsibility for the final manuscript and its scientific claims.

Reproducibility Statement

Appendix A details the temporal splits, leakage controls, and graph construction; Appendix B reports the model architecture, training settings, baselines, and computational cost; and Appendix C provides additional results and statistical testing details. The industrial payment data are proprietary and cannot be publicly released. We therefore provide anonymized statistics and implementation details to support independent reproduction of the proposed method.

References

Dmitrii Babaev, Nikita Ovsov, Ivan Kireev, Maria Ivanova, Gleb Gusev, Ivan Nazarov, and Alexander Tuzhilin. Coles: Contrastive learning for event sequences with self-supervision. In *Proceedings of the 2022 International Conference on Management of Data*, 2022.

D. T. Braithwaite, Misael Cavalcanti, R. Austin McEver, Hiroto Udagawa, Daniel Silva, Rohan Ramanath, Felipe Meneses, Arissa Yoshida, Evan Wingert, Matheus Ramos, Brian Zanfelice, and Aman Gupta. Your spending needs attention: Modeling financial habits with transformers. *arXiv preprint arXiv:2507.23267*, 2025.

Pin-Jung Chen, Sahil Bhatnagar, Sagar Goyal, Damian Konrad Kowalczyk, and Mayank Shrivastava. CASPR: Customer activity sequence-based prediction and representation. *NeurIPS Workshop on Table Representation Learning*, 2022.

Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S. Yu. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, 2020.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18932–18943. Curran Associates, Inc., 2021.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Anish Khazane, Jonathan Rider, Max Serpe, Antonia Gogoglou, Keegan Hines, C. Bayan Bruss, and Richard Serpe. DeepTrax: Embedding graphs of financial transactions. In *2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 126–133, 2019. doi: 10.1109/ICMLA.2019.00028.

Guilin Li, Yun Zhang, Xiuyuan Chen, Chengqi Li, Bo Wang, Linghe Kong, Wenjia Wang, Weiran Huang, and Matthias Hwai Yong Tan. PANTHER: Generative pretraining beyond language for sequential user behavior modeling. In *Advances in Neural Information Processing Systems*, volume 38, pp. 163622–163643. Curran Associates, Inc., 2025. doi: 10.52202/085713-5462.

Ao Liu, Yanmei Zhang, Youwei Wang, and Qiang Duan. IGT4ETH: An isotropic pre-trained graph transformer for ethereum account classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 23631–23639, 2026.

Can Liu, Yuncong Gao, Li Sun, Jinghua Feng, Hao Yang, and Xiang Ao. User behavior pre-training for online fraud detection. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.

Mingxuan Lu, Zhichao Han, Susie Xi Rao, Zitao Zhang, Yang Zhao, Yinan Shan, Ramesh Raghunathan, Ce Zhang, and Jiawei Jiang. BRIGHT: Graph neural networks in real-time fraud detection. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*, 2022.

Maxim Ostroukhov, Ruslan Mikhailov, Vladimir Iashin, Artem Sokolov, Andrei Akshonov, Vitaly Protasov, Andrey Goncharov, Dmitrii Beloborodov, Vince Mullin, Roman Yokunda Enzmann, Georgios Kolovos, Jason Renders, Pavel Nesterov, and Anton Repushko. PRAGMA: Revolut foundation model. *arXiv preprint arXiv:2604.08649v2*, 2026.

Inkit Padhi, Yair Schiff, Igor Melnyk, Mattia Rigotti, Youssef Mroueh, Pierre Dognin, Jerret Ross, Ravi Nair, and Erik Altman. Tabular transformers for modeling multivariate time series. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021.

Susie Xi Rao, Shuai Zhang, Zhichao Han, Zitao Zhang, Wei Min, Zhiyao Chen, Yinan Shan, Yang Zhao, and Ce Zhang. xFraud: Explainable fraud transaction detection. *Proceedings of the VLDB Endowment*, 15(3), 2022.

Piotr Skalski, David Sutton, Stuart Burrell, Iker Perez, and Jason Wong. Towards a foundation purchasing model: Pretrained generative autoregression on transaction sequences. In *Proceedings of the 4th ACM International Conference on AI in Finance*, 2023.

Ziming Wang, Qianru Wu, Baolin Zheng, Junjie Wang, Kaiyu Huang, and Yanjie Shi. Sequence as genes: An user behavior modeling framework for fraud transaction detection in e-commerce. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5194–5203, 2023. doi: 10.1145/3580305.3599905.

Sheng Xiang, Mingzhi Zhu, Dawei Cheng, Enxia Li, Ruihui Zhao, Yi Ouyang, Ling Chen, and Yefeng Zheng. Semi-supervised credit card fraud detection via attribute-driven graph representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 14557–14565, 2023.

Rithvik Y, Bhavuk Singhal, Shubham Jain, Akshat Garg, Karan Tanwar, Anshu Aditya, Debashis Mukherjee, and Debdoot Mukherjee. TRUST: Transaction risk via unified sequence and topology. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 40147–40156, 2026.

Chin-Chia Michael Yeh, Uday Singh Saini, Xin Dai, Xiran Fan, Shubham Jain, Yujie Fan, Jiarui Sun, Junpeng Wang, Menghai Pan, Yingtong Dou, Yuzhong Chen, Vineeth Rakesh, Liang Wang, Yan Zheng, and Mahashweta Das. TREASURE: A transformer-based foundation model for high-volume transaction understanding. In *Proceedings of the 32nd ACM SIGKDD Conference*

on Knowledge Discovery and Data Mining V.1, pp. 2518–2527, 2026. doi: 10.1145/3770854.3783942.

Zhizhi Yu, Chundong Liang, Xinglong Chang, Dongxiao He, Di Jin, and Jianguo Wei. Dynamic neighborhood modeling via node-subgraph contrastive learning for graph-based fraud detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 13115–13123, 2025.

Appendix Overview

The appendix provides additional details supporting the experimental methodology and empirical claims in the main paper. Appendix A describes the temporal protocol, leakage controls, model inputs, and graph preprocessing. Appendix B reports architecture, optimization, baseline implementation, and computational cost details. Appendix C provides partition-specific results, complementary metrics, overlap-conditioned analysis, identity-view extension experiments, component ablations, subgroup analyses, and the statistical testing protocol.

A Data and Preprocessing Details

A.1 Temporal Partitioning and Leakage Controls

Graph pretraining and supervised fraud detection follow strictly separated temporal protocols. Graph pretraining uses transactions from July–October 2025 for optimization and November 2025 for reconstruction-based checkpoint selection. Supervised training uses July–November 2025, validation uses December 2025, and the out-of-time (OOT) test set comprises three chronologically subsequent partitions from January, February, and March 2026.

All downstream checkpoints are selected exclusively using the December 2025 validation split before any OOT evaluation. The final Test metric reported in the paper is defined as the unweighted arithmetic mean over the three OOT partitions:

$$\mathrm{Test}(M) = \frac{1}{3}\left(M_{\mathrm{Jan}} + M_{\mathrm{Feb}} + M_{\mathrm{Mar}}\right),$$

where M denotes the corresponding evaluation metric. This calculation is performed separately for each training seed, after which the five seed-level test scores are averaged to obtain the reported point estimate.

Sample identifiers and physical-transaction identifiers are disjoint across supervised training, validation, and the three OOT test partitions. Account and card identities may recur across time, which reflects the intended deployment setting in which previously observed entities can generate future transactions. For every target transaction, historical context is constructed causally and contains only information available no later than the target transaction. Fraud labels from historical or target transactions are never exposed to graph pretraining.

A.2 Model Inputs and Graph Preprocessing

For each target transaction, records are ordered chronologically within the account-conditioned and card-conditioned histories. Each identity-conditioned history contains at most $L = 35$ transactions and ends at the target transaction. Occurrences associated with the same physical-transaction identifier are aligned and represented as one transaction node. Account-induced and card-induced temporal dependencies remain distinct relation types even when they connect the same pair of nodes. Corresponding reverse edge types and self-loops are subsequently added deterministically for graph message passing.

The identifier is not exposed to any learnable encoder or reconstruction objective. Similarly, manually engineered target-transaction features and fraud labels are excluded from graph pretraining. Table 3 summarizes the permitted roles of each information source.

Table 3: Model inputs and their permitted roles in MITraG.

Input	Representation	Role
Transaction features a_j^r	Numeric and categorical features	Pretraining and fine-tuning
physical-transaction identifier o_j^r	Transaction alignment key	Node merging only
Timestamp t_j^r	Temporal order and time gap	Edge ordering and time-gap encoding
Target vector x_i	102 engineered features	Supervised target branch only
Risk label y_i	Binary fraud target	Supervised objective only

A.3 Graph and Cross-Identity Overlap Statistics

In the account–card instantiation, the central structural property examined by MITraG is that identity-conditioned histories can overlap through shared physical transactions. Such transactions become shared graph nodes and provide interaction points between the corresponding temporal paths. To quantify the prevalence of this structure in the industrial dataset, Table 4 reports the share of samples with positive cross-identity overlap in each OOT test partition. Positive overlap indicates that the account- and card-conditioned histories share at least one historical transaction, excluding the target transaction.

Table 4: Positive cross-identity overlap by OOT test partition.

OOT partition	Samples	Positive-overlap share
January	6,558,187	79.03%
February	6,258,966	79.19%
March	7,357,043	78.74%
Combined	20,174,196	78.97%

The positive-overlap share remains close to 79% across all three partitions. These statistics are descriptive and are not used for model selection.

Figure A.1 provides a finer-grained view by decomposing the overlap ratio into buckets and reporting the corresponding fraud rates. Panel A reports the combined sample share and fraud rate by overlap-ratio bucket, with a bracket spanning the four positive-overlap buckets. Panel B shows the monthly and combined sample-share distributions.

B Model Configuration and Training Details

B.1 MITraG Architecture

MITraG follows the architecture described in Section 4. Numeric and categorical transaction features are first mapped into token embeddings using a tokenizer shared across identity views.

For the account–card instantiation used in the main experiments, account-conditioned and card-conditioned occurrences are processed by separately parameterized field Transformers before their representations are fused into a unique transaction-node state.

The graph encoder contains $K = 4$ relation-aware temporal attention layers. Message passing is restricted to the explicit computation edges defined by the transaction graph. The five relation types are account-forward, card-forward, account-reverse, card-reverse, and self-loop. Each non-self edge records the elapsed time between its endpoint transactions. This time gap is transformed using $\log(1 + \Delta t)$ before being encoded by the time-gap network.

After graph message passing, the contextualized target-anchor representation is combined with an attention-pooled graph summary to produce the graph representation g_i. During downstream fine-tuning, Target & Graph Interaction Fusion (TGIF) combines the projected target-transaction rep-

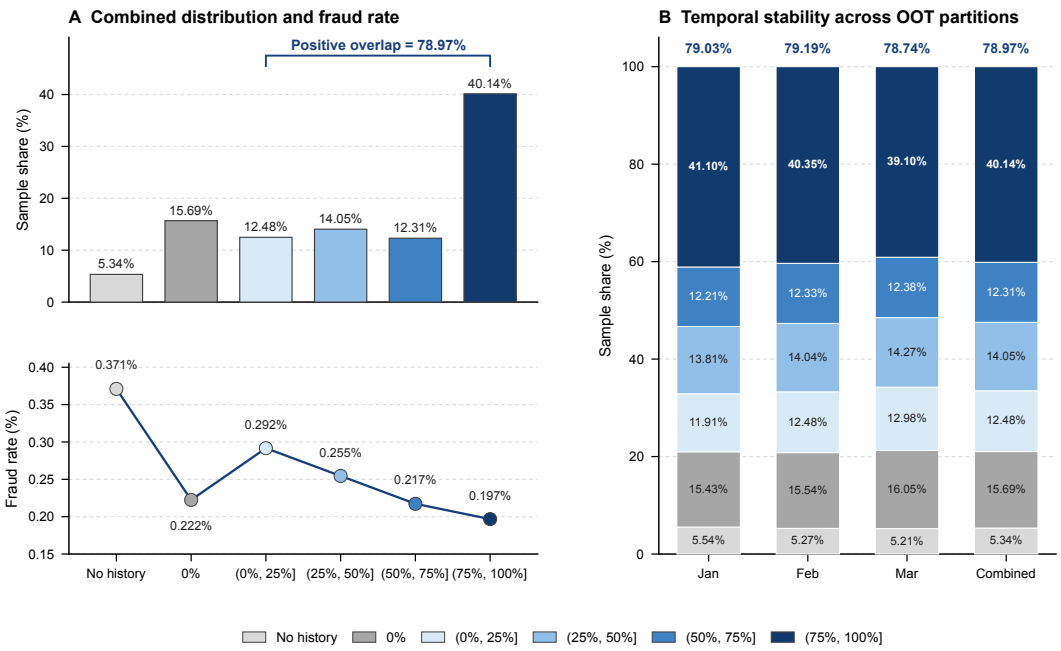

Figure A.1: Cross-identity overlap across the three OOT test partitions.

resentation and graph representation through direct features, element-wise products, and absolute differences.

For architectural reproducibility, Table 5 reports the dimensions used by the final pretraining and fine-tuning configurations.

Table 5: MITraG architecture configuration used for the reported experiments.

Architecture setting	**Value**
Maximum history length per identity	35
Graph encoder layers	4
Computation relation types	5
Field-token embedding dimension	192
Field Transformer layers / heads	3 / 8
Graph hidden dimension	512
Graph attention heads	8
Graph FFN dimension	2,048
Relation embedding dimension	512
Time-gap encoding dimension	512
TGIF projection dimension	128
Classification-head dimensions	$512 \rightarrow 128 \rightarrow 2$

B.2 Pretraining and Fine-Tuning Hyperparameters

Table 6 summarizes the optimization settings used for MITraG. During graph pretraining, checkpoints are selected by reconstruction loss on the November 2025 selection split. The selected graph encoder initializes downstream MITraG.

During supervised fine-tuning, positive examples are repeated by a factor of 50 in the training split only. Validation and all OOT test partitions preserve their natural class prevalence. No OOT partition contributes to optimization, early stopping, or checkpoint selection.

Table 6: Optimization settings used for the reported MITraG experiments.

Stage	Setting	Value
Graph	Maximum view length	35 transactions
Graph	Encoder layers	4
Pretraining	MTNR mask ratio	0.15
Pretraining	Precision / hardware	FP32 / 1 NVIDIA B300
Pretraining	Epochs	8
Pretraining	Micro-batch / accumulation	1,800 / 27
Pretraining	Effective batch size	48,600
Pretraining	Learning rate / weight decay	10^{-4} / 10^{-4}
Fine-tuning	Maximum epochs / batch size	10 / 1,024
Fine-tuning	Learning rate / weight decay	5×10^{-5} / 10^{-4}
Fine-tuning	Positive repetition	50, training split only
All stages	Optimizer	AdamW
All stages	Training seeds / independent runs	$\{42, 43, 44, 45, 46\}$ / 5

Computational cost. The graph encoder contains 34.88M parameters. Eight FP32 pretraining epochs on one NVIDIA B300 require 26 h 17 min, with peak allocated and reserved GPU memory of 57,718.4 MB and 78,330.0 MB, respectively. For the OOT partition with complete timing logs, batch inference at batch size 1,024 has mean, median, p95, and p99 latencies of 148.133, 144.707, 173.890, and 200.307 ms, respectively.

B.3 Baseline Configuration

All neural baselines follow the same temporal partitioning and validation-only checkpoint-selection protocol as MITraG. Historical inputs are restricted to information available no later than the target transaction.

FT-Transformer. FT-Transformer uses only the 102 engineered features x_i of the target transaction and therefore provides a neural transaction-level baseline without behavioral-history modeling.

LightGBM. LightGBM uses the same engineered target-transaction features as FT-Transformer and provides a strong tree-based tabular baseline.

TabBERT: Dual-Sequence Adaptation. TabBERT, the representation-learning model within the TabFormer framework Padhi et al. (2021), is adapted to the multi-identity setting by using separate encoders for the account-conditioned and card-conditioned transaction histories, each trained through masked transaction-sequence pretraining. The two identity views remain structurally separate throughout sequence encoding; their pretrained representations are fused with the target-transaction representation only during downstream fraud detection. We refer to this baseline as the dual-sequence TabBERT adaptation. It provides the primary matched sequence-pretraining comparison for testing whether preserving shared physical events within a common representation adds value beyond independently encoding the same identity-conditioned histories.

GTAN. GTAN provides the temporal-graph baseline. To avoid introducing supervision unavailable to MITraG pretraining, neither historical fraud labels nor the target transaction's fraud label is supplied as a graph-node input. Model selection follows the same OOT protocol with validation-only checkpoint selection.

Where applicable, sequence histories are truncated using the same maximum length of 35 transactions per identity view. No baseline is selected using OOT test performance.

C Additional Experimental Results

C.1 Metrics and Statistical Testing Protocol

ROC-AUC and Recall@1% are the primary evaluation metrics. ROC-AUC measures global ranking quality, while Recall@1% measures the fraction of fraudulent transactions captured when only the top-scoring 1% of transactions within an OOT partition can be reviewed.

We additionally report PR-AUC, Recall@10%, Recall-CAP, and Top-10 Area as supplementary diagnostics. PR-AUC is reported separately because its absolute value depends on fraud prevalence, which differs across temporal partitions. Top-10 Area is normalized to $[0, 1]$.

Metric implementation details. Let $R(x)$ denote the cumulative recall obtained by reviewing the highest-scoring fraction x of an evaluation partition, and let π denote the fraud prevalence in that partition. Recall-CAP is the area under the complete cumulative-recall curve,

$$A_{\mathrm{CAP}} = \int_0^1 R(x)\,\mathrm{d}x. \tag{18}$$

The implementation also provides the prevalence-adjusted quantity

$$\widetilde{A}_{\mathrm{CAP}} = \mathrm{clip}_{[-1,1]}\left(\frac{A_{\mathrm{CAP}} - 0.5}{1 - \pi/2 - 0.5}\right), \tag{19}$$

for which random, perfect, and reverse rankings correspond to 0, 1, and -1, respectively. The Recall-CAP values reported in this paper are the raw areas A_{CAP}.

Top-10 Area summarizes the same curve over the operational range $x \in [0, 0.1]$:

$$A_{\mathrm{Top10}} = \frac{1}{0.1}\int_0^{0.1} R(x)\,\mathrm{d}x. \tag{20}$$

Thus, the reported value is the interval-normalized average recall in the top 10% of the ranked population and lies in $[0, 1]$; it is not normalized against random or perfect rankings. For both metrics, equal scores are aggregated into tie groups. If an integration boundary falls within a tie group, positives are allocated proportionally within that group, equivalently using linear interpolation between group endpoints. The resulting metrics are therefore invariant to the input order of equal-scoring samples.

Statistical comparisons use paired predictions so that MITraG and each comparator are evaluated on exactly the same test transactions. Test-sample uncertainty is quantified using $B = 2{,}000$ paired stratified influence-bootstrap replicates with bootstrap random seed 42. Positive examples are resampled nonparametrically, while the million-scale negative component uses a paired joint influence-function Gaussian approximation that preserves comparator covariance. The same resampling indices are applied to all methods and all five seed-level prediction sets. In each replicate, the metric difference is recomputed separately for every seed and OOT month; the final effect is obtained by taking the unweighted mean across the three months within each seed and then averaging across seeds. We report percentile 95% confidence intervals and raw two-sided bootstrap p-values using the finite-replicate $+1$ correction. Holm correction is applied separately to ROC-AUC and Recall-CAP, with four prespecified comparator tests in each metric family.

C.2 Partition-Specific Main Results

Table 7 expands the main comparison into individual OOT partitions. The three OOT partitions form a strictly chronological test horizon. The resulting rankings show that MITraG maintains its advantage under temporal distribution shift.

MITraG achieves the highest ROC-AUC in all three OOT partitions, with an average of 0.8174. LightGBM is the strongest comparator by mean ROC-AUC at 0.7935, followed by GTAN at 0.7857; TabBERT reaches 0.7623. MITraG also obtains the highest mean Recall@1% at 0.1360, followed by GTAN at 0.1207.

Figure C.1 visualizes performance across the three OOT test partitions. Panels A and B show partition-specific ROC-AUC and Recall@1%, respectively, while Panels C and D show the corresponding three-partition means. Exact values are reported in Table 7.

Table 7: Partition-specific primary results on the three OOT partitions. R@1% denotes Recall@1%.

Model	Metric	Jan.	Feb.	Mar.	Avg.
FT-Transformer	AUC	0.7712	0.7761	0.7538	0.7670
	R@1%	0.1145	0.1182	0.1018	0.1115
LightGBM	AUC	0.7827	0.7908	0.8071	0.7935
	R@1%	0.0980	0.1051	0.1325	0.1119
TabBERT	AUC	0.7747	0.7683	0.7438	0.7623
	R@1%	0.0898	0.0766	0.0807	0.0824
GTAN	AUC	0.7887	0.7818	0.7866	0.7857
	R@1%	0.1169	0.1151	0.1302	0.1207
MITraG	AUC	**0.8200**	**0.8162**	**0.8161**	**0.8174**
	R@1%	**0.1280**	**0.1294**	**0.1506**	**0.1360**

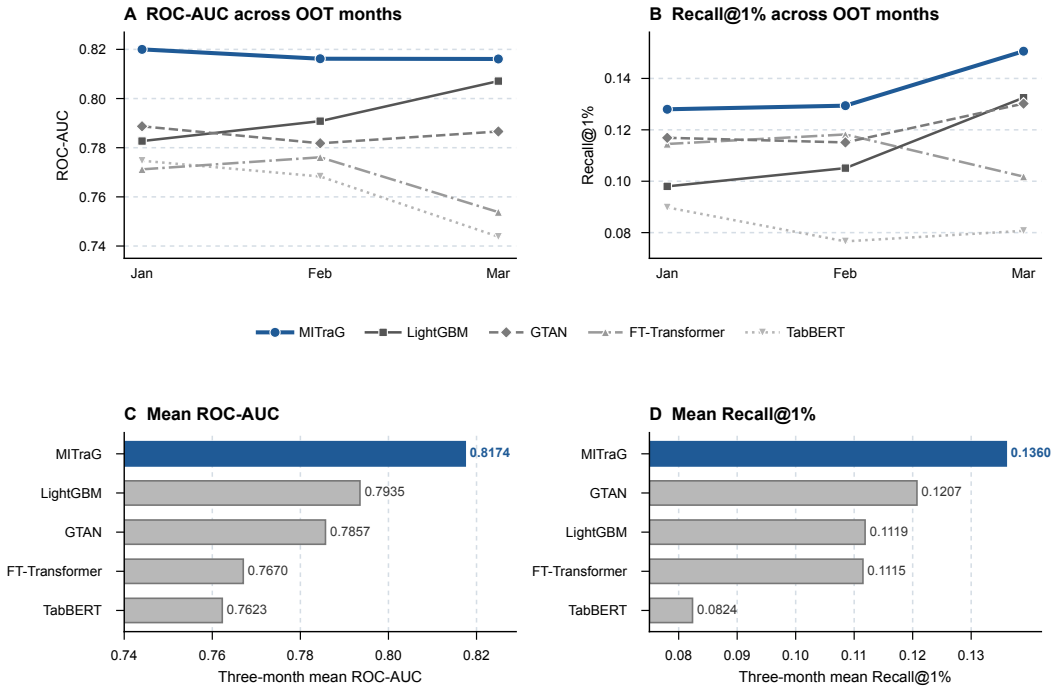

Figure C.1: Performance across the three OOT test partitions.

C.3 Complementary Risk-Capture Metrics

Table 8 reports supplementary metrics across the OOT test partitions, with ranking performed independently within each temporal partition.

Table 8: Supplementary risk-capture results on the OOT test partitions.

Model	Partition	PR-AUC	R@1%	R@10%	Recall-CAP	Top-10 Area
FT-Transformer	Jan. 2026	0.0118	0.1145	0.4094	0.7706	0.2616
	Feb. 2026	0.0129	0.1182	0.4116	0.7755	0.2668
	Mar. 2026	0.0138	0.1018	0.3805	0.7531	0.2427
LightGBM	Jan. 2026	0.0108	0.0980	0.4268	0.7821	0.2617
	Feb. 2026	0.0111	0.1051	0.4314	0.7902	0.2685
	Mar. 2026	0.0170	0.1325	0.4701	0.8064	0.3040
TabBERT	Jan. 2026	0.0100	0.0898	0.3961	0.7741	0.2424
	Feb. 2026	0.0080	0.0766	0.3760	0.7677	0.2229
	Mar. 2026	0.0103	0.0807	0.3455	0.7432	0.2112
GTAN	Jan. 2026	**0.0147**	0.1169	0.4256	0.7881	0.2741
	Feb. 2026	**0.0139**	0.1151	0.4130	0.7812	0.2635
	Mar. 2026	0.0212	0.1302	0.4271	0.7858	0.2816
MITraG	Jan. 2026	0.0137	**0.1280**	**0.4804**	**0.8194**	**0.3079**
	Feb. 2026	0.0135	**0.1294**	**0.4802**	**0.8155**	**0.3054**
	Mar. 2026	**0.0225**	**0.1506**	**0.4863**	**0.8152**	**0.3262**

MITraG achieves the highest Recall@10%, Recall-CAP, and Top-10 Area in all three OOT partitions. It also has the highest Recall@1% in all three OOT partitions. GTAN has the highest PR-AUC in January and February, while MITraG achieves the highest PR-AUC in March. PR-AUC remains sensitive to partition-level fraud prevalence and is therefore treated as a supplementary measure rather than direct evidence of temporal generalization.

C.4 Component Ablations

Table 10 reports partition-specific results for the component ablations corresponding to Table 2 in the main paper. All variants use the same supervised temporal protocol and validation-only model-selection procedure as the complete MITraG model. Matched Target-Only retains MITraG's target inputs and downstream architecture but removes preceding histories.

Table 9: Input and architecture matching for the Matched Target-Only control.

Component	MITraG	Matched Target-Only	FT-Transformer
Target 102	✓	✓	✓
Target base fields	✓	✓	–
Historical transactions	✓	–	–
Shared historical nodes	✓	–	–
TGIF	✓	✓	–
Classification head	MITraG	MITraG	FT-Transformer

Compared with Matched Target-Only, MITraG improves mean ROC-AUC by *0.0146*, supporting the value of historical context. Restricting the graph to either the account-conditioned or card-conditioned view shows that neither identity history alone reproduces the complete multi-identity representation. Removing shared-transaction merging also reduces AUC in all three partitions, supporting the representation of overlapping histories through shared physical events rather than as independent trajectories. Together with the overlap-conditioned analysis, this result links the observed performance gain to the shared-event structure that MITraG is designed to preserve.

Table 10: Partition-specific component ablations on the three OOT partitions. All variants follow the same training and model-selection procedure.

Variant	Metric	Jan.	Feb.	Mar.	Avg.
MITraG	AUC	**0.8200**	**0.8162**	**0.8161**	**0.8174**
	R@1%	**0.1280**	0.1294	**0.1506**	**0.1360**
Matched Target-Only	AUC	*0.8159*	*0.8146*	*0.7779*	*0.8028*
	R@1%	*0.1243*	***0.1312***	*0.1096*	*0.1217*
w/o MTNR Pretraining	AUC	0.8088	0.8033	0.7964	0.8028
	R@1%	0.1166	0.1169	0.1281	0.1205
w/o Shared-Transaction Merging	AUC	0.8115	0.8079	0.7980	0.8058
	R@1%	0.1229	0.1168	0.1259	0.1219
w/o Relation Types	AUC	0.8097	0.8050	0.7967	0.8038
	R@1%	0.1199	0.1121	0.1044	0.1121
w/o Time-Gap Encoding	AUC	0.8158	0.8111	0.8031	0.8100
	R@1%	0.1211	0.1210	0.1259	0.1227
Account-Conditioned Graph Only	AUC	0.8038	0.7977	0.7829	0.7948
	R@1%	0.1134	0.1056	0.1125	0.1105
Card-Conditioned Graph Only	AUC	0.8067	0.8033	0.8002	0.8034
	R@1%	0.1136	0.1093	0.1210	0.1146
Concatenation-Only Fusion	AUC	0.8104	0.8065	0.8008	0.8059
	R@1%	0.1211	0.1197	0.1267	0.1225

Removing relation types or time-gap encoding further degrades AUC, indicating that identity-specific temporal relations and irregular timing contribute beyond graph connectivity alone. Removing MTNR reduces downstream performance, supporting self-supervised transaction-context learning from unlabeled histories. Replacing TGIF with direct concatenation also reduces AUC consistently across OOT periods.

Recall@1% is more variable across individual temporal partitions than ROC-AUC. We therefore base component-level conclusions on the complete three-partition evaluation rather than interpreting isolated Recall@1% changes from a single month.

C.5 Overlap-Conditioned Shared-Event Analysis

We examine whether the benefit of shared-transaction modeling is associated with the amount of cross-identity overlap. Samples are grouped by the overlap ratio between the account- and card-conditioned histories, excluding the target transaction. Within each group, MITraG is compared with the variant without shared-transaction merging on the same transactions.

Figure C.2 reports the resulting ROC-AUC differences. Panel A shows the monthly differences with paired DeLong 95% confidence intervals, while Panel B reports the arithmetic mean across the three OOT partitions, with whiskers spanning the monthly minimum and maximum. Differences are computed as MITraG minus the no-merging variant.

The gain from shared-transaction modeling is concentrated in samples with positive cross-identity overlap and generally increases as the overlap ratio grows. In contrast, the no-history and zero-overlap groups do not show the same pattern. This result links the observed performance improvement to the shared-event structure that MITraG is designed to preserve.

C.6 Extension to a Third Identity View

To evaluate whether the proposed representation extends beyond the account–card instantiation, we introduce a device-conditioned history as an additional identity view under the same temporal protocol. MITraG denotes the account–card model, while MITraG-3V applies the same shared-event graph construction with the additional device-conditioned relations. Table 11 reports the partition-specific point estimates and their three-partition means.

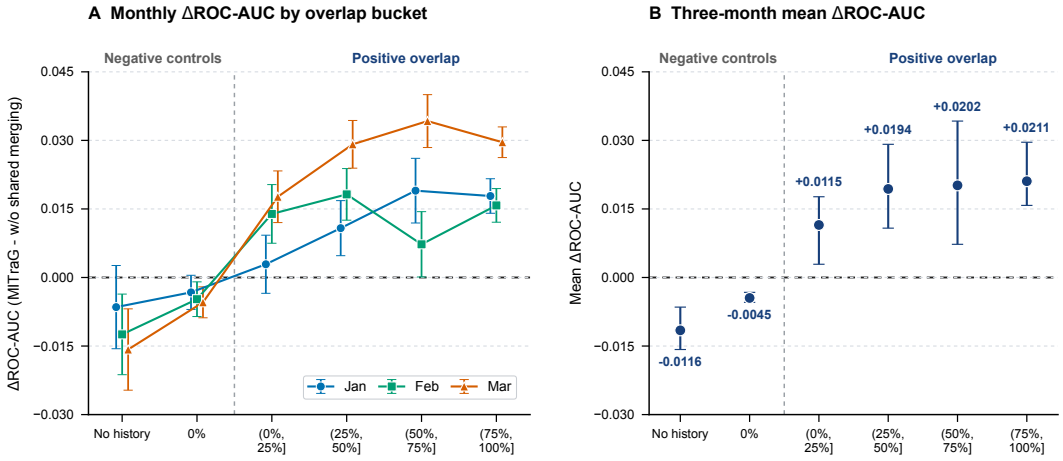

Figure C.2: Overlap-conditioned gain from shared-transaction merging.

Table 11: Extension from the account–card MITraG model to MITraG-3V, which additionally incorporates a device-conditioned history, under the same OOT protocol. Δ denotes MITraG-3V minus MITraG.

Partition	Metric	MITraG	MITraG-3V	Δ
Jan. 2026	ROC-AUC ↑	0.8200	**0.8338**	+0.0138
	PR-AUC ↑	0.0137	**0.0176**	+0.0039
	Recall@1% ↑	0.1280	**0.1444**	+0.0164
	Recall@10% ↑	0.4804	**0.5132**	+0.0328
	Recall-CAP AUC ↑	0.8194	**0.8331**	+0.0137
Feb. 2026	ROC-AUC ↑	0.8162	**0.8333**	+0.0171
	PR-AUC ↑	0.0135	**0.0183**	+0.0048
	Recall@1% ↑	0.1294	**0.1476**	+0.0182
	Recall@10% ↑	0.4802	**0.5179**	+0.0377
	Recall-CAP AUC ↑	0.8155	**0.8327**	+0.0172
Mar. 2026	ROC-AUC ↑	0.8161	**0.8315**	+0.0154
	PR-AUC ↑	0.0225	**0.0297**	+0.0072
	Recall@1% ↑	0.1506	**0.1816**	+0.0310
	Recall@10% ↑	0.4863	**0.5220**	+0.0357
	Recall-CAP AUC ↑	0.8152	**0.8306**	+0.0154
Avg.	ROC-AUC ↑	0.8174	**0.8329**	+0.0155
	PR-AUC ↑	0.0166	**0.0219**	+0.0053
	Recall@1% ↑	0.1360	**0.1579**	+0.0219
	Recall@10% ↑	0.4823	**0.5177**	+0.0354
	Recall-CAP AUC ↑	0.8167	**0.8321**	+0.0154

Across the three OOT partitions, MITraG-3V achieves a higher value than MITraG for every reported metric. Mean ROC-AUC increases from 0.8174 to 0.8329, and mean Recall@1% increases from 0.1360 to 0.1579. These results provide empirical evidence that the shared-event representation is not tied to a fixed account–card pair.

C.7 Additional Sequence Controls

In addition to the primary dual-sequence TabBERT adaptation, we evaluate several sequence controls to separate the effects of multi-identity input availability, self-supervised pretraining, and explicit shared-event graph construction.

Table 12: Sequence-control results on the three OOT test partitions.

Model	Metric	Jan.	Feb.	Mar.
FT-Transformer	AUC	0.7712	0.7761	0.7538
	R@1%	0.1145	0.1182	0.1018
Target + Card-Conditioned Sequence	AUC	0.7964	0.7458	0.7685
	R@1%	0.1136	0.0808	0.1036
Target + Account-Conditioned Sequence	AUC	0.8082	0.7609	0.7800
	R@1%	0.1220	0.0791	0.1142
TabBERT	AUC	0.7747	0.7683	0.7438
	R@1%	0.0898	0.0766	0.0807
MITraG	AUC	**0.8200**	**0.8162**	**0.8161**
	R@1%	**0.1280**	**0.1294**	**0.1506**

Single-view sequence models improve over the target-only FT-Transformer, providing evidence that identity-conditioned historical context is informative. The dual-sequence TabBERT adaptation provides the closest matched sequence-pretraining control because it applies masked self-supervised pretraining separately to the account- and card-conditioned histories while keeping the two views structurally independent. MITraG further improves ROC-AUC consistently across the OOT periods, supporting the benefit of representing overlapping identity-conditioned histories through shared physical events rather than keeping them as structurally independent trajectories.

C.8 History Length and Identity-View Availability

We further stratify test transactions by the amount of available identity-conditioned history. History length is defined as the maximum number of preceding account- or card-conditioned transactions after excluding the target transaction. Within every subgroup, FT-Transformer and MITraG are evaluated on exactly the same samples.

Table 13 reports history-length and identity-availability analyses under the same OOT protocol. The results show a consistent MITraG advantage across history lengths and across mutually exclusive identity-view availability groups, including cold-start transactions without preceding identity-conditioned history. Pooled N, Pos., and Rate aggregate the three OOT test partitions; FT Avg. and MITraG Avg. are unweighted means of monthly ROC-AUC, and Δ is MITraG minus FT-Transformer.

Figure C.3 summarizes performance and data composition by history length and identity-view availability. The upper panels compare the three-partition mean ROC-AUC of FT-Transformer and MITraG, with error bars spanning the monthly minimum and maximum. The lower panels report the sample share and fraud rate of each subgroup.

MITraG outperforms FT-Transformer in all history-length strata. The gain remains positive even for the zero-history group. Because this subgroup contains no preceding identity-conditioned transaction, the improvement should be interpreted as cold-start behavior of the complete MITraG system rather than as evidence of historical graph message passing. The Matched Target-Only control in Appendix C.4 directly evaluates the contribution of historical context while holding the target-transaction inputs and downstream architecture fixed.

Table 13: Subgroup results on the three OOT test partitions.

Group	Pooled N	Pos.	Rate (%)	FT Avg.	MITraG Avg.	Δ
History length						
0	1,076,963	3,996	0.3710	0.6460	**0.7225**	0.0765
1–4	3,258,091	12,454	0.3822	0.7159	**0.7911**	0.0752
5	646,995	2,271	0.3510	0.7206	**0.8033**	0.0827
6–10	2,613,756	7,596	0.2906	0.7376	**0.8073**	0.0697
11–20	3,416,133	7,839	0.2295	0.7594	**0.8219**	0.0625
21+	9,162,258	12,765	0.1393	0.7840	**0.8208**	0.0368
Identity-view availability						
Account only	2,859,243	5,436	0.1901	0.8482	**0.8843**	0.0361
Card only	169,244	723	0.4272	0.6005	**0.7173**	0.1167
Both histories	16,068,746	36,766	0.2288	0.7596	**0.8125**	0.0529
No history	1,076,963	3,996	0.3710	0.6460	**0.7225**	0.0765

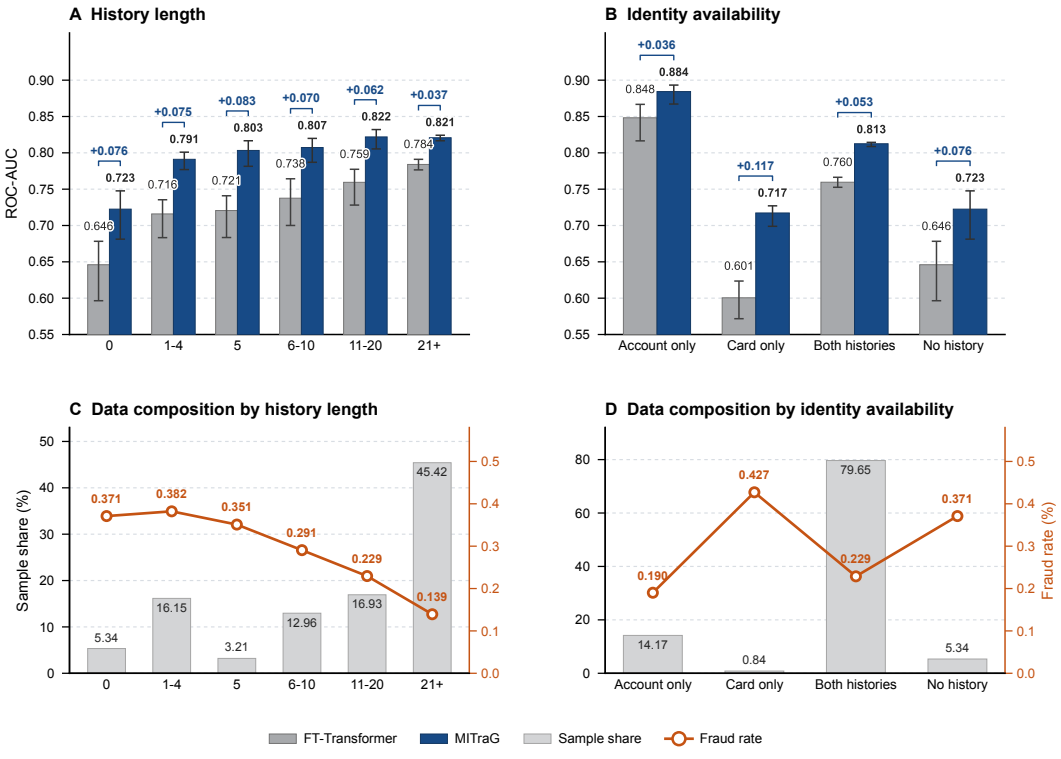

Figure C.3: Subgroup performance and data composition.

Absolute ROC-AUC generally increases with available history length, while MITraG continues to improve over the target-only baseline in the longest-history group. Across the three-month means of the four mutually exclusive identity-availability groups, MITraG also exceeds FT-Transformer in every group.

The subgroup populations differ substantially in their sample composition. Consequently, absolute AUC differences between the account-only, card-only, dual-identity, and no-history groups do not identify the causal contribution of a particular identity view. The controlled single-view graph ablations in Section C.4 are used for that purpose.

C.9 Statistical Significance

Paired significance analysis follows the protocol defined in Section C.1.

Effect sizes are computed as MITraG minus the comparator in the unweighted mean across the three OOT test partitions. Intervals are percentile 95% intervals from 2,000 paired stratified influence-bootstrap replicates. All comparisons remain significant after Holm correction, with an adjusted p-value of 0.003998.

Figure C.4 visualizes the paired effects on ROC-AUC and Recall-CAP. Points denote mean differences, and horizontal bars denote 95% confidence intervals. Positive values favor MITraG.

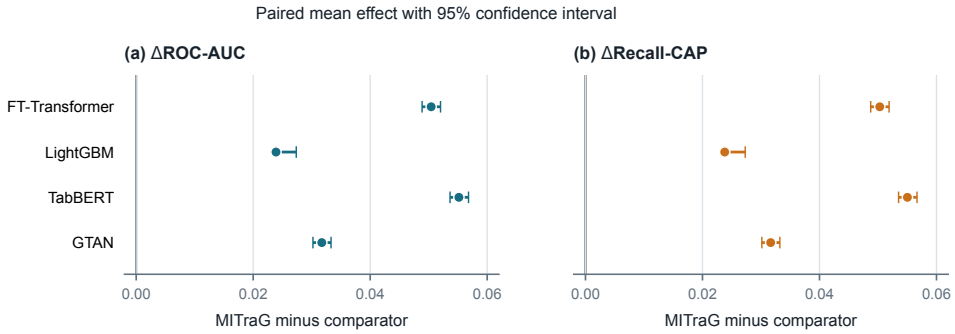

Figure C.4: Paired effects on ROC-AUC and Recall-CAP.

Recall@1% is treated as an operational point metric and is therefore excluded from the interval-based significance analysis. The five independent training runs characterize training-seed variation, whereas the paired-bootstrap intervals condition on the five frozen prediction sets and quantify uncertainty due to the finite evaluation sample.