PANTHER: Generative Pretraining Beyond Language for Sequential User Behavior Modeling

Guilin Li[2,*] **Yun Zhang**[2,*] **Xiuyuan Chen**[1,*] **Chengqi Li**[1,3] **Bo Wang**[2]
Linghe Kong[1] **Wenjia Wang**[5] **Weiran Huang**[1,3,†] **Matthias Hwai Yong Tan**[4]

[1]Shanghai Jiao Tong University [2]WeChat Pay, Tencent
[3]Shanghai Innovation Institute [4]City University of Hong Kong
[5]Hong Kong University of Science and Technology (Guangzhou)

Abstract

Large language models (LLMs) have shown that generative pretraining can distill vast world knowledge into compact token representations. While LLMs encapsulate extensive world knowledge, they remain limited in modeling the behavioral knowledge contained within user interaction histories. User behavior forms a distinct modality, where each action—defined by multi-dimensional attributes such as time, context, and transaction type—constitutes a behavioral token. Modeling these high-cardinality, sparse, and irregular sequences is challenging, and discriminative models often falter under limited supervision. To bridge this gap, we extend generative pretraining to user behavior, learning transferable representations from unlabeled behavioral data analogous to how LLMs learn from text. We present PANTHER, a hybrid generative–discriminative framework that unifies user behavior pretraining and downstream adaptation, enabling large-scale sequential user representation learning and real-time inference. PANTHER introduces: (1) Structured Tokenization to compress multi-dimensional transaction attributes into an interpretable vocabulary; (2) Sequence Pattern Recognition Module (SPRM) for modeling periodic transaction motifs; (3) a Unified User-Profile Embedding that fuses static demographics with dynamic transaction histories, enabling both personalized predictions and population-level knowledge transfer; and (4) Real-time scalability enabled by offline caching of pre-trained embeddings for millisecond-level inference.Fully deployed and operational online at WeChat Pay, PANTHER delivers a 25.6% boost in next-transaction prediction HitRate@1 and a 38.6% relative improvement in fraud detection recall over baselines. Cross-domain evaluations on public benchmarks (CCT, MBD, MovieLens-1M, Yelp) show strong generalization, achieving up to 21% HitRate@1 gains over transformer baselines, establishing PANTHER as a scalable, high-performance framework for industrial user sequential behavior modeling.

1 Introduction

Online payment platforms, such as WeChat Pay, Alipay, and PayPal, process billions of transactions daily, supported by critical applications like fraud detection, credit risk assessment, and personalized marketing [1]. Modeling payment behavior at this scale is challenging: data volumes are extreme; categorical features (e.g., merchant category, payment channel) are high-cardinality; and real-time risk decisions must be delivered under 100 ms.

[*]Equal contribution. Contact: {guilinli,aaayunzhang}@tencent.com.
[†]Corresponding author. Contact: weiran.huang@outlook.com.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

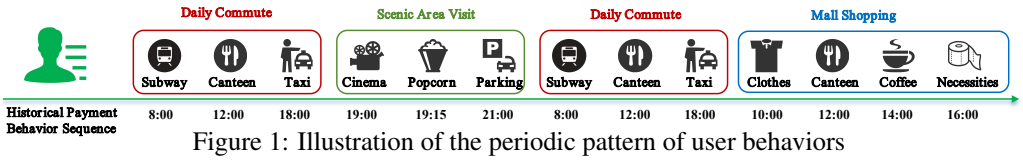

Figure 1: Illustration of the periodic pattern of user behaviors

Recent advances in self-supervised pretraining have transformed representation learning in language, vision, and recommendation [2–6]. Large language models (LLMs) demonstrate that generative pretraining on unlabeled text can compress extensive *world knowledge* into token representations. Payment platforms, however, require models that capture *behavioral knowledge*—the individualized regularities, intents, and deviations embedded in users' interaction histories. In this modality, each action is a structured event rather than a word; we therefore view a transaction as a *behavioral token* defined by multi-dimensional attributes (time, context, device, counterparty, amount). Extracting user-relevant information from these high-cardinality, sparse, and irregular sequences is essential for understanding users and for operational decision-making at scale.

Mainstream industrial approaches, adapted from recommendation systems (e.g., DeepFM [7], DCN [8], AutoFIS [9], DIEN [10]), rely primarily on supervised discriminative models for transaction-level classification. At billion-user scale they face four persistent limitations: (1) *label scarcity*—positive examples are too few to span the combinatorial feature space; (2) *overfitting to high-dimensional categories*—models memorize spurious co-occurrences rather than meaningful risk patterns; (3) *latency-driven truncation* of long histories—weakening the ability to leverage long-range behavior; and (4) *static embeddings*—biased toward frequent labels and brittle under cold-start and long-tail distributions [11].

Although sequential recommenders increasingly adopt generative objectives, they typically use generation as an *end task* (next-item prediction) (for example HSTU [12]) rather than as a *pretraining mechanism* to compress *user knowledge* into transferable representations. In contrast, we adopt a *pretrain→adapt* perspective: learn generalizable user embeddings from unlabeled behavioral logs *offline*, then *online* adapt them with lightweight discriminative heads to satisfy production latency constraints. This hybrid generative–discriminative design targets *user understanding* rather than only item generation, and supports multiple downstream decisions (fraud detection, transaction prediction, recommendation).

Orthogonal to this objective/framework difference, user behavior sequence data inherently differs from other sequential modalities. Unlike natural language, which is governed by grammatical structures, user behavior sequences exhibit rich recurring patterns—daily routines, weekly cycles, and seasonal trends—reflecting habitual user behaviors (illustrated in Figure 1). Standard sequential architectures, including Transformers, process events individually through self-attention mechanisms. Although powerful, these methods inadequately capture local periodicities and global relational patterns inherent to payment data, often diluting signals from lengthy sequences and missing subtle yet crucial anomalies indicative of fraud.

We propose **PANTHER** (Pattern AttentioN Transformer with Hybrid User ProfilER), a hybrid generative–discriminative framework that unifies *user behavior pretraining* with *downstream adaptation* for real-time inference. *Offline*, a PANTHER transformer is pretrained on billions of transactions to predict subsequent events, producing compact user-profile embeddings and event likelihoods that encode long-term intent and temporal dynamics. *Online*, a lightweight classifier fuses these cached representations with current transaction context to compute risk within milliseconds. This design leverages generative pretraining for representational power while keeping inference efficient for high-throughput systems.

To make pretraining effective on user behavior sequences and inference feasible online, PANTHER introduces three modeling components and one systems mechanism:

1. **Token Space Compression.** A *structured tokenization* scheme integrates contextual and counterparty attributes into unified tokens and applies frequency-aware compression to reduce dimensionality, filtering noise and stabilizing generative learning over heterogeneous inputs.
2. **Pattern-Aware Convolutional Cross-Attention.** A *Sequence Pattern Recognition Module (SPRM)* blends multi-scale (depthwise) convolutions with cross-attention to capture local periodicities (e.g., daily/weekly cycles) together with broader contextual dependencies—preserving cyclical signals while maintaining global relations.

3. **User Profile Embedding with Contrastive Personalization.** A dedicated *user-profile token* provides persistent access to user context across the sequence; a contrastive objective arranges users with similar payment behaviors nearby in latent space, improving personalization under sparsity and cold-start.
4. **Real-Time Hybrid Inference.** Pretrained user/profile embeddings are cached offline and fused online with context, recent patterns, and deviation features to produce millisecond-level posterior scores—meeting production latency constraints.

We empirically validate PANTHER on real-world WeChat Pay data, demonstrating strong generalization across fraud detection, transaction prediction, personalized user modeling, and recommendation. PANTHER yields a 25.6% improvement over Transformer baselines on internal WeChat Pay benchmarks and a 21% HR@1 gain on MovieLens-1M; on Yelp, it improves NDCG@5 by 29.6% over DCN. A production PANTHER-based fraud system at WeChat Pay improves Top-0.1% recall by 38.6% in online A/B tests, enhancing security across billions of daily transactions.

In summary, PANTHER provides a scalable, efficient approach to modeling complex sequential user behaviors by extending generative pretraining beyond language to the behavioral modality, compressing user knowledge into transferable representations, and bridging pretraining with real-time inference for industrial sequential decision-making.

2 Related Work

Sequential Deep Learning and Generative Recommendation. Deep learning methods for sequential modeling have significantly advanced recommendation and personalization, evolving from early RNN- and CNN-based models (e.g., GRU4Rec [13], Caser [14]) to recent transformer-based approaches (e.g., SASRec [15], BERT4Rec [16]). Generative sequential models, such as HSTU [12], TIGER [17], DiffuRecSys [18], and HLLM [19] have further advanced the field by modeling complex temporal dependencies and uncertainties. Despite these advancements, several aspects remain under-explored: explicit modeling of periodic behaviors (e.g., daily or weekly patterns), dedicated long-term personalized user embeddings, and computational strategies for low-latency inference. PANTHER uniquely addresses these challenges by incorporating convolutional cross-attention for periodic behavior modeling, contrastive user personalization embeddings, and cached representations enabling efficient real-time inference.

Fraud Detection in Financial Systems. Fraud detection in financial systems involves severe class imbalance, label scarcity, and real-time inference constraints. Early supervised approaches, including logistic regression, decision trees, and gradient boosting, perform effectively on structured data but face challenges with sparse, noisy, high-dimensional transaction data and extreme class imbalance [20]. Graph-based methods like R-GCNs [21] and heterogeneous GNNs [22] effectively capture relational patterns among entities, yet scalability and real-time latency in billion-user scenarios remain open research areas. PANTHER complements these methods by leveraging transformer-based generative pretraining, efficiently modeling temporal user behaviors and maintaining millisecond-level inference.

3 PANTHER

3.1 Model Overview and Problem Formulation

PANTHER employs a two-stage architecture for payment fraud detection, combining offline generative pretraining with real-time inference. Let $\mathcal{U}$ denote our user base where each user $u \in \mathcal{U}$ generates a sequence of payment events $\mathbf{X}_u = [x_1, x_2, \ldots, x_L]$ with $x_t \in \mathcal{V}$ representing compressed transaction tokens (see §3.2). Our system aims to estimate the fraud probability:

$$Pr(y = 1 \mid x_{\text{new}}, \mathbf{c}_{\text{new}}, \mathbf{X}_u), \tag{1}$$

for each new transaction x_{new} with contextual features $\mathbf{c}_{\text{new}}$, given the user's historical sequence $\mathbf{X}_u$.

Stage 1: Pretraining for Next Transaction Prediction. We first learn user behavior patterns through next payment behavior prediction. Each user behavior sequence is augmented with a learnable profile token x_{profile} encoding static attributes (see §3.4). Our transformer-based model with SPRM

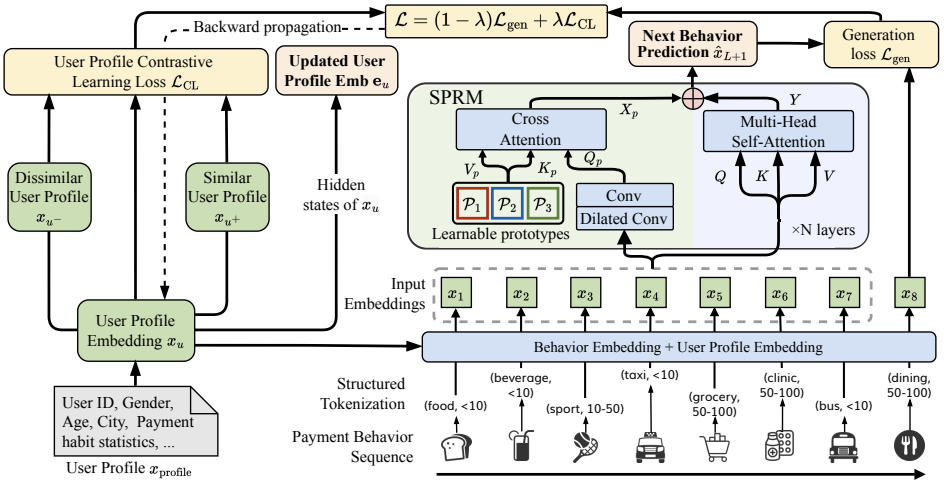

Figure 2: Key components of PANTHER: Structured Tokenization, SPRM and user profile embedding modules (§3.3) optimizes:

$$\mathcal{L}_{\text{gen}} = \mathbb{E}_{u\sim\mathcal{U}}\left[\sum_{t=1}^{L} -\log Pr_\theta(x_t \mid x_{<t}, x_{\text{profile}})\right], \tag{2}$$

where the loss is the negative log-likelihood of the ground-truth next behaviors over the sequence. This produces two key pretrained outputs: (1) user profile embeddings $\mathbf{e}_u \in \mathbb{R}^d$, and (2) behavior predictors $\mathbf{X}_u$ that predict next behavior $Pr_\theta(\hat{x}_{L+1}|\mathbf{X}_u)$ through a linear and softmax layer.

Stage 2: Hybrid Inference for Real-Time Fraud Detection. For live transactions, we compute risk scores through feature fusion:

$$Pr(y=1|\cdot) = g_\phi\Big(\underbrace{\psi(\mathbf{c}_{\text{new}})}_{\text{context}}, \underbrace{f_{\text{enc}}(\mathbf{X}_u^{[\text{L-100:L}]})}_{\text{recent patterns}}, \underbrace{\mathbf{e}_u}_{\text{long-term profile}}, \underbrace{\Delta(\hat{x}_{L+1}, x_{\text{new}})}_{\text{behavior deviation}}\Big), \tag{3}$$

where the function g_ϕ represents a general downstream discriminative model, which can take various flexible forms depending on specific tasks or use cases, $\psi(\cdot)$ embeds transaction context features, f_{enc} encodes the last 100 transactions via TextCNN [23], and Δ measures the distance between predicted and observed behaviors.

This design strikes a balance between accurate long-term behavior modeling and responsive real-time decision-making, making it particularly well-suited for high-throughput fraud detection systems.

3.2 Structured Tokenization for Payment Behaviors

Unlike natural language, payment behaviors lack predefined semantic units, as each transaction is defined by a combination of *contextual features* (e.g., payment channel, discretized amount) and *counterparty attributes* (e.g., merchant category, risk level). To efficiently capture this structured, multi-dimensional information, we introduce a *structured tokenization* framework that integrates these attributes into a unified representation.

Each transaction token is formed as the Cartesian product of contextual and counterparty features:

$$\tau = (c_i, a_j, m_k, r_l) \in \mathcal{C} \times \mathcal{A} \times \mathcal{M} \times \mathcal{R}, \tag{4}$$

where, for instance, c_i might represent a payment channel such as `CreditCard` or `RedPocket`, and a_j could capture the transaction amount, discretized into ranges like `$10-50` and `$50-100`. On the counterparty side, m_k can capture the merchant category, while r_l indicate the associated risk level, reflecting the merchant's historical reliability (e.g., `HighRisk`, `LowRisk`). This tokenization approach captures the key transactional semantics by transforming raw payment behaviors into compact, domain-specific tokens (e.g., (e.g., `CreditCard_$50-100_Fuel_LowRisk`).), effectively embedding transactions within a structured, context-rich feature space. This design allows the

pretraining model to learn meaningful representations of transaction behavior, preserving critical financial signals while reducing sparsity.

However, the raw combinatorial space is $|\mathcal{V}| = |\mathcal{C}| \times |\mathcal{A}| \times |\mathcal{M}| \times |\mathcal{R}| \approx 2\text{M}$, which is prohibitively large, leading to severe sparsity and overfitting risks. To address this, we adopt a *frequency-based compression*, leveraging real-world transaction distributions to retain only the top $K = 60,000$ most frequent tokens, covering over 96% of historical transactions. Less frequent, long-tail combinations are mapped to a unified `[UNK]` token, effectively reducing the vocabulary size by 97% (from 2M to 60k), while preserving high-impact and interpretable patterns. It maintains critical transaction semantics while significantly reducing the model's computational footprint.

3.3 Sequence Pattern Recognition Module (SPRM)

Payment sequences, unlike natural language, lack a formal grammar, yet exhibit structured recurring patterns, such as sequential routines and periodic behaviors (as illustrated in Figure 1). Accurately modeling these patterns is critical for applications like next-payment prediction and fraud detection, as they encapsulate context-aware user habits. However, standard self-attention in Transformers processes each event independently, incurring a quadratic complexity and lacking inductive biases for local and periodic structures. This inefficiency can dilute signals in long sequences and obscure subtle deviations from the routine. To address these limitations, we introduce the Sequence Pattern Recognition Module (SPRM), which incorporates two lightweight inductive biases to explicitly capture local and periodic patterns in payment sequences. Operating in parallel with the multi-head self-attention, the SPRM enhances the final representation by adding its output to the Transformer's output, forming a composite result that leverages both global context and structured pattern recognition.

(i) Local Pattern Aggregation. To capture the multi-scale nature of transactional routines, we apply depthwise dilated convolutions to the token embeddings $H \in \mathbb{R}^{T \times d}$, using a range of kernel sizes w and dilation rates r:

$$H_p^{(k)} = \text{Conv}_{\text{dil}=r_k,\, w=w_k}(H), \quad H_p = \text{Concat}_k\big(H_p^{(k)}\big).$$

Here, kernels with smaller dilation rates (e.g. $w = 3$, $r = 1$) capture immediate temporal clusters, while larger dilated kernels (e.g., $w = 3$, $r = 2$) capture periodic or recurring patterns, even in the presence of occasional noise or sporadic behaviors. This convolutional aggregation efficiently captures multiscale transactional patterns in linear time, providing a compact representation that encodes both short-term and long-term transactional routines while remaining robust to random fluctuations within those patterns.

(ii) Prototype-Driven Pattern Matching. To further enrich these multiscale embeddings, we introduce m learnable prototypes $\mathcal{P} \in \mathbb{R}^{m \times d}$, each representing a canonical spending motif (e.g., *weekend leisure*, *weekday commute*). These prototypes act as structured anchors in the embedding space, providing a scaffold for the model to map observed behaviors to known patterns.

$$Q = H_p W_Q, \quad K = \mathcal{P} W_K, \quad V = \mathcal{P} W_V, \quad X_p = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V.$$

This cross-attention mechanism compresses each local segment onto the nearest prototypes, effectively aligning real-world sequences to interpretable, high-level motifs, while preserving local context. Unlike vanilla self-attention, which treats every token as context-free, this approach enforces a structured alignment, encouraging the model to form sparse, interpretable codes that highlight subtle departures from routine behavior.

Complexity Analysis. For a sequence of length T, standard self-attention incurs a quadratic $\mathcal{O}(T^2)$ complexity. In contrast, SPRM introduces only a linear cost for dilated convolutions, $\mathcal{O}(T)$, and a modest $\mathcal{O}(Tm)$ for prototype cross-attention, where $m \ll T$ (typically $m = 64$ in practice). This results in an overall complexity of approximately $\mathcal{O}(T)$, making it feasible for long payment sequences without sacrificing long-range dependency capture.

3.4 User Profile Embedding for Personalized Payment Predictions

In PANTHER, we propose a novel *user profile embedding* that enables personalized transaction predictions by learning shared behavioral patterns among demographically similar users. This embedding combines static user attributes with dynamic transaction histories to form a compact latent

Table 1: Next-transaction prediction results on WeChat Pay. Relative improvements reflect PANTHER's gains over the Transformer baseline.

Method	HR@1	HR@5	HR@10
Transformer	0.1952	0.4121	0.5308
SASRec	0.2041	0.4280	0.5347
HSTU	0.2089	0.4271	0.5320
PANTHER (SPRM only)	0.2243(+14.9%)	0.4493(+9.03%)	0.5421(+2.13%)
+ Profile as First Token	0.2301(+17.9%)	0.4634(+12.45%)	0.5468(+3.01%)
+ Profile as Positional Encoding	0.2351(+20.4%)	0.4774(+15.85%)	0.5568(+4.90%)
+ Profile + CL	**0.2452 ± 0.0005 (+25.6%)**	**0.4837 ± 0.0009 (+17.37%)**	**0.5647 ± 0.0012 (+6.39%)**

representation, which simultaneously serves two purposes: (1) as a personalized positional encoding that contextualizes user-specific transaction sequences, and (2) as a learnable similarity anchor that adaptively refines historical behavior patterns through contrastive learning.

Our contrastive objective follows an information-theoretic formulation:

$$\mathcal{L}_{\text{CL}} = -\sum_{(i,j)\in Pos} \log \frac{\exp(-\|e_i - e_j\|_2/\tau)}{\sum_{k\in\mathcal{U}\setminus\{i\}} \exp(-\|e_i - e_k\|_2/\tau)}, \tag{5}$$

where Pos denotes positive pairs of users sharing demographic attributes (age$\pm$2, same geographic region, etc.), $\mathcal{U}$ represents the user population, and τ controls the similarity concentration temperature. This objective maximizes mutual information between demographically similar users while maintaining separation from dissimilar counterparts through the denominator's hard negative mining over all non-positive pairs.

The complete optimization objective combines both components:

$$\mathcal{L} = \underbrace{(1-\lambda)\mathcal{L}_{\text{gen}}}_{\text{Individual fidelity}} + \underbrace{\lambda\mathcal{L}_{\text{CL}}}_{\text{Population structure}}. \tag{6}$$

This dual-objective formulation yields three key advantages: (1) geometrically meaningful embeddings where user similarity correlates with both demographic alignment and behavioral consistency, (2) improved sample efficiency through knowledge transfer between similar users, and (3) inherent regularization that prevents overfitting to individual transaction outliers.

4 Experiments

4.1 Real-World Deployment & Performance Validation

We validate the core contributions of PANTHER through its large-scale deployment at WeChat Pay, focusing on two key aspects: pretraining efficacy (next-transaction prediction) and downstream fraud detection capabilities. This deployment addresses the challenges identified in Section 3.1, showcasing how PANTHER can enhance fraud detection and personalized user services in real-world setting.

4.1.1 Next-Transaction Prediction Benchmark

Task & Dataset: models are pretrained on 5.3B anonymized transactions (38M users over 6 months) to model user-specific behavior. The raw transactions data, consisting six attributes (amount, merchant category, etc.), are tokenized by 60k interpretable tokens with the structured tokenization scheme.

Experimental Setup: In the next-transaction prediction task, we evaluate PANTHER's ability to predict the next transaction based on a user's historical data. The model leverages the SPRM and unified user-profile embeddings for this purpose. We employ two widely-adopted evaluation metrics: HR@K (Hit Ratio at K) and NDCG@K (Normalized Discounted Cumulative Gain at K). Specifically, HR@K measures the fraction of test instances in which the ground-truth item appears among the top-K predicted items. NDCG@K assesses the ranking quality by assigning higher weights to relevant items placed at top positions, normalized by the ideal discounted gain. The code is available at https://github.com/WeChatPay-Pretraining/PANTHER.

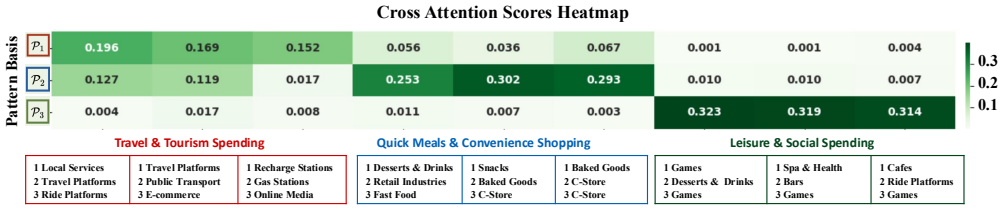

Figure 3: Demonstration of the cross attention score between recurring three-gram user payment behaviors and the learnable pattern prototypes.

We benchmark PANTHER against several strong baseline models, including Transformer [24], SASRec [15], and HSTU [12].

Key Findings: As shown in Table 1, PANTHER outperforms baseline models significantly. The SPRM module alone improves HR@1 by 14.9%, highlighting the value of sequential behavior patterns. Incorporating user profiles via learnable positional encoding boosts HR@1 by an additional 5.5%. Adding contrastive learning objectives results in a total HR@1 improvement of 25.6%, demonstrating the effectiveness of context-aware embeddings and knowledge transfer.

Visualization of Learned Patterns: Figure 3 shows how PANTHER maps consecutive user behaviors to learned behavior prototypes (P1, P2, P3). Each column represents a sequence of three consecutive behaviors identified as recurring patterns in the user's history. The heatmap values indicate the strength of alignment between behaviors and prototypes, with higher values signifying stronger matches. This visualization highlights how the model captures and recognizes repeating transaction motifs over time.

4.1.2 Hybrid Real-Time Fraud Detection

›We evaluate PANTHER's fraud detection performance through a 10-day full-traffic A/B test on WeChat Pay's production system, using the DeepFM model [7] as the baseline. DeepFM relies on handcrafted features and a TextCNN encoder for sequence processing. PANTHER operates in a hybrid configuration, combining offline-pretrained user-profile embeddings (e_u) and behavioral anomaly scores (Δ) with real-time transaction features (Equation 3), achieving substantial recall improvements, as shown in Figure 4.

The hybrid PANTHER model improves fraud recall by 109.5% at the 0.01% threshold, with smaller gains at higher thresholds (0.1% **+38.6%**, 1% +12.1%). The larger gains at extreme thresholds highlight the model's effectiveness in detecting low-frequency, high-risk fraud cases using personalized embeddings (e_u) and behavioral anomaly scores (Δ).

Deployment Strategy: This hybrid approach introduces minimal overhead (5ms higher than the baseline), while offering three key advantages: (1) personalized fraud detection via user-specific embeddings, (2) explainable anomaly detection through interpretable deviation scores, and (3) scalable production deployment by separating compute-intensive pretraining from real-time inference.

4.1.3 Merchant Risk Assessment via Behavior Sequence Pretraining

In addition to fraud detection, we developed a framework for merchant risk assessment based on behavior sequence pretraining. This approach identifies merchants potentially involved in fraud by analyzing deviations in user behavior at the merchant level, incorporating components for behavior prediction, deviation measurement, risk aggregation, and merchant classification.

Next Behavior Deviation Scoring. Deviation is measured as the standardized difference between the predicted likelihood of a behavior and the user's typical behavior distribution. It reduces false positives for users with naturally varied transaction patterns: $\Delta_{u,m} = \frac{P_u(m) - \mu_u}{\sigma_u}$, where μ_u and σ_u are the mean and Std Dev of predicted probabilities across all potential behaviors for user u.

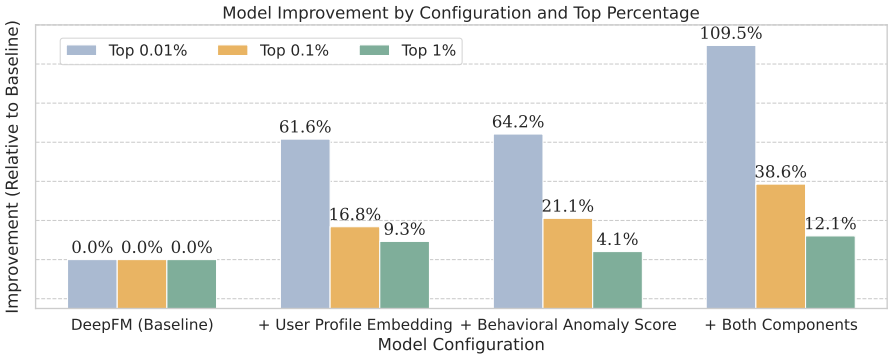

Figure 4: Fraud recall improvement at operational thresholds (Recall@Top-K).

Merchant Risk Scoring and Classification. We aggregate deviation scores for all users transacting with merchant m and compute the quantile-based risk score:

$$R_m = \text{Quantile}_q(\{\Delta_{u,m} \mid u \in \mathcal{U}_m\}),$$

where $\mathcal{U}_m$ is the set of users interacting with merchant m. Robust statistical features are extracted and used as inputs for classifier (e.g., XGBoost) to label merchants as high or low risk.

Practical Impact. The framework showed strong performance in practice, achieving 76% accuracy for flagged high-risk merchants – a significant improvement over the baseline method's 50% accuracy.

4.2 Benchmark Performance

After validating PANTHER's real-world viability, we evaluate its performance on public benchmark datasets to quantify its improvements over existing methods. This section compares PANTHER against strong baselines on public transaction datasets and recommendation benchmark, demonstrating consistent performance gains across domains beyond the WeChat Pay setting.

4.2.1 Datasets & Tasks

We evaluate PANTHER's pretraining performance on four datasets, including two financial datasets and two recommendation datasets, to assess its effectiveness and generalizability. For downstream tasks, we compare fraud detection performance on CCT and recommendation tasks on Yelp (since only these datasets include user profile features).

1. **Credit Card Transactions (CCT)** [25]: A synthetic dataset with 20 million transactions from 2,000 users, used for fraud detection with embedded anomalies.
2. **MBD-mini** [26]: An anonymized banking dataset tracking monthly product purchases, focusing on repetitive consumer behaviors.
3. **MovieLens-1M** [27]: A widely-used recommendation dataset with 1 million user-item interactions, ideal for evaluating sequential models in non-financial domains.
4. **Yelp** [28]: A public recommendation dataset containing millions of user-business interactions (e.g., restaurant reviews), commonly used for sequential recommendation tasks.

Each dataset is split chronologically into training, validation, and test subsets.

Implementation details: We apply the same model hyperparameters to the open benchmark datasets as those used for the WeChat Pay dataset. Full implementation details are provided in Section B and our supplementary code.

4.2.2 Next Payment Behavior Prediction Benchmarking

We assess PANTHER on next behavior prediction. As summarized in Table 2 for MBD-mini, CCT, MovieLens-1M and Yelp, it consistently outperform strong baselines like SASRec and HSTU.

Table 2: Experimental results of various methods on transaction and recommendation datasets.

	Dataset	Method	HR@1	HR@5	HR@10
Payment	MBD-mini	Transformer	.0441	.1735	.2864
		SASRec	.0442(+0.23%)	.1729(-0.35%)	.2871(+0.24%)
		HSTU	.0432(-2.04%)	.1713(-1.27%)	.2851(-0.45%)
		PANTHER	**.0454(+2.95%)**	**.1766(+1.79%)**	**.2923± .0005(+2.06%)**
	CCT	Transformer	.0500	.1381	.1979
		SASRec	.0447(-10.60%)	.1387(+0.43%)	.1994(+0.76%)
		HSTU	.0537(+7.40%)	.1419(+2.75%)	.2025(+2.32%)
		PANTHER	**.0576(+15.20%)**	**.1554(+12.53%)**	**.2176± .0004(+9.95%)**
Recommendation	MovieLens-1M	Transformer	.0579	.1826	.2773
		SASRec	.0627(+8.29%)	.1906(+4.38%)	.2853(+2.88%)*
		HSTU	.0699(+20.73%)	.1972(+7.99%)	.3043(+9.74%)
		PANTHER	**.0705(+21.76%)**	**.2103(+15.17%)**	**.3078± .0007(+11.01%)**
	Yelp	Transformer	.0851	.1769	.2300
		SASRec	.0875(+2.74%)	.1820(+2.91%)	.2385(+3.71%)
		HSTU	.0879(+3.21%)	.1878(+6.19%)	.2496(+8.55%)
		PANTHER	**.0929(+9.17%)**	**.2204(+24.63%)**	**.2924± .0006(+27.14%)**

* The higher HR@10 reported SASRec [15] is due to sampled negative evaluation (∼100 items). Our protocol follows HSTU with full-ranking over ∼3,700 items.

Table 3: Recommendation Performance on Yelp

Model	HR@1	HR@5	NDCG@5
DCN	0.612	0.963	0.534(baseline)
+ PANTHER	0.773	0.982	0.692(+29.6%)

Table 4: Fraud detection performance on CCT

Model	Recall	Accuracy	F1 Score
TabBERT-MLP	-	-	0.760 (baseline)
TabBERT-LSTM	-	-	0.860 (+13.2%)
DCN	0.931	0.871	0.888 (+ 16.8%)
+ PANTHER	0.978	0.896	0.911 (+ 19.9%)

Key Observations: On WeChat Pay, PANTHER-large achieves a 25.56% HR@1 improvement over Transformer, reflecting its capability to model sporadic, large-scale financial transactions. On MBD-mini and CCT, PANTHER improves HR@1 by 2.95% and 15.2%, respectively, demonstrating its broad applicability to other payment transaction data. On the MovieLens-1M and Yelp datasets, PANTHER improves HR@1 by 21.8% and 9.17% over Transformer, surpassing HSTU and showing strong generalization beyond payment data. Overall, PANTHER demonstrates robustness across domains and highlights the advantages of larger model configurations for complex user-item interactions.

4.2.3 Hybrid Fraud Detection & Recommendation

We evaluate the PANTHER on downstream fraud detection and recommendation tasks, by introducing its pretrained embeddings (e_u) and next-behavior predictions ($\hat{X}_t$) to the baseline models.

Hybrid Recommendation on Yelp. We demonstrate that PANTHER not only excels in fraud detection but also performs effectively in recommendation tasks. For example, on the Yelp dataset, the NDCG@5 metric improves by 29.6% over the DCN baseline (Table 3), highlighting the value of pretrained embeddings in enhancing recommendation performance.

Hybrid Fraud Detection on CCT. On the CCT dataset, incorporating PANTHER's pretrained embeddings and predictions improves fraud detection recall by 19.9% over the TabBERT-MLP ([29]) baseline (Table 4), enhancing the model's effectiveness in detecting fraudulent activity.

These results confirm that PANTHER's hybrid method significantly boosts performance across tasks by leveraging learned user profiles and behavior predictions.

4.3 Transferability

We examine PANTHER's ability to transfer learned representations across new users, datasets, and domains, demonstrating the power of generative pretraining in low-label and cross-domain scenarios. PANTHER shows exceptional transferability, with a 301.4% improvement over cold-start baselines when transferring across user demographics. Additionally, fine-tuning on external datasets after pretraining on WeChat Pay leads to an average HR@1 improvement of 16.66% on MBD-mini and CCT, showcasing the model's adaptability across diverse transaction contexts. More detailed results and experiments are provided Section C. These findings highlight PANTHER's ability to generalize effectively with minimal retraining, making it highly suitable for real-world applications with sparse labeled data.

4.4 Summary of Experimental Findings

Our experiments show that PANTHER consistently outperforms strong baselines across tasks. It achieves robust next-transaction prediction with noisy, sparse data by leveraging the Sequence Pattern Recognition Module and adaptive user embeddings. PANTHER also demonstrates strong transferability, excelling across diverse domains, including transaction data (CCT, MBD-mini) and recommendation tasks (MovieLens-1M, Yelp). Real-world deployment at WeChat Pay shows a 38% improvement in fraud recall at top 0.1%. Ablation studies confirm that key components, like SPRM and contrastive learning, significantly enhance performance. Overall, it proves to be a versatile, scalable solution for sequential behavior modeling, with strong generalization across domains.

5 Conclusions

We introduced PANTHER, a generative pretraining framework that addresses real-world payment data complexities by combining noise suppression, pattern recognition, and user personalization. Through token space compression and innovative attention mechanisms, PANTHER uncovers subtle, cyclic behaviors and incorporates long-term user context, producing high-fidelity user embeddings. This design supports critical financial applications such as fraud detection, credit scoring, next-payment prediction, and user segmentation. Beyond industry use, PANTHER highlights an important direction for the machine learning community: leveraging massive unlabeled transaction logs to enhance efficiency and adaptability. As financial and e-commerce data volumes rise, PANTHER's integration of generative modeling and personalization enables more secure, accurate, and user-centric services. However, its lack of interpretability is a limitation, as complex representations may hinder transparency in high-stakes applications. Future work will focus on improving explainability for broader applicability in regulated domains.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 62406192), Opening Project of the State Key Laboratory of General Artificial Intelligence (No. SKLAGI2024OP12), and Tencent WeChat Rhino-Bird Focused Research Program.

References

[1] Enterprise Apps Today. Global digital payment platforms user statistics (2022). https://www.enterpriseappstoday.com/stats/online-payment-statistics.html, 2022. Accessed: 2025-01-23.

[2] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[3] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. Ernie 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8968–8975, 2020.

[4] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020.

[5] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[6] Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Michael He, et al. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. *arXiv preprint arXiv:2402.17152*, 2024.

[7] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: A factorization-machine based neural network for ctr prediction. *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1725–1731, 2017.

[8] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*, pages 1–7. 2017.

[9] Bin Liu, Chenxu Zhu, Guilin Li, Weinan Zhang, Jincai Lai, Ruiming Tang, Xiuqiang He, and Zhenguo Li. Autofis: Automatic feature interaction selection in factorization models for click-through rate prediction. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2636–2645, 2020.

[10] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Cai, Yang Liu, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, and Kun Gai. Deep interest evolution network for click-through rate prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(1):5941–5948, 2019.

[11] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys)*, pages 191–198, 2016.

[12] Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Michael He, et al. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. *arXiv preprint arXiv:2402.17152*, 2024.

[13] Balazs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*, 2016. URL https://arxiv.org/abs/1511.06939.

[14] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining (WSDM)*, pages 567–575, 2018. doi: 10.1145/3159652.3159656.

[15] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *IEEE International Conference on Data Mining (ICDM)*, pages 1236–1242, 2018. doi: 10.1109/ICDM.2018.00152.

[16] Fei Sun, Jun Liu, and Jian Wu. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformers. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 1441–1450, 2019. doi: 10.1145/3357384.3357895.

[17] Shubham Rajput, Mingxuan Chen, and Fei Sun. Tiger: Token-based generative retrieval for sequential recommendation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 789–798, 2023. doi: 10.48550/arXiv.2306.01234.

[18] Majid Zolghadr, Mohsen Jamali, and Jiawei Zhang. Diffurecsys: Diffusion-based generative modeling for sequential recommendation. *Proceedings of the ACM Web Conference (WWW)*, pages 2156–2165, 2024. doi: 10.1145/3545678.3557899.

[19] Junyi Chen, Lu Chi, Bingyue Peng, and Zehuan Yuan. Hllm: Enhancing sequential recommendations via hierarchical large language models for item and user modeling. *arXiv preprint arXiv:2409.12740*, 2024.

[20] Richard J Bolton and David J Hand. Statistical fraud detection: A review. *Statistical Science*, 17(3):235–255, 2002. doi: 10.1214/ss/1042727940.

[21] Humberto Acevedo-Viloria, Juan Martinez, and Maria Garcia. Relational graph convolutional networks for financial fraud detection. *IEEE Transactions on Knowledge and Data Engineering*, 33(7):1357–1370, 2021. doi: 10.1109/TKDE.2020.3007655.

[22] Hao Wang, Wei Zhang, and Fei Sun. Heterogeneous graph neural networks for user behavior prediction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 765–774, 2019. doi: 10.1145/3357384.3357907.

[23] Yoon Kim. Convolutional neural networks for sentence classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, 2014.

[24] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[25] Eric Altman et al. Credit card transactions (cct) dataset, 2019. URL https://www.kaggle.com/datasets/ealtman2019/credit-card-transactions. Accessed: 2025-01-31.

[26] AI Lab et al. Mbd-mini dataset, 2020. URL https://huggingface.co/datasets/ai-lab/MBD-mini. Accessed: 2025-01-31.

[27] GroupLens et al. Movielens-1m dataset, 2000. URL https://grouplens.org/datasets/movielens/1m/. Accessed: 2025-01-31.

[28] Yelp. Yelp dataset, 2014. URL https://business.yelp.com/data/resources/open-dataset/.

[29] 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 *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3565–3569. IEEE, 2021.

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: The abstract accurately claims the the background, introduced method and its performance.

   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: We briefly discussed the limitations of PANTHER in the conclusion chapter.

   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: [NA]

Justification: The paper does not include theoretical results. Focus of this work is on the problem formulation and design of pre-training framework.

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: Our experiments on public benchmark datasets are reproducible. Experiment configurations are fully disclosed.

   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: [Yes]

Justification: The code to reproduce the experiments on public datasets is submitted.

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: Settings of experiments on public datasets are specified in the paper and appendix.

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: [Yes]

Justification: The standard error of the repeated experiments is reported for our proposed method, providing a clear indication of the statistical significance and reliability of the results.

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: [Yes]

   Justification: The required compute resources are reported in the appendix.

   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: The research conforms with the NeurIPS Code of Ethics.

   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: Potential societal impacts are discussed in the conclusion section.

    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: [NA]

    Justification: The paper poses no such risks.

    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: Existing assets used in the paper are cited and credited. Licenses are included in the appendix.

    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: [Yes]

    Justification: The released code of the paper are well documented.

    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 paper does not involve crodwsourcing nor research with human subjects.

    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: The paper does not involve crowdsourcing nor research with human subjects.

    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: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    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.

Appendix

A Notation Table

We summarize the frequently used notations in Table 5.

Table 5: Notations used in this paper

Notation	Descrpition
$\mathbf{X}_u = [x_1, x_2, \ldots, x_L]$	The payment behavior sequence of length L for a user $u \in \mathcal{U}$.
$x_t \in \mathcal{V}$	The transaction token at time step t, $\mathcal{V}$ is the set of possible transaction tokens.
$P(y = 1 \mid x_{\text{new}}, \mathbf{c}_{\text{new}}, \mathbf{X}_u)$	Fraud probability for new transaction x_{new} with contextual feature $\mathbf{c}_{\text{new}}$, given $\mathbf{X}_u$. The y is an indicator variable for fraud.
$\mathcal{L}_{\text{gen}}$	The generative loss function for next behavior prediction
$P_\theta(x_t \mid x_{<t}, x_{\text{profile}})$	The probability of the t-th transaction x_t given the previous transactions $x_{<t}$, user profile features x_{profile} and model parameters θ.
e_u	The learned user profile embedding for user u.
$g_\phi(\cdot)$	Real-time fraud detection model that integrates features including recent patterns, predicted deviations, and user profiles for risk prediction, typically a DCN network.
$\psi(\cdot)$	The embedding function for transaction context features, typically in the form of a linear layer.
$f_{\text{enc}}(\cdot)$	Sequence encoder for user's recent short-term payment behaviors, typically a TextCNN model.
$\Delta(x_{\text{new}}, \hat{x}_{L+1})$	The deviation between the observed new transaction x_{new} and the predicted next behavior $\hat{x}_{L+1}$.
$\tau = (c_i, a_j, m_k, r_l) \in \mathcal{C} \times \mathcal{A} \times \mathcal{M} \times \mathcal{R}$	Example definition of transaction token formed by the Cartesian product of the contextual and counterparty features.
$\mathrm{P}_{\text{SPRM}} \in \mathbb{R}^{m \times d}$	The set of m learnable prototypes of embedding dimension d in the Sequence Pattern Recognition Module (SPRM).
$H \in \mathbb{R}^{T \times d}$	Token embeddings of d dimension for an input sequence of length T
$\mathcal{L}_{\text{CL}}$	The contrastive loss function for user profile embedding learning
$(i, j) \in P_{pos}$	positive pairs of users sharing similar demographic attributes
e_i, e_j	User profile embeddings for users i and j, respectively.
λ	The hyper-parameter balancing the two losses: $\mathcal{L}_{\text{gen}}$, $\mathcal{L}_{\text{CL}}$
R_m	The risk score for merchant m, computed from the user deviation scores interacting with the merchant.

B Implementation Details

In this section, we provide the full implementation details for MBD-mini, CCT, Movielens, and Yelp datasets, including model configurations, training procedures, and tokenization methods.

B.1 CCT, Yelp, and MBD-mini Experiments

For the CCT, Yelp, and MBD-mini dataset, we configured PANTHER with **4 layers** and **2 attention heads**. Training was conducted with a **batch size of 128** at learning rate $\mathbf{1 \times 10^{-3}}$. The training utilized a single GPU over a span of 2 hours on CCT, 6 hours on Yelp, and 12 hours hours on MBD-mini.

The Transformer, SASRec, and HSTU models were configured with the same learning rate, number of layers, and batch size as PANTHER.

B.2 MovieLens-1M Experiments

For the MovieLens-1M experiments, PANTHER was trained with a **batch size of 128** and a learning rate of 1×10^{-3}. Specifically, PANTHER was built with a **2-layer**, **1-head** configuration. All baseline models were configured identically to PANTHER.

These experiments followed the same training pipeline as WeChatPay, with hierarchical tokenization and discretization applied to transaction attributes. Given the smaller dataset sizes, training was completed within a **shorter time frame** while preserving model scalability.

B.3 Tokenization of benchmark datasets

For the CCT dataset, user behavior tokens are constructed from payment amounts, payment methods, and merchant categories, resulting in a vocabulary of 16,847 tokens. User profiles include available card and card-holder information such as card brand, card type, and user age. For the MBD-mini dataset, user behavior tokens are derived from transaction attributes including amount, currency, event type, and the source and destination types, yielding a vocabulary of 40,791 distinct tokens.

In the Yelp dataset, user behavior tokens are formed by combining a business's city, category, star rating, and review count, where continuous features are bucketized. The original vocabulary of 40K tokens can be compressed to 17K tokens, covering 95% of all user-business interactions. User profiles consist of attributes such as the number of friends, number of reviews, and average star ratings. For the Movielens dataset, movie IDs are directly used as behavior tokens.

B.4 Efficiency Evaluation

We compare GPU memory usage and inference time of the SPRM against the Transformer baseline across increasing sequence lengths, in Table 6. These experiments illustrate how SPRM scales more efficiently in both memory consumption and latency, particularly when handling longer sequences where the Transformer model fails due to memory overflow.

Table 6: Efficiency comparison between Transformer and SPRM across different sequence lengths

Sequence Length	Transformer Memory(GB)	Transformer Inference Time(s)	SPRM Memory(GB)	SPRM Inference Time(s)
1024	8.4	72.2	1.9	65.4
2048	29.7	113.9	3.2	70.3
4096	OOM	-	5.7	74.1
8192	OOM	-	10.8	93.8

C Transferability of PANTHER

We examine PANTHER 's capacity to transfer learned representations to new users, new datasets, and new domains. The experiments showcase the advantage of generative pretraining in label scarcity and cross-domain scenarios.

Table 7: Comparison of PANTHER with cold-start and user transfer settings on WeChatPay.

	WeChatPay			
	HR@1	HR@5	HR@10	HR@50
Cold-Start	.0581	.0834	.0963	.1308
User-Transfer	**.2332(+301%)**	**.4494(+439%)**	**.5417(+463%)**	**.7280(+457%)**

C.1 User-Level Transferability

We pre-train PANTHER on one set of WeChatPay users and evaluate on another disjoint set for a cold-start recommendation setting. As shown in Table 7, the pre-trained PANTHER significantly outperforms the cold-start baseline, demonstrating its ability to preserve learned behavioral patterns and adapt to new users with minimal retraining. This finding is crucial for financial applications where new users frequently arrive and labeled data are sparse.

C.2 Data-Level Transferability

Table 8: Experimental results of PANTHER on CCT and MBD-mini datasets after pretraining on WeChatPay dataset. The values in parentheses indicate the relative improvement compared to directly finetuning on these datasets

Dataset	HR@1	HR@5	HR@10	HR@50
MBD-mini	.0539(+16.66%)	.1980(+10.43%)	.3143(+6.72%)	.7123(+3.82%)
CCT	.0248(+13.76%)	.0729(+10.79%)	.1050(+6.59%)	.2706(+4.08%)

To assess cross-dataset adaptability, we pre-train PANTHER on WeChatPay and fine-tune it on MBD-mini and CCT. Table 8 shows an average HR@1 improvement of 16.66%, demonstrating effective transfer of our generative pretraining across diverse transaction contexts. The model retains valuable cross-data signals, such as user spending patterns, even when the merchant or product space changes.

D Ablation Experiments

To examine the sensitivity of PANTHER to the balance between $\mathcal{L}_{\text{gen}}$ and $\mathcal{L}_{\text{CL}}$, we vary the loss coefficient λ and report the corresponding performance. The results are summarized in Table 9.

Table 9: Performance on WeChat Pay under different values of the loss coefficient λ.

λ	HR@1	HR@5	HR@10
0.1	0.2452	0.4837	0.5647
0.2	0.2441	0.4862	0.5644
0.4	0.2435	0.4869	0.5653
0.8	0.2430	0.4832	0.5629