# SCALING TRANSFERABLE BEHAVIORAL REPRESENTATIONS: BRIDGE THE SEMANTIC GAP FROM NEXT-EVENT PREDICTION TO DOWNSTREAM TASKS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Internet platforms collect abundant behavioral data from a broad population, yet labels for downstream decision tasks are often scarce, delayed, and expensive. A central question is whether large-scale behavioral pre-training can learn representations that transfer across the substantial semantic gap between self-supervised behavior prediction and downstream objectives. We systematically study how pre-training objective, data scale, and model scale shape such cross-task transfer. We introduce BRIDGE (Behavioral Representations for Improving Downstream Generalization and Efficiency), which learns transferable representations of people from large-scale heterogeneous behavioral sequences together with structured features (static or slowly varying person-level attributes, commonly termed user-profile features in recommendation and credit-risk modeling) through autoregressive next-event prediction (NEP). We compare NEP with representative masked, multi-token, distribution-based, and embedding-based objectives. Objective preference depends on scale: embedding-based pre-training leads at the smaller evaluated scale, whereas NEP overtakes it at the larger scale. Despite the weak semantic alignment between NEP and downstream decision objectives, the learned representations transfer effectively to tasks such as credit-risk assessment and payment fraud detection. Across the evaluated scales, downstream out-of-time (OOT) performance improves consistently with both pre-training data volume and model size, while lower pre-training loss is associated with better downstream discrimination. With only $10\%$ of the credit-risk labels, BRIDGE improves the OOT Kolmogorov–Smirnov (KS) statistic by $5.31\%$ relative to end-to-end (E2E) supervised training under the same label budget and even surpasses E2E trained on the full labeled dataset. These results show that scaling behavioral pre-training can substantially improve cross-task transfer and label efficiency, while suggesting that the preferred pre-training objective itself can depend on scale.

## 1 INTRODUCTION

Scaling has become a central empirical principle of modern pre-training, with larger datasets, larger models, and greater compute budgets consistently improving performance on the objectives used for training. In language modeling, Kaplan et al. (2020) characterize how predictive loss varies with model size, data volume, and compute, while Hoffmann et al. (2022) show that balancing model and data scale improves compute efficiency. Similar trends have emerged in sequential recommendation: HSTU examines recommendation quality as training compute increases (Zhai et al., 2024), and OneRec demonstrates generative recommendation at production scale (Zhou et al., 2025). These studies primarily evaluate the training objective or closely related tasks. Less is known about the *transferability of scaling*: whether scaling self-supervised behavioral pre-training improves downstream performance when both the objective and data distribution differ substantially and labeled samples are scarce. This question is particularly relevant to credit-risk assessment and fraud detection, where behavioral histories are abundant but reliable outcome labels are limited and delayed.

Behavioral pre-training and downstream risk prediction differ in objective, label horizon, and data distribution, a mismatch we call the *semantic gap* (Appendix G). Nevertheless, future behavior and risk outcomes may depend on shared latent factors reflected in behavioral histories and structured



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.

features. We address this question with **BRIDGE**. BRIDGE learns representations from these inputs through NEP, then uses the pre-trained embeddings and encoder to initialize full fine-tuning on a smaller labeled dataset. We evaluate transfer to credit-risk assessment and payment fraud detection against an E2E supervised baseline trained from scratch with the same architecture. This comparison isolates the benefit of pre-training from differences in model capacity. We organize the study around three research questions.

**RQ1: Does behavioral pre-training transfer across the semantic gap, and which objectives transfer best?** We compare NEP with masked-behavior prediction (MBP) (Sun et al., 2019; Wu et al., 2020), multi-token prediction (MTP) (DeepSeek-AI, 2024), multi-scale stochastic distribution prediction (MSDP) (Fu et al., 2023), and Bootstrapping Your Behavior (BYB) (Wu et al., 2025). BYB performs best at smaller evaluated data scale, while NEP leads at the larger scale, motivating its use in BRIDGE. Under strict temporal isolation, BRIDGE outperforms E2E on both credit-risk assessment and payment fraud detection, providing evidence of transfer across the semantic gap.

**RQ2: How do pre-training data volume and model size affect downstream generalization?** Using NEP, we vary pre-training data volume and model size while keeping the downstream fine-tuning task fixed. Across the evaluated scales, increasing either data volume or model size improves OOT performance, and lower pre-training loss is associated with better downstream discrimination. These results show that the benefits of scaling extend to a downstream task with a substantially different objective.

**RQ3: To what extent does behavioral pre-training improve downstream label efficiency?** We fine-tune BRIDGE on progressively smaller fractions of the credit-risk labels and compare it with E2E at matched label budgets, using E2E trained on the full labeled dataset as an additional reference. With only $10\%$ of the labels, BRIDGE surpasses both baselines in OOT KS. This result demonstrates that representations learned from abundant behavioral data can support effective risk prediction with substantially less task-specific supervision.

Together, these experiments characterize how pre-training objective, data scale, and model size influence cross-task transfer and label efficiency. BRIDGE provides evidence that scaling behavioral pre-training can improve downstream decisions despite substantial differences in task objectives and data distributions.

## 2 Related Work

### 2.1 Supervised Modeling

For click-through-rate prediction, the Deep Interest Network models candidate-conditioned behavioral interests (Zhou et al., 2018), and search-based interest models retrieve relevant events from long histories (Pi et al., 2020). More recently, SparseCTR combines personalized sequence chunking with sparse attention for efficient long-history modeling (Lai et al., 2026). Behavioral sequences also support cross-domain fraud detection with limited target-domain labels (Zhu et al., 2020). These task-specific approaches motivate learning from abundant unlabeled behavior before fine-tuning for risk prediction with scarce or delayed outcome labels.

### 2.2 Behavioral Pre-training

Behavioral pre-training learns from sequence-derived supervision through different predictive and representation-learning objectives. NEP extends next-item prediction to behavioral events; MBP reconstructs hidden events from context, as in BERT4Rec and PTUM (Sun et al., 2019; Wu et al., 2020); and MTP, adapted here from language modeling, supervises multiple future positions (DeepSeek-AI, 2024). MSDP uses distributional targets at multiple behavioral scales (Fu et al., 2023), while BYB predicts future-window embeddings without a manually selected discrete behavior vocabulary and evaluates financial-risk transfer (Wu et al., 2025). Related approaches learn transferable representations through contrastive objectives (Wu et al., 2022; Cheng et al., 2021) or universal sequence pre-training (Hou et al., 2022). PANTHER combines generative transaction pre-training with downstream adaptation, including fraud detection at WeChat Pay (Li et al., 2025). For recommendation, progressive post-training separates downstream adaptation via linear probing and full fine-tuning from business-metric alignment via reinforcement fine-tuning (Choi et al., 2026).

Rather than introducing risk transfer itself, BRIDGE compares NEP, MBP, MTP, MSDP, and BYB under a shared scarce-label risk protocol to examine which objective transfers best and how this preference changes with pre-training data scale.

### 2.3 Scaling Laws

In language modeling, Kaplan et al. (2020) relate predictive loss to model size, data, and compute, while Hoffmann et al. (2022) study compute-efficient model–data allocation. Industrial recommendation systems also demonstrate scaling benefits, including HSTU (Zhai et al., 2024), FuXi-$\alpha$ (Ye et al., 2025), and OneRec (Zhou et al., 2025). Yet the evaluation target matters: Brüel Gabrielsson (2026) shows that loss-optimal configurations for behavioral event models need not optimize deployment ranking metrics, and Xu et al. (2026) find task-dependent empirical scaling ceilings and production constraints on downstream gains. Transfer scaling is likewise established: Hernandez et al. (2021) quantify effective data gains from language-model pre-training; ATLAS models multilingual transfer across pre-training and fine-tuning (Longpre et al., 2026); and Zivic et al. (2024) show that larger sequential recommenders retain benefits when fine-tuned on smaller task-specific domains. BRIDGE does not introduce downstream scaling gains or loss–metric mismatch itself. Instead, it examines how behavioral pre-training loss relates to *supervised risk decisions* under population shift and sparse labels, across data and model scales, with additional transfer evaluation on payment fraud detection.

## 3 Behavioral Pre-training Framework

### 3.1 Problem Formulation

We consider a common human-centered learning setting with a pronounced supervision imbalance. A large corpus records behavioral histories from a broad population and contains enough samples to support robust self-supervised pre-training. In contrast, a particular downstream task may concern a different concept, follow a different data distribution, and provide only a small labeled dataset. For example, a population-scale collection of viewing histories can supply abundant self-supervision through NEP, while datasets labeled for financial interest or credit fraud may contain far fewer training samples.

**Multi-domain sequential data.** Let $i$ index a sample, and let $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_M\}$ denote $M$ behavioral domains. The observations in domain $\mathcal{D}_d$ form a chronologically ordered sequence

$$S_i^{(d)} = \left(e_{i,1}^{(d)}, e_{i,2}^{(d)}, \ldots, e_{i,T_{i,d}}^{(d)}\right), \qquad d \in \{1, \ldots, M\}. \tag{1}$$

We preserve the temporal order within every domain and organize the domain streams according to a consistent logical order. Denoting this ordering operation by $\mathrm{Ord}$, the resulting multi-domain sequence is

$$S_i = \mathrm{Ord}\left(S_i^{(1)}, \ldots, S_i^{(M)}\right). \tag{2}$$

For example, when the streams are aligned by logical step, this operation can produce

$$S_i = \left[e_{i,1}^{(1)}, e_{i,1}^{(2)}, \ldots, e_{i,1}^{(M)}, e_{i,2}^{(1)}, e_{i,2}^{(2)}, \ldots, e_{i,2}^{(M)}, \ldots\right]. \tag{3}$$

More generally, $\mathrm{Ord}$ respects event time and uses the logical domain order to resolve aligned events. This abstraction accommodates heterogeneous domains without tying the formulation to a particular application or event schema.

**Structured features.** We additionally associate sample $i$ with the explicitly structured feature set

$$\mathcal{F}_i = (a_{i,1}, a_{i,2}, \ldots, a_{i,P}), \qquad a_{i,p} \in \mathcal{X}_p, \tag{4}$$

where each feature space $\mathcal{X}_p$ may be categorical or numerical. These features encode stable or slowly varying context and long-horizon summaries that may not be identifiable from a finite recent sequence alone. Recommendation models commonly combine behavioral sequences with structured

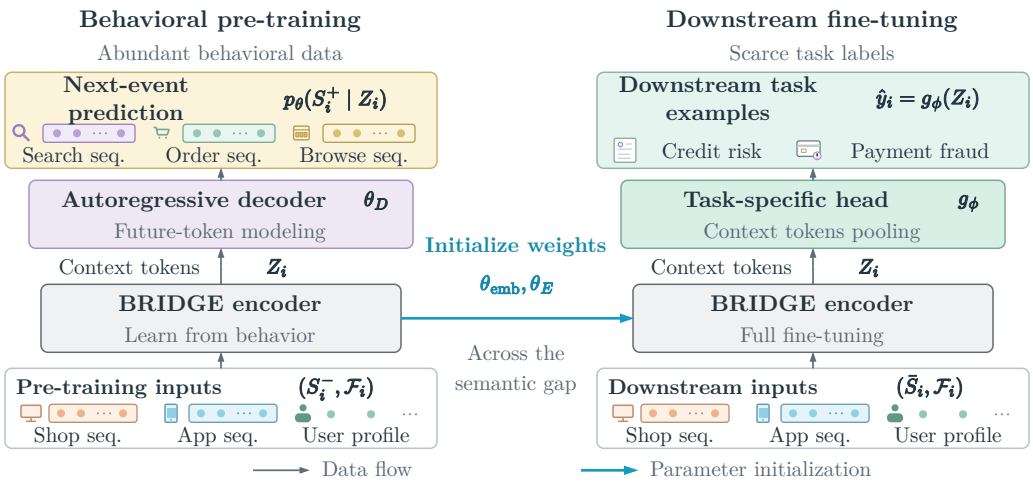



Figure 1: **Overview of BRIDGE. Left:** behavioral pre-training encodes past sequences $S_i^-$ and structured features $\mathcal{F}_i$ into $Z_i$ for autoregressive NEP. **Right:** the pre-trained embeddings and encoder initialize downstream learning with available inputs $(\bar{S}_i, \mathcal{F}_i)$. The decoder is replaced by mean pooling and a task-specific head $g_\phi$, and the embeddings, encoder, and new head are optimized jointly. The horizontal arrow denotes parameter initialization. Credit risk and payment anti-fraud are example downstream tasks; the icons are illustrative.

features for this reason (Zhou et al., 2018). Together, the two modalities provide a more complete description of a sample than either modality in isolation.

The model receives $\big(S_i, \mathcal{F}_i\big)$ as input, where $S_i$ retains both event order and domain identity. We use the abundant, broadly distributed samples to pre-train a shared representation by next-event prediction, and then adapt that representation using a much smaller labeled set drawn from a potentially shifted downstream distribution. Our focus is not a domain-specific event schema or a specialized architecture, but the transferability, label efficiency, and scaling behavior of the learned representation across this pre-training–downstream discrepancy.

### 3.2 Architecture

Inspired by the LONGER architecture (Chai et al., 2025), we propose BRIDGE as a pre-training–then–fine-tuning pipeline for multi-domain sequential and structured data. The architectural components are intentionally standard: our aim is to test whether representation transfer and scaling persist when the pre-training and downstream tasks differ substantially in both objective and data distribution. Figure 1 summarizes the two-stage pipeline and the transfer of pre-trained embedding and encoder parameters to downstream tasks.

#### 3.2.1 Model Structure

For each pre-training sample, we draw a prediction time $t_0$ and form ordered past and future windows of equal duration $T$:

$$S_i^- = \mathrm{Subseq}(S_i; t_0 - T \le \tau(e) < t_0)\,, \qquad S_i^+ = \mathrm{Subseq}(S_i; t_0 \le \tau(e) < t_0 + T)\,, \tag{5}$$

where $\tau(e)$ is event time and Subseq preserves event order. The encoder combines the past sequence with structured features $\mathcal{F}_i$. Let $H_i = \mathrm{SeqEnc}(S_i^-)$ be the full embedded sequence, $U_i = \mathrm{StructEnc}(\mathcal{F}_i)$ the structured tokens, and $R_{i,K}^{(d)}$ the most recent $K$ embedded tokens from domain $d$. Structured and recent behavioral tokens initialize a short set of global queries,

$$Q_i^{(0)} = [U_i; R_{i,K}^{(1)}; \ldots; R_{i,K}^{(M)}]. \tag{6}$$

An $n$-layer Transformer encoder (Vaswani et al., 2017) refines these queries. We use two architectural changes to support multi-domain, long-sequence transfer. First, encoder blocks *interleave*

cross-attention from global queries to the full behavioral sequence with self-attention among the global queries, repeating one cross-attention block followed by three self-attention blocks. This lets a compact query set repeatedly gather information from a long history without applying full self-attention to every behavioral token. Second, a *long-range residual connection* carries the initial queries directly to the encoder output:

$$\begin{aligned} Z_i &= \mathrm{RMSNorm}\Big(Q_i^{(n)} + \mathrm{Gate}_{\theta_E}(Q_i^{(0)})\Big) \\ &= \mathrm{Enc}_{\theta_{\mathrm{emb}},\theta_E}(S_i^-, \mathcal{F}_i). \end{aligned} \tag{7}$$

Here, $\mathrm{Gate}_{\theta_E}$ applies a learnable elementwise gate to the initial queries, and RMSNorm denotes root-mean-square normalization. This connection is intended to preserve broad, early cross-domain information when NEP disproportionately emphasizes domains that are easiest to predict; otherwise, the learned initialization may underrepresent domains important to downstream tasks. Encoder depth $n$ and width vary across the model scales in Table 3. During pre-training, a four-layer Transformer decoder conditions on $Z_i$ to predict $S_i^+$; during fine-tuning, the decoder is discarded. In both stages, the encoder receives sequential and structured features in the same form, although the available downstream sequence need not be split at $t_0$. Embeddings, projections, and attention blocks are detailed in Appendix B. The detailed model architecture is shown in Figure 4 in Appendix A.

### 3.2.2 PRE-TRAINING

Denote the temporally ordered, domain-aware future sequence by

$$S_i^+ = \big((d_{i,1}, z_{i,1}), \ldots, (d_{i,T_i^+}, z_{i,T_i^+})\big), \tag{8}$$

where $d_{i,q}$ is the domain of the $q$-th future token and $z_{i,q} \in \mathcal{V}_{d_{i,q}}$. Conditioned on the encoder state $Z_i$, the decoder autoregressively models its joint probability through next-event factorization,

$$p_\theta(S_i^+ \mid S_i^-, \mathcal{F}_i) = p_\theta(S_i^+ \mid Z_i) = \prod_{q=1}^{T_i^+} p_{\theta,d_{i,q}}(z_{i,q} \mid (d,z)_{i,<q}, Z_i)\,, \tag{9}$$

where $(d,z)_{i,<q}$ is the ordered prefix of domain–token pairs. The first equality holds because the decoder accesses the encoder input only through the deterministic state $Z_i = \mathrm{Enc}_{\theta_{\mathrm{emb}},\theta_E}(S_i^-, \mathcal{F}_i)$. If $o_{i,q}$ is the four-layer decoder state at position $q$, each domain uses its own output projection and softmax,

$$p_{\theta,d}(\cdot \mid o_{i,q}) = \mathrm{softmax}(W_d o_{i,q} + b_d) \in \Delta^{|\mathcal{V}_d|}, \qquad d = 1, \ldots, M. \tag{10}$$

Consequently, heterogeneous domains do not share a unified output vocabulary: both input tokenization and next-event normalization use the corresponding domain-specific vocabulary $\mathcal{V}_d$. The parameter set $\theta = (\theta_{\mathrm{emb}}, \theta_E, \theta_D)$ contains the embeddings, encoder, decoder, and domain-specific output heads. Let $\mathcal{I}_{\mathrm{pre}}$ index the pre-training samples and $N_{\mathrm{pre}} = \sum_{i\in\mathcal{I}_{\mathrm{pre}}} T_i^+$ be the total number of future tokens. We minimize the average next-event negative log-likelihood

$$\mathcal{L}_{\mathrm{pre}}(\theta) = -\frac{1}{N_{\mathrm{pre}}} \sum_{i\in\mathcal{I}_{\mathrm{pre}}} \sum_{q=1}^{T_i^+} \log p_{\theta,d_{i,q}}(z_{i,q} \mid (d,z)_{i,<q}, Z_i)\,. \tag{11}$$

Temporal order supplies supervision without downstream labels, allowing training to scale with the abundant behavioral corpus. The key intuition is that predicting future behavior from past behavior and structured context requires the encoder to capture latent factors that govern the future actions of the person represented by a sample. Such factors can reflect future interests for recommendation or risk-relevant behavioral tendencies for credit-risk and fraud detection, even though none of these downstream labels is used during pre-training.

### 3.2.3 FINE-TUNING

Let $\bar{S}_i$ denote all sequential information available for sample $i$ at the downstream prediction time. Unlike the artificial split used to construct pre-training pairs, $\bar{S}_i$ is not restricted to $S_i^-$ defined by the pre-training cutoff $t_0$. Let $\mathcal{I}_{\mathrm{down}}$ index the labeled set $\{(\bar{S}_i, \mathcal{F}_i, y_i)\}_{i\in\mathcal{I}_{\mathrm{down}}}$, where $y_i$ is observed after the input window and may represent, for example, whether the person represented by the

sample accepts a recommendation or is exposed to fraud risk. As illustrated on the right of Figure 1, we discard the pre-training decoder. If $Z_i = (Z_{i,1}, \ldots, Z_{i,G_i})$ contains $G_i$ refined global tokens, we aggregate them by mean pooling and attach a task-specific head $g_\phi$:

$$Z_i = \mathrm{Enc}_{\theta_{\mathrm{emb}},\theta_E}(\bar{S}_i, \mathcal{F}_i), \qquad \hat{y}_i = g_\phi(Z_i). \tag{12}$$

For a general downstream task, let $\ell_{\mathrm{task}}$ denote its appropriate supervised loss. Fine-tuning minimizes

$$\mathcal{L}_{\mathrm{down}}(\theta_{\mathrm{emb}}, \theta_E, \phi) = \frac{1}{|\mathcal{I}_{\mathrm{down}}|} \sum_{i \in \mathcal{I}_{\mathrm{down}}} \ell_{\mathrm{task}}(\hat{y}_i, y_i). \tag{13}$$

For binary outcomes such as fraud-risk detection, this general form reduces to binary cross-entropy with $\ell_{\mathrm{task}}(\hat{y}_i, y_i) = -y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i)$; other task heads and losses support multiclass classification, regression, ranking, or other downstream objectives.

# 4 Experimental Setup

**Data and splits.** Pre-training uses heterogeneous behavioral logs from an internet platform, including browsing, search, ordering, and login events. We evaluate downstream transfer on credit-risk assessment and payment fraud detection. The **Test** split is a held-out subset from the same time window as the training data, with no overlap in samples, and measures in-time generalization. The **OOT** split comes from a strictly later period and measures generalization under temporal distribution shift. We enforce strict temporal isolation to prevent label leakage; Appendix E describes the data and temporal-isolation protocol for credit risk.

**Evaluation metrics.** For pre-training, we report **validation loss** and **next-event accuracy**. For downstream credit risk, we report the area under the receiver operating characteristic curve (**AUC**) and **KS**, two standard ranking metrics for risk decisions. We report KS as a percentage (i.e., the raw statistic scaled by 100), while AUC is reported as a decimal. Implementation details are provided in Appendix H.

**Comparison protocol and reporting.** We compare behavioral pre-training followed by full fine-tuning with the same architecture trained end-to-end from scratch, using the same downstream optimization settings. Unless otherwise noted, downstream results are averaged over 3 independent runs with different random seeds. To preserve confidentiality, we report approximate data and model sizes rather than exact counts; when absolute business metrics are sensitive, we report metric changes instead.

# 5 Experimental Results

## 5.1 Bridging the Semantic Gap (RQ1)

To answer RQ1, we examine whether objective choice and increased behavioral pre-training data yield representations that transfer beyond behavioral prediction. We first compare five objectives on credit risk using the same encoder architecture and downstream fine-tuning protocol: NEP, MBP, MTP, MSDP, and BYB. Each objective is evaluated after pre-training on approximately 1B and 10B samples. The E2E baseline uses the same downstream architecture and optimization but is trained from scratch without behavioral pre-training.

**Objective preference depends on data scale.** Table 1 shows that BYB, which learns consistency between past and future behavior embeddings, achieves the highest downstream KS and AUC at $\sim$1B samples, whereas NEP leads on both metrics at $\sim$10B. Across these two scales, NEP gains 0.61 KS points, compared with 0.13 for BYB. Thus, NEP is not uniformly the best objective, but its stronger performance at the larger evaluated scale motivates its use in BRIDGE. A plausible interpretation is that consistency between past and future behavior embeddings may stabilize representation learning when observations are limited. On the one hand, BYB's consistency objective favors information about a person that remains invariant across past and future windows, potentially underemphasizing behavioral changes between them. This may limit the timely capture of a person's evolving behavior, even though such changes can be crucial for many downstream tasks. On

Table 1: Comparison of pre-training objectives at two data scales.

| | ∼1B samples | | ∼10B samples | |
|---|---|---|---|---|
| Method | OOT KS (↑) | OOT AUC (↑) | OOT KS (↑) | OOT AUC (↑) |
| E2E (no pre-training) | 64.20 | 0.8990 | 64.20 | 0.8990 |
| NEP (ours) | 64.97 | 0.9039 | **65.58** | **0.9066** |
| MBP | 64.02 | 0.8988 | 65.09 | 0.9042 |
| MTP | 64.74 | 0.9025 | 65.47 | 0.9059 |
| MSDP | 64.39 | 0.9004 | 64.88 | 0.9026 |
| BYB | **65.17** | **0.9043** | 65.30 | 0.9050 |

Table 2: Anti-fraud transfer: pre-training on ∼30B samples followed by fine-tuning, compared with E2E training from scratch. Both use a trainable encoder. Δ denotes relative improvement over E2E.

| Metric | Pre-train + fine-tune | E2E | Δ |
|---|---|---|---|
| OOT KS (↑) | **27.43** | 21.68 | +26.5% |
| OOT AUC (↑) | **0.6868** | 0.6456 | +6.4% |
| Test KS (↑) | **46.60** | 35.92 | +29.7% |
| Test AUC (↑) | **0.8066** | 0.7404 | +8.9% |

the other hand, with abundant observations, direct NEP may instead capture richer behavioral regularities, reducing reliance on additional target-design constraints. By using the observed behavioral tokens directly as prediction targets, NEP avoids compressing future behavior into an embedding or an aggregated distribution, preserving finer-grained information in the supervision signal.

**Behavioral prediction transfers to downstream risk tasks.** Behavioral pre-training improves performance on both credit risk and payment fraud detection relative to training from scratch. On credit risk (Table 1), NEP improves OOT KS over E2E from 64.20 to 64.97 at ∼1B samples and to 65.58 at ∼10B, corresponding to gains of 0.77 and 1.38 KS points. OOT AUC likewise increases from 0.8990 to 0.9039 and 0.9066. We further evaluate payment fraud detection, whose labels, time horizons, and failure modes differ from those of credit default (Zhu et al., 2020). We fine-tune the backbone pre-trained on ∼30B samples and compare it with an E2E baseline trained from scratch on fraud-detection data, with a trainable encoder in both cases. As shown in Table 2, pre-training improves every reported metric, including OOT KS by +26.5% (27.43 vs. 21.68) and OOT AUC by +6.4% (0.6868 vs. 0.6456). Together, these results support RQ1: NEP pre-training on abundant behavioral samples can help bridge the semantic gap from behavioral prediction to distinct downstream risk tasks. To assess whether these benefits also hold in real-world production environments, we conduct online experiments across three business lines, with details and results reported in Appendix D.

### 5.2 Scaling Pre-training Data and Model Size (RQ2)

To answer RQ2, we fix the pre-training objective to NEP and vary data volume and model size, holding the other axis fixed. We evaluate pre-training performance and assess downstream credit-risk performance after full fine-tuning.

**Data scaling.** We fix the backbone at the reference configuration (∼50M parameters) and increase the number of pre-training samples across five scales, approximately 1B, 5B, 10B, 20B, and 30B samples, expanding population coverage rather than extending histories for a fixed cohort. As shown in Figure 2, downstream KS rises from 64.97 to 66.11 (+1.14 KS points), while AUC increases from 0.9039 to 0.9092. The increase from 1B to 5B samples adds +0.54 KS points, compared with +0.13 for the increase from 20B to 30B samples. Over the full range, the trend is approximately log-linear, averaging about +0.23 KS points per doubling of pre-training data. Pre-training validation loss falls from 8.96 to 8.62 between 1B and 5B samples, with smaller reductions thereafter; lower loss is associated with higher downstream KS/AUC across the evaluated scales.

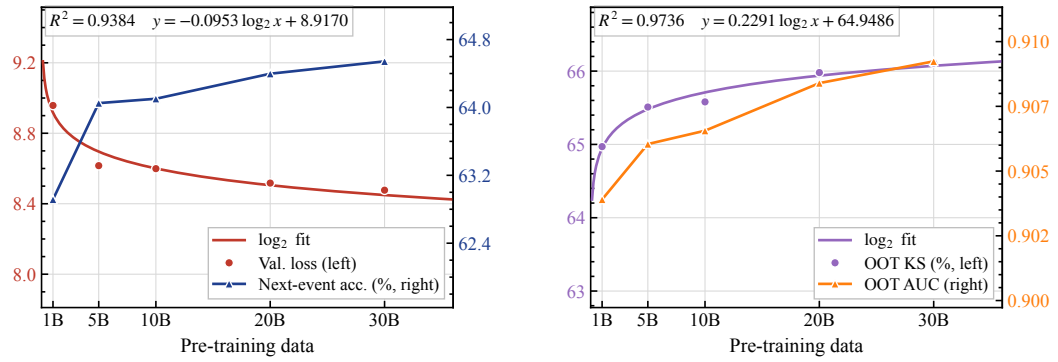



Figure 2: Data scaling at the reference ∼50M model size, with full-label downstream fine-tuning. **Left:** pre-training validation loss (↓) and next-event accuracy (↑). **Right:** downstream OOT KS and OOT AUC, both of which increase monotonically with pre-training data.

Table 3: Model scaling with ∼10B pre-training samples and full-label downstream fine-tuning. For each configuration we report the encoder width $d_{\text{model}}$, depth $n$ (total encoder layers: self-attention plus interleaved cross-attention), and their ratio $d_{\text{model}}/n$. "Best ep." denotes the selected fine-tuning epoch, at which all metrics are reported.

| # Params | $d_{\text{model}}$ | Depth $n$ | $d_{\text{model}}/n$ | OOT KS (↑) | OOT AUC (↑) | Best ep. |
|---|---|---|---|---|---|---|
| ∼10M | 128 | 4 | 32.0 | 64.11 | 0.9003 | 38 |
| ∼15M | 192 | 7 | 27.4 | 65.20 | 0.9045 | 26 |
| ∼30M | 288 | 9 | 32.0 | 65.48 | 0.9062 | 14 |
| ∼50M | 384 | 10 | 38.4 | 65.58 | 0.9066 | 9 |
| ∼100M | 512 | 16 | 32.0 | **65.62** | **0.9068** | 5 |

**Model scaling.** We fix the pre-training data at ∼10B samples and vary the backbone from ∼10M to ∼100M parameters across five configurations (Table 3). We increase both the per-layer hidden width $d_{\text{model}}$ and encoder depth $n$ while retaining the same global-query design, interleaved attention pattern, and decoder. Thus, this axis changes capacity rather than the overall pre-training task or architecture. Both OOT KS and AUC increase monotonically across the five configurations, with diminishing gains. KS rises from 64.11 to 65.20, 65.48, 65.58, and 65.62, while AUC increases from 0.9003 to 0.9045, 0.9062, 0.9066, and 0.9068, respectively. Increasing model size from ∼10M to ∼50M yields +1.47 KS points, whereas doubling the ∼50M model to ∼100M adds only +0.04 KS points and +0.0002 AUC. Although the ∼100M model achieves the highest values on both metrics, the ∼50M configuration offers a reasonable balance between parameter count and performance at this data scale.

**Complementary analyses.** Beyond data and model scaling, Appendix C examines whether longer pre-training on a fixed corpus with a fixed model configuration further improves downstream transfer. The overall gains across checkpoints suggest that continued behavioral pre-training translates into better downstream discrimination. In practical downstream applications, pre-trained embeddings may also be reused as input features without fine-tuning the encoder for each task. Motivated by this use case, Appendix F trains only lightweight prediction heads on frozen representations at different encoder depths for credit-risk prediction, assessing their utility without encoder adaptation and informing the choice of representation layer.

### 5.3 Reducing Downstream Label Requirements (RQ3)

To answer RQ3, we distinguish performance gains at a fixed label budget from the labels needed to match full-label E2E training. On the credit-risk task, we fine-tune on nested subsets containing 1%, 5%, 10%, 25%, 50%, and 100% of the available labeled data. We compare pre-trained backbones with an E2E baseline of the same architecture trained from scratch. Figure 3 reports Test/OOT KS

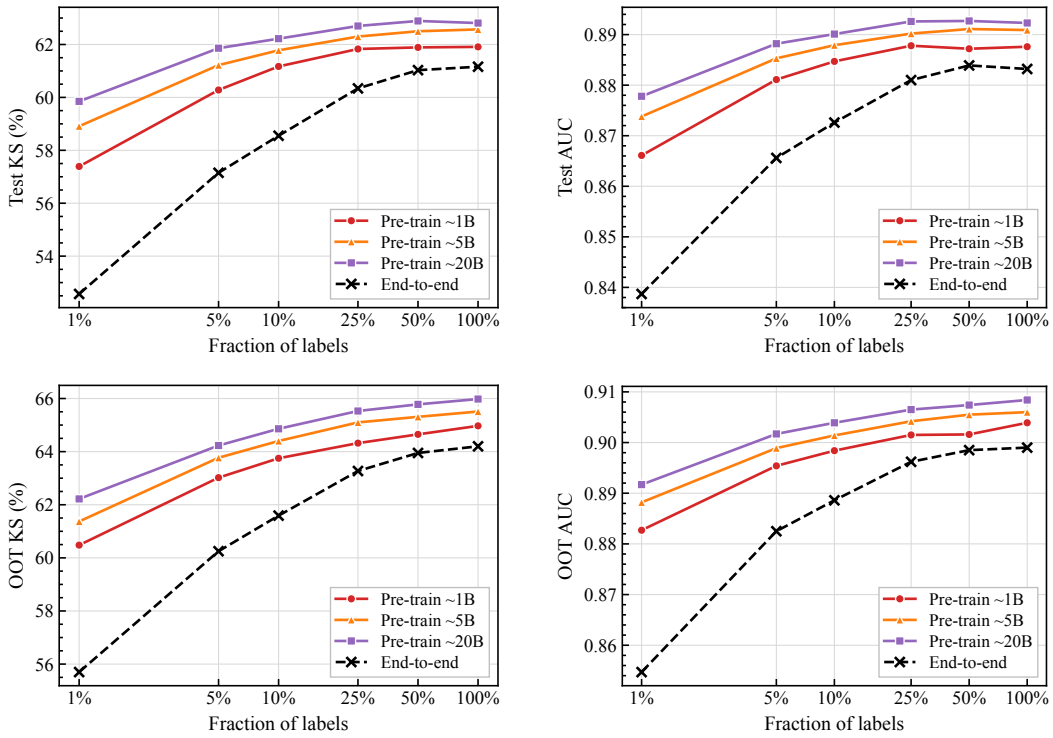



Figure 3: Credit-risk label efficiency at three pre-training data scales (∼1B/∼5B/∼20B), compared with E2E training (dashed). The label-fraction axis is logarithmic. **Top:** Test KS and AUC. **Bottom:** OOT KS and AUC. All pre-trained curves remain above E2E across the evaluated fractions.

and AUC for three pre-training data scales (∼1B to ∼20B) and the E2E baseline; we focus below on OOT KS to assess transfer under temporal distribution shift.

**Gains at the same label budget.** At 10% labels, the ∼20B pre-trained backbone achieves OOT KS of 64.86, compared with 61.59 for E2E: a gain of 3.27 KS points, or 5.31% relative improvement. The advantage is larger when labels are scarcer: at 1%, OOT KS is 62.22 versus 55.70 (+6.52 points), whereas at 100% the gain is 1.78 points (65.98 vs. 64.20). Across the evaluated label fractions, OOT KS is non-decreasing with pre-training data scale from ∼1B through ∼5B to ∼20B, extending the data-scaling pattern in Section 5.2 to limited-label settings.

**Matching full-label E2E with fewer labels.** A separate comparison measures the reduction in label requirements. With ∼20B pre-training samples, fine-tuning on 10% of the risk labels yields OOT KS of 64.86, exceeding E2E trained with 100% of the labels (64.20) by 0.66 KS points. Even 5% of the labels yields 64.23, exceeding full-label E2E by 0.03 points. Thus, among the evaluated label fractions, 5% suffices to surpass the full-label E2E reference.

## 6 Conclusion

Under strict temporal isolation, behavioral pre-training bridges the semantic gap between NEP and downstream risk tasks, outperforming capacity-matched E2E training on credit risk and payment fraud detection, with greater benefits when labels are scarce. Across NEP, MBP, MTP, MSDP, and BYB, BYB performs best with smaller pre-training corpora, whereas NEP leads at larger evaluated scales. Lower pre-training loss is associated with higher downstream KS, and frozen-encoder probing favors shallower representations for credit-risk prediction. Future work will extend behavioral pre-training across tasks and platforms, investigate how data diversity, model scale, and supervision shape transfer, and address continual adaptation, privacy, fairness, and reliability.

## REFERENCES


Rickard Brüel Gabrielsson. Scaling laws for behavioral foundation models over user event sequences. *arXiv preprint arXiv:2606.05257*, 2026.

Zheng Chai, Qin Ren, Xijun Xiao, Huizhi Yang, Bo Han, Sijun Zhang, Di Chen, Hui Lu, Wenlin Zhao, Lele Yu, Xionghang Xie, Shiru Ren, Xiang Sun, Yaocheng Tan, Peng Xu, Yuchao Zheng, and Di Wu. LONGER: Scaling up long sequence modeling in industrial recommenders. In *Proceedings of the 19th ACM Conference on Recommender Systems (RecSys)*, 2025.

Mingyue Cheng, Fajie Yuan, Qi Liu, Xin Xin, and Enhong Chen. Learning transferable user representations with sequential behaviors via contrastive pre-training. In *IEEE International Conference on Data Mining (ICDM)*, 2021.

Oseong Choi, Hoeinn Kim, Jihoon Lee, Byungsoo Kang, and Taeyeong Jang. Progressive alignment of recommender foundation model through multi-phase post-training. *arXiv preprint arXiv:2608.06792*, 2026. Accepted to RecSys 2026, Industry Track, as reported in the arXiv record.

DeepSeek-AI. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Chilin Fu, Weichang Wu, Xiaolu Zhang, Jun Hu, Jing Wang, and Jun Zhou. Robust user behavioral sequence representation via multi-scale stochastic distribution prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 4567–4573, 2023.

Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. Scaling laws for transfer. *arXiv preprint arXiv:2102.01293*, 2021.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. Towards universal sequence representation learning for recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Weijiang Lai, Beihong Jin, Di Zhang, Siru Chen, Jiongyan Zhang, Yuhang Gou, Jian Dong, and Xingxing Wang. Unleashing the potential of sparse attention on long-term behaviors for CTR prediction. In *Proceedings of the ACM Web Conference 2026*, 2026. doi: 10.1145/3774904.3792808.

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. *arXiv preprint arXiv:2510.10102*, 2025. Revised March 2026.

Shayne Longpre, Sneha Kudugunta, Niklas Muennighoff, I-Hung Hsu, Isaac Caswell, Alex Pentland, Sercan Arik, Chen-Yu Lee, and Sayna Ebrahimi. ATLAS: Adaptive transfer scaling laws for multilingual pretraining, finetuning, and decoding the curse of multilinguality. In *International Conference on Learning Representations (ICLR)*, 2026. OpenReview conference record.

Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM)*, 2020.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Chuhan Wu, Fangzhao Wu, Tao Qi, Jianxun Lian, Yongfeng Huang, and Xing Xie. PTUM: Pre-training user model from unlabeled user behaviors via self-supervision. *arXiv preprint arXiv:2010.01494*, 2020.

Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. UserBERT: Pre-training user model with contrastive self-supervision. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022.

Weichang Wu, Xiaolu Zhang, Jun Zhou, Yuchen Li, and Wenwen Xia. Bootstrapping your behavior: A new pretraining strategy for user behavior sequence data. *arXiv preprint arXiv:2506.11053*, 2025.

Qiuling Xu, Ko-Jen Hsiao, and Moumita Bhattacharya. Towards generalizable and efficient large-scale generative recommenders. *arXiv preprint arXiv:2605.23312*, 2026. RecSys 2026, as reported in the arXiv record.

Yufei Ye, Wei Guo, Jin Yao Chin, Hao Wang, Hong Zhu, Xi Lin, Yuyang Ye, Yong Liu, Ruiming Tang, Defu Lian, and Enhong Chen. FuXi-$\alpha$: Scaling recommendation model with feature interaction enhanced transformer. In *Companion Proceedings of the ACM on Web Conference 2025*, 2025. doi: 10.1145/3701716.3715448.

Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Jiayuan He, Yinghai Lu, and Yu Shi. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. In *International Conference on Machine Learning (ICML)*, 2024.

Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2018.

Guorui Zhou, Jiaxin Deng, Jinghao Zhang, Kuo Cai, Lejian Ren, Qiang Luo, et al. OneRec technical report. *arXiv preprint arXiv:2506.13695*, 2025.

Yongchun Zhu, Dongbo Xi, Bowen Song, Fuzhen Zhuang, Shuai Chen, Xi Gu, and Qing He. Modeling users' behavior sequences with hierarchical explainable network for cross-domain fraud detection. In *Proceedings of The Web Conference (WWW)*, 2020.

Pablo Zivic, Hernan Vazquez, and Jorge Sánchez. Scaling sequential recommendation models with transformers. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1567–1577, 2024. doi: 10.1145/3626772.3657816.

## A Model Architecture Overview

Figure 4 presents the detailed model architecture, illustrating how the sequential and structured embeddings, the global-query initialization, the interleaved cross- and self-attention encoder blocks, the long-range residual connection, and the domain-specific decoder heads compose the end-to-end pipeline. Appendix B specifies each component in turn.

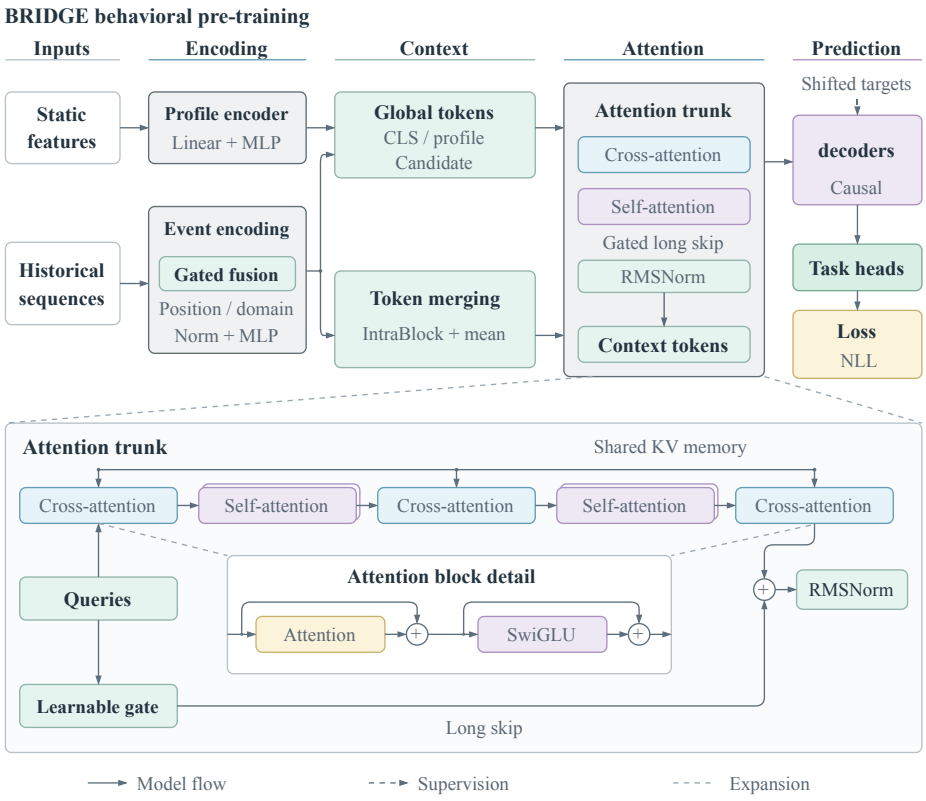



Figure 4: Detailed model architecture of BRIDGE, showing the sequential and structured embeddings, the global-query initialization, the interleaved cross- and self-attention encoder blocks, the long-range residual connection, and the domain-specific decoder heads used for autoregressive NEP. NLL denotes negative log-likelihood, KV denotes key–value, and CLS denotes a classification token.

## B Architecture Details

**Sequential and structured embeddings.** Each sequential domain $\mathcal{D}_d$ has a token vocabulary $\mathcal{V}_d$, embedding map $E_d$, and projection multilayer perceptron (MLP) $P_d$. The pair $(E_d, P_d)$ is shared across events in that domain; projections map every domain to the common width $d_{\text{model}}$. For an event $e_{i,t}^{(d)} \in \mathcal{V}_d$,

$$h_{i,t}^{(d)} = P_d\Big(E_d(e_{i,t}^{(d)})\Big) \in \mathbb{R}^{d_{\text{model}}}, \qquad H_i = \text{SeqEnc}(S_i^-) = \text{Ord}\Big(\{h_{i,t}^{(d)} : e_{i,t}^{(d)} \in S_i^-\}_{d,t}\Big). \quad (14)$$

Here, SeqEnc denotes embedding, projection, and chronological/logical ordering, not an additional attention network. Each structured feature $a_{i,p}$ has its own pre-token embedding $E_p^{\mathcal{F}}$ and projection

MLP $P_p^{\mathcal{F}}$, without parameter sharing across structured fields:

$$u_{i,p} = P_p^{\mathcal{F}}\left(E_p^{\mathcal{F}}(a_{i,p})\right), \qquad U_i = \mathrm{StructEnc}(\mathcal{F}_i) = [u_{i,1}; \ldots; u_{i,P}]. \tag{15}$$

Let $R_{i,K}^{(d)}$ contain up to the $K$ most recent embedded events from domain $d$. Concatenating $U_i$ and all $R_{i,K}^{(d)}$ along the token dimension gives $Q_i^{(0)}$, the initial global-query sequence. Its length is at most $P + MK$, while $H_i$ retains the full input history. In Figure 4, "Profile" refers to the structured tokens $U_i$, which encode a sample's static features, and "Candidate" refers to the recent-$K$ behavioral tokens $R_{i,K}^{(d)}$, not a separate recommendation candidate.

**Interleaved encoder attention.** The encoder has $n$ layers. Layers $\ell \in \{1, 5, 9, \ldots\} \cap \{1, \ldots, n\}$ use cross-attention from the global queries to the full behavioral memory; the intervening layers use global-query self-attention. Thus, each complete four-layer cycle comprises one cross-attention block and three self-attention blocks, and a final incomplete cycle uses its available layers. With $\bar{Q}_i^{(\ell-1)} = \mathrm{RMSNorm}(Q_i^{(\ell-1)})$ and $\bar{H}_i = \mathrm{RMSNorm}(H_i)$, the multi-head attention (MHA) and feed-forward network (FFN) updates are given below. The FFN uses a Swish-gated linear unit (SwiGLU):

$$A_i^{(\ell)} = \begin{cases} \mathrm{MHA}_{\mathrm{cross},\ell}\left(\bar{Q}_i^{(\ell-1)}, \bar{H}_i, \bar{H}_i\right), & \ell \equiv 1 \pmod 4, \\ \mathrm{MHA}_{\mathrm{self},\ell}\left(\bar{Q}_i^{(\ell-1)}, \bar{Q}_i^{(\ell-1)}, \bar{Q}_i^{(\ell-1)}\right), & \text{otherwise}, \end{cases} \tag{16}$$

$$\widetilde{Q}_i^{(\ell)} = Q_i^{(\ell-1)} + A_i^{(\ell)}, \qquad Q_i^{(\ell)} = \widetilde{Q}_i^{(\ell)} + \mathrm{SwiGLU}_\ell\left(\mathrm{RMSNorm}(\widetilde{Q}_i^{(\ell)})\right). \tag{17}$$

The first layer therefore attends to the full memory before any query-only self-attention. At every cross-attention layer, queries can revisit all input events; self-attention mixes information among the compact global tokens. If the history has $L_i$ tokens and there are $G_i$ global queries, a cross-attention block uses attention proportional to $G_i L_i$, and a query self-attention block to $G_i^2$, rather than applying $L_i^2$ self-attention to the full history at each layer. Finally, the initial queries pass through the learnable gate before being added to the final-layer queries, and RMSNorm normalizes the sum to obtain $Z_i$, as specified in Section 3.2 and illustrated in Figure 4. The same embedding and encoder operations are used on an available downstream sequence $\bar{S}_i$ during fine-tuning.

## C Effect of Pre-training Duration

We examine the effect of pre-training duration by evaluating downstream performance at checkpoints from epochs 1–20. As shown in Figure 5, downstream KS increases overall from $65.50$ at epoch 1 to $66.11$ at epoch 20, a gain of $0.61$ KS points. Although performance fluctuates across checkpoints, later checkpoints generally achieve higher KS, with the best result observed at epoch 20. These findings suggest that progress in behavioral pre-training translates into improved downstream discrimination.

## D Online Deployment

We deployed the fine-tuned downstream model in production to augment the existing risk-scoring model. The behavioral representation serves as an additional risk signal rather than replacing the existing system. We measure relative KS improvements over the production baseline across three independent business lines with different populations and decision policies.

Table 4 reports positive relative KS improvements on all three business lines: $+8.7\%$ on Business Line A, $+2.8\%$ on Line B, and $+3.4\%$ on Line C. Based on business estimates, these improvements are expected to generate incremental revenue of several hundred million RMB. The largest gain occurs on Business Line A, where labels are scarce. This observation is consistent with the offline label-efficiency results. The ordering of the three gains echoes the offline pattern of Section 5.3, in which the benefit of pre-training grows as labels become scarcer, providing cross-setting consistency between the offline analysis and online deployment.

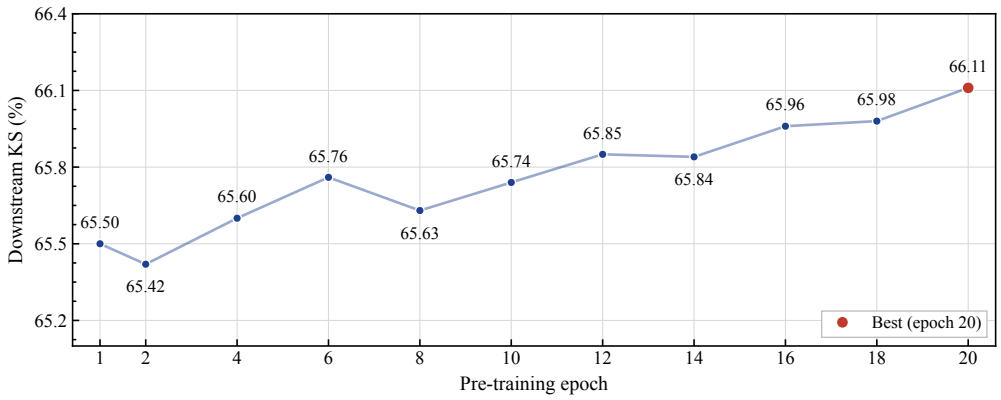



Figure 5: Downstream KS across pre-training checkpoints (approximately 30B pre-training samples and a 50M-parameter encoder).

Table 4: Relative KS improvement over the production baseline after adding the behavioral risk signal, across three business lines.

| Business line | Relative KS improvement over production baseline (↑) |
|---|---|
| Business Line A | +8.7% |
| Business Line B | +2.8% |
| Business Line C | +3.4% |

## E Data and Temporal Isolation

Pre-training uses heterogeneous behavioral logs collected from an internet platform, including browsing, search, ordering, and login events. The downstream task is a real-world credit-risk task. We enforce strict **temporal isolation** to prevent label leakage. Behavioral features are drawn from an observation window ending at $T_{\text{behavior}}$. The credit decision is made at $T_{\text{decision}}$, and the default outcome is observed at $T_{\text{outcome}}$. These times satisfy

$$T_{\text{behavior}} < T_{\text{decision}} < T_{\text{outcome}}. \tag{18}$$

No information from the decision or outcome windows is allowed to enter the behavioral representation. Within the training period, we further set aside an in-time **Test** split: a held-out subset of samples drawn from the *same time window* as the training data (an in-time holdout, disjoint from the training samples), used to evaluate generalization under the training-time distribution. We also hold out an **OOT** period that strictly follows the training period to assess temporal robustness under distribution shift. Thus the Test split reports in-time generalization, whereas the OOT split reports generalization under temporal shift; unless otherwise noted, headline results use OOT.

## F Layer-wise Analysis of Transferable Representations

As a complementary analysis, we examine how useful frozen representations at different encoder depths are for credit-risk discrimination. Unlike the full fine-tuning experiments above, this probe evaluates risk information accessible to a lightweight prediction head without adapting the encoder.

**Setup.** We fix the ∼50M encoder pre-trained on ∼20B samples, which has 10 layers. For each layer $l$, we freeze the entire encoder, attach a lightweight binary risk head to that layer's output embedding, and train it on labeled credit-risk data. We report KS under this frozen-encoder protocol, with layers indexed from 1 to 10. The comparison characterizes performance under the chosen probe rather than an intrinsic, task-independent quality of each representation.

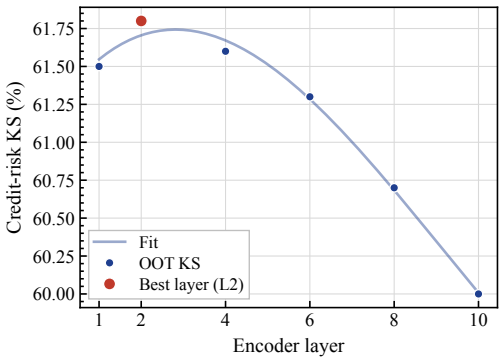



Figure 6: Layer-wise credit-risk probing with a frozen encoder. Among the displayed layers, KS peaks at layer 2 and declines at greater depths; the best displayed layer is highlighted.

As shown in Figure 6, among the displayed layers (1, 2, 4, 6, 8, and 10), KS increases from 61.50 at layer 1 to a peak of 61.80 at layer 2, then decreases to 61.60, 61.30, 60.70, and 60.00 at layers 4, 6, 8, and 10, respectively. The best-performing layer exceeds the deepest displayed layer by 1.80 KS points. These results favor shallow representations among the displayed layers under the frozen-encoder probing protocol. One possible interpretation is that deeper layers become more specialized for NEP, while shallower layers retain information more accessible to the risk head; the probe alone does not establish this mechanism. The findings motivate checking intermediate layers when using frozen embeddings, but do not determine the best representation depth after full fine-tuning.

## G Defining the Semantic Gap

We use *semantic gap* to denote the mismatch between the self-supervised pre-training task and the downstream objective along three concrete dimensions, rather than as an informal notion. (i) *Objective mismatch*: pre-training optimizes a next-event likelihood over a large behavioral vocabulary, whereas the downstream task optimizes a binary risk-classification objective; the two are neither identical nor a re-parameterization of each other. (ii) *Label-horizon delay*: the downstream label is realized only after a long outcome horizon ($T_{\text{behavior}} < T_{\text{decision}} < T_{\text{outcome}}$), so the supervised signal is temporally detached from the behaviors that predict it, unlike next-event targets that are immediately available from the sequence. (iii) *Distribution shift*: the labeled population, feature distribution, and the concept being predicted (default risk) differ from those governing generic next-event behavior, so the pre-training and downstream data are not drawn from the same conditional distribution. This decomposition distinguishes our setting from standard transfer learning and domain adaptation, where the objective is typically preserved and only the input distribution shifts, and it anchors the comparisons made throughout the paper.

## H Implementation Details

**Comparison protocol and reporting.** We compare behavioral pre-training followed by full fine-tuning with the same architecture trained end-to-end from scratch, using the same downstream optimization settings. To preserve confidentiality, we report approximate data and model sizes rather than exact counts; the pre-training data scales are approximately 1B, 5B, 10B, 20B, and 30B samples. When absolute business metrics are sensitive, we report metric changes ($\Delta$) instead. The largest configuration is trained at the $\sim$30B-sample scale. Unless otherwise noted, all reported downstream results are averaged over 3 independent runs with different random seeds.

**Full fine-tuning and transfer.** Our main experiments use *full fine-tuning*: the embedding parameters $\theta_{\text{emb}}$, encoder parameters $\theta_E$, and task-head parameters $\phi$ are all optimized on the downstream labeled set. Pre-training provides the initialization for the embeddings and encoder rather than a frozen feature extractor. Although the encoder input has the same sequential-and-structured form in both stages, the learning problems differ substantially: pre-training is autoregressive prediction

over abundant future behavioral tokens, whereas fine-tuning optimizes a potentially unrelated objective on scarce labels from a shifted distribution. We compare this pre-training–then–fine-tuning procedure with training the identical architecture from scratch under the same downstream optimization settings. The resulting gain isolates the value of the pre-trained initialization and demonstrates transfer across the task and data-distribution discrepancies.

**Training configuration.** All models are trained on a distributed GPU cluster using data and model parallelism. Both stages use AdamW with weight decay $10^{-5}$, but they use *different* learning-rate schedules tailored to their respective optimization regimes. Within each sampled pre-training window, we retain at most $512$ past tokens per sequential domain for the encoder, use the $K = 8$ most recent tokens per domain to initialize the global queries, and retain at most $64$ future tokens for the decoder target. Pre-training runs for up to $25$ epochs with a batch size of $512$ at a learning rate of $5 \times 10^{-6}$. Downstream fine-tuning runs for up to $50$ epochs, with the forward pass performed in fp32 for numerical stability; it adopts a *smaller* learning rate together with a *warm-up* schedule, which we found stabilizes adaptation of the pre-trained weights and improves downstream fit. During downstream fine-tuning, the embeddings, encoder, and task-specific head are optimized jointly.

# I AI USE STATEMENT

In this work, we used generative AI tools to draft portions of the manuscript, to edit the paper for readability and language polishing, and to support literature retrieval and the identification of relevant prior work. We have not used generative AI tools for generating experimental results or claims. We have reviewed all AI-assisted work, verifying cited literature against original sources and checking generated text for accuracy. We take responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.