Towards a Densing Law for User Representation Learning at Billion-Scale Capacity

Bin Dou[1,*], **Junru Zhang**[1,2,*], **Zhaoyi Yuan**[1,2,*], **Wuliang Huang**[1], **Letian Gong**[1], **Baokun Wang**[1,†], **Huan Li**[2,†], **Yu Cheng**[1], **Weiqiang Wang**[1]

[1]DeepFind Team, Ant Group, [2]Zhejiang University

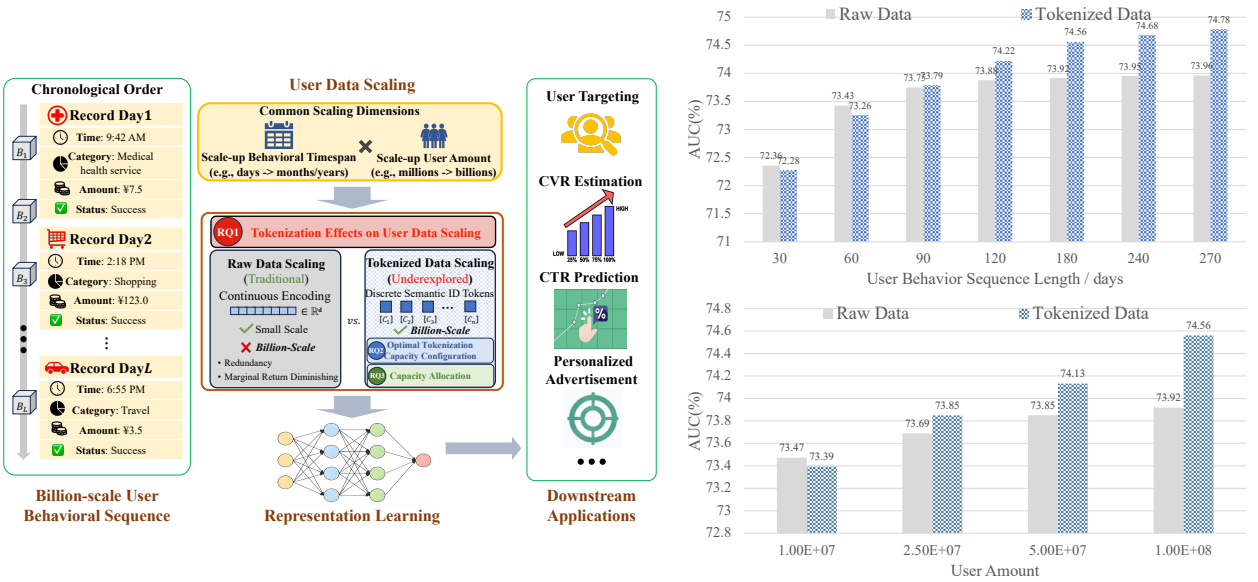
[*]Equal Contribution, [†]Corresponding author yike.wbk@antgroup.com, lihuan.cs@zju.edu.cn

User representation learning in real-world industrial scenarios is commonly scaled by increasing user amount, behavioral sequence length, and model size. However, existing methods suffer from two issues: (i) *Bottleneck for raw data scaling at billion-scale capacity*, as performance gains diminish with larger-scale raw text user behavioral input which can be alleviated by tokenization. (ii) *Lack of quantitative analysis of how tokenization configurations should scale with data size*. In this paper, we propose *User Behavioral Densing Law* for characterizing the quantitative relationship between data scale and the minimum sufficient tokenization capacity. Firstly, we conduct pilot study on raw & tokenized scaling comparison on billion-scale Alipay dataset, revealing the raw data scaling bottleneck and breakthrough via tokenization. To derive the scaling pattern governing the minimum sufficient tokenization configuration at different data scales, theoretical and experimental analyses are employed to summarize the quantitative scaling pattern. In addition, we propose a new tokenization method named ALGN inspired by the proposed Densing Law. Experimental evaluation over different data sources, tokenization methods and downstream tasks proves the generalizability and reliability of the Densing Law, which provides the guidance for configuration selection in large -scale user representation learning. Also ALGN surpasses existing baselines on both performance and efficiency.

1 Introduction

The remarkable success of deep representation learning has fundamentally transformed modern personalized platforms, with downstream performance consistently improving as user behavioral sequences and model parameters scale (Kang and McAuley, 2018; Sun et al., 2019; Pi et al., 2019). Today, unified user embeddings extracted from raw behavioral logs serve as a shared foundation across recommendation, CTR/CVR prediction, retrieval, and natural language user targeting (Xiao et al., 2025; Zhang et al., 2021). Figure 1 (a) sketches this paradigm. However, scaling along practical axes, including more users, longer historical windows, and larger encoders (Ardalani et al., 2022; Guo et al., 2024; Zhang et al., 2024b), presents significant challenges. User behavior logs are heavily shaped by daily routines and repeated habits, meaning data volume grows rapidly while novel, task-relevant information grows much more slowly. This discrepancy introduces a critical question:

Can we identify the key properties that make scaled models effective, thus improving representation quality without simply processing a greater volume of raw behavioral events?



(a) Billion-Scale User Representation Learning

(b) Performance — Training User Data Scaling Trend

Figure 1 (a) User representation learning pipeline. Once trained on user behavioral sequences, a representation model can support various downstream applications, including text-based retrieval for user targeting, U2U retrieval for recommendation system and classification for advertising and risk control scenarios. In industrial settings with billion-scale users, when scaling up input data volume, commonly via increasing behavioral timespan and user amount, models directly consume raw data (e.g., raw user behavioral text description), will suffer from performance saturation beyond practical scaling thresholds while tokenization will alleviate the limitation and enable sustained performance gains, as shown in (b).

While existing industrial systems rely on scaling raw sequence data, the empirical manifestation of behavioral redundancy and its relationship to downstream performance remain underexplored. In this work, we provide a comprehensive empirical analysis of what we term the *raw behavioral scaling wall*, using large-scale real-world Alipay PayBill data to ensure practical relevance and ecological validity.

Definition 1.1. We define the *raw behavioral scaling wall* as the phenomenon where incrementally scaling raw behavioral sequence length, user population, or model capacity yields strictly diminishing downstream returns, culminating in a *loss and quality dissociation* where pre-training loss decreases without improving downstream accuracy.

Through systematic evaluations across multiple scaling axes, we demonstrate that industrial user representation learning exhibits severe redundancy saturation. Extending observation windows or user bases primarily introduces repetitive, low-information events. Most notably, when scaling a user encoder from 0.2B to 0.4B parameters, we observe that the larger model merely fits redundant behavioral details rather than extracting additional task-relevant signal, leaving downstream representation accuracy almost unchanged.

Definition 1.2. We define *behavioral densing* as the process of transforming long raw histories into compact representations that preserve task-relevant distinctions while suppressing redundancy, thereby optimizing the amount of downstream-relevant signal each effective input unit carries.

This density-oriented perspective shows a fundamental insight: information density, rather than model capacity alone, is a key bottleneck limiting the expressivity of user encoders. We hypothesize

that moving user representations toward a better performance-cost frontier, governed by the *behavioral densing law*, can recover the gains that raw scaling fails to provide. This law rests on three principles: *Bounded Raw Gains*, where marginal utility drops as redundancy dominates; *Density Driven Scaling*, where denser representations shift the saturation frontier; and *Minimal Sufficient Capacity*, where optimal representations allocate only enough capacity to preserve necessary task-relevant information.

To test this hypothesis, we instantiate behavioral densing using residual quantized behavioral tokenization via RQ-VAE. This acts as a practical density operator, mapping repetitive event histories into compact sequences of discrete codes. Furthermore, recognizing that behavioral information is not uniformly distributed, we propose an *entropy-based variable length tokenization* strategy. By allocating more discrete codes to high-entropy segments and fewer to predictable routines, this mechanism explicitly adapts to behavioral complexity and approaches minimal sufficient representation capacity.

Our empirical evaluations demonstrate that under matched token and compute budgets, the tokenized representations overcome the raw scaling wall. The proposed framework yields consistent performance gains and improved encoder utilization on large-scale Alipay PayBill production data.

The key contributions of this work are:

1. **Scaling diagnosis and tokenization effects.** We identify diminishing marginal returns when scaling raw behavioral data, and reveal that tokenized data deliver more sustained gains thereby mitigating performance saturation at larger scales. Together with evidence of a model-capacity bottleneck, these findings indicate that data information density, rather than model parameter number alone, is a primary constraint on industrial user representation learning, especially at billion-scale data capacity.
2. **Predictive tokenization configuration recipe.** We formulate the optimal tokenization capacity configuration trajectory corresponding to input data volume as the *Behavioral Densing Law* for user representation learning, enabling required capacity configurations to be estimated from behavioral data scale and lightweight statistics after tokenizer-method calibration.
3. **Adaptive tokenization allocation.** We propose *Adaptive Length Gated Network*, a novel approach which adaptively allocates tokenization capacity via variable residual depth across user samples according to the quantization residual and expression uncertainty, enabling more effective allocation of the discrete representation space and exhibiting advantages as the input data scales up from the *Densing Law* perspective.

2 Related Work

Existing studies related to this work can be organized into three directions: scaling raw behavioral data, constructing tokenized behavioral representations, and adapting token usage across inputs. Table 1 summarizes their coverage of the dimensions considered in this study.

Raw behavioral data scaling. Large scale user modeling studies examine how behavioral data, model capacity, and computation affect downstream utility (Ardalani et al., 2022; Shin et al., 2023; Zhang et al., 2024a; Zivic et al., 2024; Guo et al., 2024; Zhai et al., 2024). Long history methods improve the utilization of extended behavioral sequences through interest extraction, retrieval, and long sequence architectures (Zhou et al., 2019; Li et al., 2019; Cen et al., 2020; Pi et al., 2020; Ren et al., 2025; Cao et al., 2022; Si et al., 2024; Zhou et al., 2024; Chai et al., 2025). However, additional records

Table 1 Comparison of related work across the dimensions considered in this study.

Research direction	Raw Data	Tokenized Data	Capacity Configuration	Capacity Allocation
User data scaling (Ardalani et al., 2022; Shin et al., 2023; Zhang et al., 2024a; Zivic et al., 2024; Guo et al., 2024; Zhai et al., 2024; Shen et al., 2025; Zhang et al., 2026)	✓	×	×	×
Behavioral tokenization (Feng et al., 2026; Liu et al., 2024a; Zhu et al., 2024; Liu et al., 2024b; Hou et al., 2025)	×	✓	×	×
Residual quantization based tokenization (van den Oord et al., 2017; Lee et al., 2022; Rajput et al., 2023; He et al., 2025)	×	✓	×	×
Adaptive quantization (Huijben et al., 2024; Seo and Kang, 2024; Chae et al., 2025; Kusupati et al., 2022)	×	✓	×	✓
Ours	✓	✓	✓	✓

may contain repetitive or weakly informative signals, resulting in diminishing marginal gains (Shen et al., 2025; Zhang et al., 2026). These studies primarily characterize the scaling behavior of raw behavioral data.

Tokenized behavioral data. Behavioral tokenization converts raw user data into compact discrete representations. Existing methods learn reusable behavioral units or vocabularies for user understanding (Feng et al., 2026; Liu et al., 2024a; Zhu et al., 2024; Liu et al., 2024b; Hou et al., 2025; Deng et al., 2025; Wang et al., 2026). VQ-VAE (van den Oord et al., 2017) learns discrete latent codes through vector quantization, while RQ-VAE (Lee et al., 2022) recursively quantizes residual information using multiple codebooks. Based on this formulation, TIGER constructs semantic identifiers for generative retrieval (Rajput et al., 2023), and U^{2}QT introduces multi view quantization for compact user tokens (He et al., 2025). These methods generally use fixed tokenization configurations at selected data scales.

Capacity configuration and adaptive tokenization. Codebook size, residual depth, and token length determine the information capacity of tokenized representations. Existing methods commonly rely on manually selected configurations (Rajput et al., 2023; Liu et al., 2024a; Zhu et al., 2024; He et al., 2025). Adaptive quantization further allows different inputs to activate different numbers of quantization levels (Huijben et al., 2024; Seo and Kang, 2024; Chae et al., 2025). Related mechanisms have also been studied in flexible embeddings and adaptive computation (Kusupati et al., 2022; Graves, 2016; Dehghani et al., 2019). However, these approaches focus on input level allocation and generally assume that the maximum available capacity has already been specified.

3 Preliminary

This section presents the overall research settings used in this paper, including the notation, the representation model pre-training pipeline which serves as the reference system for subsequent analyses, and the downstream evaluation protocol.

3.1 Notation

On a comprehensive internet platform such as Alipay, extensive nonsensitive user information and interactions are accessible. Let $\mathcal{U} = \{u_1, u_2, \ldots, u_N\}$ denotes the behavioral sequence dataset of N users, and each user's behavior u_n is organized according to the chronological order, such as daily, thus $u_n = \{u_n^1, u_n^2, \ldots, u_n^D\}$ across D days. The behavior u_n^d contains multi-source data, for instance, PayBill B, Super Position Model (SPM) records S, and MiniProgram usage descriptions

Table 2 Notations, corresponding descriptions, and key statistics. Pretraining statistics are shown above the dashed line, and downstream evaluation settings are shown below.

Notation	Description	Statistics
N	Number of users	100M–2B
D	Length of the user behavior sequence	30-270 days
P	Model size of the pretrained user embedding model	0.05B–0.4B
u_n^d	Behavior information of user n within the d-th time window, including Paybill B_n^d, SPM S_n^d, and Miniprogram M_n^d	Daily events 0–10 per user; median ~ 6
s	Behavior text tokens used for training	$\sim$ 2200 per user
$\mathcal{D}_{test}$	Test dataset for downstream evaluation	Classification: 50 datasets, 0.5M users per set Text retrieval: 22 datasets, 0.5M users per set U2U retrieval: 22 datasets, 0.5M users per set

M, as $u_n^d = \{B_n^d, S_n^d, M_n^d\}, n = 1 \ldots N, d = 1 \ldots D$. Information of each data source can be expressed in textual form, as $B_n^d \in \mathcal{T}^{L_n^d(B)}$, $S_n^d \in \mathcal{T}^{L_n^d(S)}$ and $M_n^d \in \mathcal{T}^{L_n^d(M)}$ represent the raw text for PayBill, SPM and MiniProgram descriptions, respectively, where $\mathcal{T}$ denotes the predefined text-token dictionary and $L_n^d(\cdot)$ represents the number of behavioral events recorded from each data source for user n within the d-th time window ($L_n^d(\cdot)$ may vary across users and time windows due to heterogeneous activity levels). Table 2 summarizes the notations and corresponding description that used throughout this report.

3.2 Model Pre-training Framework

To establish a representation learning pipeline for user behavioral sequences, we utilize the self-supervised pre-training framework via behavior-text contrastive learning following (Dou et al., 2025), as shown in Figure 2.

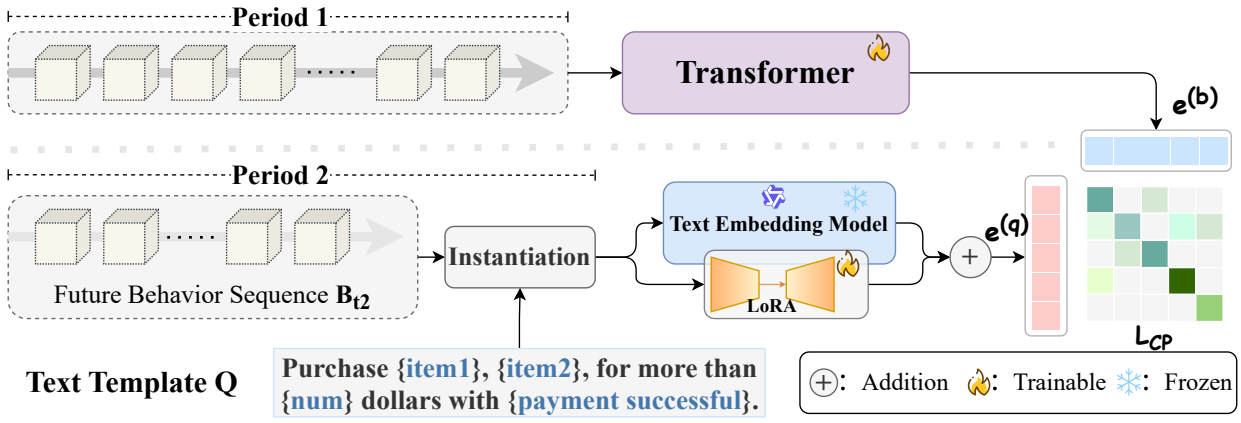


Figure 2 Pretraining framework for general-purpose user representation. Each user's behavioral sequence is split into a past and future segment, as the former part is encoded by a Transformer encoder and the latter is fed to a LoRA-tuned LLM-based model to generated embeddings, which are further aligned via a contrastive loss.

Specifically, for each behavioral information item u_n^d, textual templates $\mathcal{Q}$ are designed to handle them to structured expression. It should be noticed that different templates are employed according to different data sources. For example, the paybill information occurred by a user will be strung together into a complete description purchasing a certain product, belonging to a certain category, transaction amount and payment channel. Templates take the form as:

Template for purchase category: The user purchased {items} amounting more than {num} dollars with {status} payment.

For pre-training, we first collect and curate corpus by splitting each behavioral sequence u_n into two segments: historical part $u_n^{past} = (u_n^1, \ldots, u_n^m)$ and future part $u_n^{future} = (u_n^{m+1}, \ldots, u_n^D)$, where m corresponds to the split time window boundary. Then a Transformer-based user encoder f_θ processes the past segment u_n^{past} to generate the embedding $\mathbf{e}_n^b \in \mathbb{R}^d$. Concurrently a textual description is instantiated from the template $\mathcal{Q}$ which takes the future segment u_n^{future} as input and sampled to generate the supervisory signal. This description is encoded by a LoRA(Hu et al., 2021)-tuned LLM-based embedding model g_ϕ (e.g., Qwen3-Embedding(Zhang et al., 2025)) to produce an embedding $\mathbf{e}_n^q \in \mathbb{R}^d$. The base model is kept frozen and only low rank adapters with rank 16 is applied. As for the pretraining objective, two embeddings are aligned via the contrastive loss Info-NCE(Oord et al., 2018), which can be denoted as:

$$\mathcal{L}_{\text{CP}} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(sim(\mathbf{e}_i^b, \mathbf{e}_i^q)/\tau)}{\sum_{j=1}^{B} \exp(\text{sim}(\mathbf{e}_i^b, \mathbf{e}_j^q)/\tau)} \tag{1}$$

where B means the training batch size, sim represents the cosine similarity function and τ is a learnable temperature parameter. The denominator sums over all B in batch pairs treating the description from the same user as positives and non matching description pairs as negatives.

3.3 Downstream Evaluation Protocol

We evaluate the quality of learned user representations under different scaling settings using a fixed downstream evaluation protocol. The evaluation comprises three common tasks for user representation learning in real-world industrial application: (i) **Classification** (ii) **Text-based Retrieval** (iii) **U2U Retrieval**. It can be noticed that evaluation datasets for each task span diverse scenarios, including user preference, risk control, recommendation, marketing, etc., and the reported values are the metric averages across all scenario-specific datasets to assess the performance and generalizability of the learned representation. Detailed dataset statistics are provided in the Table 2.

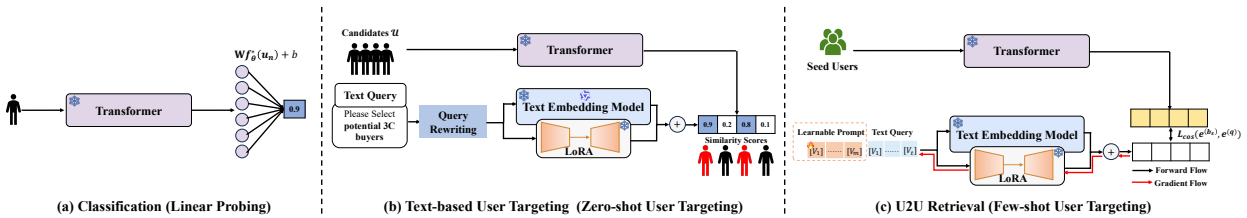


Figure 3 Three downstream tasks for user representation evaluation.

3.3.1 Classification

Representation-based classification task is typically related to industrial scenarios such as advertising, marketing and risk control directly, and linear probing protocol is commonly adopted to evaluate the classification performance(Radford et al., 2021; Cherti et al., 2023). Specifically, we freeze the pretrained user encoder and train only a lightweight linear classifier to fit a given number of labeled samples on top of the extracted embeddings.

As for each pretrained user encoder f_θ^*, we freeze and use it to extract user behavioral embeddings in all downstream datasets, which is further employed as:

$$y_n^{cls} = \mathbb{I}\left[\mathbf{W} f_\theta^*(u_n) + b > \mathcal{T}^{cls}\right] \tag{2}$$

, where $\mathbf{W} \in \mathbb{R}^d, b \in \mathbb{R}$ is the linear network for binary classification which is trained and evaluated on the held-out training and test dataset, $\mathcal{T}^{cls}$ is the classification threshold commonly set at 0.5 .

Metrics. For the linear probing task, three complementary metrics are reported: **AUC**, **KS**, and **Accuracy**. **AUC (Area Under the ROC Curve)** measures the model's overall ability to rank positive instances above negative ones across all classification thresholds. **KS (Kolmogorov–Smirnov)** value measures the maximum separation between the cumulative score distributions of positive and negative instances, reflecting the model's strongest discriminative ability at an optimal threshold. **Accuracy** measures the proportion of correctly classified instances under a specified decision threshold, capturing overall prediction correctness. Together, these metrics evaluate threshold-independent ranking quality, class separability, and threshold-dependent classification performance, providing a comprehensive assessment of downstream classification effectiveness.

3.3.2 Text-based Retrieval

User retrieval based on the text query constitutes a key component of the zero-shot user targeting task, which means judging whether each user belongs to the target audience corresponding to the given natural-language query Q. As shown in Figure , by computing the cosine similarity between the text-query embedding and each candidate user embedding, users whose similarity scores exceed a predefined threshold are selected and targeted.

$$\begin{gathered} y_n^{tbr} = \mathbb{I}\left[sim(f_\theta^*(u_n), \mathcal{E}_\theta^*(\text{Tok}[\mathcal{R}(q)])) > \mathcal{T}^{tbr}\right], \\ \mathcal{U}_q = \left\{ u_n \in \mathcal{U} \;\middle|\; y_n^{tbr} = 1 \right\} \end{gathered} \tag{3}$$

, where sim denotes the cosine similarity function used during pretraining, $\mathcal{T}^{tbr}$ is the retrieval threshold (typically set to 0.5).

Metrics. **AUC**, **Precision** and **Recall** are employed to assess the text-based user retrieval task. **AUC** measures the overall ranking quality between relevant and irrelevant users across all decision thresholds. **Precision** measures the proportion of selected users who are truly relevant to the targeting query, reflecting the reliability of the retrieval. **Recall** measures the proportion of all relevant users that are successfully retrieved, capturing the coverage of the targeting results. Together, these metrics assess ranking quality, targeting accuracy and coverage, providing a comprehensive evaluation of text-based user targeting performance.

3.3.3 U2U Retrieval

User-to-user retrieval based on a small set of seed users is a crucial step in building recommendation systems. Therefore we also design an evaluation approach for the few-shot task as illustrated in Figure 3 (c). The few-shot retrieval task is accomplished via prompt-tuning. Inspired by previous studiesZhou et al. (2022), the seed users $\mathcal{D}_s = \{x_i\}_{i=1}^K$ can be utilized as labels to learn contexts

which improves the descriptive ability of the prompt, by adding learnable tokens $\mathbf{P} = [V_1, \ldots, V_m]$ to the input text.

$$\mathbf{P}^* = \arg\min_{\mathbf{P}} \mathcal{L}_{\cos}(f_\theta^*(u_n), f_\theta^*(\mathbf{P} \oplus \text{Tok}(q)) \tag{4}$$

, where $\mathcal{L}_{\cos}$ represents the cosine similarity loss. Once trained, the tuned prompt can be used as the input to achieve user targeting via pipeline in Figure 3 (b) following Eq 3, as:

$$\begin{aligned} y_n^{u2ur} &= \mathbb{I}\left[sim(f_\theta^*(u_n), \mathcal{E}_\theta^*(\mathbf{P} \oplus \text{Tok}[q])) > \mathcal{T}^{u2ur}\right], \\ \mathcal{U}_q &= \left\{u_n \in \mathcal{U} \;\middle|\; y_n^{u2ur} = 1\right\} \end{aligned} \tag{5}$$

Metrics. The same retrieval metrics **AUC**, **Precision** and **Recall** used in Section 3.3.2 are adopted.

4 Scaling Law Analysis: Raw Data Scaling Wall

Following the protocol established in Section 3.3, we empirically characterize the scaling behavior of user representations along three practical dimensions, namely temporal horizon D, user population N, and model capacity M. For each dimension, we evaluate downstream probing performance using the AUC, KS, and Accuracy metrics defined in Section 3.3. Figure 4 visualizes the raw scaling surface across the (N, D) grid, where a clear saturation pattern emerges. Across all three dimensions, we observe that simply increasing raw behavioral data or model size yields rapidly diminishing downstream returns. This demonstrates a raw behavioral scaling wall in industrial user representation learning, where data volume grows much faster than useful task relevant information.

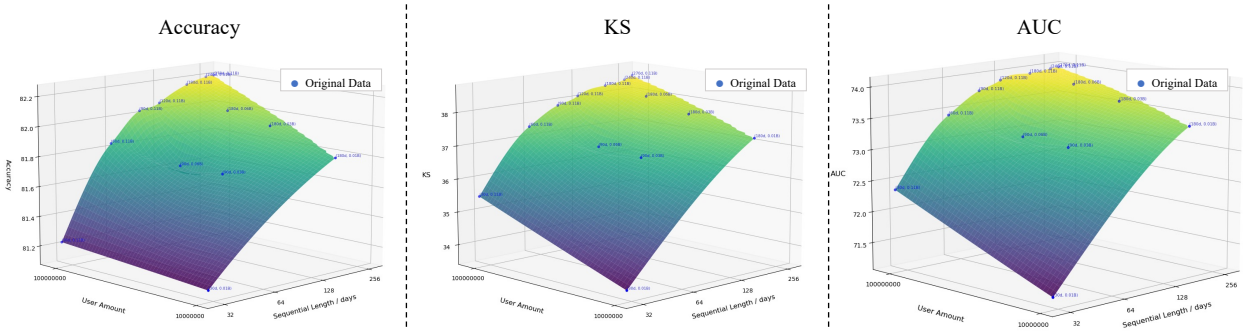


Figure 4 Raw behavioral scaling saturates beyond approximately 0.03B users and 60 days. All three probing metrics exhibit a flattening trend around the saturation thresholds, indicating that additional raw data volume yields diminishing downstream returns.

4.1 Saturation in User Population (N)

> ***Takeaway 1:*** *More users improve representation quality only before behavioral diversity is sufficiently covered. After that point, redundant behavioral patterns dominate the additional data.*

We first examine whether increasing the number of users can continuously improve representation quality. To this end, we vary the user population N from 0.01B to 0.1B while fixing the temporal horizon at $D = 180$ days. The encoder size is fixed at approximately 0.1B parameters.

As shown in Figure 4, increasing the user population initially improves downstream probing performance. When N grows from 0.01B to approximately 0.03B, all three metrics show a clear upward trend. This indicates that adding more users in the low data regime introduces useful behavioral diversity and improves the separability of learned user representations.

However, the improvement quickly diminishes once N exceeds approximately 0.03B. Further increasing the user population to 0.1B leads to only marginal performance gains. This flattening trend suggests that the major behavioral patterns have already been sufficiently covered. Beyond this point, additional users mainly contribute redundant signals rather than new task relevant information.

4.2 Redundancy in Temporal Horizon (D)

> ***Takeaway 2:*** *Extending the behavioral window helps in the early stage, but long raw histories mainly introduce repetitive and low information events.*

We next study whether longer behavioral histories provide more useful information for user representation learning. We vary the historical observation window D from 30 to 120 days while fixing the user population at $N = 0.1$B.

Figure 4 shows that extending the temporal horizon also leads to diminishing returns. Increasing D from 30 to 60 days provides a clear performance gain, suggesting that recent historical behaviors help the encoder capture more complete user preferences and consumption patterns.

Nevertheless, the benefit of longer histories becomes much weaker beyond 60 days. Extending the observation window from 60 to 120 days brings only limited improvement in downstream metrics, despite doubling the amount of historical input. This suggests that long horizon PayBill sequences contain substantial repetition, including recurring payments and habitual consumption behaviors. As a result, additional history increases sequence length much faster than it increases downstream relevant information. This saturation pattern is also reflected in the pretraining dynamics shown in Appendix Figure 12. As the temporal horizon becomes longer, the contrastive alignment loss converges to increasingly similar final values, indicating that additional historical context does not provide proportionally more useful supervision.

4.3 Quality Dissociation in Model Capacity (P)

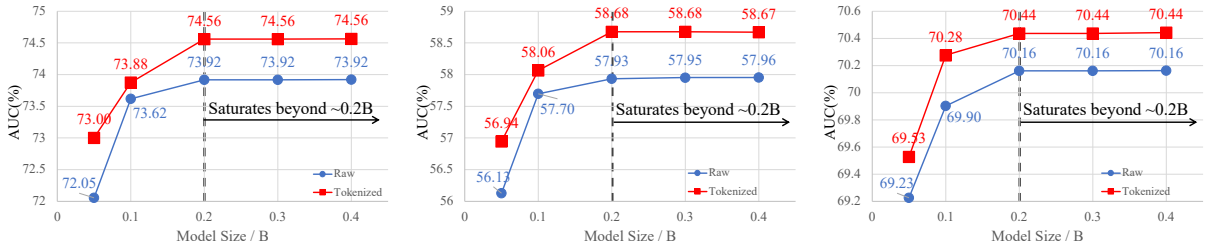
> ***Takeaway 3:*** *Larger encoders achieve lower pretraining loss, but downstream accuracy saturates. This shows that information density, rather than model capacity alone, is the key bottleneck.*

We further examine whether increasing model capacity can overcome the saturation caused by redundant behavioral data. We fix both the user population and the temporal horizon at $N = 0.1$B and $D = 180$ days, which already lies in the saturated data regime identified above. We then scale the Transformer-based user encoder from 0.05B to 0.4B parameters while keeping all other training configurations unchanged.

Figure 5 shows a clear mismatch between optimization and downstream representation quality. Larger models consistently achieve lower training loss, confirming that increasing model capacity improves the ability to fit the pretraining objective. However, downstream evaluation shows a different trend. Increasing the model size from 0.05B to 0.2B improves test accuracy, but further

scaling to 0.4B produces almost no additional gain. Despite the larger parameter budget and higher computational cost, the learned user representations do not become more discriminative.

This result exposes a loss and quality dissociation. A lower pretraining loss does not necessarily translate into better downstream representation quality. In the saturated data regime, extra parameters mainly fit redundant behavioral details rather than learning more generalizable user features. Therefore, the limiting factor is not simply model expressivity, but the amount of downstream relevant signal contained in the raw behavioral input.



(a) Classification Task Evaluation (b) Text-Retrieval Task Evaluation (c) U2U-Retrieval Task Evaluation

Figure 5 Scaling model parameters under fixed behavioral data exhibits strong diminishing returns. Larger models continue to achieve lower training loss, but downstream evaluation accuracy rapidly saturates beyond moderate model scales. This indicates that representation quality is constrained by information density rather than model capacity alone.

4.4 Implications and Motivation for Behavioral Densing

> ***Overall Takeaway:*** *The key to further scaling is no longer processing more raw behavioral events, but increasing the downstream relevant information carried by each effective input unit.*

Taken together, the steepest performance improvements are confined to the region bounded by $N \leq 0.03$B, $D \leq 60$ days, and $M \leq 0.2$B parameters. Beyond these thresholds, scaling user population, temporal horizon, or model capacity introduces substantial additional cost while yielding negligible downstream gains. These empirical results show that raw data volume and model size improve user representations only before redundancy becomes dominant.

These observations suggest that brute force scaling is *not* a sustainable path for industrial user representation learning. Once raw behavioral data enters the saturation regime, further increasing data volume, compute, or model parameters leads to rapidly diminishing returns. The bottleneck is therefore not merely insufficient model capacity, but the low information density of raw behavioral sequences, where many events are repetitive, predictable, or only weakly relevant to downstream tasks.

This motivates a shift from volume oriented scaling to density oriented scaling. Rather than relying on more raw behavioral events, the goal is to make each effective input unit carry more downstream relevant signal under a fixed data and compute budget. The next section instantiates this idea through residual quantization. Instead of feeding long and repetitive raw sequences directly into the user encoder, we first transform them into compact discrete behavioral tokens, providing a practical mechanism for increasing behavioral information density and improving encoder utilization.

5 Densing Gains from Tokenization

We identify a raw behavioral scaling wall. Once redundancy dominates user histories increasing the user population extending temporal windows or enlarging model capacity yields diminishing downstream returns. This suggests that further improvement should come from increasing the information density of behavioral inputs rather than simply processing more raw events.

To operationalize this density oriented approach we employ residual quantization as a practical behavioral tokenizer. We adopt existing tokenization methods, e.g., RQ-VAE (Lee et al., 2022; He et al., 2025), as the quantization backbone for industrial user behavior modeling. Specifically we compress long multi source PayBill histories into compact discrete behavioral tokens and then train the same Transformer based user encoder on these tokenized sequences. This design allows us to test whether the gain comes from a denser input representation rather than from changes in the downstream encoder architecture.

Figure 6 illustrates the overall pipeline. Raw multi source PayBill behaviors are first mapped into continuous behavioral embeddings. The RQ-VAE tokenizer then compresses these embeddings into multi level discrete tokens through residual quantization. The resulting token sequence is fed into the same user encoder and optimized with the same contrastive user text alignment objective as in the raw sequence setting.

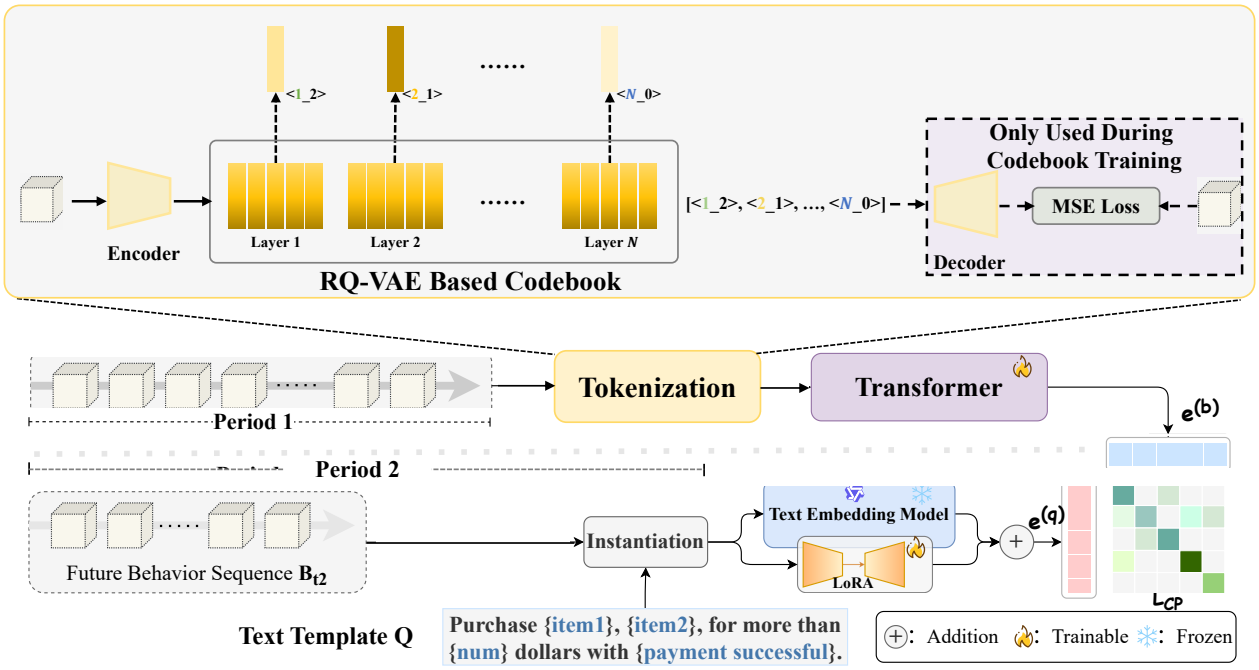


Figure 6 Overview of the RQ-VAE based tokenization method used for behavioral densing.

5.1 Behavioral Tokenization with RQ-VAE

We now describe how residual quantization converts long and redundant behavioral histories into compact discrete tokens. To adapt RQ-VAE from image processing to industrial behavioral modeling we make three task specific modifications upfront. First instead of quantizing image latents the tokenizer operates on multi source PayBill behavior embeddings. Second we decouple the output token length H from the raw temporal horizon D so long histories can be represented by

a fixed number of behavioral tokens. Third we use the tokenizer as a behavioral densing module for downstream user representation learning rather than only optimizing it for reconstruction quality.

Addressing the first modification given a user u_n let $\mathbf{X}_n = (\mathbf{x}_{n,1}, \mathbf{x}_{n,2}, \ldots, \mathbf{x}_{n,L_n})$ denote the continuous embedding sequence derived from the raw behavioral history B_n where each $\mathbf{x}_{n,\ell} \in \mathbb{R}^d$. Directly modeling $\mathbf{X}_n$ preserves all raw events but also retains substantial redundancy from repeated routines and weakly informative interactions.

To implement the second modification and increase representation density we apply a local aggregation step that maps the raw sequence into a compact latent representation $\mathbf{Z}_n = (\mathbf{z}_{n,1}, \mathbf{z}_{n,2}, \ldots, \mathbf{z}_{n,H})$ where $H \ll L_n$ is the fixed token budget of the behavioral tokenizer. This step decouples the effective input length from the raw temporal horizon and encourages the tokenizer to retain the most informative behavioral semantics under a compact budget.

For each latent vector $\mathbf{z}_{n,h}$ RQ-VAE applies multi stage residual quantization. At the m-th stage a learned codebook $\mathcal{C}^{(m)} = \{\mathbf{c}_1^{(m)}, \ldots, \mathbf{c}_{K_m}^{(m)}\}$ quantizes the current residual by selecting its nearest codeword. The process can be written compactly as

$$\begin{aligned} \mathbf{r}_{n,h}^{(1)} &= \mathbf{z}_{n,h}, \\ k_{n,h}^{(m)} &= \arg\min_{k \in \{1,\ldots,K_m\}} \left\| \mathbf{r}_{n,h}^{(m)} - \mathbf{c}_k^{(m)} \right\|_2^2, \\ \mathbf{r}_{n,h}^{(m+1)} &= \mathbf{r}_{n,h}^{(m)} - \mathbf{c}_{k_{n,h}^{(m)}}^{(m)}, \\ \hat{\mathbf{z}}_{n,h} &= \sum_{m=1}^{M} \mathbf{c}_{k_{n,h}^{(m)}}^{(m)}, \\ \mathbf{t}_{n,h} &= \left(k_{n,h}^{(1)}, k_{n,h}^{(2)}, \ldots, k_{n,h}^{(M)}\right). \end{aligned} \tag{6}$$

The full user history is therefore represented as a compact token sequence $\mathbf{T}_n = (\mathbf{t}_{n,1}, \mathbf{t}_{n,2}, \ldots, \mathbf{t}_{n,H})$.

This residual formulation provides a natural coarse to fine structure for behavioral modeling. Early codebooks capture dominant macro level semantics such as stable consumption categories routine transportation or frequent payment scenarios. Later codebooks refine residual variations such as merchant level preferences or subtle changes in spending behavior which may still be predictive for downstream targeting. Compared with a single codebook residual quantization can preserve finer behavioral distinctions under the same compact token budget.

The tokenizer is trained with a reconstruction objective and a commitment loss

$$\mathcal{L}_{\mathrm{RQ}} = \mathcal{L}_{\mathrm{rec}}(\mathbf{Z}_n, \hat{\mathbf{Z}}_n) + \sum_{m=1}^{M} \left\| \mathrm{sg}[\mathbf{r}^{(m)}] - \hat{\mathbf{r}}^{(m)} \right\|_2^2 + \beta \sum_{m=1}^{M} \left\| \mathbf{r}^{(m)} - \mathrm{sg}[\hat{\mathbf{r}}^{(m)}] \right\|_2^2 \tag{7}$$

where $\mathcal{L}_{\mathrm{rec}}$ denotes the reconstruction loss $\mathrm{sg}[\cdot]$ is the stop gradient operator and β controls the commitment penalty.

Reflecting the third modification after tokenizer pretraining we freeze the RQ-VAE and train the user encoder on $\mathbf{T}_n$ using the same alignment objective as the raw sequence baseline. This separation allows us to attribute downstream gains strictly to behavioral densing rather than additional encoder capacity.

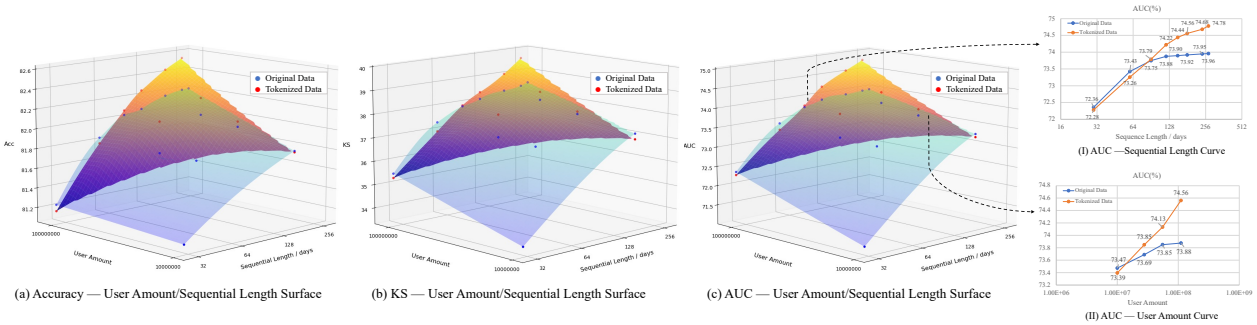


Figure 7 Tokenized representations outperform raw sequences at matched data scales with the gap widening where raw modeling saturates. Subplot (a) shows the full two dimensional scaling surface. Subplots (b) and (c) show cross sections at fixed user population and fixed temporal horizon.

5.2 Experimental Evaluation of Discrete Tokenization

To evaluate whether residual quantization alleviates the raw behavioral scaling wall we repeat the scaling protocol from Section 4. We compare raw sequence modeling with tokenized sequence modeling under matched data and compute budgets. In the raw setting the user encoder is trained directly on PayBill sequences. In the tokenized setting raw histories are first converted into RQ-VAE tokens and the same user encoder is trained on the resulting discrete token sequences. All models use the same contrastive user text alignment objective and are evaluated with the same downstream probing tasks and metrics.

Figure 7 compares the two settings across different scales. Along the temporal dimension tokenized representations become increasingly advantageous as the observation window grows longer. At shorter horizons raw sequences remain competitive because behavioral redundancy is still limited. Around $D \approx 64$ days the tokenized curve begins to outpace the raw baseline and the advantage becomes clearer as the horizon further increases. This indicates that residual quantization better distills useful behavioral signal from long and repetitive histories.

A similar pattern appears along the user population dimension. At smaller user scales raw and tokenized models perform similarly since adding users still introduces useful behavioral diversity. As the user population grows and raw scaling approaches saturation the tokenized representation shows a clearer advantage. In our experiments this crossover appears around $N \approx 1.2 \times 10^7$ after which the tokenized representation consistently outperforms the raw baseline. This suggests that behavioral densing is most helpful when the marginal utility of additional raw data becomes low.

To quantify this effect let $\mathcal{P}_{\text{raw}}(s)$ and $\mathcal{P}_{\text{tok}}(s)$ denote downstream probing performance under raw and tokenized modeling where s can represent temporal horizon D or user population N. We define the tokenization gain as

$$\Delta(s) = \mathcal{P}_{\text{tok}}(s) - \mathcal{P}_{\text{raw}}(s). \tag{8}$$

Empirically $\Delta(s)$ is not uniformly positive at all scales. Instead it becomes positive and grows larger when raw modeling enters the diminishing return regime. This pattern confirms that RQ-VAE tokenization does not merely shorten the input. It increases the effective density of behavioral information by suppressing repetitive details and preserving task relevant distinctions.

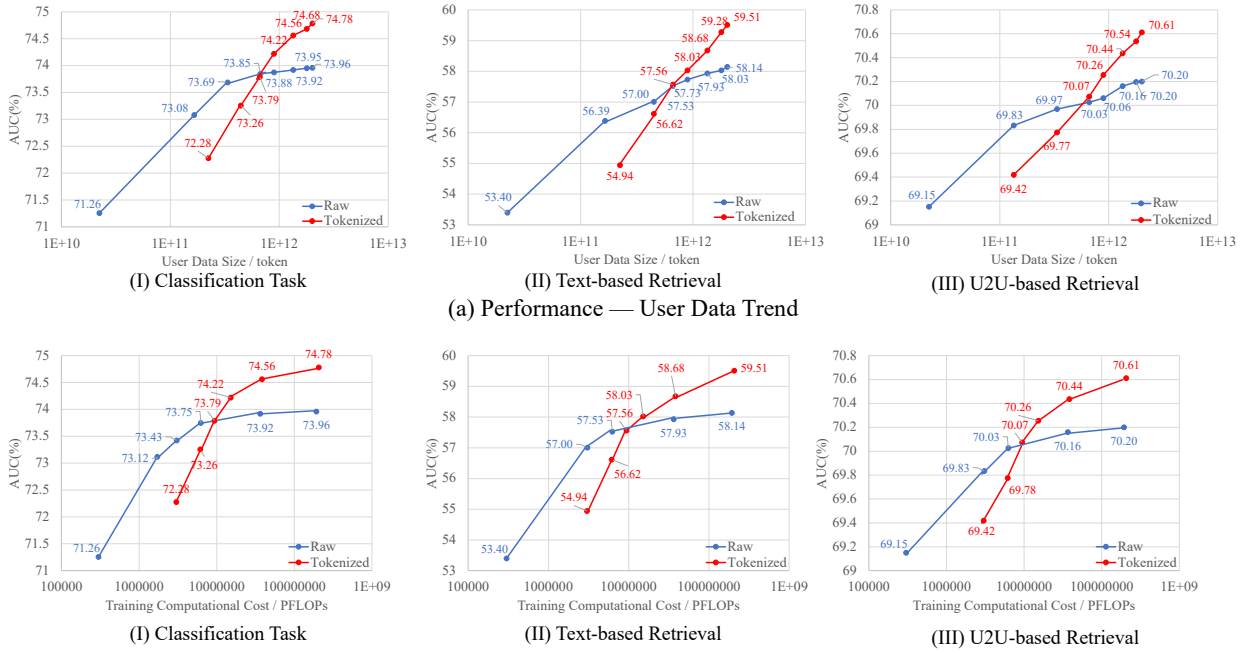


Figure 8 Downstream task performance with data size & training computes scaling.

5.3 From Raw Scaling to Density Scaling

The empirical comparison supports the central motivation of behavioral densing. Once raw behavioral data reaches saturation representation quality is no longer primarily limited by the volume of users the length of the historical window or the size of the encoder. Instead the fundamental limiting factor is the amount of downstream relevant signal contained in each effective input unit. Here RQ-VAE tokenization acts as a practical density operator. It shifts the learning problem from modeling long redundant raw sequences to modeling compact and semantically concentrated tokens. By suppressing repetitive routines and preserving task relevant distinctions the tokenizer enables moderate capacity encoders to better utilize long behavioral histories without a proportional increase in computational burden.

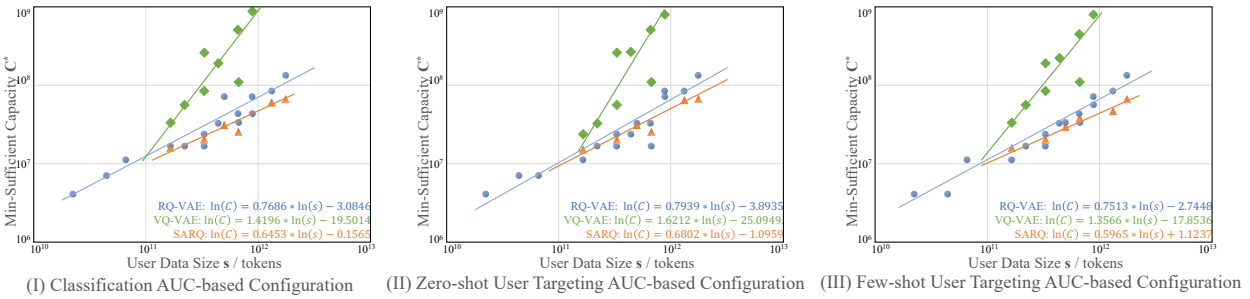
However these results also reveal a limitation of fixed length tokenization. Behavioral information is naturally heterogeneous. Some history segments contain rich and diverse signals while others consist mostly of predictable routines. A uniform token budget across all segments is therefore suboptimal. This observation directly motivates the next section where we allocate token capacity adaptively according to behavioral complexity rather than applying a fixed sequence length.

6 The Behavioral Densing Law

We reveal that raw behavioral scaling eventually saturates and that tokenization improves performance precisely in this saturated regime. This benefit reflects a fundamental density principle where an effective representation must maximize downstream relevant information while minimizing the cost of modeling redundant histories.

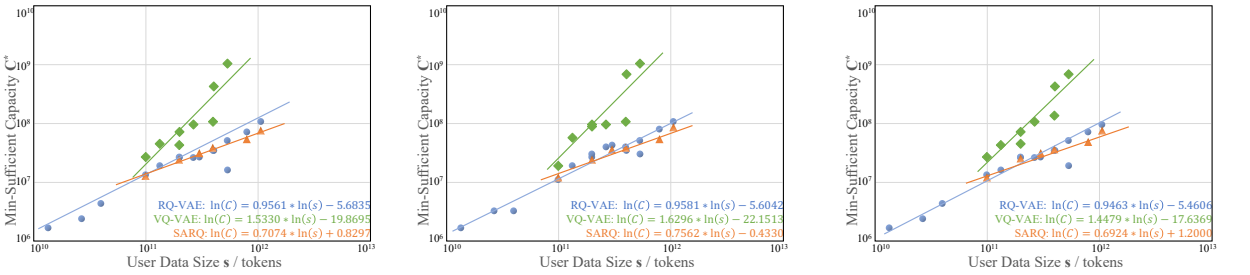
We formalize this principle as the Behavioral Densing Law. Rather than treating densing as a heuristic configuration table we formulate it as a performance and cost Pareto optimization

problem. The law characterizes how the Pareto optimal representation capacity evolves with behavioral scale. RQ-VAE tokenization serves as one concrete solver for estimating this trajectory under large scale behavior data.



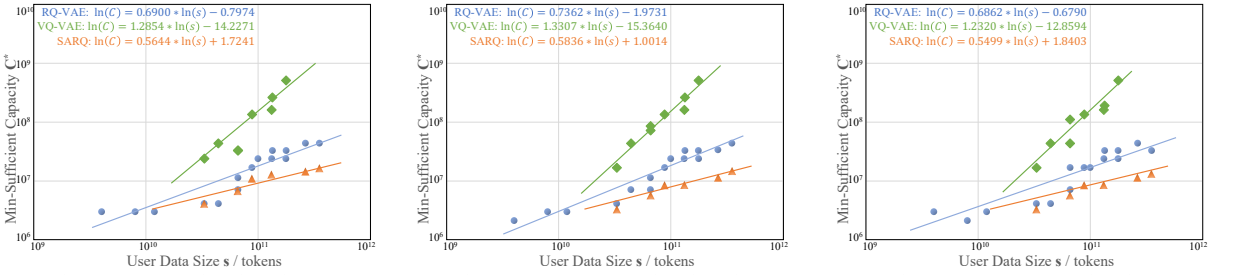
(I) Classification AUC-based Configuration (II) Zero-shot User Targeting AUC-based Configuration (III) Few-shot User Targeting AUC-based Configuration

(a) Bill Data Densing Law, $\mathcal{U}_d = 0.5513$



(I) Classification AUC-based Configuration (II) Zero-shot User Targeting AUC-based Configuration (III) Few-shot User Targeting AUC-based Configuration

(b) SPM Data Densing Law, $\mathcal{U}_d = 0.5891$



(I) Classification AUC-based Configuration (II) Zero-shot User Targeting AUC-based Configuration (III) Few-shot User Targeting AUC-based Configuration

(c) Miniprogram Data Densing Law, $\mathcal{U}_d = 0.5292$

Figure 9 Behavioral Densing Law across different data sources and tokenization methods, derived from three tasks.

6.1 Behavioral Density and Pareto Optimal Densing

To explicitly quantify this principle we define behavioral density. Let $\mathbf{s} = (s_1, s_2, \ldots, s_r)$ denote a generalized behavioral scale state where each component captures one source of behavioral complexity such as user population temporal horizon modality count or scenario diversity.

Let $C(\phi)$ be the actual cost of a densified representation ϕ and let $\widehat{C}_{\text{raw}}(\mathbf{s})$ denote the effective raw budget which is the cost required for raw sequence modeling to achieve the same downstream utility. The optimal behavioral density at scale state $\mathbf{s}$ is defined as

$$\rho^*(\mathbf{s}) = \frac{\widehat{C}_{\text{raw}}(\mathbf{s})}{C^*(\mathbf{s})}. \tag{9}$$

The goal of behavioral densing is to maximize this density ratio navigating the representation toward a better performance and cost frontier.

At a given scale state $\mathbf{s}$ each representation transformation ϕ yields a downstream utility $U(\phi, \mathbf{s})$ and incurs a representation cost $C(\phi)$. Introducing a deployment tradeoff parameter λ the Pareto optimal representation is formulated as

$$\phi^*(\lambda, \mathbf{s}) = \arg\max_{\phi \in \Phi} \left[U(\phi, \mathbf{s}) - \lambda C(\phi) \right]. \tag{10}$$

The parameter λ controls the cost sensitivity of the system. A larger λ favors cheaper representations while a smaller λ allows higher capacity when it brings sufficient downstream gain. Consequently densing does not aim to maximize the compression ratio or the tokenizer size. It specifically targets the representation with the best utility and cost tradeoff at a given scale. We define the Behavioral Densing Law as the solution map induced by this Pareto objective.

6.2 Minimal Sufficient Tokenization Capacity

We instantiate this theoretical solution using different tokenization methods. Let q_κ denote an RQ-VAE tokenizer configured by $\kappa = (K, M, d, H)$ where K is the codebook size M is the number of residual quantization stages d is the code embedding dimension and H is the output token length.

For RQ-VAE tokenization we instantiate the general representation cost $C(\phi)$ with the discrete tokenizer capacity $C_{\text{tok}}(\kappa)$. We quantify this capacity as

$$C_{\text{tok}}(\kappa) = HM \log K + \eta HMd, \tag{11}$$

where the first term measures discrete code capacity and the second term accounts for continuous code embedding cost.

For a given scale (N, D) the theoretical Pareto objective is to find the optimal configuration κ^*. However downstream performance is measured with stochastic training noise making exact continuous optimization over λ unstable. We therefore approximate the Lagrangian Pareto objective using a robust constrained form. We first compute the near optimal utility

$$U^*(N, D) = \max_{\kappa \in \mathcal{K}} U(q_\kappa, N, D), \tag{12}$$

and then rigorously select the smallest configuration that reaches this utility bound

$$\kappa^*(N, D) = \arg\min_{\kappa \in \mathcal{K}} C_{\text{tok}}(\kappa) \quad \text{subject to} \quad U(q_\kappa, N, D) \geq U^*(N, D) - \epsilon. \tag{13}$$

This constrained form selects a Pareto efficient point that is robust to stochastic training noise and discrete tokenizer configurations. It safely identifies the minimal sufficient capacity avoiding larger configurations whose additional capacity provides negligible downstream gain.

6.3 The Induced Scale Law

To obtain a compact empirical form of this solution map we study the trajectory of Pareto optimal solutions across varying scale states. In the redundancy dominated regime the utility frontier of tokenized representations exhibits diminishing returns with respect to representation capacity C. A local approximation of this frontier is

$$U(C, \mathbf{s}) = U_\infty(\mathbf{s}) - a(\mathbf{s}) C^{-b}, \tag{14}$$

where $U_\infty(\mathbf{s})$ is the maximum attainable utility at scale $\mathbf{s}$ while $a(\mathbf{s})$ measures the remaining utility gap at limited capacity and $b > 0$ controls the curvature of the frontier.

Substituting Equation 14 into the continuous Pareto objective yields

$$C^*(\lambda, \mathbf{s}) = \arg\max_C \left[U(C, \mathbf{s}) - \lambda C \right]. \tag{15}$$

The first order condition provides

$$a(\mathbf{s}) b \left(C^*(\lambda, \mathbf{s}) \right)^{-b-1} = \lambda, \tag{16}$$

which solves to

$$C^*(\lambda, \mathbf{s}) = \left(\frac{a(\mathbf{s}) b}{\lambda} \right)^{\frac{1}{b+1}}. \tag{17}$$

Assuming the scale dependent gap term grows according to a general power law $a(\mathbf{s}) = a_0 \prod_{i=1}^{r} (s_i / s_{i,0})^{\gamma_i}$ we can rewrite Equation 17 as

$$\ln C^*(\lambda, \mathbf{s}) = \beta + \sum_{i=1}^{r} \alpha_i \ln \frac{s_i}{s_{i,0}} - \alpha_\lambda \ln \lambda, \tag{18}$$

where $\alpha_i = \gamma_i / (b+1)$. In practical deployments the tradeoff λ typically remains fixed. Equation 18 thus reduces to the general operational form of the Behavioral Densing Law

$$\boxed{\ln C^*(\mathbf{s}) = \beta + \sum_{i=1}^{r} \alpha_i \ln \frac{s_i}{s_{i,0}}} \tag{19}$$

This derivation provides the scale dependent form of Pareto optimal capacity under a diminishing return utility frontier. It dictates that the required capacity of the optimal densified representation grows as a power law function of behavioral complexity.

In our scaling experiments the behavioral scale state is specifically instantiated as $\mathbf{s} = (N, D)$ where N represents the user population and D represents the temporal horizon. Equation 21 therefore reduces to the task specific empirical form

$$\ln C^*(N, D) = \beta + \alpha_N \ln \frac{N}{N_0} + \alpha_D \ln \frac{D}{D_0} \tag{20}$$

This confirms that N and D are not strict prerequisites for the law but rather two controllable dimensions used in this work to parameterize behavioral scale.

We refer to Equation 21 as the *Behavioral Densing Law*. In the scalar setting, it becomes

$$\boxed{\ln C^*(s) = \beta + \alpha \ln(s / s_0)} \tag{21}$$

.

For further analysis, the coefficients α_i should summarize the capacity trajectory under a particular behavioral data distribution and code-space expression form, which can be represented as

$$\alpha_i = f_i \left(\mathcal{U}_d, \mathcal{E}_\phi \right), \tag{22}$$

where $\mathcal{U}_d$ captures data properties such as behavioral diversity and redundancy, while $\mathcal{E}_\phi$ captures how tokenization method ϕ expresses the available code space, including codebook organization, residual depth, and capacity allocation. It's analyzed that the intra-datasource diversity should

be positively correlated with the tokenization capacity, as richer information should be mapped to a larger representation space to avoid conflicts. In this study, for the quantitative description of the property, we utilize the *Mean k-NN Cosine Distance* of LLM-based text embedding (e.g., Qwen3-Embedding) as the measurement, which can be denoted as:

$$\mathcal{U}_d = \frac{1}{Mk} \sum_{i=1}^{M} \sum_{j \in \text{kNN}(i)} d(i,j) \tag{23}$$

6.4 Empirical Validation of the Solution Trajectory

Figure 9 reports the empirical minimal sufficient tokenization configurations under varying measured data scales. The measured configurations display a clear structural pattern. As either the user population or the temporal horizon increases the minimal sufficient tokenization capacity also increases monotonically.

By fitting Equation 20 over these empirical configurations we estimate the scaling coefficients in Figure 9 for different data sources. This quantitative result confirms our theoretical formulation. In the redundancy dominated regime the Pareto optimal capacity of behavioral tokenization grows strictly as a power law function of user population and temporal horizon.

Crucially this does not imply that larger tokenizers are unconditionally better. Instead it dictates that larger behavioral scales require a larger minimal sufficient capacity. Exceeding this optimal capacity yields negligible performance benefits while severely degrading cost efficiency. The measured configurations and the resulting fitted coefficients thus represent the empirical solution trajectory of the Behavioral Densing Law estimated under large scale behavioral data.

We also estimate coefficients from empirical minimal-capacity trajectories, and obtain the approximate relation between α_i and $\mathcal{U}_d$ as $\alpha_i \propto \mathcal{U}_d^2$, as we conducted a series of experiments to figure the minimal sufficient configurations tailored to each input user data size and utilize the Pareto-Optimal method to fit the law (similar to (Cherti et al., 2023)).

To ensure generalizability and reliability, the validation experiments are performed on three tokenization methods: RQ-VAE, VQ-VAE and SARQ, and three different user data sources: Paybill, SPM and Miniprogram. We also evaluated all the three tasks in Sec 3.3. For each tokenization method's capacity configuration, size of the representation space for all available SID is calculated, as for RQ-VAE it's obtained by $C_{RQ-VAE} = K^M$, C_{VQ-VAE} is determined by the searching space specified during training for VQ-VAE, and C_{SARQ} is the activated SID capacity space for SARQ. For the optimal configuration C^*, searching approach in Section 6.1 is utilized.

The experimental validation results are presented in Figure 9. According to the results we summarize the following patterns:

Approximately Linear Relationship on the Logarithmic Input Scale. Across all input data sources and evaluation tasks, the optimal capacity and input user data size exhibit an approximately linear relationship on a logarithmic scale ($ln(C^*)$ and $ln(s)$), as indicated by the distribution of the data points.

Identical Trend for Different Downstream Tasks. It's observed that the trends of the minimal sufficient configuration and input data size share the similar pattern under all tasks. At the given tokenization method and data source (the same color within Figure 9 (a), (b) or (c)), the slopes

between $ln(C^*)$ and $ln(s)$ obtained from the Pareto frontier in (I), (II) and (III) can be considered identical within the numerical estimation error.

Slope Affected by the Tokenization Method. For different tokenization approaches, the fitted distribution varies, as illustrated in different color lines in each subfigure. It's also observed that VQ-VAE retains higher slope value compared to RQ-VAE, and SARQ reaches the lowest value. Our analysis suggests that this is related to each method's representation space redundancy $\mathcal{E}_\phi$. VQ-VAE suffers from redundant storage, as similar representations require multiple complete SID and wastes shared structures. In contrast, RQ-VAE enables compositional reuse of SID thereby reducing $\mathcal{E}_\phi$ in the codebook. However, among the theoretically possible K^M combinations in RQ-VAE, many are invalid or have an extremely low probability of being selected, which can be alleviated to some extent in SARQ.

Slope Proportional to Intra-Source Uniqueness. By comparing the slopes of the same colored line from Figure 9's column subfigure (e.g.,(I) in (a), (b) and (c)), we also observe that within the same tokenization method, the slopes probably retain the approximately proportional relation corresponding to the squared value of intra-datasource uniqueness $\mathcal{U}_d$, as the rate between three data sources is around $1 : 1.15 : 0.92$ while the $\mathcal{U}_d$ rate is around $1 : 1.07 : 0.96$ (within 10^{-2} numeric error). This discovery provides guidance for tokenization-based user representation learning, which means when using the common tokenization method (with fixed $\mathcal{E}_\phi$), the optimal codebook capacity configuration can be obtained using Eq 21 and Eq 22 with the calculation of $\mathcal{U}_d$ according to Eq 23.

7 New Method: Adaptive-Length-Gating-Based RQ-VAE Quantization

Following the Behavioral Densing Law in Section 6, we further move from scale-level capacity selection to instance-level capacity allocation. The law suggests that the optimal densified representation should use the minimal sufficient capacity required by the input token cost. However, fixed-length tokenization methods, including VQ-VAE and RQ-VAE based tokenizers, assign the same number of code levels to all behavioral periods. This uniform allocation ignores the heterogeneity of user behaviors. Some periods contain diverse and ambiguous behavioral signals and require deeper residual codes, while others mostly consist of routine or repetitive behaviors and can be represented with shorter codes.

This motivates a variable-length tokenization strategy. The key principle is simple: a new residual code should be used only when its expected marginal utility is larger than its marginal representation cost. Therefore, instead of choosing a single global code length for all behavioral periods, we adaptively decide whether each behavioral representation should continue to the next quantization level. This provides an instance-level realization of the Behavioral Densing Law.

7.1 Method

To implement this instance-level capacity allocation principle, as shown in Figure 10, we propose an *Adaptive Length Gated Network* (ALGN) to generate variable-length semantic IDs for user behavioral data. ALGN is built on top of the residual quantization architecture. At each quantization level, it decides whether the current behavioral representation still needs an additional residual code.

The design of ALGN follows a marginal utility view. For a behavioral representation at level l, continuing to the next residual code is beneficial only if the remaining information is still large

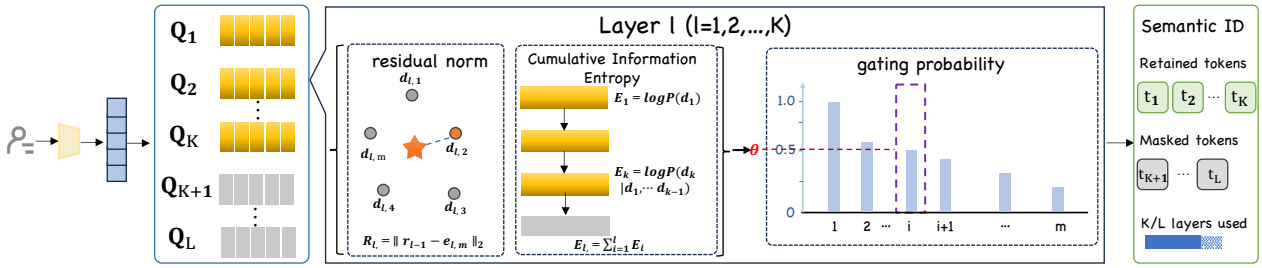


Figure 10 Adaptive variable-length tokenization method allocates more codes to high-information behavioral periods and fewer codes to routine or redundant periods, achieving comparable or better accuracy with fewer tokens on average than fixed-length baselines.

enough to justify the extra code cost. Since the true downstream marginal utility is not directly observable during tokenization, we approximate it using two signals.

First, the residual norm measures the amount of information that remains unexplained by the current code path. Let r_{l-1} denote the residual before level l, and let $e_{l,m}$ denote the selected code embedding at this level. The residual norm is defined as

$$R_l = \left\| r_{l-1} - e_{l,m} \right\|_2 . \tag{24}$$

A larger R_l indicates that the current codes are still insufficient to reconstruct the behavioral representation, so an additional residual code may bring meaningful utility. A smaller R_l suggests that the remaining information has already been mostly explained, and continuing the quantization process may lead to diminishing returns.

Second, the code uncertainty measures how ambiguous the current semantic assignment is. Let $p(m_l \mid m_{<l})$ denote the conditional probability of selecting code m_l given previous code selections. We define the cumulative uncertainty as

$$E_l = -\sum_{k=1}^{l} \log p(m_k \mid m_{<k}). \tag{25}$$

A larger E_l means that the code path is more uncertain, which usually corresponds to more complex, ambiguous, or information-rich behavioral patterns. A smaller E_l indicates that the tokenizer is confident about the current semantic path, so additional code levels are less necessary.

Based on these two proxies, ALGN estimates the continuation probability at level l as

$$g_l = \sigma\left(\mathrm{sp}(w_R) R_l + \mathrm{sp}(w_E) E_l - \mathrm{sp}(w_C) c_l + b\right), \tag{26}$$

where w_R, w_E, w_C, and b are learnable parameters, $\sigma(\cdot)$ is the sigmoid function, and $\mathrm{sp}(\cdot)$ is the Softplus function used to keep the weights positive. Here c_l denotes the marginal cost of activating level l, such as the cumulative code length or level index. This form encodes the intended monotonic behavior: the continuation probability increases when the residual information or uncertainty is high, and decreases when the additional code cost becomes large.

The activated semantic ID length is then determined by the first level where the continuation probability falls below a threshold:

$$L_{\text{act}} = \min\left\{ l : g_l \le \theta \right\}, \tag{27}$$

where θ is a stopping threshold. If no level satisfies the stopping condition, the tokenizer uses the maximum allowed length $L_{\max}$.

We train the variable-length tokenizer with a reconstruction objective and a length distribution regularizer:

$$\mathcal{L} = \mathcal{L}_{\text{rec}}^{\text{act}} + \lambda \text{KL}\left(P_{\text{len}} \,\|\, Q\right), \tag{28}$$

where $\mathcal{L}_{\text{rec}}^{\text{act}}$ is the reconstruction loss computed using only the activated residual levels, and P_{len} is the empirical distribution of activated code lengths induced by ALGN. The prior Q controls the expected length distribution. In practice, we use a decaying prior such as the geometric distribution

$$Q(l) = (1-\gamma)^{l-1}\gamma. \tag{29}$$

This regularizer prevents the tokenizer from trivially activating all residual levels and encourages a compact length distribution. Compared with directly penalizing the length of each sample, a distributional prior is more flexible because it can incorporate prior knowledge about the long-tail nature of user behaviors.

Overall, ALGN implements a local marginal utility rule for behavioral tokenization. Fixed-length RQ-VAE assigns the same capacity to every behavioral period, while ALGN dynamically allocates capacity according to the remaining residual information, quantization uncertainty, and marginal code cost. Therefore, high-information periods receive deeper semantic IDs, while routine or redundant periods stop early. This converts the Behavioral Densing Law from a scale-level capacity principle into an instance-level adaptive tokenization mechanism.

7.2 Experimental Results

We first compare downstream representation quality and the utilized SID capacity of each method. As studied in Section 5, all classification and retrieval tasks exhibits the similar performance trend thus only classification metrics on paybill data are reported. The performance comparison is conducted using dataset size at 180 days sequence length and 0.1B user count, with the input setting held constant across all methods. We set the length prior Q in Eq. 28 as a geometric distribution with $\gamma = 0.3$ and compare ALGN with four baselines: fixed-length RQ-VAE, a prior-based heuristic method, a residual-norm-only variant, and an information-entropy-only variant. This comparison evaluates whether joint gating with residual information and semantic uncertainty provides better capacity allocation than fixed, prior-only, or single-signal allocation.

7.2.1 Comparison

Table 3 Comparison of different tokenization methods. SID capacity usage is used to measure efficiency, with classification metrics reported for performance evaluation.

Method	Capacity (%)↓	AUC (%)↑	KS (%)↑	Acc (%)↑
RQ-VAE(Lee et al., 2022)	100.00	74.56	39.02	82.44
VQ-VAE(van den Oord et al., 2017)	316.23	73.45	37.95	82.47
RQ-Kmeans(Deng et al., 2025)	100.00	73.77	38.34	82.85
SARQ(Wang et al., 2026)	76.24	75.36	41.28	83.16
ALGN	**63.47**	**76.43**	**43.31**	**83.52**

Results in Table 3 shows that ALGN surpasses the best baseline SARQ by increasing 1.07%/2.03%/0.36% AUC/KS/Acc and saving 13.23% capacity.

The comparison among variable-length variants further supports the design of ALGN. The heuristic method slightly reduces SID usage but brings only marginal performance gains, suggesting that prior length statistics alone are insufficient for instance-level capacity allocation. Residual norm and information entropy each provide stronger improvements, indicating that both remaining reconstruction information and semantic uncertainty are useful signals for estimating the marginal utility of additional code levels. ALGN achieves the best performance and the lowest SID usage by jointly modeling residual information, code uncertainty, and marginal code cost.

These results provide an instance-level validation of the Behavioral Densing Law. Instead of assigning a fixed capacity to all behavioral periods, ALGN allocates more residual codes only when additional capacity is likely to bring useful information. As a result, high-information periods receive deeper semantic IDs, while routine or redundant periods stop early. This improves downstream representation quality while reducing average tokenization cost.

7.2.2 Ablation Study

Table 4 Ablation study on each component of ALGN.

Method	Capacity(%) ↓	AUC (%)↑	KS (%)↑	Acc (%)↑
RQ-VAE(Lee et al., 2022)	100.00	74.56	39.02	82.44
Heuristic Variable-Length Baseline	86.91	74.43	38.10	82.39
w/o Uncertainty Signal	77.83	75.11	40.21	82.77
w/o Residual Signal	76.24	75.36	41.28	83.16
w/o Regularization	73.42	76.01	42.21	83.35
w/ SID Length Regularization	**63.16**	76.22	42.99	83.47
w/ Random Distribution Regularization	70.15	76.16	42.88	83.45
ALGN	63.47	**76.43**	**43.31**	**83.52**

Compared with the heuristic variable-length baseline, ALGN reduces capacity usage from 86.91% to 63.47%, while improving AUC/KS/Acc at 2.00%/5.21%/1.14% respectively with the adaptive variable SID length control.

Removing either adaptive signal degrades efficiency and performance. Without the uncertainty signal, capacity usage increases to 77.83%, while AUC/KS/Acc decrease by 1.32/3.10/0.75% relative to ALGN. Removing the residual signal yields similar degradation.

Regularization ablation further validates the proposed design. Removing regularization increases capacity usage to 73.42% and decreases AUC/KS/Acc to 76.01%/42.21%/83.35%. Although direct SID-length regularization achieves slightly lower capacity usage, ALGN improves AUC/KS/Acc by 0.21/0.32/0.05%, demonstrating the best overall efficiency–performance trade-off with the incorporation of prior knowledge about the long-tail user behavior nature. It also outperforms random-distribution regularization by 0.27/0.43/0.07 % while reducing capacity usage by 6.68%.

Effect on Densing Law. We also conduct experiments to evaluate the ALGN's effect on our proposed densing law. Results in Figure 11 show that our method effectively decreases the scaling slope to $\sim$0.59, probably by reducing representation space redundancy $\mathcal{E}_\phi$.

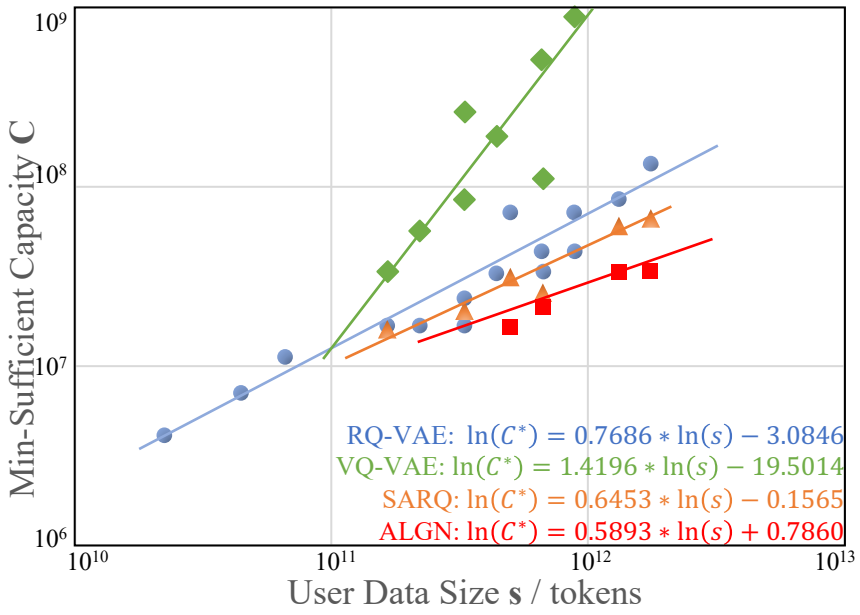


Figure 11 Effect on densing law for ALGN under classification task.

8 Conclusion & Limitation

The belief that more data and larger models always produce better user representations has underpinned years of industrial investment in behavioral sequence modeling. This work presents the first systematic, large-scale empirical test of that assumption. On Alipay's production data spanning hundreds of millions of users, we demonstrate that all three conventional scaling axes, user population, temporal horizon, and model capacity, exhibit clear saturation thresholds. Increasing users beyond 0.03B or temporal windows beyond 60 days delivers rapidly diminishing returns. More strikingly, scaling model parameters past 0.2B under fixed data continues to lower training loss but fails entirely to improve downstream accuracy, revealing that optimization objective and representation quality are not the same thing. These three findings collectively define the **scaling walls** for industrial user behavioral modeling. They converge on a single diagnosis: the bottleneck is not the volume of data, but the predictive information carried by each behavioral token.

This finding sets up our central proposal: a shift from data scaling to *density scaling*. We introduce the concept of *data densing*, drawing an explicit parallel to the densing principle at the model level that has driven progress in compact language model design. Through RQ-VAE tokenization and a matched-budget evaluation protocol, we operationalize this principle: compact, fixed-length discrete codes carry more task-relevant signal than the original raw sequences. The densified representation shifts the scaling curve to a more favorable exponent, enabling smaller encoders operating on shorter inputs to outperform larger models trained on longer raw histories.

For practitioners, our findings translate into concrete, deployable guidelines. The saturation thresholds reported here can directly inform data retention policies, training budget allocation, and model architecture decisions in production environments. The discrete tokens produced by RQ-VAE tokenization are inherently task-agnostic, reusable across user targeting, profile prediction, CTR estimation, and beyond, amortizing the one-time cost of tokenizer training across an entire application portfolio. At a time when the computational and environmental costs of indiscriminate

data scaling are drawing increasing scrutiny, density scaling offers a principled alternative: better representations from the data we already have, not more data collected at ever-increasing cost.

We acknowledge a limitation of the current study: our experiments focus on a single data modality, and the generalizability of the Densing Law to other behavioral modalities, such as video consumption, remains to be validated. We are also further evaluating the proposed law on public benchmark datasets to assess its robustness beyond data collected from a single platform.

These limitations point toward a broader research agenda. We believe that *information density* should be treated as a first-class design objective in large-scale user modeling, alongside model capacity and data volume. The implications extend beyond this work: if density can be measured and optimized for, then the practical question shifts from "how much data can we collect?" to "how do we design representations that maximize signal per token?". We invite the community to explore adaptive tokenization strategies, multi-modal density estimation, and the theoretical foundations of data densing, as a collective step **from scaling walls to densing gains**: toward representation systems that scale not by consuming more, but by extracting more from what they already have.

References

Newsha Ardalani, Carole-Jean Wu, Zeliang Chen, Bhargav Bhushanam, and Adnan Aziz. Understanding scaling laws for recommendation models. *arXiv preprint arXiv:2208.08489*, 2022.

Yue Cao, Xiaojiang Zhou, Jiaqi Feng, Peihao Huang, Yao Xiao, Dayao Chen, and Sheng Chen. Sampling is all you need on modeling long-term user behaviors for ctr prediction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2974–2983, 2022.

Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2942–2951, 2020.

Yunkee Chae, Woosung Choi, Yuhta Takida, Junghyun Koo, Yukara Ikemiya, Zhi Zhong, Kin Wai Cheuk, Marco A Martínez-Ramírez, Kyogu Lee, Wei-Hsiang Liao, et al. Variable bitrate residual vector quantization for audio coding. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

Zheng Chai, Qin Ren, Xijun Xiao, Huizhi Yang, Bo Han, Sijun Zhang, Di Chen, Hui Lu, Wenlin Zhao, Lele Yu, et al. Longer: Scaling up long sequence modeling in industrial recommenders. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, pages 247–256, 2025.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2829. IEEE, 2023.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. In *International Conference on Learning Representations*, 2019.

Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965*, 2025.

Bin Dou, Baokun Wang, Yun Zhu, Xiaotong Lin, Yike Xu, Xiaorui Huang, Yang Chen, Yun Liu, Shaoshuai Han, Yongchao Liu, et al. Transferable and forecastable user targeting foundation model. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 181–190, 2025.

Xinshun Feng, Mingzhe Liu, Yi Qiao, Tongyu Zhu, Leilei Sun, and Shuai Wang. Behavior tokens speak louder: Disentangled explainable recommendation with behavior vocabulary. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 21092–21100, 2026.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

Wei Guo, Hao Wang, Luankang Zhang, Jin Yao Chin, Zhongzhou Liu, Kai Cheng, Qiushi Pan, Yi Quan Lee, Wanqi Xue, Tingjia Shen, et al. Scaling new frontiers: Insights into large recommendation models. *arXiv preprint arXiv:2412.00714*, 2024.

Chuan He, Yang Chen, Wuliang Huang, Tianyi Zheng, Jianhu Chen, Bin Dou, Yice Luo, Yun Zhu, Baokun Wang, Yongchao Liu, et al. Learning unified user quantized tokenizers for user representation. *arXiv preprint arXiv:2508.00956*, 2025.

Yupeng Hou, Jianmo Ni, Zhankui He, Noveen Sachdeva, Wang-Cheng Kang, Ed H Chi, Julian McAuley, and Derek Zhiyuan Cheng. Actionpiece: Contextually tokenizing action sequences for generative recommendation. *arXiv preprint arXiv:2502.13581*, 2025.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Iris AM Huijben, Matthijs Douze, Matthew Muckley, Ruud JG Van Sloun, and Jakob Verbeek. Residual quantization with implicit neural codebooks. *arXiv preprint arXiv:2401.14732*, 2024.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE, 2018.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. Matryoshka representation learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 11513–11522. IEEE, 2022.

Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. Multi-interest network with dynamic routing for recommendation at tmall. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 2615–2623, 2019.

Qijiong Liu, Hengchang Hu, Jiahao Wu, Jieming Zhu, Min-Yen Kan, and Xiao-Ming Wu. Discrete semantic tokenization for deep ctr prediction. In *Companion Proceedings of the ACM Web Conference 2024*, pages 919–922, 2024a.

Qijiong Liu, Jieming Zhu, Zhaocheng Du, Lu Fan, Zhou Zhao, and Xiao-Ming Wu. Learning multi-aspect item palette: A semantic tokenization framework for generative recommendation. *arXiv preprint arXiv:2409.07276*, 2024b.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. Practice on long sequential user behavior modeling for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2671–2679, 2019.

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 & Knowledge Management*, pages 2685–2692, 2020.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36:10299–10315, 2023.

Qin Ren, Zheng Chai, Xijun Xiao, Yuchao Zheng, and Di Wu. Longretriever: Towards ultra-long sequence based candidate retrieval for recommendation. *arXiv preprint arXiv:2508.15486*, 2025.

Jiwan Seo and Joonhyuk Kang. Rate-adaptive quantization: A multi-rate codebook adaptation for vector quantization-based generative models. *arXiv preprint arXiv:2405.14222*, 2024.

Tingjia Shen, Hao Wang, Chuhan Wu, Jin Yao Chin, Wei Guo, Yong Liu, Huifeng Guo, Defu Lian, Ruiming Tang, and Enhong Chen. P-Law: Predicting quantitative scaling law with entropy guidance in large recommendation models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Kyuyong Shin, Hanock Kwak, Su Young Kim, Max Nihlén Ramström, Jisu Jeong, Jung-Woo Ha, and Kyung-Min Kim. Scaling law for recommendation models: Towards general-purpose user representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4596–4604, 2023. doi: 10.1609/aaai.v37i4.25582.

Zihua Si, Lin Guan, ZhongXiang Sun, Xiaoxue Zang, Jing Lu, Yiqun Hui, Xingchao Cao, Zeyu Yang, Yichen Zheng, Dewei Leng, et al. Twin v2: Scaling ultra-long user behavior sequence modeling for enhanced ctr prediction at kuaishou. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 4890–4897, 2024.

Fei Sun, Jun Liu, Jian Wu, Chao Pei, Xiaoyu 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)*, pages 1441–1450, 2019.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Huimu Wang, Xingzhi Yao, Yiming Qiu, Qinghong Zhang, Haotian Wang, Yufan Cui, Songlin Wang, Sulong Xu, and Mingming Li. Towards efficient and generalizable retrieval: Adaptive semantic quantization and residual knowledge transfer. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4987–4992, 2026.

Yutian Xiao, Shukuan Wang, Binhao Wang, Zhao Zhang, Yanze Zhang, Shanqi Liu, Chao Feng, Xiang Li, and Fuzhen Zhuang. Mars: Modality-aligned retrieval for sequence augmented ctr prediction. *arXiv preprint arXiv:2509.01184*, 2025.

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.

Gaowei Zhang, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, and Ji-Rong Wen. Scaling law of large sequential recommendation models. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pages 444–453. ACM, 2024a. doi: 10.1145/3640457.3688129.

Gaowei Zhang, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, and Ji-Rong Wen. Scaling law of large sequential recommendation models. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pages 444–453, 2024b.

Weinan Zhang, Tianqi Liu, Jun Luo, Likang Zou, Gang Liu, and Xing Xiong. Deep learning for matching in search and recommendation. *Foundations and Trends in Information Retrieval*, 14(2–3):102–288, 2021.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Zhicheng Zhang, Zhaocheng Du, Jieming Zhu, Jiwei Tang, Fengyuan Lu, Wang Jiaheng, Song-Li Wu, Qianhui Zhu, Jingyu Li, Hai-Tao Zheng, et al. Length-adaptive interest network for balancing long and short sequence modeling in ctr prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 28627–28635, 2026.

Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 5941–5948, 2019.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International journal of computer vision*, 130(9):2337–2348, 2022.

Wen-Ji Zhou, Yuhang Zheng, Yinfu Feng, Yunan Ye, Rong Xiao, Long Chen, Xiaosong Yang, and Jun Xiao. Encode: Breaking the trade-off between performance and efficiency in long-term user behavior modeling. *IEEE Transactions on Knowledge and Data Engineering*, 37(1):265–277, 2024.

Jieming Zhu, Mengqun Jin, Qijiong Liu, Zexuan Qiu, Zhenhua Dong, and Xiu Li. Cost: Contrastive quantization based semantic tokenization for generative recommendation. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pages 969–974, 2024.

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*, pages 1567–1577, 2024.

9 Appendix

9.1 Training Loss Convergence across Temporal Horizons

Training loss during pretraining is another signal for scaling comparison, which reflects the optimization behavior of the pretraining objective. This loss is computed according to Eq. 2.

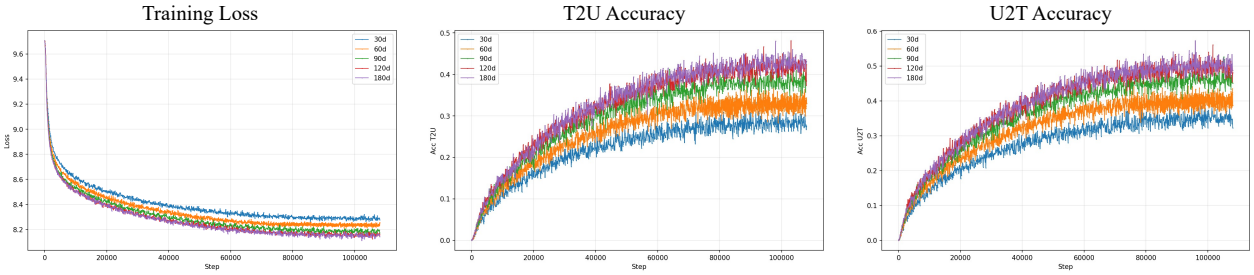


Figure 12 Contrastive alignment loss over 100,000 training steps for models pre-trained on behavioral sequences spanning $D \in \{30, 60, 90, 120, 180\}$ days. While all configurations converge, the reduction in final loss diminishes sharply as D increases. The curves for $D \geq 90$ days nearly overlap, indicating that additional historical days contribute minimal new optimization signal, consistent with the downstream metric saturation observed in Section 4.

It can be observed in Figure 12 As the behavioral timespan scales from 30 to 180 days, the training loss converges to progressively lower values, decreasing from approximately 8.28 to 8.15. Meanwhile, the contrastive-learning accuracies exhibit a consistent positive scaling trend: T2U accuracy increases from approximately 0.29 to 0.42, while U2T accuracy rises from approximately 0.35 to 0.50, indicating that larger-scale behavioral inputs improve representation alignment in both directions.

9.2 Generalization across different data sources

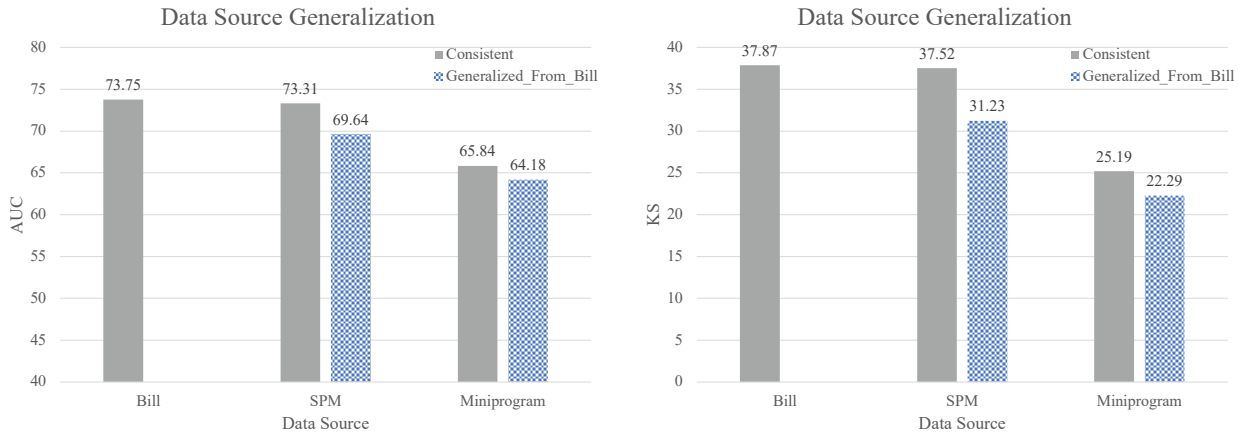


Figure 13 Data source generalization.

As shown in Figure 13 , the compression strategy exhibits strong generalizability and adaptability across heterogeneous data sources. When the compression configuration derived from the Bill dataset is directly transferred to SPM and Miniprogram, it preserves approximately 95.0% and 97.5% of the AUC achieved under the source-consistent setting, respectively. A similar trend is observed for KS, with retention rates of 83.2% on SPM and 88.5% on Miniprogram. These results indicate that the compression strategy is not strongly dependent on source-specific data characteristics and can

maintain competitive predictive performance under cross-source distribution shifts, demonstrating its robustness and practical applicability across diverse data environments.