# LLMs Reading the Rhythms of Daily Life: Aligned Understanding for Behavior Prediction and Generation

Anonymous ACL submission

## Abstract

Human daily behavior unfolds as complex sequences shaped by intentions, preferences, and context. Effectively modeling these behaviors is crucial for intelligent systems such as personal assistants and recommendation engines. While recent advances in deep learning and behavior pre-training have improved behavior prediction, key challenges remain—particularly in handling long-tail behaviors, enhancing interpretability, and supporting multiple tasks within a unified framework. Large language models (LLMs) offer a promising direction due to their semantic richness, strong interpretability, and generative capabilities. However, the structural and modal differences between behavioral data and natural language limit the direct applicability of LLMs. To address this gap, we propose Behavior Understanding Alignment (BUA), a novel framework that integrates LLMs into human behavior modeling through a structured curriculum learning process. BUA employs sequence embeddings from pretrained behavior models as alignment anchors and guides the LLM through a three-stage curriculum, while a multi-round dialogue setting introduces prediction and generation capabilities. Experiments on two real-world datasets demonstrate that BUA significantly outperforms existing methods in both tasks, highlighting its effectiveness and flexibility in applying LLMs to complex human behavior modeling. The code is available at https://anonymous.4open.science/r/dasjijio-21B2/

## 1 Introduction

Human daily life unfolds as a sequence of behaviors—ranging from habitual routines to spontaneous actions—each reflecting underlying intentions, preferences, and contextual factors. Accurately modeling and understanding these human daily behaviors is fundamental to a wide range of intelligent systems, including personalized assistants, recommender engines, and context-aware services (Chung and Lee, 2018; Tulshan and Dhage, 2019; Savcisens et al., 2023). Traditional approaches (Zhu et al., 2017; Chen et al., 2018; Yuan et al., 2023), particularly those based on deep learning, have primarily focused on behavior prediction: learning to predict the next behavior event based on historical sequences (Kang and McAuley, 2018; Sun et al., 2019). Recently, with the increasing availability of large-scale behavioral datasets and inspired by the success of pre-training paradigms in natural language processing (NLP) (Radford et al., 2019; Dubey et al., 2024), behavior pre-training has emerged as a promising technique. These methods (Gong et al., 2024; Savcisens et al., 2024; Zhai et al., 2024) exploit vast human daily behavioral corpora to capture intricate temporal dependencies and latent patterns, leading to significant improvements in predictive accuracy.

Despite these advances, existing human daily behavior modeling approaches suffer from critical limitations. First, they struggle to model long-tail behaviors, actions that occur infrequently or are newly emerging, due to inherent data sparsity issues (Hu et al., 2025; Kim et al., 2024). Second, their "black-box" nature offers limited insight into the decision-making process, creating a gap between model's predictions and human-interpretable reasoning (Lei et al., 2024). Third, most models are designed for a single task, focusing on either prediction or generation, and lack the flexibility to handle both within a unified framework.

Recent developments in Large Language Models (LLMs) offer a powerful new direction for addressing these challenges. LLMs provide several distinct advantages: (1) Their rich semantic representations, learned from vast textual corpora, can enhance the modeling of long-tail behaviors by providing crucial contextual understanding (Liu et al., 2024; Sheng et al., 2024). (2) Trained on extensive human-generated text, LLMs can process

and articulate behavioral patterns in a textual format that aligns more closely with human cognition, thereby enhancing model interpretability. (3) Their **inherent generative capabilities** support multitask learning through natural language, enabling both behavior prediction and generation within a single, unified model. Overall, integrating LLMs presents a clear opportunity to overcome the core limitations of traditional behavior modeling.

However, human daily behavioral data differs fundamentally from textual data in both structure and modality. As LLMs are trained primarily on natural language, they cannot directly interpret the features or outputs of conventional human daily behavior modeling pipelines. To address this challenge, we propose the Behavior Understanding Alignment (BUA) framework, a novel approach that integrates LLMs into human daily behavior modeling, which first leverages sequence embeddings from a pretrained behavior model as alignment anchors and guides the LLM through a structured three-stage curriculum, designed to progressively bridge the modality gap. And we further introduce a multi-round dialogue setting that incrementally incorporates prediction and generation, forming a natural reasoning chain: understanding, then predicting and generating, which effectively leveraging the model's comprehension of human daily behavior sequences to enhance both prediction and generation performance. The contributions of this work are summarized as follows:

- We are the first to train an LLM to explicitly *understand* daily behavior sequences, by aligning behavioral and language modalities, as a foundation to improve downstream prediction and generation.
- We introduce the Behavior Understanding Alignment framework, which uniquely combines a three-stage curriculum learning pipeline with a multi-round dialogue mechanism to synergistically enhance the model's capabilities in understanding, predicting, and generating human behaviors.
- Experimental results on two real-world datasets demonstrate that BUA achieves state-of-the-art performance in both prediction and generation tasks. Comprehensive ablation studies further validate the critical role of our structured curriculum and dialogue-based reasoning process in achieving these results.

## 2 Related Work

### 2.1 Behavior Modeling

Modeling daily human behavior hinges on capturing underlying regularities in user behavior sequences, typically through two tasks: behavior prediction and behavior generation. Early behavior prediction models, such as TRNN (Zhu et al., 2017), utilized time-difference-aware embeddings to enhance temporal modeling. As datasets expanded, transformer-based pretraining methods like BehaveGPT (Gong et al., 2025) and Life2Vec (Savcisens et al., 2024) became prevalent, significantly improving predictive accuracy. However, these methods often struggle with long-tail behaviors due to limited sample diversity. For behavior generation, early rule-based and agent-based models (Kim et al., 2019; Pfoser and Wenk) relied on hand-crafted logic, limiting their ability to capture real-world complexity. SAND (Yuan et al., 2023) advanced this by using neural stochastic differential equations, enabling more realistic dynamics without fixed rules, though its static generation parameters limit adaptability. Recent work fine-tunes LLMs on shopping-session logs for behavior simulation (Lu et al., 2025), while BUA focuses on embedding alignment and explicit behavior understanding. More recently, D2A (Wang et al., 2024) trained an LLM as a cognitively inspired agent guided by a dynamic value system, enhancing behavioral diversity and flexibility. However, it underutilizes the LLM's potential for sequence generation based on a deep, multimodal understanding of behavioral context.

### 2.2 Alignment in Recommendation

Our behavior prediction task is defined as predicting the next behavior based on the user's most recent $L$ behavioral events, which is similar to sequential recommendation. The two differ in focus: behavior prediction emphasizes recurring daily behaviors, whereas sequential recommendation often targets novel items. To the best of our knowledge, no prior work has focused on explicitly aligning LLMs with the underlying semantic representations of entire behavior sequences, so we draw upon related research in sequential recommendation. Existing work that incorporates large language models (LLMs) into sequential recommendation can be broadly categorized into two directions: (1) LLMs as standalone recommendation systems (Tan et al., 2024; Kim et al., 2024; Liao et al., 2024; Zhang

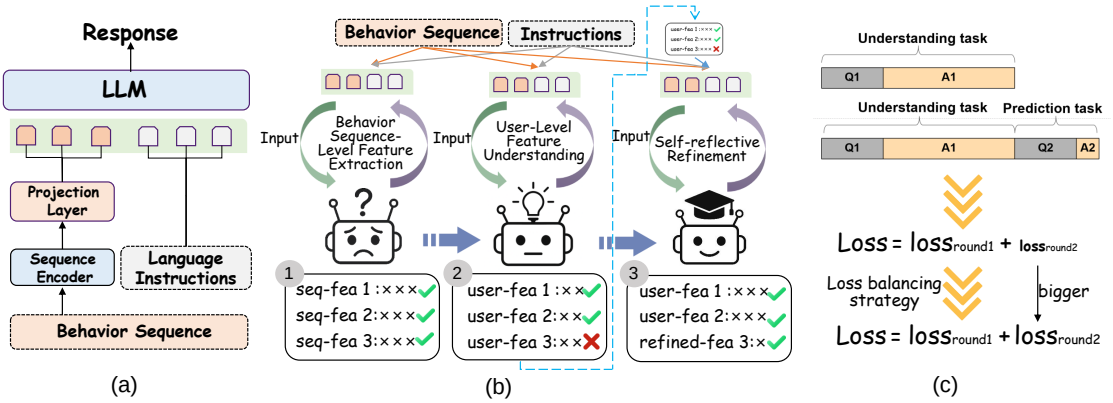


Figure 1: The framework of BUA. (a) the modality conversion process using sequence embedding. (b) curriculum for Behavior Understanding Alignment: seq-fea, user-fea, and refined-fea represent features learned in Stage 1 (Sequence-Level), Stage 2 (User-Level), and Stage 3 (Self-Reflection). The ✓ and × marks indicate the correctness of the learned features.(c) Understanding-enhanced prediction and generation via multi-round dialogue.

et al., 2025), and (2) LLMs as enhancers of traditional systems (Ren et al., 2024; Liu et al., 2024; Hu et al., 2025; Wang et al., 2024). However, works in the first category do not explore the cross-modal connection or establish a model's explicit understanding of user behavior sequences. Their item-embedding alignment-based approach limits the ability to fully leverage collaborative knowledge. The latter line of work offers superior inference efficiency compared to LLM-based systems. However, they inherit the limitations of traditional deep learning models, including poor generalization to new items and tasks, and limited interpretability due to their black-box nature(Lei et al., 2024).

## 3 Methodology

### 3.1 Problem Formulation

Let $x_i$ denote a basic behavioral event, represented as a four-tuple $(d_i, t_i, l_i, b_i)$, where $d_i$ is the Day of week index, $t_i$ the Timestamp ID, $l_i$ the Location ID, and $b_i$ the Behavior Type ID. The behavior type captures high-level daily activities—such as exercising or gaming—rather than fine-grained actions. We consider two related tasks under this representation:

**Behavior Prediction and Generation.** Given a user's recent behavior sequence $X_{\text{seq}} = \{x_1, x_2, \ldots, x_{L_1}\}$, the model is tasked with either (1) predicting the next behavior $b \in \mathcal{B}$, or (2) generating a future sequence $Y_{\text{seq}} = \{y_1, y_2, \ldots, y_{L_2}\}$, where each $y_i$ is a four-tuple $(d_i, t_i, l_i, b_i)$.

### 3.2 Overview

The framework of BUA, is illustrated in Figure 1. To effectively bridge the gap between numerical behavior data and textual semantic space, we first introduce a **Sequence Embedding Alignment** module, as detailed in Figure 1(a). Given a user behavior sequence $X_{\text{seq}}$, we utilize behaveGPT (Gong et al., 2025), a model pretrained on large-scale behavioral data, as the behavior encoder $g_\phi$. The penultimate hidden state is projected into a unified representation space via a lightweight two-layer MLP, producing the behavior sequence embedding $H_{\text{seq}}$:

$$H_{\text{seq}} = \text{MLP}(g_\phi(X_{\text{seq}})) \tag{1}$$

This embedding $H_{\text{seq}}$ serves as a **cross-modal anchor**, enabling the LLM to perceive and reason about continuous behavior signals alongside textual instructions. Notably, although we instantiate $g_\phi$ with BehaveGPT in the main experiments, BUA is not tied to a specific behavior encoder, and to optimize efficiency, $g_\phi$ remains completely frozen throughout training.

Building upon this alignment foundation, we propose a three-stage structured curriculum to progressively enhance behavior understanding: (1) Sequence-Level Understanding, (2) User-Level Feature Modeling, and (3) Self-Reflective Refinement. In the second stage, we additionally incorporate a multi-round dialogue setup that integrates prediction and generation tasks. Throughout these stages, the model is optimized by maximizing the likelihood of the target output $y$ given the multimodal input (text instruction $X_{\text{Ins}}$ and behavior embedding $H_{\text{seq}}$):

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} \log P_\theta\left(y_t \mid y_{<t}, X_{\text{Ins}}, H_{\text{seq}}\right), \tag{2}$$

where $N$ denotes the length of the response $y$.

### 3.3 Curriculum for Behavior Understanding Alignment

We propose a three-stage curriculum for user behavior understanding tasks, progressing from simple

to complex. The first stage targets sequence-level understanding, the second emphasizes user-level feature modeling, and the third incorporates self-reflective refinement to enhance behavioral representations. All tasks are detailed in Appendix A.

#### 3.3.1 Stage 1: Behavior Sequence-Level Feature Extraction

In this initial stage, the model learns to interpret behavior sequence embeddings from language modalities, building a foundation for deeper understanding of user daily behaviors in later stages. Based on empirical insights, we introduce three basic tasks:

- **Historical Sequence Reconstruction**: The model reconstructs the original behavioral sequence in natural language, capturing key temporal, spatial, and behavioral transitions. This event-level task establishes the groundwork for multimodal understanding.
- **Current Scene Summary**: The model summarizes the user's recent context over the past two hours (e.g., "morning commute"), requiring it to extract and generalize key patterns, advancing its sequence-level understanding.
- **Future Scene Inference**: The model predicts the user's likely context in the next two hours (e.g., "evening commute" or "pre-bedtime leisure"), demonstrating ability to analyze sequence and temporal trends.

In practice, we also introduce simple user-level inference tasks at this stage, such as home/workplace location identification and user hobby inference, which provide a natural transition to the more complex user-level understanding required in Stage 2. To enhance training efficiency, we freeze both the sequence encoder and the LLM during this stage, allowing only the parameters of the projection layer (MLP) to remain trainable, following the common acceleration training techniques for large multimodal language models (Liu et al., 2023).

#### 3.3.2 Stage 2: User-Level Feature Understanding

In this phase, the focus shifts to capturing deeper user-level features embedded within behavior sequences. It is not sufficient for the model to recognize superficial transitions between time, location, or activity; instead, it must extract the user features that characterize the underlying mechanisms behind observed behaviors. Such features are critical for understanding behavioral regularities and informing prediction and generation tasks. Inspired by how humans infer features regarding users' lifestyle in terms of what they behave, we design the following tasks to guide user-level feature learning:

- **User Key Behavior Identification**: The model identifies semantically rich behaviors that are frequent or mark transitions between daily phases (e.g., taking the subway after work indicates a shift to evening leisure). They are critical for inferring intent evolution.
- **User Behavior Pattern Discovery**: The model detects recurring subsequences and consistent temporal-spatial patterns (e.g., watching TikTok during commutes), revealing deeper preferences and routines.
- **User Feature Summarization**: The model abstracts high-level user features (e.g., This user prefers light entertainment during his evening commute), providing a more essential understanding of user behavior.

These tasks are intentionally sequenced from simple to complex, forming a structured learning path. To support effective user-level understanding, this hierarchical curriculum is enforced during training: the model must perform User Key Behavior Identification first, followed by Behavior Pattern Discovery, and finally User Feature Summarization. Additionally, during training at this stage, we freeze the sequence model parameters while allowing the parameters of the projection layer and LLM to be adjustable.

#### 3.3.3 Stage 3: Self-reflective Refinement

In this phase, we introduce Self-Reflective Refinement to enhance the model's understanding of user features. Despite the training in the first two phases, the performance on the User Feature Summarization task remained suboptimal. We evaluated the generated summaries and manually inspected low-scoring responses, finding that there are some recurring issues, such as unclear relationships between behavioral features. As these issues did not stem from a fundamental misunderstanding, the model might actually have developed a comprehensive understanding of user features during the Sequence-Level Understanding and User-Level Feature Extraction stages. Instead, the model only requires more "thinking" to generate reasonable and accurate user features. To address this, we propose a self-reflective iteration strategy that empowers the model to identify and correct its own shortcomings. Specifically, we summarize the recurring issues,

and design targeted correction criteria, guiding the model to review and revise its earlier outputs based on clear feedback. The task is defined as follows:

- **Self-Reflective Refinement**: The model reviews low-quality user feature summaries, identifies key issues, and refines them using its understanding of sequence embeddings, producing more accurate and coherent summaries.

Based on the foundations established in the first two stages, this strategy leverages the model's reasoning capabilities for iterative improvement, resulting in more robust and accurate user feature representations. During this phase, we freeze the sequence encoder and projection layer while keeping LLM parameters trainable.

### 3.4 Multi-round Dialogue Setup

To develop behavior prediction and generation capabilities, we introduce a multi-round dialogue in the second stage of User-Level Feature Understanding. This approach enables the model to simultaneously refine prediction and generation skills while deepening its understanding of user behavior features. In this setup, the model starts with the Key Semantic Behavior Recognition task in the first round, progressively completes all user-level behavior understanding tasks, and concludes with the corresponding prediction or generation tasks in the final round. By leveraging intermediate understanding and analysis from earlier rounds, the model enhances the accuracy and effectiveness of downstream tasks. The optimization loss for this multi-round dialogue setting is defined as:

$$Loss = \frac{1}{\sum_{\mathrm{i=1}}^{\mathrm{N}} \mathrm{T_i}} \Bigg( \sum_{i=1}^{N} \sum_{t=1}^{T_i} \log P_\theta \big( y_t^{(i)} | \; y_{<t}^{(i)}, X_{\text{Ins}}^{(i)}, Y^{(i)}, X_{\text{seq}} \big) \Bigg), \tag{3}$$

where $N$ is the total number of dialogue rounds, $T_i$ is the number of tokens in the answer for the $i$-th round, $\theta$ denotes the LLM parameters,$y_t^{(i)}$ is the $t$-th token of the $i$-th round's output, $X_{\text{Ins}}^{(i)}$ is the input Instruction, and $Y^{(i)}$ represents the corresponding answer for the $i$-th round.

However, multi-round dialogues risk imbalanced training across different rounds. Rewriting the loss from the token level to the round level, we get

$$Loss_{\text{multi-turn}} = -\sum_{i=1}^{N} \frac{T_i}{\sum_{i=1}^{N} T_i} loss_{\mathrm{i}}, \tag{4}$$

where $loss_i$ denotes the average loss for the $i$-th round. This means that rounds with longer answers dominate the overall loss, while those with shorter outputs receive less attention at the round level. This is problematic in practice, as understanding tasks typically involve long outputs (often exceeding 100 tokens), whereas prediction tasks only output the predicted behavior type (often fewer than 5 tokens). As a result, the model struggles to effectively learn shorter prediction tasks.

To address this issue, we introduce a simple yet effective loss balancing strategy that ensures equal attention across rounds. Specifically, we apply a weight $W_i = \frac{\sum_{i=1}^{N} T_i}{N T_i}$ that is inversely proportional to the length of the answer in each round, encouraging balanced learning across all rounds. The final loss function becomes:

$$Loss_{\text{weighted}} = -\frac{1}{N} \sum_{i=1}^{N} \log P_\theta \Big( y_t^{(i)} \; | \; y_{<t}^{(i)}, \; X_{\text{Ins}}^{(i)}, \; Y^{(i)}, \; X_{\text{seq}} \Big) \tag{5}$$

This balanced loss formula significantly enhances the performance of the model on the behavior prediction task without significantly reducing the effectiveness on the understanding task.

## 4 Experiments

### 4.1 Experimental Settings

**Datasets**. We evaluated our model on two real-world user behavior datasets: **Behavior dataset**: This dataset is derived from the user's mobile phone logs. After desensitization, it includes 37 daily behaviors that cover a wide range of life scenarios, including activities related to learning, work, leisure, and more. **Tencent Dataset (Shao et al., 2024)**: This dataset is derived from the user's social network and the user's movement trajectory. It includes 14 human behavior intentions, such as eating, going home, working, etc.

For both datasets, we split the users in a ratio of 8:1:1 to create training, validation, and test datasets. For more detailed information about the datasets and their splits, please refer to the Appendix B.

**Evaluation Metrics**. For **behavior prediction task**, we adopt commonly used metrics, weighted precision ($Prec_w$) and weighted recall ($Rec_w$)(equivalent to HR@1), to evaluate the overall prediction performance of the model. Additionally, user data is often unevenly distributed and exhibits a clear long-tail distribution in practice (Kim

Table 1: Experiment Results on Next Behavior Prediction

| Category | Method | Behavior Dataset | | | | | | Tencent Dataset | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $Rec_w$ | $Prec_w$ | $Overall$ | $Head$ | $Medium$ | $Tail$ | $Rec_w$ | $Prec_w$ | $Overall$ | $Head$ | $Medium$ | $Tail$ |
| Traditional | SASRec | 0.546 | 0.535 | 0.291 | 0.420 | 0.340 | 0.222 | 0.328 | 0.269 | 0.097 | 0.29 | 0.045 | 0.021 |
| | behaveGPT | 0.567 | 0.551 | 0.206 | 0.442 | 0.354 | 0.027 | 0.509 | 0.426 | 0.113 | 0.537 | 0.000 | 0.000 |
| LLM-Enhanced | PITuning | 0.617 | 0.603 | 0.408 | 0.481 | 0.444 | 0.361 | 0.524 | 0.466 | 0.120 | 0.546 | 0.009 | 0.000 |
| | AlphaFuse | 0.578 | 0.575 | 0.242 | 0.457 | 0.380 | 0.075 | 0.507 | 0.435 | 0.118 | 0.547 | 0.001 | 0.000 |
| LLM-Based | Deepseek-V3 | 0.492 | 0.495 | 0.237 | 0.330 | 0.265 | 0.191 | 0.318 | 0.282 | 0.119 | 0.303 | 0.083 | 0.038 |
| | TALLRec | 0.617 | 0.607 | 0.398 | 0.452 | 0.434 | 0.355 | 0.561 | 0.543 | 0.134 | 0.513 | 0.044 | 0.019 |
| | A-LLMRec | 0.584 | 0.557 | 0.348 | 0.422 | 0.394 | 0.299 | 0.539 | 0.523 | 0.140 | 0.542 | 0.037 | 0.025 |
| | CoLLM | 0.618 | 0.596 | 0.408 | 0.453 | 0.448 | 0.363 | 0.560 | 0.543 | 0.152 | 0.530 | 0.067 | 0.034 |
| | LLaRA | 0.615 | 0.608 | 0.404 | 0.462 | 0.439 | 0.361 | 0.564 | 0.545 | 0.152 | 0.543 | 0.060 | 0.033 |
| | **BUA** | **0.644** | **0.642** | **0.471** | **0.538** | **0.489** | **0.446** | **0.600** | **0.574** | **0.207** | **0.62** | **0.114** | **0.041** |
| | Improv | 4.2% | 5.6% | 15.4% | 11.9% | 9.2% | 22.9% | 6.4% | 5.3% | 36.2% | 13.4% | 37.4% | 7.9% |

Table 2: Experiment Result on Behavior Sequence Generation

| Method | Event | | | Timestamp | | | Location | | |
|---|---|---|---|---|---|---|---|---|---|
| | Bleu ↑ | TVD ↓ | JSD ↓ | Bleu ↑ | TVD ↓ | JSD ↓ | Bleu ↑ | TVD ↓ | JSD ↓ |
| behaveGPT | 0.009 | 0.945 | 0.632 | — | — | — | — | — | — |
| SAND | 0.142 | 0.304 | 0.083 | 0.344 | 0.204 | 0.038 | — | — | — |
| D2A | 0.315 | 0.183 | 0.039 | 0.287 | 0.223 | 0.049 | 0.396 | 0.529 | 0.173 |
| Ours | 0.354 | 0.140 | 0.020 | 0.541 | 0.147 | 0.020 | 0.711 | 0.065 | 0.005 |

et al., 2024). Following relevant work (Liu et al., 2019; Shi et al., 2024), we construct a long-tail intent test set and adopt global accuracy(denoted as $Overall$), high-frequency behavior accuracy(denoted as $Head$), medium-frequency behavior accuracy(denoted as $Medium$), and long-tail behavior accuracy (denoted as $Tail$) to fairly evaluate the model's performance across different behavior categories. For the calculation methods and additional details on all six metrics, please refer to the Appendix C. For **behavior generation task**, we adopt commonly used metrics $BLEU$, $TVD$, and $JSD$ to measure the time, location, and behavior similarity between the generated sequence and the real sequence data. The calculation methods for the metrics are outlined in the Appendix D.

**Baselines**. For **behavior prediction task**, We selected representative algorithms from various categories to compare with our proposed algorithm. For traditional deep learning methods, we chose SASRec (Kang and McAuley, 2018), BehaveGPT (Gong et al., 2025). For pure LLM-based prediction methods, we selected DeepSeek-V3 (DeepSeek-AI et al., 2025) and TallRec (Bao et al., 2023). For methods that use modality fusion and LLM as recommendation systems (similar to ours), we selected A-LLMRec (Kim et al., 2024), CoLLM (Zhang et al., 2025), and LLaRa (Liao et al., 2024). For methods that employ modality fusion and LLM as recommendation system enhancers, we selected PI-Tuing (Gong et al., 2024) and AlphaFuse (Hu et al., 2025). For **behavior generation task**, We chose SAND (Yuan et al., 2023), a representative method based on deep learning, and D2A (Wang et al., 2024), which uses LLM for user behavior activity generation based on Maslow's Theory. For further details on the baselines, please refer to the Appendix E. For fairness, all LLM-based baselines that require external behavior embeddings, including A-LLMRec, CoLLM, and LLaRa, use the same frozen BehaveGPT encoder as BUA; only the alignment and training strategies differ.

**Implementation Details**. The hardware used in this experiment consists of 8 NVIDIA A100 40G GPUs. We chose Qwen2.5-7B (Qwen et al., 2025) as the backbone model for the experiment. More details are in the Appendix F.

## 4.2 Overall Performance

### 4.2.1 Behavior Prediction Experiment

We evaluated our model on two real-world datasets against all baselines. Our model consistently outperformed all baselines across all metrics, confirm-

ing its effectiveness, as shown in Table 1. Further insights are detailed below.

**Overall Comparison**. The results show that our method(BUA) outperforms all baselines on both $Prec_w$ and $Rec_w$ under the real data distribution. Additionally, most LLM-based methods surpass traditional models like SASRec, underscoring the value of semantic information in behavior prediction. Notably, item embedding fusion approaches (e.g., LLaRA, A-LLMRec) offer no clear advantage over the pure LLM method, TallRec, indicating that item embeddings alone are insufficient to fully leverage sequential knowledge.

**Different Categories of Behaviors Comparison**. The results show that our method(BUA) achieves substantial gains across high-frequency, medium-frequency, and long-tail behaviors, with an overall average improvement of 25.8% over the best baseline on both datasets. While the LLM-based TallRec outperforms SASRec by over 50% on long-tail behaviors, it shows only a 10% gain on high-frequency ones on Behavior dataset, emphasizing the importance of semantic information for long-tail prediction. Although the item-embedding fusion method LLaRA slightly outperforms the pure LLM TallRec, the margin is small compared to BUA, further confirming the advantage of BUA.

#### 4.2.2 Behavior Generation Experiment

We evaluated our method on the Behavior dataset against all baselines. To ensure fair comparison with SAND, which outputs fixed-time behaviors, we generated one day of future behaviors. Our model consistently outperformed all baselines across all metrics, demonstrating strong robustness and effectiveness (see Table 2). Note that $-$ in table indicates the model lacks generation capability and is not applicable for evaluation. Additionally, we found that BehaveGPT, despite large-scale pretraining, performs poorly on the generation task even after fine-tuning, revealing limited capability. While SAND generates more accurate timestamps than D2A, it lags in behavior accuracy. These results underscore the importance of behavioral semantics and the difficulty LLMs face with temporal and numerical features. Our method addresses this by first understanding and summarizing behavioral patterns, leading to superior performance. In addition, we conducted experiments using the generated data on downstream prediction tasks, refer to Appendix G for details.

Table 3: Cross-Model Enhancement via Behavioral Understanding Transfer

| Method | $Rec_w$ | $Prec_w$ | $Overall$ | $Head$ | $Medium$ | $Tail$ |
|---|---|---|---|---|---|---|
| Tallrec | 0.609 | 0.587 | 0.384 | 0.460 | 0.440 | 0.319 |
| Tallrec-cross | 0.620 | 0.610 | 0.405 | 0.467 | 0.463 | 0.341 |
| Llara | 0.605 | 0.585 | 0.362 | 0.460 | 0.412 | 0.296 |
| Llara-cross | 0.610 | 0.595 | 0.403 | 0.468 | 0.466 | 0.335 |

### 4.3 Cross-Model Enhancement via Behavioral Understanding Transfer

To further validate the effectiveness of our model's behavior understanding, we evaluate whether its extracted user features could enhance other models. These features were added to TallRec and LLARA, resulting in TallRec-cross and LLARA-cross. Specifically, BUA generated user summaries for 20,000 samples to supplement each model's input. As shown in Table 3, this consistently improved prediction performance, with notable gains on long-tail behaviors.

Table 4: Ablation Study for Behavior Prediction Task

| Method | $Rec_w$ | $Prec_w$ | $Overall$ | $Head$ | $Medium$ | $Tail$ |
|---|---|---|---|---|---|---|
| Ours | 0.644 | 0.642 | 0.471 | 0.538 | 0.489 | 0.446 |
| w/o stage1 | 0.613 | 0.607 | 0.370 | 0.485 | 0.452 | 0.286 |
| w/o stage2 | 0.587 | 0.577 | 0.456 | 0.400 | 0.159 | 0.291 |
| w/o stage3 | 0.592 | 0.586 | 0.334 | 0.420 | 0.357 | 0.300 |
| w/o loss balance | 0.560 | 0.552 | 0.285 | 0.385 | 0.364 | 0.197 |
| item-emb | 0.631 | 0.626 | 0.465 | 0.523 | 0.491 | 0.437 |

### 4.4 Ablation Study

We ablate key components on the Behavior dataset by removing: (1) Stage 1: Sequence-Level Feature Understanding (w/o stage1), (2) Stage 2: User-Level Feature Extracting (w/o stage2), and (3) Stage 3: Self-reflective Refinement (w/o stage3).

**Behavior Prediction** To further analyze the contributions of each component to the Behavior Prediction task, we additionally evaluate: (4) Removal of loss balancing strategy of multi-turn dialogue in the second stage (w/o loss balance), (5) Use of item embedding instead of sequence embedding for modality alignment (item-emb). The results are shown in Table 4, with key findings as follows: All components contribute to overall performance, with the loss balancing strategy in the second-stage multi-turn dialogue having the most significant impact on prediction. Without it, the model favors the understandng task due to its longer output, reducing prediction effectiveness. Although the first and third stages do not directly target prediction, they improve behavior sequence understanding, indirectly enhancing prediction. Finally, replacing sequence embeddings with item embeddings for

modality fusion leads to performance degradation, confirming the superiority of sequence-level representations. Furthermore, we conducted a more detailed analysis of the sources of model errors, which can be found in the appendix I.

**Behavior Generation** All design components contribute to the model's overall performance. Removing the user-level feature extraction in the second stage has the greatest impact, showing that explicitly generating user behavior features enhances understanding and guides future behavior generation. Additionally, the behavior understanding tasks in the first and third stages improve the model's grasp of behavior sequences, as their removal weakens pattern understanding and indirectly reduces generation performance. Please refer to Appendix H for detailed results. In addition, SASRec replacement and efficiency results further support BUA's encoder-agnostic design and comparable inference cost; see Appendices M and K.

### 4.5 Qualitative and Quantitative Analysis of Interpretability

To comprehensively evaluate interpretability, we conducted both qualitative case studies and quantitative human evaluations.

**Qualitative Analysis: Evolution of User Profiling.** We demonstrate the model's ability to capture semantic modalities through the evolution of User Feature Summarization across the three curriculum stages. For clarity, we present only a representative sub-feature, as full outputs are too lengthy.

> ***Stage1:****The user frequently reads news throughout the day and is a news enthusiast.*
> **Stage2:**The user has a strong habit of consuming news, **often checking it multiple times in quick succession**, suggesting **a desire to stay informed** about current events.
> **Stage3:Information-Seeking Behavior:** The user has a strong habit of **consuming news** and **checking the weather**, indicating **a desire to stay informed** about current events and environmental conditions. This behavior is **consistent throughout the week, with slight variations in timing**.

The outputs demonstrate a clear progression. Stage 1 produces simple summaries (e.g., "likes news"). Stage 2 advances to identifying behavioral patterns (e.g., "checking multiple times in rapid succession") and inferring intent. Finally, Stage 3 generates comprehensive features by abstracting actions into high-level categories, grouping behaviors such as "reading news" and "checking the weather" into higher-level "information-seeking" with richer detail, confirming the effectiveness of our design.

Table 5: Average Scores of Human Evaluation

| Type | Rationality | Interpretability |
|---|---|---|
| Human | 2.51 | 2.55 |
| BUA | 2.46 | 2.39 |
| No Tune | 1.90 | 1.83 |

Table 6: Best Score Probability per Sample

| Category | Human | BUA | No Tune |
|---|---|---|---|
| Best in Rationality | 39.8% | 37.9% | 22.3% |
| Best in Interpretability | 45.3% | 33.3% | 21.4% |

**Quantitative Analysis: Human Evaluation.** Following standard protocols (Lei et al., 2024), we assessed feature quality and interpretability using 120 random samples from the Behavior dataset. We compared features from three sources: (1) Human annotators, (2) BUA (Ours), and (3) the Base Model (Qwen2.5-7B w/o fine-tuning). Blinded samples were scored on a 0–3 scale based on two dimensions:**Rationality** (The degree to which the features align with the user's historical behavior) and **Interpretability** (The extent to which the features help explain the user's future behavior).

As shown in Tables 5 and 6, BUA significantly improves upon the Base Model (raising interpretability from 1.83 to 2.39), and achieves near-human level performance in "best score probability". These results demonstrate that BUA attains a deep behavior understanding approaching human levels. For the more detailed experimental setup and analysis, please refer to Appendices P and Q.

**Broader Societal Impacts and Ethical Considerations.** Behavior modeling may raise risks of profiling, behavioral fingerprinting, misuse for surveillance or manipulation, and uneven performance across populations. We discuss these societal and ethical considerations in Appendix W.

## 5 Conclusion

We present BUA, which leverages structured curriculum learning to align behavior and language modalities via sequence embeddings. This approach effectively unifies prediction and generation within a single framework. Experiments confirm that BUA outperforms existing baselines, achieving state-of-the-art results on two datasets.

## Limitations

**Dependence on behavior encoders:** Our proposed method(BUA) uses a pretrained behavior encoder to provide sequence-level behavioral anchors, so its performance may be affected when the encoder has limited coverage in new domains. However, BUA is not tied to a specific encoder: our SASRec-based replacement study shows that the proposed alignment framework remains effective with alternative behavior encoders. Future work will further explore cross-domain adaptation with more general behavior foundation models. And we further discuss practical encoder requirements and generalization directions in Appendix M.

**Joint task optimization** BUA jointly optimizes prediction and generation through multi-round dialogue, but the two tasks may converge at different rates, making unified checkpoint selection challenging. We provide a preliminary mitigation strategy based on adaptive task weighting in Appendix J, which improves the unified checkpoint performance. Nevertheless, fully balancing all sub-metrics across both tasks remains an open direction for future multi-task optimization.

## References


Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 1007–1014.

Charles Chen, Sungchul Kim, Hung Bui, Ryan Rossi, Eunyee Koh, Branislav Kveton, and Razvan Bunescu. 2018. Predictive analysis by leveraging temporal user behavior and user embeddings. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 2175–2182.

Hyunji Chung and Sangjin Lee. 2018. Intelligent virtual assistant knows your life. *arXiv preprint arXiv:1803.00466*.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Jiahui Gong, Jingtao Ding, Fanjin Meng, Guilong Chen, Hong Chen, Shen Zhao, Haisheng Lu, and Yong Li. 2024. A population-to-individual tuning framework for adapting pretrained lm to on-device user intent prediction. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 896–907.

Jiahui Gong, Jingtao Ding, Fanjin Meng, Chen Yang, Hong Chen, Zuojian Wang, Haisheng Lu, and Yong Li. 2025. Behavegpt: A foundation model for large-scale user behavior modeling. *arXiv preprint arXiv:2505.17631*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Guoqing Hu, An Zhang, Shuo Liu, Zhibo Cai, Xun Yang, and Xiang Wang. 2025. Alphafuse: Learn id embeddings for sequential recommendation in null space of language embeddings. *arXiv preprint arXiv:2504.19218*.

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

Joon-Seok Kim, Hamdi Kavak, Umar Manzoor, Andrew Crooks, Dieter Pfoser, Carola Wenk, and Andreas Züfle. 2019. Simulating urban patterns of life: A geo-social data generation framework. In *Proceedings of the 27th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 576–579.

Sein Kim, Hongseok Kang, Seungyoon Choi, Donghyun Kim, Minchul Yang, and Chanyoung Park. 2024. Large language models meet collaborative filtering: an efficient all-round llm-based recommender system. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1395–1406.

Yuxuan Lei, Jianxun Lian, Jing Yao, Xu Huang, Defu Lian, and Xing Xie. 2024. Recexplainer: Aligning large language models for explaining recommendation models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1530–1541.

Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. 2024. Llara: Large language-recommendation assistant. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1785–1795.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning.

Qidong Liu, Xian Wu, Yejing Wang, Zijian Zhang, Feng Tian, Yefeng Zheng, and Xiangyu Zhao. 2024. Llm-esr: Large language models enhancement for long-tailed sequential recommendation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. 2019. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2537–2546.

Yuxuan Lu, Jing Huang, Yan Han, Bennet Bei, Yaochen Xie, Dakuo Wang, Jessie Wang, and Qi He. 2025. Beyond believability: Accurate human behavior simulation with fine-tuned llms. *arXiv preprint arXiv:2503.20749*.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Andreas Züfle Pfoser and Carola Wenk. Towards large-scale agent-based geospatial simulation.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Representation learning with large language models for recommendation. In *Proceedings of the ACM on Web Conference 2024*, pages 3464–3475.

Germans Savcisens, Tina Eliassi-Rad, Lars Kai Hansen, Laust Hvas Mortensen, Lau Lilleholt, Anna Rogers, Ingo Zettler, and Sune Lehmann. 2023. Using sequences of life-events to predict human lives. *Nature Computational Science*, pages 1–14.

Germans Savcisens, Tina Eliassi-Rad, Lars Kai Hansen, Laust Hvas Mortensen, Lau Lilleholt, Anna Rogers, Ingo Zettler, and Sune Lehmann. 2024. Using sequences of life-events to predict human lives. *Nature Computational Science*, 4(1):43–56.

Chenyang Shao, Fengli Xu, Bingbing Fan, Jingtao Ding, Yuan Yuan, Meng Wang, and Yong Li. 2024. Beyond imitation: Generating human mobility from context-aware reasoning with large language models. *arXiv e-prints*, pages arXiv–2402.

Leheng Sheng, An Zhang, Yi Zhang, Yuxin Chen, Xiang Wang, and Tat-Seng Chua. 2024. Language representations can be what recommenders need: Findings and potentials. *arXiv preprint arXiv:2407.05441*.

Jiang-Xin Shi, Tong Wei, Zhi Zhou, Jie-Jing Shao, Xin-Yan Han, and Yu-Feng Li. 2024. Long-tail learning with foundation model: Heavy fine-tuning hurts. In *Forty-first International Conference on Machine Learning*.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450.

Juntao Tan, Shuyuan Xu, Wenyue Hua, Yingqiang Ge, Zelong Li, and Yongfeng Zhang. 2024. Idgenrec: Llm-recsys alignment with textual id learning. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 355–364.

Amrita S Tulshan and Sudhir Namdeorao Dhage. 2019. Survey on virtual assistant: Google assistant, siri, cortana, alexa. In *Advances in Signal Processing and Intelligent Recognition Systems: 4th International Symposium SIRS 2018, Bangalore, India, September 19–22, 2018, Revised Selected Papers 4*, pages 190–201. Springer.

Yiding Wang, Yuxuan Chen, Fangwei Zhong, Long Ma, and Yizhou Wang. 2024. Simulating human-like daily activities with desire-driven autonomy. *arXiv preprint arXiv:2412.06435*.

Yuan Yuan, Huandong Wang, Jingtao Ding, Depeng Jin, and Yong Li. 2023. Learning to simulate daily activities via modeling dynamic human needs. In *Proceedings of the ACM Web Conference 2023*, pages 906–916.

Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Michael He, and 1 others. 2024. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. *arXiv preprint arXiv:2402.17152*.

Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. 2025. Collm: Integrating collaborative embeddings into large language models for recommendation. *IEEE Transactions on Knowledge & Data Engineering*, (01):1–12.

Yu Zhu, Hao Li, Yikang Liao, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. 2017. What to do next: Modeling user behaviors by time-lstm. In *IJCAI*, volume 17, pages 3602–3608. Melbourne, VIC.

## A Summary of all understanding tasks

Summary as shown in Figure 2.

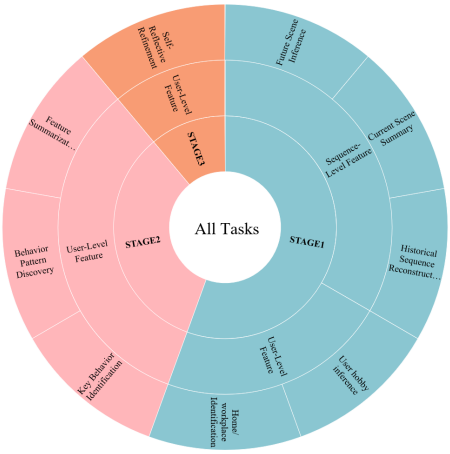


Figure 2: Summary of all understanding tasks

## B Dataset Information

**Behavior dataset**: This proprietary large-scale dataset is derived from mobile phone usage logs. When users interact with their mobile phones, various types of logs are generated, desensitized, and reported with user consent. After desensitizing the original data, we extract 37 daily behaviors that are reliably extracted from raw logs and also cover broad life scenarios, including activities related to learning, work, entertainment, leisure, etc. The dataset spans from March 1, 2024, to April 29, 2024, and consists of over 50 million behavior events from 24,133 anonymous users. We preprocess the dataset and construct samples in the format of $(weekday, timestamp, location, behaviortype)$. Since our target is fine-tuning the LLM instead of training from scratch, we only randomly select a subset (100,000) of data for experiments.

**Tencent dataset**: The Tencent Trajectory dataset, collected from a social network, captures user mobility trajectories. Points of interest (POIs) are manually annotated with detailed intent types. The dataset includes 2,000 users and spans from October 8 to December 31, 2019, comprising 320,516 records. Each record contains an anonymized location ID, the associated intent type, and a timestamp.

**Data reachability.** The Behavior dataset is proprietary and is currently undergoing internal compliance review for potential public release. While the full dataset cannot be redistributed at submission time, we provide several materials to support reproducibility: (1) complete training and inference code, (2) fully desensitized data-format samples that preserve the input structure, (3) detailed descriptions of the data schema and behavior-sequence format, and (4) complete experimental settings and hyperparameters. We also evaluate BUA on public datasets, including the Tencent dataset and the Carat Top1000 Users AppUsage dataset, to provide reproducible benchmarks beyond the proprietary dataset. Links to the public datasets are provided in the published anonymous codebase and below.

**Summary of links to the public datasets.**

- **Tencent Dataset:** Available via the official CoPB GitHub repository at `https://github.com/tsinghua-fib-lab/CoPB`, with a direct download link at `https://pan.baidu.com/s/1K4pUhIiBu7RsAmfoqwF5cg?pwd=821m`. The access password is `821m`.
- **Carat Dataset:** Publicly available as the *Carat Top 1000 Users Long-Term App Usage Dataset* at `https://www.cs.helsinki.fi/group/carat/data-sharing/`.

The characteristics of both datasets are presented in Table 7.

Table 7: Statistics of the datasets

| Dataset | # Users | # Behav. Type | # Logs |
|---|---|---|---|
| Behavior Dataset | 24,133 | 37 | 100,000 |
| Tencent Dataset | 2,000 | 14 | 320,516 |

**Data Privacy and Anonymization Protocols**

To ensure the ethical handling of user data and strict adherence to privacy standards, we have implemented the following protection measures:

- **Anonymization Procedures**: We employ a rigorous ID anonymization mechanism. Each user is assigned a randomized anonymous identifier, which is **periodically refreshed** (e.g., every three months) to prevent long-term user tracking and linkage attacks.
- **Location Data Obfuscation**: We strictly avoid collecting precise geospatial coordinates. Instead, we utilize a rule-based aggregation algorithm prior to data upload. This algorithm identifies the user's top-10 frequent activity zones and maps them to **low-resolution area IDs**. Consequently, only these abstract

representations—rather than raw latitude and longitude data—are stored, effectively preserving location privacy.

- **Long-Term Privacy Assurance**: Collectively, these measures render it computationally infeasible to re-identify specific individuals through longitudinal behavioral analysis or location inference.

# C Details of Used Metrics for Prediction

## C.1 Behavior category classification

For *long-tailed learning*, following the settings of relevant work (Liu et al., 2019; Shi et al., 2024), we evaluate four accuracy metrics based on behavior occurrence frequencies: category-average accuracy across all behaviors ($Overall$), head-category accuracy for behaviors with $> 5.0\%$ frequency ($Head$), medium-category accuracy for those between 1.0% and 5.0% frequency ($Medium$), and tail-category accuracy for the remaining low-frequency behaviors ($Tail$).

## C.2 Behavior prediction Metrics

The formula for $Prec_w$ :

$$Prec_w = \frac{\sum_{c \in C}(\mathrm{TP}_c + \mathrm{FP}_c) \cdot \mathrm{Precision}_c}{\sum_{c \in C}(\mathrm{TP}_c + \mathrm{FP}_c)} \quad (6)$$

The formula for $Rec_w$ :

$$Rec_w = \frac{\sum_{c \in C}(\mathrm{TP}_c + \mathrm{FN}_c) \cdot \mathrm{Recall}_c}{\sum_{c \in C}(\mathrm{TP}_c + \mathrm{FN}_c)} \quad (7)$$

The formula for $Overall$ :

$$Accuracy = \frac{1}{|C|} \sum_{c \in C} \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FN}_c} \quad (8)$$

The formula for $Head$ :

$$Accuracy = \frac{1}{|C_h|} \sum_{c \in C_h} \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FN}_c} \quad (9)$$

The formula for $Medium$ :

$$Accuracy = \frac{1}{|C_m|} \sum_{c \in C_m} \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FN}_c} \quad (10)$$

The formula for $Tail$ :

$$Accuracy = \frac{1}{|C_t|} \sum_{c \in C_t} \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FN}_c} \quad (11)$$

Where $|C|$ represents the total number of classes, $|C_h|$ represents the total number of classes belonging to the head category, Where $|C_m|$ represents the total number of classes belonging to the medium category, Where $|C_h|$ represents the total number of classes belonging to the tail category. True Positives ($TP_c$) denotes the number of samples correctly classified as class $c$, False Positives ($FP_c$) represents the number of samples incorrectly classified as class $c$, and False Negatives ($FN_c$) stands for the number of samples incorrectly classified as other classes instead of class $c$. And $\mathrm{Precision}_c$ and $\mathrm{Recall}_c$ respectively refer to the precision and recall of class $c$.

# D Details of Used Metrics for Generation

The formula for $BLEU$:

$$\mathrm{BLEU} = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \quad (12)$$

Where $\mathrm{BP} = \min\left(1, e^{1-r/c}\right)$ is the brevity penalty, $p_n$ is the modified $n$-gram precision, $r$ is the reference length, and $c$ is the candidate length.

The formula for $TVD$:

$$\mathrm{TVD}(P, Q) = \frac{1}{2} \sum_{i=1}^{k} |P(i) - Q(i)| \quad (13)$$

Where $P$ and $Q$ are probability distributions over $k$ classes, $P(i)$ denotes the predicted probability of class $i$, $Q(i)$ denotes the ground truth probability.

The formula for $JSD$:

$$\mathrm{JSD}(P \| Q) = \sqrt{\frac{1}{2} D_{\mathrm{KL}}(P \| M) + \frac{1}{2} D_{\mathrm{KL}}(Q \| M)} \quad (14)$$

Where $M = \frac{1}{2}(P+Q)$ is the midpoint distribution, and $D_{\mathrm{KL}}$ denotes the Kullback-Leibler divergence:

$$D_{\mathrm{KL}}(P \| Q) = \sum_{i=1}^{k} P(i) \log \frac{P(i)}{Q(i)} \quad (15)$$

# E Details of Baselines

**SASRec (Kang and McAuley, 2018).** uses self-attention mechanisms to model user behavior sequences. It captures both short-term and long-term dependencies in sequential data, allowing it to focus on the most relevant items in the user's interaction history for recommendation.

**BehaveGPT (Gong et al., 2025)** is a transformer-based model pre-trained specifically on user behavior data. Its novel pre-training paradigm enables it to learn complex behavior patterns and support various downstream tasks,

including next behavior prediction, long-term generation, and cross-domain adaptation.

**PITuning (Gong et al., 2024)** loads pre-trained Large Language Model (LLM) parameters to acquire textual knowledge and then designs an adaptive unlearning strategy to address the long-tail preference issue, achieving excellent performance in user behavior prediction.

**AlphaFuse (Hu et al., 2025)** is a simple yet effective language-guided learning strategy that addresses long-tail intent modeling by learning ID embeddings within the null space of language embeddings.

**TALLRec (Bao et al., 2023)** is one of the earlier methods to integrate Large Language Models (LLMs) with the recommendation domain. It employs a two-stage tuning process—Alpaca Tuning and Rec-Tuning—to finetune LLMs for recommendations, enabling effective and efficient adaptation of LLMs with only a small number of tuning samples.

**A-LLMRec (Kim et al., 2024)** bridges the knowledge between the language and recommendation domains by training an alignment network with a variety of tasks, targeting both warm and cold-start scenarios.

**CoLLM (Zhang et al., 2025)** captures collaboration information using external traditional models and maps it into the LLM's input embedding space as collaboration embeddings. This external integration allows effective modeling of collaboration without modifying the LLM, enabling flexible use of various collaboration modeling techniques.

**LLaRa (Liao et al., 2024)** introduces a hybrid prompting method that integrates both world knowledge and behavioral patterns into item representations. It conducts curriculum prompt tuning to achieve modality alignment.

For comparison, we also consider LLMs that are not fine-tuned on behavioral data, i.e., Deepseek-V3 (DeepSeek-AI et al., 2025), which is a powerful Mixture-of-Experts (MoE) language model with 671B total parameters and 37B activated per token, offering performance comparable to GPT-4 (OpenAI et al., 2024) at a lower cost.

# F Implementation Details

The hardware used in this experiment consists of 8 NVIDIA A100 40G GPUs. We selected Qwen2.5-7B (Qwen et al., 2025) as the backbone for our experiments. Our experiments utilized the AdamW optimizer with a cosine annealing learning rate schedule, setting the warm-up proportion to 0.03. The maximum learning rate for cosine annealing was set to 5e-5, while both the minimum and initial warm-up learning rates were set to 1e-6. We conducted LoRA (Hu et al., 2022) fine-tuning and parallel training acceleration. All experiments were performed with a maximum of 3 training epochs and a batch size of 96, selecting the best-performing model on the validation set for testing. Our experiments are typically completed within 8 hours. To assess statistical reliability, we additionally repeat the main behavior prediction experiment on the Behavior dataset with five different random seeds. Detailed seed settings and the corresponding mean and standard deviation results are provided in Appendix U.

# G Details of Practical Applications

## G.1 Data Generation Process

For the behavior dataset, we use our model and baselines to generate one day of user behavior data based on a history sequence of 100 behaviors (spanning over one day). From the generated output, we take the most recent 41 behaviors and use the first 40 to predict the final one.

## G.2 Downstream Task Experimental Settings

To evaluate the utility of the generated synthetic data, we employed SASRec as the downstream behavior prediction model. To ensure a fair comparison across datasets of varying sizes (e.g., real data vs. real + synthetic data) and to address concerns regarding gradient steps, we adopted a "train to convergence" strategy. Instead of fixing the total number of gradient steps, we utilized **Early Stopping** with a patience of 5 epochs (monitoring validation loss). This approach ensures that all models, regardless of the training data volume, are trained to their maximum potential without overfitting or underfitting. The specific hyperparameters used for the downstream SASRec model are consistent with standard settings and are detailed as follows:

- **Model Architecture**:
  - Hidden Units: 50
  - Number of Blocks (Layers): 2
  - Number of Attention Heads: 1
  - Dropout Rate: 0.1
  - Max Sequence Length: 40

- **Optimization**:
  - Optimizer: Adam
  - Learning Rate: 0.001
  - Batch Size: 16
  - L2 Embedding Regularization (`l2_emb`): 0.01
- **Training Config**:
  - Maximum Epochs: 200
  - Early Stopping Patience: 5 epochs

# H Ablation Study for Behavior Generation Task

The ablation results on the generation task are shown in the following Table 8.

# I Analysis of Error Sources in Behavior Prediction

We additionally conducted error analysis experiments to better analyze the sources of error in the behavior prediction task. Specifically, for the three progressive subtasks in the second stage, we replaced the model-generated outputs with ground-truth values from their respective supervised training tasks. The experimental setup includes four groups, as shown below(Note that understanding task1, task2, and task3 in the table represent the User Key Behavior Identification, User Behavior Pattern, and User Feature Summarization Discovery tasks, respectively.). In the table, "pred" indicates that the corresponding feature uses the model's own generated result (which may contain errors), while "label" denotes the use of the ground-truth value from the supervised training tasks.

Under these four experimental settings, we analyzed changes in the accuracy on the long-tail test set. The results are summarized in the table below. In the table, $r2w$ represents the percentage of data that changed from correct to incorrect predictions compared to the previous row's settings, while $w2r$ represents the opposite, and *Difference* indicates the net accuracy improvement (the difference between $w2r$ and $r2w$).

These results indicate that early errors in User Key Behavior Identification can propagate through subsequent pattern discovery and feature summarization. In BUA, Stage 3 self-reflective refinement partially mitigates this issue by prompting the model to review and correct inconsistent or weakly supported user features. However, this is mainly a downstream correction strategy rather than a root-cause solution. Future work may further reduce error propagation by improving the robustness of key-behavior identification, introducing confidence-aware intermediate reasoning, or allowing the model to revise earlier Stage 2 outputs during multi-round dialogue.

# J Joint Optimization

**Joint Optimization** – Consider using adaptive learning rate schedules to resolve convergence mismatch between prediction and generation tasks

We implemented an **adaptive learning rate schedule** by dynamically adjusting the task loss weights based on the ratio of current loss to initial loss. This effectively assigns a higher weight to the prediction task and a lower weight to the generation task, accelerating convergence of the former while slowing down the latter. Below is a detailed description of the strategy:

## J.1 Dynamic Task Weighting Strategy

Let:

- $L_i^{(0)}$: the initial loss of task $i$
- $\hat{L}_i$: the current exponentially moving averaged (EMA) loss of task $i$
- $r_i = \frac{\hat{L}_i}{L_i^{(0)}}$: the loss ratio of task $i$
- $\bar{r} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} r_i$: the average loss ratio across **valid tasks**
- $s_i = \frac{\bar{r}}{r_i}$: the **relative learning speed** of task $i$ (slower tasks will have larger values)
- $\alpha$: a tunable exponent to control the sensitivity of the weighting

The normalized task weight $w_i$ is computed as:

$$w_i = \begin{cases} \frac{s_i^\alpha}{\sum_{j \in \mathcal{V}} s_j^\alpha} \cdot |\mathcal{V}|, & \text{if } L_i^{(0)} > 0 \\ 1, & \text{otherwise} \end{cases}$$

Where:

$$\mathcal{V} = \left\{ i \mid L_i^{(0)} > 0 \right\}$$

is the set of valid tasks (i.e., those with positive initial loss values).

After applying this method, the step corresponding to the lowest total loss shifted from **2600 to 3200**, with corresponding unweighted prediction

Table 8: Ablation Study for Behavior Generation Task

| Method | Event | | | Timestamp | | | Location | | |
|---|---|---|---|---|---|---|---|---|---|
| | Bleu ↑ | TVD ↓ | JSD ↓ | Bleu ↑ | TVD ↓ | JSD ↓ | Bleu ↑ | TVD ↓ | JSD ↓ |
| Ours | 0.354 | 0.140 | 0.020 | 0.541 | 0.147 | 0.020 | 0.711 | 0.065 | 0.005 |
| w/o stage1 | 0.309 | 0.167 | 0.028 | 0.500 | 0.162 | 0.024 | 0.640 | 0.093 | 0.007 |
| w/o stage2 | 0.304 | 0.189 | 0.029 | 0.580 | 0.095 | 0.008 | 0.745 | 0.064 | 0.006 |
| w/o stage3 | 0.343 | 0.146 | 0.022 | 0.523 | 0.156 | 0.025 | 0.708 | 0.079 | 0.008 |

Table 9: Experimental Setup for Error Analysis

| ID | *understanding task1* | *understanding task2* | *understanding task3* |
|---|---|---|---|
| 1 | pred | pred | pred |
| 2 | label | pred | pred |
| 3 | label | label | pred |
| 4 | label | label | label |

Table 10: Error Analysis Results on Long-Tail Test Set

| ID | $Overall$ | $r2w$ | $w2r$ | *Difference* |
|---|---|---|---|---|
| 1 | 0.336 | - | - | - |
| 2 | 0.417 | 6.3% | 14.4% | 8.1% |
| 3 | 0.432 | 2.7% | 4.2% | 1.5% |
| 4 | 0.480 | 1.8% | 6.6% | 4.8% |

and generation losses improving slightly to **0.2674** and **0.271** (from **0.2690** and **0.2726**). These data show that this method does make the convergence speed of prediction and generation tasks more matched.

The table below shows performance comparisons, where "No optimization" refers to results without multi-task optimization (as in the paper)

### J.1.1 Prediction Task

Table 11: Performance of the Prediction Task Under Different Optimization Methods

| Optimization method | $Prec_w$ | $Rec_w$ | $Overall$ | $Head$ | $Medium$ | $Tail$ |
|---|---|---|---|---|---|---|
| No optimization | 0.644 | 0.642 | 0.471 | 0.538 | 0.489 | 0.446 |
| Adaptive learning rate | 0.638 | 0.648 | 0.479 | 0.528 | 0.497 | 0.452 |

### J.1.2 Generation Task

As shown, while some metrics improved, results are not consistently better across all tasks. This suggests that **multi-task optimization requires more sophisticated strategies**, and we plan to explore further methods (e.g., separate optimizers or gradient balancing techniques) in future work. Although this strategy does not simultaneously maximize every sub-metric, it provides a practical way to select a more balanced unified checkpoint.

## K Efficiency Comparison

**Efficiency Comparison** – Compare inference time and memory usage with baseline models

Regarding computational cost during inference:

- **Hardware:** All inference tests were conducted on NVIDIA A100 (40GB).
- **Inference Time (on 20,000 samples from the Honor dataset):**
  - $<$ 3 *minutes:* SASRec, BehaveGPT, PITuning, AlphaFuse
  - ~25 *minutes:* TALLRec, A-LLMRec, CoLLM, LLaRa
  - ~40 *minutes:* BUA
  - *Not available:* DeepSeek (API-based)
- **Memory Usage:**
  - *Low ($<$ 2GB):* SASRec (~1GB), BehaveGPT, PITuning, AlphaFuse (~2GB)
  - *High (~30˘32GB):* TALLRec, A-LLMRec, CoLLM, LLaRa, BUA

BUA's inference efficiency is comparable to other LLM-based baselines, though higher than traditional methods—reflecting a broader trend in LLM-based approaches. We anticipate continued advances in LLM optimization that will help narrow this efficiency gap in the near future.

## L Cross-Cultural Contexts Evaluation

**Cross-Cultural Contexts Evaluation** – Test the model on datasets from different domains or cultural backgrounds

To address this, we incorporated a new dataset: the **Carat Top 1000 Users App Usage Dataset**, which collects app usage and battery data from volunteers across multiple countries, including the U.S., Japan, and the U.K., and notably excludes China. This provides a complementary perspective to the Honor and Tencent datasets used in our original submission.

Table 12: Performance of the Generation Task Under Different Optimization Methods

| Method | Event | | | Timestamp | | | Location | | |
|---|---|---|---|---|---|---|---|---|---|
| | Bleu ↑ | TVD ↓ | JSD ↓ | Bleu ↑ | TVD ↓ | JSD ↓ | Bleu ↑ | TVD ↓ | JSD ↓ |
| None | 0.354 | 0.140 | 0.020 | 0.541 | 0.147 | 0.020 | 0.711 | 0.065 | 0.005 |
| Adaptive learning rate | 0.363 | 0.141 | 0.020 | 0.553 | 0.146 | 0.019 | 0.708 | 0.079 | 0.008 |

We compared our method (BUA) with the best-performing baselines from each category on this dataset. The results are shown in Table 13.

Table 13: Performance on Carat Top 1000 Users App Usage Dataset

| Method | $Prec_w$ | $Rec_w$ | $Overall$ | $Head$ | $Medium$ | $Tail$ |
|---|---|---|---|---|---|---|
| BehaveGPT | 0.299 | 0.318 | 0.210 | 0.303 | 0.219 | 0.052 |
| PITuning | 0.352 | 0.356 | 0.357 | 0.362 | <u>0.425</u> | 0.152 |
| CoLLM | <u>0.400</u> | <u>0.365</u> | <u>0.367</u> | <u>0.377</u> | 0.418 | <u>0.233</u> |
| Ours (BUA) | **0.447** | **0.418** | **0.400** | **0.409** | **0.451** | **0.267** |

As shown, our method continues to achieve strong performance on a dataset with a markedly different demographic and geographical distribution, further validating its generalizability.

## M Pretrained Base Model Replacement

**Pretrained Base Model Replacement** – Evaluate the effect of replacing the current pre-trained base model

We have replaced BehaveGPT with SASRec as the pretrained behavior sequence encoder. The performance is shown in Table 14.

Table 14: Performance Comparison of Different Pretrained Base Models

| Pretrained Model | $Prec_w$ | $Rec_w$ | $Overall$ | $Head$ | $Medium$ | $Tail$ |
|---|---|---|---|---|---|---|
| SASRec | 0.561 | 0.589 | 0.331 | 0.466 | 0.389 | 0.247 |
| BehaveGPT | 0.644 | 0.642 | 0.471 | 0.538 | 0.489 | 0.446 |

While SASRec underperforms compared to BehaveGPT, our method still achieves notable gains over SASRec alone, demonstrating its effectiveness. BUA requires the encoder to preserve temporal order and distinguish major behavior patterns at the sequence level, but it does not require a large pretrained foundation model. The SASRec replacement study shows that a simpler sequential encoder can also serve as the behavioral anchor, although stronger encoders may provide more informative representations.

## N Self-Reflection Details

**Self-Reflection Details** – Provide a more detailed explanation of the self-reflection optimization method.

To clarify, the model identifies recurring shortcomings in initial profiles through prompt-guided reflection, focusing on issues such as:

1. insufficient abstract summarization,
2. inadequate detail association and reasoning,
3. poor structural clarity,
4. weak information hierarchy,
5. inaccurate temporal pattern analysis, and
6. lack of personalized expression.

The corresponding correction criteria are designed as follows:

- **For abstract summarization**: Elevate surface-level behaviors to infer deeper cognitive traits (e.g., deducing "information-driven lifestyle" from frequent news consumption).
- **For temporal analysis**: Calibrate behavior frequencies and highlight periodic patterns.
- **For structure**: Implement a three-layer hierarchy—from cognitive-level traits to habit interactions and specific behavioral anchors.
- **For personalization**: Emphasize distinctive, user-specific behavioral descriptors while avoiding vague generalities.

Importantly, this self-reflective process is not limited to output refinement. As described above, feedback from these reflections is also used to update model parameters via supervised fine-tuning, leading to further performance improvements.

## O Data Granularity

**Data Granularity** – Clarify what is meant by "high-level daily activities" and how they are represented in the data.

To clarify the granularity of "Behavior Type ID," we define it at the level of high-level daily activities—neither raw sensor signals nor overly abstract categories. Below is a simplified example of a typical user's day to illustrate the scope:

- **Morning**: Alarm clock, check weather
- **Commute (to work)**: Subway, watch news, payment
- **Work hours**: Editing video, online meeting
- **Lunch break**: Ordering takeout, watching video
- **Commute (to home)**: Subway, watching video, payment
- **Evening**: Online shopping, gaming, watching video

Due to space limitations, this example condenses some activity details, but it reflects the typical granularity used across different scenarios.

## P Human Evaluation

**Human Evaluation** – Human evaluation experiments on the quality and interpretability of model-generated profile features

We have conducted a human evaluation study to more systematically assess the interpretability of the generated portrait features.

Following the methodology of prior work [1], we randomly selected 120 test samples from the Honor dataset. For each sample, portrait features were generated by three sources: (1) human annotators, (2) our proposed model (BUA), and (3) the base model (Qwen2.5-7B without fine-tuning). This resulted in a total of 360 portrait feature samples, which were evaluated by a separate group of human judges using consistent evaluation criteria. The evaluation focused on two dimensions:

- **Rationality**: the degree to which the portrait features align with the user's historical behavior
- **Interpretability**: the extent to which the portrait features help explain the user's predicted future behavior

Scores ranged from 0 to 3, with higher scores indicating better performance. To avoid bias, the order and source of the portrait features were anonymized and randomly shuffled for each evaluation instance. (Note: If multiple sources achieve the highest score for a sample, the credit is divided equally among them.)

Table 15: Average Scores of Human Evaluation

| Type | Rationality | Interpretability |
|---|---|---|
| Human | 2.51 | 2.55 |
| BUA | 2.46 | 2.39 |
| No Tune | 1.90 | 1.83 |

Table 16: Best Score Probability per Sample

| Category | Human | BUA | No Tune |
|---|---|---|---|
| Best in Rationality | 39.8% | 37.9% | 22.3% |
| Best in Interpretability | 45.3% | 33.3% | 21.4% |

These results indicate that while human-written features still achieve the highest overall performance, our fine-tuned model (BUA) significantly outperforms the base model in both rationality and interpretability. Moreover, BUA's performance approaches that of human-written features, demonstrating meaningful gains in interpretability.

## Q Questionnaire Setup Details

**Task Description:**

Your task is to evaluate the quality of user features based on two dimensions

**Dimension 1 (Rationality):**

Does the user feature accurately reflect the user's historical behavior sequence?

**Dimension 2 (Interpretability):**

Can the user feature help explain the predicted next behavior of the user? Does it provide a reasonable basis for why the predicted behavior might occur?

---

**Scoring Criteria for Dimension 1 (Rationality: 0–3 points)**

**Scoring Criteria for Dimension 2 (Interpretability: 0–3 points)**

## R Empirical Analysis of Curriculum Learning Strategy

To empirically validate the necessity and effectiveness of our proposed three-stage curriculum

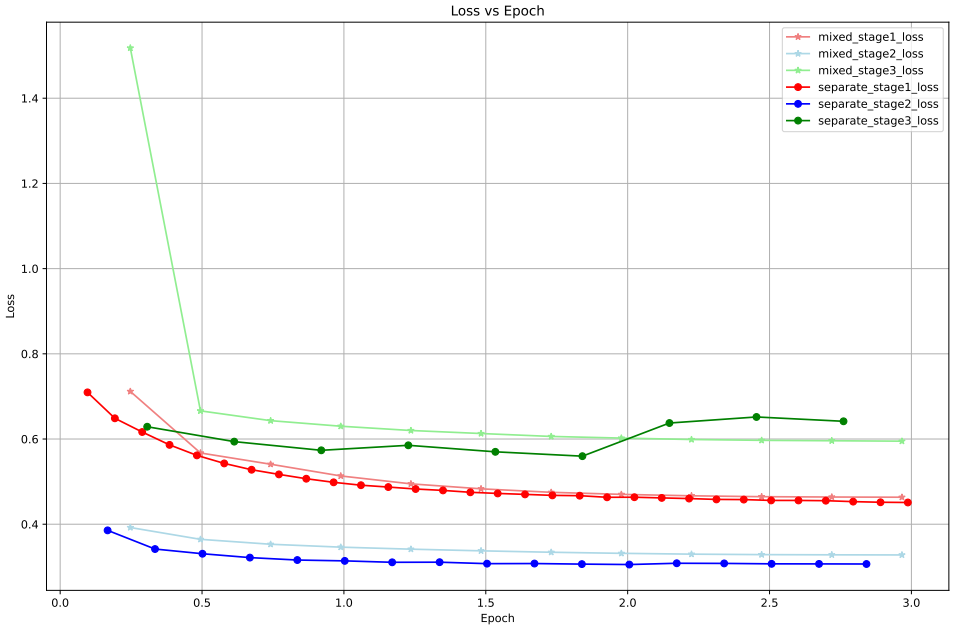


Figure 3: Validation loss comparison between our Staged Curriculum (Separate) and Joint Training (Mixed) strategies. The solid lines represent our Staged Training, while the faded lines with stars represent Joint Training.

| Score | Description |
|---|---|
| 0 | **No Match**: The profile feature is not reflected at all in the user's behavior sequence. |
| 1 | **Weak Match**: The profile feature is only partially reflected in the behavior sequence. |
| 2 | **Basic Match**: The feature is generally reflected in the behavior sequence but is overly broad (e.g., "user likes playing games"). |
| 3 | **Strong Match**: The feature is clearly and specifically reflected in the behavior sequence (e.g., "user likes playing games on Friday nights after watching short videos"). |

| Score | Description |
|---|---|
| 0 | **No Match**: The profile feature is completely unrelated to the predicted user behavior. |
| 1 | **Weak Match**: The feature can be loosely connected to the predicted behavior (e.g., "user often engages in leisure activities" → predicted behavior: "playing games"). |
| 2 | **Basic Match**: The feature aligns with the predicted behavior but is too general (e.g., "user likes playing games" → predicted behavior: "playing games"). |
| 3 | **Strong Match**: The feature directly and specifically supports the predicted behavior (e.g., "user likes playing games on Friday nights after reading the news" → predicted behavior: "playing games"; it is Friday night and the user has just read the news). |

design, we conducted a comprehensive ablation study comparing the convergence dynamics of our approach against a standard multi-task learning baseline. Specifically, we contrasted our proposed **Staged Training** strategy, where the model is optimized sequentially through Sequence-Level Understanding (Stage 1), User-Level Feature Modeling (Stage 2), and Self-Reflective Refinement (Stage 3), against a **Joint Training** baseline. In the Joint setting, the model is trained simultaneously on all tasks across the three stages from scratch, disregarding the hierarchical dependencies inherent in behavioral understanding. This comparison aims to verify whether the structured, easy-to-hard progression provides tangible optimization benefits over simple joint optimization.

The validation loss curves for each stage's specific tasks under both settings are presented in Fig-

ure 3. The comparative analysis reveals three critical insights regarding the training dynamics:

- **Comparable Performance on Baselines (Stage 1):** For the most fundamental task, Sequence-Level Understanding, the loss curves for both Separate (Red solid line) and Mixed (Red faded line) settings are relatively close. This indicates that simple semantic alignment is less sensitive to the training strategy and can be adequately learned via joint optimization.

- **Superiority in User-Level Modeling (Stage 2):** A significant divergence appears in the more difficult Stage 2 tasks. The Separate Training (Blue solid line) achieves a consistently lower minimum validation loss compared to the Mixed setting (Blue faded line). This confirms that a solid foundation in Stage 1 is essential for mastering complex user features, as the model benefits from pre-aligned semantic representations.

- **Cold-Start Challenge in Self-Reflection (Stage 3):** Notably, the Mixed_Stage3_Loss (Green faded line) starts at an extremely high value ($> 1.5$), indicating that the model struggles to perform self-reflective refinement without a pre-established user profile context. In contrast, the Staged approach (Green solid line) allows the model to tackle Stage 3 with initialized understanding, resulting in a smoother optimization landscape and lower final loss.

These empirical results strongly corroborate the theoretical foundation of our curriculum design, rooted in cognitive scaffolding and curriculum learning (Bengio et al., 2009). The convergence patterns demonstrate that while joint training is sufficient for aligning basic semantic representations, it struggles with higher-order reasoning tasks without established prerequisites. By enforcing a structured, easy-to-hard learning progression, our staged approach ensures that the model acquires a robust understanding of fundamental behavioral semantics before tackling complex user profiling and self-reflection, achieving superior performance.

## S Use of LLMs

We used LLMs to assist in writing the paper, such as identifying typos and correcting grammatical errors, as well as polishing some paragraphs.

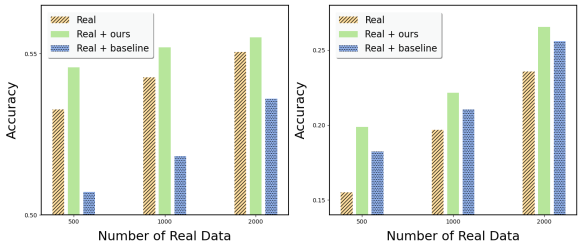


Figure 4: hybrid scenario

| Method | $Rec_w$ | $Prec_w$ | Overall | Head | Medium | Tail |
|---|---|---|---|---|---|---|
| BUA | 0.643±0.004 | 0.642±0.002 | 0.473±0.004 | 0.537±0.003 | 0.488±0.004 | 0.448±0.004 |

Table 17: Multi-seed results of BUA on the Behavior dataset for behavior prediction.

## T Synthetic Data for Downstream Prediction Task

To further assess the usability of the generated data, following (Yuan et al., 2023), we evaluated our model in a hybrid setting that augments real data with synthetic data. Using the standard SASRec model for next-behavior prediction (see Appendix G). As shown in Figure 4, our generated data consistently outperforms the strongest baseline, D2A, in both overall and average accuracy, significantly boosting model performance. This confirms our model's ability to generate high-fidelity behavior sequences that capture underlying user patterns.

## U Multi-seed Robustness Analysis

To evaluate the statistical stability of BUA, we repeat the main behavior prediction experiment on the Behavior dataset with five random seeds: 42, 2024, 2025, 2026, and 999. Table 17 reports the mean and standard deviation.

The standard deviations are consistently small across all metrics, indicating that BUA's performance is stable across different random seeds. The variance is also much smaller than the main performance gaps over strong baselines, supporting the reliability of the reported improvements.

## V Discussion on Population-Level Behavior Modeling

BUA focuses on extracting user-specific behavioral patterns from individual behavior histories. Population-level behavioral regularities are currently captured only implicitly through the pretrained behavior encoder, which may encode shared behavioral rhythms from large-scale pretraining

data. However, explicitly modeling population-level patterns is a promising direction for future work.

One possible extension is to introduce population-level prototypes that represent common behavioral clusters, such as commuting routines, workday patterns, or weekend activities. User-specific sequence representations could then be compared with these prototypes through contrastive learning or hierarchical modeling. Such mechanisms may improve cold-start prediction for users with limited histories, support anomaly detection by distinguishing rare population behaviors from user-specific preferences, and enhance interpretability by contextualizing individual patterns against population-level norms.

We leave the explicit trade-off between population-level generalization and user-level personalization as future work.

## W Broader Societal Impacts and Ethical Considerations

Human behavior modeling can support user-beneficial applications such as personal assistants, context-aware services, and personalized recommendation systems. However, because BUA models longitudinal behavioral traces and supports both future behavior prediction and generation, it also raises important societal and ethical concerns. We discuss these risks and responsible deployment considerations below.

**Privacy, consent, and behavioral fingerprinting.** Although our data processing follows anonymization and location obfuscation protocols, technical anonymization alone may not fully eliminate privacy risks. Long-term behavioral sequences can reveal stable routines, preferences, and activity patterns, which may serve as behavioral fingerprints and create potential re-identification risks even without explicit personal identifiers. Moreover, models may infer high-level user characteristics beyond directly observed actions. Therefore, behavior modeling systems should rely on explicit, informed, and granular user consent. Users should be clearly informed about what types of behavioral patterns may be inferred, how long the data will be retained, and how they can opt out or request deletion of their data and inferred profiles.

**Dual-use risks and user autonomy.** The ability to predict and generate future behaviors can be beneficial when used to assist users, for example by supporting scheduling, context-aware reminders, or personalized services under user control. However, the same capability could also be misused for intrusive profiling, surveillance, manipulative personalization, or behavioral nudging. These risks are especially concerning in settings where data collectors have greater power than users, such as large platforms, workplaces, or commercial service environments. We therefore view BUA as appropriate only for voluntary, consent-based, and user-beneficial scenarios. It should not be used for non-consensual monitoring, employee surveillance, or targeting vulnerable users based on inferred behavioral states.

**Responsible deployment guidelines.** Practical deployment of behavior modeling systems should preserve user autonomy and agency. We recommend several safeguards: purpose limitation, data minimization, transparent disclosure of inferred behavioral profiles, user-accessible controls for reviewing or deleting inferred patterns, opt-out mechanisms, strict access control, audit logs, and human oversight in sensitive use cases. Where feasible, on-device or hybrid inference should be considered to reduce unnecessary transmission of sensitive behavioral data or embeddings. For high-stakes applications, additional ethical review and external auditing should be required before deployment.

**Fairness and cultural generalizability.** Behavioral routines can vary substantially across cultures, regions, occupations, age groups, and socioeconomic contexts. Although our additional evaluation on the Carat dataset provides evidence of generalization beyond a single data source, it does not guarantee fairness across all demographic or cultural groups. A model trained or evaluated on one population may underperform on another, or may incorrectly treat population-specific routines as universal behavioral norms. Responsible deployment should therefore include subgroup-level evaluation, domain-shift monitoring, and fairness auditing, especially when the system is applied to new populations or behavioral domains.

**Environmental and deployment considerations.** BUA relies on an LLM and a behavior encoder, and can therefore be more computationally demanding than lightweight sequential models. We do not suggest that LLM-based behavior modeling should replace simpler models in all scenarios. Instead,

the choice of model should consider the trade-off among predictive performance, interpretability, generation capability, privacy requirements, and computational sustainability. Our implementation freezes the behavior encoder and uses parameter-efficient tuning to reduce training cost, but future work should further explore smaller backbones, model compression, early-exit inference, and on-device or hybrid inference where appropriate.

Overall, detailed behavior modeling should be developed and deployed with caution. While BUA aims to improve interpretable behavior understanding for beneficial applications, its use should be constrained by transparency, consent, user control, fairness evaluation, and application-specific risk assessment.