Beyond Static Profiles: Intent-Aware Behavioural Control for LLM-based Recommendation Simulation

Xinye Wanyan
xinye.wanyan@student.rmit.edu.au
RMIT University
Melbourne, VIC, Australia

Chenglong Ma
chenglong.ma@rmit.edu.au
RMIT University
Melbourne, VIC, Australia

Danula Hettiachchi
danula.hettiachchi@rmit.edu.au
RMIT University
Melbourne, VIC, Australia

Ziqi Xu
ziqi.xu@rmit.edu.au
RMIT University
Melbourne, VIC, Australia

Jeffrey Chan
jeffrey.chan@rmit.edu.au
RMIT University
Melbourne, VIC, Australia

Abstract

Large language models (LLMs) are increasingly used in recommender systems for user modelling, ranking, and simulation. Among these applications, LLM-based user simulators provide a flexible way to evaluate recommender systems without relying on live user feedback. However, existing simulators typically reduce user behaviour to static preference matching based on fixed user profiles, overlooking the fact that real decisions may also reflect recent preference shifts and exploratory behaviour. To address this limitation, we propose **IntRecSim** (**Int**ent-aware **Rec**ommendation **Sim**ulation), a plug-in behavioural control framework for LLM-based user simulation. IntRecSim introduces a lightweight intent mechanism that guides whether the simulated decision should follow long-term preferences, recent behavioural evidence, or exploratory behaviour. Since standard recommendation datasets lack explicit intent annotations, we construct pseudo-intent supervision from historical interactions using hybrid category and semantic preference signals. The inferred intent is injected into the LLM simulator as a dynamic control signal, making the framework compatible with different profile-generation backbones. Experiments on three benchmark datasets, covering discrimination and ranking tasks show that IntRecSim improves simulation performance in most evaluated settings. Further analysis highlights the importance of explicit behavioural control for modelling realistic user decisions in LLM-based recommender simulation. [1]

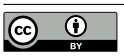
CCS Concepts

• **Information systems → Recommender systems**; • **Computing methodologies → Interactive simulation**; **Agent / discrete models**.

Keywords

Recommender System, Generative AI, User Behaviour Simulation

[1]The source code is available at https://github.com/WennyXY/IntRecSim.



This work is licensed under a Creative Commons Attribution 4.0 International License.
CIKM '26, Rome, Italy
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2539-5/2026/11
https://doi.org/10.1145/3799682.3840925

ACM Reference Format:
Xinye Wanyan, Chenglong Ma, Danula Hettiachchi, Ziqi Xu, and Jeffrey Chan. 2026. Beyond Static Profiles: Intent-Aware Behavioural Control for LLM-based Recommendation Simulation. In *Proceedings of the 35th ACM International Conference on Information and Knowledge Management (CIKM '26), November 07–11, 2026, Rome, Italy*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3799682.3840925

1 Introduction

LLMs have shown strong potential in recommender systems, extending their role beyond ranking and content generation to user behaviour simulation for offline evaluation [12, 13, 19, 21, 26, 29, 30]. Conventional recommender system evaluation mainly relies on historical interaction logs or online A/B testing [8]. However, logged data is static, incomplete, and often affected by selection bias, while online experiments are costly, time-consuming, and may expose users to suboptimal recommendations [6, 7, 35]. Recent studies have explored LLM-based user simulators as controllable and low-risk environments for recommender system evaluation [3, 4, 14, 20, 31, 38, 39]. In these frameworks, users are typically instantiated as agents equipped with profile, memory, and action modules, enabling sequential behaviours such as browsing, clicking, and rating. More broadly, LLM agents have also been studied as human-centred recommendation assistants and user-side intermediaries [18, 27, 40]. These studies demonstrate that LLM agents can capture richer user-side behaviour, but how to explicitly control the decision mode of simulated users remains underexplored.

Despite this progress, existing LLM-based simulators often reduce user behaviour to preference matching, where an agent selects items that best align with a static or loosely updated representation of historical preferences [25, 37]. While useful, such profiles are insufficient for modelling dynamic user decisions. Real users do not always choose items that best match their long-term interests; their decisions may also reflect recent behavioural shifts, temporary needs, or exploratory tendencies [2, 36]. As a result, simulators relying mainly on static profiles may fail to capture how users balance long-term preferences, recent evidence, and novelty during sequential decision-making. Although memory modules are commonly used to store historical interactions, they mainly provide behavioural evidence about what the user has consumed [22, 32, 33] and do not explicitly specify how long-term and recent signals should influence the next decision [24]. This motivates an intent-aware formulation [28], where a dynamic intent signal

controls whether the simulated decision should follow stable preferences, recent preference shifts, or exploratory behaviour. In this view, intent is not a replacement for memory, but a decision-control layer derived from user history.

Based on these observations, this paper investigates three research questions: (**RQ1**) How can high-level behavioural intent be inferred from observable preference transitions in historical interactions? (**RQ2**) To what extent can conditioning LLM-based user simulators on inferred intent improve the alignment between simulated and real user behaviour? (**RQ3**) How can intent-aware behavioural control be used as a plug-in module across different user profile backbones and simulation settings?

To address these challenges, we propose **IntRecSim**, a plug-in framework for intent-aware LLM-based user simulation in recommender systems. Instead of treating simulation as static profile-conditioned preference matching, IntRecSim explicitly models the behavioural mode that may guide the user's next decision. The framework introduces a dynamic intent layer that determines whether the simulated decision should primarily follow long-term preferences, recent behavioural evidence, or exploratory behaviour. The inferred intent is then injected into the LLM-based simulator as a high-level control signal, enabling the agent to adapt how preference evidence is used during sequential decision-making. By separating behavioural control from profile construction, IntRecSim is designed to be compatible with different user-profile backbones and simulation settings. Since standard recommendation datasets do not provide explicit behavioural intent annotations, we construct pseudo-intent supervision from historical interaction sequences using hybrid category and semantic preference signals. This design enables intent signals to capture both explicit item attributes and fine-grained semantic transitions across domains with different item characteristics. The main contributions of this paper are summarised as follows:

- We identify a key limitation of existing LLM-based user simulators: simulated decisions are often reduced to static profile-conditioned preference matching, leaving the user's current decision mode implicit.
- We propose **IntRecSim**, a plug-in intent-aware simulation framework that explicitly controls LLM-based user decisions through three behavioural modes: *stay_profile*, *follow_recent*, and *explore*.
- We develop a weakly supervised pseudo-intent construction strategy based on hybrid category and semantic preference signals, enabling intent learning from standard interaction sequences without manual annotations.
- We conduct experiments on three recommendation datasets across discrimination and ranking tasks, showing that intent-aware control consistently improves simulation alignment while revealing domain-specific benefits and limitations through intent-wise analysis.

2 Related Work

Recent studies have explored LLM-based agents as user simulators for recommender-system evaluation, aiming to provide controllable alternatives to static offline logs and costly online experiments [11, 16, 17, 23, 34]. RecAgent [22] instantiates users as LLM agents equipped with profile, memory, and action modules, enabling simulated users to browse, click, search, and interact within recommendation environments. Agent4Rec [32] further develops a recommender-specific generative-agent framework with factual and emotional memory, social traits, and page-by-page actions. These frameworks demonstrate the feasibility of using LLM agents to generate realistic recommendation behaviours. However, their decision processes are still largely conditioned on user profiles and retrieved memories, leaving the LLM to implicitly infer whether the next action should follow long-term preferences, recent behaviour, or exploratory interests. As a result, existing simulators mainly model user behaviour as preference matching, without explicitly controlling the behavioural mode underlying each decision step.

A growing line of work attempts to improve simulation quality by enriching the information provided to the agent. LLM-Powered User Simulator introduces a logical-and-statistical preference inference pipeline for predicting user engagement, while PUB incorporates personality traits to increase behavioural diversity and realism [15, 37]. APG4RecSim focuses on profile construction and shows that task-aligned user profiles can improve simulation performance across discrimination, ranking, and rating tasks [25]. These methods improve the quality of behavioural evidence available to the simulator, including preferences, traits, and task-relevant policies. Nevertheless, they primarily operate at the level of profile construction or stable user attributes, leaving an important research gap: existing frameworks do not explicitly model how different preference sources should guide the current decision step.

User preferences are inherently dynamic, and recent interactions may reflect temporary interests or evolving behavioural tendencies. Recent work such as DyTA4Rec incorporates dynamic profile updating and adaptive aggregation to improve sensitivity to behavioural dynamics [24]. These approaches primarily focus on improving how behavioural evidence is represented and updated, often through iterative LLM-based profile refinement. In contrast, our work focuses on a complementary aspect of simulation: explicitly modelling how different behavioural signals should influence the current decision step. We introduce a lightweight plug-in intent-control layer that explicitly guides whether the next simulated action should prioritise stable long-term preferences, recent behavioural tendencies, or exploratory interests. In this formulation, profile and memory modules provide behavioural evidence, while intent control modulates how that evidence is used during generation.

3 Methodology

This section defines the intent-aware user-simulation problem and then describes pseudo-intent construction, intent-model training, and intent-guided simulation. Figure 1 illustrates the overall framework of IntRecSim, where stable user preferences and dynamic behavioural control are modelled separately to guide simulated decisions.

3.1 Problem Definition

3.1.1 User Behaviour Simulation. For each user u, we denote the chronological interaction history as:

$$\mathcal{H}_u = \{(i_k, r_k, \tau_k)\}_{k=1}^{n}, \tag{1}$$

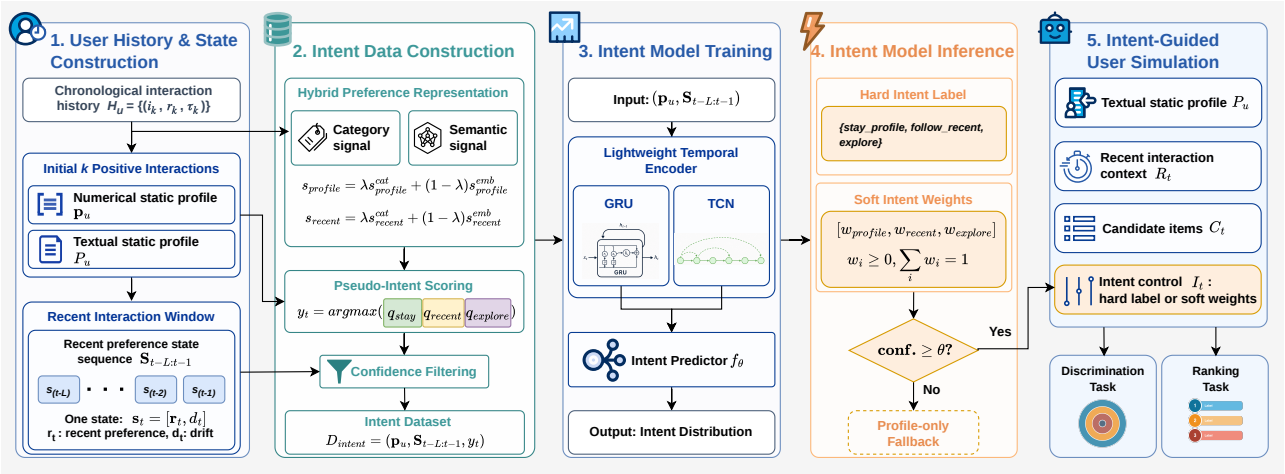


Figure 1: Framework overview of intent-aware LLM-based user simulation. Historical interactions are used to construct a static profile and recent preference state sequence, from which pseudo-intent labels are derived using hybrid category–semantic signals. A lightweight temporal classifier predicts behavioural intent, which is used as either a hard label or soft control weights to guide the LLM simulator for discrimination and ranking tasks.

where i_k is the interacted item, r_k denotes the observed feedback signal, such as a rating, click, or review, and τ_k is the timestamp.

At a decision step t, the system provides a candidate item set:

$$C_t = \{c_1, c_2, \ldots, c_m\}. \tag{2}$$

The goal of user simulation is to predict the likely response to the candidate set. Depending on the evaluation task, the response may be a selected item, a ranking over candidates, or a predicted rating. In this work, the task is primarily formulated as predicting the item $\hat{i}_t \in C_t$ most likely to be selected by the simulated user.

Existing LLM-based user simulators commonly condition the LLM on a user profile, historical interactions, and candidate items. However, such formulations often do not explicitly model the behavioural mode that determines how long-term preferences and recent interaction evidence should be used at each decision step. For example, a user decision may be guided by stable long-term preferences, short-term recent interests, or exploratory behaviour. We therefore formulate user simulation as an intent-aware behavioural control problem, where a dynamic behavioural signal is used to modulate the LLM-based simulator.

3.1.2 Static Profile and Recent Preference State. For each user u, we distinguish between long-term preference and recent behavioural dynamics. Let $\mathbf{p}_u$ denote the numerical static profile used by the intent predictor, and let P_u denote the corresponding textual profile provided to the LLM simulator. The static profile is constructed from the user's initial interactions and serves as the long-term preference baseline. In our framework, $\mathbf{p}_u$ can include both category-level and semantic preference representations, while P_u expresses the same long-term preference information in natural language for LLM prompting.

At each time step t, we define a recent preference state as: $\mathbf{s}_t = [\mathbf{r}_t, d_t]$, where $\mathbf{r}_t$ encodes the user's recent preference representation before the current decision, and d_t is a drift signal measuring how much the recent preference deviates from the static profile. Thus, $\mathbf{r}_t$ captures short-term behavioural evidence, while d_t describes its relative shift from the user's long-term preference baseline.

For intent prediction, we construct a target-independent recent state sequence:

$$\mathbf{S}_{t-L:t-1} = [\mathbf{s}_{t-L}, \ldots, \mathbf{s}_{t-1}], \tag{3}$$

where L is the sequence length. This sequence summarises how the user's recent preferences evolve relative to the long-term profile before the current decision step. Importantly, $\mathbf{S}_{t-L:t-1}$ only uses information available before time t; the target item at time t is used later only for constructing pseudo-intent supervision.

3.1.3 Intent-Aware Behavioural Control. We define user intent as an operational behavioural control signal rather than an explicitly observed psychological state. Specifically, intent represents how a simulated user balances long-term preference, recent behavioural evidence, and exploratory behaviour when making the next interaction. This signal can be instantiated either as a discrete mode label or as a soft distribution over behavioural modes. Formally, we define three intent modes:

$$\mathcal{Z} = \{\texttt{stay_profile}, \texttt{follow_recent}, \texttt{explore}\}. \tag{4}$$

A discrete intent is represented as $z_t \in \mathcal{Z}$, while a soft intent is represented as $\boldsymbol{\alpha}_t \in \Delta^{|\mathcal{Z}|}$, where each element denotes the weight assigned to one behavioural mode.

The three modes correspond to different decision strategies: `stay_profile` indicates that the decision is mainly guided by stable long-term preferences; `follow_recent` indicates stronger reliance on recent preference patterns; and `explore` indicates a decision weakly constrained by both long-term and recent preferences.

Under this formulation, intent-aware simulation consists of two steps. First, an intent predictor estimates the current behavioural

control signal from information available before the decision:

$$\hat{\mathbf{z}}_t = g_\theta(\mathbf{p}_u, \mathbf{S}_{t-L:t-1}), \tag{5}$$

where $\hat{\mathbf{z}}_t$ denotes either a discrete intent label $\hat{z}_t$ or a soft intent distribution $\hat{\boldsymbol{\alpha}}_t$, $\mathbf{p}_u$ denotes the user-level preference representation, $\mathbf{S}_{t-L:t-1}$ denotes a compact state sequence over the recent L steps, and $g_\theta(\cdot)$ is the intent prediction function.

Second, the predicted intent signal is injected into the LLM-based simulator as a high-level behavioural control signal:

$$\hat{i}_t = \mathrm{LLM}_\phi(P_u, R_t, \hat{\mathbf{z}}_t, C_t), \tag{6}$$

where P_u is the textual user profile, R_t is the recent interaction context exposed to the LLM, C_t is the candidate item set, and $\hat{i}_t$ is the simulated choice. Here, $\mathbf{S}_{t-L:t-1}$ provides numerical states for intent prediction, whereas R_t provides item-level context for LLM reasoning over candidates.

3.2 Intent Data Construction

Since explicit ground-truth behavioural intent is unavailable in standard recommendation datasets, our goal is not to recover latent psychological intent. Instead, we define intent as an operational behavioural-control signal derived from observable preference transitions. We validate this signal through downstream utility and sanity checks: intent-enhanced simulators outperform profile-only simulators across most settings; random intent does not produce similar gains; pseudo-oracle intent provides stronger or complementary improvements; and per-intent analysis shows that different behavioural modes exhibit distinct simulation effects. These results suggest that, despite inevitable noise in pseudo-intent supervision, the constructed signal captures useful decision-control information beyond static profiles.

We therefore construct pseudo-intent supervision from observed user histories. For each target interaction at time step t, we compare the target item with the user's static profile and recent preference representation to estimate whether the interaction is better explained by long-term preference, recent preference, or exploration. The target item is used only for constructing pseudo-intent supervision; the intent predictor itself receives only pre-decision information, namely the static profile and recent state sequence.

Hybrid preference representation. We use both category-level and semantic embedding representations to capture complementary preference signals. Category information provides explicit and interpretable preference structure, which is particularly useful for movie recommendation, where genres largely reflect user interests. However, in book and beauty-product recommendation, coarse categories are often insufficient to capture real shopping preferences, where author, topic, brand, form, and fine-grained semantic similarity may be important. Therefore, we combine category-based similarity with embedding-based similarity to construct more robust intent supervision across domains.

Let $\mathbf{p}_u^{\mathrm{cat}}$ and $\mathbf{p}_u^{\mathrm{emb}}$ denote the category- and embedding-based static profiles of user u, and let $\mathbf{r}_t^{\mathrm{cat}}$ and $\mathbf{r}_t^{\mathrm{emb}}$ denote the corresponding recent preference representations before time t. For the target item, let $\mathbf{v}_t^{\mathrm{cat}}$ and $\mathbf{v}_t^{\mathrm{emb}}$ denote its category and embedding representations. We compute target–profile and target–recent alignment scores as:

$$\tilde{s}_{\mathrm{profile}} = \lambda \mathrm{sim}_{\mathrm{cat}}(\mathbf{v}_t^{\mathrm{cat}}, \mathbf{p}_u^{\mathrm{cat}}) + (1-\lambda)\cos(\mathbf{v}_t^{\mathrm{emb}}, \mathbf{p}_u^{\mathrm{emb}}), \tag{7}$$

$$\tilde{s}_{\mathrm{recent}} = \lambda \mathrm{sim}_{\mathrm{cat}}(\mathbf{v}_t^{\mathrm{cat}}, \mathbf{r}_t^{\mathrm{cat}}) + (1-\lambda)\cos(\mathbf{v}_t^{\mathrm{emb}}, \mathbf{r}_t^{\mathrm{emb}}), \tag{8}$$

where $\lambda \in [0,1]$ controls the category–embedding trade-off, and $\mathrm{sim}_{\mathrm{cat}}(\cdot)$ denotes the category-level similarity function. To account for user-specific similarity ranges, $\tilde{s}_{\mathrm{profile}}$ and $\tilde{s}_{\mathrm{recent}}$ are min–max normalised within each user's interaction sequence, producing $s_{\mathrm{profile}}, s_{\mathrm{recent}} \in [0,1]$. This normalisation makes profile- and recent-alignment scores comparable when constructing intent scores.

Pseudo-intent score construction. We define three behavioural-mode scores corresponding to the intent modes in $\mathcal{Z}$:

$$q_{\mathrm{stay}} = s_{\mathrm{profile}}, \quad q_{\mathrm{recent}} = s_{\mathrm{recent}}, \tag{9}$$

$$q_{\mathrm{explore}} = \eta(1 - s_{\mathrm{profile}})(1 - s_{\mathrm{recent}}). \tag{10}$$

where η controls the relative strength of the exploration mode. The intuition is that a target item strongly aligned with the static profile indicates `stay_profile`, a target item more aligned with recent preferences indicates `follow_recent`, and an item weakly aligned with both long-term and recent preferences indicates `explore`.

The soft intent distribution is obtained by normalising the three scores:

$$\boldsymbol{\alpha}_t = \mathrm{Normalize}(q_{\mathrm{stay}}, q_{\mathrm{recent}}, q_{\mathrm{explore}}), \tag{11}$$

where $\boldsymbol{\alpha}_t \in \Delta^{|\mathcal{Z}|}$ represents the relative strength of the three behavioural modes.

When a discrete pseudo-intent label is required, we assign:

$$y_t = \arg\max_{z \in \mathcal{Z}} q_z, \tag{12}$$

where $\mathcal{Z}$ is the set of intent modes defined in Eq. (4).

To reduce noisy pseudo-supervision, we compute the annotation confidence as:

$$c_t = \max_{z \in \mathcal{Z}} \alpha_{t,z}. \tag{13}$$

Low-confidence instances can be filtered during intent-model training, while simulation evaluation instances are kept unchanged.

Each training instance consists of the static profile, recent state sequence, soft intent distribution, and hard pseudo-label. The soft distribution $\boldsymbol{\alpha}_t$ supports soft supervision or soft behavioural control, while y_t provides a hard pseudo label for discrete intent prediction and prompt-based control.

3.3 Intent Model Training

Given the constructed intent dataset $\mathcal{D}_{\mathrm{intent}}$, we train a lightweight intent predictor to estimate the behavioural control signal for the next simulated decision. The predictor does not model item selection directly; instead, it predicts how the LLM simulator should balance long-term preference, recent behavioural evidence, and exploration.

3.3.1 Input Representation. For each instance, the model takes $(\mathbf{p}_u, \mathbf{S}_{t-L:t-1})$ as input, where $\mathbf{p}_u$ represents the user's long-term preferences and $\mathbf{S}_{t-L:t-1}$, defined in Eq. (3), is the target-independent recent-state sequence available before time t. Each state encodes recent preference information and its deviation from the static profile.

This representation allows the model to capture both category-level shifts and semantic preference changes relative to the user's long-term profile.

The prediction target can be either a hard pseudo-intent label $y_t \in \mathcal{Z}$ or a soft intent distribution $\boldsymbol{\alpha}_t \in \Delta^{|\mathcal{Z}|}$, where $\mathcal{Z}$ is the set of behavioural intent modes defined in Eq. (4).

3.3.2 Model Architecture. We formulate intent prediction as a three-way sequence classification problem:

$$\hat{\boldsymbol{\alpha}}_t = f_\theta(\mathbf{p}_u, \mathbf{S}_{t-L:t-1}), \tag{14}$$

where $\hat{\boldsymbol{\alpha}}_t \in \Delta^{|\mathcal{Z}|}$ is the predicted distribution over intent modes. Since our goal is not to design a new intent predictor, we consider two lightweight temporal encoders: GRU [5] and TCN [1]. The GRU captures recurrent transitions in recent preference states, while the TCN captures local and multi-step temporal patterns through causal convolutions. In both cases, the static profile is used as conditioning information so that recent dynamics are interpreted relative to the user's long-term preference context.

3.3.3 Training Objective. For hard pseudo-label supervision, we train the intent predictor with weighted cross-entropy and label smoothing:

$$\mathcal{L}_{\text{intent}} = -\sum_{z \in \mathcal{Z}} w_z \tilde{y}_{t,z} \log \hat{\boldsymbol{\alpha}}_{t,z}, \tag{15}$$

where $\hat{\boldsymbol{\alpha}}_t$ is the predicted intent distribution, w_z is the class weight for intent mode z, and $\tilde{\mathbf{y}}_t$ is the smoothed target distribution derived from the hard pseudo label y_t. We use a smoothing factor of 0.1. Class weighting reduces the effect of class imbalance, while label smoothing mitigates over-confidence under noisy pseudo-label supervision.

The framework can also use soft intent supervision by replacing $\tilde{\mathbf{y}}_t$ with the constructed soft intent distribution $\boldsymbol{\alpha}_t$. Models are selected using validation macro-F1. For discrete prompt-based control, the predicted intent is obtained by $\hat{z}_t = \arg\max_{z \in \mathcal{Z}} \hat{\boldsymbol{\alpha}}_{t,z}$; alternatively, $\hat{\boldsymbol{\alpha}}_t$ can be directly used for soft behavioural control.

3.4 Intent Model Inference

At inference time, the model predicts $\hat{\boldsymbol{\alpha}}_t$ using Eq. (14), and the discrete intent is obtained as $\hat{z}_t = \arg\max_{z \in \mathcal{Z}} \hat{\boldsymbol{\alpha}}_{t,z}$, where $\mathcal{Z}$ denotes the intent-mode set defined in Eq. (4).

To reduce the effect of uncertain predictions, we compute the confidence score as $\hat{c}_t = \max_{z \in \mathcal{Z}} \hat{\boldsymbol{\alpha}}_{t,z}$. We apply a confidence gate: if $\hat{c}_t < \theta$, the intent signal is omitted and the simulator falls back to the profile-only prompt; otherwise, $\hat{z}_t$ is injected into the prompt as a behavioural instruction. Specifically, `stay_profile` emphasises long-term preferences, `follow_recent` emphasises recent interaction patterns, and `explore` allows less profile-constrained choices.

3.5 Intent-Guided User Simulation

After intent inference, the predicted intent signal is injected into the LLM-based user simulator as a high-level behavioural control signal. For each simulation instance, the LLM receives the textual static profile P_u, recent interaction context R_t, candidate items C_t, and an intent signal I_t. The simulator then predicts the user's response over the candidate set, either by selecting item(s) in the discrimination task or ranking all candidates in the ranking task:

$$\hat{y}_t = \text{LLM}_\phi(P_u, R_t, I_t, C_t), \tag{16}$$

where $\hat{y}_t$ denotes the simulated output.

We consider two forms of intent control. In the hard-intent setting, $I_t = \hat{z}_t$, where $\hat{z}_t \in \mathcal{Z}$ is the predicted behavioural mode. The intent label is provided to the LLM as an explicit instruction: `stay_profile` encourages decisions consistent with long-term preferences, `follow_recent` emphasises recent interaction patterns, and `explore` allows choices less constrained by both long-term and recent preferences.

In the soft-intent setting, $I_t = \hat{\boldsymbol{\alpha}}_t$, where $\hat{\boldsymbol{\alpha}}_t \in \Delta^{|\mathcal{Z}|}$ is the predicted distribution over intent modes. Rather than assigning a single mode, soft intent prompting asks the LLM to jointly consider long-term preference, recent evidence, and exploration according to their predicted weights.

As a comparison, we also use a profile-only simulator, which conditions the LLM only on the static profile and recent context. Comparing profile-only, hard-intent, and soft-intent simulation allows us to evaluate whether explicit behavioural control improves LLM-based user simulation beyond profile and history conditioning.

4 Experiments

This section presents a comprehensive experimental evaluation of the proposed method, detailing the experimental setup, baselines, and results across multiple tasks and datasets. Furthermore, we conduct rigorous ablation studies, intent quality analysis, and robustness and efficiency analysis.

4.1 Experimental Setup

4.1.1 Datasets. We conduct experiments on three public recommendation datasets: MovieLens-1M (ML-1M) [9], Amazon-Book [10], and Amazon-Beauty [10]. For each dataset, interactions are sorted chronologically by timestamp. We retain users with at least 30 interactions to ensure sufficient history for profile construction, intent learning, and simulation evaluation. For each user, the first $K = 10$ interactions are used to initialise the static profile, the next $N_{\text{train}} = 10$ interactions are used to construct intent-training instances, and the following $N_{\text{val}} = 5$ interactions are reserved for intent-model validation. All remaining interactions are used for downstream simulation evaluation.

4.1.2 Baselines. We apply IntRecSim to five baseline simulators: **Recent Items**, **RecAgent** [22], **Agent4Rec** [32], **APG4RecSim** [25], and a non-LLM **Statistic Profile**. The Statistic Profile is constructed from structured item attributes without using LLMs: genre for ML-1M, genre and author for Amazon-Book, and form and brand for Amazon-Beauty. For each baseline, we compare the profile-only simulator with two intent-enhanced variants, **+GRU Intent** and **+TCN Intent**, where the intent signal is predicted by the corresponding temporal encoder.

4.1.3 Evaluation Settings. We consider two simulation tasks. In the **discrimination task**, the simulator selects the ground-truth next item from a candidate set containing one positive item and sampled negatives. Performance is evaluated by Hit@1 under 1:6 and 1:10 candidate settings. In the **ranking task**, the simulator ranks candidate items for the next interaction, and performance is evaluated by

Table 1: Intent prediction performance on constructed pseudo-intent labels. Macro-F1 and per-class F1 are reported to account for class imbalance.

Dataset	Model	Acc	Macro-F1	F1-stay	F1-rec	F1-exp
ML-1M	GRU	0.4904	0.4501	0.6225	0.3261	0.4018
	TCN	0.4808	0.4406	0.6135	0.3572	0.3510
Book	GRU	0.6503	0.5890	0.7533	0.5208	0.4928
	TCN	0.6396	0.5776	0.7476	0.5058	0.4792
Beauty	GRU	0.4882	0.4381	0.6283	0.3717	0.3143
	TCN	0.4612	0.3479	0.6187	0.3404	0.0845

nDCG@5 and HR@3. Note that while both metrics are hit-based, they are used under different evaluation protocols. Hit@1 measures exact match accuracy in the discrimination task, whereas HR@3 evaluates whether the ground-truth item is retrieved within the top-3 ranked candidates in the ranking task, reflecting a softer retrieval objective. All methods are evaluated using the same candidate sets, recent interaction context, LLM backbone, and decoding settings for fair comparison. At the distribution level, we compare real and simulated genre distributions using Pearson correlation, Spearman rank correlation, and Jensen–Shannon divergence (JSD). Higher correlation coefficients and lower JSD indicate better distributional alignment.

4.1.4 Implementation Details. We use `gpt-4o-mini` [2] as the LLM-based user simulator through Azure OpenAI [3]. The decoding temperature is set to 0.1 to reduce randomness while preserving limited response diversity. To improve efficiency, simulation requests are executed asynchronously with a concurrency limit of 8. Each simulation instance is repeated $R = 3$ times, and the final output is determined by majority voting to improve robustness against stochastic LLM responses.

4.2 Intent Analysis

To answer RQ1, we evaluate whether the constructed preference-state sequences contain sufficient behavioural signal for predicting pseudo-intent labels.

4.2.1 Intent prediction quality. Table 1 shows that the intent predictors achieve moderate but useful performance, suggesting that the constructed preference-state sequences contain meaningful behavioural signals. Amazon-Book yields the strongest results, with GRU reaching 0.6503 accuracy and 0.5890 macro-F1, while ML-1M and Amazon-Beauty are more challenging. Across datasets, `stay_profile` is predicted most reliably, whereas `follow_recent` and especially `explore` are harder to identify, reflecting the ambiguity of recent-shift and exploratory behaviours under weak pseudo-label supervision. GRU generally outperforms TCN, particularly on Amazon-Beauty. Despite imperfect intent prediction, the downstream results show that the predicted intent signals remain effective for improving user simulation.

[2]https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/
[3]https://azure.microsoft.com/en-au/products/ai-foundry/models/openai

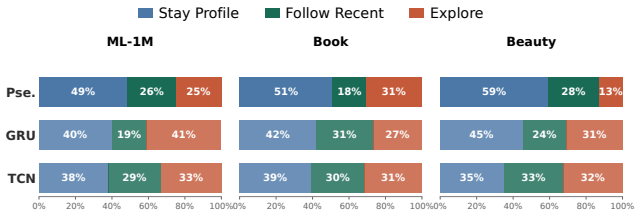


Figure 2: Distribution of constructed and predicted intent labels across datasets. Pse. denotes the constructed pseudo-intent distribution, while GRU and TCN denote the distributions predicted by the corresponding intent models. The bars show the proportion of samples assigned to each behavioural mode.

4.2.2 Predicted intent distribution. We compare the constructed pseudo-intent distribution with the distributions predicted by the GRU and TCN intent models (Figure 2). Across datasets, `stay_profile` is the dominant mode in the pseudo labels, suggesting that many interactions remain consistent with long-term preferences. The learned models broadly recover this dominant pattern, indicating that the constructed intent labels contain stable behavioural structure beyond random assignment. However, the models tend to assign fewer instances to `stay_profile` and more to `follow_recent` or `explore`, especially on ML-1M and Amazon-Beauty. This suggests that recent-shift and exploratory behaviours are harder to separate under weak supervision, which is consistent with the intent prediction results in Table 1. Despite these distributional differences, the downstream experiments show that the predicted intent signals still provide useful behavioural control for LLM-based simulation.

4.2.3 Fine-grained Intent Analysis. To better understand when intent control improves simulation, we group test instances by their constructed pseudo-intent labels and report per-group Hit@1 changes over the profile-only simulator. Figure 3 shows that intent control improves overall performance across datasets, but the gains are not uniform across behavioural modes.

The results show that `stay_profile` instances generally benefit from intent control, especially on ML-1M and Amazon-Beauty, suggesting that explicit intent helps the simulator rely more consistently on stable long-term preferences. For `follow_recent`, the effect is more domain-dependent: intent control improves ML-1M and Amazon-Book in several cases, but can hurt Amazon-Beauty. This indicates that recent-shift behaviour is more sensitive to pseudo-intent quality and domain-specific item semantics. For `explore`, pseudo-oracle intent improves Amazon-Beauty, while learned intent models are less stable, suggesting that exploration is useful but difficult to predict reliably.

Overall, this analysis supports the value of intent-aware control while revealing an important limitation: different behavioural modes vary in predictability and usefulness across domains. This explains why intent control improves overall simulation performance but may still underperform in specific intent groups or datasets.

4.2.4 Stability of Intent-wise Predictability Patterns. Beyond analysing the magnitude of intent-wise gains, we further examine whether

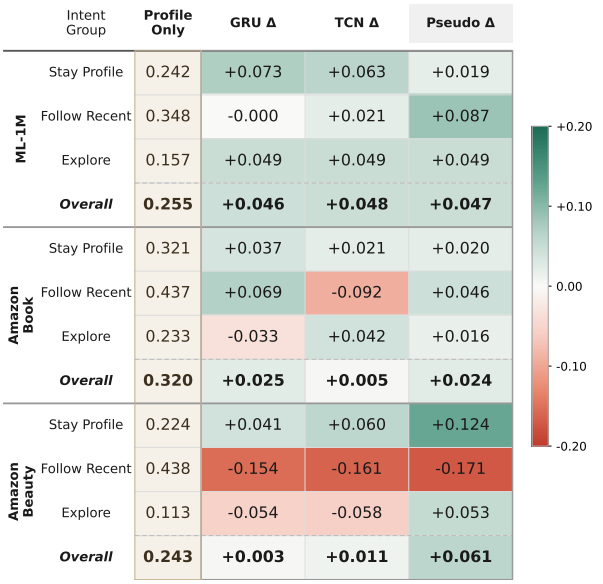


Figure 3: Per-intent discrimination performance relative to the profile-only simulator. Test instances are stratified by pseudo-oracle intent labels. The Profile Only column reports the baseline Hit@1 on the 1:10 discrimination task, while GRU, TCN, and Pseudo-Intent Label report ΔHit@1 relative to the profile-only setting. Green denotes performance gains, whereas red denotes performance degradation.

Table 2: Intent-wise predictability consistency. FR, SP, and EX denote `follow_recent`, `stay_profile`, and `explore`, respectively.

Dataset	Method	Order by Hit@1	ρ
ML-1M	Profile, GRU, TCN, Pseudo	FR > SP > EX	1.00
Book	Profile, GRU, TCN, Pseudo	FR > SP > EX	1.00
Beauty	Profile, GRU	FR > SP > EX	1.00
Beauty	TCN, Pseudo	SP > FR > EX	0.50
Perfect order ratio		10/12	
Mean Spearman ρ		0.917	

the constructed pseudo-intent groups exhibit stable behavioural predictability patterns. For each dataset–method setting, we rank the three intent groups by absolute Hit@1, using the profile-only value plus the reported Δ in Figure 3. As shown in Table 2, 10 out of 12 settings follow the order FR > SP > EX, with a mean Spearman correlation of 0.917. This suggests that recent-aligned decisions are generally easiest to recover, profile-aligned decisions are intermediate, and exploratory decisions are the most challenging. The two exceptions occur on Amazon-Beauty under TCN and pseudo-intent control, where SP slightly exceeds FR, indicating that domain-specific item semantics can affect the relative difficulty of recent-shift behaviours.

Table 3: Genre-level distribution alignment on ML-1M.

Simulator	Pearson↑	Spearman↑	JSD ↓
Profile baseline	0.9749	0.8759	0.1246
IntRecSim	0.9840	0.9133	0.1018

4.3 Main Results

4.3.1 Overall effectiveness. **Item-level.** To answer RQ2, we compare profile-only simulation with intent-enhanced simulation across discrimination and ranking tasks. Table 4 reports the main results on three datasets, five backbones, and two simulation tasks. Overall, intent-aware control improves most profile-only baselines, showing that the predicted intent signal provides useful behavioural guidance beyond static profile and recent-history conditioning. The improvements are observed in both discrimination and ranking tasks, suggesting that intent control helps the simulator not only identify the next likely item but also produce better candidate orderings.

Distribution-level. Beyond next-item recovery, we evaluate whether simulated interactions preserve coarse-grained distributional properties. Figure 4a shows genre-level interaction distributions on ML-1M. Both profile-only and IntRecSim achieve high correlation with real user distributions, indicating that profile-based conditioning already captures macro-level preference structure. IntRecSim further improves Pearson correlation (0.975→0.984), Spearman correlation (0.876→0.913), and reduces Jensen–Shannon divergence (0.125→0.102).

4.3.2 Complementarity across profile backbones. To answer RQ3, we apply the same intent-control mechanism to multiple profile backbones, including LLM-generated and non-LLM profiles. For the simple **Recent Items** baseline, intent control substantially improves ranking performance; for example, HR@3 on ML-1M increases from 43.56 to 55.56/56.44 with GRU/TCN intent. For stronger LLM-based profiles, including **RecAgent**, **Agent4Rec**, and **APG4RecSim**, intent-enhanced variants also generally outperform their profile-only counterparts. In particular, APG4RecSim improves on ML-1M from 38.70 to 48.45/50.01 in Hit@1 under the 1:6 discrimination setting, and from 49.73 to 53.60/53.06 in nDCG@5. These results indicate that intent control is complementary to profile construction rather than a replacement for it.

4.3.3 Effect of backbone strength. The magnitude of improvement suggests that intent control is especially useful when the profile backbone provides limited behavioural guidance. **Recent Items** and **Simple Profile** obtain substantial gains in several ranking settings, indicating that explicit intent can help the simulator better interpret sparse or coarse preference evidence. However, the gains are not limited to weaker profiles: **APG4RecSim** also benefits consistently from intent control. This suggests that intent-aware control can both compensate for missing behavioural structure in simpler profiles and complement stronger profiles by specifying how profile evidence should be used at the current decision step.

4.3.4 Intent predictors and failure cases. Comparing GRU and TCN, both temporal encoders are effective, but neither consistently dominates across all datasets and metrics. The grey-shaded cells in

Table 4: Main simulation results on ML-1M, Amazon-Book, and Amazon-Beauty. We evaluate five baselines and their intent-enhanced variants using GRU- and TCN-based intent predictors. For each backbone, the baseline uses the corresponding profile plus the same recent interaction context as the intent-enhanced variants, but without explicit intent control. Recent Items uses the raw features of the five most recent interacted items as the profile. Simple Profile denotes our LLM-free statistical profile constructed from available item attributes. Grey shading marks cases where the intent-enhanced variant does not outperform the corresponding profile-only baseline. Results are reported as mean performance with standard deviation over repeated runs, and are expressed as percentages (×100).

Model	ML-1M				Amazon-Book				Amazon-Beauty			
	Discrimination		Ranking		Discrimination		Ranking		Discrimination		Ranking	
	1:6	1:10	nDCG@5	HR@3	1:6	1:10	nDCG@5	HR@3	1:6	1:10	nDCG@5	HR@3
Recent Items	39.83(±2.46)	21.62(±1.91)	41.02(±2.51)	43.56(±2.51)	35.14(±1.27)	26.07(±0.96)	41.75(±1.07)	43.02(±0.64)	27.44(±1.60)	17.46(±1.92)	35.06(±1.22)	38.13(±1.61)
+GRU Intent	40.28(±2.21)	21.62(±1.91)	46.49(±2.04)	55.56(±1.26)	35.81(±0.64)	26.17(±0.00)	41.88(±0.77)	45.78(±0.94)	29.94(±2.89)	23.14(±1.60)	37.13(±1.90)	40.18(±1.94)
+TCN Intent	41.22(±3.19)	23.54(±2.20)	47.77(±0.88)	56.44(±1.57)	38.13(±1.60)	27.11(±0.31)	41.88(±0.99)	45.33(±0.63)	29.47(±1.28)	21.32(±1.28)	37.78(±1.54)	41.78(±1.61)
RecAgent	36.68(±0.96)	26.12(±0.64)	44.37(±2.69)	48.22(±3.46)	38.51(±0.00)	31.28(±0.32)	44.61(±0.80)	45.11(±1.26)	28.73(±1.30)	23.15(±2.29)	39.08(±0.80)	42.99(±2.28)
+GRU Intent	38.00(±0.63)	27.77(±2.51)	45.08(±1.37)	50.44(±1.57)	39.78(±0.31)	30.01(±0.94)	45.00(±1.39)	48.99(±2.21)	30.34(±1.30)	21.99(±0.65)	42.07(±1.94)	45.29(±1.63)
+TCN Intent	38.89(±0.94)	26.40(±2.21)	45.04(±1.36)	50.22(±1.89)	40.31(±0.64)	30.63(±0.96)	45.39(±1.65)	50.34(±1.90)	31.94(±0.98)	22.92(±0.65)	39.93(±1.47)	47.03(±1.61)
Agent4Rec	33.10(±0.63)	25.50(±1.27)	46.45(±0.70)	48.44(±0.63)	44.44(±2.24)	29.99(±1.57)	50.25(±1.25)	53.56(±1.26)	36.13(±1.98)	22.67(±1.60)	43.78(±1.71)	46.30(±1.64)
+GRU Intent	37.56(±1.26)	31.55(±1.57)	52.37(±1.71)	56.00(±1.57)	45.50(±0.64)	34.45(±1.57)	50.46(±0.52)	53.56(±1.57)	38.09(±2.89)	26.30(±1.28)	45.65(±2.27)	50.23(±1.96)
+TCN Intent	38.67(±0.63)	30.01(±1.57)	53.30(±2.14)	57.33(±1.26)	45.19(±0.63)	34.44(±1.89)	50.50(±2.24)	54.22(±3.14)	36.53(±1.94)	25.86(±2.24)	44.09(±1.62)	47.82(±1.30)
APG4RecSim	38.70(±1.27)	25.45(±1.59)	49.73(±0.86)	50.67(±1.89)	44.07(±2.53)	31.99(±1.57)	47.10(±1.12)	50.22(±0.63)	33.34(±1.65)	24.26(±1.32)	47.76(±1.29)	52.61(±1.92)
+GRU Intent	48.45(±1.57)	30.01(±1.57)	53.60(±1.26)	55.78(±1.89)	47.20(±0.95)	34.45(±1.89)	51.10(±1.07)	52.44(±0.31)	39.59(±1.64)	24.60(±0.98)	50.59(±1.10)	56.62(±0.97)
+TCN Intent	50.01(±2.83)	30.23(±1.26)	53.06(±0.94)	56.00(±0.94)	48.99(±1.27)	33.78(±0.94)	50.95(±1.29)	52.00(±1.26)	34.70(±2.24)	25.39(±0.96)	49.25(±1.00)	54.11(±1.61)
Simple Profile	35.36(±1.27)	26.38(±0.98)	37.53(±1.44)	38.93(±1.90)	33.33(±0.63)	24.78(±0.32)	42.50(±1.06)	47.75(±1.91)	35.65(±1.64)	25.23(±0.98)	43.66(±1.87)	44.52(±1.65)
+GRU Intent	42.28(±0.63)	30.21(±1.58)	48.65(±2.19)	53.69(±1.90)	37.55(±0.31)	25.50(±0.00)	43.05(±1.75)	48.65(±2.23)	36.39(±2.32)	25.87(±0.66)	44.39(±1.76)	48.36(±1.33)
+TCN Intent	41.83(±2.21)	28.39(±1.90)	48.60(±2.09)	54.14(±2.21)	37.33(±0.63)	27.07(±0.95)	43.03(±1.69)	48.30(±1.92)	34.97(±0.66)	26.34(±0.99)	44.48(±2.05)	49.65(±1.98)

Table 4 indicate cases where intent-enhanced variants do not outperform the corresponding profile-only baseline. These cases are limited, but they suggest several conditions under which intent control may become less reliable.

First, intent prediction can be affected by data sparsity. Although we filter users with at least 30 interactions, some datasets still provide relatively limited training instances for learning dynamic intent, especially for less frequent or more ambiguous modes such as `follow_recent` and `explore`. As shown in the intent prediction analysis, these modes are harder to predict than `stay_profile`. In such cases, an incorrect intent signal may misguide the simulator, particularly in harder discrimination settings where candidate items are more difficult to distinguish.

Second, the marginal benefit of explicit intent may depend on how much behavioural evidence is already exposed to the LLM. When the prompt contains rich recent-item context, such as actual item titles or detailed item descriptions, the LLM may already infer short-term preference shifts from the context itself. In this setting, the predicted intent signal can become partially redundant with information already available in the prompt, reducing its additional contribution. By contrast, when recent preferences are represented in a more compressed form, such as genre or attribute distributions, the user's current decision mode is less directly observable, and explicit intent control may provide more useful guidance.

Overall, these failure cases suggest that intent-aware control is most effective when the recent context is informative but not already sufficient for the LLM to infer the decision strategy. Its effectiveness depends not only on the quality of the intent predictor, but also on data sparsity, candidate difficulty, domain characteristics, metadata quality, and the level of behavioural detail provided in the simulator prompt.

4.4 Ablation Study

Table 5 reports ablation results using APG4RecSim as the profile backbone on ML-1M and Amazon-Book. We examine the effects of recent states (RS), pseudo-intent control, confidence gating, and learned intent predictors.

Adding recent states improves the profile-only baseline on both datasets, showing that short-term behavioural information provides useful evidence beyond the static profile. For example, on ML-1M, nDCG@5 increases from 50.65 to 52.78, and HR@3 increases from 50.89 to 54.81. In contrast, adding random intent substantially degrades Hit@1 on both datasets, from 48.00 to 40.72 on ML-1M and from 43.25 to 38.97 on Amazon-Book. This confirms that the improvement does not come from adding arbitrary prompt instructions, but from meaningful behavioural intent.

Pseudo-intent variants further demonstrate the value of intent control. Compared with RS alone, pseudo soft intent improves Hit@1 from 48.00 to 49.12 on ML-1M and from 43.25 to 48.77 on Amazon-Book. It also improves HR@3 on ML-1M from 54.81 to 57.33. Pseudo hard intent is generally weaker than pseudo soft intent, suggesting that soft intent weights better capture mixed behavioural tendencies than a single discrete mode. Adding confidence gating to pseudo soft intent further improves Amazon-Book, reaching the best Hit@1 of 49.89 and nDCG@5 of 50.21 among pseudo-intent variants.

The learned intent predictors also improve over the RS-only baseline in most settings. On Amazon-Book, GRU and TCN intent improve Hit@1 from 43.25 to 46.39 and 45.86, and nDCG@5 from 48.19 to 50.83 and 51.03, respectively. Adding confidence gating further improves ranking performance, with GRU intent plus confidence gating achieving the best Amazon-Book HR@3 of 52.44. On ML-1M, TCN intent plus confidence gating achieves the best Hit@1 of 50.01, while GRU intent plus confidence gating achieves

Table 5: Ablation results using APG4RecSim as the profile backbone on ML-1M and Amazon-Book. We evaluate the effects of recent states (RS), pseudo-intent control, confidence gate (CG), and learned intent predictors. Discrimination is evaluated by Hit@1 under the 1:6 setting, and ranking is evaluated by nDCG@5 and HR@3. Results are reported as mean performance with standard deviation over repeated runs, and are expressed as percentages (×100).

Model	ML-1M			Amazon-Book		
	Discrimination	Ranking		Discrimination	Ranking	
	1:6	nDCG@5	HR@3	1:6	nDCG@5	HR@3
Profile Only	47.33(±0.00)	50.65(±0.83)	50.89(±0.94)	42.01(±0.94)	47.10(±1.12)	50.22(±0.63)
+ Recent States (RS)	48.00(±0.63)	52.78(±1.80)	54.81(±2.53)	43.25(±0.96)	48.19(±1.21)	50.67(±0.94)
+ RS + Random Intent	40.72(±1.27)	50.15(±1.10)	53.11(±2.20)	38.97(±2.23)	47.60(±1.45)	51.11(±0.94)
+ RS + Pseudo Hard Intent	47.42(±1.90)	52.31(±2.09)	56.00(±1.89)	42.57(±1.27)	49.58(±1.27)	50.44(±1.57)
+ RS + Pseudo Soft Intent	49.12(±1.26)	52.76(±1.09)	57.33(±0.94)	48.77(±0.95)	49.55(±0.67)	51.11(±1.26)
+ RS + Pseudo Soft Intent + CG	49.01(±2.53)	52.92(±0.89)	57.33(±0.94)	49.89(±0.32)	50.21(±1.80)	51.33(±2.20)
+ RS + GRU Intent	47.33(±2.20)	52.61(±0.97)	54.44(±1.26)	46.39(±0.64)	50.83(±1.14)	51.78(±1.89)
+ RS + TCN Intent	48.23(±1.89)	53.03(±1.70)	55.78(±1.26)	45.86(±0.95)	51.03(±1.01)	51.33(±0.63)
+ RS + GRU Intent + CG	48.45(±1.57)	53.60(±1.26)	55.78(±1.89)	47.20(±0.95)	51.10(±1.07)	52.44(±0.31)
+ RS + TCN Intent + CG	50.01(±2.83)	53.06(±0.94)	56.00(±0.94)	47.55(±0.31)	50.95(±1.29)	52.00(±1.26)

the best nDCG@5 of 53.60. Confidence gating provides consistent gains across backbones and datasets, although the optimal threshold θ is dataset-dependent and does not exhibit a consistent trend across settings. We therefore tune θ on the validation set and report results using the selected configuration. A full sensitivity analysis is omitted as it does not reveal a stable pattern across datasets.

Overall, the ablation confirms three findings: recent states provide useful short-term evidence, intent control must be behaviourally meaningful rather than random, and confidence gating can improve robustness by suppressing uncertain intent predictions. These results support the design of IntRecSim as a plug-in intent-control layer on top of profile-based simulation.

4.5 Generalisation across LLM Simulators

To examine whether intent-aware control is specific to a single LLM simulator, we conduct an additional ranking evaluation on Amazon-Book using APG4RecSim as the profile backbone. Table 6 compares profile-only prompting with GRU- and TCN-based intent prompting across different LLM backbones, while keeping the profile representation and simulation pipeline fixed. The results suggest that intent-aware control can transfer across LLM simulators, but its effectiveness is model-dependent. With Llama-3.3-70B-Instruct [4], intent prompting improves over the profile-only baseline: GRU intent increases HR@3 from 0.5200 to 0.5422, while TCN intent improves nDCG@5 from 0.4819 to 0.5005. This indicates that explicit behavioural guidance can be beneficial when the simulator is able to interpret and follow the intent-conditioned prompt.

For grok-4-1-fast-non-reasoning[5], the profile-only baseline is already stronger, achieving 0.6289 nDCG@5 and 0.6711 HR@3. Under this setting, TCN intent provides only a marginal nDCG@5 improvement and matches the baseline HR@3, while GRU intent does not improve performance. This suggests that intent-aware control does not uniformly benefit all LLM–intent-predictor combinations, especially when the base simulator already captures useful behavioural signals from the profile and recent context.

[4]https://github.com/meta-llama/llama-models
[5]https://docs.x.ai/developers/models/grok-4-1-fast-non-reasoning

Table 6: Generalisation of intent control across LLM simulators on Amazon-Book. Values report performance of profile-only and intent-enhanced simulation using APG4RecSim as the profile backbone.

LLM Simulator	Model	nDCG@5	HR@3
Llama-3.3-70B -Instruct	Profile Only	0.4819(±0.0907)	0.5200(±0.0974)
	+GRU Intent	0.4823(±0.0830)	**0.5422(±0.0911)**
	+TCN Intent	**0.5005(±0.1072)**	0.5378(±0.1163)
grok-4-1-fast -non-reasoning	Profile Only	0.6289(±0.0030)	**0.6711(±0.0031)**
	+GRU Intent	0.6272(±0.0082)	0.6644(±0.0063)
	+TCN Intent	**0.6327(±0.0070)**	**0.6711(±0.0063)**

Overall, this analysis provides preliminary evidence for the plug-in applicability of IntRecSim across different LLM simulators. However, the mixed results indicate that intent-aware control should be viewed as a complementary guidance mechanism rather than a guaranteed source of improvement. Its effectiveness depends on both the quality of the inferred intent signal and the extent to which the LLM simulator can operationalise that signal during decision-making.

4.6 Sensitivity and Efficiency Analysis

4.6.1 Window size sensitivity. Figure 4b examines the effect of the recent-state window size L. The base setting $L = 5$ achieves the best performance on both ML-1M and Amazon-Book, while shorter windows provide insufficient transition context and longer windows may introduce less relevant historical signals. This suggests that a moderate recent window is effective for capturing behavioural transitions used by the intent predictor.

4.6.2 Efficiency. We further analyse the computational overhead introduced by intent-aware simulation. As shown in Figure 4c, intent-enhanced prompting increases the average token usage from 4,207 to 4,573 tokens per call, corresponding to an average overhead

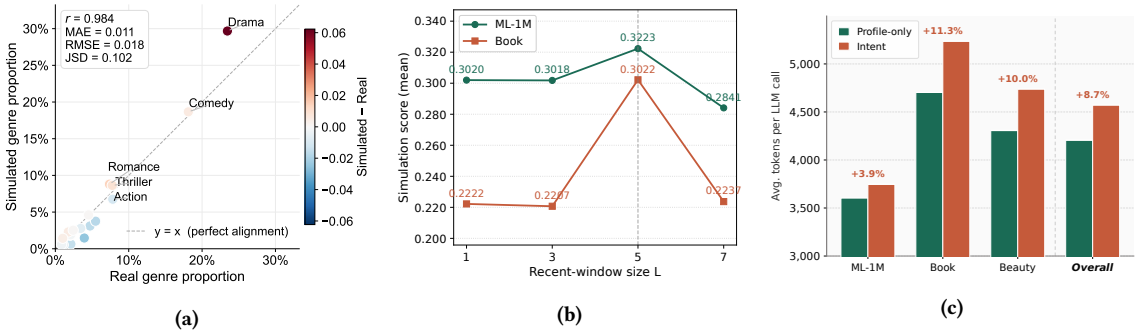


Figure 4: (a) Genre-level interaction distribution in real and simulated data on ML-1M. (b) Sensitivity analysis of recent-state window size L. (c) Average token usage of profile-only and intent-enhanced simulation. Values are averaged across ranking and discrimination tasks within each dataset.

of approximately 8.7%. This increase mainly comes from the additional behavioural control instruction in the prompt. The overhead is moderate relative to the consistent performance gains observed in the main results.

In terms of latency, intent control introduces no measurable runtime overhead over profile-only simulation. The average response time is 4.21s for profile-only prompting and 4.06s for intent-enhanced prompting, with similar median and p95 latency. The slight difference is within normal API response-time variance rather than a systematic speedup or slowdown. This is because the total simulation time is dominated by the LLM API call, while local intent-model inference takes less than 50ms per instance. Therefore, IntRecSim improves simulation quality with only a modest token increase and negligible additional latency.

Overall, the experimental results answer the three research questions as follows. For RQ1, the intent prediction results show that high-level behavioural intent can be inferred from preference-state transitions with moderate but useful accuracy. For RQ2, the main results and ablation study demonstrate that intent-aware control improves simulation alignment beyond profile-only prompting and recent-history conditioning. For RQ3, the gains across multiple profile backbones show that IntRecSim can serve as a plug-in behavioural control layer for different LLM-based user simulations.

5 Limitations and Future Work

This work has several limitations. First, the proposed intent labels are weakly supervised signals derived from interaction histories rather than directly observed user intent. Although this makes the framework applicable to standard recommendation datasets, the constructed labels may contain noise and may not fully capture the complexity of real user decision processes. Future work could incorporate richer feedback signals, user studies, or semi-supervised refinement to obtain more reliable behavioural intent annotations.

Second, we model intent using three coarse decision modes: `stay_profile`, `follow_recent`, and `explore`. Explore is a meaningful but difficult regime; identifying it is possible, but controlling LLM behaviour under exploration remains challenging. Furthermore, while these modes capture important behavioural patterns, real user decisions may involve more fine-grained factors such as context, mood, social influence, price sensitivity, or task-specific goals. Extending the intent space to support more adaptive or continuous behavioural states is a promising direction.

Finally, the framework relies on LLM prompting for user simulation, which can be affected by prompt sensitivity, model choice, and inference cost. Developing more robust prompt designs, confidence-aware control strategies, and efficient distilled simulators would make intent-aware user simulation more scalable and practical.

6 Conclusion

In this work, we proposed **IntRecSim**, a plug-in behavioural control framework for LLM-based user simulation in recommendation. IntRecSim moves beyond static profile-conditioned preference matching by introducing an operational intent signal that guides how the simulator balances long-term preference, recent behavioural evidence, and exploratory behaviour.

We construct this signal from historical interactions using weak pseudo-intent supervision based on hybrid category–semantic preference signals. Rather than claiming to recover latent psychological intent, we treat intent as a practical control mechanism derived from observable preference transitions. Experiments on three public datasets show that IntRecSim improves simulation performance in most settings across baseline backbones and evaluation tasks. Further ablations and intent-wise analyses indicate that the gains are not due to arbitrary prompting and that the constructed intent groups capture meaningful differences in behavioural predictability.

Overall, our results suggest that explicit behavioural control is a promising direction for developing more controllable and realistic LLM-based user simulators. Future work should improve pseudo-intent estimation and evaluate intent-aware simulation in multi-step interactive and downstream recommender-system settings.

Acknowledgments

This work was conducted at the ARC Centre of Excellence for Automated Decision-Making and Society (ADM+S) and funded by the Australian Research Council (CE200100005). It was supported by computing resources from RACE (RMIT Advanced Computing Ecosystem).

GenAI Usage Disclosure

This research complies with the CIKM 2026 GenAI usage policy. The authors disclose the following use of Generative AI (GenAI) tools during the research process:

- **Writing:** ChatGPT (OpenAI) was used to assist in proofreading, rephrasing technical sentences, and improving the clarity of the manuscript. All substantive content, including ideas, methods, results, and analysis, was written and verified by the authors.
- **Code:** No GenAI tools were used to generate any code in this research. All code was developed by the authors.
- **Data:** No GenAI tools were used to generate or augment the data used in this research. All datasets were obtained from publicly available sources as described in the paper.
- **Experiments and Analysis:** No GenAI tools were used to generate experimental results or statistical analyses.

The authors confirm that all intellectual contributions are original and that the use of GenAI tools did not compromise the scientific integrity or originality of the work.

References

[1] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. 2018. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *CoRR* abs/1803.01271 (2018). arXiv:1803.01271 http://arxiv.org/abs/1803.01271

[2] Veronika Bogina, Tsvi Kuflik, Dietmar Jannach, Maria Bielikova, Michal Kompan, and Christoph Trattner. 2022. Considering temporal aspects in recommender systems: a survey. *User Modeling and User-Adapted Interaction* 33, 1 (July 2022), 81–119. doi:10.1007/s11257-022-09335-w

[3] Nicolas Bougie and Narimawa Watanabe. 2025. SimUSER: Simulating User Behavior with Large Language Models for Recommender System Evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, Georg Rehm and Yunyao Li (Eds.). Association for Computational Linguistics, Vienna, Austria, 43–60. doi:10.18653/v1/2025.acl-industry.5

[4] Luyu Chen, Quanyu Dai, Zeyu Zhang, Xueyang Feng, Mingyu Zhang, Pengcheng Tang, Xu Chen, Yue Zhu, and Zhenhua Dong. 2025. RecUserSim: A Realistic and Diverse User Simulator for Evaluating Conversational Recommender Systems. In *Companion Proceedings of the ACM on Web Conference 2025* (Sydney NSW, Australia) *(WWW '25)*. Association for Computing Machinery, New York, NY, USA, 133–142. doi:10.1145/3701716.3715258

[5] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *CoRR* abs/1412.3555 (2014). arXiv:1412.3555 http://arxiv.org/abs/1412.3555

[6] Yashar Deldjoo, Zhankui He, Julian McAuley, Anton Korikov, Scott Sanner, Arnau Ramisa, René Vidal, Maheswaran Sathiamoorthy, Atoosa Kasirzadeh, and Silvia Milano. 2024. A review of modern recommender systems using generative models (gen-recsys). In *Proceedings of the 30th ACM SIGKDD conference on Knowledge Discovery and Data Mining*. 6448–6458.

[7] Chongming Gao, Shijun Li, Wenqiang Lei, Jiawei Chen, Biao Li, Peng Jiang, Xiangnan He, Jiaxin Mao, and Tat-Seng Chua. 2022. KuaiRec: A fully-observed dataset and insights for evaluating recommender systems. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 540–550.

[8] Alexandre Gilotte, Clément Calauzènes, Thomas Nedelec, Alexandre Abraham, and Simon Dollé. 2018. Offline A/B Testing for Recommender Systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (Marina Del Rey, CA, USA) *(WSDM '18)*. Association for Computing Machinery, New York, NY, USA, 198–206. doi:10.1145/3159652.3159687

[9] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (Dec. 2015), 19 pages. doi:10.1145/2827872

[10] Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. 2024. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952* (2024).

[11] Jiahao Liu, Shengkang Gu, Dongsheng Li, Guangping Zhang, Mingzhe Han, Hansu Gu, Peng Zhang, Tun Lu, Li Shang, and Ning Gu. 2025. AgentCF++: Memory-enhanced LLM-based Agents for Popularity-aware Cross-domain Recommendations. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Padua, Italy) *(SIGIR '25)*. Association for Computing Machinery, New York, NY, USA, 2566–2571. doi:10.1145/3726302.3730161

[12] Qidong Liu, Xiangyu Zhao, Yuhao Wang, Yejing Wang, Zijian Zhang, Yuqi Sun, Xiang Li, Maolin Wang, Pengyue Jia, Chong Chen, et al. 2025. Large Language Model Enhanced Recommender Systems: Methods, Applications and Trends. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*. 6096–6106.

[13] Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Chris Leung, Jiajie Tang, and Jiebo Luo. 2024. Llm-rec: Personalized recommendation via prompting large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*. 583–612.

[14] Chenglong Ma, Xinye Wanyan, Danula Hettiachchi, Ziqi Xu, Yongli Ren, and Jeffrey Chan. 2026. Verifiable User Simulation for Search and Recommendation Systems. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Australia) *(SIGIR '26)*. Association for Computing Machinery, New York, NY, USA, 5340–5343. doi:10.1145/3805712.3808645

[15] Chenglong Ma, Ziqi Xu, Yongli Ren, Danula Hettiachchi, and Jeffrey Chan. 2025. PUB: An LLM-Enhanced Personality-Driven User Behaviour Simulator for Recommender System Evaluation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR*. 2690–2694.

[16] Yu Shang, Peijie Liu, Yuwei Yan, Zijing Wu, Leheng Sheng, Yuanqing Yu, Chumeng Jiang, An Zhang, Fengli Xu, Yu Wang, Min Zhang, and Yong Li. 2025. AgentRecBench: Benchmarking LLM Agent-based Personalized Recommender Systems. In *Advances in Neural Information Processing Systems*, D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (Eds.), Vol. 38, Main Conference. Curran Associates, Inc. doi:10.52202/085713-5156

[17] Yunxiao Shi, Wujiang Xu, Zhang Zeqi, Xing Zi, Qiang Wu, and Min Xu. 2025. PersonaX: A recommendation agent-oriented user modeling framework for long behavior sequence. In *Findings of the Association for Computational Linguistics: ACL 2025*. 5764–5787.

[18] Yubo Shu, Haonan Zhang, Hansu Gu, Peng Zhang, Tun Lu, Dongsheng Li, and Ning Gu. 2024. RAH! RecSys–assistant–human: A human-centered recommendation framework with LLM agents. *IEEE Transactions on Computational Social Systems* 11, 5 (2024), 6759–6770.

[19] Alicia Tsai, Adam Kraft, Long Jin, Chenwei Cai, Anahita Hosseini, Taibai Xu, Zemin Zhang, Lichan Hong, Ed H Chi, and Xinyang Yi. 2024. Leveraging llm reasoning enhances personalized recommender systems. In *Findings of the Association for Computational Linguistics: ACL 2024*. 13176–13188.

[20] Marco Valentini. 2025. Cooperative and competitive llm-based multi-agent systems for recommendation. In *European Conference on Information Retrieval*. Springer, 204–211.

[21] Lu Wang, Di Zhang, Fangkai Yang, Pu Zhao, Jianfeng Liu, Yuefeng Zhan, Hao Sun, Qingwei Lin, Weiwei Deng, Dongmei Zhang, Feng Sun, and Qi Zhang. 2025. LettinGo: Explore User Profile Generation for Recommendation System. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2* (Toronto ON, Canada) *(KDD '25)*. Association for Computing Machinery, New York, NY, USA, 2985–2995. doi:10.1145/3711896.3737024

[22] Lei Wang, Jingsen Zhang, Hao Yang, Zhi-Yuan Chen, Jiakai Tang, Zeyu Zhang, Xu Chen, Yankai Lin, Hao Sun, Ruihua Song, et al. 2025. User behavior simulation with large language model-based agents. *ACM Transactions on Information Systems* 43, 2 (2025), 1–37.

[23] Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Yanbin Lu, Xiaojiang Huang, and Yingzhen Yang. 2024. Recmind: Large language model powered agent for recommendation. In *Findings of the Association for Computational Linguistics: NAACL 2024*. 4351–4364.

[24] Xinye Wanyan, Danula Hettiachchi, Chenglong Ma, Ziqi Xu, and Jeffrey Chan. 2025. Temporal-Aware User Behaviour Simulation with Large Language Models for Recommender Systems. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management* (Seoul, Republic of Korea) *(CIKM '25)*. Association for Computing Machinery, New York, NY, USA, 5335–5339. doi:10.1145/3746252.3760878

[25] Xinye Wanyan, Chenglong Ma, Danula Hettiachchi, Ziqi Xu, and Jeffrey Chan. 2026. Task-Aware Automated User Profile Generation for Recommendation Simulation Using Large Language Models. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Australia) *(SIGIR '26)*. Association for Computing Machinery, New York, NY, USA, 1937–1947. doi:10.1145/3805712.3809707

[26] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. 2024. A survey on large language models for recommendation. *World Wide Web* 27, 5 (2024), 60.

[27] Wujiang Xu, Yunxiao Shi, Zujie Liang, Xuying Ning, Kai Mei, Kun Wang, Xi Zhu, Min Xu, and Yongfeng Zhang. 2025. iAgent: LLM Agent as a Shield between User and Recommender Systems. In *Findings of the Association for Computational Linguistics: ACL 2025*, Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (Eds.). Association for Computational Linguistics, Vienna, Austria, 18056–18084. doi:10.18653/v1/2025.findings-acl.928

[28] Xiaochuan Xu, Zeqiu Xu, Peiyang Yu, and Jiani Wang. 2025. Enhancing user intent for recommendation systems via large language models. In *International Conference on Artificial Intelligence and Machine Learning Research (CAIMLR 2024)*, Vol. 13635. SPIE, 46–54.

[29] Ziqi Xu, Chenglong Ma, Yongli Ren, Jeffrey Chan, Wei Shao, and Feng Xia. 2025. Towards Better Evaluation of Recommendation Algorithms with Bi-directional Item Response Theory. In *Companion Proceedings of the ACM on Web Conference 2025* (Sydney NSW, Australia) *(WWW '25)*. Association for Computing Machinery, New York, NY, USA, 1455–1459. doi:10.1145/3701716.3715540

[30] Yuwei Yan, Yu Shang, Qingbin Zeng, Yu Li, Keyu Zhao, Zhiheng Zheng, Xuefei Ning, Tianji Wu, Shengen Yan, Yu Wang, et al. 2025. Agentsociety challenge: Designing llm agents for user modeling and recommendation on web platforms. In *Companion Proceedings of the ACM on Web Conference 2025*. 2963–2967.

[31] Xiaopeng Ye, Chen Xu, Zhongxiang Sun, Jun Xu, Gang Wang, Zhenhua Dong, and Ji-Rong Wen. 2025. Llm-empowered creator simulation for long-term evaluation of recommender systems under information asymmetry. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 201–211.

[32] An Zhang, Yuxin Chen, Leheng Sheng, Xiang Wang, and Tat-Seng Chua. 2024. On Generative Agents in Recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC, USA) *(SIGIR '24)*. Association for Computing Machinery, New York, NY, USA, 1807–1817. doi:10.1145/3626772.3657844

[33] Guangping Zhang, Peng Zhang, Jiahao Liu, Zhuoheng Li, Dongsheng Li, Hansu Gu, Tun Lu, and Ning Gu. 2025. EvalAgent: Towards Evaluating News Recommender Systems with LLM-based Agents. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management* (Seoul, Republic of Korea) *(CIKM '25)*. Association for Computing Machinery, New York, NY, USA, 4086–4095. doi:10.1145/3746252.3761127

[34] Junjie Zhang, Yupeng Hou, Ruobing Xie, Wenqi Sun, Julian McAuley, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2024. AgentCF: Collaborative Learning with Autonomous Language Agents for Recommender Systems. In *Proceedings of the ACM Web Conference 2024* (Singapore, Singapore) *(WWW '24)*. Association for Computing Machinery, New York, NY, USA, 3679–3689. doi:10.1145/3589334.3645537

[35] Shuo Zhang and Krisztian Balog. 2020. Evaluating Conversational Recommender Systems via User Simulation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 1512–1520. doi:10.1145/3394486.3403202

[36] Shengzhe Zhang, Liyi Chen, Dazhong Shen, Chao Wang, and Hui Xiong. 2025. Hierarchical time-aware mixture of experts for multi-modal sequential recommendation. In *Proceedings of the ACM on Web Conference 2025*. 3672–3682.

[37] Zijian Zhang, Shuchang Liu, Ziru Liu, Rui Zhong, Qingpeng Cai, Xiangyu Zhao, Chunxu Zhang, Qidong Liu, and Peng Jiang. 2025. LLM-powered user simulator for recommender system. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence (AAAI'25/IAAI'25/EAAI'25)*. AAAI Press, Article 1483, 9 pages. doi:10.1609/aaai.v39i12.33456

[38] Yurou Zhao, Yingfei Zhang, Quan Zhou, Shuang Zhang, Wei Lin, and Jiaxin Mao. 2025. Adapting LLMs for Personalized Evaluation of Explanations for Recommendations: A Meta-Learning Approach based on MAML. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management* (Seoul, Republic of Korea) *(CIKM '25)*. Association for Computing Machinery, New York, NY, USA, 4390–4400. doi:10.1145/3746252.3761020

[39] Lixi Zhu, Xiaowen Huang, and Jitao Sang. 2025. A llm-based controllable, scalable, human-involved user simulator framework for conversational recommender systems. In *Proceedings of the ACM on Web Conference 2025*. 4653–4661.

[40] Xi Zhu, Yu Wang, Hang Gao, Wujiang Xu, Chen Wang, Zhiwei Liu, Kun Wang, Mingyu Jin, Linsey Pang, Qingsong Weng, et al. 2025. Recommender systems meet large language model agents: A survey. *Foundations and Trends® in Privacy and Security* 7, 4 (2025), 247–396.