LEARNING USER REPRESENTATIONS TO IMPROVE SIMULATION FIDELITY

Anonymous authors
Paper under double-blind review

ABSTRACT

User simulators enable interactive training and evaluation of AI assistants at scale. Yet current simulators struggle to reproduce the diverse behaviors of real users, leading to assistants that generalize poorly to underrepresented users. We identify user representations as an underexplored bottleneck in user simulation. User representations are natural-language descriptions that condition simulators to reproduce a particular user's behavior. A representation's quality becomes apparent in hindsight, through the behavior it induces in a simulator. To evaluate and optimize user representations, we formulate a conversation reconstruction task: given an original conversation, we generate a representation, condition a simulator on it, simulate a conversation, and then measure fidelity by comparing user behavior in the original and simulated conversations. This comparison also yields *hindsight feedback*, which identifies where the representation falls short and how it can be improved. We introduce *hindsight distillation*, an online training procedure that distills feedback-conditioned improvements into the representation generator, enabling it to generate improved representations directly from the original conversation. We present `UserRepGen-4B`, a representation generator trained with hindsight distillation. Across 700 conversations spanning seven datasets, hindsight feedback helps improve user representations at test time, and `UserRepGen-4B` generates representations that enable more faithful simulation than those generated by GPT-5.5. Beyond conversation reconstruction, learned representations improve simulation fidelity for unseen tasks, better preserve the population-level diversity of real users, and better distinguish users across conversations.

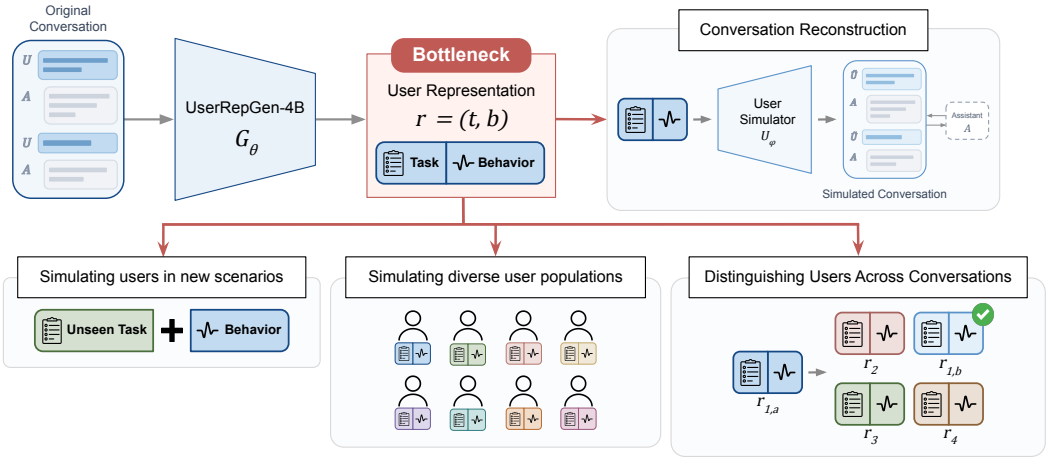


Figure 1: **User representations are a bottleneck in user simulation.** They capture the information a simulator needs to faithfully reproduce a user's behavior. We train `UserRepGen-4B` to generate high-quality user representations and demonstrate their importance for reconstructing user conversations, simulating users in new scenarios, simulating diverse user populations, and distinguishing users across conversations.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Simulating user behavior is central to the development of human-centered AI assistants. User simulators provide scalable environments in which AI assistants can be trained (Wu et al., 2025; Zhou et al., 2024b) and evaluated (Yao et al., 2024; Zhou et al., 2024a) on the kinds of interactions they encounter in the real world. Yet current simulators struggle to faithfully simulate diverse users and instead demonstrate homogeneous behaviors that tend to resemble those of a generic user (Mehri et al., 2026b; Xiao et al., 2026). Reliance on these simulators exposes assistants to a narrow range of behaviors during development, leading to poor generalization to users that are underrepresented in simulation (Yu et al., 2026; Liu et al., 2023). It is therefore essential that user simulators capture this diversity and reproduce the behaviors of individual users with high fidelity.

To simulate a particular user, a simulator is conditioned on a user representation and then used to generate responses. A user representation is a natural-language description intended to capture the information needed to reproduce a user's behavior. When relevant information is missing, simulators fill in the gaps with assumptions about how the user might behave, often resulting in generic behaviors rather than the particular user's behavior. Existing works that improve the simulation pipeline primarily focus on optimizing the user simulator to generate more realistic responses, while representation generation receives comparatively little attention and is typically done with an ad hoc prompt and a general-purpose large language model (LLM) (Naous et al., 2026; Wu et al., 2026; Zhou et al., 2026). In this work, we identify user representations as an underexplored bottleneck in user simulation, and optimize the representation generator for downstream simulation fidelity while keeping the user simulators fixed.

The quality of a user representation becomes apparent in hindsight through the behavior it induces in a simulator. We operationalize this with a conversation reconstruction task: we generate a user representation from a user–assistant conversation, condition a fixed simulator on it, and have the simulator interact with an assistant to produce a simulated conversation. We evaluate the representation by measuring how faithfully the simulator reproduces the user behavior from the original conversation. Comparing the two conversations also yields *hindsight feedback*, a description of where the representation falls short and how it can be improved. The representation generator can be conditioned on this feedback to generate an improved representation. Collecting hindsight feedback, however, requires expensive multi-turn simulations. We therefore introduce *hindsight distillation*, an online training procedure in which we distill feedback-conditioned improvements into the representation generator. At each training step, we sample a representation, improve it with hindsight feedback, and then use it as a supervised target. This teaches the representation generator to produce better representations directly from the original conversation.

We present `UserRepGen-4B`, a representation generator trained with hindsight distillation, and evaluate it on 700 conversations spanning seven datasets of user–AI interactions. `UserRepGen-4B` generates representations that enable more faithful simulation than those generated by proprietary models such as GPT-5.5, without requiring hindsight feedback at inference time. Improving representations with hindsight feedback also improves simulation fidelity at test time, with further gains when repeated for multiple iterations. Beyond conversation reconstruction, the learned representations improve simulation fidelity for unseen tasks, produce less homogeneous simulations that better preserve the population-level diversity of real users, and capture user-specific behaviors that better distinguish users across conversations.

The main contributions of our work can be summarized as follows:

- **User representations and conversation reconstruction.** We identify user representations as an underexplored bottleneck in user simulation and formulate conversation reconstruction as a task for evaluating and training representation generators (§ 3).
- **Structured user representations.** We structure a user representation into separate task and behavior descriptions, enabling a user's behavior description to be paired with unseen tasks or different behavior descriptions to be paired with a fixed task (§ 3).
- **Hindsight distillation.** We introduce *hindsight distillation*, an online training procedure in which we train a representation generator using *hindsight feedback*, which describes how to improve user representations by comparing real and simulated user behavior. (§ 4 & § 5).
- **`UserRepGen-4B`.** We present `UserRepGen-4B`, a representation generator trained with hindsight distillation, whose representations enable more faithful simulation that those gener-

ated by proprietary models such as GPT-5.5 (§ 6.3). Improving representations with hindsight feedback also improves fidelity at test time, and can be applied for multiple iterations (§ 6.2).

- **Characterizing learned representations.** We present a systematic analysis of how user representations change after being optimized for simulation fidelity (§ 6.4).
- **Broader utility of learned representations.** Beyond conversation reconstruction, the learned representations improve simulation fidelity on unseen tasks (§ 7.1), produce less homogeneous simulations that preserve the population-level diversity of real users (§ 7.2), and better distinguish users across conversations (§ 7.3).

2 Related Work

User simulation. LLMs have emerged as promising tools for simulating human behavior (Park et al., 2023; Yao et al., 2024; Anthis et al., 2025; Mehri et al., 2026a). Yet prompting general-purpose LLMs has revealed several limitations, motivating efforts to train specialized simulators with objectives such as profile consistency, goal alignment, latent-state alignment, and realism (Wang et al., 2025; Mehri et al., 2026c; Naous et al., 2026; Wu et al., 2026; Zhou et al., 2026; Wang et al., 2026c). Most recently, studies have documented behavioral homogenization among simulated users and the resulting mismatch with real user populations (Mehri et al., 2026b; Xiao et al., 2026). Our work addresses this problem by conditioning simulators on learned user representations that improve simulation fidelity.

User representation. Early works introduce personas to enable assistants to generate responses personalized to and consistent with different individuals (Li et al., 2016; Zhang et al., 2018). User simulation has explored several forms of user representation, including describing user goals (Schatzmann et al., 2007; Naous et al., 2026), demographic or biographical attributes (Argyle et al., 2023), interview transcripts (Park et al., 2024), and learned vector representations (Li et al., 2016; Ning et al., 2025; Chen et al., 2025). User representations are central to simulation reliability, directly shaping the behaviors that simulators produce (Li et al., 2025). To improve user representations, recent work has explored reducing underspecification (Lin et al., 2026), scaling generation (Ge et al., 2025), generating diverse populations of personas (Chopra et al., 2026), and matching the distribution of behaviors in a reference population (Liu et al., 2026b). We learn to generate user representations that capture the user-specific information necessary for faithful simulation.

Learning from privileged information. Recent work incorporates environmental feedback into model training through reinforcement learning and self-distillation (Song et al., 2026; Shi et al., 2026; Sun et al., 2026), often by distilling from a teacher conditioned on privileged information (Zhao et al., 2026; Hübotter et al., 2026; Mendes et al., 2026). For interactive agents, privileged information is obtained from multi-turn rollouts and used to improve the agent (Buening et al., 2026; Wang et al., 2026a;b; Yeo et al., 2026; Li et al., 2026; Liu et al., 2026a). In contrast, our setting uses the simulator behavior induced by a representation in a multi-turn conversation as privileged information for improving the representation generator.

3 Task Description

Let $\mathcal{C} = (u_1, a_1, \ldots, u_n, a_n)$ be a conversation between a real user and an assistant, where u_i and a_i denote the user and assistant utterances at turn i, respectively. The goal of user simulation is to generate responses that reproduce the behavior exhibited by the user in $\mathcal{C}$. A typical pipeline consists of two stages. First, a representation generator G_θ generates a user representation r from $\mathcal{C}$, which is a natural language description that captures the information needed to simulate the user. Second, an LLM-based user simulator U_ϕ is conditioned on r and interacts with an assistant to generate a simulated conversation $\tilde{\mathcal{C}} = (\tilde{u}_1, \tilde{a}_1, \ldots, \tilde{u}_m, \tilde{a}_m)$.

Recent work has largely focused on the second stage, optimizing the user simulator U_ϕ to generate responses that resemble real humans (Naous et al., 2026; Wu et al., 2026; Zhou et al., 2026). By contrast, representation generation has received comparatively little attention and is typically performed using a general-purpose LLM and an ad hoc prompt. The representations may fail to capture all relevant information about a user's behavior, which can cause unwanted behaviors from U_ϕ.

For example, the simulator may fill in the gaps with assumptions about the user, and demonstrate generic behaviors rather than those of the particular user in $\mathcal{C}$. This motivates treating representation generation as its own learning problem and optimizing G_θ for downstream simulation fidelity.

User Representations. One approach for representing users is through personas that describe demographic or biographical attributes (Anthis et al., 2025), such as age, occupation, location, experiences, interests, personality, or background (Zhang et al., 2018; Ge et al., 2025). These attributes provide only indirect cues about user behavior, leaving the simulator to fill in the gaps with potentially unsupported or stereotypical assumptions. Another approach represents users through an intent that describes their overall goal without specifying any behaviors (Naous et al., 2026). This may suffice for simulating a generic user pursuing a given task, but it does not capture the unique behaviors of individual users, which can result in homogeneous simulations that fail to reflect the diversity of real users (Mehri et al., 2026b; Xiao et al., 2026).

In this work, we introduce a structured user representation $r = (t, b)$ that separates the user's conversation-specific task from their behavioral patterns that persist across conversations. The task t describes what the user is trying to accomplish in $\mathcal{C}$, and behavior b describes the user's general behavior, such as how they formulate and specify requests, respond to the assistant, and provide feedback. This structure enables composing representations, either by simulating diverse user behaviors on a fixed task, or in simulating multiple tasks with a fixed user behavior.

Conversation Reconstruction. We train and evaluate representation generators through a conversation reconstruction task. Given a conversation $\mathcal{C}$, we measure how well a user representation r enables user simulators to generate conversations that preserve the user behavior observed in $\mathcal{C}$.

We formalize this task with a simulation environment $e = (U_e, A_e, J_e)$, consisting of a fixed user simulator U_e, assistant A_e, and judge J_e. For each simulation environment, the simulator U_e is conditioned on r and interacts with the assistant A_e to produce a simulated conversation $\tilde{\mathcal{C}}_e = (\tilde{u}_1, \tilde{a}_1, \ldots, \tilde{u}_m, \tilde{a}_m)$. The interaction ends when U_e is satisfied and emits a termination signal, or when the conversation reaches a maximum length. The judge J_e compares $\tilde{\mathcal{C}}_e$ with $\mathcal{C}$ and produces a scalar score $s_e \in \mathbb{R}$ that measures simulation fidelity across multiple dimensions (see Appendix C). To account for differences in simulator behavior caused by different assistant responses, J_e evaluates fidelity based on the given interaction context rather than requiring an exact reconstruction.

Our goal is to train G_θ to generate representations that maximize fidelity across fixed simulators.

4 Hindsight Feedback

A user representation may appear reasonable without capturing all the relevant information needed to reproduce a user's behavior. Its quality becomes apparent in hindsight, after it is used to simulate the user. The behaviors that the simulator reproduces faithfully indicate what the representation captures well, whereas the behaviors it reproduces poorly or fails to reproduce reveal what the representation does not capture well. We describe these in natural language with *hindsight feedback*.

Given a conversation $\mathcal{C}$ and an initial user representation $r^{(0)}$, we collect hindsight feedback for the representation using an ensemble of simulation environments $\mathcal{E} = \{e_k = (U_k, A_k, J_k)\}_{k=1}^{K}$, each built on a different underlying model. For each environment e_k, we roll out a simulated conversation:

$$\tilde{\mathcal{C}}_k \sim \text{rollout}\Big(U_k(\cdot \mid r^{(0)}), A_k\Big)\,. \tag{1}$$

The judge J_k then compares the user behavior in $\tilde{\mathcal{C}}_k$ with that in $\mathcal{C}$ and generates hindsight feedback f_k. We collect the feedback from all simulation environments into $f = (f_1, \ldots, f_K)$, where

$$f_k \sim J_k\Big(\cdot \mid \mathcal{C}, \tilde{\mathcal{C}}_k\Big)\,, \qquad k = 1, \ldots, K. \tag{2}$$

Improving user representations with hindsight feedback. We leverage the in-context learning capabilities of LLMs to improve a user representation. We generate an improved representation by conditioning G_θ on the conversation $\mathcal{C}$, the current representation r, and the hindsight feedback f. We repeat this improvement for I iterations, as each iteration may expose previously unseen failures

or reveal that earlier improvements were insufficient. At each round i, we collect feedback $f^{(i)}$ for $r^{(i)}$ and generate:

$$r^{(i+1)} \sim G_\theta\Big(\cdot \mid \mathcal{C}, r^{(i)}, f^{(i)}\Big), \tag{3}$$

The final user representation is denoted by $r^{(I)}$.

5 Hindsight Distillation

Hindsight feedback enables G_θ to improve a user representation, but obtaining this feedback requires simulating and evaluating a multi-turn conversation. To avoid this during inference, we distill the feedback-conditioned improvements into G_θ, enabling it to generate improved representations directly from $\mathcal{C}$.

At training step n, the current policy samples an initial user representation for a conversation $\mathcal{C}$

$$r^{(0)} \sim G_{\theta_n}(\cdot \mid \mathcal{C}). \tag{4}$$

We apply I rounds of hindsight feedback to obtain $r^{(I)} = (r_1^{(I)}, \ldots, r_L^{(I)})$, which we use as a training target under the negative log-likelihood objective:

$$\mathcal{L}(\theta_n) = -\frac{1}{L}\sum_{j=1}^{L} \log G_{\theta_n}\left(r_j^{(I)} \mid \mathcal{C}, r_{<j}^{(I)}\right). \tag{5}$$

This distills hindsight feedback-conditioned improvements into $G_\theta(\cdot \mid \mathcal{C})$, teaching it to generate better representations without the hindsight feedback. The complete training procedure is depicted in Figure 2 and Algorithm 1 in Appendix A.

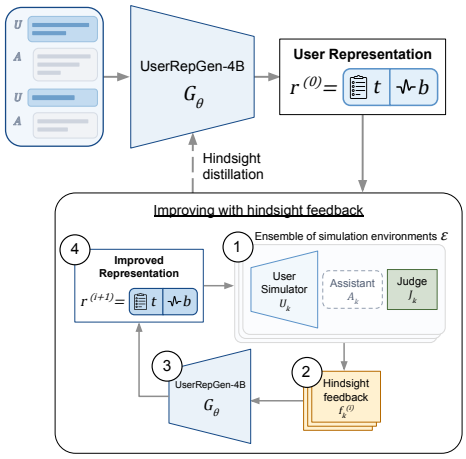


Figure 2: Hindsight distillation for G_θ. A representation is sampled, and is improved with I rounds of hindsight feedback to get $r^{(I)}$, which serves as a training target.

6 Learning User Representations

6.1 Experimental Setup

Datasets. We use five datasets of real-world conversations with AI assistants: WildChat (Zhao et al., 2024), LMSYS-Chat-1M (Zheng et al., 2024), PRISM (Kirk et al., 2024), SWE-Chat (Baumann et al., 2026), and ThoughtTrace (Jin et al., 2026). We additionally use two datasets of simulated conversations: WildChat Coding and WildChat Writing (Mehri et al., 2026b). For each dataset, we apply a preprocessing stage that removes near-duplicate conversations based on n-gram overlap, and retains safe English conversations with at least four user turns.

Training. We use a fixed set of 3,200 conversations, approximately 500 conversations from each dataset and 200 from ThoughtTrace. Hindsight feedback is generated using an ensemble of two simulation environments based on Qwen3.5-4B (Team, 2026) and Gemma3-4B (Team et al., 2025). Within each environment, the corresponding model serves as the user simulator, assistant and judge. During training, the judges compare only user utterances from the real and simulated conversations, which we found to be a reliable proxy for the full-conversation evaluation when using smaller and less capable judge models. Additionally, we stabilize training by using the initial policy as our teacher policy (Zhao et al., 2026) and to generate improved user representations.

Evaluation. We sample 100 conversations from each dataset, yielding 700 conversations in total. We follow the evaluation process described in Section 3, using GPT-5.4 mini as the user simulator, assistant, and judge. We specifically use a different model than training to ensure a more reliable and unbiased evaluation that tests whether the learned representations generalize to unseen simulation environments. The judge compares user behavior in simulated conversations with that in the corresponding original conversation and assigns an overall fidelity score from 1 to 10. We validate

the judge with a human preference study in Appendix B. Lastly, simulators sometimes demonstrated failures in staying aligned with their user representations (Mehri et al., 2026c). To account for this, we simulate three conversations for each representation, and report *best-of-1* and *best-of-3* fidelity. All prompts are provided in Appendix C and hyperparameters are reported in Appendix D.

6.2 Test-Time Improvement with Hindsight Feedback

We investigate whether hindsight feedback improves user representations directly at test time. For each evaluation conversation, we generate an initial user representation $r^{(0)}$ using Qwen3.5-4B. Using the ensemble $\mathcal{E}$ of Qwen3.5-4B and Gemma3-4B environments, we apply $I = 10$ rounds of improvement using hindsight feedback, as described in Section 4. We evaluate representations after every iteration (Figure 3).

Hindsight feedback helps generate better user representations at test-time, which leads to higher simulation fidelity scores. After one round of feedback, best-of-1 fidelity increases from 3.41 to 4.93, and best-of-3 increases from 4.42 to 6.14. Performance continues to improve over subsequent rounds, with best-of-1 reaching 6.10 after ten iterations and best-of-3 peaking at 7.07 at nine. Most gains occur in the first two rounds, and performance largely plateaus after six rounds.

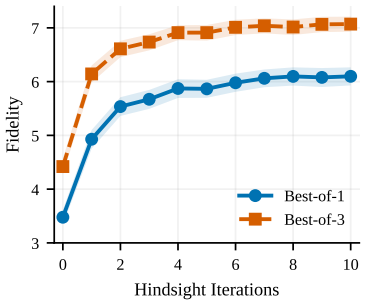


Figure 3: Simulation fidelity across iterations of hindsight feedback. Shaded regions show 95% bootstrap confidence intervals over conversations.

6.3 Hindsight Distillation

We then distill hindsight feedback into the representation generator to generate better representations directly from a conversation. Unless otherwise stated, hindsight feedback is collected using the ensemble of simulation environments described in Section 6.1. We explore four training approaches:

- **Supervised Fine-Tuning (SFT):** For each conversation $\mathcal{C}$ in the training set, the base policy samples an initial user representation $r^{(0)} \sim G_{\theta_0}(\cdot \mid \mathcal{C})$. We collect hindsight feedback and generate an improved user representation $r^{(1)}$. We then train the representation generator to predict $r^{(1)}$ directly from $\mathcal{C}$ using the standard negative log-likelihood objective.
- **Group Relative Policy Optimization (GRPO) (Shao et al., 2024):** For each conversation $\mathcal{C}$, the current policy samples a group of n user representations $\{r_i\}_{i=1}^{n}$. We assign each group to one of the simulation environments $e = (U_e, A_e, J_e)$ and roll out a simulated conversation $\tilde{\mathcal{C}}_{e,i} \sim \text{rollout}(U_e(\cdot \mid r_i), A_e)$ for each representation r_i. To compute the reward, the judge assigns a scalar score $s_i = J_e(\mathcal{C}, \tilde{\mathcal{C}}_{e,i})$ measuring simulation fidelity. We compute group-relative advantages and update the policy using the clipped GRPO objective.
- **On-Policy Self-Distillation (OPSD) (Zhao et al., 2026):** The current policy samples an initial user representation $r^{(0)} \sim G_\theta(\cdot \mid \mathcal{C})$. Hindsight feedback f is collected, and the teacher policy is conditioned on the privileged context $(\mathcal{C}, r^{(0)}, f)$, while the student policy is conditioned only on $\mathcal{C}$. We update the student policy by minimizing the token-level forward KL divergence from the teacher distribution to the student distribution along $r^{(0)}$.
- **Online Supervised Fine-Tuning (OSFT):** As described in Section 5, the current policy samples an initial user representation $r^{(0)} \sim G_\theta(\cdot \mid \mathcal{C})$ and receives hindsight feedback f. Rather than using token-level supervision from the teacher policy, we condition the policy on $(\mathcal{C}, r^{(0)}, f)$ to generate an improved representation $r^{(1)}$. The policy is updated to predict $r^{(1)}$ directly from $\mathcal{C}$ using the standard negative log-likelihood objective.

Results for training Qwen3.5-4B using the different approaches are reported in Table 1. OSFT performs best, achieving 4.95 best-of-1 and 6.02 best-of-3. We refer to this model as `UserRepGen-4B`. OPSD achieves 4.46 best-of-1 and 5.60 best-of-3. Like OSFT, OPSD samples user representations from the current policy and collects hindsight feedback online. However, OPSD distills this feedback through dense token-level supervision, whereas OSFT generates an improved representation and uses it as a supervised target. The stronger performance of OSFT suggests that explicitly supervising on an improved representation is more effective for this task. SFT simi-

larly trains on improved user representations using negative log-likelihood, but generates its targets only once from the base policy rather than continually doing so using the latest policy. It scores of 4.64 best-of-1 and 5.68 best-of-3, therefore supporting prior findings and showing that an online approach for collecting feedback and improving representations is more effective (Ross et al., 2011; Agarwal et al., 2024; Xiong et al., 2024; Shao et al., 2024).

GRPO achieves 3.57 best-of-1 and 4.59 best-of-3, yielding only modest improvements over the base model. Two factors may help explain this result. First, scalar rewards provide limited information about what a user representation omitted or how it should be improved (Song et al., 2026; Hübotter et al., 2026; Zhao et al., 2026; Sun et al., 2026). This limitation is especially costly in our setting because computing each reward requires an expensive multi-turn simulation. Second, we observe low within-group reward diversity during training, with a median reward variance of 0.029 (mean 0.048). This can reduce the learning signal, and when all samples receive identical rewards, their advantages are zero and have no gradient signal (Hübotter et al., 2026; Zhao et al., 2026).

Table 1: Simulation fidelity across user representation generators. We report Best-of-1 and Best-of-3 scores. Subscripts indicate absolute improvements over the Qwen3.5-4B, and darker blue indicates stronger scores.

Model	Best-of-1	Best-of-3
Qwen3.5-4B	3.41	4.42
+ GRPO	$3.57_{\uparrow 0.16}$	$4.59_{\uparrow 0.17}$
+ SFT	$4.64_{\uparrow 1.23}$	$5.68_{\uparrow 1.26}$
+ OPSD	$4.46_{\uparrow 1.05}$	$5.60_{\uparrow 1.18}$
+ OSFT	$4.95_{\uparrow 1.54}$	$6.02_{\uparrow 1.60}$
Qwen3.5-9B	$3.72_{\uparrow 0.31}$	$4.68_{\uparrow 0.26}$
Qwen3.5-27B	$3.79_{\uparrow 0.38}$	$4.83_{\uparrow 0.41}$
Qwen3.5-122B-A10B	$3.53_{\uparrow 0.12}$	$4.55_{\uparrow 0.13}$
GPT-5.4 mini	$3.68_{\uparrow 0.27}$	$4.71_{\uparrow 0.29}$
GPT-5.5	$3.90_{\uparrow 0.49}$	$4.90_{\uparrow 0.48}$

General-purpose LLMs struggle to generate user representations. We also investigate whether stronger general-purpose LLMs generate better user representations without training and report results in Table 1. Despite substantial differences in model scale and general capability, all models perform with a similar range. GPT-5.5, the strongest off-the-shelf model, outperforms Qwen3.5-4B by only 0.49 best-of-1 and 0.48 best-of-3 points. In contrast, `UserRepGen-4B` surpasses GPT-5.5 and raises scores by 1.05 and 1.12 points. These results suggest that user representation generation depends less on general model capability and more on explicitly learning what information is important for downstream simulation fidelity.

6.4 How Does Training Change User Representations?

We analyze how user representations change when optimized for downstream simulation fidelity. Quantitatively, we measure their average length and ROUGE-L precision (Lin, 2004), which measures lexical overlap with the source conversation. Table 2 shows that the representation lengths stay within the same general range, with slight increase in the behavior descriptions. ROUGE-L precision decreases slightly, indicating that improvements in fidelity do not come from simply copying from the source conversation.

Table 2: Quantitative analysis of user representations. We report the average word counts of the task, behavior and complete user representations, as well as ROUGE-L precision against the source conversation.

Model	Task	Behavior	Total	R-L (%)
Qwen3.5-4B	61	101	163	27.2
+ GRPO	61	101	162	27.8
+ SFT	58	126	185	25.8
+ OPSD	72	111	183	25.3
+ OSFT	62	148	210	23.9

We next qualitatively analyze the task and behavior descriptions. After training, task descriptions become more standardized and comprehensive, more consistently capturing constraints introduced in later turns, shifts in direction, and multiple objectives pursued within a single conversation. Behavior descriptions become more specific and actionable. Rather than relying on broad characterizations such as "goal-oriented" or "directive", they describe observable patterns that a simulator can reproduce. They also use more negative and contrastive instructions, such as specifying that a user does not provide all constraints upfront but instead introduces them incrementally. Lastly, the descriptions capture dimensions that were previously overlooked, including how users respond to the assistant, formulate requests, communicate, provide context, and engage across turns. Appendix E uses GPT-5.4 mini to analyze these observations and provides qualitative examples.

7 Broader Utility of Learned Representations

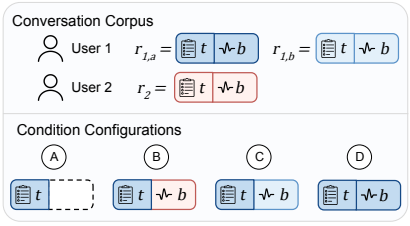
Conversation reconstruction provides an objective for learning and evaluating user representations based on simulation fidelity. We next demonstrate their broader utility beyond reconstruction.

7.1 Simulating Users in New Scenarios

One use case for user simulators is to use evidence from a user's prior interactions to simulate how they might behave in new scenarios. Our primary evaluation, however, evaluates a representation through reconstruction of the same conversation from which it was generated. We therefore evaluate how faithfully a behavior description enables user simulation for unseen tasks.

We select all 147 users from ThoughtTrace who have at least two conversations and do not appear in our training set. For each user, let $\mathcal{C}$ denote the target conversation and $\mathcal{C}'$ another conversation from the same user. We generate representations $r = (t, b) \sim G_\theta(\mathcal{C})$ and $r' = (t', b') \sim G_\theta(\mathcal{C}')$. We follow Section 6.1 and use GPT-5.4 mini as the user simulator, assistant and judge. We condition it under four configurations while keeping task description t fixed:

Ⓐ **Same task only:** Task t from $\mathcal{C}$ without a behavior description; Ⓑ **Same task, different user, different conversation:** Task t and behavior $\bar{b}$ from a randomly selected conversation by a different user; Ⓒ **Same task, same user, different conversation:** Task t and behavior b' from the same user's other conversation $\mathcal{C}'$; Ⓓ **Same task, same user, same conversation:** Task t and behavior b from $\mathcal{C}$.



Model	Ⓐ	Ⓑ	Ⓒ	Ⓓ
Qwen3.5-4B	4.02	$4.16_{\uparrow 0.14}$	$4.17_{\uparrow 0.01}$	$4.68_{\uparrow 0.51}$
+ GRPO	3.95	$4.23_{\uparrow 0.28}$	$4.29_{\uparrow 0.06}$	$4.65_{\uparrow 0.36}$
+ SFT	4.01	$5.37_{\uparrow 1.36}$	$5.70_{\uparrow 0.33}$	$6.35_{\uparrow 0.65}$
+ OPSD	4.25	$5.20_{\uparrow 0.95}$	$5.51_{\uparrow 0.31}$	$5.96_{\uparrow 0.45}$
+ OSFT	4.16	$5.50_{\uparrow 1.34}$	$5.80_{\uparrow 0.30}$	$6.59_{\uparrow 0.79}$

Figure 4: Simulation fidelity when conditioned under configurations Ⓐ, Ⓑ, Ⓒ, and Ⓓ. Each cell reports Best-of-3 scores. Subscripts indicate absolute improvement relative to the condition immediately to the left.

For each configuration, the simulator interacts with the assistant to generate a simulated conversation $\tilde{\mathcal{C}}$. The judge compares the user behavior in $\tilde{\mathcal{C}}$ with that in $\mathcal{C}$ and assigns a fidelity score. We generate three roll outs per configuration and report best-of-3 fidelity in Figure 4.

Ⓑ ≫ Ⓐ: Behavior descriptions encourage realistic simulation. Adding a behavior description improves fidelity even when it is generated from a different user's conversation. The descriptions provide concrete guidance, such as how a user formulates requests or responds to the assistant. This helps the simulator avoid defaulting to assistant-like behaviors and be more representative of real users in general, despite not identifying the behavior of the particular user being simulated. The larger gain for trained representation generators indicates more effective behavior descriptions.

Ⓒ > Ⓑ: Behavior descriptions improve simulation for unseen tasks. Condition Ⓒ tests whether behavior extracted from one of a user's conversation enables faithful simulation of the same user in another conversation with an unseen task. Ⓒ outperforms Ⓑ for every trained representation generator, providing evidence that the descriptions capture user-specific behaviors that generalize to new scenarios. The improvements are modest, however, and remain well below Ⓓ, an upper bound using behavior extracted directly from the target conversation.

7.2 Preserving Behavioral Diversity at the Population Level

We next evaluate user representations at the population level, asking whether the learned representations produce less homogeneous simulations that better preserve the behavioral diversity of the original conversations. We measure the distributional gap between original and simulated user behaviors using the approach proposed by Mehri et al. (2026b): (1) Qwen3.5-122B-A10B-FP8 (Team, 2026) describes the user behavior in each original and simulated conversation along six dimensions; (2) Qwen3-Embedding-8B (Zhang et al., 2025) embeds the resulting descriptions; (3) the embeddings are quantized using k-means into $k = 500$ clusters; (4) the cluster assignments for each conversation are used to construct discrete distributions for the original and simulated user behaviors; and (5) the two distributions are compared using Jensen–Shannon divergence (JS). See Appendix F for details.

For each representation generator, we generate a user representation for each of the 700 evaluation conversations. Using GPT-5.4 mini as both the user simulator and assistant, we produce a corresponding simulated conversation and measure the distributional gap. We additionally report Self-BLEU (Zhu et al., 2018), which measures the lexical overlap between each simulated conversation with the other simulated conversations. We conduct this evaluation under two configurations: conditioning the simulator on only the generated task description and conditioning it on both the task and behavior descriptions. Results are reported in Table 3.

Table 3: Distributional gap between original and simulated user behaviors for our different representation generators when the simulator is conditioned on the task description alone or both the task and behavior descriptions. We report (JS) divergence and Self-BLEU (SB). Darker shades of blue and lower values indicate better performance.

	Task Only		**Task + Behavior**	
Model	JS ↓	SB ↓	JS ↓	SB ↓
Qwen3.5-4B	0.389	0.257	0.315	0.227
+ GRPO	0.356	0.256	0.338	0.217
+ SFT	0.353	0.264	0.270	0.196
+ OPSD	0.326	0.239	0.308	0.202
+ OSFT	0.354	0.246	0.273	0.192

Across both Task Only and Task + Behavior, the trained representation generators generally achieve lower JS divergence and Self-BLEU than the base model. Moreover, conditioning the simulator on both the task and behavior descriptions instead of just on the task reduces both metrics for every representation generator. Lower JS divergence indicates closer alignment between original and simulated distributions of user behavior, and lower Self-BLEU indicates less lexical homogeneity among the simulated conversations. Together, these results show that learned representations better preserve behavioral diversity at the population level.

7.3 Distinguishing Users Across Conversations

Finally, we consider the task of distinguishing users across conversations. We frame this as a multiple-choice task: given a query and four candidate conversations, the goal is to select the candidate from the same user as the query. We use the same 147 ThoughtTrace users from Section 7.1. For each user, one conversation serves as the query and their other conversation serves as the correct candidate. The remaining candidates are conversations from randomly selected other users. We ask GPT-5.4 mini to select the correct candidate and report accuracy. Random selection would yield 25% accuracy, whereas GPT-5.4 mini achieves 44.22% accuracy.

Table 4: GPT-5.4 mini accuracy at distinguishing users across conversations. Subscripts indicate performance compared to the full conversation baseline, and darker blue indicates higher accuracy.

Representation Generator	**Accuracy (%)**
Full conversation (baseline)	44.22
Qwen3.5-4B	$38.78_{\downarrow 5.44}$
+ GRPO	$41.50_{\downarrow 2.72}$
+ SFT	$43.54_{\downarrow 0.68}$
+ OPSD	$50.34_{\uparrow 6.12}$
+ OSFT	$51.02_{\uparrow 6.80}$

We next ask whether user representations can improve performance. For each representation generator, we replace every conversation with its generated behavior description and ask GPT-5.4 mini to solve the same task. Table 4 reports the results. Representations generated by the base model yield the lowest accuracy at 38.78%, 5.44% under the full conversation baseline. GRPO and SFT narrow the gap, but also remain below the full conversation baseline. OPSD and OSFT achieve the highest accuracies at 50.34% and 51.02%, exceeding the full conversation baseline by 6.12% and 6.80%. These results demonstrate that the learned representations capture user-specific characteristics that distinguish a user from other users and persist across conversations.

8 Conclusion

User representations are a bottleneck in user simulation. A simulator conditioned on a representation can only reproduce the behaviors that the representation captures. When relevant behaviors are missing, the simulator fills in the gaps with assumptions, often producing generic rather than user-specific behavior. We therefore present `UserRepGen-4B`, a representation generator that we optimize for downstream simulation fidelity using hindsight distillation. Our results show that learned representations enable more faithful simulations of a particular user, improve simulation fidelity for unseen tasks, better preserve the population-level diversity of real users, and better distinguish users across conversations. Simulation fidelity therefore depends not only on the ability to generate realistic user behavior, but also on capturing the relevant information with user representations.

AI USE STATEMENT

In this work, generative AI tools were used to help with writing code and implementing methods, obtaining feedback on research methodology and experiments, supporting qualitative analysis of user representations, and assisting with interpreting results. We did not use generative AI tools to develop conceptual frameworks, propose or refine hypotheses, or to directly clean and reformat datasets. Additionally, we used generative AI tools to analyze existing literature, search for information, and improve writing and readability. Writing was limited to expressing the authors' intended content. Mathematical claim formulation, proofs, and translations are not applicable to this work. The use of generative AI tools were to support rather than replace the authors' work. All AI-assisted work was reviewed and verified, and we take responsibility for the final content of this paper, including all text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

`UserRepGen-4B` is intended to support research on user simulation and the interactive training and evaluation of AI assistants. More faithful simulations may help developers account for a broader range of user behaviors, but also introduce risks of privacy loss, mischaracterization, and misuse. User representations are inferred from a single conversation and may retain sensitive information or make unsupported assumptions about an individual. Our results in Section 7.3 show that learned behavior descriptions can help link conversations from the same user, highlighting potential privacy or anonymity considerations. Although `UserRepGen-4B` enables more faithful simulation of individual users, it should not be used to impersonate them or present simulated interactions as their own. These simulations should not be used to replace real users.

Our training and evaluation use datasets of real English conversations collected through publicly accessible chatbot interfaces, paid participant studies, and agent coding session records in public GitHub repositories. We also use LLM-generated simulated conversations. These sources do not ensure representative coverage of user populations, and we do not confirm whether the observed improvements in simulation fidelity will generalize to other languages and user populations. Furthermore, although our LLM judge was validated against human preferences, we acknowledge that its evaluations may still reflect model biases and inconsistencies.

REPRODUCIBILITY STATEMENT

`UserRepGen-4B` will be made publicly available, and we also plan to release our code to support reproducibility and future research. We describe the dataset sources, preprocessing criteria, and training and evaluation setup in Section 6.1. Prompts for representation generation, simulation, and evaluation are provided in Appendix C.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024. URL `https://proceedings.iclr.cc/paper_files/paper/2024/file/5be69a584901a26c521c2b51e40a4c20-Paper-Conference.pdf`.

Jacy Reese Anthis, Ryan Liu, Sean M Richardson, Austin C. Kozlowski, Bernard Koch, Erik Brynjolfsson, James Evans, and Michael S. Bernstein. Position: LLM social simulations are a promising research method. In *Forty-second International Conference on Machine Learning Position Paper Track*, 2025. URL `https://openreview.net/forum?id=cRBg1dtj7o`.

Lisa P. Argyle, Ethan C. Busby, Nancy Fulda, Joshua R. Gubler, Christopher Rytting, and David Wingate. Out of one, many: Using language models to simulate human samples. *Political Analysis*, 31(3):337–351, 2023. doi: 10.1017/pan.2023.2.

Joachim Baumann, Vishakh Padmakumar, Xiang Li, John Yang, Diyi Yang, and Sanmi Koyejo. Swe-chat: Coding agent interactions from real users in the wild, 2026. URL https://arxiv.org/abs/2604.20779.

Thomas Kleine Buening, Jonas Hübotter, Barna Pásztor, Idan Shenfeld, Giorgia Ramponi, and Andreas Krause. Aligning language models from user interactions, 2026. URL https://arxiv.org/abs/2603.12273.

Runjin Chen, Andy Arditi, Henry Sleight, Owain Evans, and Jack Lindsey. Persona vectors: Monitoring and controlling character traits in language models, 2025. URL https://arxiv.org/abs/2507.21509.

Harshita Chopra, Kshitish Ghate, Aylin Caliskan, Tadayoshi Kohno, Chirag Shah, and Natasha Jaques. Beyond cooperative simulators: Generating realistic user personas for robust evaluation of LLM agents. In *Second Workshop on Social Simulation with LLMS: Fidelity in Applications*, 2026. URL https://openreview.net/forum?id=cJVWKXCws0.

Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. Scaling synthetic data creation with 1,000,000,000 personas, 2025. URL https://arxiv.org/abs/2406.20094.

Jonas Hübotter, Frederike Lübeck, Lejs Deen Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=QkfkxyRizZ.

Chuanyang Jin, Binze Li, Haopeng Xie, Cathy Mengying Fang, Tianjian Li, Shayne Longpre, Hongxiang Gu, Maximillian Chen, and Tianmin Shu. Thoughttrace: Understanding user thoughts in real-world llm interactions, 2026. URL https://arxiv.org/abs/2605.20087.

Hannah Rose Kirk, Alexander Whitefield, Paul Röttger, Andrew Bean, Katerina Margatina, Juan Ciro, Rafael Mosquera, Max Bartolo, Adina Williams, He He, Bertie Vidgen, and Scott Hale. The prism alignment dataset: What participatory, representative and individualised human feedback reveals about the subjective and multicultural alignment of large language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 105236–105344. Curran Associates, Inc., 2024. doi: 10.52202/079017-3342. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/be2e1b68b44f2419e19f6c35a1b8cf35-Paper-Datasets_and_Benchmarks_Track.pdf.

Ang Li, Haozhe Chen, Hongseok Namkoong, and Tianyi Peng. Llm generated persona is a promise with a catch. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc., 2025. doi: 10.52202/085713-2226. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/5fc150b75e8b509735564de16bc3527b-Paper-Position_Paper_Track.pdf.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. A persona-based neural conversation model. In Katrin Erk and Noah A. Smith (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 994–1003, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1094. URL https://aclanthology.org/P16-1094/.

Xiaozhe Li, Tianyi Lyu, Yang Li, Yichuan Ma, Peiji Li, Linyang Li, Qipeng Guo, Dahua Lin, and Kai Chen. What and when to distill: Selective hindsight distillation for multi-turn agents, 2026. URL https://arxiv.org/abs/2605.19447.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Victoria Lin, Taedong Yun, Maja Mataric, John Canny, Arthur Gretton, and Alexander Nicholas D'Amour. The illusion of intervention: Your LLM-simulated experiment is an observational study. In *ICML 2026 Statistical Frameworks for Uncertainty in Agentic Systems*, 2026. URL https://openreview.net/forum?id=7EZrgYo07P.

Haoran Liu, Yuwei Zhang, Xiyao Li, Bohan Lyu, and Jingbo Shang. Hero: Hindsight-enhanced reflection from environment observations for agentic self-distillation, 2026a. URL https://arxiv.org/abs/2606.11559.

Xinyi Liu, Rinat Khaziev, Hooshang Nayyeri, Emine Yilmaz, Charith Peris, and Hari Thadakamalla. Beyond individual personas: Aligning synthetic dialogue to population-level behavior distributions, 2026b. URL https://arxiv.org/abs/2606.07893.

Yajiao Liu, Xin Jiang, Yichun Yin, Yasheng Wang, Fei Mi, Qun Liu, Xiang Wan, and Benyou Wang. One cannot stand for everyone! leveraging multiple user simulators to train task-oriented dialogue systems. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1–21, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.1. URL https://aclanthology.org/2023.acl-long.1/.

Shuhaib Mehri, Priyanka Kargupta, Tal August, and Dilek Hakkani-Tür. Multisessioncollab: Learning user preferences with memory to improve long-term collaboration, 2026a. URL https://arxiv.org/abs/2601.02702.

Shuhaib Mehri, Philippe Laban, Sumuk Shashidhar, Marwa Abdulhai, Sergey Levine, Michel Galley, and Dilek Hakkani-Tür. Measuring and mitigating the distributional gap between real and simulated user behaviors. *arXiv preprint arXiv:2605.07847*, 2026b.

Shuhaib Mehri, Xiaocheng Yang, Takyoung Kim, Gokhan Tur, Shikib Mehri, and Dilek Hakkani-Tür. Goal alignment in llm-based user simulators for conversational ai. *Transactions of the Association for Computational Linguistics*, 14:852–876, 2026c.

Ethan Mendes, Jungsoo Park, and Alan Ritter. Making expert reasoning learnable with self-distillation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=JG6f02X29a.

Tarek Naous, Philippe Laban, Wei Xu, and Jennifer Neville. Flipping the dialogue: Training and evaluating user language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ykSmkVqzn4.

Lin Ning, Luyang Liu, Jiaxing Wu, Neo Wu, Devora Berlowitz, Sushant Prakash, Bradley Green, Shawn O'Banion, and Jun Xie. User-llm: Efficient llm contextualization with user embeddings. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, pp. 1219–1223, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400713316. doi: 10.1145/3701716.3715463. URL https://doi.org/10.1145/3701716.3715463.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701320. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

Joon Sung Park, Carolyn Q. Zou, Jonne Kamphorst, Niles Egan, Aaron Shaw, Benjamin Mako Hill, Carrie Cai, Meredith Ringel Morris, Percy Liang, Robb Willer, and Michael S. Bernstein. LLM Agents Grounded in Self-Reports Enable General-Purpose Simulation of Individuals. *arXiv e-prints*, art. arXiv:2411.10109, November 2024. doi: 10.48550/arXiv.2411.10109.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/ross11a.html.

Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. Agenda-based user simulation for bootstrapping a POMDP dialogue system. In Candace Sidner, Tanja Schultz,

Matthew Stone, and ChengXiang Zhai (eds.), *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pp. 149–152, Rochester, New York, April 2007. Association for Computational Linguistics. URL https://aclanthology.org/N07-2038/.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Taiwei Shi, Sihao Chen, Bowen Jiang, Linxin Song, Longqi Yang, and Jieyu Zhao. Experiential reinforcement learning. In *ICLR 2026 Workshop on Lifelong Agents: Learning, Aligning, Evolving*, 2026. URL https://openreview.net/forum?id=OSYyL6hqdP.

Yuda Song, Lili Chen, Fahim Tajwar, Rémi Munos, Deepak Pathak, Drew Bagnell, Aarti Singh, and Andrea Zanette. Expanding the capabilities of reinforcement learning via text feedback. In *The 1st Workshop on Scaling Post-training for LLMs*, 2026. URL https://openreview.net/forum?id=om12baYnKM.

Weiwei Sun, Xuhui Zhou, Jiarui Liu, Weihua Du, Haojia Sun, Yiqing Xie, Qianou Ma, Sihao Chen, Mengting Wan, Longqi Yang, Pei Zhou, Sherry Wu, Sean Welleck, Graham Neubig, Yiming Yang, and Maarten Sap. Reinforcing human behavior simulation via verbal feedback, 2026. URL https://arxiv.org/abs/2605.20506.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Hao Wang, Guozhi Wang, Han Xiao, Yufeng Zhou, Yue Pan, Jichao Wang, Ke Xu, Yafei Wen, Xiaohu Ruan, Xiaoxin Chen, and Honggang Qi. Skill-sd: Skill-conditioned self-distillation for multi-turn llm agents, 2026a. URL https://arxiv.org/abs/2604.10674.

Kuang Wang, Xianfei Li, Shenghao Yang, Li Zhou, Feng Jiang, and Haizhou Li. Know you first and be you better: Modeling human-like user simulators via implicit profiles. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21082–21107, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1025. URL https://aclanthology.org/2025.acl-long.1025/.

Shengjie Wang, Guanghe Li, Zonghan Yang, and Yang Gao. Hindsight hint distillation: Scaffolded reasoning for swe agents from cot-free answers, 2026b. URL https://arxiv.org/abs/2605.11556.

Yingshan Susan Wang, Cedegao E. Zhang, Linlu Qiu, Zexue He, Pengyuan Li, Alex Pentland, Roger P. Levy, and Yoon Kim. Learning user simulators with turing rewards, 2026c. URL https://arxiv.org/abs/2606.19336.

Shirley Wu, Michel Galley, Baolin Peng, Hao Cheng, Gavin Li, Yao Dou, Weixin Cai, James Zou, Jure Leskovec, and Jianfeng Gao. CollabLLM: From passive responders to active collaborators. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=DmH4HHVb3y.

Shirley Wu, Evelyn Choi, Arpandeep Khatua, Zhanghan Wang, Joy He-Yueya, Tharindu Cyril Weerasooriya, Wei Wei, Diyi Yang, Jure Leskovec, and James Zou. Humanlm: Simulating users with state alignment beats response imitation. *arXiv preprint arXiv:2603.03303*, 2026.

Yunze Xiao, Vivienne J. Zhang, Chenghao Yang, Ningshan Ma, Weihao Xuan, and Jen tse Huang. The chameleon's limit: Investigating persona collapse and homogenization in large language models, 2026. URL https://arxiv.org/abs/2604.24698.

Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. Iterative preference learning from human feedback: Bridging theory and practice for RLHF under KL-constraint. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 54715–54754. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/xiong24a.html.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains, 2024. URL https://arxiv.org/abs/2406.12045.

Woongyeong Yeo, Yumin Choi, Taekyung Ki, and Sung Ju Hwang. Hint-sd: Targeted hindsight self-distillation for long-horizon agents, 2026. URL https://arxiv.org/abs/2605.17873.

Simon Yu, Nicholas Tomlin, Marwa Abdulhai, Ximing Lu, Derek Chong, Abe Hou, Dilara Soylu, Sergey Levine, Christopher D Manning, and Weiyan Shi. One frozen simulator is not enough: Simulator collapse in multi-agent rl. *arXiv preprint arXiv:2608.12253*, 2026.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2204–2213, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1205. URL https://aclanthology.org/P18-1205/.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models, 2025. URL https://arxiv.org/abs/2506.05176.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=Jpxfof0EaS.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. Wildchat: 1m chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Bl8u7ZRlbM.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric Xing, Joseph E Gonzalez, Ion Stoica, and Hao Zhang. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 22225–22257, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/5f9bfdfe3685e4ccdbc0e7fb29cccf2a-Paper-Conference.pdf.

Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and Maarten Sap. Sotopia: Interactive evaluation for social intelligence in language agents. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 40975–41019, 2024a. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/b3075b88e583a0e98d8b24338a613060-Paper-Conference.pdf.

Xuhui Zhou, Weiwei Sun, Weihua Du, Jiarui Liu, Haojia Sun, Qianou Ma, Tongshuang Wu, Yiming Yang, and Maarten Sap. Odyssim: Building foundation models for human behavior simulation. *arXiv preprint arXiv:2606.14199*, 2026.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. ArCHer: Training language model agents via hierarchical multi-turn RL. In *Forty-first International Conference on Machine Learning*, 2024b. URL https://openreview.net/forum?id=b6rA0kAHT1.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, pp. 1097–1100, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356572. doi: 10.1145/3209978.3210080. URL https://doi.org/10.1145/3209978.3210080.

A HINDSIGHT DISTILLATION ALGORITHM

Algorithm 1 presents our hindsight distillation algorithm. This algorithm distills hindsight feedback-conditioned improvements into a representation generator model, enabling it to generate improved representations directly from an input conversation.

Algorithm 1 Hindsight Distillation

input: representation generator G_θ; conversation dataset $\mathcal{D}$; ensemble of simulation environments $\mathcal{E} = \{(U_k, A_k, J_k)\}_{k=1}^{K}$; refinement rounds I
repeat
  sample a conversation $\mathcal{C} \sim \mathcal{D}$.
  sample an initial user representation $r^{(0)} \sim G_\theta(\cdot \mid \mathcal{C})$.
  for $i = 0, \ldots, I-1$ **do**
    for all $e_k = (U_k, A_k, J_k) \in \mathcal{E}$ **do**
      roll out $\tilde{\mathcal{C}}_k^{(i)} \sim \text{rollout}(U_k(\cdot \mid r^{(i)}), A_k)$.
      generate hindsight feedback $f_k^{(i)} \sim J_k(\cdot \mid \mathcal{C}, \tilde{\mathcal{C}}_k^{(i)})$.
    end for
    collect feedback $f^{(i)} = (f_1^{(i)}, \ldots, f_K^{(i)})$.
    sample a refined user representation $r^{(i+1)} \sim G_\theta(\cdot \mid \mathcal{C}, r^{(i)}, f^{(i)})$.
  end for
  update θ with gradient descent on $\mathcal{L}(\theta)$ to predict $r^{(I)}$ from $\mathcal{C}$ alone.
until converged

B HUMAN VALIDATION OF LLM JUDGE

In this section, we validate whether GPT-5.4 mini's fidelity scores align with human judgements. We construct a human-labeled validation set of 100 original conversations from our evaluation set, each paired with two simulations generated using representations from Qwen3.5-4B and `UserRepGen-4B`, with GPT-5.4 mini as both user simulator and assistant. We select only pairs for which the LLM judge assigns unequal fidelity scores. For each pair, presented in randomized order and without its source or scores, we label the simulation whose user behavior is more faithful to the original user. We then measure how often the judge's ranking agrees with these labels.

The judge's ranking agrees with our labels on 88% of pairs. As shown in Table 5, agreement increases with the absolute difference between the two fidelity scores, from 80.5% when the gap is one point to 97.4% when the gap is at least three points.

Table 5: Agreement between our manually labeled preferences and the LLM judge. The score gap is the absolute difference between the two simulated conversations' overall fidelity scores.

Score gap	Agreement	Rate
1	33 / 41	80.5%
2	17 / 20	85.0%
≥ 3	38 / 39	97.4%
Overall	88 / 100	88.0%

These results support the LLM judge as a measure of simulation fidelity. Moreover, our evaluations for population-level diversity and distinguishing users across conversations in Sections 7.2 and 7.3 provide complementary evidence, by demonstrating that representations optimized for simulation fidelity improve performance on downstream tasks that do not rely on the fidelity judge.

C PROMPTS

User Representation Generation Prompt

```
You are an expert at analyzing user behavior in conversations between a user and an AI
     assistant. Given the conversation below, your task is to produce a representation of
     the user.

Treat the conversation as evidence about the user, and use it to infer how they generally
     behave and how they would typically act in other conversations. Focus on stable,
     general tendencies such as: how they phrase and specify requests, how they decompose
     goals across turns, how they respond to and give feedback to the assistant, how and
     when they provide context, their communication style, the dialog acts they use, and
     their degree and depth of engagement. Keep this description general: do not include
     topic-specific details or content from the conversation, do not quote or closely
     paraphrase full user messages, and do not refer to specific moments in this
     conversation (e.g. "here", "in this conversation", "when they asked about X"). Short
     distinctive words or phrases may be quoted sparingly when they are useful examples of
      the user's communication style. Describe their patterns in your own general terms
     instead.

Separately, you will also describe the task the user is trying to accomplish. Here you may
      include topic-specific details about what the user is doing in the conversation.

Conversation:
{conversation_history}

Output a single valid JSON object of the form {{"task": "...", "user_behavior": "..."}}.
     Do not wrap the output in markdown or add any other text.
```

Improving User Representation Prompt

```
You are an expert at analyzing user behavior in conversations between a user and an AI
     assistant. You have a representation of the user based on the conversation below. We
     used that representation to simulate a user, and then collected feedback from a judge
     . Your task is to incorporate that feedback and generate an improved user
     representation.

The user representation consists of a task and user_behavior field. The task field
     describes what the user is trying to accomplish, and can include topic-specific
     details about what the user is doing in the conversation. The user_behavior field
     describes how they generally behave and how they would typically act in conversations
     . It focuses on stable, general tendencies such as: how they phrase and specify
     requests, how they decompose goals across turns, how they respond to and give
     feedback to the assistant, how and when they provide context, their communication
     style, the dialog acts they use, and their degree and depth of engagement. This
     description should be general: it should not include topic-specific details or
     content from the conversation, it should not quote or closely paraphrase full user
     messages, and it should not refer to specific moments in this conversation (e.g. "
     here", "in this conversation", "when they asked about X"). Short distinctive words or
      phrases may be quoted sparingly when they are useful examples of the user's
     communication style. It should describe their patterns in general terms.

Additional guidelines for improving the representation:
- Optimize for behavioral simulation, not efficient task completion: preserve multi-turn
     repetition, incremental follow-ups, verbose constraints, and inefficient interaction
     patterns when they appear in the real user.
- Make user_behavior operational: specify what the simulator should do and should not do,
     including negative constraints from judge feedback (e.g. do not terminate early,
     combine turns, over-polish, change formatting style, or solve the task directly).
- Be explicit about role distinctions: if the real user asks the assistant/agent to
     produce commands, JSON, code, tool calls, or formatted output, say whether the user
     writes instructions to the assistant or directly provides the final artifact.
- When feedback says the simulation is too short, too generic, or collapsed multiple turns
     , specify the desired utterance form and interaction arc: approximate turn count,
     near-duplicate vs incremental turns, what each turn should contain, and what not to
     compress.
- For prompt-author or agent-control users, state if user turns should be long prompt
     blocks with role instructions, constraints, allowed actions, response schemas,
     evaluation criteria, and goals; also state not to replace those blocks with a bare
     command, JSON object, or final answer.

Conversation:
{conversation_history}
```

```
Current User Representation:
task: {task}
user_behavior: {user_behavior}

Judge Feedback:
{feedback}

Generate an updated representation that incorporates the judge feedback. This updated
     representation should follow the guidelines above, and should be in the same format
     as the current representation.
Output only valid JSON with exactly this shape:
{{"task": "<improved task>", "user_behavior": "<improved behavior representation>"}}
```

User Simulator System Prompt

```
You are a user simulator. You will be provided with a user representation and your task is
      to role-play as that user in a conversation with an AI assistant.

# User Representation
{user_representation}

# Instructions
- Stay consistent with the information provided in the user representation
- Never describe the user or your own behavior, never break character, and never mention
     that you are simulating, role-playing, or following a description.
- When the task requires concrete material for the assistant to work on (e.g. a paragraph
     to analyze, a passage to translate, code to debug, an essay to edit, data to
     summarize), invent a plausible, self-contained example yourself and include it inline
      in your message. Never ask the assistant to work on something you have not actually
     provided.
- When this user's goal has been met, or when they would naturally end the chat, respond
     with exactly \"{termination_signal}\" and nothing else.
```

Assistant System Prompt

```
You are a helpful AI assistant. Respond to the user's messages accurately and helpfully.
```

Judge Prompt - Full Conversation

```
You are an expert analyzing user behavior in human-AI conversations. Your task is to
     evaluate how well a simulated user reproduces the behavior of a real user.

You are given the full real and simulated conversations, including assistant turns. Use
     the assistant turns only as context for interpreting the user's behavior; score only
     how faithfully the SIMULATED user reproduces the REAL user's behavior. Do not score
     assistant quality.

Evaluate how well the simulated user reproduces the real user along the following
     dimensions:

- Communication style:
    - Do they use similar punctuation, capitalization, spelling/grammar, typos, slang,
         emoji, contractions, and other surface-level writing habits? Compare concrete
         surface artifacts, not just the general vibe.
    - Do they have similar response lengths? Are they terse or verbose in the same way?
    - Do they use similar formatting, such as paragraphs, markdown, bullets, or line
         breaks?
    - Do they have similar tone/formality, politeness, greetings/thanks, directness,
         warmth, and social conventions?
- Engagement (The REAL user took {real_user_turns} turn(s); the SIMULATED user took {
     sim_user_turns} turn(s)):
    - Do they spend a similar number of turns on the same interaction arc or subtask? Do
         they persist or move on quickly in a similar way?
    - Do they continue, stop, or terminate at a similar point?
- Requests:
    - Do they pursue a similar number of distinct asks or subtasks? Ignore the exact
         domain/topic, but compare the interaction structure.
    - Do they ask for things in a similar way: direct commands vs questions, broad asks vs
          narrow asks, explicit constraints vs implicit cues?
    - Do they front-load requirements or reveal them gradually across turns in a similar
         way?
```

```
    - Do they keep separate real-user requests separate, rather than combining them too
        early?
    - Are their requests similarly open-ended, verification-oriented, correction-oriented,
         or solution-oriented?
- Responses:
    - Given the assistant turns shown in each transcript, do the users respond to
        assistant outputs in similar ways?
    - Do they accept, reject, correct, validate, express dissatisfaction, refine
        constraints, or redirect the conversation similarly?
    - Is their feedback similarly explicit or implicit, terse or explanatory, polite or
        blunt?
- Context:
    - Do they provide a similar amount and type of context: examples, data, code, prior
        attempts, constraints, personal background?
    - Is context front-loaded or revealed gradually in a similar way?

# REAL conversation
{real}

# SIMULATED conversation
{simulated}

# Additional instructions
- Focus ONLY on the USER behavior, and judge HOW the user behaves rather than WHAT they
    are working on: the specific domain, entities, technologies, or subject matter are
    not important. Do not reward or penalize the simulated user for working on the same
    or a different topic than the real user.
- Use assistant turns only to understand what the user is responding to. Do not penalize
    the simulated user because the assistant gave different content, unless the simulated
     user's behavioral response pattern is also different.
- Account for the fact that the simulated user may need to act differently because the
    simulated assistant behaved differently. A faithful user may ask a different follow-
    up, skip a correction, or introduce a clarification at a different point if the
    assistant's answer created a different opportunity; judge whether the underlying
    behavioral tendency is similar.
- Pay attention to whether the simulated user reproduces the full arc of the real user's
    user turns. For example, if the real user works on multiple tasks across the
    conversation but the simulated user works on only one, there is a behavioral mismatch
    .
- Penalize generic simulator behavior such as asking "How can I help?", ending immediately
    , over-polishing terse users, using markdown when the real user does not, or
    combining multiple real-user follow-ups into one turn.

For each dimension, write a one-sentence rationale, and then assign a score from 1 (poorly
     reproduces the real user's behavior) to 10 (highly reproduces the real user's
    behavior). Finally, give a single 1-10 overall score that measures the simulator's
    overall behavioral similarity.

Output a valid JSON object with exactly this shape and nothing else (no markdown):
{{"dimensions": {{"communication_style": {{"rationale": "<one sentence>", "score":
    <1-10>}}, "engagement": {{"rationale": "<one sentence>", "score": <1-10>}}, "requests
    ": {{"rationale": "<one sentence>", "score": <1-10>}}, "responses": {{"rationale": "<
    one sentence>", "score": <1-10>}}, "context": {{"rationale": "<one sentence>", "score
    ": <1-10>}}}}, "overall": <1-10>}}
```

Judge Prompt - User Utterances Only

```
You are an expert analyzing user behavior in human-AI conversations. Your task is to
    evaluate how well a simulated user reproduces the behavior of a real user.

You are given the user utterances from a real and simulated conversation between a user
    and an AI assistant. Evaluate how well the simulated user reproduces the behavior of
    the real user along the following dimensions:

- Communication style:
    - Do they use similar punctuation, capitalization, spelling/grammar, typos, slang,
        emoji, contractions, and other surface-level writing habits? Compare concrete
        surface artifacts, not just the general vibe.
    - Do they have similar response lengths? Are they terse or verbose in the same way?
    - Do they use similar formatting, such as paragraphs, markdown, bullets, or line
        breaks?
    - Do they have similar tone/formality, politeness, greetings/thanks, directness,
        warmth, and social conventions?
- Engagement (The REAL user took {real_user_turns} turn(s); the SIMULATED user took {
    sim_user_turns} turn(s)):
    - Do they spend a similar number of turns on the same interaction arc or subtask? Do
        they persist or move on quickly in a similar way?
```

```
    - Do they continue, stop, or terminate at a similar point?
- Requests:
    - Do they pursue a similar number of distinct asks or subtasks? Ignore the exact
        domain/topic, but compare the interaction structure.
    - Do they ask for things in a similar way: direct commands vs questions, broad asks vs
         narrow asks, explicit constraints vs implicit cues?
    - Do they front-load requirements or reveal them gradually across turns in a similar
        way?
    - Do they keep separate real-user requests separate, rather than combining them too
        early?
    - Are their requests similarly open-ended, verification-oriented, correction-oriented,
         or solution-oriented?
- Responses:
    - Based only on the visible user follow-up turns, do they appear to address/engage
        with prior assistant outputs in a similar way?
    - Do they accept, reject, correct, validate, express dissatisfaction, refine
        constraints, or redirect the conversation similarly?
    - Is their feedback similarly explicit or implicit, terse or explanatory, polite or
        blunt?
- Context:
    - Do they provide a similar amount and type of context: examples, data, code, prior
        attempts, constraints, personal background?
    - Is context front-loaded or revealed gradually in a similar way?

# REAL user utterances
{real}

# SIMULATED user utterances
{simulated}

# Additional instructions
- Focus ONLY on the USER behavior, and judge HOW the user behaves rather than WHAT they
    are working on: the specific domain, entities, technologies, or subject matter are
    not important. Do not reward or penalize the simulated user for working on the same
    or a different topic than the real user.
- Pay attention to whether the simulated user reproduces the full arc of the real user's
    user turns. For example, if the real user works on multiple tasks across the
    conversation but the simulated user works on only one, there is a behavioral mismatch
    .
- Penalize generic simulator behavior such as asking "How can I help?", ending immediately
    , over-polishing terse users, using markdown when the real user does not, or
    combining multiple real-user follow-ups into one turn.
- Do not infer hidden assistant content. If a response pattern is not visible from the
    user utterances themselves, do not give credit for it.

For each dimension, write a one-sentence rationale, and then assign a score from 1 (poorly
     reproduces the real user's behavior) to 10 (highly reproduces the real user's
    behavior). Finally, give a single 1-10 overall score that measures the simulator's
    overall behavioral similarity.

Output a valid JSON object with exactly this shape and nothing else (no markdown):
{{"dimensions": {{"communication_style": {{"rationale": "<one sentence>", "score":
    <1-10>}}, "engagement": {{"rationale": "<one sentence>", "score": <1-10>}}, "requests
    ": {{"rationale": "<one sentence>", "score": <1-10>}}, "responses": {{"rationale": "<
    one sentence>", "score": <1-10>}}, "context": {{"rationale": "<one sentence>", "score
    ": <1-10>}}}}}, "overall": <1-10>}}
```

D TRAINING HYPERPARAMETERS

Table 6: Training hyperparameters for GRPO, OPSD, SFT, and OSFT.

Hyperparameter	**GRPO**	**OPSD**	**SFT**	**OSFT**
Learning rate	5×10^{-6}	5×10^{-6}	5×10^{-6}	5×10^{-6}
Effective batch size	32	32	32	32
Training steps	100	100	100	100
LoRA rank (r)	32	64	64	64
LoRA alpha (α)	64	128	128	128
Maximum prompt length	8,192	8,192	8,192	8,192
Maximum completion length	1,024	1,024	1,024	1,024
Maximum sequence length	9,216	9,216	9,216	9,216
Generations per prompt	8	1	–	1
Sampling temperature	0.7	1.1	–	1.1
Top-p	0.8	0.95	–	0.95
Top-k	20	20	–	20

E FURTHER ANALYSIS ON HOW TRAINING CHANGES USER REPRESENTATIONS

E.1 PAIRWISE ANALYSIS WITH GPT-5.4 MINI

In Section 6.4, we qualitatively analyze how training changes user representations by manually comparing representations generated by the base and trained models. We use the resulting observations to guide a more larger-scale, systematic analysis over the 700 conversations in our evaluation set. For each conversation, we present GPT-5.4 mini with the representations generated by Qwen3.5-4B and `UserRepGen-4B` in randomized order, and ask it to answer 11 questions about how they differ:

Task descriptions.

1. Which task description better captures constraints introduced in later turns?
2. Which task description better captures shifts in direction or goal?
3. Which task description better captures multiple distinct objectives or subtasks?
4. Which task description has less repetition?

Behavior descriptions.

5. Which behavior description is more specific and actionable?
6. Which behavior description uses more useful negative and contrastive instructions?
7. Which behavior description better captures the user's response policy?
8. Which behavior description better captures the user's communication style?
9. Which behavior description better captures the user's engagement patterns, including how long they persist, when they move on, and when they end the conversation?
10. Which behavior description better captures how the user makes requests, including how many requests or subtasks they pursue, how directly they ask, and whether they provide requirements upfront or gradually?
11. Which behavior description better captures how the user provides context, including the amount and type of context and whether it is provided upfront or gradually?

For each question, GPT-5.4 mini must select one of the two representations. We use the following prompt:

How Training Changes User Representations - Judge Prompt

```
You are an expert in analyzing user representations. A user representation is a natural-
     language description generated from a conversation and used to condition an LLM-based
      user simulator. Its purpose is to give the simulator the information needed to
     reproduce the user behavior observed in the original conversation. User
     representations are made up of two components: (1) a task description that describes
     what the user is trying to accomplish, and (2) a behavior description that describes
     how the user behaves.

Your task is to help me analyze the difference between the two user representations. You
     will be given two representations generated from the same conversation, presented in
     randomized order. The order carries no information about which models produced them
     or which representation is better. Carefully compare the content of the two
     representations for every criterion and answer the questions below.

# REPRESENTATION 1
task: {representation_1_task}
behavior: {representation_1_behavior}

# REPRESENTATION 2
task: {representation_2_task}
behavior: {representation_2_behavior}

# For each criterion of the following criteria, select either `representation_1` or `
     representation_2`:

## Task Description
1. Which representation better captures constraints introduced in later turns?
2. Which representation better captures shifts in direction or goal?
3. Which representation better captures multiple distinct objectives or subtasks?
4. Which representation has less repetition?

## Behavior Description
5. Which behavior description is more specific and actionable? Prefer descriptions of
     observable patterns that a simulator can reproduce over broad characterizations such
     as  goal -oriented or  directive .
6. Which behavior description uses more useful negative and contrastive instructions?
7. Which behavior description better captures the user's response policy?
8. Which behavior description better captures the user's communication style?
9. Which behavior description better captures the user's engagement patterns, including
     how long they persist, when they move on, and when they end the conversation?
10. Which behavior description better captures how the user makes requests, including how
     many requests or subtasks they pursue, how directly they ask, and whether they
     provide requirements upfront or gradually?
11. Which behavior description better captures how the user provides context, including
     the amount and type of context and whether it is provided upfront or gradually?

# Output format

Output only the following JSON object and nothing else. For each question, provide your
     selection and one brief sentence explaining the most important difference:

{{
  "q1": {{"selection": "representation_1|representation_2", "reasoning": "<brief sentence
       >"}}, "q2": {{"selection": "representation_1|representation_2", "reasoning": "<
       brief sentence>"}}, "q3": {{"selection": "representation_1|representation_2", "
       reasoning": "<brief sentence>"}}, "q4": {{"selection": "representation_1|
       representation_2", "reasoning": "<brief sentence>"}}, "q5": {{"selection": "
       representation_1|representation_2", "reasoning": "<brief sentence>"}}, "q6": {{"
       selection": "representation_1|representation_2", "reasoning": "<brief sentence>"}},
        "q7": {{"selection": "representation_1|representation_2", "reasoning": "<brief
       sentence>"}}, "q8": {{"selection": "representation_1|representation_2", "reasoning
       ": "<brief sentence>"}}, "q9": {{"selection": "representation_1|representation_2",
       "reasoning": "<brief sentence>"}}, "q10": {{"selection": "representation_1|
       representation_2", "reasoning": "<brief sentence>"}}, "q11": {{"selection": "
       representation_1|representation_2", "reasoning": "<brief sentence>"}}
}}
```

Our results are presented in Table 7. GPT-5.4 mini selects the task description from `UserRepGen-4B` as better capturing constraints introduced in later turns in 63.6% of comparisons, shifts in direction or goal in 61.0%, and multiple objectives in 58.3%. Together, these results support our qualitative observation that training produces more comprehensive task descriptions. However, `UserRepGen-4B` is selected as less repetitive in only 33.1% of comparisons, indicating that Qwen3.5-4B usually produces less repetitive task descriptions. For behavior descriptions, GPT-5.4 mini selects the description from `UserRepGen-4B` as containing more useful negative

Table 7: Comparing user representations generated by our trained and untrained models along several dimensions. We report the percentage of times that GPT-5.4 mini selects the representation generated by `UserRepGen-4B` over the one generated by Qwen3.5-4B.

Question	`UserRepGen-4B` Selected (%)
Task description	
Captures constraints introduced in later turns	63.6
Captures shifts in direction or goal	61.0
Captures multiple objectives or subtasks	58.3
Has less repetition	33.1
Behavior description	
Is more specific and actionable	87.7
Uses more useful negative and contrastive instructions	91.6
Captures the user's response policy	83.1
Captures the user's communication style	88.1
Captures the user's engagement patterns	90.0
Captures how the user makes requests	83.9
Captures how the user provides context	85.9

and contrastive instructions in 91.6% of comparisons, better capturing user engagement patterns in 90.0%, better capturing communication style in 88.1%, and being more specific and actionable in 87.7%. It also selects `UserRepGen-4B` as better describing how the user provides context in 85.9%, how the user makes requests in 83.9%, and how the user responds to the assistant in 83.1%. These results support our qualitative observation that training produces behavior descriptions that are more specific and actionable, and that capture dimensions of user behavior previously overlooked.

E.2 User Representation Examples

We provide two examples of user representations generated by Qwen3.5-4B and `UserRepGen-4B`. Yellow highlighting marks the differences most relevant to our analysis.

Example 1. This example illustrates how `UserRepGen-4B` generates task descriptions that capture the full sequence of user requests, whereas the base model summarizes the conversation at a high-level. The behavior description replaces broad terms such as "curious", "direct", and "inquisitive" with observable patterns that a simulator can reproduce, such as specifying that the user asks short factual questions, provides no initial context, and pivots to new angles rather than giving fedback.

User representation generated by Qwen3.5-4B

Task description. The user seeks to understand the core ideas and historical impact of a specific philosopher, Čaadaev, and subsequently explores the philosophical connections between him and other notable figures like Dostoevsky.

Behavior description. The user initiates inquiries with broad, open-ended questions about a topic's fundamental nature and relevance. They progressively narrow the scope by requesting specific follow-up information on influences and related concepts. Their engagement is characterized by a high degree of curiosity and a desire for concise summaries, often explicitly requesting structural constraints (such as a sentence limit) when formulating their requests. The communication style is direct and inquisitive, treating the assistant as a knowledge repository without providing extensive initial context.

User representation generated by `UserRepGen-4B`

Task description. The user seeks to understand the philosophical contributions of Ivan Čaadaev and their influence on later thinkers, specifically focusing on Fyodor Dostoevsky. The interaction involves a series of sequential inquiries about Čaadaev's core themes, modern relevance, intellectual lineage, and a specific request for a concise summary of Dostoevsky's philosophy.

Behavior description. The user engages in a multi-turn dialogue characterized by short, direct, and factual questions that progressively narrow down the scope of inquiry from general biography to specific thematic analysis and finally to a constrained output format (two sentences). They decompose complex topics into discrete sub-questions without providing initial context or background information, relying instead on the assistant's responses to inform subsequent queries. The communication style is terse and impersonal, avoiding conversational fillers, emotional expressions, or requests for formatting details like markdown headers. The user does not provide feedback on previous answers but immediately pivots to new angles or deeper dives into related figures based on the assistant's output. They maintain a consistent pattern of asking distinct follow-ups rather than combining multiple goals into single prompts or requesting broad overviews before drilling down. The engagement level is moderate; they do not terminate the conversation early nor do they expand significantly beyond the immediate topic area, preferring a linear progression of specific facts and summaries.

Example 2. In this example, the behavior description from `UserRepGen-4B` specifies how the user formulates requests: short, direct questions, often phrased as yes/no inquiries, with little upfront context. It also describes how the user responds to the assistant, noting that they neither critique nor refine previous answer but instead use them to guide the next question. The base model, by contrast, relies on broad characterizations.

Representation generated by Qwen3.5-4B

Task description. The user is investigating the safety, ethical implications, and management challenges of nuclear energy, specifically focusing on waste disposal methods. They are exploring alternatives to traditional nuclear power and considering how artificial intelligence can contribute to solving complex environmental and safety problems.

Behavior description. The user engages in a structured inquiry by posing direct questions that progressively narrow down from general safety concerns to specific technical solutions and ethical considerations. They actively seek alternative perspectives and practical applications (such as AI's role) rather than accepting single answers. Their communication style is concise and focused on high-level concepts, demonstrating a degree of engagement that balances curiosity with a desire for actionable information.

Representation generated by `UserRepGen-4B`

Task description. The user is exploring the safety, morality, and future implications of nuclear energy, specifically focusing on waste disposal methods and seeking potential AI-driven solutions for these challenges.

Behavior description. The user engages in a multi-turn dialogue with a distinct communication style characterized by short, direct questions often phrased as simple yes/no inquiries or brief statements. They decompose complex topics into sequential sub-questions without providing extensive background context upfront, relying instead on the assistant's responses to guide subsequent turns. The user maintains a low level of engagement per turn, avoiding elaboration, justification, or follow-up analysis beyond the immediate question. They do not provide feedback, corrections, or alternative perspectives; their interaction pattern is strictly linear and interrogative. The user consistently asks new, related questions rather than refining previous ones or requesting specific formatting changes.

F Measuring the Gap Between Distributions of Original and Simulated User Behaviors

To evaluate how well user representations preserve behavioral diversity at the population level, we measure the distributional gap between original and simulated user behaviors. We follow the pipeline and hyperparameters of Mehri et al. (2026b), which we detail below.

User behavior descriptions. We use Qwen3.5-122B-A10B-FP8 to describe the user behavior in each original and simulated conversation along six behavioral dimensions: requests, responses, context, communication style, and dilaog acts under the Dialog Act Markup in several Layers (DAMSL) framework and Schema-Guided Dataset (SGD) taxonomies. Each prompt instructs the model to describe general behavioral patterns while excluding task-specific details. We use the following prompts:

User Behavior Description: Requests Dimension

```
You are an expert analyzing user behaviour in human-AI conversations. The user has a goal,
      and the assistant helps them achieve it. Your task is to describe the user's
     behavior according to the criteria below.

# User Goal
{user_goal}
# User Utterances
{conversation_history}

# Analysis Criteria
1. Specification and Articulation
   - How specified are the requests? Is the first request underspecified, and clarified in
        subsequent turns? Or are the requests exhaustive?
   - What type of information is left underspecified/specified? (e.g. constraints, edge
       cases, context, output format, etc.)
   - Does the user explicitly command specific tasks, or do they rely on indirect cues (e.
       g., presenting an error without explicitly asking for a fix)?

2. User Goal Decomposition
   - How is the user goal decomposed across utterances?
      - Single-shot: The entire goal is expressed in one utterance with no further
          decomposition.
      - Top-down: High-level goal stated upfront, then progressively refined or elaborated
          .
      - Bottom-up: Individual preconditions or sub-tasks addressed first, building toward
          the overall goal.
      - Chained: Each request builds purely on the immediate previous turn rather than
          referring back to a central goal.

3. Relevance to Goal
   - Are the requests directly related to the user goal? Or does the user introduce
       secondary/perpendicular/emergent needs?
   - What functions do the requests serve beyond achieving the user goal? (setting context
       , probing AI capabilities, verifying intermediate outputs, logistics,
       troubleshooting, side-effects, exploring related sub-tasks, etc.)

# Instructions
- Generate terse, concise bullet points, not full sentences.
- Abstract away from the specific topic/domain
- IMPORTANT:Do NOT use task-specific terms (e.g., "coding," "booking," "Python", "CSV").
     Use generic substitutes (e.g., "executing a task," "providing constraints," "the
     target artifact").

Output a valid JSON object using the exact format below. Do not include any text outside
     the JSON.

{{
   "specification_and_articulation": {{
      "specification_level": "how specified are requests?",
      "underspecified_aspects": "what types of information are left underspecified/
          specified? (provide high-level descriptions without task-specific details)",
      "articulation_mode": "how does the user articulate their requests (explicit
          directives, indirect cues, mixed)?"
   }},
   "goal_decomposition_strategy": "describe how the user goal decomposed across utterances
       ?",
   "relevance_to_goal": {{
      "goal_adherence": "Are the user's requests directly related to the user goal? If not
          , what other functions do they serve? (provide high-level descriptions without
          task-specific details)"
   }}
}}
```

User Behavior Description: Responses Dimension

```
You are an expert analyzing user behaviour in human-AI conversations. The user has a goal,
      and the assistant helps them achieve it. Your task is to describe the user's
     behavior according to the criteria below.

# User Goal
{user_goal}

# Conversation
{conversation_history}
```

```
# Analysis Criteria
1. Engagement and Evaluation
   - Does the user engage with the agent responses, or ignore/skip over them?
   - How does the user evaluate the agent's output? (e.g., explicit validation, implicit
       acceptance by continuing, partial acceptance with corrections, rejection)
   - Does the user provide specific or actionable feedback, or only surface-level
       acknowledgment?

2. Response Composition
   - What types of actions are present in the user's responses? Are they reactive (e.g.,
       validation, acknowledgment, answering agent questions, corrections, feedback),
       proactive (e.g., follow-up requests, new constraints/preferences, suggestions,
       questions), or self-directed (e.g., thinking out loud, expressing uncertainty,
       narrating their process)?

3. Steering Mechanism
   - Does the user steer through direct follow-up requests, or through indirect means? (e.
       g., asking questions that implicitly request action, expressing dissatisfaction
       without stating what to change, providing hints or examples rather than directives
       )
   - Does the user introduce new preferences, constraints, or feedback as part of their
       responses, effectively reshaping the task incrementally?

# Instructions
- Generate terse, concise bullet points, not full sentences.
- Abstract away from the specific topic/domain.
- IMPORTANT: Do NOT use task-specific terms (e.g., "coding," "booking," "Python", "CSV").
    Use generic substitutes (e.g., "executing a task," "providing constraints," "the
    target artifact").

Output a valid JSON object using the exact format below. Do not include any text outside
    the JSON.

{{
   "engagement_and_evaluation": {{
      "engagement_level": "does the user engage with the agent's responses or skip over
          them?",
      "evaluation_mode": "how does the user evaluate the agent's output? (explicit
          validation, implicit acceptance, partial acceptance, rejection, etc.)",
      "feedback_specificity": "does the user provide specific/actionable feedback or only
          surface-level acknowledgment?"
   }},
   "response_composition": {{
      "action_types": "What types of actions are present in the user's responses? Are they
           reactive (e.g., validation, acknowledgment, answering agent questions,
          corrections, feedback), proactive (e.g., follow-up requests, new constraints/
          preferences, suggestions, questions), or self-directed (e.g., thinking out loud
          , expressing uncertainty, narrating their process)?"
   }},
   "steering_mechanism": {{
      "directness": "does the user steer through explicit follow-up requests or indirect
          means (questions, hints, expressed dissatisfaction)?",
      "incremental_reshaping": "does the user introduce new preferences/constraints/
          feedback that reshape the task within their responses?"
   }}
}}
```

User Behavior Description: Context Dimension

```
You are an expert analyzing user behaviour in human-AI conversations. The user has a goal,
     and the assistant helps them achieve it. Your task is to describe the user's
    behavior according to the criteria below.

# User Goal
{user_goal}

# Conversation
{conversation_history}

# Analysis Criteria
1. Context Richness
   - How much context does the user provide overall? Do they share background about
       themselves, their situation, what they've already tried, what they're struggling
       with, or what they're thinking?
   - Does the user provide all relevant context needed for the agent to help effectively,
       or is essential context left out?
```

```
   - Does the user provide context at all, or do they simply issue directives without
       situating the task?

2. Context Type
   - What types of context does the user provide? (e.g., personal background, goals/
       motivations, prior attempts, existing solutions, constraints, preferences, domain
       knowledge, emotional state, thought process)

3. Context Delivery
   - Does the user front-load all relevant context in their first message, or reveal it
       incrementally across turns? What types of context are introduced later? (e.g.,
       constraints they forgot, preferences they didn't think to mention, background that
        becomes relevant as the task evolves)
   - Is incremental context revealed in response to agent questions, or volunteered
       unprompted?

# Instructions
- Generate terse, concise bullet points, not full sentences.
- Abstract away from the specific topic/domain.
- IMPORTANT: Do NOT use task-specific terms (e.g., "coding," "booking," "Python", "CSV").
    Use generic substitutes (e.g., "executing a task," "providing constraints," "the
    target artifact").

Output a valid JSON object using the exact format below. Do not include any text outside
    the JSON.

{{
   "context_richness": {{
      "depth": "how much context does the user provide? do they share background, prior
          attempts, thought process, etc.?",
      "completeness": "does the user provide all relevant context or leave essential
          information out?",
      "contextualization_vs_directives": "does the user situate the task with context, or
          simply issue directives?"
   }},
   "context_type": "what types of context does the user provide? (personal background,
       goals/motivations, prior attempts, existing solutions, constraints, preferences,
       domain knowledge, emotional state, thought process, etc.)",
   "context_delivery": {{
      "distribution": "does the user front-load context or reveal it incrementally? What
          types of context are introduced in later turns?",
      "trigger": "is incremental context revealed in response to agent questions or
          volunteered unprompted?"
   }}
}}
```

User Behavior Description: Communication Style Dimension

```
You are an expert analyzing user behaviour in human-AI conversations. The user has a goal,
     and the assistant helps them achieve it. Your task is to describe the user's
    behavior according to the criteria below.

# User Goal
{user_goal}

# Conversation
{conversation_history}

# Analysis Criteria
1. Register and Tone
   - What is the user's register? (e.g., formal, casual, conversational, terse,
       professional)
   - What is the user's emotional tone? (e.g., neutral, frustrated, enthusiastic,
       impatient, apologetic, deferential)
   - Does the register/tone shift across the conversation? (e.g., starts polite but
       becomes curt after errors)

2. Verbosity and Structure
   - How verbose are the user's messages? Are they minimal and compressed, or expansive
       and elaborated?
   - Does the user use formatting conventions? (e.g., bullet points, numbered lists, code
       blocks, markdown, all caps, punctuation patterns)
   - How does message length and structure change across turns?

3. Social Conventions
   - Does the user use greetings, pleasantries, expressions of gratitude, or sign-offs?
```

```
   - Does the user treat the agent as a tool (purely transactional) or as a social
       interlocutor (rapport-building, politeness, humor)?

4. Request Framing
   - How does the user syntactically frame their requests? (e.g., imperative commands,
       questions, hedged suggestions, statements of problems without explicit asks)

# Instructions
- Generate terse, concise bullet points, not full sentences.
- Abstract away from the specific topic/domain.
- IMPORTANT: Do NOT use task-specific terms (e.g., "coding," "booking," "Python", "CSV").
    Use generic substitutes (e.g., "executing a task," "providing constraints," "the
    target artifact").

Output a valid JSON object using the exact format below. Do not include any text outside
    the JSON.

{{
   "register_and_tone": {{
      "register": "what is the user's register? (formal, casual, conversational, terse,
          professional, etc.)",
      "emotional_tone": "what is the user's emotional tone? (neutral, frustrated,
          enthusiastic, impatient, apologetic, deferential, etc.)",
      "tone_shifts": "does the tone shift across the conversation, and in response to what
          ?"
   }},
   "verbosity_and_structure": {{
      "verbosity": "how verbose are the user's messages? minimal and compressed, or
          expansive and elaborated?",
      "formatting": "does the user use formatting conventions? (bullet points, numbered
          lists, code blocks, markdown, etc.)",
      "evolution": "how does message length and structure change across turns?"
   }},
   "social_conventions": {{
      "politeness_markers": "does the user use greetings, pleasantries, gratitude, or sign
          -offs?",
      "agent_relationship": "does the user treat the agent as a tool (transactional) or as
           a social interlocutor (rapport-building, politeness, humor)?"
   }},
   "request_framing": "how does the user syntactically frame their requests? (imperative
       commands, questions, hedged suggestions, problem statements without explicit asks,
        etc.)"
}}
```

User Behavior Description: DAMSL Dialog Acts Dimension

```
You are an expert analyzing user behaviour in human-AI conversations. Your task is to
    analyze a specific user utterance in a multi-turn conversation and provide a
    description of the user's behaviour.

# Conversation History
{conversation_history}

# Target Utterance
Read the conversation history for context, but your analysis MUST strictly apply ONLY to
    this utterance:
{target_user_utterance}


# Dialogue Act Markup in Several Layers (DAMSL) Annotation Framework
DAMSL defines utterances based on the intentions of the speaker. Each utterance is
    analyzed along several dimensions.
1. INFORMATION LEVEL: Characterizes the semantic content of the utterance:
    - **Task**: Directly advances (or attempts to advance) the goals of the domain task.
    - **Task-management**: Addresses the problem-solving process/procedure rather than
        performing the task itself.
    - **Communication-management**: Concerned exclusively with maintaining the
        communication channel.
2. FORWARD LOOKING FUNCTION: Characterizes what effect the utterance has on the subsequent
     dialog:
    - **Statement**: Makes a claim (Assert, Re-assert, Other-statement).
    - **Info-request**: Asks for information.
    - **Influencing-addressee-future-action**: Directly influences the addressee's future
        non-communicative actions, directs or suggests the addressee to perform an action
        .
    - **Committing-speaker-future-action**: Potentially commits the speaker to some future
         action.
```

```
    - **Conventional**: Conventional conversational actions such as greeting, farewells,
        thanking, or responding to thanks.
3. BACKWARD LOOKING FUNCTION: Characterizes how the current utterance relates to the
    previous discourse:
    - **Agreement**: Accept, reject, partial accept/reject, holds off a response.
    - **Answer**: Provides requested information.
    - **Understanding**: Signals comprehension (acknowledgement, repetition/reformulation,
        collaborative completion) or lack thereof.


# Output Instructions
Analyze the target utterance along DAMSL's dimensions and write a concise, 1 sentence
    description of the user's behaviour and its functional role in the target utterance:
- Generate terse, concise bullet points, not full sentences.
- Abstract away from the specific topic/domain
- IMPORTANT:Do NOT use task-specific terms (e.g., "coding," "booking," "Python", "CSV").
    Use generic substitutes (e.g., "executing a task," "providing constraints," "the
    target artifact").

Output only the concise description as plain text, without any other text or formatting.
```

User Behavior Description: SGD Dialog Acts Dimension

```
You are an expert analyzing user behaviour in human-AI conversations. Your task is to
    analyze a specific user utterance in a multi-turn conversation and classify the user'
    s dialogue act(s).

# Conversation History
{conversation_history}

# Target Utterance
Read the conversation history for context, but your analysis MUST strictly apply ONLY to
    this utterance:
{target_user_utterance}


# Dialogue Act Taxonomy (adapted from Schema-Guided Dialogue)
Classify the target utterance using one or more of the following user dialogue acts:

1. INFORM: User provides information, states preferences, or supplies constraints to the
    agent.
2. REQUEST: User asks the agent for specific information or details.
3. AFFIRM: User agrees with, confirms, or accepts a proposition, value, or suggestion made
     by the agent.
4. NEGATE: User disagrees with, denies, or rejects a proposition, value, or suggestion
    made by the agent.
5. SELECT: User chooses between multiple options or alternatives presented by the agent.
6. INFORM_INTENT: User states, introduces, or shifts to a new goal or task objective.
7. AFFIRM_INTENT: User confirms or agrees with a goal or task objective suggested by the
    agent.
8. NEGATE_INTENT: User rejects or denies a goal or task objective suggested by the agent.
9. REQUEST_ALTS: User asks for different or additional options beyond what has been
    presented.
10. THANK: User expresses gratitude or appreciation.
11. GOODBYE: User signals the end of the conversation or a conversational closing.
12. GREET: User initiates the conversation with a greeting or opening.


# Output Instructions
- Assign ALL applicable dialogue acts to the target utterance (utterances may have
    multiple acts).

Output the dialogue acts, separated by commas, without any other text or formatting.
```

Embedding and quantization. For each conversation, we concatenate the descriptions across all six behavioral dimensions and embed the resulting text using Qwen3-Embedding-8B. We retain the first 1,024 embedding dimensions and apply ℓ_2 normalization. For each comparison between original and simulated conversations, we pool the two set of embeddings and apply PCA to retain 90% of the variance. We then quantize the embeddings using k-means with $k = 500$ clusters, using five restarts with up to 500 iterations each and retaining the solution with the lowest clustering objective. The cluster counts are used to create discrete distributions for the original and simulated user behaviors.