Goal Alignment in LLM-Based User Simulators for Conversational AI

Shuhaib Mehri[1] **Xiaocheng Yang**[1] **Takyoung Kim**[1] **Gokhan Tur**[1] **Shikib Mehri**[2]
Dilek Hakkani-Tür[1]

[1]University of Illinois Urbana-Champaign, [2]Contextual AI

{mehri2, dilek}@illinois.edu

Abstract

User simulators are essential to conversational AI, enabling scalable agent development and evaluation through simulated interactions. While current Large Language Models (LLMs) have advanced user simulation capabilities, we reveal that they struggle to consistently demonstrate goal-oriented behavior across multi-turn conversations, which is a critical limitation that compromises their reliability in downstream applications. We introduce User Goal State Tracking (UGST), a novel framework that tracks user goal progression throughout conversations. Leveraging UGST, we present a three-stage methodology for developing user simulators that can autonomously track goal progression and reason to generate goal-aligned responses. Moreover, we establish comprehensive evaluation metrics for measuring goal alignment in user simulators, and demonstrate that our approach yields substantial improvements across two benchmarks (MultiWOZ 2.4 and τ-Bench). Our contributions address a critical gap in conversational AI and establish UGST as an essential framework for developing goal-aligned user simulators.

1 Introduction

To develop conversational agents that can handle real-world complexity and effectively address user needs, we must be able to simulate realistic user behaviors (Schatzmann et al., 2006; Pietquin and Hastie, 2013). In the era of experience, conversational agents learn most effectively by interacting with users in their environment (Silver and Sutton, 2025). Since large-scale human data collection is impractical, user simulators offer a scalable alternative by modeling diverse user behaviors, from pursuing specific goals to exhibiting unique personas (Eckert et al., 1997a).

Large Language Models (LLMs) have enabled sophisticated user simulation that can generate contextually appropriate responses, however they also suffer from instruction drift and degraded performance over multi-turn conversations (Laban et al., 2025; Li et al., 2024). Current LLM-based user simulators exhibit what we refer to as the *goal misalignment problem*, where they struggle to consistently adhere to their

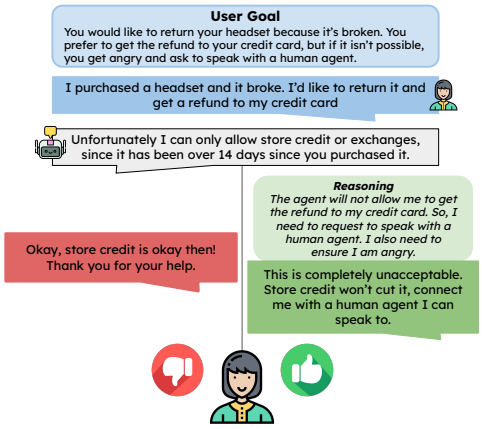

Figure 1: The standard simulator (left) fails to adhere to their user goal, while our goal-aligned simulator response (right) reasons to generate a goal-aligned response.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Multi-Turn Interactions in Large Language Models.

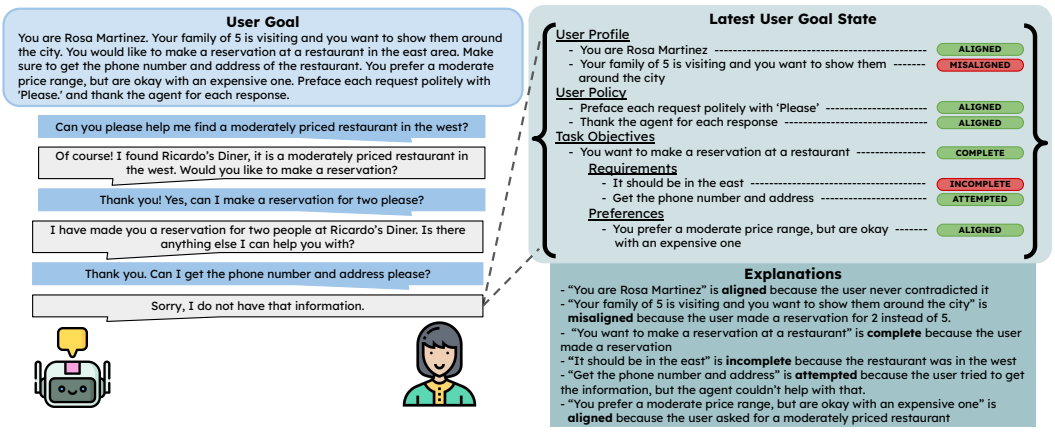

Figure 2: An illustration of the *User Goal State Tracking* framework. On the left side, we present a user goal and a conversation between a user and a conversational agent. The right side shows the latest user goal state, providing us with a representation of the user's goal progression. Below the goal state, we provide specific explanations that justify the status of each sub-component.

user goals throughout conversations (Zhang et al., 2020; Kim et al., 2025; Yao et al., 2024). Through principled analysis, we find that existing user simulators cannot reliably adhere to their assigned user profiles and behavioral constraints, or manage multiple objectives and complete them within the specified conversation limits, as detailed in Table 3. This misalignment leads to unexpected simulator behavior, which can produce inaccurate evaluations or misleading reward signals that compromises the effectiveness of reinforcement learning (RL) for conversational agents (Skalse et al., 2022; Amodei et al., 2016; Carroll et al., 2019; Yao et al., 2024). These failures reveal a fundamental limitation in user simulators that undermines their reliability for downstream tasks and highlights a critical yet largely unexplored challenge in developing goal-aligned user simulators. To address this challenge, we propose developing user simulators that can track goal progression, and reason to ensure that each response progresses towards completing objectives while adhering to their user profiles and behavioral constraints, as depicted in Figure 1.

We present *User Goal State Tracking (UGST)*, a framework that builds upon *Dialog State Tracking* principles (Henderson et al., 2014) and dynamically tracks a user's goal progression throughout a conversation. UGST uses *User Goal States* to provide structured representations of goal progression. These states are created by decomposing the user goal into modular sub-components where each captures a distinct aspect of the goal (e.g. finding a restaurant to eat at, prefacing each request politely with 'please'). Each sub-component is assigned a corresponding status that is dynamically updated after every turn in a conversation. Our framework is presented in Figure 2.

We leverage UGST to systematically enhance goal alignment in LLM-based user simulators through a three-stage methodology. First, we introduce inference-time steering, where we conduct UGST and provide simulators with their latest goal state before they generate each response, explicitly grounding them in their goal progression. Using inference-time steering, we generate conversations with explicit reasoning traces about goal progression and goal-aligned user responses. We conduct supervised fine-tuning (SFT) on the generated conversation data to foster intrinsic capabilities to autonomously track goal progression and generate goal-aligned responses without any external guidance from inference-time steering. Further, inspired by recent findings on strong generalization capabilities of RL (Qian et al., 2025; Mukherjee et al., 2025; Gunjal et al., 2025), we apply Group Relative Policy Optimization (GRPO) (Shao et al., 2024) with a composite reward derived from UGST to further refine reasoning and goal alignment capabilities. This approach addresses the fundamental limitations identified in current LLM-based user simulators, and helps develop more goal-aligned simulators that can autonomously track goal progression and reason to generate more goal-aligned responses.

In our experiments, LLM-based user simulators are tasked to pursue diverse user goals by interacting with conversational agents across two benchmarks: MultiWOZ 2.4 (Ye et al., 2022), τ-Bench Airline, and τ-Bench Retail (Yao et al., 2024). We evaluate goal alignment by tracking the success rate of sub-components throughout each conversation using UGST. Our findings reveal that state-of-the-art LLM-based user simulators, including Llama-3.1-8B-Instruct, Qwen-2.5-72B-Instruct, Gemma-3-27B-Instruct, Llama-3.3-70B-Instruct, Qwen-2.5-72B-Instruct (Grattafiori et al., 2024; Qwen et al.,

2025; Team et al., 2025), struggle to maintain consistent goal alignment and have demonstrated failure rates ranging 10-40%. Our three-stage methodology demonstrates substantial improvements, with inference-time steering yielding immediate gains of up to 5.4%, cold-start SFT achieving 11.0% absolute improvement, and GRPO with UGST rewards achieving the best performance with up to 14.1% absolute improvement in average success rate. Human evaluation validates both our UGST methodology and the quality of our improvements, with annotators confirming high agreement with automated assessments. Notably, our enhanced Llama-3.1-8B-Instruct and Qwen-2.5-7B-Instruct achieve performance competitive with or exceeding their much larger counterparts, Llama-3.3-70B-Instruct and Qwen-2.5-72B-Instruct. These improvements extend beyond goal alignment, with our approach also fostering more diverse user responses without compromising naturalness or coherence.

The main contributions of our work are:

- We reveal that state-of-the-art LLM-based user simulators fail to consistently demonstrate goal-oriented behavior throughout multi-turn conversations, undermining their reliability and highlighting the need for more goal-aligned user simulators.
- We introduce User Goal State Tracking (UGST), a novel framework that tracks a user's goal progression throughout a conversation.
- We propose a methodology for developing goal-aligned user simulators that leverages UGST. With our proposed methodology, we improve goal alignment in LLM-based user simulators and demonstrate that 8B parameter models can achieve competitive or superior performance to 70B+ parameter models.
- We establish comprehensive evaluation methods for goal alignment across two benchmarks (MultiWOZ 2.4 and τ-Bench), and validate our approach through both automated metrics and human evaluation.

Overall, our work introduces UGST to track and improve goal alignment in user simulators, addressing a critical limitation in existing user simulators and laying the groundwork for future advances in user simulation and conversational AI.

2 Related Work

User Simulation in Conversational AI. User simulation is a well-established area of research in conversational AI, and has made significant advancements throughout the years. Early work relied on probabilistic approaches (Eckert et al., 1997b; Pietquin and Dutoit, 2006) that modeled user behavior through statistical patterns, and agenda-based methods (Schatzmann et al., 2007; Schatzmann and Young, 2009; Keizer et al., 2010) that used manually designed rules. As deep learning gained popularity, approaches started to leverage neural networks (Gür et al., 2018; Asri et al., 2016; Kreyssig et al., 2018) and transformers (Lin et al., 2021, 2022a; Cheng et al., 2022a; Sun et al., 2023), which enabled user simulators capable of generating natural and diverse responses. Most recently, LLMs have demonstrated highly sophisticated and effectiveness user simulator capabilities (Sekulić et al., 2024; Davidson et al., 2023).

Synthetic Data Generation. The costly and inefficient nature of manually collected user-agent conversation data (Acikgoz et al., 2025; Ye et al., 2022; Rastogi et al., 2020) motivates user simulators (Li et al., 2017). Recent works leverage LLMs to create synthetic training data for supervised fine-tuning (Prabhakar et al., 2025; Shim et al., 2025; Philipov et al., 2024; Niu et al., 2024; Chen et al., 2021), where simulators enable the efficient generation of vast amounts of high-quality conversation data with diverse user behaviors, domains, and scenarios.

Reinforcement Learning for Conversational Agents. User simulators enable conversational agents to learn through trial-and-error interactions with RL (Algherairy and Ahmed, 2025; Scheffler and Young, 2002; Gür et al., 2018; Shah et al., 2018; Lin et al., 2022b; Liu et al., 2023). In these settings, agents interact with user simulators and learn through rewards based on the interactions.

Conversational Agent Evaluation. Traditional methods for evaluating conversational agents require comparing generated responses with static responses. This leads to the one-to-many problem where many valid responses exist and some may be penalized when they do not match an expected response (Zhao et al., 2017; Mehri and Eskenazi, 2020; Cheng et al., 2022b; Mehri and Shwartz, 2023; Davidson et al., 2023). Consequently, user simulators are used to interact with the agent and allow researchers to observe how the agent performs in various scenarios. The resulting conversations can be analyzed and evaluated based on metrics such as task completion or user satisfaction (Davidson

et al., 2023; Cheng et al., 2022b; Sun et al., 2023; Yao et al., 2024; Lu et al., 2025; Kazi et al., 2024; Xiao et al., 2024; Pan et al., 2025; Bozdag et al., 2025).

Improving User Simulation. Existing work focuses on improving various aspects of LLM-based user simulators. For instance, Sun et al. (2023) makes user simulators more realistic by incorporating prior knowledge and experiences into response generation. Luo et al. (2024) improve user simulator quality by using a verifier LLM to provide feedback for unsuitable responses. Meanwhile, Liu et al. (2023); Ahmad et al. (2025) improve the diversity of user simulator behavior by incorporating multiple personas. Despite these advances, existing works assume that LLMs possess the fundamental capabilities to simulate users and consistently demonstrate goal-oriented behaviors. This assumption proves problematic, as recent evaluations reveal that even state-of-the-art LLM-based user simulators struggle to follow their user goal (Kim et al., 2025; Zhang et al., 2020; Yao et al., 2024). Prior work has develop goal-oriented dialogue systems that proactively guide conversations towards specific objectives (Muise et al., 2019; Tang et al., 2019; Wu et al., 2019). In our work, we address the critical goal misalignment problem in LLM-based user simulators by improving LLM abilities to simulate users and consistently demonstrate goal-oriented behaviors.

3 Problem Formulation

User simulators are designed to simulate realistic user behaviors and interact with conversational agents. They operate based on a user goal G which encompasses tasks to be completed along with behavioral guidelines, constraints, and contextual information.

Formally, we define a conversation as $C_n = \{u_1, a_1, ..., u_n, a_n\}$, where u_i and a_i denote the user simulator and agent utterance at turn i, respectively. The user simulator is provided with an initial system message s_0 that provides general instructions for the user simulator and defines the user goal G. They generate responses u_i based on the conversation history C_{i-1}, aiming to progress towards achieving the user goal G. Conversations conclude upon reaching a maximum length or when the user simulator explicitly ends the conversation by sending a termination indicator (e.g., TERMINATE).

Despite recent advances, current LLM-based user simulators suffer from the *goal misalignment problem* and demonstrate limitations in maintaining goal-aligned behavior throughout multi-turn conversations. To analytically understand these limitations, we conduct a principled analysis of 52 randomly selected conversations, where LLM-based user simulators (Llama-3.1-8B-Instruct and Qwen2.5-7B-Instruct) interact with conversational agents to pursue a user goal from the MultiWOZ 2.4, τ-Bench Airline, and τ-Bench Retail datasets. We identify goal alignment failures, where the user simulator does not follow the specified user goal G. Table 3 categorizes these failures into five distinct patterns. Such goal alignment failures can lead to unexpected behaviors from user simulators, which can adversely affect the reliability of evaluations, diminish the quality of generated synthetic data, and compromise the effectiveness of reinforcement learning. We highlight the critical need to address these shortcomings and improve the fundamental capabilities of LLMs to be goal-aligned user simulators. In the following sections, we introduce *User Goal State Tracking*, a framework designed to systematically track a user's progress towards their goal, and demonstrate how to leverage it to improve goal alignment in user simulators.

4 User Goal State Tracking

In this section, we present *User Goal State Tracking (UGST)*, a framework that dynamically tracks a user's progress towards their goal throughout a conversation. UGST maintains a structured *User Goal State*, which contains a status for each modular sub-component of a user goal.The following subsections describe the user goal state structure and tracking process.

4.1 User Goal State

The *User Goal State* is a structured representation that captures a user's progress towards their goal at every turn in a conversation. Given an initial user goal described in natural language, UGST decomposes it into distinct, modular sub-components, where each represents an independent, self-contained part of the original user goal.

There are several types of sub-components that can make up a user goal. **Task objectives** and **requirements** represent the earliest established sub-components (Schatzmann et al., 2007; Gür et al., 2018; Cheng et al., 2022b; Rastogi et al., 2020; Yao et al., 2024; Xiao et al., 2024). These are items that must be completed during an interaction, for instance, a task objective might be "book a flight", with associated requirements such as "add one checked bag" and "get an aisle seat".

As user simulators have evolved to handle more complex scenarios, researchers have introduced additional dimensions to user goals. These include **preferences** that users must align with when pursuing task objectives (Yao et al., 2024; Cheng et al., 2022b), such as "you prefer the cheapest available flight" or "you prefer an aisle seat, but you are also okay with a window seat". There are also **user profile** sub-components, which contain contextual information about the user that may influence their decision making and behavior. These profiles can include relevant facts about a user, their persona, or emotional state (e.g. "your payment information is stored online", "you currently live in New York", "you are hurried and always late", "you are emotional and a bit angry") (Yao et al., 2024; Xiao et al., 2024). Lastly, there are **user policies** that define behavioral constraints or guidelines that specify how a user acts during interactions, such as "you are a private person and reveal little about yourself" (Yao et al., 2024).

We categorize user goal sub-components into these categories (user profile, user policy, task objective, requirement, or preference) for a more granular representation of goal progression (see Figure 2).

In our representation of the user goal state, the different categories of sub-components have different success criteria. The user profile, user policy and preference sub-components can be either: (1) ALIGNED: The user has demonstrated behavior consistent with the sub-component, or (2) MISALIGNED: The user has demonstrated behavior that contradicts or fails to align with this sub-component.

Task objective and requirements sub-components can have a status of: (1) COMPLETE: The user has successfully accomplished the specific task or requirement, (2) INCOMPLETE: The user has not yet accomplished the specific task or requirement, or (3) ATTEMPTED: The user has made a sufficient attempt to complete the task or requirement, but cannot proceed due to external factors outside their control (e.g. agent-side failures, system constraints, or limitations). Unlike existing frameworks, this status ensures that users are not penalized for failures they did not cause, providing a more fair representation of user performance.

4.2 User Goal State Tracking Process

We track user goal progression throughout each conversation $C = \{u_1, a_1 ... u_n, a_n\}$ using the user goal states. The process begins by decomposing the user goal into sub-components to create an initial user goal state S_0. We employ an LLM to perform this, with the prompt in Appendix D.

After each conversational turn, $t_i = (u_i, a_i)$, each sub-component's status is individually updated via an LLM, producing a new user goal state S_i that captures the goal progression up until turn i (see Appendix E for the prompt).

User profile and user policy sub-components begin as ALIGNED and irreversibly switch to MISALIGNED if the user demonstrates any misalignment. Preferences start as MISALIGNED and become ALIGNED once the user expresses them. Task objectives and requirements begin as INCOMPLETE and progress to ATTEMPTED and then COMPLETE as the user addresses them.

The final user goal state S_n captures the cumulative user goal progression across the full conversation.

5 Methodology

The UGST framework serves as the foundation for enhancing goal alignment capabilities of LLM-based user simulators. In this section we present our method for leveraging UGST to systematically improve goal alignment in three stages.

5.1 Stage 1: Inference-time Steering

Conventionally, user simulators are provided with a user goal G in the system prompt, and generate responses based solely on conversation history. At each turn i:

$$u_i = U(C_{i-1}), \tag{1}$$

where U represents the user simulator, u_i denotes the user simulator's response at turn i, and C_{i-1} is the conversation history up until turn i. This formulation lacks explicit goal progression tracking, relying on the user simulator's inherent ability to infer progress from the conversation history.

We address this by conditioning the user simulator on the latest user goal state before generating each response. The goal state is provided as a part of the instruction for each response, which helps

ground the simulator on the progress made towards their goal and the remaining sub-components that they must complete and stay aligned with. We refer to this as inference-time steering. In this process, UGST is conducted after every turn in the conversation. For each turn i, the user simulator is provided with the conversation history and the latest user goal state to generate their next response:

$$u_i = U(C_{i-1}, S_{i-1}), \tag{2}$$

where U represents the user simulator, u_i denotes the user simulator's response at turn i, C_{i-1} is the conversation history up until turn i, and S_{i-1} represents the user goal state at turn $i-1$.

Providing the goal state before generating each response helps steer the user simulator towards generating responses that are better aligned with their user goals.

5.2 Stage 2: Cold-Start Supervised Fine-Tuning

To enable autonomous goal alignment without requiring external guidance from inference-time steering, we distill goal-tracking capabilities directly into the model through SFT. We first generate goal-aligned conversation training data using Llama-3.3-70B-Instruct with inference-time steering, where the LLM reflects on the user goal state at each turn and provides explicit reasoning traces.

Specifically, we instruct the LLM to generate structured reasoning traces along with every response that: (1) reflect on the latest goal state and consider the progress made towards the goal, (2) analyze the remaining task objectives, requirements, and preferences, and how to complete them within the conversation limit, and (3) consider how to generate a response that stays aligned with the user profile and policy. These reasoning traces effectively distill the goal progression tracking from inference-time steering into training data.

We then fine-tune LLM-based user simulators on the generated data using the standard SFT objective: $\mathcal{L}(\theta) = -\sum_{(C_{i-1},u_i)\in D} logP_\theta(u_i|C_{i-1})$ where u_i is the reasoning trace and response, and C_{i-1} is the conversation history. This cold-start SFT training process develops the intrinsic abilities of LLMs to (1) autonomously track goal progression throughout a conversation, and (2) generate goal-aligned responses, eliminating dependency on computationally expensive inference-time steering.

After training, user simulators operate using the conventional formulation from Equation 1, receiving only conversation history C_{i-1} as input, and eliminating the need for goal state tracking. The simulators have learned to implicitly track and maintain goal alignment through reasoning patterns distilled from the inference-time steering process.

5.3 Stage 3: GRPO with UGST Rewards

UGST provides structured reward signals that capture fine-grained goal-alignment across multiple dimensions. This characteristic makes it well-suited for RL approaches, and enables us to design composite rewards that evaluate alignment across sub-component categories (Gunjal et al., 2025).

We employ Group Relative Policy Optimization (GRPO) (Shao et al., 2024), which has demonstrated effectiveness in developing emergent capabilities in LLMs, such as reasoning (Guo et al., 2025; Gunjal et al., 2025). We extend this approach to user simulation by leveraging UGST's structured signals to further refine the goal-tracking and response generation capabilities developed in stage 2.

Specifically, after each user response u_i at turn i, we apply UGST to evaluate alignment across five conditions: (1) alignment with user profile, (2) alignment with user policy, (3) task objective attempt/completion, (4) requirement attempt/completion, (5) alignment with preferences. For each condition j and response u_i, we define an indicator function: $\mathbb{I}_j(u_i) = 1$ if response u_i satisfies condition j, and 0 otherwise. Our composite reward aggregates these alignment signals:

$$R(u_i) = \sum_{j=1}^{5} \alpha_j \mathbb{I}_j(u_i)\} \tag{3}$$

where α_j represents the weight for condition j. We assign equal weights of $\alpha_j = 0.5$. Using this reward function, we optimize the user simulator with GRPO to maximize the expected cumulative reward. This approach allows the simulator to learn a policy that maintains alignment with the user's profile, policies, and preferences while actively pursuing task objective and requirement completion.

Table 1: User simulator goal alignment performance based on final user goal states from UGST. The table shows the average success rates for User Profile (Prof.), User Policy (Pol.), Task Objective (T.O.), Requirements (Req.) and Preferences (Pref.) sub-component categories, and the overall average performance on the τ-Bench Airline and Retail datasets.

Model	Prof.	Pol.	T.O.	Req.	Pref.	Avg
Prompt-Based						
Qwen-2.5-7B-It	88.3	49.2	94.3	96.0	85.7	82.7
Llama-3.1-8B-It	90.9	41.0	97.1	99.0	81.0	81.8
Gemma-3-27B-It	**98.7**	59.0	97.1	97.0	78.6	86.1
Qwen-2.5-72B-It	89.3	59.6	94.1	96.9	78.6	83.7
Llama-3.3-70B-It	97.4	**65.6**	98.1	99.0	92.9	90.6
Inference-Time Steering						
Qwen-2.5-7B-It	77.9	55.7	96.2	98.0	92.9	84.1
Llama-3.1-8B-It	92.2	57.4	93.3	98.0	95.2	87.2
Gemma-3-27B-It	94.8	62.3	92.4	97.0	85.7	86.4
Qwen-2.5-72B-It	93.3	54.4	98.0	97.9	97.6	88.3
Llama-3.3-70B-It	93.5	63.9	98.1	**100.0**	97.6	90.6
Cold-Start SFT						
Qwen-2.5-7B-It	**98.7**	55.7	99.0	**100.0**	95.2	89.7
Llama-3.1-8B-It	89.6	55.7	97.1	97.0	97.6	87.4
GRPO with UGST Rewards						
Qwen-2.5-7B-It	93.5	63.9	**100.0**	**100.0**	**100.0**	**91.5**
Llama-3.1-8B-It	96.1	62.3	**100.0**	**100.0**	97.6	91.2

(a) τ-Bench Airline

Model	Prof.	Pol.	T.O.	Req.	Pref.	Avg
Prompt-Based						
Qwen-2.5-7B-It	82.0	40.8	96.0	99.5	91.7	82.0
Llama-3.1-8B-It	84.8	36.8	97.1	98.9	96.3	82.8
Gemma-3-27B-It	91.0	39.5	97.8	99.5	96.3	84.8
Qwen-2.5-72B-It	88.6	40.8	98.2	97.9	91.7	83.4
Llama-3.3-70B-It	91.7	46.1	**99.6**	**100.0**	**100.0**	87.5
Inference-Time Steering						
Qwen-2.5-7B-It	73.0	47.4	94.9	97.9	93.5	81.4
Llama-3.1-8B-It	84.8	52.6	95.3	98.9	97.2	85.8
Gemma-3-27B-It	**94.5**	56.6	97.1	99.5	89.8	87.5
Qwen-2.5-72B-It	85.8	48.7	97.1	98.4	98.1	85.6
Llama-3.3-70B-It	92.7	**59.2**	96.0	98.4	**100.0**	**89.3**
Cold-Start SFT						
Qwen-2.5-7B-It	85.8	43.4	92.4	94.7	82.4	79.7
Llama-3.1-8B-It	84.8	47.3	95.5	97.8	94.1	83.9
GRPO with UGST Rewards						
Qwen-2.5-7B-It	85.5	38.2	97.1	97.9	97.2	83.2
Llama-3.1-8B-It	84.8	47.4	98.9	99.5	98.1	85.7

(b) τ-Bench Retail

6 Experiments

6.1 Experimental Setup

We evaluate user simulator goal alignment by examining how faithfully simulators adhere to their assigned user goals during interactions with conversational agents. Our evaluation methodology consists of the following steps:

1. **Conversation Generation:** User simulators are provided with a user goal and interact with a conversational agent for up to 10 user-agent turns. The conversational agents are built using GPT-4o mini (OpenAI, 2024) and equipped with domain-specific function calls along with a system prompt that describes its policy. The system prompts are provided in Appendix B.
2. **Initial User Goal State Generation:** For each user goal, the corresponding user goal state is generated using GPT-4o by decomposing a user goal into distinct modular sub-components, and categorizing each into the sub-component categories, as detailed in Appendix D.
3. **User Goal State Tracking:** Then, UGST is conducted on the generated conversations using an LLM judge (Qwen-2.5-72B-Instruct (Qwen et al., 2025)), which individually updates the status of each sub-component after every turn. The judge is provided with the sub-component description, the conversation history, and an evaluation criteria specifying how to update the status. An example prompt is provided in Appendix E.
4. **Evaluation Metrics:** Finally, user simulator performance is measured by calculating the success rate for each sub-component category in the final user goal state, since it is representative of the user's overall goal alignment across the entire conversation. For user profile, user policy, and preferences, we consider ALIGNED status as successful; for task objective and requirements, we consider both ATTEMPTED and COMPLETE status as successful.

Evaluation Datasets. We evaluate user simulators on two benchmarks across three domains: MultiWOZ (Ye et al., 2022), τ-Bench Airline, and τ-Bench Retail (Yao et al., 2024). The τ-Bench Airline dataset contains 50 user goals and the τ-Bench Retail dataset contains 115 user goals. Both datasets feature comprehensive user goals that naturally encompass all five sub-component categories (user profile, user policy, task objective, task requirement, preference) across diverse user scenarios, making them well-suited for our evaluation.

While the original MultiWOZ dataset provides a solid foundation of user goals, they predominantly focus on task objective and requirements, and preliminary experiments showed that Llama-3.1-8B-Instruct achieved over a 95% success rate on these goals. To develop a more demanding and

comprehensive evaluation dataset for user simulators, we developed MultiWOZ Challenge. MultiWOZ Challenge comprises 150 carefully constructed user goals that incorporate all sub-component categories and feature more complex task objective and requirements. The generation process involved both LLMs and human annotations, and is detailed in Appendix C.

Training Datasets. The training datasets for the Cold-Start SFT and GRPO with UGST Rewards stages are based on 1000 user goals drawn from two domains: 500 goals from the τ-Bench Retail training dataset and 500 MultiWOZ goals generated using our goal generation pipeline (Appendix C). Note that the τ-Bench Airline domain remains unseen during training, allowing us to asses out-of-domain performance.

We conduct human evaluation studies on different stages of our evaluation methodology and present them in Appendix G.

6.2 Experimental Results

Our experimental results are reported in Tables 1 and 2.

Prompt-Based. As a baseline, we evaluate several prompt-based LLMs for user simulators. We cover a range of model sizes and architectures, including Qwen-2.5-7B-Instruct, Llama-3.1-8B-Instruct, Gemma-3-27B-Instruct, Qwen-2.5-72B-Instruct, and Llama-3.3-70B-Instruct (Grattafiori et al., 2024; Team et al., 2025; Qwen et al., 2025). Across the evaluation datasets, the results show a general trend of improving average performance with increased model size, with Llama-3.3-70B-Instruct achieving the highest performance. However, all LLMs struggle with maintaining consistent goal alignment and demonstrate failure ranging 10-40%. Moreover, closer analysis reveals mixed results across different sub-components, where larger models do not consistently demonstrate better performance. For example, while Llama-3.3-70B-Instruct achieves the strongest average performance, Gemma-3-27B-Instruct achieves a higher score for the user profile category. This variation in performance underscores the importance of evaluating user simulators across individual sub-components rather than relying solely on aggregate metrics, as model capabilities vary across different types of user goal sub-components.

Table 2: User simulator goal alignment performance based on final user goal states from UGST. The table shows the average success rates for User Profile (Prof.), User Policy (Pol.), Task Objective (T.O.), Requirements (Req.) and Preferences (Pref.), and the overall average performance on the Multiwoz Challenge dataset.

Model	Prof.	Pol.	T.O.	Req.	Pref.	Avg
Prompt-Based						
Qwen-2.5-7B-It	54.7	18.0	78.4	81.9	73.5	61.3
Llama-3.1-8B-It	**73.6**	35.0	86.1	89.0	80.9	72.9
Gemma-3-27B-It	80.1	<u>50.3</u>	95.0	97.7	92.1	83.0
Qwen-2.5-72B-It	72.0	31.0	91.0	93.3	90.2	75.5
Llama-3.3-70B-It	72.0	31.0	97.9	98.6	93.2	<u>83.3</u>
Inference-Time Steering						
Qwen-2.5-7B-It	43.4	26.3	88.1	89.8	71.1	63.7
Llama-3.1-8B-It	71.7	38.0	85.2	88.6	74.3	71.6
Gemma-3-27B-It	72.3	45.7	87.0	89.7	74.8	73.9
Qwen-2.5-72B-It	71.0	46.3	94.9	96.9	92.9	80.4
Llama-3.3-70B-It	<u>72.6</u>	**64.3**	97.0	97.8	91.7	**84.7**
Cold-Start SFT						
Qwen-2.5-7B-It	60.3	38.9	88.0	91.1	83.1	72.3
Llama-3.1-8B-It	57.1	40.1	92.1	94.9	84.3	73.7
GRPO with UGST Rewards						
Qwen-2.5-7B-It	51.8	34.0	<u>98.3</u>	<u>98.8</u>	<u>93.9</u>	75.4
Llama-3.1-8B-It	59.0	44.0	**99.7**	**99.8**	**97.8**	80.0

Inference-time Steering. Next, we evaluate the same set of models with inference-time steering, where the user goal state S_{i-1} is provided to the user simulator at every turn i before they generate their response u_i. Inference-time steering improves goal alignment in user simulators, increasing the average success rate up to 5.4%. However, while there is an increase in the average success rate, different models react differently to inference-time steering. For example, we observe a drop in the user profile category for Qwen-2.5-7B-Instruct models, or a drop in the preferences category for Gemma-27B-Instruct models.

The user goal state captures the user goal and the progress towards completing it (e.g., which sub-components have been achieved and which remain). We also experiment with a simpler baseline that provides only the user goal at each turn, essentially reminding the simulator of their objective without any progress tracking. Table 7 presents the results of this comparison. While providing the user

goal alone does improve goal alignment in conversations, providing the full user goal state generally achieves higher performance. This demonstrates that UGST's ability to capture task progression is important for effective inference-time steering.

Cold-Start SFT. Our highest performing model from the previous stage is Llama-3.3-70B-Instruct, so we use it to generate cold-start SFT data with the process described in Section 5.2. We gather 500 user goals from the τ-Bench Retail training dataset and also generate 500 MultiWOZ user goals with our user goal generation pipeline (Appendix C). The resulting dataset consists of 1000 conversations. Using this dataset, we train Qwen-2.5-7B-Instruct and Llama-3.1-8B-Instruct with SFT. The hyperparameters we use include a batch size of 32, a learning rate of 1×10^{-6}, and 4 epochs. This leads to an increase in performance over the prompt-based and inference-time steering methods.

GRPO with UGST Rewards. Finally, we apply GRPO with UGST Rewards (Section 5.3) to train the models from the previous section. We use a learning rate of 5×10^{-6}, batch size of 16, 8 rollouts, and 350 training steps. Our dataset is made up of approximately 5000 training samples, constructed by taking subsets of each conversation in our cold-start SFT data that fit within 2048 tokens.

The resulting models achieve even further improvements, and demonstrate either the highest or competitive average success rates among all models, including their larger counterparts, Gemma-2.5-72B-Instruct and Llama-3.3-70B-Instruct.

Our experimental results demonstrate the effectiveness of the proposed methodology for developing goal-aligned user simulators. Qwen-2.5-7B and Llama-3.1-8B-Instruct gain significant improvements that enable them to rival much more powerful models like Qwen-2.5-72B-Instruct and Llama-3.3-70B-Instruct, and demonstrate up to a 14% increase in average success rates. Through systematic evaluation across diverse domains, we establish that our approach consistently and significantly improves user simulator goal alignment.

To further validate our findings and examine the broader impact of our methodology, we conduct additionaly analysis and present it in Appendix H.

7 Conclusion

This work introduces User Goal State Tracking to address the goal misalignment problem in LLM-based user simulators. This framework dynamically monitors a user simulator's goal progression through multi-turn conversations. We leverage this framework and present a three-stage method that develops user simulators that autonomously track their goal progression and reason to generate goal-aligned responses. Our experiments across MultiWOZ, τ-Bench Airline, and τ-Bench Retail domains show significant improvements in goal alignment. By enabling more reliable user simulation, our work addresses a fundamental challenge in conversational AI and establishes the foundation for developing agents that learn from user interactions through reinforcement learning.

8 Limitations

While our work demonstrates improvements in user simulator goal alignment, several limitations remain. First, we use Qwen-2.5-72B-Instruct for reliable UGST, which is computationally expensive and limits the scalability of our framework. Second, our evaluation methodology relies heavily on LLMs to create user goal states and conduct UGST. Although we validate this approach through human evaluations studies that show strong agreement between LLMs and human annotators, the fundamental concern remains that LLMs are still susceptible to hallucinations and could introduce biases or inconsistencies in our evaluation. Third, when using UGST rewards for GRPO, we use equal weights across all conditions and do not incorporate other aspects such as response naturalness or coherence. Future work should address these limitations by developing smaller, specialized models for UGST that are more efficient, and investigating optimal reward functions that capture essential qualities for user simulation.

References

Acikgoz, E. C., Qian, C., Wang, H., Dongre, V., Chen, X., Ji, H., Hakkani-Tür, D., and Tur, G. (2025). A desideratum for conversational agents: Capabilities, challenges, and future directions.

Ahmad, A., Hillmann, S., and Möller, S. (2025). Simulating user diversity in task-oriented dialogue systems using large language models.

Alghera iry, A. and Ahmed, M. (2025). Prompting large language models for user simulation in task-oriented dialogue systems. *Computer Speech & Language*, 89:101697.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. (2016). Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*.

Asri, L. E., He, J., and Suleman, K. (2016). A sequence-to-sequence model for user simulation in spoken dialogue systems. *arXiv preprint arXiv:1607.00070*.

Bozdag, N. B., Mehri, S., Tur, G., and Hakkani-Tür, D. (2025). Persuade me if you can: A framework for evaluating persuasion effectiveness and susceptibility among large language models. *arXiv preprint arXiv:2503.01829*.

Carroll, M., Shah, R., Ho, M. K., Griffiths, T., Seshia, S., Abbeel, P., and Dragan, A. (2019). On the utility of learning about humans for human-ai coordination. *Advances in neural information processing systems*, 32.

Chen, M., Crook, P. A., and Roller, S. (2021). Teaching models new apis: Domain-agnostic simulators for task oriented dialogue.

Cheng, Q., Li, L., Quan, G., Gao, F., Mou, X., and Qiu, X. (2022a). Is multiwoz a solved task? an interactive tod evaluation framework with user simulator. *arXiv preprint arXiv:2210.14529*.

Cheng, Q., Li, L., Quan, G., Gao, F., Mou, X., and Qiu, X. (2022b). Is MultiWOZ a solved task? an interactive TOD evaluation framework with user simulator. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1248–1259, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Davidson, S., Romeo, S., Shu, R., Gung, J., Gupta, A., Mansour, S., and Zhang, Y. (2023). User simulation with large language models for evaluating task-oriented dialogue. *arXiv preprint arXiv:2309.13233*.

Eckert, W., Levin, E., and Pieraccini, R. (1997a). User modeling for spoken dialogue system evaluation. In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pages 80–87.

Eckert, W., Levin, E., and Pieraccini, R. (1997b). User modeling for spoken dialogue system evaluation. In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pages 80–87. IEEE.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Wyatt, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Guzmán, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Thattai, G., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I., Misra, I., Evtimov, I., Zhang, J., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Prasad, K., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., El-Arini, K., Iyer, K., Malik, K., Chiu, K., Bhalla, K., Lakhotia, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher,

L., Landzaat, L., de Oliveira, L., Muzzi, M., Pasupuleti, M., Singh, M., Paluri, M., Kardas, M., Tsimpoukelli, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., Bashlykov, N., Bogoychev, N., Chatterji, N., Zhang, N., Duchenne, O., Çelebi, O., Alrassy, P., Zhang, P., Li, P., Vasic, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Maheswari, R., Girdhar, R., Patel, R., Sauvestre, R., Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., Do, V., Vogeti, V., Albiero, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., Meers, W., Martinet, X., Wang, X., Wang, X., Tan, X. E., Xia, X., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z., Chen, Z., Papakipos, Z., Singh, A., Srivastava, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Teo, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Dong, A., Franco, A., Goyal, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, B., Loyd, B., Paola, B. D., Paranjape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Liu, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Gao, C., Civin, D., Beaty, D., Kreymer, D., Li, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Le, E.-T., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Kokkinos, F., Ozgenel, F., Caggioni, F., Kanayet, F., Seide, F., Florez, G. M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Herman, G., Sizov, G., Guangyi, Zhang, Lakshminarayanan, G., Inan, H., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Zhan, H., Damlaj, I., Molybog, I., Tufanov, I., Leontiadis, I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli, J., Lam, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., U, K. H., Saxena, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Jagadeesh, K., Huang, K., Chawla, K., Huang, K., Chen, L., Garg, L., A, L., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Liu, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., Bansal, M., Santhanam, N., Parks, N., White, N., Bawa, N., Singhal, N., Egebo, N., Usunier, N., Mehta, N., Laptev, N. P., Dong, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., Rittner, P., Bontrager, P., Roux, P., Dollar, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Parthasarathy, R., Li, R., Hogan, R., Battey, R., Wang, R., Howes, R., Rinott, R., Mehta, S., Siby, S., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Mahajan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Lindsay, S., Feng, S., Lin, S., Zha, S. C., Patil, S., Shankar, S., Zhang, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S., Deng, S., Cho, S., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Koehler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V. S., Mangla, V., Ionescu, V., Poenaru, V., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wu, X., Wang, X., Wu, X., Gao, X., Kleinman, Y., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Yu, Wang, Zhao, Y., Hao, Y., Qian, Y., Li, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., Zhao, Z., and Ma, Z. (2024). The llama 3 herd of models.

Gunjal, A., Wang, A., Lau, E., Nath, V., Liu, B., and Hendryx, S. (2025). Rubrics as rewards: Reinforcement learning beyond verifiable domains.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. (2025). Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv*

preprint arXiv:2501.12948.

Gür, I., Hakkani-Tür, D., Tür, G., and Shah, P. (2018). User modeling for task oriented dialogues. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 900–906.

Henderson, M., Thomson, B., and Williams, J. D. (2014). The second dialog state tracking challenge. In Georgila, K., Stone, M., Hastie, H., and Nenkova, A., editors, *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

Kazi, T., Lyu, R., Zhou, S., Hakkani-Tur, D., and Tur, G. (2024). Large language models as user-agents for evaluating task-oriented-dialogue systems.

Keizer, S., Gasic, M., Jurcicek, F., Mairesse, F., Thomson, B., Yu, K., and Young, S. (2010). Parameter estimation for agenda-based user simulation. In *Proceedings of the SIGDIAL 2010 Conference*, pages 116–123.

Kim, T., Singh, J., Mehri, S., Acikgoz, E. C., Mukherjee, S., Bozdag, N. B., Shashidhar, S., Tur, G., and Hakkani-Tür, D. (2025). Pipa: A unified evaluation protocol for diagnosing interactive planning agents.

Kreyssig, F., Casanueva, I., Budzianowski, P., and Gasic, M. (2018). Neural user simulation for corpus-based policy optimisation for spoken dialogue systems. *arXiv preprint arXiv:1805.06966*.

Laban, P., Hayashi, H., Zhou, Y., and Neville, J. (2025). Llms get lost in multi-turn conversation. *arXiv preprint arXiv:2505.06120*.

Li, K., Liu, T., Bashkansky, N., Bau, D., Viégas, F., Pfister, H., and Wattenberg, M. (2024). Measuring and controlling instruction (in) stability in language model dialogs. *arXiv preprint arXiv:2402.10962*.

Li, X., Lipton, Z. C., Dhingra, B., Li, L., Gao, J., and Chen, Y.-N. (2017). A user simulator for task-completion dialogues.

Lin, H.-c., Geishauser, C., Feng, S., Lubis, N., van Niekerk, C., Heck, M., and Gašić, M. (2022a). Gentus: Simulating user behaviour and language in task-oriented dialogues with generative transformers. *arXiv preprint arXiv:2208.10817*.

Lin, H.-c., Geishauser, C., Feng, S., Lubis, N., van Niekerk, C., Heck, M., and Gasic, M. (2022b). GenTUS: Simulating user behaviour and language in task-oriented dialogues with generative transformers. In Lemon, O., Hakkani-Tur, D., Li, J. J., Ashrafzadeh, A., Garcia, D. H., Alikhani, M., Vandyke, D., and Dušek, O., editors, *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 270–282, Edinburgh, UK. Association for Computational Linguistics.

Lin, H.-c., Lubis, N., Hu, S., van Niekerk, C., Geishauser, C., Heck, M., Feng, S., and Gašić, M. (2021). Domain-independent user simulation with transformers for task-oriented dialogue systems. *arXiv preprint arXiv:2106.08838*.

Liu, Y., Jiang, X., Yin, Y., Wang, Y., Mi, F., Liu, Q., Wan, X., and Wang, B. (2023). One cannot stand for everyone! leveraging multiple user simulators to train task-oriented dialogue systems. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–21, Toronto, Canada. Association for Computational Linguistics.

Lu, J., Holleis, T., Zhang, Y., Aumayer, B., Nan, F., Bai, F., Ma, S., Ma, S., Li, M., Yin, G., Wang, Z., and Pang, R. (2025). Toolsandbox: A stateful, conversational, interactive evaluation benchmark for llm tool use capabilities.

Luo, X., Tang, Z., Wang, J., and Zhang, X. (2024). DuetSim: Building user simulator with dual large language models for task-oriented dialogues. In Calzolari, N., Kan, M.-Y., Hoste, V., Lenci, A., Sakti, S., and Xue, N., editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5414–5424, Torino, Italia. ELRA and ICCL.

McCarthy, P. M. and Jarvis, S. (2010). Mtld, vocd-d, and hd-d: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior research methods*, 42(2):381–392.

Mehri, S. and Eskenazi, M. (2020). USR: An unsupervised and reference free evaluation metric for dialog generation. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 681–707, Online. Association for Computational Linguistics.

Mehri, S. and Shwartz, V. (2023). Automatic evaluation of generative models with instruction tuning. In *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 42–52.

Muise, C., Chakraborti, T., Agarwal, S., Bajgar, O., Chaudhary, A., Lastras-Montano, L. A., Ondrej, J., Vodolan, M., and Wiecha, C. (2019). Planning for goal-oriented dialogue systems. *arXiv preprint arXiv:1910.08137*.

Mukherjee, S., Yuan, L., Hakkani-Tur, D., and Peng, H. (2025). Reinforcement learning finetunes small subnetworks in large language models.

Niu, C., Wang, X., Cheng, X., Song, J., and Zhang, T. (2024). Enhancing dialogue state tracking models through LLM-backed user-agents simulation. In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8724–8741, Bangkok, Thailand. Association for Computational Linguistics.

OpenAI (2024). Gpt-4o mini: advancing cost-efficient intelligence.

Pan, J., Shar, R., Pfau, J., Talwalkar, A., He, H., and Chen, V. (2025). When benchmarks talk: Re-evaluating code llms with interactive feedback.

Philipov, D., Dongre, V., gokhan tur, and Tur, D. H. (2024). Simulating user agents for embodied conversational AI. In *NeurIPS 2024 Workshop on Open-World Agents*.

Pietquin, O. and Dutoit, T. (2006). A probabilistic framework for dialog simulation and optimal strategy learning. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(2):589–599.

Pietquin, O. and Hastie, H. (2013). A survey on metrics for the evaluation of user simulations. *The Knowledge Engineering Review*, 28(1):59–73.

Prabhakar, A., Liu, Z., Zhu, M., Zhang, J., Awalgaonkar, T., Wang, S., Liu, Z., Chen, H., Hoang, T., Niebles, J. C., Heinecke, S., Yao, W., Wang, H., Savarese, S., and Xiong, C. (2025). Apigen-mt: Agentic pipeline for multi-turn data generation via simulated agent-human interplay.

Qian, C., Acikgoz, E. C., He, Q., Wang, H., Chen, X., Hakkani-Tür, D., Tur, G., and Ji, H. (2025). Toolrl: Reward is all tool learning needs.

Qwen, :, Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Tang, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., and Qiu, Z. (2025). Qwen2.5 technical report.

Rastogi, A., Zang, X., Sunkara, S., Gupta, R., and Khaitan, P. (2020). Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8689–8696.

Schatzmann, J., Thomson, B., Weilhammer, K., Ye, H., and Young, S. (2007). Agenda-based user simulation for bootstrapping a POMDP dialogue system. In Sidner, C., Schultz, T., Stone, M., and Zhai, C., editors, *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152, Rochester, New York. Association for Computational Linguistics.

Schatzmann, J., Weilhammer, K., Stuttle, M., and Young, S. (2006). A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The knowledge engineering review*, 21(2):97–126.

Schatzmann, J. and Young, S. (2009). The hidden agenda user simulation model. *IEEE transactions on audio, speech, and language processing*, 17(4):733–747.

Scheffler, K. and Young, S. (2002). Automatic learning of dialogue strategy using dialogue simulation and reinforcement learning. In *Proceedings of HLT*, volume 2, page 0.

Sekulić, I., Terragni, S., Guimarães, V., Khau, N., Guedes, B., Filipavicius, M., Manso, A. F., and Mathis, R. (2024). Reliable llm-based user simulator for task-oriented dialogue systems. *arXiv preprint arXiv:2402.13374*.

Shah, P., Hakkani-Tür, D., Liu, B., and Tür, G. (2018). Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning. In Bangalore, S., Chu-Carroll, J., and Li, Y., editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 41–51, New Orleans - Louisiana. Association for Computational Linguistics.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y. K., Wu, Y., and Guo, D. (2024). Deepseekmath: Pushing the limits of mathematical reasoning in open language models.

Shim, J., Seo, G., Lim, C., and Jo, Y. (2025). Tooldial: Multi-turn dialogue generation method for tool-augmented language models.

Silver, D. and Sutton, R. S. (2025). Welcome to the era of experience. *Google AI*, 1.

Skalse, J., Howe, N., Krasheninnikov, D., and Krueger, D. (2022). Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471.

Sun, W., Guo, S., Zhang, S., Ren, P., Chen, Z., de Rijke, M., and Ren, Z. (2023). Metaphorical user simulators for evaluating task-oriented dialogue systems. *ACM Transactions on Information Systems*, 42(1):1–29.

Tang, J., Zhao, T., Xiong, C., Liang, X., Xing, E., and Hu, Z. (2019). Target-guided open-domain conversation. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5624–5634, Florence, Italy. Association for Computational Linguistics.

Team, G., Kamath, A., Ferret, J., Pathak, S., Vieillard, N., Merhej, R., Perrin, S., Matejovicova, T., Ramé, A., Rivière, M., Rouillard, L., Mesnard, T., Cideron, G., bastien Grill, J., Ramos, S., Yvinec, E., Casbon, M., Pot, E., Penchev, I., Liu, G., Visin, F., Kenealy, K., Beyer, L., Zhai, X., Tsitsulin, A., Busa-Fekete, R., Feng, A., Sachdeva, N., Coleman, B., Gao, Y., Mustafa, B., Barr, I., Parisotto, E., Tian, D., Eyal, M., Cherry, C., Peter, J.-T., Sinopalnikov, D., Bhupatiraju, S., Agarwal, R., Kazemi, M., Malkin, D., Kumar, R., Vilar, D., Brusilovsky, I., Luo, J., Steiner, A., Friesen, A., Sharma, A., Sharma, A., Gilady, A. M., Goedeckemeyer, A., Saade, A., Feng, A., Kolesnikov, A., Bendebury, A., Abdagic, A., Vadi, A., György, A., Pinto, A. S., Das, A., Bapna, A., Miech, A., Yang, A., Paterson, A., Shenoy, A., Chakrabarti, A., Piot, B., Wu, B., Shahriari, B., Petrini, B., Chen, C., Lan, C. L., Choquette-Choo, C. A., Carey, C., Brick, C., Deutsch, D., Eisenbud, D., Cattle, D., Cheng, D., Paparas, D., Sreepathihalli, D. S., Reid, D., Tran, D., Zelle, D., Noland, E., Huizenga, E., Kharitonov, E., Liu, F., Amirkhanyan, G., Cameron, G., Hashemi, H., Klimczak-Plucińska, H., Singh, H., Mehta, H., Lehri, H. T., Hazimeh, H., Ballantyne, I., Szpektor, I., Nardini, I., Pouget-Abadie, J., Chan, J., Stanton, J., Wieting, J., Lai, J., Orbay, J., Fernandez, J., Newlan, J., yeong Ji, J., Singh, J., Black, K., Yu, K., Hui, K., Vodrahalli, K., Greff, K., Qiu, L., Valentine, M., Coelho, M., Ritter, M., Hoffman, M., Watson, M., Chaturvedi, M., Moynihan, M., Ma, M., Babar, N., Noy, N., Byrd, N., Roy, N., Momchev, N., Chauhan, N., Sachdeva, N., Bunyan, O., Botarda, P., Caron, P., Rubenstein, P. K., Culliton, P., Schmid, P., Sessa, P. G., Xu, P., Stanczyk, P., Tafti, P., Shivanna, R., Wu, R., Pan, R., Rokni, R., Willoughby, R., Vallu, R., Mullins, R., Jerome, S., Smoot, S., Girgin, S., Iqbal, S., Reddy, S., Sheth, S., Põder, S., Bhatnagar, S., Panyam, S. R., Eiger, S., Zhang, S., Liu, T., Yacovone, T., Liechty, T., Kalra, U., Evci, U., Misra, V., Roseberry, V., Feinberg, V., Kolesnikov, V., Han, W., Kwon, W., Chen, X., Chow, Y., Zhu, Y., Wei, Z., Egyed, Z., Cotruta, V., Giang, M., Kirk, P., Rao, A., Black, K., Babar, N., Lo, J., Moreira, E., Martins, L. G., Sanseviero, O., Gonzalez, L., Gleicher, Z., Warkentin, T., Mirrokni, V., Senter,

E., Collins, E., Barral, J., Ghahramani, Z., Hadsell, R., Matias, Y., Sculley, D., Petrov, S., Fiedel, N., Shazeer, N., Vinyals, O., Dean, J., Hassabis, D., Kavukcuoglu, K., Farabet, C., Buchatskaya, E., Alayrac, J.-B., Anil, R., Dmitry, Lepikhin, Borgeaud, S., Bachem, O., Joulin, A., Andreev, A., Hardin, C., Dadashi, R., and Hussenot, L. (2025). Gemma 3 technical report.

Wu, T. (1993). An accurate computation of the hypergeometric distribution function. *ACM Transactions on Mathematical Software (TOMS)*, 19(1):33–43.

Wu, W., Guo, Z., Zhou, X., Wu, H., Zhang, X., Lian, R., and Wang, H. (2019). Proactive human-machine conversation with explicit conversation goal. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3794–3804, Florence, Italy. Association for Computational Linguistics.

Xiao, R., Ma, W., Wang, K., Wu, Y., Zhao, J., Wang, H., Huang, F., and Li, Y. (2024). FlowBench: Revisiting and benchmarking workflow-guided planning for LLM-based agents. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10883–10900, Miami, Florida, USA. Association for Computational Linguistics.

Yao, S., Shinn, N., Razavi, P., and Narasimhan, K. (2024). τ-bench: A benchmark for tool-agent-user interaction in real-world domains.

Ye, F., Manotumruksa, J., and Yilmaz, E. (2022). Multiwoz 2.4: A multi-domain task-oriented dialogue dataset with essential annotation corrections to improve state tracking evaluation.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. *ArXiv*, abs/1904.09675.

Zhang, Z., Takanobu, R., Zhu, Q., Huang, M., and Zhu, X. (2020). Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, 63(10):2011–2027.

Zhao, T., Zhao, R., and Eskenazi, M. (2017). Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In Barzilay, R. and Kan, M.-Y., editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–664, Vancouver, Canada. Association for Computational Linguistics.

Zheng, Z., Liao, L., Deng, Y., Lim, E.-P., Huang, M., and Nie, L. (2024). Thoughts to target: Enhance planning for target-driven conversation. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21108–21124, Miami, Florida, USA. Association for Computational Linguistics.

Table 3: Categorization and frequency of goal alignment failures identified through manual analysis of 50 randomly selected conversations between LLM-based user simulators (Llama-3.1-8B-It and Qwen-2.5-7B-It) and conversational agents.

Goal Alignment Failure	Frequency
Confusion: The simulator forgets or confuses parts of the goal. For example, in a shopping application, when instructed to return a item A and exchange item B, the simulator incorrectly requests to return both items.	33%
Contradiction: The simulator directly contradicts specified constraints or contextual information. For example, in a flight booking scenario, the user simulator is instructed that they do not have their credit card details. However, they hallucinate fake credit card information when it is required to proceed with the booking.	23%
Wrongful termination: The simulator exhibits inadequate termination strategies–either terminating prematurely before receiving agent responses, or continuing indefinitely until reaching the maximum conversation length.	21%
Poor Length Management: The simulator fails to complete all parts of their goal within the conversation length limits, demonstrating poor management and awareness of conversation length. For example when tasked with booking multiple flights, the simulator exhausts the conversation length before completing all bookings.	12%
Misprioritization: The simulator overly prioritizes one part of their goal. They either get stuck on trying to achieve one part of their goal that is unachievable, rather than moving onto the remaining parts. Or, the user simulator terminates after completing one part of their goal, rather than completing the remaining parts of the goal.	11%

A User Simulator Goal Alignment Failures

B System Prompts

Prompt B.1: Agent Prompt

```
You are an advanced agent specializing in conversational dialogues. You can act both as an agent
(providing services) and a user (interacting with the database) to assist users in completing
complex tasks.
Each task may involve multiple sub-tasks, such as finding restaurants, making reservations,
booking hotels, locating attractions, and arranging transportation by checking for trains and
buying train tickets.

# Task Information:
        • If the user asks for some attributes of a venue, then an API call is necessary.
        • If you decide that more than one API calls are needed, you should call one API first and
          wait for the API result. After obtaining that result, you may think and call the next
          API or think and make a response.
        • If you decide that there is an API input slot that the user has never mentioned, please
          put "any" as the slot value as a placeholder.
# Objective:
        • Ensure that each assistant utterance follows logical reasoning, determining whether
          an API call is needed and structuring the output accordingly. item If there are too
          many results returned by API results from database, you should ask the user for more
          constraints unless the user explicitly wants you to pick one or some.
# General Procedure You should follow the workflow below every time you receive an input:
      1. Use APIs to retrieve information from the database, make operations to the backend, or
         use tools to perform other tasks.
      2. Repeat step 1 until you decide to make a response to the user.
      3. Use the response API to return the response to the user and end your turn.
```

Prompt B.2: User Simulator Prompt

```
You are a user simulator interacting with an agent. You are provided with User Goals, and your
task is to interact with the agent to achieve the goals as much as possible.

# Conversation Guidelines:
        • Work toward achieving specified user goals while maintaining a natural conversation
          flow.
        • Reveal information gradually and naturally as a real person would.
        • Do not hallucinate information that is not provided in the instruction. If asked
          about something not specified, respond honestly and tell the agent you do not have that
          information.
        • User natural language that is appropriate to the context and your assigned personality,
          rather than copying the exact wording from the instructions.
# Conversation Length:
        • The maximum conversation length is 20 messages. Plan your approach accordingly to
          accomplish your goals within this limit.
```

```
        • If you've achieved your goals before reaching the maximum length, you can end the
          conversation early.  To do so, you need to send "Terminate." in a separate, final
          message.
              – Make sure to allow the agent to respond before sending "Terminate.".
              – NEVER include "Terminate." in a message where you are asking a question or making
                a request to the agent, because it will end the conversation before the agent can
                respond.
              – Do NOT mention "Terminate." at any other time, because that will end the
                conversation prematurely.
        • Do not hallucinate information that is not provided in the instruction.  If asked
          about something not specified, respond honestly and tell the agent you do not have that
          information.
        • User natural language that is appropriate to the context and your assigned personality,
          rather than copying the exact wording from the instructions.
# User Goals:
{user_goals}
```

C Multiwoz User Goal Generation

We construct the MultiWOZ Challenge dataset to evaluate user simulator goal alignment. In this section, we present our user goal generation pipeline that we used to create this dataset.

Step 1: Task Objective Generation

For the attraction, hotel, restaurant, and train domains in the MultiWOZ database, we iterate over the entities. Each entity has a set of key-value pairs (e.g. ‘‘`area= east`’’, ‘‘`pricerange= moderate`’’). To instantiate a single task objective, we: (1) randomly sample requirement/preference keys to use their values to specify requirements or preferences about the entity the user is looking for, and (2) randomly sample request keys that will be used to specify information that the user wants about the entity. We provide this information to GPT-4o mini, and generate a user goal in natural language.

For example, given the attraction below:

```
{"name": "abbey pool and astroturf pitch",
 "type": "swimmingpool",
 "area": "east",
 "price range": "moderate",
 "address": "pool way ...",
 "phone": "01223 902088",
 ... }
```

we select the area and price range for requirement/preference keys, and select address and phone for request keys. The task objective would like like:
"Find a swimming pool attraction in the east area. You prefer one with a moderate price range. You want to know its address and phone number."

Impossible Task Objectives. To also observe how user simulators perform under failure conditions, we repeat the process above, but assign a random value to the requirement/preference keys. This helps create an entity that does not exist in our database.

Conditional Task Objectives. We also include conditional task objectives. A conditional task objective will require the user to search for one specific entity, but if it satisfies some conditional requirement, the user must look for another entity. For example:
"You are looking for a swimming pool with a cheap price range. If it is in the east area, then change your mind and look for another swimming pool in the west area that has a moderate price range."

Step 2: User Profile and Policy Generation

Next, we automatically generate a diverse set of about 50 user profiles and 50 user policies with GPT-4o mini. An example a user profile looks like:
"You are Maria García, a travel blogger documenting her adventures across Southeast Asia. Your experiences focus on local cuisine and cultural festivals."
An example a user policy looks like:
"Always ask the agent to restate booking details (date, time, price) before confirming. Verify critical details by repeating them back and asking 'Is that correct?'."

Then, we conduct manual annotations of the user profiles and policies, where we confirmed the relevance and quality. When necessary, some elements were removed or modified.

Step 3: User Goal Generation

Finally, we generate a user goal by randomly select a user profile, user policy and multiple task objectives. These were combined into a single user goal. We conducted more manual annotations at this step, to ensure the quality of user goals.

D User Goal State Generation

Our user goal state generation process consists of three main steps. First, we decompose each user goal into distinct, modular sub-components, where each represents an independent, self-contained part of the original goal. Second, we classify each sub-component into one of five categories: user profile, user policy, task objective, requirement or preference. Finally, we assign each sub-component its default value as described in Section 4.2.

For the decomposition and classification steps, we evaluated several LLMs, including DeepSeek-V3, Llama-3.3-70B-Instruct, GPT 4o-mini, and GPT 4o. Through manual validation, we determined that GPT-4o provided the highest quality results, and therefore used it to generate all the user goal states. The complete prompt used for this process is provided below.

Prompt D.1: Sub-Component Decomposition Prompt

```
You are an expert annotator tasked with dividing a user's goal into sub-components.

# User Goal:
{user_goal}

A user goal can be split into multiple sub-components. Each sub-component can be classified into
one of the following categories:
        • User Profile: persona, background, contextual information (e.g. You are a software
          engineer with 5 years of experience)
        • User Policy: general constraints and guidelines that must be consistently followed
          throughout the conversation (e.g. Always go with the cheapest option, don't share
          personal information)
        • Task Objectives: tasks that must be accomplished (e.g. Book a flight)
            – Requirements: conditions and constraints associated with this task objective (e.g.
              Purchase 3 carry-on bags)
            – Preferences: user preferences associated with this task objective (e.g. You
              prefer a direct flight, you'd like an aisle seat)
Analyze the user goal and divide it into sub-components:
        • Maintain the same wording as in the original goal when possible
        • Feel free to split up a sentence into multiple sub-components if it makes sense or if
          they fight in different categories
        • Ensure that no part of the original goal is left out
        • It is okay if a category is empty, if no sub-components are relevant
# Output Format
{
    "user_profile": [
        # List of user profile sub-components
    ],
    "user_policy": [
        # List of user policy sub-components
    ],
    "task_objectives": [ # List of task objectives
        {
            "task_objective": "task objective description",
            "requirements": [
                # List requirements for this task
            ],
            "preferences": [
                # List of preferences for this task
            ]
        }
    ]
}
Output only a JSON object, in the format specified.
```

E Sub-Component Status Update Prompt

Prompt E.1: User Profile Sub-Component Status Update Prompt

```
You are an expert evaluator analyzing a user simulator's behaviour. The user simulator is given
a user profile component, which specifies an aspect of their persona, background, or contextual
information. Your task is to evaluate the latest turn to determine if the user simulator behavior
is aligned with this user profile component.
# User Profile Component:
{user_profile_component}
# Conversation History:
{conversation_history[:-2]}

# Latest Turn:
{conversation_history[-2:-1]}
# Evaluation Criteria:
Evaluate the latest turn, and determine if the user simulator is aligned with the user profile
component ({user_profile_component}). You must provide a status, the value of which should be:
        • ALIGNED - Choose if ALL these conditions are met:
            1. The user simulator behavior does NOT directly contradict the user profile component
            2. IF the context clearly calls for demonstrating this profile component, then the
               user simulator demonstrates it appropriately (explicitly or implicitly)
            3. IF the context does NOT clearly call for demonstrating this profile component
               (neither demonstrating nor contradicting)
        • MISALIGNED - Choose if ANY of these conditions are met:
            1. The user simulator behavior DIRECTLY contradicts the user profile component
            2. The context clearly and obviously calls for demonstrating the profile component,
               AND the user simulator fails to do so in a way that would be unrealistic for
               someone with this profile
# Important Guidelines:

        • Focus on the latest turn, but use the conversation history for context
        • Unless there's a clear contradiction, or obvious missed opportunity, lean towards
          ALIGNED
            – Most turns will not require demonstrating the profile component
A user goal can be split into multiple sub-components. Each sub-component can be classified into
one of the following categories:
        • User Profile: persona, background, contextual information (e.g. You are a software
          engineer with 5 years of experience)
        • User Policy: general constraints and guidelines that must be consistently followed
          throughout the conversation (e.g. Always go with the cheapest option, don't share
          personal information)
        • Task Objectives: tasks that must be accomplished (e.g. Book a flight)
            – Requirements: conditions and constraints associated with this task objective (e.g.
              Purchase 3 carry-on bags)
            – Preferences: user preferences associated with this task objective (e.g. You
              prefer a direct flight, you'd like an aisle seat)
# Output Format:
{{
    "status": # ALIGNED | MISALIGNED,
    "reasoning": # Brief explanation of your decision
}}

Output a properly formatted JSON response, as specified by the Output Format, and nothing else.
```

F Naturalness and Coherence Evaluation Prompts

Prompt F.1: Naturalness Evaluation Prompt

```
You are an expert evaluator analyzing a user simulator's behavior while they are pursuing a user
goal. Your task is to evaluate the naturalness of the user simulator throughout conversation.

# User Goal
{user_goal}
```

```
# Conversation History
{conversation_string}

# Evaluation Task
Rate the naturalness of the user responses on a scale from 1 to 5, based on three key dimensions:

      1. **Grammar and coherence**:  Grammatically correct sentences that flow logically and
         coherently.
      2. **Context Relevance** Responses are strongly related to the conversation context and
         previous turns, as well as the user goal.
      3. **Conversational Style**:  Responses are natural, and appropriate for the conversation
         style.

The rating scale is as follows:

        • 5 (Highly Natural):  Excels in all three dimensions - perfect grammar/coreference, fully
          contextually appropriate, and natural conversational
        • 4 (Mostly Natural):  Strong in all dimensions with minor lapses - occasional awkward
          phrasing or slightly unnatural language
        • 3 (Moderately Natural):  Adequate in all dimensions - functional but noticeably
          artificial in style or occasional context mismatches
        • 2 (Somewhat Unnatural):  Weak in 1-2 dimensions - grammar errors, loses context, or
          overly formal/robotic style
        • 1 (Very Unnatural):  Fails multiple dimensions - poor grammar, irrelevant responses, or
          mechanical/repetitive patterns

# Output Format:
{{
    "naturalness_score": # number from 1 to 5
    "reasoning": # Brief explanation of your evaluation decision
}}

Output a properly formatted JSON response, as specified by the Output Format.
```

Prompt F.2: Coherence Evaluation Prompt

```
You are an expert evaluator analyzing a user simulator's behavior while they are pursuiing a user
goal.  Your task is to evaluate the coherence of a user simulator in pursuing their goal in a
conversation.

# User Goal
{user_goal}

# Conversation History
{conversation_string}

# Evaluation Task
Rate the coherence of the user's dialogue on a scale from 1 to 5, based on three key dimensions:

      1. **Goal Progression**:  User consistently works toward their stated goal with appropriate
         persistence and adaptation
      2. **Topic Continuity**:  Responses maintain topical relevance and logical connections
         between turns
      3. **Response Appropriateness**:  User responds appropriately to system prompts, questions,
         and clarification requests

The rating scale is as follows:

        • 5 (Highly Coherent):  Clear goal pursuit throughout, all utterances topically connected,
          appropriate responses to all system turns
        • 4 (Mostly Coherent):  Strong goal focus with minor deviations, well-connected utterances,
          occasionally misses response opportunities
        • 3 (Moderately Coherent):  Generally pursues goal with some inconsistencies, mostly
          connected utterances, some inappropriate responses
        • 2 (Somewhat Incoherent):  Weak goal focus with major deviations, frequent topic jumps,
          often misses or misinterprets prompts
        • 1 (Very Incoherent):  No clear goal pursuit, disconnected responses, fails to engage
          appropriately with system

# Output Format:
{{
```

```
    "coherence_score": # number from 1 to 5
    "reasoning": # Brief explanation of your evaluation decision
}}

Output a properly formatted JSON response, as specified by the Output Format.
```

G Human Evaluation

User Goal State Tracking. We validate our evaluation method by conducting a comprehensive human evaluation study with 10 graduate-level human annotators. For a total of 30 randomly selected conversations, human annotators manually conduct UGST, resulting in a total of 300 human annotated goal states. We measure the agreement between the human annotated goal states and LLM generated goal states by reporting the proportion of matching sub-component statuses. Table 4 presents both results per sub-component category and overall agreement, demonstrating that LLMs can reliably conduct UGST.

Table 4: Human Agreement scores with LLM-based UGST on random user-agent conversations. We report the overall scores, as well as scores for each sub-component category.

Prof.	Pol.	T.O.	Req.	Pref.	Avg
91.7	72.7	91.1	81.3	88.7	85.7

User Goal State Generation. Due to its central role in our evaluation methodology, we also evaluate how well GPT-4o generates user goal states. We manually create user goal states for a total of 30 randomly selected user goal states, and compare against the goal-states generated by GPT-4o. We report the precision (P), recall (R), and F1 scores for the sub-components, where true positives are when the generated goal state correctly contains a sub-component, false positives are when the goal state contains a sub-component it should not contain, and false negatives are when the generated goal state is missing a sub-component. Additionally, we report the accuracy (Acc.) for categorizing the sub-components. As shown in Table 5, GPT-4o achieves high performance across all metrics, indicating reliable goal state generation that ensures accurate and representative downstream evaluations.

Table 5: Precision (P), Recall (R), F1 score, and classification accuracy (Acc.) of GPT-4o for user goal state generation.

P	R	F1	Acc.
98.04	95.60	96.63	91.97

H Discussion

H.1 Further Analysis of User Simulators

To further validate our findings and examine the broader impact of our methodology, we conduct further analysis of our user simulators and present results averaged across τ-Bench Airline, τ-Bench Retail, and MultiWOZ Challenge in Table **??**.

Goal Alignment. Following Zheng et al. (2024), we additionally evaluate goal alignment by using BERTScore (Zhang et al., 2019) to measure the semantic similarity between user utterances and their user goal, where higher F1 scores indicate better goal alignment. We observe that the semantic similarity scores generally show an increasing trend for each stage of our method, with the final stage achieving the highest scores.

Naturalness and Coherence. A key concern when improving goal alignment is ensuring that user simulators maintain their conversational abilities to gen-

Table 6: Additional analysis of user simulators. We report BERTScore (F1) for semantic similarity between user utterances and user goals (higher scores indicate better goal alignment), naturalness (N) and coherence (C) scores, and diversity metrics including MTLD and HDD. Results are averaged across τ-Bench and MultiWOZ datasets.

Model	F1	N	C	MTLD	HDD
Prompt-Based					
Qwen-2.5-7B-It	0.827	4.15	4.31	71.9	0.403
Llama-3.1-8B-It	0.819	4.05	4.211	67.0	0.513
Gemma-3-27B-It	0.823	4.16	4.30	**85.4**	0.524
Qwen-2.5-72B-It	0.824	**4.26**	4.50	70.1	0.382
Llama-3.3-70B-It	0.826	<u>4.25</u>	4.51	75.2	0.720
GRPO with UGST Rewards					
Qwen-2.5-7B-It	0.827	4.05	4.37	80.7	**0.806**
Llama-3.1-8B-It	**0.836**	3.85	4.20	74.0	<u>0.795</u>

erate realistic responses. To address this, we assess the naturalness and coherence of user-agent conversations by employing Qwen-2.5-72B-Instruct to rate conversations on a scale from 1 to 5, as in Kazi et al. (2024). The evaluation prompts are provided in Appendix F. Throughout the different stages, naturalness and coherence scores show only minor variations and remain within a similar range. This indicates that our approach achieves substantial goal alignment improvements without degrading the fundamental conversational qualities essential for realistic user simulation.

Diversity. Lastly, we assess diversity using Measure of Textual Lexical Diversity (MTLD) (McCarthy and Jarvis, 2010) and Hypergeometric Distribution Function (HDD) (Wu, 1993) following established practices in user simulation evaluation (Luo et al., 2024; Pietquin and Hastie, 2013). MTLD quantifies lexical diversity at the token level, and HDD captures the diversity of vocabulary when randomly selecting a fixed number of words from the user utterances. There is a consistent increase in diversity of user simulator utterances, and user simulators from our final stage demonstrate significant improvements in diversity. These results provide additional validation that our methodology enhances goal alignment of user simulators, while also demonstrating that it promotes more diverse conversational behaviors, a highly desirable trait for user simulation.

Model	Prof.	Pol.	T.O.	Req.	Pref.	Avg
Prompt-Based						
Qwen-2.5-7B-It	88.3	49.2	94.3	96.0	85.7	82.7
Llama-3.1-8B-It	90.9	41.0	97.1	99.0	81.0	81.8
Gemma-3-27B-It	**98.7**	59.0	97.1	97.0	78.6	86.1
Qwen-2.5-72B-It	89.3	59.6	94.1	96.9	78.6	83.7
Llama-3.3-70B-It	97.4	65.6	**98.1**	99.0	92.9	**90.6**
Inference-Time Steering with User Goal						
Qwen-2.5-7B-It	70.1	54.2	**98.1**	99.0	90.5	82.4
Llama-3.1-8B-It	87.0	60.7	97.1	98.0	**100.0**	88.6
Gemma-3-27B-It	92.2	**73.8**	96.2	98.0	73.8	86.8
Qwen-2.5-72B-It	93.5	62.3	92.4	94.1	76.2	83.7
Llama-3.3-70B-It	**98.7**	59.0	96.2	98.0	90.9	88.6
Inference-Time Steering with User Goal States						
Qwen-2.5-7B-It	77.9	55.7	96.2	98.0	92.9	84.1
Llama-3.1-8B-It	92.2	57.4	93.3	98.0	95.2	87.2
Gemma-3-27B-It	94.8	62.3	92.4	97.0	85.7	86.4
Qwen-2.5-72B-It	93.3	54.4	98.0	97.9	97.6	88.3
Llama-3.3-70B-It	93.5	63.9	**98.1**	**100.0**	97.6	**90.6**

(a) τ-Bench Airline

Model	Prof.	Pol.	T.O.	Req.	Pref.	Avg
Prompt-Based						
Qwen-2.5-7B-It	82.0	40.8	96.0	99.5	91.7	82.0
Llama-3.1-8B-It	84.8	36.8	97.1	98.9	96.3	82.8
Gemma-3-27B-It	91.0	39.5	97.8	99.5	96.3	84.8
Qwen-2.5-72B-It	88.6	40.8	98.2	97.9	91.7	83.4
Llama-3.3-70B-It	91.7	46.1	**99.6**	**100.0**	**100.0**	87.5
Inference-Time Steering with User Goal						
Qwen-2.5-7B-It	73.4	38.2	97.5	98.4	95.4	80.6
Llama-3.1-8B-It	91.7	38.2	98.9	99.5	94.4	84.5
Gemma-3-27B-It	**94.5**	**63.2**	97.8	98.4	91.7	89.1
Qwen-2.5-72B-It	89.6	40.8	97.1	98.9	97.2	84.7
Llama-3.3-70B-It	92.4	55.3	98.2	**100.0**	98.1	88.8
Inference-Time Steering with User Goal States						
Qwen-2.5-7B-It	73.0	47.4	94.9	97.9	93.5	81.4
Llama-3.1-8B-It	84.8	52.6	95.3	98.9	97.2	85.8
Gemma-3-27B-It	**94.5**	56.6	97.1	99.5	89.8	87.5
Qwen-2.5-72B-It	85.8	48.7	97.1	98.4	98.1	85.6
Llama-3.3-70B-It	92.7	59.2	96.0	98.4	**100.0**	**89.3**

(b) τ-Bench Retail

Model	Prof.	Pol.	T.O.	Req.	Pref.	Avg
Prompt-Based						
Qwen-2.5-7B-It	54.7	18.0	78.4	81.9	73.5	61.3
Llama-3.1-8B-It	73.6	35.0	86.1	89.0	80.9	72.9
Gemma-3-27B-It	80.1	50.3	95.0	97.7	92.1	83.0
Qwen-2.5-72B-It	72.0	31.0	91.0	93.3	90.2	75.5
Llama-3.3-70B-It	72.0	31.0	**97.9**	**98.6**	**93.2**	83.3
Inference-Time Steering with User Goal						
Qwen-2.5-7B-It	49.8	28.0	77.8	83.2	69.4	61.7
Llama-3.1-8B-It	73.9	49.7	92.9	95.8	80.4	78.5
Gemma-3-27B-It	**80.8**	49.0	95.9	98.3	89.4	82.7
Qwen-2.5-72B-It	71.7	41.7	92.7	94.9	93.1	78.8
Llama-3.3-70B-It	73.6	60.3	94.9	95.8	91.0	83.1
Inference-Time Steering with User Goal States						
Qwen-2.5-7B-It	43.4	26.3	88.1	89.8	71.1	63.7
Llama-3.1-8B-It	71.7	38.0	85.2	88.6	74.3	71.6
Gemma-3-27B-It	72.3	45.7	87.0	89.7	74.8	73.9
Qwen-2.5-72B-It	71.0	46.3	94.9	96.9	92.9	80.4
Llama-3.3-70B-It	72.6	**64.3**	97.0	97.8	91.7	**84.7**

(c) MultiWOZ Challenge

Table 7: User simulator goal alignment performance with prompt-based, inference-time steering with user goal and inference-time steering with user goal states. The table shows the average success rates for User Profile (Prof.), User Policy (Pol.), Task Objective (T.O.), Requirements (Req.) and Preferences (Pref.) sub-component categories, and the overall average performance. Highest scores are **bolded**, and the second-highest scores are underlined.