Thinking Alignment of Scenario-Oriented User Simulation

Xiaoting Wu[1], Yi Huang[1,2*], Chunyang Gao[1], Mengfei Guo[1], Jingyu Yao[1] and Junlan Feng[1]
[1]JIUTIAN Research, [2]Department of Computer Science and Technology, Tsinghua University, China
{wuxiaoting,huangyi,gaochunyang,guomengfei,yaojingyu,fengjunlan}@cmjt.chinamobile.com

Abstract

Existing user simulators based on prompting to role-play or SFT are generally confined to imitating users' textual utterances, without adequately considering the multi-faceted cognitive processes that underlie human decision-making during interactions. To facilitate better alignment with real human thinking patterns, we construct the LMSYS-UserThinking dataset, in which we augment 51k human–LLM conversations by reconstructing the user's inner reasoning both during and at the end of each dialogue. Furthermore, to enhance controllability and situational coherence, we introduce scenario settings that describe the global context and user goals throughout multi-turn conversations. Using this dataset, we train user simulators called ThinkingUS on different base models. We evaluate our approach from both offline and online user simulation perspectives, ultimately demonstrating its effectiveness.

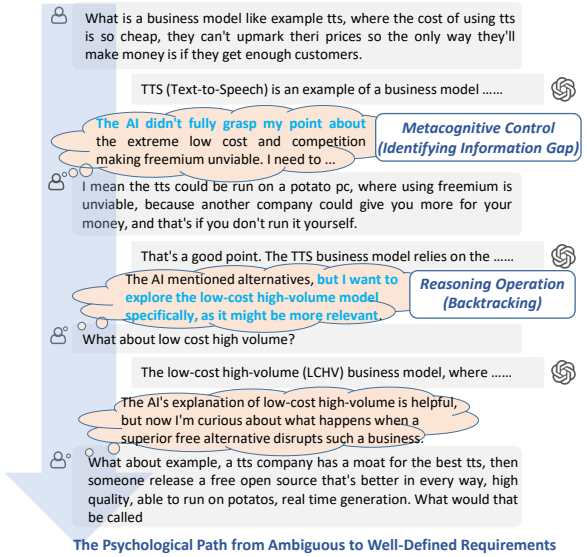

Figure 1: User's psychological process for exploring complex problems, including metacognitive control and reasoning operations.

1 Introduction

User simulation, which aims to replicate the conversational behaviors of real human users, thereby providing an interaction environment that closely resembles real-world scenarios, has emerged as a critical technology in interactive intelligence research. Building a high-fidelity user simulator can not only provide large-scale, high-quality multi-turn interaction data for assistant language models (LMs) (Ding et al., 2023; Kong et al., 2024b), but also support online reinforcement learning training and interactive evaluation of assistant LMs (Qian et al., 2025a,b; Shea et al., 2025). While user simulation has a rich history spanning decades, the advent of large language models (LLMs) has fundamentally transformed both the capabilities and methodologies in this field.

Current research on user simulators falls mainly into two technical paradigms. The first is the prompt-based user simulation approach, which relies on predefined user personas to characterize different users and uses instructions to guide LLMs in role-playing to generate simulated user utterances (Xu et al., 2023; Ding et al., 2023; Wu et al., 2025; Li et al., 2024). However, these prompt-based simulators often diverge from the direct, pragmatic, and unpredictable behavioral patterns typical of real human users (Ivey et al., 2024). The second paradigm is the supervised fine-tuning (SFT) method based on real dialogue data, which uses real human–LLM conversation data as the training foundation to enable the model to learn authentic user interaction behavior patterns. The Socratic user simulator (Kong et al., 2024b) built by PlatoLM takes real user queries from the ShareGPT dataset as learning targets and achieves highly human-like multi-turn query generation by fine-tuning the LLaMA model. USP (Wang et al., 2025a) employs the LMSYS-Chat-1M dataset (Zheng et al., 2024) for

*Corresponding authors

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 42932–42945
July 2-7, 2026

fine-tuning, enabling the simulator to generate interactions based on implicit personas. UserLM (Naous et al., 2025) trains a dedicated user LM capable of following user intentions using millions of real human–LLM interaction data from the WildChat dataset (Zhao et al., 2024).

Despite increasing attention and efforts towards user simulation, existing work has largely focused on imitating utterances of users, overlooking the multi-layered psychological processes grounded in individual cognition during interactions. In real-world scenarios, conversations between human users and assistant LMs are often situation-driven and frequently involve vague, evolving, or implicitly expressed goals (Qian et al., 2025a). Under such circumstances, merely imitating utterances remains superficial—it captures the "what" but fails to explain the "why". Simulating the internal cognitive processes that drive user behavior can yield behavior patterns that are more authentic to the underlying mechanisms and more generalizable across contexts. As illustrated in Figure 1, when exploring complex problems, a user's psychological process encompasses a series of cognitive elements—including metacognitive control (identifying information gaps and adjusting strategies) and reasoning operations (such as verification, backtracking, and principle refinement)—which collectively render the user's behavior more well-grounded.

In this paper, we first consider how to represent the rich, situation-driven user context. Inspired by CoSER (Wang et al., 2025c), we fully characterize the user context through scenario setting, user thought, and user utterance. To better align with genuine human thinking patterns, we construct user thought approximation of users based on real human–LLM dialogue data, guided by a four-dimensional cognitive framework (Kargupta et al., 2025b). For convenience, we refer to the enhanced dataset as **LMSYS-UserThinking**. To test whether our constructed user thoughts help replicate real human users, we train user simulators, **ThinkingUS**, on LMSYS-UserThinking. We construct offline and online evaluation on ThinkingUS. The experimental results demonstrate that ThinkingUS significantly outperforms methods that prompting to user role-playing and user utterance SFT. Our key contributions are summarized below:

- We devise a user context representation that includes scenario setting, user thought, and user utterance, presenting a more complete view of both visible and invisible user context.
- We construct LMSYS-UserThinking dataset which captures psychological processes of human users during multi-turn dialogue with assistant LMs, enabling better alignment of user thinking for internal cognitive simulation.
- We conduct extensive experiments to evaluate the effectiveness of our thinking-augmented user simulation method. Experiments on offline and online settings demonstrate that our thinking-augmented method is more effective in imitating user behaviors.

2 Related Work

Prompt-based Methods. Recent research has focused on utilizing sophisticated prompting to steer LLMs into realistic user simulations without modifying model weights. A primary direction involves aligning models with specific human traits and perspectives by incorporating diverse life narratives, backstories, and historical user opinions, which enhances the consistency and demographic representation of the simulated subjects (Hwang et al., 2023; Moon et al., 2024; Herlihy et al., 2024; Zhang et al., 2024; Dongre et al., 2025; Ferreira et al., 2024). Beyond static personas, architectural frameworks that integrate memory, reflection, and planning have enabled LLMs to serve as interactive generative agents, simulating complex social dynamics and emergent human-like behaviors in sandbox environments (Park et al., 2023; Holderried et al., 2024). Furthermore, user simulation via prompting has proven effective for task-oriented objectives, such as boosting zero-shot reasoning capabilities through persona-driven contexts or providing multi-dimensional, human-consistent metrics for text evaluation (Kong et al., 2024a; Wu et al., 2023; Kargupta et al., 2024, 2025a). However, prompt-based simulators often exhibit excessive cooperativeness and politeness bias, failing to capture the unpredictable or critical nature of real human behavior.

Fine-tuning Methods. To internalize consistent interaction patterns directly within model parameters, another line of work leverages supervised fine-tuning and reinforcement learning to build more robust user simulators. Early efforts focused on scaling high-quality, multi-turn instructional dialogues to improve the general conversational breadth and reasoning of models acting as

human-like interlocutors (Wan et al., 2022; Ding et al., 2023; Kong et al., 2024b; Sekuli'c et al., 2024; Dhole, 2024; Liu et al., 2023). To achieve higher authenticity, modern frameworks have transitioned toward modeling implicit user traits and narrative chains (Wang et al., 2025a; Wu et al., 2024). Building upon these methods, recent studies have increasingly employed reinforcement learning to optimize simulators for specific interaction objectives or behavioral diversity, enabling them to better generalize across dynamic scenarios by pursuing long-term rewards (Wang et al., 2025b; Wei et al., 2025). However, these fine-tuning approaches typically focus on surface-level imitation of what a user says without modeling the underlying why. This lack of cognitive alignment leads to simulations that are behaviorally consistent but cognitively hollow. Our work addresses this gap by incorporating inner thought processes into the fine-tuning stage, bridging the gap between behavioral imitation and cognitive simulation.

3 Approach

This section details the methodology for user simulation, proceeding from context representation to behavioral generation. As shown in Figure 2, we construct a dataset based on the user context representation and perform user thought alignment training.

3.1 User Context Representation

Existing user simulation datasets often encompass only dialogue data. However, the user context encompasses not only the visible user input, but also substantial implicit information behind the user saying. Our aim is to develop a dataset that captures more comprehensive user contexts. Specifically, we posit that user verbal behavior is motivated by contextual goals and modulated by internal cognition and interactional expectations. We model the user context as comprising three components to represent such comprehensive user information.

Scenario Setting involves a detailed description of the context in which users pose questions, encompassing objectives, behaviors, domains, intentions, functional requirements, constraints, user identities, and emotional attitudes. This setting is applicable to the entire multi-turn dialogue.

User Thought follow-up with the assistant based on its response and encompasses unexpressed internal cognitive processes, such as the user's perception of the response, interaction expectations, adaptive behavioral adjustments, and emotional states.

User Utterance refers to the content articulated by the user in each dialogue turn, which is directly observable by the assistant model.

3.2 Data Construction

We leverage authentic dialogue datasets to construct user contexts, thereby enabling more human-like, diverse, and complex scenarios as well as interaction patterns. Existing dialogue datasets inherently contain user utterances but lack scenario settings and user thoughts. We complement the dialogue by adding scenario settings and user thought approximations to obtain comprehensive user context. The user context construction process is formulated as:

$$(U, A) \longrightarrow (S, M, T), \tag{1}$$

where U and A denote the user requests and assistant responses, respectively, in a multi-turn dialogue. The scenario setting is represented as scenario description S and user motivation M. The scenario description comprises background information of the user dialogue, while user motivation refers to the underlying purpose or conversational goal of the user. T is the user thought approximation of each turn, which is categorized into in-process user thought (enabling the reasonable inference of the user's utterance) and terminal user thought (guiding appropriate conversation termination based on user goals and dialogue progression).

Scenario Setting Generation. The scenario description and the user motivation are used to represent the implicit and explicit factors driving multi-turn dialogues, respectively. We instruct advanced LLM to synthesize these two components from the human-LLM dialogue while taking the holistic context of the conversation into account.

In-Process User Thought Approximation. We generate the user thought for each turn based on real conversational data consisting of alternating utterances between the user and the assistant LM. Thus, the in-process user thought serves as a bridge connecting the previous assistant response to the current user utterance. When generating such user thoughts, it is essential to first understand the user's core needs throughout the entire interaction, and then focus on the adjacent turns around the target utterance—specifically, what the user said in the

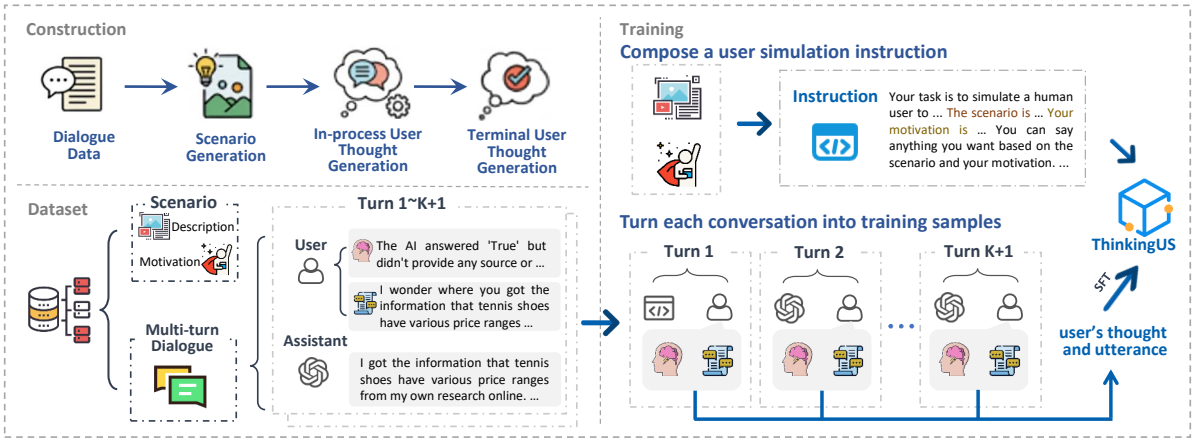

Figure 2: Overview of dataset and training process of ThinkingUS. Left is an overview of the dataset. It contains rich user contextual information of scene descriptions and thinking processes of users. Right is the training process. We construct a unified instruction template and convert each multi-turn conversation with K turns into K+1 training samples. Each sample trains the LLM to portray a specific user in a conversation, conditioning both on the scenario setting and dialogue history to generate the next user utterance.

previous turn, how the assistant responded, and what the user says in the current turn. It should be determined whether the current user utterance is an extension of the prior user turn or a follow-up to the assistant's last response. If it builds upon the previous user utterance, one should first consider the differences between the two user turns before reasoning about the user's possible thought in light of the assistant's reply. If it follows from the assistant's response, the inference should be grounded in the logical flow of the conversation. Similar to the process of scenario setting generation, the LLM was provided with the complete conversation history and instructed to produce a version of the dialogue in which a user thought is inserted before each user utterance, without altering any of the original user utterances. Prompt details can be found in the Appendix A.

Terminal User Thought Approximation. We initially attempted to generate both the in-process and terminal user thoughts in a single step using the aforementioned method. However, manual inspection reveals a relatively high divergence between the generated terminal user thought and human judgment. This discrepancy primarily stems from that generating the terminal user thought lacks a subsequent user utterance as a direct reference, and the general user-thought-generation prompt proves insufficient to effectively guide the model. To address this, we designed a two-step prompting approach. First, the LLM is prompted to revisit the user motivation outlined in the scenario setting and assess whether the primary demands of the user are fulfilled. Subsequently, it is instructed to synthesize this reflection to generate the user's thought at the end of the conversation. This two-step method, analogous to chain-of-thought reasoning, helps decompose the task and improves the quality of the generated terminal user thought.

3.3 Training

To enhance the controllability of user simulation, we employ conditional supervised fine-tuning based on scenario settings, following prior works (Wang et al., 2025a; Naous et al., 2025). We transform each dialogue with K turns into K+1 training samples. For the first turn, the initial user utterance is generated conditioned on the scenario setting. For subsequent turns, the next concatenated user thought and user utterance are generated conditioned on both the scenario setting and the dialogue state. Here, the user thought is wrapped with "<thought>" and "</thought>" tags to distinguish it from the user utterance. We train LLMs on the concatenation of thought and utterance using the following training objective:

$$\mathcal{L} = -\sum_{i=0}^{K} logP(u_i|a_{i-1}, u_{i-1}, ..., a_0, u_0, S, M), \tag{2}$$

where u_i denotes the *i-th* user response which is the concatenation of user thought and user utterance, a_i is the *i-th* assistant response, $P(u_i|a_{i-1}, u_{i-1}, ..., a_0, u_0, S, M)$ is the conditional probability of the model generating the *i-th*

user response given the scenario setting and the history turns. K is the total number of interaction turns.

4 Experiments

To validate the effectiveness of user simulation based on user thinking, conditioned on scenario setting, we conduct several experiments. In Section 4.1, we report the performance of the user simulation on offline test set. In Section 4.2, we evaluate the dialogues of online interactions between user simulators and the assistant LM in designated scenarios.

Training Data. We use LMSYS-Chat-1M (Zheng et al., 2024), which contains one million human-LLM conversations. Following prior work (Wang et al., 2025a), we filter out non-English, toxic, and redundant samples during data preprocessing. Then we leverage DeepSeek-V3[1] to generate scenario settings and user thoughts as described in Section 3.2. The generated data that do not meet the requirements are filtered out, ultimately resulting in 53,818 conversations (51,140 for training, 2,678 for testing).

Training Details. We perform full-parameter fine-tuning of Llama-3.1-8B[2] and Llama-3.1-8B-Instruct[3] over 3 epochs at a learning rate of 1e-5, max sequence length of 4,096, and a global batch size of 32.

Baselines. We compare our ThinkingUS with non-thinking and thinking baselines (the base LLM adopted is Llama-3.1-8B-Instruct): Prompt-based (**Prompt**, via prompt directly conditioned on scenario settings), Prompt thinking (**Prompt Thinking**, via prompt that output thought processes and responses conditioned on scenario settings), SFT-based (**Utterance SFT**, based on user utterance SFT directly) user simulators.

Benchmarks. We evaluate on held-out test samples from LMSYS-UserThinking (15,015 samples in 2,678 conversations). To assess more general user simulation capability, we extract part of the data from WildChat (Zhao et al., 2024) as the out-of-domain test set. Specifically, we filter out non-English, toxic, and redundant samples. Subsequently, we randomly select part of the processed data and apply the scenario setting generation procedure to the selected WildChat conversations, resulting 1,560 conversations (3,381 turns) as the WildChat test set.

4.1 Offline Evaluation

We evaluate the user simulation performance during and at the end of the interaction separately. In-process user evaluation is achieved through pairwise comparison on user utterances. Terminal user evaluation focus on how well the assessments from user simulation match the distribution of human user assessments of the assistant at the end of conversations.

4.1.1 In-Process User Evaluation

Metrics. We leverage DeepSeek-V3 to benchmark responses of each method against the Prompt outputs (details are displayed in Section A), employing a win-rate metric (Ji et al., 2024) to evaluate which user response aligns better with real user utterances. This metric is formulated as:

$$r_w = \frac{N_w - N_l}{N_w + N_l + N_e}, \tag{3}$$

where r_w denotes the win-rate, while N_w, N_l and N_e represent the number of wins, losses, and draws compared to the corresponding utterance generated by the same Prompt method. The comparison is conducted along two dimensions: user intent (measuring the alignment between the simulator's output and the ground truth in terms of goals, needs, or purposes) and linguistic expression (assessing the similarity in language style, tone, and level of detail—such as the use of analogous phrasing, keywords, or emotional tone—between the simulator's output and the ground truth).

Results. First, **thinking user simulators outperform non-thinking simulators**, even when compared to user simulator fine-tuned on real human utterances but not explicitly designed for thinking. This advantage remains consistent across two different test sets, indicating that simulating the thinking process of users plays a significant role in user modeling. During conversations with assistant models, human users typically act as the primary driver of dialogue. Before each utterance, they make decisions based on their goals, personal knowledge, and the prior responses of assistant. Thinking models are able to partially emulate these internal cognitive mechanisms, thereby generating dialogue behaviors that more closely approx-

[1] https://huggingface.co/deepseek-ai/DeepSeek-V3

[2] https://huggingface.co/meta-llama/Llama-3.1-8B

[3] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

Method	LMSYS-Chat				WildChat			
	win.	loss.	draw.	r_ω	win.	loss.	draw.	r_ω
Utterance SFT	8791	5862	362	0.1951	2066	1278	37	0.2331
Prompt Thinking	8674	5426	915	0.2163	2175	1093	113	0.3200
ThinkingUS (on Instruct)	9259	5450	306	0.2537	2267	1080	34	0.3511
ThinkingUS (on Base)	9931	4791	293	**0.3423**	2472	889	20	**0.4682**

Table 1: In-process user evaluation results. We leverage DeepSeek-V3 to benchmark responses of each method against the Prompt outputs, determining their win-rates, i.e., r_ω.

imate the complexity and coherence of real human interaction. Second, **LMSYS-UserThinking dataset demonstrates significant effectiveness**. On both in-domain and out-of-domain test sets, the ThinkingUS models achieve higher win-rate than the Prompt Thinking model, indicating that LMSYS-UserThinking can effectively enhance the ability of user simulators to mimic real human intentions and expression patterns in given scenarios. Third, **base model is more suitable than instruction-tuned model for training user simulator**. ThinkingUS starting from a base model achieves a higher win-rate than that starting from an instruction-tuned model, with a clear margin of advantage, on both in-domain and out-of-domain test sets. This is because instruction-tuned models are typically trained to act as helpful assistants, and much of the instruction data is synthetic, which often differs substantially from real human user requests. Finally, all methods achieve higher win-rates on the WildChat test set compared to LMSYS-Chat, indicating a considerable performance gap between the Prompt method and each model on this dataset. Since WildChat data consists of real interaction logs voluntarily shared by users, it better reflects genuine usage scenarios. Consequently, simulating authentic user interactions solely through prompt instructions poses a greater challenge.

We also fine-tune Qwen3 models of two sizes (4B and 8B) to validate the generalization capability of our thinking-aligned user simulation method (details are displayed in Section B). Table 5 presents the evaluation results on the LMSYS-Chat and the WildChat test set. The ThinkingUS method significantly outperforms the utterance-only fine-tuning approach on both Qwen3-8B and Qwen3-4B, which demonstrate that our method **not only improves user simulation on Llama models but also achieves gains on Qwen3 models**, indicating its generalizability across different model architectures.

4.1.2 Terminal User Evaluation

In addition to conducting realistic conversations, a user simulator should also accurately assess the performance of assistant models. To achieve this, we measure the evaluation of the assistant included in the terminal user thought. We employ the Cramér–von Mises (CV-M) divergence (Williams, 2007) to quantify the agreement of the score distribution regarding the assistant between user simulators and real users. To obtain the numerical score, we map the terminal thoughts to a 5-level rating system consisting of Excellent, Good, Average, Poor, and Extremely Poor by DeepSeek-V3. The divergence of the simulator-generated terminal thought is calculated using the score mapped from the annotated terminal user thought in the test data as the golden result. See Appendix C for additional details on computing the CV-M divergence.

Method	Divergence ↓
Prompt Thinking	0.1583
ThinkingUS (ours)	**0.0687**

Table 2: Terminal user evaluation results. A lower divergence indicates that the simulator's evaluation of the assistant is closer to the golden results.

Table 2 shows the CV-M divergence achieved by Prompt Thinking and ThinkingUS on the dialogues of the LMSYS test set. ThinkingUS achieves the lower divergence than the Prompt Thinking method, which indicates that ThinkingUS has a higher degree of consistency between the distribution of its evaluations of the assistant in the final turn and that of the real user evaluations.

4.2 Online Evaluation

In the setting of online evaluation, the user simulator engages in online multi-turn interactions with the assistant LM based on a given scenario and user motivation, aiming to accomplish user goals within the specified scenario. Unlike offline evalu-

Type	Criteria
Disclosure	Score 1: The user's utterances reflect this goal (the interpretation can be inferred from the dialogue). Score 0: The relationship is unclear or ambiguous (insufficient evidence to infer). Score -1: The user's utterances conflict with this goal (the interpretation contradicts the dialogue).
Progression	Score 1: Dialogue progresses toward the user's goals and user logic is consistent. Score 0: User goals are stagnating or regressing.
Additional Information	Score 1: if at least one user utterance introduces information beyond the original scenario setting. Score 0: if none of the user utterances introduce information beyond the original scenario setting.

Table 3: Scoring criteria for both goal orientation and behavioral pattern aspects.

Method	LMSYS-Chat			WildChat		
	Disc.	Prog.	Add Info.	Disc.	Prog.	Add Info.
Utterance SFT	0.2110	0.2846	0.2340	0.2903	0.3120	0.2700
ThinkingUS (on Instruct)	0.3899	**0.5292**	0.4120	0.4439	**0.4950**	0.3420
ThinkingUS (on Base)	**0.4318**	0.4297	**0.4220**	**0.4968**	0.4620	**0.4140**

Table 4: Online evaluation results of three simulators (non-thinking and thinking alignment method). Each simulator is evaluated on its coverage of the goal disclosure, dialogue progression, and the additional information introduction.

ation—where each sample input to the user simulator comes from a test set and the dialog history in the sample is not actually generated by the simulator being evaluated—online evaluation requires the user simulator to respond in real time based on its own actual interaction history. This approach reflects the simulator's performance across multiple turns better.

Setup. We select scenarios from LMSYS-Chat and WildChat test set for online evaluation. Specifically, we select 500 scenarios from each test set randomly for online conversation, and train a classifier to end the interaction. In all conversations, we use the same assistant LM (Llama-3.1-8B-Instruct) and the same dialogue end prediction model, keeping these aspects of the simulation fixed.

Metrics. We evaluate the user simulator from two aspects: goal orientation and behavioral patterns. For the goal orientation aspect, we develop evaluation metrics from two dimensions: referential (comparison with source data) and inherent (assessment of independent quality). In the referential dimension, we focus on **goal disclosure** (Disc.), which measures whether the simulator adequately reveals the intention information from the scenario setting during the dialogue process. In the inherent dimension, we focus on **dialogue progression** (Prog.), which evaluates whether the simulator follows an intrinsic goal to advance or steer the dialogue. For the behavioral pattern aspect, we emphasize the ability to introduce **additional information** (Add Info.), since human users typically provide extra information to the assistant model based on their cognition when the goals are not yet achieved, in order to facilitate goal accomplishment. For the specific implementation, for the disclosure metric, we adapt the methodology of FactScore (Yao et al., 2023). The scenario settings are decomposed into atomic goals through DeepSeek-V3. Then, each <dialogue, atom goal> pair is scored based on a predefined rule by DeepSeek-V3. The final score for a target is computed as the average score of all its atomic goals. The progression metric directly scores multi-turn dialogues based on scoring criteria, while the additional information metric assigns scores by comparing the scenario setup with the dialogue content. Further criteria details are provided in Table 3. All metrics are evaluated by prompting DeepSeek-V3 as the judge.

Results. First, the thinking user simulator significantly outperforms the non-thinking simulator in both goal orientation and behavioral patterns. Second, the user simulator based on the base model surpassed the model based on the instruction-tuned version on most metrics, further demonstrating that

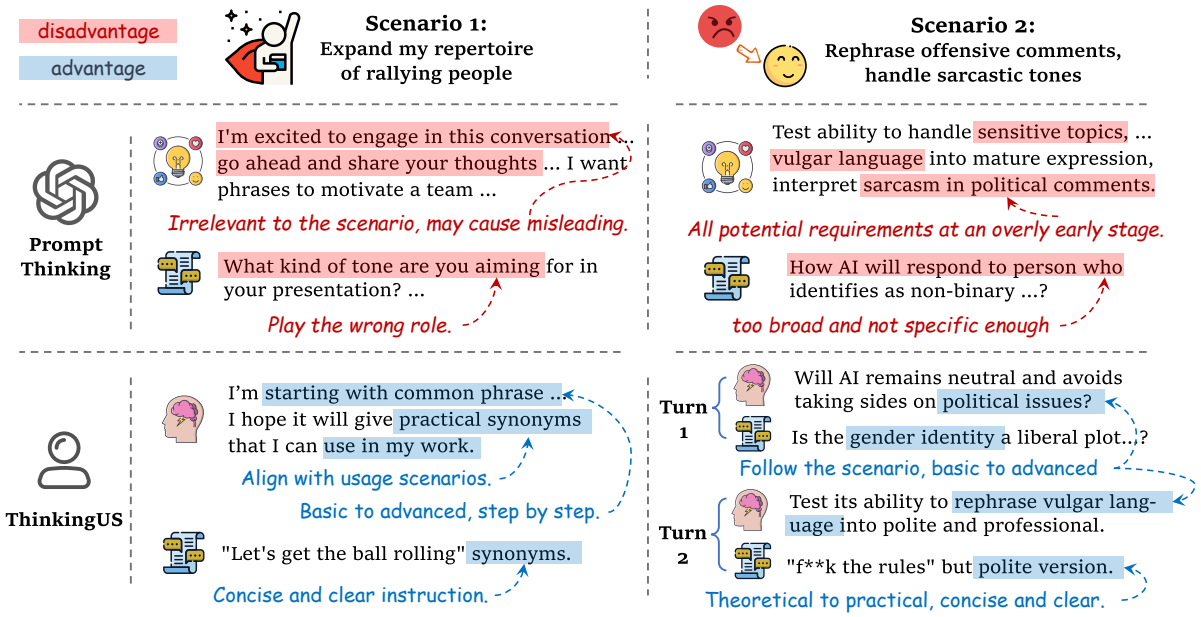

Figure 3: Case study of Prompt Thinking vs. ThinkingUS in multi-turn dialogues. Red highlight is used to indicate poorer-performing thoughts and behaviors, such as those that are scenario-irrelevant or involve role misalignment; blue highlight denotes better-performing thoughts and behaviors, which are scenario-adherent and exhibit human-like qualities.

base models are more suitable as initial models for training user simulators, which is consistent with offline evaluation results. Third, thinking user simulators achieve notably higher scores in goal disclosure than non-thinking simulators, typically outperforming the latter by 70% to 80%. Although the disclosure scores of all three simulators still fall short of the maximum, this aligns with real human user behavior, as humans also selectively disclose primary intentions while avoiding unnecessary details. Finally, all three simulators—which utilized real human dialogue data to enhance alignment with human behavior—are able to introduce additional information not specified in the original scenario setting. Among them, the thinking simulators incorporate more such information to facilitate dialogue progression.

4.3 Case Study

We select representative cases to provide a comparative exposition of the thought process between prompting Thinking and ThinkingUS, as presented in Figure 3. There are significant differences in cognitive style between the two methods. The thought process of Prompt Thinking is conducted from the assistant's perspective, encompassing not only the user's own thinking patterns but also a simulated chain of thought that mimics these patterns. This can sometimes cause the user simulator to shift from being a questioner to a problem-solver, resulting in role confusion (shown in *Scenario 1* in Figure 3). Furthermore, Prompt Thinking tends to present all requirements to the LLM at an earlier stage, and then replaces keywords for the same type of questions in the middle and later phases of the conversation, by conducting hypothetical extensions and conceptual expansions of the current dialogue context. The content generated by ThinkingUS is more concise, with its summaries of previous-turn content and reasoning for subsequent-turn responses being more straightforward. Taking *Scenario 2* in Figure 3 as an example, the thinking mode of ThinkingUS tends to follow a from-easy-to-hard progression: it proceeds from general scenarios, to specific one, and from conceptual hypotheses to practical problem-solving, and is more likely to terminate the current topic once a satisfactory answer is obtained. In contrast, Prompt Thinking retains the characteristic of prompting the LLM to sustain the conversation, although this approach enriches the content of the conversation, it increases the deviation from the thinking patterns of real human beings.

5 Conclusion

In this work, we train models to achieve "cognitive simulation" that goes beyond mere behavioral imi-

tation, aligning the thought patterns and cognitive processes of user simulators with those of human users to guide their decision-making. We construct a dataset for user thought alignment and compare the performance of thinking versus non-thinking, prompt-based versus SFT-based user simulators under multiple settings, including offline test data and online real-time dialogues. The results show that thinking user simulators significantly enhance the human-like quality of behavioral pattern imitation. Nevertheless, a gap remains compared to real human behavior. Our future work will explore more granular multi-level cognitive simulation and improved internal consistency.

Limitations

This study has several limitations that should be considered when interpreting the results. First, although the training data consisted of a dialogue dataset derived from real human users, factors such as interaction purposes, interlocutors, and time constraints may have introduced biases in the comprehensiveness and diversity of scenarios, which could deviate from real-world usage contexts. Second, the user thoughts constructed in this work were based on dialogue context completion from real conversations, which may not fully align with the actual cognitive processes of human users. Despite these limitations, this study—through offline and online evaluation experiments across multiple base models—is the first to demonstrate that thinking alignment can enhance the ability of user simulators to emulate human behavioral patterns more closely. Future work will explore more fine-grained, multi-level cognitive modeling and improve internal consistency in simulation.

Acknowledgments

This work is funded by China Mobile Strategic Project (R26110S3, R24113J4).

Ethics Considerations

This study strictly adheres to ethical guidelines and responsible AI practices. All data are obtained from public, authorized dialogue datasets (LMSYS-Chat-1M, WildChat) with full anonymization. The constructed LMSYS-UserThinking dataset and ThinkingUS simulator are used exclusively for academic research to evaluate and improve assistant models, not for impersonation, deception, or harmful applications. This work prioritizes privacy, fairness, transparency, and safety to promote ethical development of interactive user simulation technologies.

References

Kaustubh Dhole. 2024. KAUCUS - knowledgeable user simulators for training large language models. In *Proceedings of the 1st Workshop on Simulating Conversational Intelligence in Chat (SCI-CHAT 2024)*, pages 53–65, St. Julians, Malta. Association for Computational Linguistics.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 3029–3051. Association for Computational Linguistics.

Vardhan Dongre, Xiaocheng Yang, Emre Can Acikgoz, Suvodip Dey, Gokhan Tur, and Dilek Hakkani-Tur. 2025. ReSpAct: Harmonizing reasoning, speaking, and acting towards building large language model-based conversational AI agents. In *Proceedings of the 15th International Workshop on Spoken Dialogue Systems Technology*, pages 72–102, Bilbao, Spain. Association for Computational Linguistics.

Rafael Ferreira, David Semedo, and Joao Magalhaes. 2024. Multi-trait user simulation with adaptive decoding for conversational task assistants. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16105–16130, Miami, Florida, USA. Association for Computational Linguistics.

Christine Herlihy, Jennifer Neville, Tobias Schnabel, and Adith Swaminathan. 2024. On overcoming miscalibrated conversational priors in LLM-based chatbots. In *The 40th Conference on Uncertainty in Artificial Intelligence*.

Friederike Holderried, Christian Stegemann-Philipps, Anne Herrmann-Werner, Teresa Festl-Wietek, Martin Holderried, Carsten Eickhoff, Moritz Mahling, and 1 others. 2024. A language model–powered simulated patient with automated feedback for history taking: Prospective study. *JMIR Medical Education*, 10(1):e59213.

EunJeong Hwang, Bodhisattwa Majumder, and Niket Tandon. 2023. Aligning language models to user opinions. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5906–5919, Singapore. Association for Computational Linguistics.

Jonathan Ivey, Shivani Kumar, Jiayu Liu, Hua Shen, Sushrita Rakshit, Rohan Raju, Haotian Zhang, Aparna Ananthasubramaniam, Junghwan Kim, Bowen Yi, Dustin Wright, Abraham Israeli, Anders Giovanni Møller, Lechen Zhang, and David

Jurgens. 2024. Real or robotic? assessing whether llms accurately simulate qualities of human responses in dialogue. *ArXiv*, abs/2409.08330.

Jiaming Ji, Boyuan Chen, Hantao Lou, Donghai Hong, Borong Zhang, Xuehai Pan, Tianyi Qiu, Juntao Dai, and Yaodong Yang. 2024. Aligner: Efficient alignment by learning to correct. *Advances in Neural Information Processing Systems 37*.

Priyanka Kargupta, Ishika Agarwal, Tal August, and Jiawei Han. 2025a. Tree-of-debate: Multi-persona debate trees elicit critical thinking for scientific comparative analysis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29378–29403, Vienna, Austria. Association for Computational Linguistics.

Priyanka Kargupta, Ishika Agarwal, Dilek Hakkani Tur, and Jiawei Han. 2024. Instruct, not assist: LLM-based multi-turn planning and hierarchical questioning for socratic code debugging. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9475–9495, Miami, Florida, USA. Association for Computational Linguistics.

Priyanka Kargupta, Shuyue Stella Li, Haocheng Wang, Jinu Lee, Shan Chen, Orevaoghene Ahia, Dean Light, Thomas L. Griffiths, Max Kleiman-Weiner, Jiawei Han, Asli Celikyilmaz, and Yulia Tsvetkov. 2025b. Cognitive foundations for reasoning and their manifestation in llms. *Preprint*, arXiv:2511.16660.

Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, Xin Zhou, Enzhi Wang, and Xiaohang Dong. 2024a. Better zero-shot reasoning with role-play prompting. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4099–4113, Mexico City, Mexico. Association for Computational Linguistics.

Chuyi Kong, Yaxin Fan, Xiang Wan, Feng Jiang, and Benyou Wang. 2024b. PlatoLM: Teaching LLMs in multi-round dialogue via a user simulator. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7841–7863, Bangkok, Thailand. Association for Computational Linguistics.

Ruosen Li, Ruochen Li, Barry Wang, and Xinya Du. 2024. IQA-EVAL: automatic evaluation of human-model interactive question answering. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Yajiao Liu, Xin Jiang, Yichun Yin, Yasheng Wang, Fei Mi, Qun Liu, Xiang Wan, and Benyou Wang. 2023. One cannot stand for everyone! leveraging multiple user simulators to train task-oriented dialogue systems. In *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.

Suhong Moon, Marwa Abdulhai, Minwoo Kang, Joseph Suh, Widyadewi Soedarmadji, Eran Kohen Behar, and David M. Chan. 2024. Virtual personas for language models via an anthology of backstories. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19864–19897, Miami, Florida, USA. Association for Computational Linguistics.

Tarek Naous, Philippe Laban, Wei Xu, and Jennifer Neville. 2025. Flipping the dialogue: Training and evaluating user language models. *ArXiv*, abs/2510.06552.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. 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. Association for Computing Machinery.

Cheng Qian, Zuxin Liu, Akshara Prabhakar, Zhiwei Liu, Jianguo Zhang, Haolin Chen, Heng Ji, Weiran Yao, Shelby Heinecke, Silvio Savarese, Caiming Xiong, and Huan Wang. 2025a. Userbench: An interactive gym environment for user-centric agents. *Preprint*, arXiv:2507.22034.

Cheng Qian, Zuxin Liu, Akshara Prabhakar, Jielin Qiu, Zhiwei Liu, Haolin Chen, Shirley Kokane, Heng Ji, Weiran Yao, Shelby Heinecke, Silvio Savarese, Caiming Xiong, and Huan Wang. 2025b. Userrl: Training interactive user-centric agent via reinforcement learning. *Preprint*, arXiv:2509.19736.

Ivan Sekuli'c, Silvia Terragni, Victor Guimarães, Nghia Khau, Bruna Guedes, Modestas Filipavicius, André Ferreira Manso, and Roland Mathis. 2024. Reliable llm-based user simulator for task-oriented dialogue systems. *ArXiv*, abs/2402.13374.

Ryan Shea, Yunan Lu, Liang Qiu, and Zhou Yu. 2025. Sage: A top-down bottom-up knowledge-grounded user simulator for multi-turn agent evaluation. *ArXiv*, abs/2510.11997.

Dazhen Wan, Zheng Zhang, Qi Zhu, Lizi Liao, and Minlie Huang. 2022. A unified dialogue user simulator for few-shot data augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3788–3799, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kuang Wang, Xianfei Li, Shenghao Yang, Li Zhou, Feng Jiang, and Haizhou Li. 2025a. Know you first and be you better: Modeling human-like user simulators via implicit profiles. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21082–21107, Vienna, Austria. Association for Computational Linguistics.

Lei Wang, Jingsen Zhang, Hao Yang, Zhi-Yuan Chen, Jiakai Tang, Zeyu Zhang, Xu Chen, Yankai Lin, Hao Sun, Ruihua Song, Xin Zhao, Jun Xu, Zhicheng Dou, Jun Wang, and Ji-Rong Wen. 2025b. User behavior simulation with large language model-based agents. *ACM Trans. Inf. Syst.*, 43(2).

Xintao Wang, Heng Wang, Yifei Zhang, Xinfeng Yuan, Rui Xu, Jen-tse Huang, Siyu Yuan, Haoran Guo, Jiangjie Chen, Shuchang Zhou, Wei Wang, and Yanghua Xiao. 2025c. Coser: Coordinating llm-based persona simulation of established roles. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net.

Tianjun Wei, Huizhong Guo, Yingpeng Du, Zhu Sun, Chen Huang, Dongxia Wang, and Jie Zhang. 2025. Mirroring users: Towards building preference-aligned user simulator with user feedback in recommendation. *arXiv preprint arXiv:2508.18142*.

J. Williams. 2007. A method for evaluating and comparing user simulations: The cramér-von mises divergence. *2007 IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, pages 508–513.

Ning Wu, Ming Gong, Linjun Shou, Shining Liang, and Daxin Jiang. 2023. Large language models are diverse role-players for summarization evaluation. In *Natural Language Processing and Chinese Computing*, pages 695–707, Cham. Springer Nature Switzerland.

Shujin Wu, Yi R. Fung, Cheng Qian, Jeonghwan Kim, Dilek Hakkani-Tur, and Heng Ji. 2025. Aligning llms with individual preferences via interaction. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 7648–7662. Association for Computational Linguistics.

Weiqi Wu, Hongqiu Wu, Lai Jiang, Xingyuan Liu, Hai Zhao, and Min Zhang. 2024. From role-play to drama-interaction: An LLM solution. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3271–3290, Bangkok, Thailand. Association for Computational Linguistics.

Canwen Xu, Daya Guo, Nan Duan, and Julian J. McAuley. 2023. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 6268–6278. Association for Computational Linguistics.

Yuxuan Yao, Han Wu, Qiling Xu, and Linqi Song. 2023. Fine-grained conversational decoding via isotropic and proximal search. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 58–70, Singapore. Association for Computational Linguistics.

Tong Zhang, Chen Huang, Yang Deng, Hongru Liang, Jia Liu, Zujie Wen, Wenqiang Lei, and Tat-Seng Chua. 2024. Strength lies in differences! improving strategy planning for non-collaborative dialogues via diversified user simulation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 424–444, Miami, Florida, USA. Association for Computational Linguistics.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. Wildchat: 1m chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. 2024. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. *Preprint*, arXiv:2309.11998.

A Prompt

Scenario Setting and User Thought Generation. Figure 4 shows the prompt we used to generate scenario settings and user thoughts of LMSYS-Chat-1M and Wildchat datasets. We provide DeepSeek-V3 with the full conversation history between the user and the assistant, instructing it to generate a summary of the scenario setting based on the holistic context of the dialogue, as well as to produce contextualized user thoughts preceding each user utterance.

LLM-as-Judge for Authenticity. Figure 5 presents the prompt used to evaluate the authenticity of user responses generated by different methods. Using golden responses from real human users within the same conversational context as references, the comparison assesses the similarity to these golden responses along two dimensions: intent and expression style.

B Extra Offline Evaluation

To validate the generalization capability of the thinking-aligned user simulation method across different model architectures and sizes, we conduct additional supervised fine-tuning experiments on the Qwen3 model series. Specifically, we selected two base models, Qwen3-8B and Qwen3-4B, and train them using two schemes: supervised fine-tuning based solely on user utterances (Utterance SFT) and the proposed thinking-aligned fine-tuning method (ThinkingUS). We evaluate the similarity between the generated user utterances and real human utterances on both the in-domain LMSYS-

Base Model	Method	LMSYS-Chat				WildChat			
		win.	loss.	draw.	r_ω	win.	loss.	draw.	r_ω
Qwen3-8B	Utterance SFT	10769	3949	297	0.4542	2459	904	18	0.4599
	ThinkingUS (ours)	11304	3432	279	**0.5243**	2628	729	24	**0.5617**
Qwen3-4B	Utterance SFT	10815	3894	306	0.4609	2468	885	28	0.4682
	ThinkingUS (ours)	11333	3436	246	**0.5259**	2636	723	22	**0.5658**

Table 5: Offline evaluation results for Qwen3 models with different sizes.

Chat test set and the out-of-domain WildChat test set. The evaluation follows the win-rate metric r_ω based on the DeepSeek-V3 judge model described in Section 4.1.1. Table 5 details the counts of wins, losses, ties, and final win rates for each model on the two test sets.

On the Qwen3-8B model, Utterance SFT achieves a win rate of 0.4542 on the LMSYS-Chat test set, whereas the ThinkingUS method boosts the win rate to 0.5243, representing an absolute improvement of 7.01 percentage points and a relative improvement of approximately 15.4%. On the more challenging out-of-domain WildChat test set, the performance gain is even more pronounced: the win rate jumps from 0.4599 to 0.5617, an absolute increase of 10.18 percentage points and a relative increase of about 22.1%. Similarly, on the smaller Qwen3-4B model, the ThinkingUS method demonstrates consistent gains: the win rate on LMSYS-Chat increases from 0.4609 to 0.5259 (+6.5 percentage points, relative +14.1%), and on WildChat from 0.4682 to 0.5658 (+9.76 percentage points, relative +20.8%). Overall, the ThinkingUS method significantly outperforms the utterance-only fine-tuning approach on both Qwen3-8B and Qwen3-4B, validating the universal effectiveness of the thinking alignment strategy in enhancing the authenticity of user simulation.

C Calculation of CV-M divergence

We employ the Cramér–von Mises (CV-M) divergence (Williams, 2007) to quantify the agreement of the score distribution regarding the assistant between user simulators and real users:

$$D(F_0||F_1) = \alpha \sqrt{\sum_{i=1}^{N_0} (F_0(x^0_{(i)}) - F_1(x^0_{(i)}))^2}, \tag{4}$$

where F_j denotes the empirical distribution function (EDF) of the score list $X_j = (x^j_{(1)}, ..., x^j_{(N_j)})$. F_0 and F_1 are the EDF of the golden scores and the predicted scores, respectively, which can be calculated as:

$$F_j(x) = \frac{1}{N_j} \sum_{i=1}^{N_j} \begin{cases} 1 & \text{if } x^j_{(i)} < x \\ \frac{1}{2} & \text{if } x^j_{(i)} = x \\ 0 & \text{if } x^j_{(i)} > x \end{cases}. \tag{5}$$

$\alpha = \sqrt{((12N_0)/(4N_0^2 - 1))}$ is a normalizing constant which scales $D(F_0||F_1)$ to the range [0, 1]. The score x_i refers to the numerical value obtained by mapping the thoughts about assistant of users in the final turn of a conversation to a fixed interval. We map the thoughts to a 5-level rating system consisting of Excellent, Good, Average, Poor, and Extremely Poor. Here, x^0_i and x^1_i denote the golden score and the predicted score, respectively.

D LLM Usage Clarification

Throughout the paper, the use of LLMs is solely restricted to the polishing of textual elements, such as lexical or phrasal substitutions, and does not extend beyond this scope.

Prompt Template for Scenario Setting and User Thought Generation

```
Complete the dialogue scenario, user motivation, and user thoughts for the
given multi-turn dialogue, adhering to the following rules:

1. The dialogue scenario refers to the background information of the
conversation, and user motivation refers to the starting point or purpose of
the user's dialogue. The dialogue scenario and user motivation apply globally
to the entire multi-turn dialogue.
2. User thoughts are specific to each round of the assistant model's
response, containing the user's unexpressed inner activities, including the
user's emotions and motivations for speaking, enclosed within "<thought>" and
"</thought>" tags.
3. You should carefully consider the user's thoughts and simulate a human user
as realistically as possible.
4. Do not alter the original text's output; only supplement the user's inner
thoughts for each round of dialogue.
5. When supplementing user thoughts after the assistant's response in the final
round, include the reason for ending the dialogue.

Input Example:

Output Example:

Now, please complete the dialogue scenario, user motivation, and user
thoughts for the following dialogue.

{DIALOGUE}
```

Figure 4: Prompt template used to generate scenario setting and user thought from conversations with DeepSeek-V3.

Prompt Template for Offline Evaluation

```
** Task Description:
Please evaluate which of the two user simulator outputs is closer to the golden
output based on the given dialogue scenario, dialogue history, and the real human
user's output (golden). Your response should clearly indicate the choice (A
or B) and provide a brief explanation based on user intent and expression style.

** Input Information:
 - Dialogue Scenario:
 - Dialogue History:
 - Real Human User Output (Golden):
 - User Simulator Output A:
 - User Simulator Output B:

** Output Requirements:
1. Judgment Result: Clearly state which user simulator output (A or B) is closer
to the golden output.
2. Explanation: Briefly explain the reasoning, focusing on:
 - User Intent: Compare the consistency between the simulator output and the
golden output in terms of goals, needs, or objectives.
 - Expression Style: Compare the similarity between the simulator output and
the golden output in terms of language style, tone, and level of detail (e.g.,
whether similar expressions, keywords, or emotional tones are used).
3. You should carefully consider the user's thoughts and simulate a human user
as realistically as possible.
 - No Length Bias: Do not assume longer responses are better.
 - No Position Bias: Do not favor A or B based on order.

First, begin your evaluation by comparing the two statements with the
golden user utterance, explaining which one is consistent with the golden
standard in terms of user intent and expression style.

After your explanation, output your final verdict strictly as: "[[A]]"
if user simulator A is better, "[[B]]" if user simulator B is better, or
"[[C]]" for a tie.
```

Figure 5: Prompt template used to benchmark responses of each method against the Prompt outputs.