SpeakRL: Synergizing Reasoning, Speaking, and Acting in Language Models with Reinforcement Learning

Emre Can Acikgoz[1], **Jinoh Oh**[2], **Jie Hao**[2], **Joo Hyuk Jeon**[2],
Heng Ji[2], **Dilek Hakkani-Tür**[2], **Gokhan Tur**[2], **Xiang Li**[2], **Chengyuan Ma**[2], **Xing Fan**[2]
[1]University of Illinois Urbana-Champaign, [2]Amazon Alexa
acikgoz2@illinois.edu, ojino@amazon.com

Abstract

Effective human-agent collaboration is increasingly prevalent in real-world applications. Current trends in such collaborations are predominantly unidirectional, with users providing instructions or posing questions to agents, where agents respond directly without seeking necessary clarifications or confirmations. However, the evolving capabilities of these agents require more proactive engagement, where agents should dynamically participate in conversations to clarify user intents, resolve ambiguities, and adapt to changing circumstances. Existing prior work under-utilize the conversational capabilities of language models (LMs), thereby optimizing agents as better followers rather than effective speakers. In this work, we introduce **SpeakRL**, a reinforcement learning (RL) method that enhances agents' conversational capabilities by rewarding proactive interactions with users, such as asking right clarification questions when necessary. To support this, we curate **SpeakER**, a synthetic dataset that includes diverse scenarios from task-oriented dialogues, where tasks are resolved through interactive clarification questions. We present a systematic analysis of reward design for conversational proactivity and propose a principled reward formulation for teaching agents to balance *asking* with *acting*. Empirical evaluations demonstrate that our approach achieves a 20.14% absolute improvement in task completion over base models without increasing conversation turns even surpassing even much larger proprietary models, demonstrating the promise of clarification-centric user-agent interactions.

1 Introduction

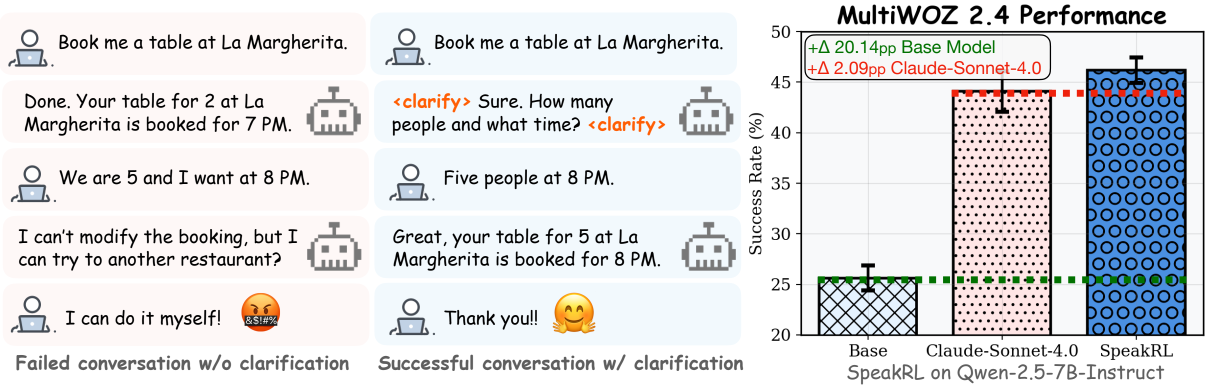
The integration of language models (LMs) into real-world applications has transformed human–agent collaboration, enabling systems that assist users with tasks ranging from planning travel itineraries (Xie et al., 2024; Yao et al., 2024) to managing smart home ecosystems (Gottardi et al., 2022; Padmakumar et al., 2022). However, during these interactions, agents often encounter vague or underspecified user queries, making task execution more challenging. In such situations, the agent must either make a potentially risky assumption or fail to complete the task (Purver et al., 2001).

Mechanistically, asking clarification questions serves as a proactive error-correction mechanism in conversational agents: by querying for missing details early, agents minimize uncertainty, refine task understanding, and prevent downstream failures (Acikgoz et al., 2025d). This dynamic is illustrated in Figure 1, where an ambiguous restaurant booking request fails without clarification (left) but succeeds when the agent seeks key details (right), highlighting clarification's role in robust, multi-turn dialogues. Thus, we treat clarification as a control primitive: detect underspecification, ask for the missing variables, then execute. This loop grounds actions in user constraints—reducing plan entropy, avoiding risky commitments, and yielding robust task completion.

Existing methods for integrating LMs with clarification capabilities generally fall into two categories: (i) designing hand-crafted, rule-based prompts with predefined instructions (Dongre et al., 2024) and (ii) fine-tuning models explicitly to generate clarification questions (Zhang and Choi, 2025; Zhang et al., 2025) for better interactions with users (Li et al., 2025; Andukuri et al., 2024). In parallel to these, reinforcement learning (RL) (Sutton and Barto, 2018) has gained significant traction in improving reasoning capabilities in Large Reasoning Models (LRMs) like OpenAI-o1 (Jaech et al., 2024) and DeepSeek-R1 (Guo et al., 2025), employing techniques such as GRPO (Shao et al., 2024a) to enhance problem-solving skills through experiential reward feedback.

However, applying reinforcement learning with verifiable rewards (RLVR) to interactive user clari-

Proceedings of the 16th International Workshop on Spoken Dialogue Systems Technology, pages 312–325
February 26–March 1, 2026. ©2026 Association for Computational Linguistics



Book me a table at La Margherita.

Done. Your table for 2 at La Margherita is booked for 7 PM.

We are 5 and I want at 8 PM.

I can't modify the booking, but I can try to another restaurant?

I can do it myself!

Book me a table at La Margherita.

<clarify> Sure. How many people and what time? <clarify>

Five people at 8 PM.

Great, your table for 5 at La Margherita is booked for 8 PM.

Thank you!!

Figure 1: **Impact of SpeakRL on Success Rate Performance in MultiWOZ 2.4. Left:** Example dialogues showing failure without clarification (left) versus success with proactive clarification (right). **Right:** Success rates for the Base Model (25.63%), Claude-Sonnet-4.0 (44.08%), and SpeakRL (46.17%) on MultiWOZ 2.4. SpeakRL attains 80% higher success rate than the Base Model and ∼5% higher than Claude-Sonnet-4.0, demonstrating the substantial impact of reinforcement-learned clarification in multi-turn conversation settings.

fication remains underexplored and introduces several challenges: (i) **Multi-turn Conversation and Clarification**—the agent must balance between directly responding and selectively requesting clarification only when ambiguity arises, which demands sophisticated multi-turn interaction capabilities; (ii) **Reward Design**—creating an effective reward function that clearly defines when and what to ask for clarification remains challenging, as it is uncertain whether simple outcome-based rewards can guide agents to consistently and meaningfully generate clarification questions; and (iii) **RL Optimization**—integrating clarification behaviors into RL training for LLMs in a stable and efficient manner is still an open problem.

To address the aforementioned challenges, we introduce SpeakRL, a novel RLVR algorithm that empowers LLMs to resolve ambiguity through user-directed clarification in multi-turn conversations by learning both *when* and what to *ask* to fill particular slots (Lison, 2013; Louvan and Magnini, 2020). Our SpeakRL uses Group Relative Policy Optimization (GRPO), an on-policy RL algorithm, with two complementary verifiable reward signals. We first introduce structured special tokens that separates internal uncertainty reasoning (**<think>**. . . **</think>**) and clarification questions (**<clarify>**. . . **<clarify>**), giving the agent precise control over the timing of clarification requests by learning to produce these tokens appropriately. Second, we define an LLM-as-judge reward model that evaluates and optimizes the quality of clarification questions, teaching the model to formulate more effective queries. Together, this RLVR-based optimization enables agents to learn strategic question-asking behavior without explicit task completion rewards. Notably, by optimizing solely for clarification quality rather than task success, we demonstrate that effective interactive conversation naturally leads to more successful and efficient task completion.

In summary, our primary contributions are as follows: (1) We introduce SpeakRL, an end-to-end RLVR framework that enables LLM agents to iteratively improve their ability to ask clarification questions in multi-turn, goal-oriented dialogues. (2) We construct SpeakER, a synthetic dataset of 25,000 task-oriented multi-turn conversations, explicitly designed to include ambiguous scenarios annotated with user clarification turns. (3) We design reward strategies within RLVR that guide agents on both when and what clarification questions to ask. (4) We show that post-training with SpeakRL enables LLMs to proactively ask clarification questions in uncertain or ambiguous contexts, improving task success while reducing dialogue length, thereby fostering more accurate and efficient collaborative human–agent interactions.

2 Related Work

Reinforcement Learning for Task-Oriented Dialogue. RL has been applied to learn dialogue behaviors beyond supervised imitation, from optimizing open-ended generation with long-horizon rewards (Li et al., 2016) to traditional TOD agents that act for information access (Dhingra et al., 2017). Prior work also explores interactive improvements via self-play and online RL (Shah et al., 2018), collaborative multi-agent RL dialogue training (Papangelis et al., 2019), and the importance of action-space design for effective RL (Zhao et al., 2019). Motivated by the fact that modern LLM-based TOD systems are end-to-end (Acikgoz et al., 2025b), we use RLVR to directly reward clarification behavior in natural language, enabling a simple and end-to-end pipeline that teaches the model when and how to clarify without hand-crafted states or rigid dialog acts.

Reinforcement Learning and LLMs. RL has been incorporated into LLMs mainly via Reinforcement Learning from Human Feedback (RLHF), which trains a reward model from human preferences and optimizes the policy using PPO (Ouyang et al., 2022; Schulman et al., 2017). However, PPO is often unstable and requires careful hyperparameter tuning. To mitigate these issues, simpler alternatives such as Direct Preference Optimization (DPO) have been proposed, which learn directly from preference pairs without explicit reward modeling, along with several efficient variants (Rafailov et al., 2023; Ethayarajh et al., 2024; Hong et al., 2024; Meng et al., 2024). Although these methods improve computational efficiency, they are prone to off-policy issues and often fall short of the performance achieved by traditional RL techniques (Pang et al., 2024). More recently, Group Relative Policy Optimization (GRPO) has been proposed (Shao et al., 2024b; Guo et al., 2025), which bypasses the need for a reward model by employing a group-based evaluation approach and demonstrates robust enhancements in reasoning capabilities across diverse tasks and domains (Qian et al., 2025; Jin et al., 2025; Lai et al., 2025; Huang et al., 2025b). Nevertheless, the use of RL to train conversational agents for greater proactivity remains a largely untapped area of research.

Asking Clarification Questions. Prior work addresses ambiguity in user requests by teaching LLMs to ask clarification questions, using either prompting-based approaches with hand-engineered instructions (Zhang and Choi, 2025; Dongre et al., 2024) or explicit training methods (Zhang et al., 2025; Andukuri et al., 2024; Wu et al., 2025; Chen et al., 2025; Kobalczyk et al., 2025). Training-based methods employ various techniques, including supervised fine-tuning (SFT) (Andukuri et al., 2024), reinforcement learning (Chen et al., 2025; Wu et al., 2025), direct preference optimization with positive and negative samples (Zhang et al., 2025; Chen et al., 2025), and active learning (Kobalczyk et al., 2025). However, most of these approaches focus primarily on clarification question generation in isolation and underutilize the complexity of multi-turn conversational dynamics, with notable exceptions being Dongre et al. (2024) and Wu et al. (2025). Dongre et al. (2024) explores multi-turn settings but relies on hand-engineered prompts where speaking actions are conditioned as policies for specific situations, limiting generalizability. Closest to our work, Wu et al. (2025) train LLMs to ask clarification questions in multi-turn settings using user simulators and reward signals. However, their setup is domain-general and does not require task completion or agentic behaviors such as tool use, limiting realism for task-oriented settings. In contrast, we apply RLVR in multi-turn TOD with user feedback tied to task completion, requiring function calling and tool use to achieve realistic agentic behavior.

3 SpeakRL

Overview. On high level, SpeakRL teach LLM Agents to identify when user requests are underspecified or ambiguous and to ask targeted clarification questions that maximize task success in goal-oriented dialogues. Each conversation begins with a user request that the agent attempts to fulfill through iterative interaction, potentially spanning multiple related tasks (e.g., purchasing train tickets followed by booking nearby accommodations). For actual realistic conversations as in real-life, we simulate human users through a user simulator that takes both a goal message and a role-defining prompt as input (Xu et al., 2024). The agent, instantiated as an LLM, iteratively reasons about whether a clarification question is necessary and, if so, formulates the appropriate query. Over time, the agent progressively learns to enhance its internal reasoning abilities (Zelikman et al., 2022), allowing it to better grasp ambiguities and determine what to ask, thereby improving the quality of its clarification questions. Through this iterative process, the agent improves both its reasoning capabilities and clarification question quality. Conversations terminate either when the agent completes the task and signals completion, or when the maximum dialogue length of 20 turns is reached.

3.1 Task Definition

Task-oriented dialogues (TOD) can be viewed as *multi-step reasoning processes* where an agent interacts with a user (or user simulator) to accomplish a concrete goal—such as booking, ordering, or scheduling—through successive decisions and

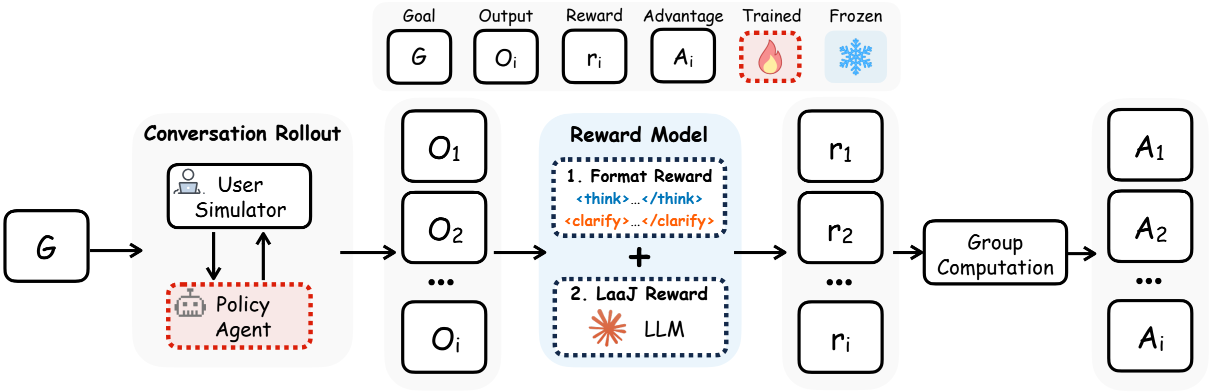


Figure 2: **GRPO algorithm with user clarification for collaborative RL.** Demonstration of GRPO training for teaching asking clarification questions (SpeakRL). During rollout, the policy agent conducts multi-turn interactions, with outputs rewarded according to format compliance and LLM-as-a-Judge (LaaJ) scores.

clarifications. Each decision in the dialogue depends on the accumulated interaction history, and the final outcome is determined by the cumulative success of these intermediate reasoning steps.

Formally, let the agent's action space be:

$$\mathcal{A} = a_1, a_2, \ldots, a_n, \tag{1}$$

where each action $a_i \in \mathcal{A}$ corresponds to one of three categories: (i) asking a clarification question, (ii) generating an actual response, or (iii) executing an API call. Given a user goal $\mathcal{G}$, the dialogue trajectory up to step k is defined as:

$$s_k = (r_1, a_1, o_1), \ldots, (r_k, a_k, o_k), \tag{2}$$

where r_i represents the agent's reasoning or internal plan at step (i), a_i denotes the chosen action, and o_i represents the observation received after executing a_i, which may include user or environment feedback.

At each step $k{+}1$, the agent interprets the current dialogue state, generates the next reasoning step r_{k+1}, selects an action $a_{k+1} \in \mathcal{A}$, and produces the corresponding utterance or API call to advance toward fulfilling $\mathcal{G}$. The agent's policy is defined as:

$$\pi : s_k \rightarrow (r_{k+1}, a_{k+1}), \tag{3}$$

with the objective of selecting the optimal action that maximizes expected reward:

$$a^*_{k+1} = \arg \max_{a_{k+1} \in A} R(s_k, a_{k+1}, o_{k+1}), \tag{4}$$

where $\mathcal{R}(\cdot)$ evaluates progress made after performing the action—reflecting factors such as effective clarification, correct slot acquisition, or successful task advancement.

While immediate rewards encourage effective reasoning and interaction at each step, the policy π is optimized to maximize the cumulative reward across the dialogue trajectory:

$$\max_{\pi} \; \mathbb{E}_{\pi} \sum_{k=1}^{K} R(s_k, a_{k+1}, o_{k+1}), \tag{5}$$

This step-wise optimization enables the agent to learn both *when to ask* and *what to ask*, balancing clarification with progression toward the final goal. Through reinforcement signals, the agent learns to navigate the trade-off between proactive information gathering and efficient task completion, ultimately leading to more robust and goal-aligned dialogue behavior.

Importantly, we focus on clarification triggered by referential ambiguity or underspecification, where multiple valid slot values remain plausible despite partial information rather than traditional slot filling for missing required fields (Lison, 2013; Louvan and Magnini, 2020), and train agents to decide when uncertainty warrants clarification rather than simply requesting unfilled slots.

3.2 Structured Reasoning and Clarification Tokens

To enable the model to autonomously reason about ambiguity and generate clarification questions, we structure its outputs using two category of special tokens: **<think>**... **</think>** and **<clarify>**... **<clarify>**. The **<think>** tokens delimit the model's internal reasoning trace, allowing it to articulate latent uncertainty and evaluate whether the current user input provides sufficient information to act. **<clarify>** tokens, in turn, marks the model's externally verbalized clarification question aimed at resolving that uncertainty. This tokenization provides a clean separation between implicit reasoning and explicit interaction, enabling precise supervision and reward assignment during RL training.

When the model emits a segment within <think>. . . </think> the content is treated as an internal thought process and excluded from the dialogue context visible to the user. If the output contains <clarify>. . . <clarify>, the enclosed text is parsed as the model's clarification question and appended to the dialogue history, triggering a response from the user simulator. The returned feedback is then incorporated into the evolving dialogue state, forming a new step in the reasoning trajectory.

Importantly, <think> and <clarify> can co-occur within a single output, allowing the model to reason, identify uncertainty, and immediately issue a targeted clarification within the same turn. The user's initial goal or query Q is provided as the starting context, and subsequent user replies are iteratively appended to form a structured multi-turn trajectory:

$$s_k = (r_1, a_1, o_1), \ldots, (r_k, a_k, o_k), \tag{6}$$

where reasoning r_i corresponds to <think> content, and clarification or response a_i corresponds to user-directed actions (via <clarify> or plain responses).

This token-level design enables reinforcement signals to be applied at fine granularity, rewarding the model not merely for end-task success but for strategic *decision-making* in ambiguity detection and question formulation. Through this structured reasoning–clarification loop, SpeakRL teaches LLMs to proactively manage uncertainty and conduct effective multi-turn dialogue grounded in user intent.

3.3 Reward Design

Reward mechanisms play a central role in reinforcement learning with verifiable rewards (RLVR), guiding the model toward desirable interactive behavior (Guo et al., 2025). In our training, we similarly adopt a reward formulation that integrates structural and semantic-quality components (Jin et al., 2025; Qian et al., 2025), implicitly teaching the model when to ask for clarification via token-level optimization, and what to ask through semantic feedback. Formally, the total reward at each step is defined as:

$$R_{\text{total}} = R_{\text{format}} + R_{\text{clarify}}, \tag{7}$$

where (R_{format}) measures adherence to the required output structure and (R_{clarify}) assesses the quality and helpfulness of clarification questions.

Format Reward. The format reward (R_{format}) verifies whether the model correctly employs the designated special tokens <think> and <clarify> in the proper order and syntactic form. The reward encourages the model to produce interpretable reasoning traces and explicitly structured clarifications, where a fixed output format simplifies verification and guides the model toward more deliberate reasoning.

$$R_{\text{format}} = \begin{cases} 1, & \text{if tokens appear correctly in valid order,} \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Additionally, partial credit can be assigned when the model produces only one of the required fields (e.g., emits <think> but omits <clarify>), which empirically stabilizes early-stage learning. This binary (or near-binary) format supervision ensures the model first learns how to produce syntactically valid clarification outputs before optimizing their content quality.

Clarification Reward. The clarification reward R_{clarify} evaluates the *semantic quality* of the clarification question enclosed within <clarify>. . . <clarify>. Because there is no single "correct" clarification for a given ambiguous query, we adopt an LLM-as-judge scoring framework that assigns verifiable feedback based on several interpretable dimensions. At each clarification step, we query an LLM with a structured rubric prompt that evaluates complementary dimensions of clarification quality as defined in Section C. Specifically, Relevance measures whether the question directly targets the ambiguous or missing information in the user request; Precision and Clarity capture whether the question is well-formulated, unambiguous, and clearly phrased; Specificity assesses whether it narrows down the uncertainty to a concrete aspect of the task; Logical Connection evaluates whether the question follows coherently from the preceding dialogue context; and Constructive Nature examines whether the question helps advance task completion rather than repeating or restating information.

The combined reward is thus expressed as: $R_{\text{total}} = R_{\text{format}} + R_{\text{clarify}}$. Unlike prior works that rely solely on outcome-based or rule-matching rewards, our design jointly optimizes structural ad-

Given the **Conversation** below, carefully read the dialogue and the final user query. First, reflect on the reasoning process—consider any ambiguity, missing information, or potential failure points. Then decide whether it is necessary to ask the user a clarification question before proceeding. The reasoning process and user clarification question are enclosed within <**think**>…</**think**> and <**clarify**>…<**clarify**> tags, respectively, i.e., <**think**>reasoning process here </**think**> <**clarify**> user clarification question here <**clarify**>. User: conversation. Agent:

Table 1: **Template for SpeakRL.** The placeholder conversation is substituted with the corresponding user query and dialogue turn during training.

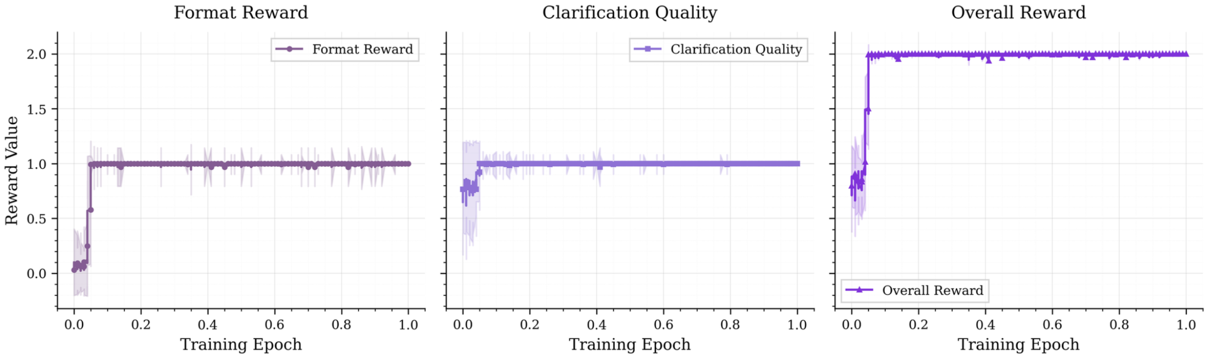


Figure 3: **RL Rewards**. SpeakRL reward progression across format, clarification quality, and overall.

herence and interaction quality. The format reward ensures syntactic precision in the model's reasoning–clarification structure, while the clarification reward provides dense, interpretable feedback on what the agent asks. This two-part signal allows SpeakRL to learn fine-grained clarification behaviors without relying on ground-truth task completion labels, ultimately leading to more adaptive, user-aware dialogue strategies.

3.4 SpeakER Dataset

We introduce the SpeakER dataset to study clarification behavior in task-oriented dialogue settings where user goals are *intrinsically ambiguous* and cannot be resolved without explicitly asking clarification questions. Unlike conventional slot-filling datasets (e.g., SLURP), where missing information can be obtained by sequentially filling required slots, SpeakER focuses on scenarios where the system has *partial but uncertain* information, and successful task completion depends on resolving ambiguity through clarification.

We build SpeakER by using MultiWOZ 2.4 (Ye et al., 2022) dialogues as seed trajectories and synthesizing new dialogue paths that intentionally introduce ambiguities at different stages of the interaction. These ambiguities may require one or multiple clarification turns to resolve. Clarification turns are explicitly annotated using special tokens <**clarify**>...<**clarify**>, enabling turn-level supervision of *when* clarification is necessary within a multi-turn context. This annotation allows us to condition learning on the full dialogue history while assigning rewards at specific clarification decision points, supporting single-step optimization with multi-turn conversational context. Importantly, SpeakER is not designed to optimize *what* clarification question to ask, but rather to supervise *whether* and *when* clarification is required; the former is studied separately in Section 3. All dialogues are filtered to remove redundant clarification questions using n-gram similarity, and only trajectories that successfully complete the task through clarification are retained. The final dataset consists of approximately 25K training dialogues.

For preference-based training, we additionally construct SpeakER-DPO. Positive samples correspond to successful clarification-based trajectories, while negative samples reuse the same user goals but omit clarification turns, leading to task failure. We use DPO (Rafailov et al., 2023) as a contrastive objective rather than human preference alignment, allowing the model to learn the consequences of asking—or failing to ask—clarification questions. All data are synthesized using `claude-sonnet-4-20250514`.

3.5 RL Training

To train the model with structured rewards, we adopt the GRPO algorithm (Shao et al., 2024b) by using the training instruction in Table 1 (See Figure 2). Unlike the original formulation, we remove the KL-divergence penalty against a reference model, allowing the policy to more freely adapt to our custom clarification format and reward structure. This design choice simplifies the training pipeline while maintaining stability and leading to faster convergence in practice. For the clarification-quality reward, we experiment with two LLM-as-judge settings: a strong external evaluator, `claude-sonnet-4-20250514`, providing high-fidelity feedback for objective scoring; and a self-judging setup, where the same model `Qwen2.5-7B-Instruct` evaluates its own clarification questions. The latter explores the potential of self-improving agents that refine their behavior through internally generated reward signals (Huang et al., 2023, 2025a; Acikgoz et al., 2025c).

During RL training, the model rapidly learns both structured and behavioral reward signals on SpeakER. As shown in Figure 3, the *Format Reward* starts low but quickly converges to a stable maximum, indicating that the model efficiently learns to follow the expected <**think**> and <**clarify**> output structure. The *Clarification Reward* (middle) begins at a moderately higher baseline and similarly converges early, suggesting that the agent quickly internalizes what constitutes an effective clarification. Together, these trends yield a stable *Overall Reward* (right), demonstrating consistent convergence and stable policy improvement throughout GRPO training.

4 Main Results

Collaborative Environment. To simulate realistic user environments, agents must engage in collaborative communication that handles real-world goal-oriented tasks. We simulate conversations between an agent and a human user-simulator (Xu et al., 2024) with access to user goals hidden from the agent. The agent must fulfill user requests that may span multiple subtasks (e.g., booking a hotel, finding an Italian restaurant, and reserving a table for 3 at 7pm), some containing ambiguities requiring clarification. The agent must interact with the user to gather necessary information and complete the task. Task completion occurs when the agent returns correct booking or reservation IDs, or terminates after a predefined turn limit. We evaluate the agent's performance using two key metrics: success rate and average number of turns where a lower value indicates better efficiency (See Section A for further details).

Method	Success (↑)	Turns (↓)
Qwen-2.5-7B-Instruct		
Prompting	25.63 ± 1.24	8.12
SFT	28.78 ± 1.15	7.32
DPO	45.73 ± 3.23	5.92
SpeakRL	**46.17 ± 1.25**	**5.82**
claude-sonnet-4.0	44.08 ± 1.99	6.28

Table 2: **Main results on collaborative user–agent dialogue evaluation.** Comparison of different training paradigms on task success (Success) and conversational efficiency (*Avg. Turns*). Higher success and fewer turns indicate better goal completion and dialogue quality.

Models. We use `Qwen2.5-7B-Instruct` as our main agent model because it is publicly available as open source, has been shown to be one of the best models for its size, and is generally used in RL finetuning. We use `claude-sonnet-4-20250514` for the user-simulator and LaaJ reward model unless otherwise specified.

Finding 1: Effective User Clarification Improves Task Success and Efficiency. Even though SpeakRL is not explicitly optimized for task success, it achieves substantial improvements over prompting. Specifically, Success (Avg@5) improves from 25.63 → 46.17, corresponding to an absolute gain of +20.54 points (80% relative). These gains indicate that effective clarification, asking when information is missing, directly enhances task completion rates. Moreover, SpeakRL reduces average turns from 8.12 → 5.82, a reduction of 2.30 turns (28%), demonstrating improved conversational efficiency. The agent learns to identify ambiguities early, ask a single targeted question, and obtain the necessary information in fewer exchanges, minimizing unnecessary dialogue cycles.

Finding 2: GRPO-Based Reinforcement Learning Outperforms Supervised and Preference-Based Methods. Among learning paradigms, GRPO-based SpeakRL achieves the strongest performance, outperforming both SFT and DPO. Compared to SFT, Success rises from 28.78 → 46.17, an absolute +17.39 gain (relative 60%). SFT overfits to dialogue trajectories–imitating structure without learning when or why to ask questions—whereas

SpeakRL learns through reward feedback. Against DPO, SpeakRL still achieves higher scores (45.73 → 46.17), showing the benefit of granular, token-level reward shaping via GRPO. These results highlight that structured RL-based reward learning can produce reasoning-capable and adaptive conversational agents beyond imitation or pairwise preference optimization.

Finding 3: Small Open Models Can Surpass Proprietary LLMs When Trained Collaboratively. Remarkably, SpeakRL fine-tuned on the open-source `Qwen2.5-7B-Instruct` (46.17%) surpasses much more larger proprietary model `claude-sonnet-4.0` (44.08%). Despite being significantly smaller, SpeakRL benefits from reinforcement-driven clarification training, enabling it to generalize beyond memorization and achieve competitive or superior task success. This finding underscores the promise of small, open, and interpretable conversational agents (Belcak et al., 2025), when trained with collaborative user-clarification feedback through RLVR, to rival and even outperform larger closed models.

5 Ablation Studies

We conduct detailed analyses to understand the internal components of SpeakRL and the underlying dynamics of the RLVR process.

Finding 4: Emergent Improvement in Latent Reasoning Depth During RLVR Training. As shown in Figure 4 (left), although the **<think>**...**</think>** token sequence is not explicitly rewarded for its length, we observe a gradual and consistent increase in the model's internal reasoning span over RLVR iterations. The average think-string length steadily rises throughout training, indicating that the model autonomously learns to engage in deeper reasoning before producing actions or clarifications. This emergent behavior reveals that GRPO not only optimizes for external task success but also implicitly fosters the development of richer internal deliberation, leading to improved reasoning quality and more stable decision-making over time.

Finding 5: Learning What to Ask Leads to Richer and More Effective Clarifications. As shown in Figure 4 (right), the model gradually learns *what to ask*, how to identify and query missing information critical for task completion. Early in training (first 100–200 samples), clarification questions are short and underspecified (around 10 tokens), often failing to resolve ambiguity. Over time, their average length increases steadily, indicating that the agent begins forming more complete and contextually grounded questions. This evolution demonstrates that reinforcement learning not only improves the model's ability to act but also shapes its inquiry behavior, enabling it to formulate richer, more purposeful clarifications that directly enhance task success and collaborative efficiency.

5.1 Qualitative Analysis

As shown in Figure 5, our qualitative analysis uncovers an emergent pattern of reflective reasoning within the **<think>**...**</think>** segments, revealing how the model progressively internalizes the principles of context-aware clarification through RL. Early in training (Epoch 0.1–0.3), the agent's reasoning remains superficial; its thought process ends prematurely with conclusions such as "The conversation has all the necessary information... No further clarification is needed". It incorrectly concludes that no clarification is needed, even though additional information is clearly required. By contrast, in later stages (Epoch 0.8–1.0), the model's internal reasoning exhibits a more structured and anticipatory nature. It begins to self-monitor and generate meta-cognitive statements such as "I need to ensure I have the correct context and any necessary details" and "I should confirm the user's current travel plans and ensure everything is clear to avoid any mix-ups". These phrases indicate that the model is learning to (1) assess the sufficiency of information, (2) reason about latent variables like time, intent, and user preferences, and (3) plan clarification queries that minimize ambiguity before acting.

6 Discussion

Conclusions In this work, we presented SpeakRL, an end-to-end RLVR framework that enables LLM agents to proactively ask effective clarification questions in multi-turn, goal-oriented dialogues. To do that we create SpeakER, a synthetic dataset of 25K conversations explicitly designed to capture ambiguous scenarios through turn-level clarification annotations. By separating reasoning and clarification using structured tokens and train with GRPO-based RLVR, SpeakRL jointly learns when and what to clarify without directly optimizing for task completion. Empir-

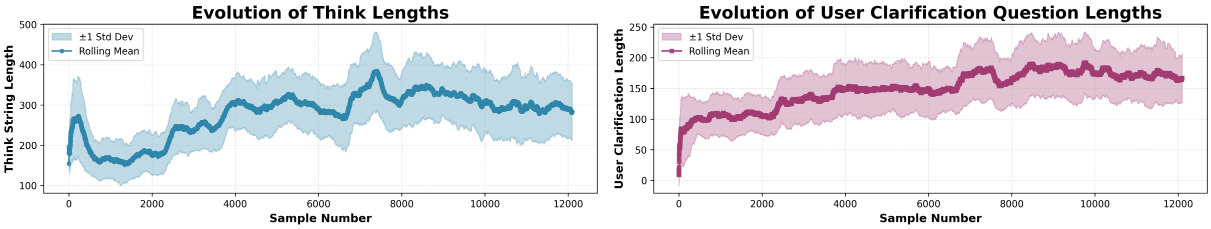
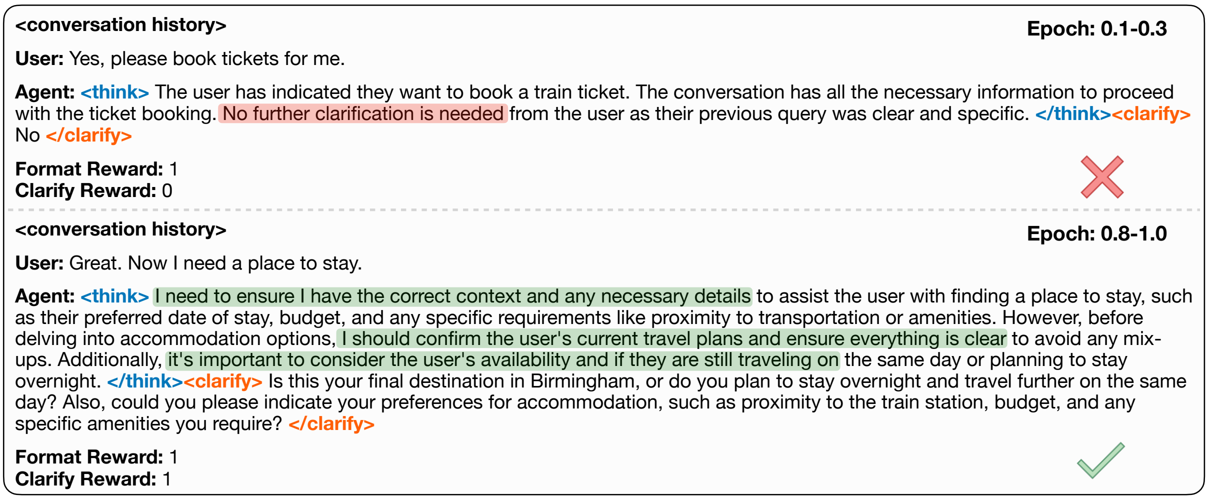


Figure 4: **Evolution of reasoning and clarification behaviors during GRPO training.** The left plot tracks the growth of <**think**> sequence lengths, reflecting deeper internal reasoning, while the right plot shows increasingly rich <**clarify**> questions, indicating the model's improved ability to identify and resolve missing information.



<conversation history> **Epoch: 0.1-0.3**

User: Yes, please book tickets for me.

Agent: **<think>** The user has indicated they want to book a train ticket. The conversation has all the necessary information to proceed with the ticket booking. No further clarification is needed from the user as their previous query was clear and specific. **</think><clarify>** No **</clarify>**

Format Reward: 1
Clarify Reward: 0 ✗

<conversation history> **Epoch: 0.8-1.0**

User: Great. Now I need a place to stay.

Agent: **<think>** I need to ensure I have the correct context and any necessary details to assist the user with finding a place to stay, such as their preferred date of stay, budget, and any specific requirements like proximity to transportation or amenities. However, before delving into accommodation options, I should confirm the user's current travel plans and ensure everything is clear to avoid any mix-ups. Additionally, it's important to consider the user's availability and if they are still traveling on the same day or planning to stay overnight. **</think><clarify>** Is this your final destination in Birmingham, or do you plan to stay overnight and travel further on the same day? Also, could you please indicate your preferences for accommodation, such as proximity to the train station, budget, and any specific amenities you require? **</clarify>**

Format Reward: 1
Clarify Reward: 1 ✓

Figure 5: **Qualitative Analysis of Thinking and Clarification.** Early in training (Epoch 0.1–0.3), the agent prematurely concludes that no clarification is needed, overlooking missing context. Later (Epoch 0.8–1.0), it demonstrates careful thinking, confirming user goals, and generating clear targeted clarification questions.

ical results demonstrate that post-training with SpeakRL leads to higher task success and shorter dialogues, resulting in more accurate, efficient, and collaborative human–agent interactions.

Limitations While SpeakRL shows promise for co-evolving user–agent interactions, it has several limitations. First, both training and evaluation rely on the training split of MultiWOZ 2.4 due to the lack of suitable task-oriented user simulators, which may introduce i.i.d. bias and limit generalization. Second, our reward design does not explicitly penalize excessive or unnecessary clarification questions. In different settings, this could encourage reward hacking, leading the agent to ask overly long or repetitive questions, potentially reducing user satisfaction in real-world deployments (Levandovsky et al., 2025). Addressing this trade-off between clarification utility and user burden is an important direction for future work.

Future Work Looking ahead, future directions include developing multi-task reward functions that jointly optimize for clarification, task execution, and response quality by extending RLVR beyond clarification to broader collaborative reasoning. Another promising direction is teaching tool-use (Qian et al., 2025) and clarification skills with RLVR in multi-turn conversations (Acikgoz et al., 2025a) in dynamic environments. Finally, self-improving LLM agents represent a promising and largely underexplored direction (Schmidhuber, 2007), especially for TOD Agents. Future work can focus on enabling agents to proactively self-improve their skills at test time (Acikgoz et al., 2025c), allowing them to adapt to new situations and better align with human preferences on the fly (Carroll et al., 2024). Beyond purely autonomous agents, an even safer and more compelling direction is the co-evolution of agents together with humans, where continual mutual adaptation enables more reliable, aligned, and effective AI systems (Weston and Foerster, 2025). Together, these efforts move toward a unified objective: building interactive conversational agents capable of reasoning, clarifying, and acting toward perfect collaboration.

References

Emre Can Acikgoz, Jeremiah Greer, Akul Datta, Ze Yang, William Zeng, Oussama Elachqar, Emmanouil Koukoumidis, Dilek Hakkani-Tür, and Gokhan Tur. 2025a. Can a single model master both multi-turn conversations and tool use? CoALM: A unified conversational agentic language model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12370–12390, Vienna, Austria. Association for Computational Linguistics.

Emre Can Acikgoz, Dilek Hakkani-Tür, and Gokhan Tur. 2025b. Conversational agents in the era of large language models [perspectives]. *IEEE Signal Processing Magazine*, 42(3):35–39.

Emre Can Acikgoz, Cheng Qian, Heng Ji, Dilek Hakkani-Tür, and Gokhan Tur. 2025c. Self-improving llm agents at test-time. *arXiv preprint arXiv:2510.07841*.

Emre Can Acikgoz, Cheng Qian, Hongru Wang, Vardhan Dongre, Xiusi Chen, Heng Ji, Dilek Hakkani-Tür, and Gokhan Tur. 2025d. A desideratum for conversational agents: Capabilities, challenges, and future directions. *arXiv preprint arXiv:2504.16939*.

Chinmaya Andukuri, Jan-Philipp Fränken, Tobias Gerstenberg, and Noah Goodman. 2024. STar-GATE: Teaching language models to ask clarifying questions. In *First Conference on Language Modeling*.

Peter Belcak, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. 2025. Small language models are the future of agentic ai. *arXiv preprint arXiv:2506.02153*.

Micah Carroll, Davis Foote, Anand Siththaranjan, Stuart Russell, and Anca Dragan. 2024. Ai alignment with changing and influenceable reward functions. In *International Conference on Machine Learning*, pages 5706–5756. PMLR.

Maximillian Chen, Ruoxi Sun, Tomas Pfister, and Sercan O Arik. 2025. Learning to clarify: Multi-turn conversations with action-based contrastive self-training. In *The Thirteenth International Conference on Learning Representations*.

Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng. 2017. Towards end-to-end reinforcement learning of dialogue agents for information access. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–495, Vancouver, Canada. Association for Computational Linguistics.

Vardhan Dongre, Xiaocheng Yang, Emre Can Acikgoz, Suvodip Dey, Gokhan Tur, and Dilek Hakkani-Tür. 2024. Respact: Harmonizing reasoning, speaking, and acting towards building large language model-based conversational ai agents. *arXiv preprint arXiv:2411.00927*.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Anna Gottardi, Osman Ipek, Giuseppe Castellucci, Shui Hu, Lavina Vaz, Yao Lu, Anju Khatri, Anjali Chadha, Desheng Zhang, Sattvik Sahai, et al. 2022. Alexa, let's work together: Introducing the first alexa prize taskbot challenge on conversational task assistance. *arXiv preprint arXiv:2209.06321*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. ORPO: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11170–11189, Miami, Florida, USA. Association for Computational Linguistics.

Audrey Huang, Adam Block, Dylan J Foster, Dhruv Rohatgi, Cyril Zhang, Max Simchowitz, Jordan T. Ash, and Akshay Krishnamurthy. 2025a. Self-improvement in language models: The sharpening mechanism. In *The Thirteenth International Conference on Learning Representations*.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. Large language models can self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1051–1068, Singapore. Association for Computational Linguistics.

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. 2025b. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*.

Kasia Kobalczyk, Nicolás Astorga, Tennison Liu, and Mihaela van der Schaar. 2025. Active task disambiguation with LLMs. In *The Thirteenth International Conference on Learning Representations*.

Yuxiang Lai, Jike Zhong, Ming Li, Shitian Zhao, and Xiaofeng Yang. 2025. Med-r1: Reinforcement learning for generalizable medical reasoning in vision-language models. *arXiv preprint arXiv:2503.13939*.

Enoch Levandovsky, Anna Manaseryan, and Casey Kennington. 2025. Learning to speak like a child: Reinforcing and evaluating a child-level generative language model. In *Proceedings of the 26th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 370–382, Avignon, France. Association for Computational Linguistics.

Belinda Z. Li, Alex Tamkin, Noah Goodman, and Jacob Andreas. 2025. Eliciting human preferences with language models. In *The Thirteenth International Conference on Learning Representations*.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Austin, Texas. Association for Computational Linguistics.

Pierre Lison. 2013. Model-based bayesian reinforcement learning for dialogue management. In *Proc. Interspeech 2013*, pages 475–479.

Samuel Louvan and Bernardo Magnini. 2020. Recent neural methods on slot filling and intent classification for task-oriented dialogue systems: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 480–496, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. In *Advances in Neural Information Processing Systems*, volume 37, pages 124198–124235. Curran Associates, Inc.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. 2022. Teach: Task-driven embodied agents that chat. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2017–2025.

Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. 2024. Iterative reasoning preference optimization. In *Advances in Neural Information Processing Systems*, volume 37, pages 116617–116637. Curran Associates, Inc.

Alexandros Papangelis, Yi-Chia Wang, Piero Molino, and Gokhan Tur. 2019. Collaborative multi-agent dialogue model training via reinforcement learning. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 92–102, Stockholm, Sweden. Association for Computational Linguistics.

Matthew Purver, Jonathan Ginzburg, and Patrick Healey. 2001. On the means for clarification in dialogue. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. 2025. Toolrl: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

Jürgen Schmidhuber. 2007. *Gödel Machines: Fully Self-referential Optimal Universal Self-improvers*, pages 199–226. Springer Berlin Heidelberg, Berlin, Heidelberg.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Pararth Shah, Dilek Hakkani-Tür, Bing Liu, and Gokhan Tür. 2018. Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning. In *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.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024a. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, et al. 2024b. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA.

Jason Weston and Jakob Foerster. 2025. Ai & human co-improvement for safer co-superintelligence. *arXiv preprint arXiv:2512.05356*.

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

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. 2024. Travelplanner: A benchmark for real-world planning with language agents. In *International Conference on Machine Learning*, pages 54590–54613. PMLR.

Heng-Da Xu, Xian-Ling Mao, Puhai Yang, Fanshu Sun, and Heyan Huang. 2024. Rethinking task-oriented dialogue systems: From complex modularity to zero-shot autonomous agent. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2748–2763, Bangkok, Thailand. Association for Computational Linguistics.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. tau-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*.

Fanghua Ye, Jarana Manotumruksa, and Emine Yilmaz. 2022. MultiWOZ 2.4: A multi-domain task-oriented dialogue dataset with essential annotation corrections to improve state tracking evaluation. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 351–360, Edinburgh, UK. Association for Computational Linguistics.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, pages 15476–15488. Curran Associates, Inc.

Michael JQ Zhang and Eunsol Choi. 2025. Clarify when necessary: Resolving ambiguity through interaction with LMs. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5526–5543, Albuquerque, New Mexico. Association for Computational Linguistics.

Michael JQ Zhang, W. Bradley Knox, and Eunsol Choi. 2025. Modeling future conversation turns to teach LLMs to ask clarifying questions. In *The Thirteenth International Conference on Learning Representations*.

Tiancheng Zhao, Kaige Xie, and Maxine Eskenazi. 2019. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1208–1218, Minneapolis, Minnesota. Association for Computational Linguistics.

Appendix

A Collaborative Environment

Overview. To simulate realistic user environments, agents must engage in collaborative communication that handles real-world goal-oriented tasks, where a single user request may encompass several tasks from different domains with varying levels of complexity. In SpeakRL, agents communicate with users in a realistic end-to-end manner, where an agent can directly respond with natural language, take actions via APIs by interacting with external databases, or ask clarification questions.

Task Generation. We generate tasks using the user goals G from MultiWOZ 2.4 (Ye et al., 2022), which provides ground truth user goals as annotations. Our environment includes five different domains: restaurant, hotel, train, attraction, and taxi. The agent must track user multi-intent goals, monitor the evolving belief state, make API calls when necessary, ask clarification questions in cases of ambiguity or underspecification to advance the task, and provide appropriate system responses (see Table 4 for further details about environment).

Collaborative Conversation. We simulate conversations between an agent and a human user-simulator (Xu et al., 2024), which has access to user goals unknown to the agent. The agent's task is to fulfill the user request, which may involve several different subtasks (e.g., booking a hotel, searching for an Italian restaurant afterward, and reserving a table for 3 persons at 7pm), some of which may include ambiguities requiring user clarification. The agent should interact with the user, gather all necessary information, and complete the task. The task is considered complete when the agent returns the correct booking or reservation IDs, or terminated after specific number of turns.

B Further Details on MultiWOZ 2.4

We evaluate the performance of our SpeakRL using dialogue-level metrics that capture both the effectiveness and efficiency of task completion. Our primary metric is Success Rate, which measures whether the agent fully satisfies all user-specified constraints and successfully completes the task. For each dialogue, we use an LLM-based judge to assess if the agent's final response fulfills every requirement defined by the user's goal, including both requested attributes (such as hotel name or train arrival time) and booking constraints (such as the number of people or destination) following Xu et al. (2024). Formally, a dialogue is considered successful if all constraints in the user's goal G are met by the end of the interaction: Success $= \mathbb{I}(\text{all constraints in } G \text{ are satisfied})$, where $\mathbb{I}(\cdot)$ denotes the indicator function. This score is computed for every dialogue and averaged across the evaluation set. To account for the stochastic nature of both model inference and LLM-based judging, we conduct five independent runs for each experimental configuration.

We report two aggregate Success Rate metrics: **Success Avg@5**, the mean and standard deviation of success rates over the five runs, providing a robust measure of typical performance and variance, and **Average Number of Turns** per conversation as an efficiency metric. This measures the average length of the dialogue required to complete the task, with lower values indicating more concise and effective interactions.

C RLVR Training Details

We conduct our experiments using the TRL framework[1] with the GRPO class. We adopt the training prompt template shown in Table 1 and report the GRPO hyperparameter settings in Table 3 to ensure reproducibility. The LaaJ prompt template used by the reward model (illustrated in Figure 2) is provided in Figure 6.

Hyperparameter	Value
Base Model	Qwen/Qwen2.5-7B-Instruct
Dataset	SpeakER 25K
Epochs	1
Batch Size (per device)	8
Gradient Accumulation Steps	8
Effective Batch Size	512
Learning Rate	1×10^{-5}
LR Scheduler	Cosine
Warmup Ratio	0.1
Optimizer	AdamW
Adam β_1 and β_2	0.9, 0.99
Weight Decay	0.1
Max Gradient Norm	0.1
GRPO β	0.04
Number of Generations (K)	8
Max Prompt Length	512
Max Completion Length	786
Precision	BF16
GPUs	8 × A100s

Table 3: GRPO training hyperparameter details.

[1] https://github.com/huggingface/trl

Domain	API Name	API Arguments	Test Samples per Domain
Restaurant	query_restaurant book_restaurant	area, pricerange, food, name name, people, day, time, pricerange, stars, type	437
Hotel	query_hotel book_hotel	area, internet, name, parking name, people, day, stay	394
Attraction	query_attraction	area, name, type	395
Train	query_train buy_train_ticket	arriveBy, day, departure, destination, leaveAt, trainID arriveBy, day, departure, destination, leaveAt, trainID, people	494
Taxi	book_taxi	arriveBy, departure, destination, leaveAt	195

Table 4: Environment details and available function calls.

LLM Judge Prompt for Quality Reward

You are a judge evaluating the quality of user clarification questions. Given a conversation agent clarification question, analyze if there are any clarification questions and evaluate their quality.
Rules:

1. If clarification questions exist, evaluate them based on:
   - Relevance to the context
   - Precision and clarity
   - Specificity
   - Logical connection to previous context
   - Constructive nature of the question
2. If no clarification questions exist, output: `0`
3. Output format:
   - For high-quality clarification questions: `1`
   - For low-quality or no clarification questions: `0`

IMPORTANT: You must *only* output the number `0` or `1`. No other text, explanations, or characters are allowed. Do not provide any reasoning. Return only an integer score in the following exact format:

Score: [YOUR BINARY 0/1 SCORE HERE]

Conversation
<conversation>

Agent Clarification Question to Judge
<clarification_question>

Your Decision (0/1)
Score: [0 or 1]

Figure 6: LLM Judge prompt used for binary quality reward evaluation.