LLM-Driven Multi-Turn Task-Oriented Dialogue Synthesis for Realistic Reasoning

Yu Zhu, Kai Yang

Department of Computer Science and Technology, Tongji University

Abstract

The reasoning capability of large language models (LLMs), defined as their ability to analyze, infer, and make decisions based on input information, is essential for building intelligent task-oriented dialogue systems. However, existing benchmarks do not sufficiently reflect the complexity of real-world scenarios, which limits their effectiveness in evaluating and enhancing LLM reasoning in practical contexts. Many current reasoning datasets are overly simplistic and abstract, often disconnected from realistic task flows, domain constraints, and operational rules, making it difficult to effectively evaluate LLMs' logical reasoning ability. In addition, data contamination from pretraining corpora undermines the reliability of evaluation results, and traditional crowdsourcing methods for dataset construction are labor-intensive and difficult to scale. To address these challenges, we propose a LLM-driven framework for synthesizing multi-turn, task-oriented dialogues grounded in realistic reasoning scenarios, leveraging trilevel optimization to enhance dialogue quality. Our method generates dialogues grounded in authentic task scenarios, enriched with real-world information, and exhibiting strong contextual coherence. Corresponding reasoning tasks are carefully designed around these dialogues and iteratively refined to continuously improve the tasks' quality and challenge. The resulting dataset serves as a valuable benchmark for assessing and advancing the realistic logical reasoning capabilities of LLMs. Experimental results show that our synthetic data–based reasoning tasks introduce non-trivial reasoning challenges and provide meaningful support for improving the reasoning capabilities of LLMs.

Datasets —
https://github.com/adventureoflingke/RealReasoning/

1 Introduction

With the rapid development of large language models, natural language processing has entered a new era(Achiam et al. 2023; Ouyang et al. 2022; Wei et al. 2022). Large-scale pretrained language models, empowered by transformer architectures and massive parameter sizes, have achieved remarkable performance across a wide range of NLP tasks, including text classification, machine translation, question answering, and summarization(Gao et al. 2025; Kernycky et al. 2024; Li et al. 2024a). These models often achieve near-human or even super-human performance on benchmark datasets, significantly pushing the boundaries of what was previously thought possible(Chang et al. 2024). However, despite these impressive achievements, when faced with logical reasoning tasks in realistic scenarios, their performance remains unsatisfactory(Li et al. 2024b). Therefore, to enhance the realistic logical reasoning capabilities of LLMs, acquiring high-quality data grounded in realistic environments has emerged as a critical direction for future research.

Reasoning tasks serve as a key means to evaluate a model's reasoning capabilities, and the quality of the datasets used in these tasks has a decisive impact on research outcomes(Lu et al. 2023; Yu et al. 2024). As models grow increasingly sophisticated(Huang and Chang 2023), the need for datasets that accurately measure reasoning capability becomes more pressing. Although multiple datasets exist for evaluating reasoning ability(Chen et al. 2022; Zhang et al. 2021), they suffer from notable limitations. First, the tasks used in these datasets are often relatively simple and abstract, present reasoning tasks in a highly intuitive manner, rarely accounting for factors appear in realistic scenarios. Logical reasoning problems that involve realistic scenarios, integrate multi-source information, and are closely tied to business rules are seldom covered, such problems typically require models to maintain coherence across long sequences, and apply context-sensitive constraints. Apart from it, realistic scenarios often involve privacy concerns for both individuals and organizations(Yao et al. 2024). The commercial value associated with real data further complicates its collection and public release, leading to an increasing scarcity of such data in public. Moreover, existing evaluation benchmarks suffer from data contamination due to the overlap between pretraining data and test sets, undermining the reliability of performance assessments(Deng et al. 2024; Magar and Schwartz 2022). Traditional approaches mainly rely on crowd-sourcing for dataset construction, and thus are inherently limited in addressing such issues. Existing works have also explored agent-based approaches for constructing dialogue datasets, which rely on pre-specified structured task information to generate dialogues(Rastogi et al. 2020; Shah et al. 2018). While these datasets provide important benchmarks for evaluating model capabilities in information extraction and task completion, their generation processes are highly dependent on predefined schemas and flows, thus limiting their ability to reflect the complex rea-

soning exhibited in real-world scenarios. Moreover, in the evaluation of synthetic dialogue quality, the process of exploring and iteratively refining evaluation metrics is often time-consuming and resource-intensive, and there remains a lack of efficient and high-quality evaluation methodologies. Therefore, developing methods to construct logical reasoning datasets grounded in real-world scenarios in a more efficient manner plays a crucial role in advancing the development and application of reasoning models in practical settings.

To address these challenges, we propose a novel approach that leverages large language model agents to simulate user behaviors based on realistic scenarios. Our goal is to overcome the limitations of traditional dataset construction methods, such as lack of realism and data contamination, by generating high-fidelity interactions in a controlled and scalable manner. Our method generates action sequences that closely resemble realistic interactions. These sequences are constructed by modeling the decision-making patterns of users during realistic scenarios, which ensures that the user agent possesses unique and realistic action memory aligned with realistic scenarios. Based on these sequences, user agents and assistant agents engage in dialogue, producing multi-turn task-oriented dialogues grounded in authentic real-world scenarios by prompting the model. Furthermore, we propose a trilevel optimization framework that jointly optimizes the evaluation metric, multi-turn dialogue prompt, and single-turn dialogue prompt, enabling automatic evaluation and iterative refinement of synthesized dialogue quality. This framework establishes an end-to-end pipeline from dialogue generation to evaluation, effectively supporting the production of high-quality synthetic dialogues. The contributions of this paper can be summarized as follows.

- We formalize dialogue generation as a trilevel optimization problem that jointly optimizes prompt design and evaluation criteria, enabling the automatic generation and iterative refinement of high-quality synthetic dialogues. By leveraging the generative and role-playing capabilities of LLM agents, our approach constructs task-oriented, logical reasoning-intensive dialogue data grounded in realistic information and aligned with scenario-specific rules.
- We present a multi-turn task-oriented dialogue dataset specifically designed to evaluate models' logical reasoning capabilities in realistic scenarios. Each instance in the dataset consists of a reasoning question derived from a realistic multi-turn dialogue context, accompanied by a corresponding label that has been manually annotated and verified. While such task-oriented datasets remain scarce, they play a pivotal role in advancing the development and deployment of reasoning models in practical settings, thereby offering significant value for research.
- Our experiments are evaluated on multiple baseline models and the results show that there is still significant room for improvement in current models when handling such reasoning tasks. This indicates that our dataset not only provides a challenging evaluation platform for existing models but also serves as a valuable resource for future research. Therefore, the proposed method has the potential to drive advancements in the reasoning capabilities of LLMs.

2 Related work

2.1 LLM-based agents

Due to their impressive planning and reasoning capabilities, large language models (LLMs) are increasingly being used as the foundation for constructing intelligent agents(Kojima et al. 2022; Li et al. 2023b; Nguyen et al. 2023; Xi et al. 2025). Specifically, LLMs serve as the central controller of these agents, enabling them to perceive external information and take actions accordingly(Shinn et al. 2023; Yao et al. 2023). LLM-based agents possess strong interaction capabilities with their environments. Moreover, thanks to their pre-training on massive corpora, LLMs have accumulated extensive knowledge across diverse domains, allowing them to generalize and transfer across tasks without task-specific fine-tuning(Chung et al. 2023; Jiao, Wang, and Yang 2025; Wang et al. 2020).

With recent advances in related research, multi-agent systems built upon LLMs have achieved notable success in solving complex problems and simulating realistic scenarios(Guo et al. 2024). LLM-based agents can communicate with each other, enabling both cooperation and competition(He et al. 2023; Hong et al. 2024; Islam, Ali, and Parvez 2024; Liu et al. 2025a; Qian et al. 2024; Zhang et al. 2024). By specializing agents in terms of roles and skills, LLM-based multi-agent systems can be assigned distinct capabilities and responsibilities, thereby enhancing performance on specific tasks through division of labor. The collaboration among agents with complementary roles leads to superior overall system performance, demonstrating the potential of LLM-based multi-agent architectures in modeling complex, realistic behaviors(Owoicho et al. 2023; Park et al. 2023). Based on existing research findings, this paper leverages the division of labor and collaboration among multiple agents to enhance the simulation of real-world scenarios. By assigning a specific agent to focus on a single task, the quality of the generated data is further improved.

2.2 Multi-trun dialogue data generation

Training machines to understand natural language and interact with humans is a challenging yet fundamental task in artificial intelligence. With the rapid advancement of deep learning techniques, especially the widespread adoption of pre-trained language models (PLMs), various dialogue systems have been proposed that are increasingly capable of engaging in multi-turn conversations(Algherairy and Ahmed 2024; Ham et al. 2020; Kulhánek et al. 2021; Lin et al. 2020; Xu, Zhao, and Zhang 2021). Compared to traditional text-based reading comprehension tasks, multi-turn dialogues are more closely aligned with spoken language and exhibit strong interactive characteristics. These dialogues typically involve multiple speakers, diverse intents, and frequent topic shifts, resulting in utterances that are highly dynamic and context-dependent(Huang, Zhu, and Gao 2020; Ma, Zhang, and Zhao 2023; Su et al. 2019). Performing reasoning tasks

in such settings introduces significantly greater complexity than understanding static, single-paragraph texts.

Traditional dialogue datasets are primarily collected and constructed through crowd-sourcing(Budzianowski et al. 2018; Feng et al. 2023; Quan et al. 2020; Serban et al. 2018; Zhu et al. 2020) . However, this approach is costly and has limited scalability. To address these challenges, synthetic data generation methods have gained development to some extent(Anaby-Tavor et al. 2020; Ding et al. 2020; Meng et al. 2022; Ye et al. 2022). In particular, for dialogue-formatted data, common approaches include enhancing existing datasets(Chen and Yang 2021; Li et al. 2023a) or directly generating data based on specific tasks, often leveraging external knowledge to produce multi-turn conversational structures(Abdullin et al. 2023; Kim et al. 2022; Qiu et al. 2024). Building upon existing research, this paper focuses on generating multi-turn task-oriented dialogue, aiming to embed necessary reasoning information within the dialogue structure, thereby enriching the complexity and practical value of the generated dialogues.

2.3 Trilevel optimization

Trilevel optimization has emerged as a powerful framework for modeling complex learning problems in which objectives and constraints are interdependent across multiple layers of control(Jiao, Yang, and Song 2025; Jiao et al. 2023; Yang et al. 2008), and it has already found numerous applications in machine learning. When gradient information is available, researchers typically adopt first-order gradient-based methods; this setting has been extensively studied, with applications including neural architecture search(Guo et al. 2020; Jiao et al. 2022), hyperparameter optimization(Raghu et al. 2021; Sato, Tanaka, and Takeda 2021), data reweighting(Garg et al. 2022), OOD generalization(Jian, Yang, and Jiao 2024) and distributed optimization(Jiao et al. 2024; Liu et al. 2025b). In scenarios where gradients are unavailable, zeroth-order information is used to solve trilevel learning problems, and a limited body of work has recently emerged in this direction(Jiao, Yang, and Jian 2025). In this paper, we make the first attempt to introduce a zeroth-order trilevel optimization framework to dialogue generation, and we specifically tailor the optimization to the structure of the loss function to more accurately evaluate dialogue quality .

3 Method

3.1 Problem Definition

This paper presents a method aimed at generating multi-turn, task-oriented dialogues to evaluate reasoning capabilities in realistic scenarios. In this setup, a user agent interacts with a assistant agent through dialogue to accomplish specific tasks in practical domains. Specifically, we denote x as a particular realistic scenario, $\mathcal{F}, \mathcal{G}$ as the user agent and the assistant agent, respectively. The resulting multi-turn dialogue is represented as $\boldsymbol{Y} = \{\boldsymbol{y}_1^{\mathcal{F}}, \boldsymbol{y}_1^{\mathcal{G}}, ..., \boldsymbol{y}_i^{\mathcal{F}}, \boldsymbol{y}_i^{\mathcal{G}}, ..., \boldsymbol{y}_n^{\mathcal{F}}, \boldsymbol{y}_n^{\mathcal{G}}\}$, where n denotes the number of dialogue rounds and $\boldsymbol{y}_i^{\mathcal{F}}, \boldsymbol{y}_i^{\mathcal{G}}$ denote the utterances generated by the user agent and the assistant agent in the i-th turn, respectively.

3.2 Framework overview

To evaluate the reasoning capabilities of LLMs, this study aims to simulate multi-turn task-oriented dialogues in realistic scenarios containing implicit information that can only be uncovered through logical reasoning. We propose a framework for generating multi-turn task-oriented dialogues driven by LLM-based agents. Given a realistic scenario, our framework inputs its description into an LLM-based user generator to get user agent, user agent observe the environment and perform actions. After each action, the user agent records its experience, and this process repeats to generate a sequence of actions. Once a complete action sequence is formed, the user agent engages in a turn-by-turn dialogue with an assistant agent. The user agent asks questions based on the action sequence and scenario context, and the assistant agent reasons about these questions using externally perceived information and provides responses. The generation process is shown in the figure 1.

After dialogue generation, we automatically evaluate and improve dialogue quality within a trilevel optimization framework by automatically searching for more accurate evaluation metrics and refining the dialogue prompts of the agents, thereby continuously enhancing the quality of the generated dialogues.

3.3 User generator

Generally, different users will have different roles, goals and problems to solve when faced with a particular realistic scenario. In the first step, we employ a LLM, referred to as the user generator, dedicated to generating users' intents and potential issues by prompting it. After generating the user's intent and problem, the created user profile is placed into a pool of candidate users, from which they can be selected for subsequent processes. By constructing various user roles, intents and problems in advance, user agents can focus on taking actions without having to generate their own intents and related content, thereby enhancing the performance of the user agents. The generation process of the condidate user by the user generator is formulated as Eq. (1):

$$\boldsymbol{U} = \mathcal{H}(x, p_1), \tag{1}$$

where p_1 is the prompt that instructs the LLM to act as the user generator, $\mathcal{H}$ represents the LLM used to generate candidate users, $\boldsymbol{U} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_i, ..., \boldsymbol{u}_m\}$ represents the generated candidate users, $\boldsymbol{u}_i$ represents the i-th candidate user and m represents the number of candidate users.

3.4 Simulated actions of the user agent

After the generation of candidate users, a user is selected from the candidate pool and the user agent will act as the selected user, adopting its role, intent, problems. The user agent takes actions based on the observed external information and its existing memory. Initially, the user's memory consists of its intent and associated problems. External information is collected using tools such as search engines, with keywords derived from the input realistic scenario, and is organized into documents to influence the user's action.

Following each action round, the user reviews its action and integrates it with its existing memory to form an updated

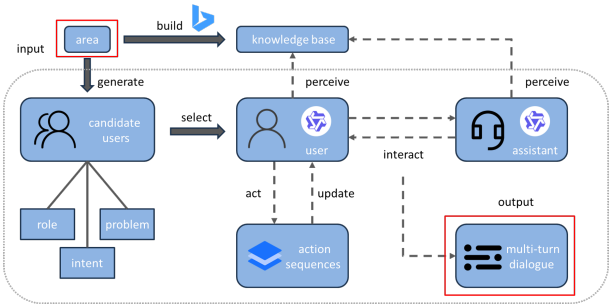


Figure 1: Schematic of the synthetic generation process.

memory state. By iteratively performing this process, a complete sequence of actions is generated, which simulates the action trajectory a real user might follow in a specific realistic scenario. The process of generating action sequence is formulated as Eq. (2):

$$\boldsymbol{a_i} = \mathcal{F}(x, \boldsymbol{A}_{i-1}, \boldsymbol{K}, p_2), \tag{2}$$

where $\boldsymbol{A}_{i-1} = \{a_1, a_2, ..., a_{i-1}\}$ represents the action sequence consisting of the first $i-1$ actions, $\boldsymbol{K}$ represents the external information collected by tools based on the input realistic scenario, p_2 represents the prompt used to control the user agent to take actions and $\boldsymbol{a_i}$ represents the action taken by the user agent in i-th action round.

By constructing a sequence of user actions, the user agent acquires stable and well-defined memory content. This enables the agent to exhibit consistent dialogue topics and maintain coherent intent throughout its interaction with the assistant agent, thereby enhancing the task-oriented nature of the dialogue. Through the incorporation of conditional information stored in memory, the resulting dialogues naturally embed implicit logical structures that require logical reasoning to uncover. Based on these inferred conclusions, we further derive corresponding questions and labels, which can be used for downstream evaluation tasks to evaluate the models' capability to perform realistic reasoning.

3.5 Interaction between the user agent and the assistant agent

Providing a fixed prompt into a large language model often results in outputs with limited diversity. The user agent engages in a dialogue with the assistant agent using memories drawn from realistic scenarios after generating its action sequence. This approach ensures that each user agent has distinct experiences, thereby generating more diverse responses during interactions with the assistant agent. Meanwhile, real-world data collected by external tools is incorporated as contextual information, further improving both the diversity and authenticity of the dialogue content. By defining the user agent's role, intent, and problem context, generating an action sequence, and providing prompt guidance during the interaction, task-oriented information is effectively integrated into the dialogue. This facilitates the generation of multi-turn task-oriented dialogues that support realistic reasoning tasks. The interaction process between the user agent and the assistant agent can be represented by Eq. (3) and Eq. (4):

$$\boldsymbol{y}_i^{\mathcal{F}} = \mathcal{F}(x, \boldsymbol{A}, \boldsymbol{Y}_{i-1}, \boldsymbol{K}, p_3), \tag{3}$$

$$\boldsymbol{y}_i^{\mathcal{G}} = \mathcal{G}(x, \boldsymbol{A}, \boldsymbol{Y}_{i-1}, \boldsymbol{y}_i^{\mathcal{F}}, \boldsymbol{K}, p_4), \tag{4}$$

where $\boldsymbol{A}$ represents the complete action sequence generated by the user agent, , $\boldsymbol{Y}_{i-1}$ represents the dialogue history up to the $i-1$-th turn, p_3 and p_4 represent the prompt used to control the user agent and the assistant agent to interact with each other, respectively.

3.6 Trilevel optimization for enhancing dialogue quality

After the initial dialogue generation, their quality must be assessed, as effective evaluation is crucial for improvement. However, traditional metrics often struggle to comprehensively capture the multidimensional nature of dialogue quality and heavily rely on reference texts, making them ill-suited for evaluating synthetic dialogues. Meanwhile, human evaluation is costly and highly subjective. Observing that multi-turn dialogues consist of a sequence of consecutive single-turn exchanges, we note that single-turn dialogues focus on the quality of individual responses and serve as the foundation for high-quality multi-turn interactions, while multi-turn dialogues further emphasize global coherence and naturalness built upon this foundation. These two levels are interdependent, and their joint optimization can be naturally formulated as a bilevel optimization problem. Within this framework, by introducing a module capable of automatically searching for the metric to evaluate dialogue quality and guide model training, the problem can be further formalized as a trilevel optimization problem. This enables end-to-end automatic generation, evaluation, and iterative refinement of multi-turn dialogues, effectively addressing the limitations of existing approaches. The trilevel optimization problem can be formulated as Eq. (5):

$$\begin{aligned} &\min_{\boldsymbol{\omega}} \hbar(\boldsymbol{\theta}, \boldsymbol{\phi}) \\ &\text{s.t.} \quad \boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}'} f(\boldsymbol{\omega}, \boldsymbol{\theta}', \boldsymbol{\phi}; x) \\ &\quad \text{s.t.} \quad \boldsymbol{\phi} = \arg\min_{\boldsymbol{\phi}'} g(\boldsymbol{\omega}, \boldsymbol{\theta}', \boldsymbol{\phi}'; x) \\ &\text{var.} \quad \boldsymbol{\omega}, \boldsymbol{\theta}, \boldsymbol{\phi}, \end{aligned} \tag{5}$$

where $\hbar$ is a dialogue quality scoring function implemented using an expert large language model, and $\boldsymbol{\omega} \in \mathbb{R}^{d_{\boldsymbol{\omega}}}$ denotes the parameters that define the metric, f and g are the metrics for evaluating multi-turn and single-turn dialogue quality, respectively, both influenced by $\boldsymbol{\omega}$, and $\boldsymbol{\theta} \in \mathbb{R}^{d_{\boldsymbol{\theta}}}, \boldsymbol{\phi} \in \mathbb{R}^{d_{\boldsymbol{\phi}}}$ represent the model parameters controlling the prompts used in multi-turn and single-turn dialogue generation. In addition, we consider that the dialogue generation process involves closed-source black-box LLMs, where first-order information is not accessible. Thus, we employ two-point zeroth-order estimate to approximate the gradient for optimizing $\boldsymbol{\theta}, \boldsymbol{\phi}$, which can be expressed by Eq. (6) and Eq. (7):

$$\hat{\nabla} f(\boldsymbol{\theta}') = \frac{f(\boldsymbol{\omega}, \boldsymbol{\theta}' + \mu_1 \boldsymbol{u}_1, \boldsymbol{\phi}; x) - f(\boldsymbol{\omega}, \boldsymbol{\theta}', \boldsymbol{\phi}; x)}{\mu_1} \boldsymbol{u}_1, \tag{6}$$

$$\hat{\nabla} g(\boldsymbol{\phi}') = \frac{g(\boldsymbol{\omega}, \boldsymbol{\theta}', \boldsymbol{\phi}' + \mu_2 \boldsymbol{u}_2; x) - g(\boldsymbol{\omega}, \boldsymbol{\theta}', \boldsymbol{\phi}'; x)}{\mu_2} \boldsymbol{u}_2, \tag{7}$$

where $\boldsymbol{u}_1, \boldsymbol{u}_2 \sim \mathcal{N}(0, \mathbf{I})$ are random direction vectors and μ_1, μ_2 are perturbation radii.

To optimize the metric-defining parameters $\boldsymbol{\omega}$, we cast the evaluation metric as a differentiable-loss-like function. However, due to its discrete structure, gradient-based optimization is infeasible, instead, we use an evolutionary algorithm. Specifically, each candidate loss function is represented as a tree-structured computational graph: the root outputs the final score, internal nodes are operators, and leaves are base dialogue quality metrics (e.g., distinct-n), whose combination is governed by $\boldsymbol{\omega}$. The graph is initialized by recursively expanding from the root—sampling operators and adding children according to their arity until at least one root-to-leaf path reaches depth d, ensuring sufficient complexity. After initializing the loss function, we optimize the prompt-control parameters $\boldsymbol{\theta}, \boldsymbol{\phi}$ based on this loss function and obtain the corresponding dialogue quality score. In each evolutionary iteration, we select the loss function that yields the highest average dialogue quality as the parent, and generate offspring using one of the following three strategies:

- **Copy**: Copy the parent's computational graph.
- **Re-initialization**: Generate a new computational graph according to the initialization procedure described above.
- **Mutation**: Apply a mutation operation to the parent's computational graph to produce an offspring.

The candidate mutation operations include the following:

- **Insertion**:Insert a random operator between a non-root node and its parent, adding a leaf if arity > 1.
- **Deletion**: Select an intermediate node for removal, then connect one of its children to its former parent.
- **Replacement**: Swap a node's operator, adjusting children to match the new arity.

The evolutionary framework enables efficient exploration of the discrete space of loss functions while leveraging downstream dialogue quality feedback to guide the search.

4 Dataset

Based on the aforementioned data generation method, we constructed a dataset named RealReasoning, aimed at evaluating the logical reasoning capabilities of LLMs in real-world scenarios. The dataset was built following the framework proposed in Section 3.1. In this study, we employed Qwen-Max as the large language model to drive the agents, used it to generate the identities of user agents, and simulate the interactions between user agents and assistant agents. The RealReasoning dataset contains 500 multi-turn task-oriented dialogues, all generated through the aforementioned framework. For each dialogue, we first manually design a logical reasoning question based on its content and then annotate a ground-truth label for the question.

4.1 Dataset description and statistical analysis

In the construction of the external knowledge source, we utilize the Bing search engine to retrieve relevant external knowledge. For each search, keywords are extracted from a predefined lexicon that covers a wide range of scenarios requiring logical reasoning, including mathematical and commonsense reasoning. The plain text content from the retrieved web pages is then used to build the external knowledge base related to the scenario.

Additionally, we provide statistical details regarding the dataset, as is shown in Table 1.

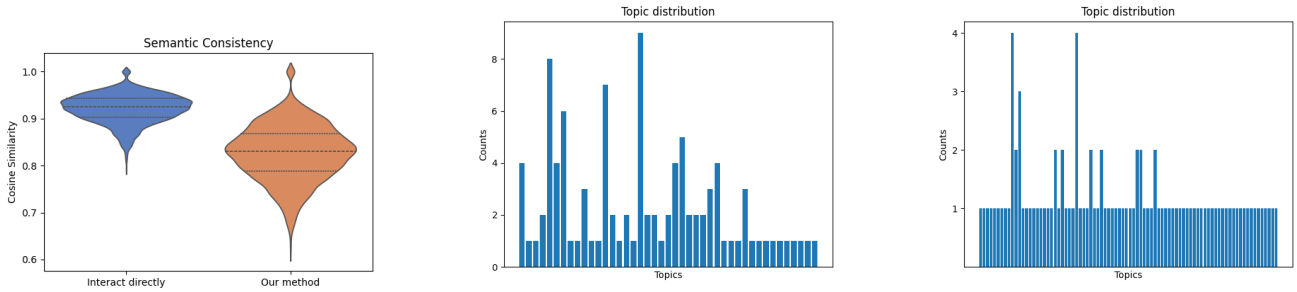
Metric	RealReasoning
Dialogues	500
Total turns	2398
Avg. turns per dialogue	4.796
Avg. tokens per turn	69.456
Unique tokens/Total tokens	0.024

Table 1: RealReasoning Dataset Statistics

4.2 Reasoning task construction and human annotation

We manually designed logical reasoning questions based on the dialogues generated by our method and annotated corresponding ground-truth labels, as the dataset inherently poses a certain level of reasoning difficulty. Even state-of-the-art LLMs struggle to provide consistently accurate answers with high reliability. Therefore, it poses certain challenges to achieve fully automated label annotation using LLMs and manual annotation continues to be an essential component in the process. In other words, if current LLMs are able to provide highly accurate and consistently reliable answers, it suggests that the generated reasoning tasks are too simplistic to effectively evaluate the models' reasoning capabilities.

We carefully read and analyzed the multi-turn task-oriented dialogues and designed questions based on the task-oriented information they contain. Logical reasoning tasks refer to tasks that require deriving new conclusions or unknown information from known information or premises through logic and rules. These tasks cannot be solved simply by extracting textual information. Specifically, we constructed two main categories of logical reasoning tasks — math word reasoning and common-sense reasoning — based on the dialogue content. Math word problem reasoning involves scenarios that describe characters, entities, and quantities. These problems typically consist of multi-step arithmetic operations and assess a model's abilities in language comprehension, semantic parsing, and mathematical reasoning. The label for a math word reasoning task is an exact number that must be obtained through multi-step calculations and cannot be directly derived. In contrast, common-sense reasoning requires synthesizing multiple pieces of information, combining explicitly stated knowledge from the text with implicit common-sense knowledge from the model to yield the final answer. Common-sense reasoning tasks use



(a) Cosine similarity of dialogues generated by two distinct approaches.

(b) Topic distribution of dialogues generated by interacting directly.

(c) Topic distribution of dialogues generated by our method.

Figure 2: The quality of generated dialogues.

binary classification labels. For each data, we provided question prompts and corresponding labels for the associated multi-turn dialogues. In the dataset, common-sense reasoning tasks account for 41%, while math word reasoning tasks account for 59%.

4.3 Iterative updating of the dataset

To further improve the quality of the data in the dataset and ensure that it adequately evaluates the reasoning capabilities of models, this paper adopts an iterative optimization approach, building upon the existing data in the dataset. Given a dataset $\mathcal{D} = \{(\boldsymbol{Y}^i, \boldsymbol{P}^i, l^i)\}_{i=1}^n$ constructed using the aforementioned method, where Y^i represents the i-th multi-turn dialogue, $\boldsymbol{P}^i$ represents the problem derived from the i-th dialogue, l^i is the corresponding label for the problem and n is the current size of the dataset. To implement the iterative optimization process, we introduce a knowledge base of difficult problems, denoted as $\boldsymbol{K_P}$, which is used to store challenging problems identified from the dataset.

For each data instance $(\boldsymbol{Y}^i, \boldsymbol{P}^i, l^i)$ in the RealReasoning dataset, where $i \in [1, n]$, we perform reasoning using k reasoning models $\{\mathcal{M}_i\}_{i=1}^k$ and obtain the corresponding results. The average accuracy of solutions for each data instance is denoted as acc^i, forming a data quadruplet $(\boldsymbol{Y}^i, \boldsymbol{P}^i, l^i, acc^i)$. Instances solved correctly by all reasoning models, in other words, data instances with $acc^i = 1$ are considered as easy problems. Instances with $acc^i < \phi$ are considered as difficult problems, where ϕ is a predefined threshold currently set to 0.5. Each difficult problem is then added to the difficult problem knowledge base $\boldsymbol{K_P}$.

After updating the knowledge base, a problem generation model $\mathcal{I}$ is utilized to generate new problems based on existing dialogues $\boldsymbol{Y}^j$, with reference to the difficult problems stored in $\boldsymbol{K_P}$. The new problem is generated based on both, which can be formulated as eq(8):

$$\hat{\boldsymbol{P}} = \mathcal{I}(\boldsymbol{Y}^j, \boldsymbol{P}^k), \tag{8}$$

where $\boldsymbol{P}^k$ is a difficult problem sampled from $\boldsymbol{K_P}$, and $\hat{\boldsymbol{P}}$ represents the newly generated problem. To maximize the proportion of high-quality data in the dataset, we select $\boldsymbol{Y}^j, \boldsymbol{P}^k$ according to specific conditions. Specifically,

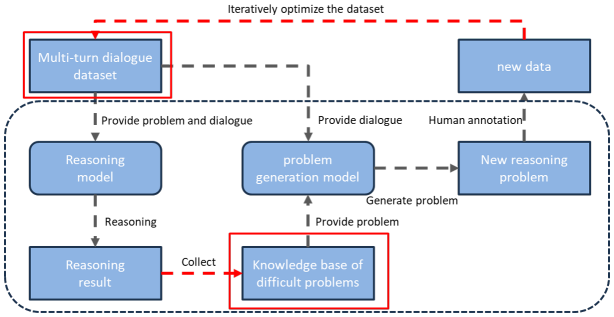


Figure 3: Iterative updating of the dataset.

for all data instances with $acc^j = 1$, the corresponding dialogue data $\boldsymbol{Y}^j$ is provided. After confirming the multi-turn dialogue, the cosine similarity is used to find the difficult problem in the difficult problem knowledge base that has the highest similarity to this dialogue, which can be formulated as eq(9),eq(10):

$$k = \arg\max_{k \in [1,m]} \text{CosineSimilarity}(\boldsymbol{Y}^j, \boldsymbol{P}^k), \tag{9}$$

$$\text{CosineSimilarity}(\boldsymbol{Y}^j, \boldsymbol{P}^k) = \frac{\boldsymbol{v}_{\boldsymbol{Y}^j} \cdot \boldsymbol{v}_{\boldsymbol{P}^k}}{\|v_{\boldsymbol{Y}_j}\| \|\boldsymbol{v}_{\boldsymbol{P}_k}\|}, \tag{10}$$

where m is the total number of problems in $\boldsymbol{K}_p$, $\boldsymbol{v}_{\boldsymbol{Y}_j}$, $\boldsymbol{v}_{\boldsymbol{P}_k}$ represent the text embeddings of the dialogue $\boldsymbol{Y}_j$ and the difficult problem $\boldsymbol{P}_k$.

Following the problem generation phase, automatic label generation and human verification are conducted to form a new data triplet $(\boldsymbol{Y}^j, \hat{\boldsymbol{P}}, \hat{l})$, which is then added to the dataset $\mathcal{D}$ to replace the data instance with $acc^j = 1$. Through this iterative process, both the dataset $\mathcal{D}$ and the difficult problem knowledge base $\boldsymbol{K_P}$ are iteratively updated. This process will be repeated continuously until there are no more data instance with simple problem. The overall workflow is illustrated in the figure 3. We utilized the Qwen3-235B-Instruct as the problem generation model and employed Qwen-Plus, Qwen-Turbo and DeepSeek-R1-Distill-Qwen-1.5B as the reasoning models. For text embedding, we used thenlper/gte-large-zh. Through this pro-

cess, we conducted the evaluation and modification of simple problems. After multiple rounds of iteration, the ratio of simple problems dropped from 26.6% to 0.6%, effectively enhancing the quality of problems in the dataset, increasing the ratio of challenging problems, and ensuring that the dataset could provide a higher level of challenge as a benchmark.

4.4 Dialogue evaluation

To evaluate the quality of the dialogues generated by the proposed method, we employ the semantic consistency metric and record the topic distribution to assess the similarity between generated contents.

We measure dialogue diversity through semantic similarity. Specifically, we use the thenlper/gte-large-zh as the text embedding model to encode utterances into vector representations and compute the cosine similarity $\text{CosineSimilarity}(\boldsymbol{Y}^a, \boldsymbol{Y}^b)$ between pairs of dialogues to quantify their semantic similarity.

The distribution of cosine similarity is presented in Figure 2a. We observe that the dialogues generated by directly interacting the user and assistant agents—without pre-assigning a user identity or generating an action sequence—exhibit significantly higher semantic similarity, with an average cosine similarity of 0.921, compared to 0.827 for dialogues produced by our framework.

Moreover, statistical analysis of topic distributions reveals that our method can still ensure sufficient diversity in conversation topics even with limited input. Specifically, we use "business travel reimbursement" as the designated domain for input and facilitate interact directly between a user agent and a assistant agent based on this keyword, either directly or after actions guided by our method. We record the multi-turn dialogues generated and employ Qwen-Max to summarize the topics of these dialogues. As shown in the figure 2b,2c, dialogues generated from direct interaction exhibit significant repetition in topics and contain fewer unique topics. In contrast, dialogues generated using our method show virtually no overlap in topics, demonstrating that varied experiences of user agents influence dialogue topics, thereby making the generated content more realistic and better suited for reasoning tasks. We provide more detailed ablation studies in Appendix A.

To further validate the effectiveness of the trilevel optimization framework in improving dialogue quality, we employ qwen-max as a proxy evaluator to score synthesized dialogues along three established dimensions of dialogue quality: Coherence, Fluency, and Diversity. The model is provided with the task description, scoring criteria, and the generated multi-turn dialogue, and assigns scores on a 1–5 scale. We include an unoptimized baseline without any metric-based optimization to demonstrate the effectiveness of different metrics and compare the impact of different automatic evaluation metrics on dialogue quality. As shown in Table 2, our approach to improving dialogue quality achieves strong performance across all three dialogue quality evaluation dimensions and attains the highest average score, demonstrating that the proposed trilevel optimization framework is effective in improving dialogue quality.

Overall, our framework enhances dialogue diversity and coherence even under minimal input, enabling scalable and realistic dialogue synthesis.

Metric	Coherence	Fluency	Diversity	Avg
Unoptimized	4.33	3.08	2.96	3.46
distinct-N	**4.91**	3.50	3.07	3.82
TF-IDF	4.35	3.28	3.10	3.57
Embedding	4.90	3.50	3.00	3.80
ours	4.90	**4.02**	**3.12**	**4.01**

Table 2: The impact of different metrics on dialogue quality.

5 Experimental results

Experiments are conducted on the RealReasoning dataset to evaluate the ability of large language models to perform realistic logical reasoning. The experimental results and corresponding analyses are provided below to explore the impact of the RealReasoning dataset and the factors influencing model reasoning performance.

5.1 Main results

We conduct reasoning tasks on the synthetic dataset using various LLMs and collect the corresponding outputs. Each model is presented with reasoning tasks together with their associated dialogue contexts. We instruct the model to output only the final result, without including any intermediate reasoning steps or internal thought processes. The generated responses are then compared to the ground truth annotations to evaluate performance. We also employ two publicly available datasets, GSM8K(Cobbe et al. 2021) and CODAH(Chen et al. 2019) , for comparison. As is shown in Table 3, the result shows that even advanced LLMs, such as qwen-plus, struggle to achieve strong performance on the realistic logical reasoning tasks presented in our dataset. Meanwhile, models that come with an inherent reasoning process, such as DeepSeek-R1, have demonstrated better performance on tasks that require complex reasoning to solve.

We primarily used models from the qwen series(Yang et al. 2024) and the deepseek series(Guo et al. 2025) for our experiments. Among them, qwen-plus, and qwen-turbo models responded to problems under a zero-shot prompting scenario. The remaining models underwent thorough reasoning and deliberation before outputting their answers, specifically, qwen-plus-thinking and qwen-turbo-thinking refer to invoking qwen-plus and qwen-turbo with 'enable_thinking' set to 'True'.

5.2 Analysis and findings

Reasoning vs No-reasoning. Based on the RealReasoning dataset, we conducted zero-shot prompting experiments using the qwen series models. The results indicate that when models are not prompted to engage in reasoning and are instead asked to directly solve reasoning tasks, their performance is suboptimal. Even large-scale models such as qwen-plus achieve only around 48.4% overall accuracy. Smaller

Model	RealReasoning math word reasoning	RealReasoning common-sense reasoning	RealReasoning	GSM8K	CODAH
qwen-plus	40.6%	59.5%	48.4%	69.1%	94.8%
qwen-plus-thinking	88.1%	74.6%	82.6%	94.2%	95.0%
qwen-turbo	31.8%	52.6%	40.4%	69.9%	90.8%
qwen-turbo-thinking	88.1%	77.5%	83.8%	**94.4**%	86.3%
deepseek-r1	**89.4**%	**85.3**%	**87.8**%	94.2%	**95.2**%
deepseek-r1-distill-qwen-32b	78.3%	71.7%	75.6%	68.8%	88.6%
deepseek-r1-distill-qwen-1.5b	37.6%	40.0%	38.6%	61.7%	35.4%
deepseek-r1-distill-llama-70b	48.4%	69.2%	57.0%	64.2%	93.5%

Table 3: Answer accuracy of different models on datasets.

models like qwen-turbo exhibit a more pronounced performance degradation on common-sense reasoning tasks, making them ineffective for realistic logical problem-solving.

Upon further examination of the output, it becomes evident that weaker models tend to mechanically select the first available option when faced with closed-ended reasoning questions. This behavior reflects an inability to deeply understand or effectively analyze specific problem contexts, significantly limiting their performance on realistic logical reasoning tasks.

Conversely, models equipped with mechanisms for reflective reasoning demonstrate significant advantages on tasks requiring multi-step inference or deep comprehension. For instance, the DeepSeek model series leverages extensive reflection to thoroughly analyze problems, leading to more accurate responses. Specifically, DeepSeek-R1 utilizes long chains of reasoning to meticulously explore each question. By virtue of its vast parameter size, DeepSeek-R1 accumulates substantial internal knowledge during pre-training, enabling superior performance on realistic logical reasoning tasks. The performance gap between qwen-turbo-thinking and qwen-turbo clearly demonstrates the effectiveness of a thorough reasoning process in solving reasoning tasks.

Math word reasoning vs Common-sense reasoning. When addressing two distinct types of reasoning tasks—math word reasoning and common-sense reasoning—the latter exhibits significantly greater difficulty. Even the best-performing model in current experiments, deepseek-r1, achieves only an 85.3% accuracy rate on common-sense reasoning tasks. This indicates that, for tasks requiring auxiliary internal knowledge embedded within model parameters, it is challenging to substantially improve reasoning accuracy solely through explicit reasoning processes. Current models with strong reasoning capabilities tend to focus more on external information provided as input, lacking sufficient engagement with the implicit internal knowledge contained within their parameters.

In contrast, math word reasoning tasks provide all necessary information externally, enabling even small-parameter models like deepseek-r1-distill-qwen-1.5b to effectively extract and utilize this information to derive correct answers. The characteristics of the math word reasoning tasks lead to a larger performance gap between models with reasoning and those without.

Summary. Experimental results demonstrate that, compared to tasks from existing public reasoning datasets that are relatively easier to solve, models still face certain difficulties when solving logical reasoning tasks in real-world scenarios. This highlights the value of the data generation method proposed in this paper for evaluating model reasoning capabilities.

6 Conclusion and Future Work

This study presents a novel framework for generating multi-turn task-oriented dialogues using LLM agents, with a focus on evaluating models' logical reasoning capabilities in realistic scenarios. By leveraging the generative and role-playing strengths of LLM agents, our approach effectively constructs dialogue data that is grounded in real-world contexts. Real-world scenarios help maintain contextual coherence across dialogue turns and inherently increase the difficulty of logical reasoning. In addition, we introduce a newly curated dataset specifically designed to assess model performance on such reasoning tasks. Experimental results demonstrate that current models still face significant challenges in handling these complex reasoning tasks, highlighting the need for further research in this direction.

Limitations. The annotation of the data generated in this study relies on manual labeling, and expanding the dataset still requires considerable human effort. Moreover, the quality of the generated data can be influenced by the performance of the LLMs used.

We plan to explore two main directions for future work. First, we aim to extend the dialogue generation framework from user-assistant interactions to multi-agent settings by incorporating more sophisticated prompting strategies and memory mechanisms. Second, we intend to investigate automated methods for labeling generated dialogue data, aiming to maintain task difficulty while reducing the reliance on labor-intensive manual annotation processes.

References

Abdullin, Y.; Molla, D.; Ofoghi, B.; et al. 2023. Synthetic Dialogue Dataset Generation using LLM Agents. In *Pro-*

ceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM), 181–191.

Achiam, J.; Adler, S.; Agarwal, S.; et al. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*.

Algherairy, A.; and Ahmed, M. 2024. A Review of Dialogue Systems: Current Trends and Future Directions. *Neural Computing and Applications*, 36(12): 6325–6351.

Anaby-Tavor, A.; Carmeli, B.; Goldbraich, E.; et al. 2020. Do Not Have Enough Data? Deep Learning to the Rescue! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 7383–7390.

Budzianowski, P.; Wen, T. H.; Tseng, B. H.; et al. 2018. MultiWOZ-A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 5016–5026.

Chang, Y.; Wang, X.; Wang, J.; et al. 2024. A Survey on Evaluation of Large Language Models. *ACM Transactions on Intelligent Systems and Technology*, 15(3): 1–45.

Chen, J.; and Yang, D. 2021. Simple Conversational Data Augmentation for Semi-Supervised Abstractive Dialogue Summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 6605–6616.

Chen, M.; D'Arcy, M.; Liu, A.; Fernandez, J.; and Downey, D. 2019. CODAH: An Adversarially-Authored Question Answering Dataset for Common Sense. In *Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP*, 63–69.

Chen, Z.; Li, S.; Smiley, C.; et al. 2022. ConvFinQA: Exploring the Chain of Numerical Reasoning in Conversational Finance Question Answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 6279–6292.

Chung, W.; Cahyawijaya, S.; Wilie, B.; et al. 2023. InstructTODS: Large Language Models for End-to-End Task-Oriented Dialogue Systems. In *Proceedings of the Second Workshop on Natural Language Interfaces*, 1–21.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Deng, C.; Zhao, Y.; Heng, Y.; Li, Y.; Cao, J.; Tang, X.; and Cohan, A. 2024. Unveiling the Spectrum of Data Contamination in Language Model: A Survey from Detection to Remediation. In *Findings of the Association for Computational Linguistics ACL 2024*, 16078–16092.

Ding, B.; Liu, L.; Bing, L.; et al. 2020. DAGA: Data Augmentation with a Generation Approach for Low-resource Tagging Tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6045–6057.

Feng, J.; Sun, Q.; Xu, C.; et al. 2023. MMDialog: A Large-scale Multi-turn Dialogue Dataset Towards Multimodal Open-domain Conversation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 7348–7363.

Gao, M.; Hu, X.; Yin, X.; Ruan, J.; Pu, X.; and Wan, X. 2025. Llm-based nlg evaluation: Current status and challenges. *Computational Linguistics*, 1–28.

Garg, B.; Zhang, L.; Sridhara, P.; Hosseini, R.; Xing, E.; and Xie, P. 2022. Learning from mistakes–a framework for neural architecture search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 10184–10192.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Guo, M.; Yang, Y.; Xu, R.; Liu, Z.; and Lin, D. 2020. When nas meets robustness: In search of robust architectures against adversarial attacks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 631–640.

Guo, T.; Chen, X.; Wang, Y.; et al. 2024. Large language model based multi-agents: A survey of progress and challenges. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 8048–8057.

Ham, D.; Lee, J.-G.; Jang, Y.; and Kim, K.-E. 2020. End-to-end neural pipeline for goal-oriented dialogue systems using GPT-2. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, 583–592.

He, Z.; Cao, P.; Chen, Y.; Liu, K.; Li, R.; Sun, M.; and Zhao, J. 2023. LEGO: A multi-agent collaborative framework with role-playing and iterative feedback for causality explanation generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 9142–9163.

Hong, S.; Zhuge, M.; Chen, J.; Zheng, X.; Cheng, Y.; Wang, J.; Zhang, C.; Wang, Z.; Yau, S. K. S.; Lin, Z.; et al. 2024. MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework. In *International Conference on Learning Representations (ICLR)*.

Huang, J.; and Chang, K. C. C. 2023. Towards Reasoning in Large Language Models: A Survey. In *Findings of the Association for Computational Linguistics: ACL 2023*, 1049–1065.

Huang, M.; Zhu, X.; and Gao, J. 2020. Challenges in building intelligent open-domain dialog systems. *ACM Transactions on Information Systems (TOIS)*, 38(3): 1–32.

Islam, M. A.; Ali, M. E.; and Parvez, M. R. 2024. MapCoder: Multi-Agent Code Generation for Competitive Problem Solving. In *62nd Annual Meeting of the Association for Computational Linguistics, ACL 2024*, 4912–4944. Association for Computational Linguistics (ACL).

Jian, C.; Yang, K.; and Jiao, Y. 2024. Tri-Level Navigator: LLM-Empowered Tri-Level Learning for Time Series OOD Generalization. *Advances in Neural Information Processing Systems*, 37: 110613–110642.

Jiao, Y.; Wang, X.; and Yang, K. 2025. Pr-attack: Coordinated prompt-rag attacks on retrieval-augmented generation in large language models via bilevel optimization. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 656–667.

Jiao, Y.; Yang, K.; and Jian, C. 2025. DTZO: Distributed Trilevel Zeroth Order Learning with Provable Non-Asymptotic Convergence. In *Forty-second International Conference on Machine Learning*.

Jiao, Y.; Yang, K.; and Song, D. 2025. Federated distributionally robust optimization with non-convex objectives: Algorithm and analysis. *IEEE Transactions on Mobile Computing*.

Jiao, Y.; Yang, K.; Song, D.; and Tao, D. 2022. Timeautoad: Autonomous anomaly detection with self-supervised contrastive loss for multivariate time series. *IEEE Transactions on Network Science and Engineering*, 9(3): 1604–1619.

Jiao, Y.; Yang, K.; Wu, T.; Jian, C.; and Huang, J. 2024. Provably convergent federated trilevel learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 12928–12937.

Jiao, Y.; Yang, K.; Wu, T.; Song, D.; and Jian, C. 2023. Asynchronous Distributed Bilevel Optimization. In *The Eleventh International Conference on Learning Representations*.

Kernycky, A.; Coleman, D.; Spence, C.; and Das, U. 2024. Evaluating the performance of llms on technical language processing tasks. In *International Conference on Human-Computer Interaction*, 75–85. Springer.

Kim, G.; Kim, S.; Yoo, K. M.; et al. 2022. Generating Information-Seeking Conversations from Unlabeled Documents. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2362–2378.

Kojima, T.; Gu, S. S.; Reid, M.; et al. 2022. Large Language Models are Zero-Shot Reasoners. *Advances in Neural Information Processing Systems*, 35: 22199–22213.

Kulhánek, J.; Hudeček, V.; Nekvinda, T.; and Dušek, O. 2021. AuGPT: Auxiliary Tasks and Data Augmentation for End-To-End Dialogue with Pre-Trained Language Models. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, 198–210.

Li, B.; Zhang, S.; Zhang, W.; Chen, Y.; Yang, C.; Hu, S.; Xu, T.; Liu, S.; and Li, J. 2023a. S2M: Converting Single-Turn to Multi-Turn Datasets for Conversational Question Answering. In *ECAI 2023*, 1365–1372. IOS Press.

Li, H.; Zhang, R.; Lee, Y.-C.; Kraut, R. E.; and Mohr, D. C. 2023b. Systematic review and meta-analysis of AI-based conversational agents for promoting mental health and well-being. *NPJ Digital Medicine*, 6(1): 236.

Li, J.; Tang, T.; Zhao, W. X.; et al. 2024a. Pre-trained Language Models for Text Generation: A Survey. *ACM Computing Surveys*, 56(9): 1–39.

Li, Z.; Cao, Y.; Xu, X.; Jiang, J.; Liu, X.; Teo, Y. S.; Lin, S.-W.; and Liu, Y. 2024b. Llms for relational reasoning: How far are we? In *Proceedings of the 1st International Workshop on Large Language Models for Code*, 119–126.

Lin, Z.; Madotto, A.; Winata, G. I.; and Fung, P. 2020. MinTL: Minimalist Transfer Learning for Task-Oriented Dialogue Systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 3391–3405.

Liu, Y.; Yang, K.; Zhu, Y.; Yang, K.; Jian, C.; Ni, W.; Ye, X.; and Ouyang, Y. 2025a. A Nested Zeroth-Order Fine-Tuning Approach for Cloud-Edge LLM Agents. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 3–15. Springer.

Liu, Y.; Yang, K.; Zhu, Y.; Yang, K.; and Zhao, H. 2025b. Argus: Federated Non-Convex Bilevel Learning Over 6 G Space-Air-Ground Integrated Network. *IEEE Transactions on Network Science and Engineering*.

Lu, P.; Qiu, L.; Yu, W.; et al. 2023. A Survey of Deep Learning for Mathematical Reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 14605–14631.

Ma, X.; Zhang, Z.; and Zhao, H. 2023. Enhanced speaker-aware multi-party multi-turn dialogue comprehension. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31: 2410–2423.

Magar, I.; and Schwartz, R. 2022. Data Contamination: From Memorization to Exploitation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 157–165.

Meng, Y.; Huang, J.; Zhang, Y.; et al. 2022. Generating Training Data with Language Models: Towards Zero-Shot Language Understanding. *Advances in Neural Information Processing Systems*, 35: 462–477.

Nguyen, H. H.; Liu, Y.; Zhang, C.; et al. 2023. CoF-CoT: Enhancing Large Language Models with Coarse-to-Fine Chain-of-Thought Prompting for Multi-domain NLU Tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 12109–12119.

Ouyang, L.; Wu, J.; Jiang, X.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744.

Owoicho, P.; Sekulic, I.; Aliannejadi, M.; et al. 2023. Exploiting Simulated User Feedback for Conversational Search: Ranking, Rewriting, and Beyond. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 632–642.

Park, J. S.; O'Brien, J.; Cai, C. J.; et al. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 1–22.

Qian, C.; Liu, W.; Liu, H.; Chen, N.; Dang, Y.; Li, J.; Yang, C.; Chen, W.; Su, Y.; Cong, X.; et al. 2024. ChatDev: Communicative Agents for Software Development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 15174–15186.

Qiu, H.; He, H.; Zhang, S.; et al. 2024. SMILE: Single-turn to Multi-turn Inclusive Language Expansion Via ChatGPT for Mental Health Support. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 615–636.

Quan, J.; Zhang, S.; Cao, Q.; et al. 2020. RiSAWOZ: A Large-Scale Multi-Domain Wizard-of-Oz Dataset with Rich Semantic Annotations for Task-Oriented Dialogue Modeling. In *Proceedings of the 2020 Conference on Empirical*

Methods in Natural Language Processing (EMNLP), 930–940.

Raghu, A.; Lorraine, J.; Kornblith, S.; McDermott, M.; and Duvenaud, D. K. 2021. Meta-learning to improve pre-training. *Advances in Neural Information Processing Systems*, 34: 23231–23244.

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

Sato, R.; Tanaka, M.; and Takeda, A. 2021. A gradient method for multilevel optimization. *Advances in Neural Information Processing Systems*, 34: 7522–7533.

Serban, I. V.; Lowe, R.; Henderson, P.; et al. 2018. A Survey of Available Corpora for Building Data-Driven Dialogue Systems: The Journal Version. *Dialogue & Discourse*, 9(1): 1–49.

Shah, P.; Hakkani-Tur, D.; Liu, B.; and Tur, G. 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)*, 41–51.

Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K.; and Yao, S. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36: 8634–8652.

Su, H.; Shen, X.; Zhang, R.; Sun, F.; Hu, P.; Niu, C.; and Zhou, J. 2019. Improving Multi-turn Dialogue Modelling with Utterance ReWriter. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 22–31.

Wang, J.; Liu, J.; Bi, W.; et al. 2020. Dual Dynamic Memory Network for End-to-End Multi-Turn Task-Oriented Dialog Systems. In *Proceedings of the 28th International Conference on Computational Linguistics*, 4100–4110.

Wei, J.; Tay, Y.; Bommasani, R.; et al. 2022. Emergent Abilities of Large Language Models. *Transactions on Machine Learning Research*.

Xi, Z.; Chen, W.; Guo, X.; et al. 2025. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 68(2): 121101.

Xu, Y.; Zhao, H.; and Zhang, Z. 2021. Topic-aware multi-turn dialogue modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 14176–14184.

Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Yang, K.; Wu, Y.; Huang, J.; Wang, X.; and Verdú, S. 2008. Distributed robust optimization for communication networks. In *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*, 1157–1165. IEEE.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; and Cao, Y. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Yao, Y.; Duan, J.; Xu, K.; et al. 2024. A Survey on Large Language Model (LLM) Security and Privacy: The Good, the Bad, and the Ugly. *High-Confidence Computing*, 100211.

Ye, J.; Gao, J.; Li, Q.; et al. 2022. ZeroGen: Efficient Zero-shot Learning via Dataset Generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 11653–11669.

Yu, F.; Zhang, H.; Tiwari, P.; et al. 2024. Natural language reasoning, a survey. *ACM Computing Surveys*, 56(12): 1–39.

Zhang, C.; Yang, K.; Hu, S.; Wang, Z.; Li, G.; Sun, Y.; Zhang, C.; Zhang, Z.; Liu, A.; Zhu, S.-C.; et al. 2024. Proagent: building proactive cooperative agents with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 17591–17599.

Zhang, Q.; Wang, L.; Yu, S.; et al. 2021. NOAHQA: Numerical Reasoning with Interpretable Graph Question Answering Dataset. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 4147–4161.

Zhu, Q.; Huang, K.; Zhang, Z.; et al. 2020. CrossWOZ: A Large-Scale Chinese Cross-Domain Task-Oriented Dialogue Dataset. *Transactions of the Association for Computational Linguistics*, 8: 281–295.

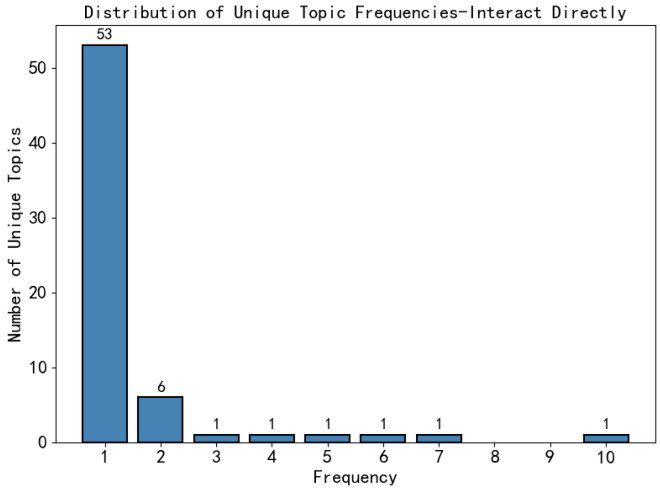
A Ablation Study on Synthetic Dialogues

This subsection validates the effectiveness of the different modules proposed in this chapter through ablation studies. All dialogues are generated under the same scenario, with user information, action sequence. Each subsequent configuration includes all previously added components, while the baseline, which is direct dialogue generation without any of these components, serves as the reference point. The impact of each component is evaluated by comparing the topic distributions of the generated dialogues and the cosine similarities between their corresponding semantic embeddings. The relevant experimental results are shown in the figure 4,5 and table 4. As shown in Figure 4, compared to the baseline of direct interaction between user and system agents, equipping the user agent with the modules proposed in this chapter leads to significantly greater diversity in topic distribution. When all modules are incorporated, more than 70% of generated dialogues exhibit a unique topic, indicating higher inter-dialogue distinctiveness. This suggests that the synthetic dialogues achieve broader coverage of the target scenario and better account for the variety of events likely to occur within it. Moreover, the proportion of unique topics consistently increases as modules are incrementally added, demonstrating that each component positively contributes to enhancing topical diversity.

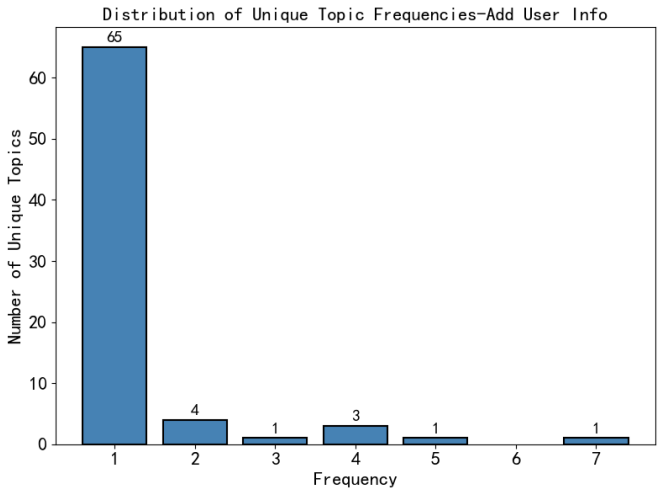
Figure 5 further reveals that incorporating the user identity module yields the most pronounced reduction in cosine similarity, while the subsequent addition of the action sequence module also exerts a beneficial effect in further lowering similarity scores.

Finally, Table 4 shows that under the scene-based direct dialogue setting, the maximum cosine similarity reaches 1.0, indicating the generation of identical dialogue pairs. This reflects a tendency of large language model agents to fall into repetitive response patterns when no auxiliary information is provided, resorting instead to conservative outputs based solely on internal priors. Notably, the introduction of action sequence information substantially reduces the minimum cosine similarity between dialogues.

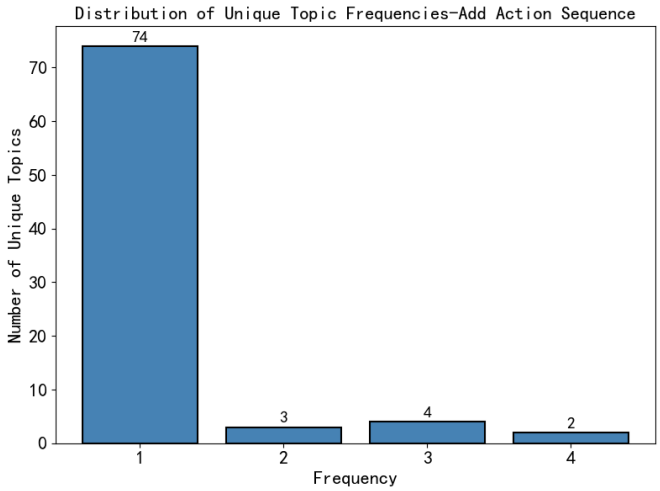
The statistics on topic diversity and repetition across different methods further validate the effectiveness of our proposed modules. Here, “Unique Topics” denotes the number of distinct dialogue topics covered, and “Max Repetition” refers to the highest frequency of any single topic. Both the user identity and action sequence modules markedly improve topic diversity, thereby contributing positively to overall dialogue quality.



(a) Distribution of Unique Topic Frequencies-Interact Directly.



(b) Distribution of Unique Topic Frequencies-Add User Info.



(c) Distribution of Unique Topic Frequencies-Add Action Sequence.

Figure 4: Impact of Ablation Components on Dialogue Topic Distribution.

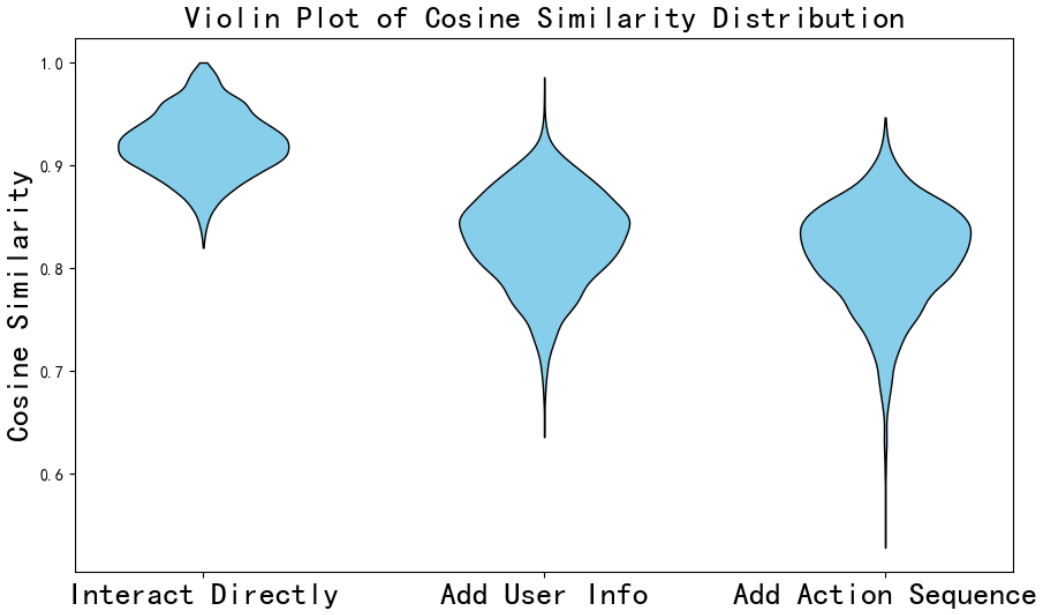


Figure 5: Impact of Ablation Components on Cosine Similarity Distribution.

Table 4: Comparison of Multi-turn Dialogue Generation Results with Different Ablation Components Combined

Method	**Cosine Similarity**			**Topic Diversity and Redundancy**	
	Max	Min	Mean	Unique Topics	Max Repetition
Direct Interaction	1.00	0.820	0.921	65	10
+ User Identity Info	0.986	0.636	0.830	75	7
+ Action Sequence Info	**0.946**	**0.528**	**0.813**	**83**	**4**

Reproducibility Checklist

Instructions for Authors:

This document outlines key aspects for assessing reproducibility. Please provide your input by editing this `.tex` file directly.

For each question (that applies), replace the "Type your response here" text with your answer.

Example: If a question appears as

```
\question{Proofs of all novel claims
are included} {(yes/partial/no)}
Type your response here
```

you would change it to:

```
\question{Proofs of all novel claims
are included} {(yes/partial/no)}
yes
```

Please make sure to:

- Replace ONLY the "Type your response here" text and nothing else.
- Use one of the options listed for that question (e.g., **yes**, **no**, **partial**, or **NA**).
- **Not** modify any other part of the `\question` command or any other lines in this document.

You can `\input` this .tex file right before `\end{document}` of your main file or compile it as a stand-alone document. Check the instructions on your conference's website to see if you will be asked to provide this checklist with your paper or separately.

1. General Paper Structure

1.1. Includes a conceptual outline and/or pseudocode description of AI methods introduced (yes/partial/no/NA) yes

1.2. Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results (yes/no) yes

1.3. Provides well-marked pedagogical references for less-familiar readers to gain background necessary to replicate the paper (yes/no) yes

2. Theoretical Contributions

2.1. Does this paper make theoretical contributions? (yes/no) yes

If yes, please address the following points:

2.2. All assumptions and restrictions are stated clearly and formally (yes/partial/no) yes

2.3. All novel claims are stated formally (e.g., in theorem statements) (yes/partial/no) yes

2.4. Proofs of all novel claims are included (yes/partial/no) no

2.5. Proof sketches or intuitions are given for complex and/or novel results (yes/partial/no) yes

2.6. Appropriate citations to theoretical tools used are given (yes/partial/no) yes

2.7. All theoretical claims are demonstrated empirically to hold (yes/partial/no/NA) yes

2.8. All experimental code used to eliminate or disprove claims is included (yes/no/NA) no

3. Dataset Usage

3.1. Does this paper rely on one or more datasets? (yes/no) yes

If yes, please address the following points:

3.2. A motivation is given for why the experiments are conducted on the selected datasets (yes/partial/no/NA) yes

3.3. All novel datasets introduced in this paper are included in a data appendix (yes/partial/no/NA) yes

3.4. All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no/NA) yes

3.5. All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations (yes/no/NA) NA

3.6. All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available (yes/partial/no/NA) NA

3.7. All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisficing (yes/partial/no/NA) NA

4. Computational Experiments

4.1. Does this paper include computational experiments? (yes/no) yes

If yes, please address the following points:

4.2. This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting (yes/partial/no/NA) yes

4.3. Any code required for pre-processing data is included in the appendix (yes/partial/no) yes

4.4. All source code required for conducting and analyzing the experiments is included in a code appendix (yes/partial/no) yes

4.5. All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no) yes

4.6. All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from (yes/partial/no) partial

4.7. If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results (yes/partial/no/NA) yes

4.8. This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks (yes/partial/no) yes

4.9. This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics (yes/partial/no) yes

4.10. This paper states the number of algorithm runs used to compute each reported result (yes/no) yes

4.11. Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information (yes/no) yes

4.12. The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank) (yes/partial/no) yes

4.13. This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments (yes/partial/no/NA) yes