DIAMONDs: A Dataset for 𝔻ynamic 𝕀nformation 𝔸nd 𝕄ental modeling 𝕆f ℕumeric 𝔻iscussions

Sayontan Ghosh*, **Mahnaz Koupaee***, **Yash Kumar Lal***, **Pegah Alipoormolabashi***, **Mohammad Saqib Hasan***, **Jun Seok Kang**†, **Niranjan Balasubramanian***
* Department of Computer Science, Stony Brook University
† Blink Health
{sagghosh, mkoupaee, ylal, palipoormola, mdshasan, niranjan}@cs.stonybrook.edu, jun.s.kang@gmail.com

Abstract

Understanding multiparty conversations demands robust Theory of Mind (ToM) capabilities, including the ability to track dynamic information, manage knowledge asymmetries, and distinguish relevant information across extended exchanges. To advance ToM evaluation in such settings, we present a carefully designed scalable methodology for generating high-quality benchmark conversation-question pairs with these characteristics. Using this methodology, we create DIAMONDs, a new conversational QA dataset covering common business, financial or other group interactions. In these goal-oriented conversations, participants often have to track certain numerical quantities (say *expected profit*) of interest that can be derived from other variable quantities (like *marketing expenses, expected sales, salary*, etc.), whose values also change over the course of the conversation. DIAMONDs questions pose simple numerical reasoning problems over such quantities of interest (e.g., *funds required for charity events, expected company profit next quarter*, etc.) in the context of the information exchanged in conversations. This allows for precisely evaluating ToM capabilities for carefully tracking and reasoning over participants' knowledge states.

Our evaluation of state-of-the-art language models reveals significant challenges in handling participant-centric reasoning, specifically in situations where participants have false beliefs. Models also struggle with conversations containing distractors and show limited ability to identify scenarios with insufficient information. These findings highlight current models' ToM limitations in handling real-world multi-party conversations.[1]

1 Introduction

Effective communication and cooperation in multi-party conversations depends on the ability to model mental states such as beliefs, intents, desires, emotions, knowledge of oneself and others (Frith, 1994; Clark, 1996; Pickering & Garrod, 2004; De Rosnay & Hughes, 2006) – i.e., Theory of Mind (ToM) abilities. Understanding multi-party conversations thus presents a strong test bed for evaluating ToM capabilities, where participants must track not only the information being shared, but also maintain dynamic models of who has access to what information at different points in time. This ToM challenge becomes particularly complex in information rich conversations with multiple participants, common in settings such as business meetings, financial discussions, planning sessions, etc. In these settings, the states of key variables (e.g., *expenses, timelines, or task allocations* in a *project budgeting discussion*) frequently change due to new information, corrections, or updates. Consider an illustrative example in Figure 1 (red arrow) where during a *project logistics discussion*, *Alex increases his initial handout requirements from* 15 *to* 20 *units*

[1]Dataset and code are available at: https://github.com/StonyBrookNLP/diamonds

while *Chen is absent*. This seemingly simple update adds multiple reasoning demands[2] for conversation understanding: tracking long-term dependencies (connecting initial estimates with later adjustments), managing dynamic information states (updating quantities as new information arrives), handling information asymmetry (Chen's outdated knowledge state leading to false belief), and distinguishing relevant details from distractors (for quantities of interest) in information-rich exchanges.

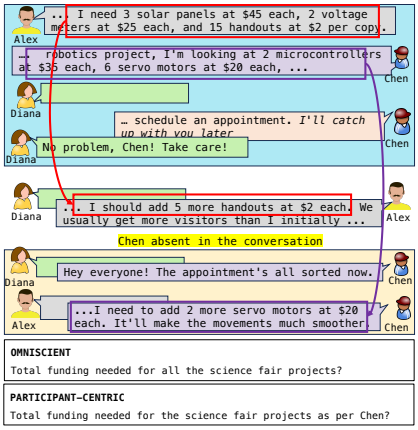

Figure 1: A conversation and question example from DIAMONDs. Alex increases his initial handout requirements from 15 to 20 units while Chen is absent. Chen also updates his motor requirement from 2 to 4 after returning.

Existing benchmarks for evaluating ToM capabilities in LLMs pose QA tasks on simple narratives, stories (Sap et al., 2022; Gandhi et al., 2023; Chen et al., 2024), or synthetic sequences of actions (Le et al., 2019b; Wu et al., 2023a; Xu et al., 2024). These resources, while valuable for basic ToM assessment, are either too simplistic to capture the complexities of real-world social interactions. Most evaluate ToM through categorical or extractive QA tasks where the context contains semantic cues directly related to the answer, avoiding the challenges of tracking and integrating information across extended exchanges. While the recently introduced FanTom dataset (Kim et al., 2023) tests ToM abilities in conversations, it mainly focuses on extractive questions where relevant information is typically localized within specific dialogue segments. In contrast, real-world multi-party conversations require sophisticated ToM capabilities to track dynamic information states, manage knowledge asymmetries across multiple participants, over entire extended conversations, and integrate this understanding with other cognitive demands like numerical reasoning - challenges that existing benchmarks fail to capture.

To address these limitations, we present DIAMONDs, a benchmark designed to stress-test ToM capabilities in multi-party conversations. DIAMONDs focuses on tracking and reasoning over multiple dynamic variables whose states changes across the conversation, in the presence of information asymmetries between participants. We embed multi-step numerical reasoning problems within multi-party conversational contexts where successful comprehension requires both robust information tracking and modeling of participant-specific knowledge states. This ensures answer to the question is unique, not-extractive while requiring multi-step reasoning over the information across the entire conversation.

Creating numerically consistent, detailed, and highly structured conversations with specific characteristics poses significant challenges for both human annotators and LLMs. To overcome this, we implemented a multi-stage synthetic data generation process where an LLM first generates a narrative script with a corresponding question-answer pair, then transforms this script into a multi-party conversation while preserving all question-relevant information. Each script consists of a ***premise*** that establishes the complete set of variable relevant to the question and their initial state, followed by a sequence of ***variable state perturbations*** that unfold throughout the conversation and perturbs the states of these variables. For example, in Figure 1, the premise relevant to the question about "total funding for science fair" sets up the space of the relevant variables such as "handouts Alex needs" (15), "servo motors Chen needs" (6), their respective prices ($2, $20), etc., along with their initial states/values. As the conversation progresses, these initial variable states are modified through perturbations such as *Chen stating that he needs* 2 *more servo motors, updates his requirement from* 6 *to* 8 or *Alex updating his handout requirement*. This combination of initial states and subsequent variable state perturbations ensures that the

[2]There are likely analogous cognitive demands for humans completing such tasks. Our goal here, however, is not draw connections with any specific set of human Theory-of-Mind capabilities but rather to design a stress test for understanding and improving models abilities to use such capabilities for conversation understanding.

state of the variables affecting the answer changes throughout the conversation, making accurate tracking of variable states essential for correct reasoning. After the LLM generates the premise and perturbation information, we integrate these elements into a coherent script using a probabilistic template sampled from a Markov process. The template contains slots for information exchanges and participant movements, defining information access patterns throughout the conversation. This approach introduces randomness into the generation process that is external to the LLM while scaffolding the premise, variable perturbation, participant movements into a coherent script.

The conversations in DIAMONDs induce several cognitive challenges that stress-test ToM capabilities: (1) tracking variable states as new information emerges, (2) resolving long-term dependencies between information shared at different points, (3) maintaining separate knowledge models for each participant based on their presence during information exchanges, (4) distinguishing relevant information from distractors, and (5) identifying information gaps. By presenting both omniscient questions (testing global understanding) and participant-centric questions (testing the ability to reason from specific participants' perspectives, including their potentially false beliefs), DIAMONDs provides a comprehensive evaluation of ToM capabilities in complex multi-party settings.

Our evaluation of state-of-the-art LLMs (`gpt-4o` (Achiam et al., 2023), `Claude-3.5-Sonnet`, and `Llama3` (Dubey et al., 2024)) on DIAMONDs reveals significant gaps in their ToM capability in context of multi-party conversation. Benchmarking reveals significant challenges in handling participant-centric reasoning, with performance dropping from 80.0% on omniscient questions to 55.1% on participant-centric questions. This performance further drops to 27.0% in participant-centric questions for participants with false beliefs. Models also show decreased performance when dealing with conversations with distractors and limited ability to identify scenarios with insufficient information. These findings highlight the limitations in ToM capability of current models needed in multi-party conversations.

2 Related Works

2.1 Conversation Comprehension: Existing conversation comprehension research has strong focus on classification tasks like slot filling (MultiWOZ (Han et al., 2021; Ye et al., 2022)), intent classification (CLINC150 (Larson et al., 2019), SILICONE (Chapuis et al., 2020)), and emotion detection (MELD Poria et al. (2019), EmoWOZ (Feng et al., 2022)). These tasks do not require complex multi-step reasoning capabilities across conversation that our work requires. Dialogue summarization represents another major research direction, with datasets like SAMSum (Gliwa et al., 2019), MeetingBank (Hu et al., 2023), DialogSum (Chen et al., 2021), and MediaSum (Zhu et al., 2021). While valuable, these tasks are primarily extractive and don't demand the precise tracking of dynamic numerical information central to our framework. Conversation-based QA has advanced through datasets such as CoQA (Reddy et al., 2019), QuAC (Choi et al., 2018), ShARC (Saeidi et al., 2018), OR-QuAC (Qu et al., 2020), and QReCC (Anantha et al., 2021). The most relevant work, Fantom, evaluates theory-of-mind capabilities in conversations with information asymmetry. However, Fantom (Kim et al., 2023) primarily tests extractive fact-based questions localized within conversation segments, whereas our work requires non-extractive, multi-step mathematical reasoning across multiple dynamic variables whose states evolves throughout entire conversations.

2.2 Theory of Mind Benchmarks: Narrative-based ToM benchmarks evaluate mental state reasoning through narrative contexts. Recent works include ToMi (Le et al., 2019a), which requires tracking belief states across multiple characters; Neural ToM (Sap et al., 2022), which tests the understanding of beliefs and intentions; HiToM (Wu et al., 2023b), which examines higher-order ToM; and OpenToM (Xu et al., 2024) introduces the dimension of participant character in the narrative context. While valuable for basic ToM assessment, these benchmarks often use simplified narratives with clear semantic cues, making them susceptible to reporting bias and lacking the dynamic complexity of real-world interactions. Conversation-based benchmarks better approximate real-world social interactions by evaluating ToM within dialogue contexts. Soubki et al. (2024) introduced the first ToM resource on real human dialogues, while FaNToM (Kim et al., 2023) introduced synthetically gener-

ated conversations with information asymmetry, and NegotiationToM (Chan et al., 2024) extended evaluation to strategic interactions. However, these benchmarks typically focus on extractive or classification tasks where answers appear in localized dialogue segments. In contrast, DIAMONDs features conversations requiring multi-step reasoning over dynamic variables that evolve throughout extended exchanges, better reflecting the cognitive demands of real-world multiparty interactions.

3 DIAMONDs: Dynamic Information And Mental modeling Of Numeric Discussions

3.1 Overview: DIAMONDs is a conversation QA dataset comprising of 3786 (C, q, a) triples of multiparty conversation C, a question q to answered based on C, and its answer a. The dataset evaluates Theory of Mind (ToM) capabilities in information-rich discussions, where values of quantities (variables) affecting the answer to the question, evolve over the conversation discourse. Each conversation in DIAMONDs follows a common structure: (1) early exchanges establish the variable space and their initial states (2) subsequent interactions introduce modifications to these states (3) participant movement patterns create information asymmetries. DIAMONDs includes both omniscient (testing global understanding) and participant-centric questions (requiring reasoning from specific perspectives, including false beliefs). It also contains conversation variants with thematically relevant distractors and deliberately underspecified scenarios resulting in unanswerable questions. All questions are non-extractive, requiring multi-step numerical reasoning over the entire conversation.

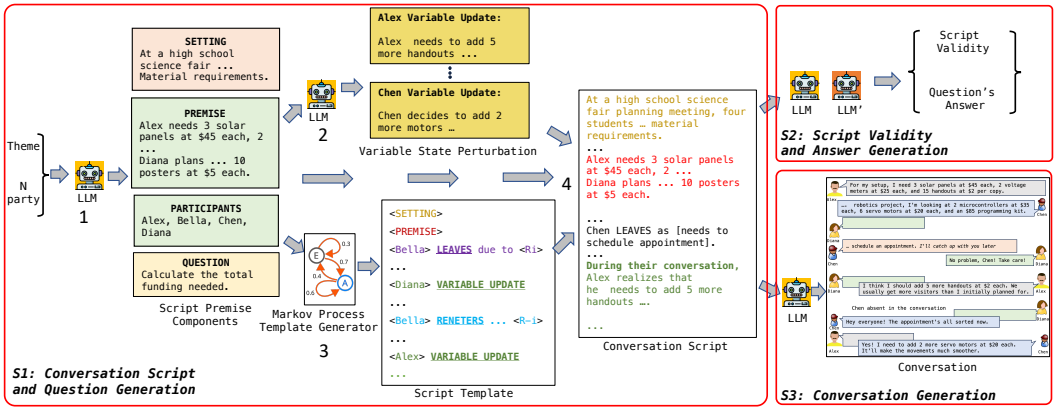

Figure 2: **Dataset Generation**: ***S1***: Script generation begins by (1) creating premise components (setting, premise, question, participants) based on a ***theme*** t and n ***participants*** (2) generating ***variable state perturbations*** that modify initial conditions, (3) producing a ***script template*** via a Markov process (4) assembling these elements into a complete ***script***. ***S2***: Script validation - evaluates logical consistency and generates verified answers. ***S3***: Conversation translation - transforms the validated ***script*** into a natural ***multi-party conversation*** while preserving all critical numerical information.

3.2 Dataset Creation We target automatic generation of conversations following works that uses LLMs (Kim et al., 2023; Xu et al., 2024) and templates (Gandhi et al., 2023) for the generation of conversations and narratives. Our primary design goals for DIAMONDs are to include problems that: (i) are non-extractive, i.e., resistant to simple localized extraction, (ii) require tracking and aggregating information across the entire conversation, (iii) have unambiguous answers enabling precise evaluation, and (iv) include information asymmetry. To this end, we want to obtain conversations with multiple thematically relevant numeric quantities in a consistent manner, include information asymmetries, and be coherent overall.

This, in itself, poses a challenging generation problem. From our initial attempts, we found that directly generating such structurally constrained conversations using few-shot prompting produced inconsistent and incoherent results. For example, LLMs frequently failed to maintain numerical consistency across long exchanges or struggled to properly implement the required information asymmetries. To address this, we leverage the strong

instruction following and refinement capabilities of LLMs to devise a 3-stage pipeline as shown in Figure 2. The key idea is to use a *script* as a basis for generating the *conversation* and it QA pairs. The *script* describes how the numerical quantities will be introduced and modified in the *conversation* and how the information asymmetries will arise. Also, generating correct QA pairs based on the script is easier and allows for effective validation.

S1: Generate conversation script and question: A conversation *script* S is a structured narrative that captures essential interactions between participants $(P_1, \cdots, P_n)$. Each script consists of two key components: (i) a ***premise*** segment establishing the initial scenario and variable states, and (ii) subsequent segments introducing ***variable state perturbations*** through participant interactions (e.g., *Chen adds two motors*), with controlled participant entry/exit events creating information asymmetry (e.g., *Chen leaves in the middle for an appointment*). To create diverse and realistic patterns of participant engagement, we model various ways participants enter, exit, and rejoin conversations. We model these dynamic participation patterns using a Markov process that generates ***script templates*** with different information access configurations. As shown in Figure 2, the process can be used to sample templates with designated slots for critical information exchanges and participant movements. A snippet of a sampled template is show below:

$$\cdots \rightarrow [P_i] \texttt{ LEAVES DUE TO } [R_m] \rightarrow \texttt{VARIABLE UPDATE BY } [P_j] \rightarrow [P_k] \texttt{ RETURNS.} \rightarrow \cdots$$
$$\cdots \rightarrow [P_k] \texttt{ LEAVES DUE TO } [R_m] \rightarrow \texttt{CASUAL TALK} \rightarrow [P_k] \texttt{ RETURNS} \rightarrow \cdots$$

As a final ***assembling*** step, these templates are then populated with relevant content and to form the complete script. The following subsections detail this process.

1. Script Premise Generation Given the number of participants n and a conversation theme t, we generate the premise components that establish the conversation context and initial variable states. This includes: (i) **setting** - A concise context description for the conversation (ii) **Script Premise** - a detailed math word problem describing the conversation involving numerical information exchanged with each of them. (iii) **question** - A question about a numeric quantity that can be derived from the information in the script premise (iv) **participants** - The list of conversation participants. We generate the script premise components using few-shot prompting with a LLM. Appendix, Figure 5 shows these components for a *science fair planning* scenario with 4 participants, with the prompt in Figure 5.

2. Variable State Perturbation Generation The numeric quantities (*variables*) of interest can change values as the conversation unfolds (e.g. the *material requirements* keep changing). We model these changes by generating *perturbations* that alter the values of the *variables* associated with each participant. The perturbations are added only for *variables* that are relevant to answering the question. For instance, in Figure 1, perturbation for *Chen* (*he needs to add* 2 *more motors*), results in increasing the number of motors he requires from $6 \rightarrow 8$. This change in turn increases the total cost for Chen and the overall *funding needed for the project*, changing the answer to the question. To maintain logical consistency and manage complexity, we implement two key design constraints: (i) we only perturb independent variables from the premise, avoiding modifications to variables constrained by multiple relationships that could create logical inconsistencies (ii) we ensure each *perturbations* is expressed independently rather than in terms of other *perturbations*, removing ordering dependencies when arranging them in the conversation. We generate these *variable state perturbations* using LLM with few-shot prompting (prompt shown in Appendix, Figure 12).

3. Script Template Generation We generate probabilistic script *templates* from a Markov process (Figure 3) that models participant movement and information flow. This *template* contains slots for *premise* information, *variable-state perturbations*, casual dialogue cues, and participant entry/exit indicators. The structure ensures discourse coherence while creating controlled information asymmetry, allowing us to precisely track which participant has access to what information throughout the conversation. As shown in Figure 3, the Markov process begins with *premise* slot for the *script premise* with all participants intially present. It then systematically models participant movement dynamics by: (1) randomly selecting participants to leave with plausible reasons, (2) determining which participant shares new information (*variable state perturbations*) next, and (3) managing re-entries of absent participants.

To maintain naturalistic conversation flow, we reduce the probability of repeated exits by the same participant and include casual conversation segments between major events. Figure 7 shows a script template for 4 participants, with full details of the template generation process in Appendix A.6.

4. Assembling the Script The final conversation *script* is assembled by combining the *template slots* with information in the *script premise components* and *variable state perturbations*. Figure 8 shows the complete *script* after filling the template slots (Figure 7) with information in *Script Premise Components* (Figure 5) and its *variable state perturbations* (Figure 6).

Generating Underspecified and Distractor Script Variants: To evaluate model robustness, we create two controlled script variants:

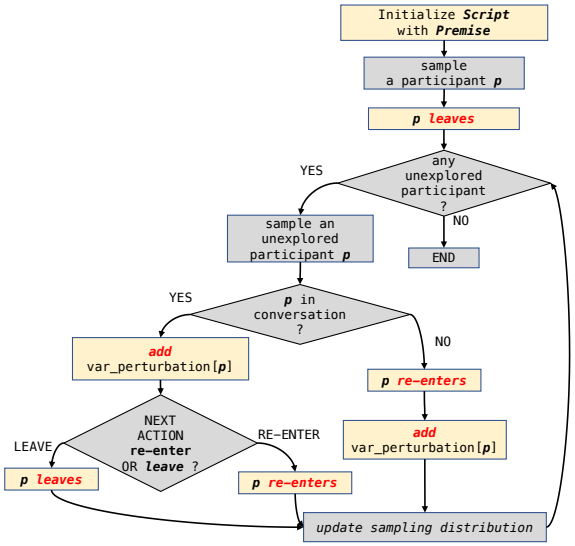

Figure 3: A simplified Markov process generating script *template* with slots for *script premise*, *variable state perturbations*, and participant movements.

a. Distractor Variants: We augment base scripts with thematically relevant but question-irrelevant information by introducing new quantities in the *premise* and their state perturbations in one of the subsequence segment. These additions maintain conversational coherence while being carefully having no impact on the answer to the question, testing models' ability to filter relevant from irrelevant information.

b. Underspecified Variants: We create underspecified scripts by systematically removing critical information needed to answer questions. This is done by either (1) eliminating key details from the premise or (2) making variable state perturbations ambiguous. These scripts test a model's ability to recognize when information is insufficient rather than forcing an incorrect answer. The details for both are described in Appendix, subsection A.10.3

S2: Script Validation and Answer Generation

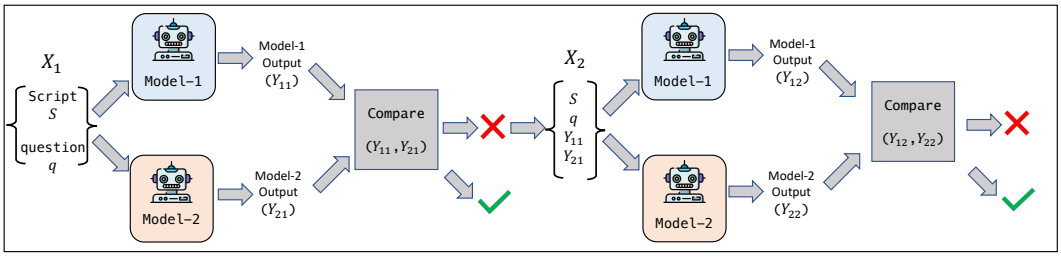

Figure 4: **Script validation and solution generation**: Two LLMs independently evaluate inputs and generate solutions in the first round. If their solutions match, the solution is accepted. Otherwise, a second round incorporates both first-round outputs for refinement. Agreement in the second round leads to acceptance, otherwise the input is rejected.

We design a two-step reflect-refine inter-model consistency (Chowdhury et al., 2024; Amiri-Margavi et al., 2024; Zhang et al., 2024) based approach to generate solution for the question based on script. As shown in Figure 4, two independent LLMs generate solutions for each script-question pair. Each solution includes evaluations of script logical consistency, question answerability, step-by-step reasoning, and a final numeric answer. If both models produce equivalent outputs on the first attempt, we accept the script and solution. When disagreements occur, we conduct a second evaluation round where both models review the script alongside both the first-round solutions. Agreement in this second round leads to acceptance, while continued disagreement results in the script being discarded.

S3: Conversation Generation We translate the script into conversation with few-shot prompting and 2-step self-reflection and refinement (Madaan et al., 2024). The task instruction has two key focuses: (i) conversation completely covers all information in the script (ii) Avoiding any extra information not in the script that might change the answer to the question. The prompt also includes additional guidance to maintain conversational coherence and naturalness. The prompt used for this step is shown in the Appendix, Figure 16.

Using this process we created, DIAMONDs, which contains 584 conversations with 4 to 7 participants, with 3786 unique conversation-question pairs covering 49 conversation themes.

3.1 Data Quality Assessment: We conducted rigorous quality evaluations on our dataset of script-conversation-question-answer tuples $\{(S, q, a, C)\}$. Our assessment focused on two critical aspects: (i) script-answer validity of (S, q, a) (ii) script-conversation alignment between (S, C). We randomly selected 66 samples for evaluation by 5 expert annotators (STEM graduate students), with each instance assessed by 2 evaluators.
(i) Script-based Answer Validity To determine the validity of a $\{(S, q, a)\}$ instance, the evaluators assessed: (a) The script S was logically consistent (b) The question q was answerable based on S (c) The solution a was correct for q in context of S. Of evaluated instances, 91.1% of (S, q, a) tuples were assessed as valid.
(ii) Script-Conversation Alignment Evaluators evaluated the alignment of a conversation C with its script S by verifying that: (i) All question-relevant script details were preserved in the conversation (a) Conversation segments maintained the discourse order established in the script (b) No additional calculations beyond those in the script were introduced (c) Participant movements and information states remained consistent Of the evaluated (S, C)instances, 100% of the conversations were correctly aligned with their scripts.

Based on these evaluations, we estimate the validity of our curated dataset to be $\sim$ 91%.

4 Benchmarking Conversation Comprehension with DIAMONDs

Conversation Comprehension Tasks: We introduce a QA task that requires Theory-of-Mind (ToM) abilities for conversation comprehension. This involves tracking, and reasoning with multiple dynamic numeric variables over a long multi-party conversation. Our task covers two settings: (i) **Omniscient Questions**: These require answering based on the entire conversation, reflecting complete knowledge of all exchanged information. (ii) **Participant-Centric Questions**: These require answering from a specific participant's perspective, using only information exchanged in the conversation that participant had access to. Here participants may miss critical updates relevant to the question due to being absent from the conversation at that time. This leads to a **false belief** situation where the answer to the question from their perspective would be different from the true answer to the question.

Evaluation Metrics: Since the task requires numeric computations, we allow a small tolerance for minor computational or rounding errors. For base and distractor conversations, we consider a model's answer correct if its final numerical response is within 2% of the true value, accounting for potential intermediate rounding errors. For underspecified conversations, a response is deemed correct if it successfully identifies the question as unanswerable. To ensure uniform comparison, we evaluate models on a common subset of data where all models provided valid responses. This approach was necessary because models occasionally failed to generate parseable responses due to invalid JSON formatting or other runtime API errors . Results on the full set of valid responses are in Appendix A.7. Models show same relative performance, and the difference on overall accuracy (complete set vs common subset) ranges from $\sim$ 0.4% for `gpt-4o` to $\sim$ 2.5% for `Llama-90B`.

Models: Our goal is to benchmark the inherent ToM capabilities of large language models and thus do not consider fine-tuning models on this task. Therefore, we benchmark three state-of-art (i) closed models: `Claude-3-5-sonnet-20241022`, `gpt-4o 2024-08-06` (ii) open models: `llama3-1 405B-instruct` , `llama3-2 90B-instruct`, and `llama3-2 11B-instruct` on the task with zero-shot prompting with chain-of-thought and 5-rounds of self-reflect and self-refinement (Madaan et al., 2024). Given the objective nature of the task, we set the generation temperature to 0 for benchmarking. Dataset generation details are mentioned

in Appendix, subsection A.8. All the prompts used for data generation are shown in subsection A.10

5 Results

5.1 Participant-Centric Questions are harder than Omniscient Questions

MODEL ACCURACY (%) ON BASE/REGULAR CONVERSATIONs			
Model	**Omniscient**	**Participant Centric**	**Overall**
claude-3-5-sonnet-20241022	80.0	55.1%	59.6%
gpt-4o-2024-08-06	73.8	39.2%	45.6%
llama3-1-405b-instruct	69.2%	37.9%	43.6%
llama3-2-90b-instruct	50.0%	24.3%	29.0%
llama3-2-11b-instruct	11.5%	9.5%	9.8%

Table 1: **Omniscient** is the % of correctly answered omniscient questions, **Participant Centric** is the % of correctly answered participant-centric ones, and **Overall** is the accuracy across all the question type.

Across all models, there is a consistent performance gap between omniscient and participant-centric questions. For base conversations (Table 1), Claude-3.5 has the highest performance with 80.0% accuracy on omniscient questions but drops to 55.1% on participant-centric ones. gpt-4o shows a similar pattern with 73.8% and 39.2% for omniscient and participant-centric respectively. The llama3 models follow similar trend trend, with performance scaling with model size - from the 405B parameter version (69.2% omniscient, 37.9%participant-centric) down to the 11B version (11.5% omniscient, 9.5% participant-centric). This substantial performance gap can be attributed to the added complexity of modeling information access in situations with information asymmetry, compared to using the entire context. These models struggle to ignore contextually present but inaccessible information, highlighting the limitations of their ToM capabilities for multi-party conversations.

5.2 Participant-Centric Performance: True vs False Belief

MODEL ACCURACY (%) ON PARTICIPANT-CENTRIC QUESTIONS IN BASE CONVERSATION			
Model	**True Belief**	**False Belief**	**Overall**
claude-3-5-sonnet-20241022	78.4	28.1	55.1
gpt-4o-2024-08-06	54.8	21.0	39.2
llama3-1-405b-instruct	51.6	21.7	37.9
llama3-2-90b-instruct	31.2	16.1	24.3
llama3-2-11b-instruct	10.5	8.6	9.5

Table 2: Model Performance on participant-centric questions in DIAMONDs's base conversations. **True Belief** is the accuracy for cases where participants have access to all the question-relevant information, **False Belief** represents cases where participants miss critical updates, resulting in outdated or incorrect knowledge states. **Participant Centric** is the overall performance across both categories.

Table 2 breaks down the model performance on participant-centric question. We distinguish between True Belief scenarios, where participants have access to the correct information needed to answer questions accurately, and False Belief cases, where participants miss critical information updates, leading to a mismatch between their knowledge state and reality. This analysis reveals two critical insights: *First*, all models show dramatic performance drop when handling false beliefs situations. Claude-3.5 achieves 78.4% accuracy with true beliefs but only 28.1% with false beliefs. Similar patterns appear across all models, with gpt-4o dropping 33.8% points and llama3-405B declining 29.9%. *Second*, even for true belief participant-centric questions, most models perform worse than on omniscient questions (Table 1). Since participants with true belief have access to correct information, performance should theoretically be comparable to omniscient question performance. While Claude-3.5 shows only a small gap of 1.6%, other models exhibit larger differences: gpt-4o drops 19%

and llama3-405B falls 17.6% compared to omniscient questions. These findings highlight models' limitations in Theory of Mind capabilities as they struggle noticeably when reasoning from perspectives with incomplete information and show degraded performance even when merely adopting a participant's viewpoint rather than an omniscient one.

5.3 Impact of Distractors on Model Performance

MODEL ACCURACY (%) ON CONVERSATIONS				
	DISTRACTOR			UNDERSPECIFIED
Model	**Omniscient**	**Participant Centric**	**Overall**	**Omniscient**
claude-3-5-sonnet-20241022	78.8	50.5	55.6	63.8
gpt-4o-2024-08-06	65.4	34.8	40.3	54.3
llama3-1-405b-instruct	67.0	35.5	41.2	73.9
llama3-2-90b-instruct	44.4	21.7	25.7	44.4
llama3-2-11b-instruct	9.1	8.1	8.3	14.6

Table 3: Model performance on (i) conversations with distractor (ii) underspecified conversations with unanswerable questions. For distractor conversations we compute accuracy for Omniscient, participant-centric and over all questions. For unswerable questions we only compute % unanswerability correctly identified. We compute this for only Omniscient questions as explained in section 5.4

Our analysis of conversations with distractors (Table 3) reveals a consistent drop in model performance across all models, especially for participant-centric type questions. For omniscient questions, the impact is relatively modest. Claude-3.5 maintains strong performance with only 1.2% drop in performance compared to that on base conversation. Similarly, Llama3-405B also shows only 2.2% drop in performance. GPT-4o experiences a more noticeable performance drop of $\sim$ 8% to 65.4% due to distractors. The challenge becomes more pronounced with participant-centric questions, where models must simultaneously filter distractors while maintaining participant-specific information models. Claude-3.5 achieves 50.5% accuracy on these more complex scenarios, while GPT-4o and Llama3-405B attain 34.8% and 35.5% respectively. These results highlight a critical limitation that current language models struggle to effectively filter out irrelevant information, especially when required to maintain a specific participant centric view of the conversation.

5.4 Handling Unanswerable Situations

For underspecified conversations, we calculate the percentage of questions where models correctly identified the unanswerability. It's important to note that for underspecified conversations, if the participant-centric question is unanswerable, its omniscient variant will also be unanswerable. Therefore, a model could potentially arrive at the correct answer (identifying unanswerability) for a participant-centric question by using omniscient reasoning. So we only report performance on Omniscient questions. As shown in Table 3, models demonstrate varying abilities to recognize information gaps. Claude-3.5 correctly identifies 63.8% of unanswerable questions, while Llama3-405B performs surprisingly well at 73.9%. GPT-4o achieves 54.3% accuracy, with smaller models showing significantly reduced capabilities. This demonstrates that even advanced models struggle to consistently recognize when they lack sufficient information to answer a question and are biased to generate answer even when it is not possible to do so.

6 Conclusion

We present DIAMONDs, a novel multiparty conversation comprehension benchmark that tests for a subset of Theory of Mind (ToM) capabilities. Our benchmark specifically targets tracking dynamic information states, managing knowledge asymmetries across multiple participants, and integrating this understanding with numerical reasoning—challenges that mirror real-world social interactions. Our evaluation of state-of-the-art language models reveals significant gaps in these types of ToM capabilities, where even strong models like Claude-3.5 struggle with participant-centric reasoning, and fare even worse in false belief settings. Additionally, DIAMONDs introduces a scalable approach for constructing complex,

information-rich conversations with controlled information asymmetries. By leveraging a multi-stage generation process that combines LLM capabilities with structured templates sampled from a Markov process, we create conversations that systematically test specific aspects of ToM reasoning while maintaining coherence and realism. This methodology offers a blueprint for developing increasingly sophisticated conversation-based evaluation resources. These findings highlight a fundamental challenge: current language models struggle to maintain accurate mental models of different participants' knowledge states throughout extended conversations. DIAMONDs thus provides valuable insights into the current limitations of language models while establishing clear directions for improving ToM capabilities essential for effective human-AI interaction in multi-party settings.

7 Acknowledgement

This work is supported in part by DARPA for the KAIROS program under agreement number FA8750-19-2-1003 and by an Amazon Research Award. We also thank Prof. Katrin E. Erk, Prof. H. Andrew Schwartz, Prof. Jordan Kodner, and Prof. Owen Rambow for providing insightful feedback that was essential for improving this work.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Alireza Amiri-Margavi, Iman Jebellat, Ehsan Jebellat, and Seyed Pouyan Mousavi Davoudi. Enhancing answer reliability through inter-model consensus of large language models. 2024. URL https://api.semanticscholar.org/CorpusID:274281085.

Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. Open-domain question answering goes conversational via question rewriting. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 520–534, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.44. URL https://aclanthology.org/2021.naacl-main.44.

Chunkit Chan, Cheng Jiayang, Yauwai Yim, Zheye Deng, Wei Fan, Haoran Li, Xin Liu, Hongming Zhang, Weiqi Wang, and Yangqiu Song. Negotiationtom: A benchmark for stress-testing machine theory of mind on negotiation surrounding. In *Conference on Empirical Methods in Natural Language Processing*, 2024. URL https://api.semanticscholar.org/CorpusID:269293744.

Emile Chapuis, Pierre Colombo, Matteo Manica, Matthieu Labeau, and Chloé Clavel. Hierarchical pre-training for sequence labelling in spoken dialog. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 2636–2648, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.239. URL https://aclanthology.org/2020.findings-emnlp.239.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. DialogSum: A real-life scenario dialogue summarization dataset. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 5062–5074, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.449. URL https://aclanthology.org/2021.findings-acl.449.

Zhuang Chen, Jincenzi Wu, Jinfeng Zhou, Bosi Wen, Guanqun Bi, Gongyao Jiang, Yaru Cao, Mengting Hu, Yunghwei Lai, Zexuan Xiong, and Minlie Huang. ToMBench: Benchmarking theory of mind in large language models. In Lun-Wei Ku, Andre Martins, and Vivek

Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15959–15983, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.847. URL https://aclanthology.org/2024.acl-long.847/.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036*, 2018.

Saurav Chowdhury, Lipika Dey, and Suyog Joshi. Cross examine: An ensemble-based approach to leverage large language models for legal text analytics. In Nikolaos Aletras, Ilias Chalkidis, Leslie Barrett, Cătălina Goanță, Daniel Preoțiuc-Pietro, and Gerasimos Spanakis (eds.), *Proceedings of the Natural Legal Language Processing Workshop 2024*, pp. 194–204, Miami, FL, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.nllp-1.16. URL https://aclanthology.org/2024.nllp-1.16.

Herbert H Clark. *Using language*. Cambridge university press, 1996.

Marc De Rosnay and Claire Hughes. Conversation and theory of mind: Do children talk their way to socio-cognitive understanding? *British journal of developmental psychology*, 24 (1):7–37, 2006.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Shutong Feng, Nurul Lubis, Christian Geishauser, Hsien-chin Lin, Michael Heck, Carel van Niekerk, and Milica Gasic. EmoWOZ: A large-scale corpus and labelling scheme for emotion recognition in task-oriented dialogue systems. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Jan Odijk, and Stelios Piperidis (eds.), *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pp. 4096–4113, Marseille, France, June 2022. European Language Resources Association. URL https://aclanthology.org/2022.lrec-1.436.

Uta Frith. Autism and theory of mind in everyday life. *Social development*, 3(2):108–124, 1994.

Kanishk Gandhi, Jan-Philipp Fränken, Tobias Gerstenberg, and Noah Goodman. Understanding social reasoning in language models with language models. *Advances in Neural Information Processing Systems*, 36:13518–13529, 2023.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In Lu Wang, Jackie Chi Kit Cheung, Giuseppe Carenini, and Fei Liu (eds.), *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pp. 70–79, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5409. URL https://aclanthology.org/D19-5409.

Ting Han, Ximing Liu, Ryuichi Takanabu, Yixin Lian, Chongxuan Huang, Dazhen Wan, Wei Peng, and Minlie Huang. Multiwoz 2.3: A multi-domain task-oriented dialogue dataset enhanced with annotation corrections and co-reference annotation. In *Natural Language Processing and Chinese Computing: 10th CCF International Conference, NLPCC 2021, Qingdao, China, October 13–17, 2021, Proceedings, Part II*, pp. 206–218, Berlin, Heidelberg, 2021. Springer-Verlag. ISBN 978-3-030-88482-6. doi: 10.1007/978-3-030-88483-3_16. URL https://doi.org/10.1007/978-3-030-88483-3_16.

Yebowen Hu, Timothy Ganter, Hanieh Deilamsalehy, Franck Dernoncourt, Hassan Foroosh, and Fei Liu. MeetingBank: A benchmark dataset for meeting summarization. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16409–16423, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.906. URL https://aclanthology.org/2023.acl-long.906.

Hyunwoo Kim, Melanie Sclar, Xuhui Zhou, Ronan Bras, Gunhee Kim, Yejin Choi, and Maarten Sap. FANToM: A benchmark for stress-testing machine theory of mind in interactions. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14397–14413, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.890. URL https://aclanthology.org/2023.emnlp-main.890.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. An evaluation dataset for intent classification and out-of-scope prediction. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1311–1316, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1131. URL https://aclanthology.org/D19-1131.

Matthew Le, Y-Lan Boureau, and Maximilian Nickel. Revisiting the evaluation of theory of mind through question answering. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5872–5877, Hong Kong, China, November 2019a. Association for Computational Linguistics. doi: 10.18653/v1/D19-1598. URL https://aclanthology.org/D19-1598/.

Matthew Le, Y-Lan Boureau, and Maximilian Nickel. Revisiting the evaluation of theory of mind through question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5872–5877, 2019b.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36, 2024.

Martin J Pickering and Simon Garrod. Toward a mechanistic psychology of dialogue. *Behavioral and brain sciences*, 27(2):169–190, 2004.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 527–536, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1050. URL https://aclanthology.org/P19-1050.

Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W. Bruce Croft, and Mohit Iyyer. Open-retrieval conversational question answering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, pp. 539–548, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450380164. doi: 10.1145/3397271.3401110. URL https://doi.org/10.1145/3397271.3401110.

Siva Reddy, Danqi Chen, and Christopher D Manning. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 2019.

Marzieh Saeidi, Max Bartolo, Patrick Lewis, Sameer Singh, Tim Rocktäschel, Mike Sheldon, Guillaume Bouchard, and Sebastian Riedel. Interpretation of natural language rules in conversational machine reading. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2087–2097, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1233. URL https://aclanthology.org/D18-1233.

Maarten Sap, Ronan Le Bras, Daniel Fried, and Yejin Choi. Neural theory-of-mind? on the limits of social intelligence in large LMs. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3762–3780, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.248. URL https://aclanthology.org/2022.emnlp-main.248/.

Adil Soubki, John Murzaku, Arash Yousefi Jordehi, Peter Zeng, Magdalena Markowska, Seyed Abolghasem Mirroshandel, and Owen Rambow. Views are my own, but also yours: Benchmarking theory of mind using common ground. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14815–14823, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.880. URL https://aclanthology.org/2024.findings-acl.880/.

Yufan Wu, Yinghui He, Yilin Jia, Rada Mihalcea, Yulong Chen, and Naihao Deng. Hi-ToM: A benchmark for evaluating higher-order theory of mind reasoning in large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10691–10706, Singapore, December 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.717. URL https://aclanthology.org/2023.findings-emnlp.717/.

Yufan Wu, Yinghui He, Yilin Jia, Rada Mihalcea, Yulong Chen, and Naihao Deng. Hi-tom: A benchmark for evaluating higher-order theory of mind reasoning in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023b. URL https://aclanthology.org/2023.findings-emnlp.717.

Hainiu Xu, Runcong Zhao, Lixing Zhu, Jinhua Du, and Yulan He. OpenToM: A comprehensive benchmark for evaluating theory-of-mind reasoning capabilities of large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8593–8623, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.466. URL https://aclanthology.org/2024.acl-long.466/.

Fanghua Ye, Jarana Manotumruksa, and Emine Yilmaz. MultiWOZ 2.4: A multi-domain task-oriented dialogue dataset with essential annotation corrections to improve state tracking evaluation. In Oliver Lemon, Dilek Hakkani-Tur, Junyi Jessy Li, Arash Ashrafzadeh, Daniel Hernández Garcia, Malihe Alikhani, David Vandyke, and Ondřej Dušek (eds.), *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 351–360, Edinburgh, UK, September 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.sigdial-1.34. URL https://aclanthology.org/2022.sigdial-1.34.

Wenqi Zhang, Yongliang Shen, Linjuan Wu, Qiuying Peng, Jun Wang, Yueting Zhuang, and Weiming Lu. Self-contrast: Better reflection through inconsistent solving perspectives. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3602–3622, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.197. URL https://aclanthology.org/2024.acl-long.197.

Chenguang Zhu, Yang Liu, Jie Mei, and Michael Zeng. MediaSum: A large-scale media interview dataset for dialogue summarization. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5927–5934, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.474. URL https://aclanthology.org/2021.naacl-main.474.

A Appendix

A.1 Script Premise Components

setting:
At a high school science fair planning meeting, four students – Alex, Bella, Chen, and Diana – are discussing their project budgets and material requirements.

question:
Calculate the total amount of funding needed for all four science fair projects combined.

participants:
['Alex', 'Bella', 'Chen', 'Diana']

script premise:
Alex is working on a renewable energy project and needs 3 solar panels at $45 each, 2 voltage meters costing $25 each, and plans to create 15 informational handouts at $2 per copy. Bella's chemistry experiment requires 8 different chemicals costing $12 each, 3 sets of safety equipment at $30 per set, and 5 specialized glass beakers at $15 each. Chen's robotics project needs 2 microcontrollers at $35 each, 6 servo motors costing $20 each, and a programming kit for $85. He also needs to print 20 pages of documentation at $0.50 per page. Diana's environmental science project requires 4 soil testing kits at $28 each, 6 plant specimens costing $8 each, and 2 digital thermometers at $22 each. She plans to create 10 presentation posters at $5 each.

Figure 5: **Script Premise** Components for a Travel Agency Meeting with 4 participants.

A.2 Variable State Perturbation

Alex:
During their conversation, Alex realizes he needs to add 5 more informational handouts at the same cost per copy to reach more visitors.

Bella:
During their conversation, Bella discovers she needs one additional set of safety equipment at the same cost per set.

Chen:
During their conversation, Chen decides to add 2 more servo motors at the same cost per unit to improve his robot's functionality.

Diana:
During their conversation, Diana decides to reduce her poster count by 2 as she found more efficient ways to present her information.

Figure 6: **Variable State Perturbation** information for each participant, that perturbs a variable state established in the script premise (Figure 5)

A.3 Template Generation

```
<SETTING>
<SCRIPT PREMISE>
Chen leaves the conversation because of - "[need to schedule
another appointment]"

Some casual conversation goes on between ['Alex', 'Bella',
'Diana'].
<Alex> var change
Diana leaves because of reason "[need to take care of some
personal matters]"

Some casual conversation goes on between ['Alex', 'Bella'].
Chen re-enters, after leaving earlier due to "[need to schedule
another appointment]"
Some casual conversation goes on between ['Alex', 'Bella',
'Chen']. (They do not talk about the specific details of their
past conversations that Chen missed.)
<Chen> var change

Some casual conversation goes on between ['Alex', 'Bella',
'Chen'].
<Bella> var change
Chen leaves because of reason "[have unexpected visitor]"

Some casual conversation goes on between ['Alex', 'Bella'].
Diana re-enters, after leaving earlier due to "[need to take
care of some personal matters]"
Some casual conversation goes on between ['Alex', 'Bella',
'Diana']. (They do not talk about the specific details of their
past conversations that Diana missed.)
<Diana> var change
```

Figure 7: **Script Template** with participant movement (red and green colored) and slots (`<name> var change`) for pariable state perturbation information to be exchanged during the conversation.

A.4 Assembling the Script

```
At a high school science fair planning meeting, four students - Alex, Bella, Chen, and
Diana - are discussing their project budgets and material requirements. Alex is working on
a renewable energy project and needs 3 solar panels at $45 each, 2 voltage meters costing
$25 each, and plans to create 15 informational handouts at $2 per copy. Bella's chemistry
experiment requires 8 different chemicals costing $12 each, 3 sets of safety equipment at
$30 per set, and 5 specialized glass beakers at $15 each. Chen's robotics project needs 2
microcontrollers at $35 each, 6 servo motors costing $20 each, and a programming kit for
$85. He also needs to print 20 pages of documentation at $0.50 per page. Diana's
environmental science project requires 4 soil testing kits at $28 each, 6 plant specimens
costing $8 each, and 2 digital thermometers at $22 each. She plans to create 10
presentation posters at $5 each.
Chen leaves the conversation because of - "[need to schedule another appointment]"

Some casual conversation goes on between ['Alex', 'Bella', 'Diana'].
During their conversation, Alex realizes he needs to add 5 more informational handouts at
the same cost per copy to reach more visitors.
Diana leaves because of reason "[need to take care of some personal matters]"

Some casual conversation goes on between ['Alex', 'Bella'].
Chen re-enters, after leaving earlier due to "[need to schedule another appointment]"
Some casual conversation goes on between ['Alex', 'Bella', 'Chen']. (They do not talk
about the specific details of their past conversations that Chen missed.)
During their conversation, Chen decides to add 2 more servo motors at the same cost per
unit to improve his robot's functionality.

Some casual conversation goes on between ['Alex', 'Bella', 'Chen'].
During their conversation, Bella discovers she needs one additional set of safety
equipment at the same cost per set.
Chen leaves because of reason "[have unexpected visitor]"

Some casual conversation goes on between ['Alex', 'Bella'].
Diana re-enters, after leaving earlier due to "[need to take care of some personal
matters]"
Some casual conversation goes on between ['Alex', 'Bella', 'Diana']. (They do not talk
about the specific details of their past conversations that Diana missed.)
During their conversation, Diana decides to reduce her poster count by 2 as she found more
efficient ways to present her information.
```

Figure 8: **Assmebling the script** by filling in the template slot (Figure 7) with relevant (i) setting, (ii) premise (Figure 5 and (iii) variable state perturbations (Figure 6).

A.5 Conversation Generation from Script

```
Alex : Thanks everyone for meeting up to discuss our science fair projects! I'm really excited
about my renewable energy demonstration.
Bella : Me too! Let's go through our materials lists to make sure we have everything covered.
Alex : For my setup, I need 3 solar panels at $45 each, 2 voltage meters at $25 each, and 15
handouts at $2 per copy.
Chen : Sounds comprehensive! For my robotics project, I'm looking at 2 microcontrollers at $35
each, 6 servo motors at $20 each, and an $85 programming kit.
Diana : That's quite high-tech, Chen! I'm keeping mine more earth-focused.
Chen : What's your plan, Diana?
Diana : I need 4 soil testing kits at $28 each, 6 plant specimens at $8 each, and 2 digital
thermometers at $22 each. Plus 10 posters at $5 each.
Bella : While we're sharing, my chemistry experiment needs 8 chemicals at $12 each, 3 safety
equipment sets at $30 each, and 5 special beakers at $15 each.
Chen : Oh! I almost forgot - I also need to print 20 pages of documentation at 50 cents per page.
Alex : Looks like we've all put a lot of thought into this!
Chen : Sorry everyone, I just remembered I need to schedule an appointment. I'll catch up with you
later!
Bella : No problem, Chen! Take care!
Alex : Have you seen the previous years' winning projects? They're really inspiring.
Bella : Yes! The presentation quality keeps getting better each year.
Diana : Speaking of presentation, that reminds me...
Alex : You know, looking at those projects made me realize something about my handouts.
Bella : What's on your mind, Alex?
Alex : I think I should add 5 more handouts at $2 each. We usually get more visitors than I
initially planned for.
Diana : That's smart thinking! Oh - I just remembered I need to take care of some personal
matters. I'll catch you all later!
Bella : See you later, Diana! Take care!
Alex : The science fair venue is perfect this year, don't you think?
Bella : Absolutely! The lighting is much better for demonstrations.
Chen : Hey everyone! The appointment's all sorted now.
Alex : Chen! Perfect timing - we were just discussing the venue.
Bella : The layout should work great for your robot demonstration, Chen!
Chen : Actually, speaking of the robot, I've been thinking...
Alex : Made some design improvements?
Chen : Yes! I need to add 2 more servo motors at $20 each. It'll make the movements much smoother.
Bella : That sounds like it'll make a big difference!
Alex : The judging criteria this year seems pretty comprehensive.
Chen : Safety is a big focus, from what I've heard.
Bella : Oh, that reminds me! After reviewing the safety guidelines...
Alex : Everything okay with your chemistry setup, Bella?
Bella : I need to add another set of safety equipment at $30. Better to be over-prepared than
under!
Chen : Always better to- Oh! Someone's at my door unexpectedly. Got to run!
Alex : No problem, catch you later!
Alex : The weather should be perfect for the science fair weekend.
Bella : Perfect for getting all our equipment in without rain!
Diana : Hi everyone! Sorry about stepping out earlier.
Alex : Diana! Welcome back! We were just talking about the weather for the fair.
Bella : The forecast looks great for your plant specimens!
Diana : That's good to hear! Actually, I've been rethinking my presentation approach...
Alex : Oh? What's changed?
Diana : I found some better ways to present my data, so I'll reduce my poster count by 2. Still at
$5 each, but fewer needed.
Bella : Sometimes less is more with presentations!
```

Figure 9: **Conversation from Script:** Final conversation based on the conversation script (Figure 5).

A.6 Script Template Generation

As shown in Figure 10, the generation follows a Markov Chain process that creates dynamic script templates based on the number of participants, and a set of leaving reasoning mentioned in Kim et al. (2023). The process begins by establishing a premise slot that sets the foundation for the conversation. Initially, all participants have an equal probability of leaving the conversation. The generation unfolds in several steps:

1. First, a participant is randomly selected to leave the conversation, with their departure reason chosen from a predefined list of plausible excuses.

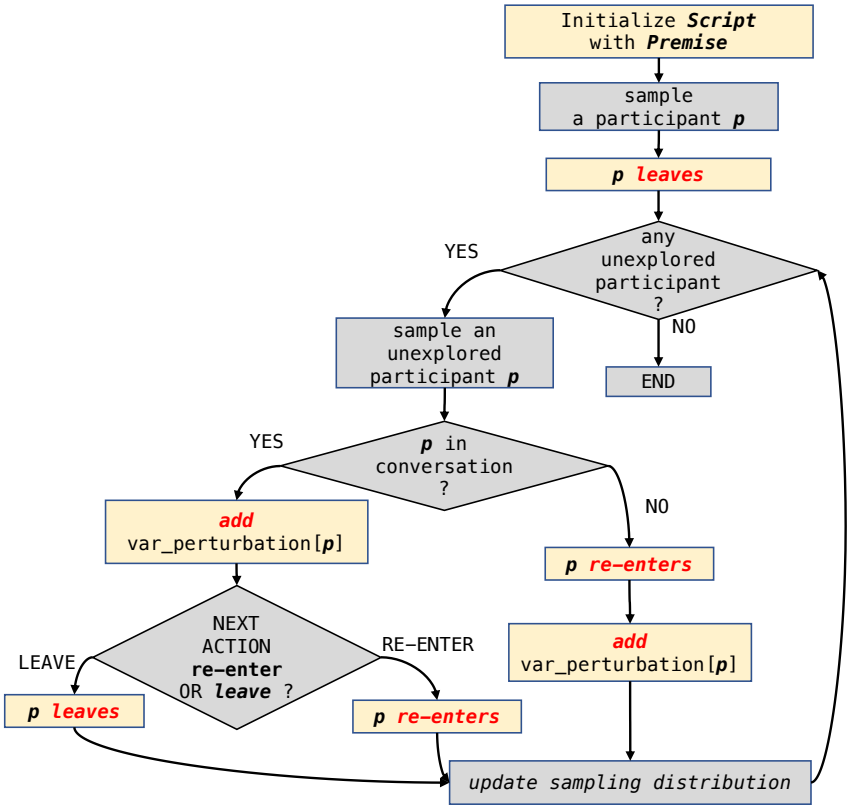

Figure 10: Markov Chain process for generating the discourse of the script which will be translated into a conversation for the data in DIAMONDs.

2. Next, the generator selects a participant to share information (variable state perturbation) from those who haven't yet contributed to the conversation. This can happen in two ways:
   - If the selected participant is currently present in the conversation, they directly share their information
   - If the selected participant is absent, they first re-enter the conversation before sharing their information
3. After each information sharing event, the generator randomly decides between two possible actions:
   - Having another participant leave the conversation
   - Having a previously departed participant re-enter the conversation

To make the conversation flow more natural, the probability of a participant leaving again is reduced by 0.75 each time they exit the conversation. This prevents unrealistic patterns of repeated exits and entries by the same participant. Between these major events (leaving, re-entering, and information sharing), the generator includes segments of casual conversation among the current participants. When a participant re-enters, the generator explicitly notes that the others don't discuss the specific details of conversations that occurred during their absence.

A.7 Benchmarking Models on their Complete Set of Data with Valid Prediction

MODEL PERFORMANCE ON BASE CONVERSATIONS (FULL DATA)			
Model	**Omniscient (%)**	**Participant Centric (%)**	**Overall (%)**
claude-3-5-sonnet-20241022	82.6	57.4	81.8
gpt-4o-2024-08-06	69.4	40.9	45.95
llama3-1-405b-instruct	66.0	36.0	41.2
llama3-2-90b-instruct	48.5	23.7	27.94
llama3-2-11b-instruct	9.8	8.7	8.9

Table 4: Model accuracy (%) on regular/base conversations without any distractor. Numbers reported are for models' performance on their full set of valid, parsable predictions.

MODEL ACCURACY (%) ON CONVERSATIONS (FULL DATA)				
	DISTRACTOR			**UNDERSPECIFIED**
Model	**Omniscient**	**Participant Centric**	**Overall**	**Omniscient**
claude-3-5-sonnet-20241022	78.1	55.7	24.	58.1
gpt-4o-2024-08-06	63.7	35.6	40.4	44.6
llama3-1-405b-instruct	65.0	35.3	40.6	70.3
llama3-2-90b-instruct	43.8	22.1	25.8	44.3
llama3-2-11b-instruct	9.1	8.5	8.6	13.0

Table 5: Model accuracy (%) on conversations with distractor and underspecified conversations where questions are unanswerable. The distractor conversations we compute accuracy for Omniscient, participant-centric and over all questions. For unswerable questions we only compute % unanswerable

We report models performance in Table 1, 3 on a common subset of data where all models provided valid responses. Here we report the models' performance based on their full set of valid, parsable predictions across the entire dataset. Table 4, 5 reports the model performance on regular/base, distractor and unanswerable conversation in DIAMONDs.

A.8 Dataset Generation

Our complete dataset generation pipeline uses LLMs in a multi-stage fashion to generate the conversation, question, answer (C, q, a). Given the mathematically nuanced and constrained nature of our dataset, we use the latest state of art models `claude-3-5 sonnet 20241022`, and `gpt-4o 2024-08-06` with few-shot prompting for generating the required components of the dataset. For translating the script into their corresponding conversation, we use `claude-3-5-sonnet-20241022` with 1-shot prompting as we observed it to generate more more natural reading and longer conversations compared to `gpt-4o-2024-08-06`. However, to show that the relative performance trend of a model is not result of choice of model for translating script to conversation, we also generate data that uses `gpt-4o-2024-08-06` as script to conversation translation model and report result on it in the Appendix A.9.

A.9 Benchmarking on Data with gpt-4o as Script to Conversation Translator

In this section, we report the model performance on question for DIAMONDs conversations,

MODEL PERFORMANCE ON BASE CONVERSATIONS GPT-4o TRANSLATED CONVERSATION			
Model	**Omniscient (%)**	**Participant Centric (%)**	**Overall (%)**
claude-3-5-sonnet-20241022	66.7	43.0	47.1
gpt-4o-2024-08-06	57.1	38.0	41.3
llama3-1-405b-instruct	61.9	33.0	38.0
llama3-2-90b-instruct	47.6	27.0	30.6
llama3-2-11b-instruct	14.3	14.0	14.0

Table 6: Model accuracy (%) on regular/base conversations without any distractor. Numbers reported here are for a common subset of conversations that are translated from script using gpt-4o.

MODEL PERFORMANCE ON DISTRACTOR CONVERSATIONS GPT-4o TRANSLATED CONVERSATION			
Model	**Omniscient (%)**	**Participant Centric (%)**	**Overall (%)**
claude-3-5-sonnet-20241022	54.2	34.3	38.0
gpt-4o-2024-08-06	41.7	23.8	27.1
llama3-1-405b-instruct	41.7	21.9	25.6
llama3-2-90b-instruct	20.8	16.2	17.1
llama3-2-11b-instruct	8.3	5.8	6.3

Table 7: Model accuracy (%) on conversations with distractors. Numbers reported here are for a common subset of conversations that are translated from script using gpt-4o.

where gpt-4o is used to translate the script to conversation. Table 6 reports the model performance on base/regular conversations, Table 7 on conversations with distractor and on underspecified conversations for which the questions are unanswerable. We do not include these data as part of DIAMONDs dataset, due to the observed quality of the translated conversations to be erroneous and unnatural in generation. However, we are reporting the performance just to show that the relative performance trends of the model is same as that observed for conversations generated with Claude-3.5 (Table 1, 3).

A.10 Prompts

A.10.1 Script Premise Components

```
# Instruction
Create a simple and unambiguous math word problem (MWP) based on a situation involving conversation between
multiple participants.

## 1. Input
- A conversation theme
- number of participants in the conversation

## 2. Output Component

### 2.1 setting
- a brief, high-level description of the conversation's context.

### 2.2 Math Word Problem (mwp)
- Based on the setting, create a detailed math word problem (mwp) that includes various numerical information
associated with the participants, that are expressed by them.
- Must include:
    - Clear numerical information for each participant associated with them
    - Logical relationships between quantities if any exists
    - Realistic and consistent values
    - Clear units of measurement

### 2.3 question
- Formulate a sufficiently specified question about one numeric quantity of interest that can be answered using
the information in the mwp.

- Requirements:
    - Avoid questions that ask for multiple results, e.g.:
        ✗ How many liters of water can they purchase, and will they have enough to meet the weekly water
requirements for all their trees?
        ✗ What are the individual yields of each participant?
   - Include any extra assumptions or context necessary to make the question unambiguous.
   - 'question' should be such that answering it should require utilizing almost all the numerical information
in the 'mwp'.
   - 'question' should' yield one unambiguous answer, e.g:
        ✗ What is the company's net profit for the quarter?
        ✓ Calculate the company's quarterly profit or loss (express profit as positive and loss as negative).

        ✗ What is the remaining balance?
        ✓ Calculate the remaining balance, expressing excess as positive and shortfall as negative..

### 2.4 Participants
- List of participant names in the form they appear in the 'mwp'

## 3. Output Format
JSON object with the following structure:
```json
{{
  "setting": "string",
  "mwp": "string",
  "question": "string",
  "participants": ["string"]
}}

## 4. Examples
{Example 1}
…
{Example n}
```

Figure 11: Conversation script premise prompt
```

A.10.2 Variable State Perturbations Generation

```
# Instruction
Create logically consistent variable changes that affect the solution to a given math word problem (narrative)
while maintaining problem solvability.

## 1. Input
- A complete math word problem in JSON format containing:
  - setting: a brief, high-level description of the conversation's between participants that is described in the
narrative
  - narrative:  detailed math word problem/narrative based on the setting, that includes various numerical
information associated with the participants
  - question: a question about a quantity of interest that can be answered using the information in the
narrative
  - participants: participants in the narrative

## 2. Variable Selection Criteria

### 2.1 Eligible Variables
Variables must be:
- Explicitly mentioned in the narrative
- Used directly or indirectly in calculating the answer
- Quantifiable and clearly specified
- Modifiable while maintaining problem solvability
- Independent or unaffected by the state of other variables

## 3. Change Specifications

### 3.1 Expression Format
Must be:
- Relative to original value
- Self-contained (not referencing other participants)
- Complete with all necessary information needed to calculate the new answer due to this change
- Free of redundant information
- Avoid dependencies on other participants' changes
- Begin with "During their conversation"

Examples:
✓ "decides to add 5 more units at the same cost per unit"
✗ "increases units from 10 to 15"
✓ "reduces processing time by 10 minutes"
✗ "processing time becomes same as Bob's"

## 4. Output Format
JSON object with the following structure:
{{
  "participant1_name": "string description of change of variable associated with participant 1.",
  "participant2_name": "string description of change of variable associated with participant 2.",
  ...
  "participantn_name": "string description of change of variable associated with participant n."
}}

## 5. Examples
{Example 1}
…
{{Example n}
```

Figure 12: Variable State Perturbation: Information perturbing the state of the variables established in the remise

A.10.3 Generate Underspecified and Distractor Script Variants

Underspecific Script We generate Unanswerable or underspecified variants S^- of base script S through systematic information omission. Given a base script S with premise S_p, we create an underspecified premise S_p^- (Figure 21 (1)) by removing key information required to answer the question. We then generate two sets of variable state perturbations: (i) $V_{S_p^-}$ for the underspecified premise S_p^- and (ii) $V_{S_p}^-$ (Figure 21 (2)) for the base script premise S_p with underspecified variable information. We create unanswerable script S^- by combining (S_p^-, $V_{S_p^-}$) or (S_p, $V_{S_p}^-$, V_{S_p}) through our script template, that test models' ability to recognize insufficient information. The components for the distractor scripts are generated with few-shot prompting using a Large Language Model. The prompts for generating S_p^- and $V_{S_p}^-$ are shown in Appendix, Figure 13, and 15.

Distractor Script For generating the distractor variants S^+, we augment the base script S with additional, thematically relevant but question-irrelevant information. We first transform a base script premise S_p into an augmented premise S_p^+ by introducing information about new quantities u_i and u_j (Figure 18 (1)) associated with participants p_i and p_j. These additions maintain conversational coherence while being irrelevant to answering the target question. We then generate u_i and u_j's corresponding variable state perturbations v_i^+ and v_j^+ (Figure 18 (2)) for these new quantities. The final distractor script S^+ is assembled using an augmented template that augments the original template slots with positions for distractor variable updates. This process preserves the original answer to question q, as the added information in S^+ is designed to be independent of the solution path. Similar to the distractor scripts, the components for the unanswerable scripts are generated with few-shot prompting using a Large Language Model. The prompt template used to generate S_p^+, and v_i^+ and v_j^+ is shown in Appendix, Figure 13, 14.

```
# Instruction
You are given a 'narrative' and a 'question' about a setting involving conversation between multiple
participants. You are required to generate two variants of the narrative:
 - overspecified narrative with additional distractor information that does not change the answer to the
question
 - underspecified narrative with key information removed that makes the question unanswerable

## 1. Input (in JSON Format)
- **setting**: A brief description of the conversation context
- **narrative**: The original math word problem narrative
- **question**: The numerical question to be answered
- **participants**: List of all participants in the conversation

## 2. Task Steps

### Underspecified Variant
1. Identify key information in the **narrative** which, if removed, would make the **question** unanswerable
   - Key information must be a specific numerical value or relationship essential for calculation
   - If multiple pieces could make it unanswerable, select the most critical one
2. Create new narrative by removing the identified key information
   - Maintain the coherence and flow of the original narrative
   - Only remove the specific piece of information identified

### Overspecified Variant
1. Select exactly 2 participants from the input participant list
2. Generate one piece of distractor information for each selected participant
   - Distractor must be relevant to the setting and participant's role
   - Distractor must not affect the numerical answer to the question
   - Distractor should be quantitative information (measurements, counts, percentages, etc.)
3. Create overspecified narrative by adding both pieces of distractor information
   - Insert distractors near the original mentions of the selected participants
   - Maintain natural flow of the narrative

## 3. Output Format

{{
  "underspecified": {{
    "missing_info": "string describing the removed critical information",
    "underspecified_narrative": "string containing modified narrative"
  }},
  "overspecified": {{
    "selected_participants": ["string", "string"],
    "distractor-info": {{
      "participant1": "string containing distractor information",
      "participant2": "string containing distractor information"
    }},
    "overspecified_narrative": "string containing modified narrative"
  }}
}}

## 5. Example
{Example 1}
…
{Example n}
```

Figure 13: Prompt for generating (i) underspecified premise and (ii) premise with distractor information.

A.10.4 Distractor Variable State Perturbations Generation

```
# Instruction
Create changes to the state of participant's variable that do not affect the answer to a given question while
maintaining narrative consistency and plausibility.

## 1. Input
- A complete scenario in JSON format containing:
  - setting: a brief, high-level description of the conversation between participants that is described in the
narrative
  - narrative:  detailed math word problem/narrative based on the setting, that includes various numerical
information associated with the participants
  - question: a question about a quantity of interest that can be answered using the information in the
narrative
  - distractor-info: participants and their associated variables in narrative that do not affect the answer to
the question

## 2. Variable Selection Criteria

### 2.1 Eligible Variables
Variables must be:
- Mentioned in distractor-info
- Not used in calculating the answer
- Clearly specified with current state/value
- Modifiable while maintaining narrative plausibility
- Independent of answer-critical variables

## 3. Change Specifications

### 3.1 Expression Format
Must be:
- Relative to original value
- Self-contained within participant context
- Free of references to other participants
- Include only necessary information
- Begin with "During their conversation"

Examples:
✓ "decides to reduce water usage by 1 liter per day"
✗ "reduces water usage from 5 to 4 liters"
✓ "plans to expand farm area by 10 square meters"
✗ "expands farm to match Nina's size"

## 4. Output Format
{{
  "participant1_name": "string describing change to distractor variable",
  "participant2_name": "string describing change to distractor variable",
  ...
}}

## 5. Examples
{Example 1}
…
{Exampl. n}
```

Figure 14: Perturbing distractor information in premise with distractor

A.10.5 Underspecified Variable State Perturbations Generation

```
# Instruction
Create underspecified variable changes that affect the solution to a given math word problem (narrative) while
intentionally omitting or underspecifying key information needed for precise calculation.

## 1. Input
- A complete math word problem in JSON format containing:
  - setting: a brief, high-level description of the conversation between participants
  - narrative: detailed math word problem/narrative based on the setting, including various numerical
information
  - question: a question about a quantity of interest that can be answered using the narrative
  - participants: participants in the narrative whose variables state should change

## 2. Variable Selection Criteria

### 2.1 Eligible Variables
Variables must be:
- Explicitly mentioned in the narrative
- Used directly or indirectly in calculating the answer
- Quantifiable and clearly specified in the original problem
- Independent (not defined relative to other participants' variables)

### 2.2 Underspecification Requirements
Changes must:
- Include some numeric information
- Deliberately omit crucial details needed for precise calculation
- Avoid explicit uncertainty phrases (e.g., "is not sure", "doesn't know")
- Maintain independence from other participants' variables
- Include at least one numeric value

## 3. Change Specifications

### 3.1 Expression Format
Must be:
- Begin with "During their conversation"
- Self-contained (not referencing other participants)
- Intentionally omit key details that would be needed for calculation
- Free of explicit uncertainty markers
- Independent of other participants' changes

Examples:
✓ "decides to add 2 more powerful motors that cost extra"
✗ "decides to add 2 motors that cost $15 each"
✓ "will use 3 premium sensors"
✗ "is unsure about how many sensors to add"
✓ "plans to increase production by 20% using additonal enhanced materials"
✗ "will match Bob's production rate"

## 4. Output Format
JSON object with the following structure:

{{
  "participant1_name": "string description of underspecified change for participant 1",
  "participant2_name": "string description of underspecified change for participant 2",
  ...
  "participantn_name": "string description of underspecified change for participant n"
}}

## 5. Examples
{Example 1}
…
{Example n}
```

Figure 15: Prompt for generating underspecified variable state perturbation information that effects the answer to the question

A.10.6 Script to Conversation Translation

```
# Conversation Generation Instructions

## 1. Overview
Convert a given 'context' into a natural multi-turn 'conversation' between participants. The output should be a
JSON structure containing segmented conversations, i.e. one conversation segment for one context segment.
The convesation should accurately reflect the context while maintaining natural flow and consistent character
personalities.

## 2. Base Requirements

### 2.1 Information Coverage
- EACH and EVERY piece of information from the context must be conveyed in the conversation
- Follow the exact same discourse order as the context
- Do not include any numerical calculations/information beyond what's explicitly stated
- Maintain factual consistency throughout all segments

### 2.2 Conversation Structure
- Maximum 3 sentences per dialogue turn
- include casual exchanges before and after
    (i) someone leaves
    (ii) someone re-enters
    (iii) introducing key information
- Natural transitions between casual conversation and key information
- Break long information into multiple natural turns

### 2.3 Participant Rejoining
- When participants rejoin, DO NOT mention or summarize previous conversations
- Include appropriate welcoming remarks without discussing missed content
- Maintain conversation flow without artificial recap

## 3. Natural Conversation Guidelines

### 3.1 Casual Conversation Topics
- Topics should be relevant to the conversation setting

### 3.2 Character Consistency
- Maintain consistent speaking style for each participant
- Use appropriate level of formality based on relationship
- Maintain consistent enthusiasm and engagement level

### 3.3 Emotional Elements
- Show appropriate reactions to changes in plans
- Express genuine interest in others' updates
- Include natural concern when others leave
- Display appropriate excitement for positive developments

### 3.4 Transitions
- Natural lead-ins to important information
- Smooth transitions between speakers
- Organic conversation flow
- Appropriate closure when participants leave

## 4. Technical Format

{{
    "conversation": [
        [
            {{"participant-1": "dialogue 1"}},
            {{"participant-2": "dialogue 2"}},
            {{"participant-3": "dialogue 3"}}
        ],
        [
            {{"participant-2": "dialogue 4"}},
            {{"participant-3": "dialogue 5"}},
            {{"participant-1": "dialogue 6"}}
        ]
    ]
}}

## 5. Example
{Example 1}
…
{Example n}
```

Figure 16: Prompt for translating script to natural conversation.

A.10.7 Model Inference

```
"""You are a logical analyzer with commonsense that can do mathematical reasoning and has theory of
mind.
Given a conversation and a question based on it, analyze them step-by-step using the following process:

1. First, carefully read the provided:
Conversation: [narrative text]
Question: [question text]

2. Think through the following aspects methodically:

a) answerability: evaluate question answerability based on the conversation:
- Does the context provide all necessary information?
- Are there any missing crucial details?
- Can the question be answered definitively based on given information?

b) If the question is answerable, solve it:
- Break down the solution into clear steps
- Show all calculations and reasoning
- Derive the final numerical answer


3. Provide your analysis in the following JSON format:
{
"cot": "Detailed chain of thought explaining your analysis for each step above",
"answerability": "YES/NO/NA",
"solution": ["Step 1...", "Step 2...", ...] or "NA",
"answer": "numerical_answer or NA"
}

Task Rules:
- If you are asked a question from a participants perspective, assume that you are that participant.
To answer the question, you can only use the information that the participant has access to. For example
in this conversation:

{'Councilor Thompson': 'Found something interesting?'},
{'Councilor Chen': 'Yes, our recent preventive maintenance work has paid off. We can reduce emergency
repairs by $40,000.'},
{'Mayor Wilson': "That's excellent news!"},
{'Councilor Thompson': 'Has anyone tried the new café in the city hall lobby?'},
{'Councilor Rodriguez': "Hello everyone! Hope I didn't miss too much."},
{'Mayor Wilson': 'Welcome back! We were just discussing the new café.'},
{'Councilor Chen': 'Their coffee is quite good!'},

Councilor Rodriguez does knows that the 'emergency repairs' can be reduced by $40,000'.

Suggestion:
When asked the question from a participant's perspective, you can summarize the conversation using the
information in it that is accessible to that participant, and use the summary to answer the question.

Output Rules:
- The "cot" should show your complete reasoning process
- "answerability" is "YES" if the question is answerable based on the conversation, otherwise "NO"
- "solution" should be "NA" if answerability is "NO", otherwise ["Step 1...", "Step 2...", ...]
- "answer" should be "NA" if answerability is "NO", otherwise the final numerical answer

Conversation:\n{conversation}

Question:\n{question}

Response:

"""
```

Figure 17: Prompt used for performing inference for the task.

A.11 Distractor and Unanswerable Conversations

A.11.1 Distractor Conversations

1. Distractor information to be added to script premise:

Chen:
Chen mentions that his robotics workspace takes up 4 square meters in his garage.

Diana:
Diana notes that she spends 12 hours per week monitoring her plant specimens.

2. Variable state perturbation for distractor variables

Chen:
During their conversation, Chen mentioned he needs to expand his robotics workspace to 6 square meters to accommodate new equipment.

Diana:
During their conversation, Diana noted that she plans to increase her plant monitoring time to 15 hours per week for better data collection.

Figure 18: Components Required for Distractor Script Generation: (1) Distractor variable information to be added to the base script's premise (Figure 5) to create a distractor premise, and (2) Variable state perturbations for the distractor variables in the distractor premise.

At a high school science fair planning meeting, four students – Alex, Bella, Chen, and Diana – are discussing their project budgets and material requirements.
Alex is working on a renewable energy project and needs 3 solar panels at $45 each, 2 voltage meters costing $25 each, and plans to create 15 informational handouts at $2 per copy. Bella's chemistry experiment requires 8 different chemicals costing $12 each, 3 sets of safety equipment at $30 per set, and 5 specialized glass beakers at $15 each. Chen's robotics project needs 2 microcontrollers at $35 each, 6 servo motors costing $20 each, and a programming kit for $85. He mentions that his robotics workspace takes up 4 square meters in his garage. He also needs to print 20 pages of documentation at $0.50 per page. Diana's environmental science project requires 4 soil testing kits at $28 each, 6 plant specimens costing $8 each, and 2 digital thermometers at $22 each. She notes that she spends 12 hours per week monitoring her plant specimens. She plans to create 10 presentation posters at $5 each.
Chen leaves the conversation because of – "[need to schedule another appointment]"

Some casual conversation goes on between ['Alex', 'Bella', 'Diana'].
During their conversation, Diana noted that she plans to increase her plant monitoring time to 15 hours per week for better data collection.
Some casual conversation goes on between ['Alex', 'Bella', 'Diana'].
During their conversation, Alex realizes he needs to add 5 more informational handouts at the same cost per copy to reach more visitors.
Diana leaves because of reason "[need to take care of some personal matters]"

Some casual conversation goes on between ['Alex', 'Bella'].
Chen re-enters, after leaving earlier due to "[need to schedule another appointment]"
Some casual conversation goes on between ['Alex', 'Bella', 'Chen']. (They do not talk about the specific details of their past conversations that Chen missed.)
During their conversation, Chen decides to add 2 more servo motors at the same cost per unit to improve his robot's functionality.

Some casual conversation goes on between ['Alex', 'Bella', 'Chen'].
During their conversation, Bella discovers she needs one additional set of safety equipment at the same cost per set.
Some casual conversation goes on between ['Alex', 'Bella', 'Chen'].
During their conversation, Chen mentioned he needs to expand his robotics workspace to 6 square meters to accommodate new equipment.
Chen leaves because of reason "[have unexpected visitor]"

Some casual conversation goes on between ['Alex', 'Bella'].
Diana re-enters, after leaving earlier due to "[need to take care of some personal matters]"
Some casual conversation goes on between ['Alex', 'Bella', 'Diana']. (They do not talk about the specific details of their past conversations that Diana missed.)
During their conversation, Diana decides to reduce her poster count by 2 as she found more efficient ways to present her information.

Figure 19: The Distractor Script is created through two components: (1) incorporating distractor variable information into the base premise, and (2) applying variable state perturbations to both (violet) distractor variables and (cyan) relevant variables within the premise.

A.11.2 Unanswerable Conversations

```
Alex : Hey everyone! Ready to dive into our science fair project plans?
Bella : Definitely! I've got my budget spreadsheet ready.
Chen : Same here. My garage is already turning into a mini robotics lab!
Diana : Let's share what we need. Alex, want to start?
Alex : Sure! For my renewable energy project, I need 3 solar panels at $45 each, 2 voltage
meters at $25 each, and 15 handouts at $2 per copy.
Bella : Sounds organized! For my chemistry experiment, I need 8 chemicals at $12 each, 3
safety equipment sets at $30 each, and 5 special beakers at $15 each.
Chen : My robotics project's getting complex. I need 2 microcontrollers at $35 each, 6 servo
motors at $20 each, and an $85 programming kit.
Diana : That's quite a setup, Chen!
Chen : Yeah, it takes up 4 square meters in my garage. Oh, and I need 20 pages of
documentation at 50 cents each.
Diana : For my environmental project, I need 4 soil testing kits at $28 each, 6 plant
specimens at $8 each, and 2 digital thermometers at $22 each.
Bella : How are you managing all those plants, Diana?
Diana : I'm dedicating 12 hours weekly to monitoring them. Also planning 10 presentation
posters at $5 each.
Chen : Oh shoot - just remembered I need to schedule an appointment. Got to run!
Alex : No problem, Chen. Catch you later!
Alex : The science fair's really shaping up this year!
Bella : The variety of projects is amazing. Diana, how's your plant monitoring going?
Diana : Actually, I've been thinking about that...
Alex : Found something interesting in your data?
Diana : Yes, I'm going to increase my monitoring to 15 hours weekly. Need more detailed
observations.
Bella : That's dedication! The judges will definitely notice that.
Alex : Speaking of my project, I've been reviewing my outreach strategy...
Bella : Need to make some adjustments?
Alex : Yes, I should add 5 more handouts at $2 each. Want to reach more visitors.
Diana : That's smart thinking! Oh - I just remembered I need to take care of some personal
matters.
Bella : No worries, Diana. See you soon!
Alex : Have you seen the layout plan for the science fair?
Bella : Yes! Our projects are well-spaced this year.
Chen : Hey everyone! Finally got that appointment sorted.
Alex : Chen! Perfect timing - we were just discussing the fair layout.
Bella : How's the robot coming along?
Chen : Actually, I've been tinkering with the design...
Alex : Making improvements?
Chen : Yes, adding 2 more servo motors at $20 each. Should make the movements smoother.
Bella : Can't wait to see it in action!
Alex : The judges this year have impressive backgrounds!
Bella : That reminds me - I need to review my safety protocols...
Chen : Always better to be over-prepared.
Bella : Actually, I should add another safety equipment set at $30. Better safe than sorry!
Alex : Good call, Bella. Safety first!
Chen : Speaking of space needs, my robot's getting bigger than expected.
Bella : Need more room?
Chen : Yes, I'll need to expand to 6 square meters in the garage. Oh - someone's at my door
unexpectedly.
Alex : Go ahead, Chen. We'll catch up later!
Alex : These project deadlines are approaching fast!
Bella : But we're making good progress.
Diana : Hi everyone! Hope I didn't miss anything exciting!
Alex : Diana! Welcome back. We were just talking about deadlines.
Bella : How's your presentation planning going?
Diana : Actually, I've been rethinking my poster strategy...
Alex : Found a better approach?
Diana : Yes, I can reduce my poster count by 2. Found more efficient ways to present the
data.
Bella : Sometimes less is more!
```

Figure 20: Conversation created from distractor script (Figure 19)

1. Information to be removed from base script premise

Alex does not mention the cost per solar panel for his renewable energy project.

2. Underspecified variable perturbations for the base script premise

Chen:
During their conversation, Chen decides to add several high-torque servo motors costing $25 each and an enhanced programming interface.

Alex:
During their conversation, Alex plans to incorporate additional high-efficiency solar panels and will need extra voltage meters for the expanded setup.

Figure 21: Components Required for Unanswerable Script Generation: (1) Elements to be excluded from the base script premise (Figure 5) to create an underspecified premise, and (2) Underspecified variable state perturbations for the base premise.

At a high school science fair planning meeting, four students - Alex, Bella, Chen, and Diana - are discussing their project budgets and material requirements.
Alex is working on a renewable energy project and needs 3 solar panels at $45 each, 2 voltage meters costing $25 each, and plans to create 15 informational handouts at $2 per copy. Bella's chemistry experiment requires 8 different chemicals costing $12 each, 3 sets of safety equipment at $30 per set, and 5 specialized glass beakers at $15 each. Chen's robotics project needs 2 microcontrollers at $35 each, 6 servo motors costing $20 each, and a programming kit for $85. He also needs to print 20 pages of documentation at $0.50 per page. Diana's environmental science project requires 4 soil testing kits at $28 each, 6 plant specimens costing $8 each, and 2 digital thermometers at $22 each. She plans to create 10 presentation posters at $5 each.
Chen leaves the conversation because of - "[need to schedule another appointment]"

Some casual conversation goes on between ['Alex', 'Bella', 'Diana'].
During their conversation, Alex plans to incorporate additional high-efficiency solar panels and will need extra voltage meters for the expanded setup.
Diana leaves because of reason "[need to take care of some personal matters]"

Some casual conversation goes on between ['Alex', 'Bella'].
Chen re-enters, after leaving earlier due to "[need to schedule another appointment]"
Some casual conversation goes on between ['Alex', 'Bella', 'Chen']. (They do not talk about the specific details of their past conversations that Chen missed.)
During their conversation, Chen decides to add several high-torque servo motors costing $25 each and an enhanced programming interface.

Some casual conversation goes on between ['Alex', 'Bella', 'Chen'].
During their conversation, Bella discovers she needs one additional set of safety equipment at the same cost per set.
Chen leaves because of reason "[have unexpected visitor]"

Some casual conversation goes on between ['Alex', 'Bella'].
Diana re-enters, after leaving earlier due to "[need to take care of some personal matters]"
Some casual conversation goes on between ['Alex', 'Bella', 'Diana']. (They do not talk about the specific details of their past conversations that Diana missed.)
During their conversation, Diana decides to reduce her poster count by 2 as she found more efficient ways to present her information.

Figure 22: The Unanswerable Script is constructed by combining three elements: (1) the base script premise (Figure 5), (2) underspecified variable state perturbations (violet) (Figure 21) for some participants, and (3) sufficiently specified variable state perturbations (cyan)(Figure 6) for the remaining participants.

Alex : Hey everyone! Ready to dive into our science fair budget planning?
Bella : Yes! I've got my list all ready. This chemistry project is getting complex.
Chen : Same here with my robotics project. The parts list keeps growing!
Diana : Let's break it down one by one. Who wants to start?
Alex : I'll go first! For my renewable energy project, I need 3 solar panels at $45 each, 2 voltage meters at $25 each, and I'm planning 15 handouts at $2 per copy.
Bella : Nice! For my chemistry experiment, I need 8 chemicals at $12 each, 3 sets of safety equipment at $30 per set, and 5 special glass beakers at $15 each.
Chen : My robotics project requires 2 microcontrollers at $35 each, 6 servo motors at $20 each, and an $85 programming kit. Plus 20 pages of documentation at 50 cents each.
Diana : For my environmental science project, I need 4 soil testing kits at $28 each, 6 plant specimens at $8 each, and 2 digital thermometers at $22 each. Also planning 10 posters at $5 each.
Chen : Oh shoot, just remembered I have an appointment to schedule. Got to step out for a bit!
Alex : No problem, Chen. Catch you later!
Alex : You know, I've been doing more research on solar efficiency...
Bella : Found something interesting?
Diana : Your renewable energy project is really coming together, Alex.
Alex : Actually, yes! ~~I'm thinking of adding some high-efficiency solar panels and extra voltage meters to expand the setup.~~
Bella : That could give you much better data to present!
Diana : Oh, I just remembered - I need to take care of some personal matters. See you both later!
Alex : Take care, Diana!
Bella : Bye, Diana! Don't forget about tomorrow's meeting!
Alex : Have you seen the new science lab equipment catalog?
Bella : Yes! Some amazing new tools in there.
Chen : Hey everyone! The appointment's all sorted.
Alex : Welcome back! We were just checking out the new catalog.
Bella : Find any interesting robotics equipment lately, Chen?
Chen : Actually, yes! ~~*I'm planning to add some high-torque servo motors at $25 each*~~ and an enhanced programming interface.
Alex : That sounds like it'll give you much more precision!
Bella : Your robot's going to be amazing with those upgrades.
Alex : The science fair layout plans look great this year.
Chen : They really thought through the power supply locations.
Bella : Speaking of safety, I just realized something about my setup...
Alex : Everything okay?
Bella : Yes, but I'll need one more set of safety equipment at the same cost. Better to be over-prepared!
Chen : Oh! Someone's at my door - wasn't expecting visitors. Got to run!
Alex : See you later, Chen!
Bella : Take care!
Alex : These project deadlines are coming up fast!
Bella : Tell me about it! But it's all coming together.
Diana : Hi everyone! Hope I didn't miss anything exciting!
Alex : Diana! Welcome back. We were just talking about deadlines.
Bella : How are your plant specimens doing?
Diana : They're great! Actually, I've been thinking about my presentation format...
Alex : Found a better way to showcase your results?
Diana : Exactly! I can reduce my poster count by 2. Found some more efficient ways to present the information.
Bella : That's great! More concise can be more impactful.

Figure 23: Conversation created from an unanswerable script (Figure 19). Strike through text shows the underspecificity in the conversation.