DECISION-ORIENTED DIALOGUE FOR HUMAN–AI COLLABORATION

Jessy Lin[*1] **Nicholas Tomlin**[*1] **Jacob Andreas**[2 3] **Jason Eisner**[2 4]
[1] UC Berkeley [2] Microsoft Semantic Machines [3] MIT [4] Johns Hopkins
{jessy_lin, nicholas_tomlin}@berkeley.edu
{jaandrea, jason.eisner}@microsoft.com

ABSTRACT

We describe a class of tasks called *decision-oriented dialogues*, in which AI assistants such as large language models (LLMs) must collaborate with one or more humans via natural language to help them make complex decisions. We formalize three domains in which users face everyday decisions: (1) choosing an assignment of reviewers to conference papers, (2) planning a multi-step itinerary in a city, and (3) negotiating travel plans for a group of friends. In each of these settings, AI assistants and users have disparate abilities that they must combine to arrive at the best decision: assistants can access and process large amounts of information, while users have preferences and constraints external to the system. For each task, we build a dialogue environment where agents receive a reward based on the quality of the final decision they reach. We evaluate LLMs in self-play and in collaboration with humans and find that they fall short compared to human assistants, achieving much lower rewards despite engaging in longer dialogues. We highlight a number of challenges models face in decision-oriented dialogues, ranging from goal-directed behavior to reasoning and optimization, and release our environments as a testbed for future work.

1 INTRODUCTION

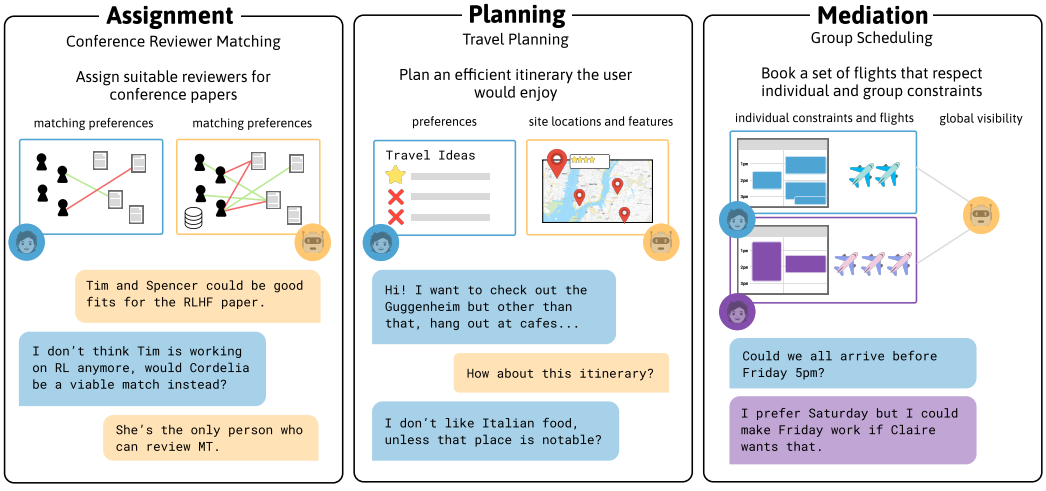

Figure 1: Overview of the three collaborative dialogue tasks that we consider. In `Assignment`, two agents with symmetric access to information play the role of area co-chairs assigning reviewers to conference papers. In `Planning`, an assistant collaborates with a user to help them plan an itinerary. In `Mediation`, an assistant must chat with multiple separate users to help them resolve a group scheduling problem.

Imagine that you are trying to book conference travel with the help of a digital assistant. Your choice of airline is flexible, but you'd rather avoid layovers, want to arrive a day or two before the conference begins, and would like to be able to check in to your hotel as soon as you arrive. Additionally, you're in charge of booking travel for a few of your colleagues, each of whom has their own preferences and budgets, some of whom will be flying in from different cities, but all of whom would like to arrive at roughly the same time and stay in a nearby area. Suddenly, you must manage and communicate about a combinatorial explosion of possible itineraries.

Similar optimization problems occur in many everyday situations. Consider consulting a friend about what computer they'd recommend with the best tradeoff of features for your use cases. Or trying to allocate funding from multiple grants to determine which students should work on which projects, while juggling student preferences. Or making strategic decisions with your colleagues about which projects your company will take on and who to hire to manage those projects. All these situations share an underlying decision problem in the face of uncertainty, where collaborating with others is often critical to arrive at the best solution.

Difficult decision problems like these are precisely where AI assistants could shine. Automated systems can handle large amounts of information and complex computations much better than humans. For example, in cases like travel booking, they can quickly search over a large number of possible itineraries and compute total costs in a way that the average user cannot. They may also be able to efficiently reason under uncertainty about the expected value of decision-relevant information, helping them determine what information may be important to share with or request from the user. On the other hand, these decisions cannot be *fully* automated either. AI assistants *complement* humans' knowledge and capabilities: people know their preferences and may have other knowledge external to the system, including knowledge about fuzzy real-world constraints that are difficult to formalize in a computer-readable format. To solve these problems, systems need to communicate with users, ideally with a flexible interface such as natural language. However, there is limited existing work evaluating model performance in these types of conversational settings. In this paper, we develop a challenging suite of decision problems in which multiple agents must collaborate with each other and make decisions via natural language. We then benchmark the abilities of language models on these tasks and release datasets and environments to encourage future modeling work in this area.

We begin by formalizing the setting of *decision-oriented dialogue*, a class of tasks in which multiple agents must communicate in order to arrive at a joint decision, perhaps from a combinatorially large space of options. Agents in these tasks are jointly rewarded according to the quality of the decision. Each agent starts out with different information: for example, the user knows their own travel preferences, while the AI assistant has a database of flight and hotel prices. Sharing their information allows them to better assess different travel plans. Critically, the large amount of information makes it unnatural and inefficient for assistants to communicate *all* of their knowledge to users, or vice versa. Instead, agents must determine what their partners already know and what information is likely to be decision-relevant, asking questions and making inferences as needed.

Within this class of tasks, we present three everyday domains where humans and agents must collaborate in order to make complicated decisions. (1) In `Assignment`, two agents take on the role of conference area chairs, assigning reviewers to conference papers when each agent has only has partial information about reviewer–paper fit. (2) In `Planning`, an assistant with knowledge of a city must assist a human with building an itinerary based on their preferences. (3) In `Mediation`, multiple users must collaborate with an assistant in order to resolve group scheduling challenges. For each task, we specify an objective measure of utility based on the quality of the final decision. We first collect human–human dialogues in order to establish a reference point for how humans naturally collaborate with each other. These are long dialogues, averaging 13 messages over 8 minutes (Table 1). We then develop extensible environments for evaluating language models on each task.

We use these environments to benchmark the relative performance of GPT-3 (Brown et al., 2020) in collaboration with humans, along with additional experiments in self-play and in a novel evaluation procedure known as *prompted self-play*, in which AI agents complete partial human dialogues. We then identify several common failure modes of GPT-3 and provide analyses of self-play dialogues.

2 TASK FORMULATION

We formalize a *decision-oriented dialogue* (DoD) task as a multi-agent problem consisting of a set of agents, an underlying world state W, each agent's partial and possibly noisy observation O_i, a set of legal messages $m \in \mathcal{M}$ (analogous to actions in an Markov decision process), a reward function R with parameters θ that evaluates decisions, and a communication cost function C. The goal of a decision-oriented dialogue is to find a decision that maximizes R while minimizing the communication cost function C. W remains fixed throughout the dialogue. Our problem can be thought of as a decentralized partially observable Markov decision process (Dec-POMDP; Bernstein et al., 2000) in which actions are messages and formal decisions.

An agent i's policy π_i maps its known information O_i and the dialogue history $\{m_1, \dots m_{t-1}\}$ to a new message m_t: $\pi_i(m_t \mid O_i, \{m_1, \dots m_{t-1}\})$. Agents send messages by sampling from their policy. Messages may specify a recipient if the number of agents > 2, and are expressed in natural language except for three special formal messages: a proposed decision, a formal acceptance of a decision, and a formal rejection. If an agent sends a proposed decision message and all other agents respond with a formal acceptance, the dialogue ends.

To illustrate the information in a DoD, consider the task of planning a travel itinerary that satisfies a user's preferences (`Planning`, as shown in Figure 1, middle). We represent the underlying world state as a weighted graph $W = (V, E, w)$ whose vertices are potential destinations. A decision is a path W' in W, representing the itinerary. Higher-weighted paths are better and the agents must communicate to improve their knowledge of the edge weights.

In general, we represent the world state W as a weighted graph and the possible decisions as subgraphs W' that satisfy task-specific constraints.[1] Edges and vertices in W have weights $w(e_{ij}), w(v_i)$ that represent rewards (which may be negative) for including them in W'. The optimal decision for this world state is a subgraph $W' \subseteq W$ that maximizes the reward

$$R_\theta(W') = \sum_{v \in W'} w(v) + \sum_{e \in W'} w(e) \tag{1}$$

In principle, the reward function could be any function of W', but we focus on the linear objective (1). For most practical tasks, the constrained optimization problem could then be expressed as an integer linear programming problem and solved using standard algorithms. We assume edge and vertex weights are determined by their features, represented by feature vectors $\phi(\cdot) \in \mathbb{R}^k$, so that:

$$w(v_i) = \theta^T \phi(v_i) \qquad w(e_{ij}) = \theta^T \phi(e_{ij}) \tag{2}$$

where θ is a preference vector.[2]

The hard constraints on W' and the form of the objective are treated as common knowledge. However, the world state W—in particular the feature vectors and the preferences θ—is only partially observed by each agent. Therefore, crucially, agents must exchange messages in order to reduce their respective uncertainties about the optimization problem. However, there is a cost to communicating (e.g., time or effort), which agents must trade off with their desire to achieve a good decision. Thus, the overall objective function for a DoD is:

$$\max_{W', \mathbf{m}} R_\theta(W') - \sum_t C(m_t) \tag{3}$$

$$\text{subject to } \textit{task-specific constraints on } W' \subseteq W$$

Other collaborative or task-oriented dialogue tasks are typically evaluated on coarse metrics such as success rate (Li et al., 2016), which measure whether a system accomplished its user's goal. In contrast, the reward in a DoD provides a *graded* measure of communication success, measuring how close to optimal a final decision is.

[1] Representing W as a graph lets us model most discrete optimization problems. A more general formulation could assume an unstructured world state; agents would communicate about random variables representing unknown quantities in the world state, rather than features of an underlying graph.

[2] To reward edges between similar or dissimilar vertices, one could define $\phi(e_{ij}) = \phi(v_i) \odot \phi(v_j)$, for example.

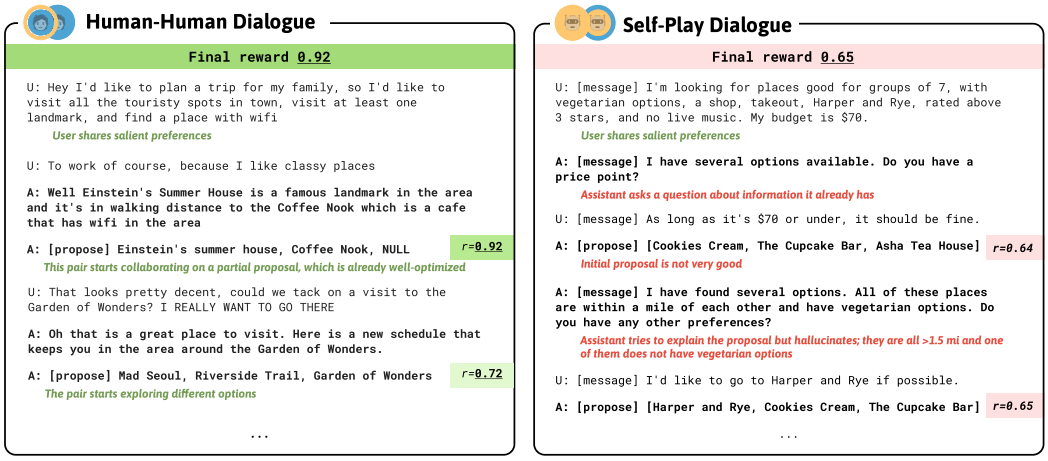

Figure 2: An annotated example of a human-human dialogue and a model-model self-play dialogue with GPT-3 in `Planning`. While humans generally exhibit diverse and flexible strategies and reach good solutions, self-play dialogues tend to be repetitive, and the assistant makes mediocre proposals and often hallucinates. We discuss more analysis in §7.

3 THE `DialOp` ENVIRONMENTS

We introduce three everyday collaborative decision-making domains formalized as DoD tasks. To instantiate them, we release `DialOp`, an open-source suite of decision-oriented dialogue environments For each task, we implement a text environment to evaluate models in self-play (as in §6.2), a graphical UI to build human user interfaces for data collection (as in §4), and a unified interface between the two to evaluate models in collaboration with humans (as in §6.1). An example dialogue for one task is highlighted in Figure 2, and example dialogues for every task can be found in Appendix C. In Appendix B.1, we describe how each task can be formalized as a DoD task. Here, we describe how we implement the environments at a high level, with more details provided in Appendix B.2.

In contrast to other dialogue tasks where evaluation is based on supervised datasets, we procedurally generate each game by sampling the parameters of the underlying decision problem (e.g. the reward parameters θ) to instantiate new dialogue contexts. This process enables future work to study how models generalize: e.g. to larger optimization problems (by changing the parameter dimensions) or new domains (by changing the "theme" while keeping the underlying parameters fixed).

Agents interact with the text environments through an OpenAI Gym-like interface (Brockman et al., 2016). Agents send messages to the environment, prefixing each with a message type (`[message]`, `[propose]`, `[accept]`, or `[reject]`), which the environment parses to determine how to interpret the message. Messages are forwarded to other agents. Proposals are parsed and scored; on the next turn, the only valid actions for the other agents are `[accept]` and `[reject]`. Formal rejections clear the current proposal, and formal acceptances terminate the dialogue. Below, we describe how the environments implement each of the decision domains we introduce.

3.1 `Assignment`

Our first task is an idealized bipartite matching problem, motivated by the scenario of conference organizers assigning reviewers to submitted papers (Figure 1, left). Although reviewer matching is sometimes automated via approaches like the Toronto Paper Matching System (TPMS; Charlin & Zemel, 2013), organizers often have incomplete and partially-overlapping knowledge about which reviewers fit which papers. Further, fit cannot necessarily be described on an absolute scale, so when working together on an assignment, organizers must discuss relative edge weights ("Alice would be a better choice than Bob for paper 8"). TPMS could in principle be replaced by an AI agent that joins this dialogue as an additional participant. We consider a simplified version of this problem in which two agents must find a one-to-one matching between reviewers and papers.

Environment Implementation For each game, we sample a random 8×8 table of reviewer-paper affinity scores (edge weights). Each cell is shown to each agent with probability $p_{\text{observed}} = 0.4$, so that a given cell may be shown to just one agent, to both, or to neither. To discourage reviewers from communicating affinity scores in the form of numbers—which would not be natural in the real-world version of this scenario—we scale all scores shown to each agent by a random positive constant, so that they are not comparable across agents but can still be discussed in relative terms such as "X is much better than Y." Each player observes a subset of the reviewer-paper affinity scores, scaled by some constant unknown to them. The final reward is the sum of edge weights in the final matching, normalized by the value of the best matching with the agents' pooled knowledge, computed by assuming the values neither of them knows are the average value.

3.2 `Planning`

Next, we consider the scenario in which a user is planning an itinerary in a city with the assistance of a travel agent (Figure 1, middle). While existing systems can assist with parts of travel such as recommendation or booking, they often expect users to provide close-to-full specifications of their requests, rather than working toward a solution together. Ideally, systems would be able to assist us in the comprehensive way that a human travel agent would: starting with an under-specified set of "things we'd like to do," comprehensively exploring multi-day itineraries based on the user's preferences and domain knowledge, and iteratively refining the plan with the user based on feedback.

Environment Implementation In each game, the assistant must propose a set of three sites. The environment comes with a set of sites (e.g., restaurants, parks, museums). On each game, the environment randomizes the features of each site (e.g., expected price range). The environment also has a set of features with natural language labels (e.g., a preference for "Wi-Fi available") and randomly generates the user's preference vector θ with $s = 10$ nonzero elements.

To simulate the fact that people cannot quantify their actual preferences on an absolute scale, the user only observes natural language descriptions of their nonzero preferences, without the numerical preference weights. The assistant only observes the inventory of sites and their features. The environment optionally provides API calls to search over sites, either via (1) a simple domain-specific language (DSL) that can query specific `fields` (e.g. name, category, price) of a site, `filter` over fields, `sort_by` field values (including `distance_to` another destination), and search by `text_query` in freeform natural language or (2) a LLM prompted with examples in the DSL as query executor, which permits simple generalizations from our DSL.

When the assistant proposes a complete or partial itinerary, the proposal reward (while unknown to the assistant) is automatically computed for the user's convenience, including a breakdown of the contributions to the reward from each site, travel times, and budget constraints. With this information, the user can make judgments about aspects of the itinerary (e.g., that it is worth spending extra travel time to visit a particularly desirable site). The game ends when the user accepts a full itinerary of k sites. The final reward is the score of the itinerary, range-normalized by the scores of the best and worst possible k-site itineraries.

3.3 `Mediation`

Finally, we introduce a coordination scenario where the assistant serves as the role of mediator among multiple users (Figure 1, right). The users are attempting to book flights from their respective cities to all arrive at some shared destination at around the same time, e.g., to meet up for an event or vacation. Assistants could be helpful to negotiate individual constraints and consider all the configurations efficiently. We consider a setting where n users can only coordinate through the single assistant. In the task, each user wants to choose a flight that is inexpensive and avoids conflicts with the user's calendar commitments, but that arrives close to the arrival times of other users. The assistant has access to each user's flight options and work calendar, but doesn't observe the user's personal calendar, nor the user's preferences about which meetings are important.

Environment Implementation In each game, the assistant must coordinate flights for two users. The environment generates a random set of personal calendar events, work calendar events, and

Figure 3: Human-LM and self-play scores compared to human dialogues, plotted against dialogue lengths in words. LM assistants achieve lower scores than human assistants on average, and also tend to have longer dialogues. Models in self-play have even lower scores and longer dialogues since they must also play the role of a cooperative user. Marginal histograms show the marginal distributions of the # words and score. The dashed line shows the average score of a random proposal.

importance weights for each event indicating how important it is. The environment also generates a list of flights for each user, each with randomized features for price, arrival time, and departure time.

The user observes their own personal and work calendar and flight set, while the assistant observes the work calendars and flight sets of *both* users (but not their personal calendars, and without the meeting importances). The assistant has one-on-one chats with each user and is allowed to talk to any user at any time; deciding which user to talk to is itself a strategic decision. The assistant can propose a flight to one or both users. When the assistant proposes a flight to a user, the user observes the score breakdown in terms of missed meetings, price, and closeness to the other user's flight (when known). The game ends when the assistant proposes a set of flights for all users and all users accept. The final reward is the sum of their scores, range-normalized by the best and worst possible scores.

4 Dataset

In order to study the communication strategies used by humans and establish baseline performance numbers, we collected a set of human-human dialogues. For each task, we built a multi-player online interface and collected high-quality human-human dialogues in randomized games using a mixture of workers hired directly and through Amazon Mechanical Turk, resulting in a total of 409 dialogues, consisting of 5253 messages and over 58K words across domains. Human players take a median time of 8min 19sec across tasks. Humans achieve an average of roughly 90% of the maximum possible score on both the assignment and planning domains, and close to 100% performance in the mediation domain. We provide additional data statistics and example dialogues for each task in Appendix C.

Human have access to the same information as evaluated models, but presented in a graphical UI rather than as pure text. A side-by-side depiction of the interface for humans and models is shown in Appendix Figure 5. In each task, each annotator played the role of an assistant or user. For ease of play, annotators were not required to take turns, but used a chat interface where they could send a message at any time. Consecutive messages from the same annotator were concatenated into a "turn." Although real-world users know their own preferences, our annotators are emulating users that we have generated programmatically, so we must tell them what their preferences are. This setup gives us full knowledge of user preferences so that we can objectively evaluate the quality of the decision.

5 Baseline Models

Future AI agents for decision-oriented dialogue may benefit from incorporating explicit reasoning over possible world states and possible decisions. However, as a baseline approach, this paper evaluates few-shot prompted LLMs as the AI agents. These have the benefit that they can attempt a wide variety of dialogue interactions without the need for domain-specific training or modeling. In particular, we

focus our evaluations on the instruction-tuned GPT-3 model known as `text-davinci-003`, prompted for each task with 1-2 human-human dialogue examples from the dataset for each task. Models receive the same information that human annotators do, presented through a text-based environment instead of a UI. If models fail to generate a valid message (e.g., user simulator model attempting to send proposals), we append the generated message to the prompt, along with any error message from the game, and continue generating, allowing the model to revise its previous generation. Generally, we simply prompt models with player information in context, with some exceptions we note here. For `Planning`, we noted that models needed particularly complex reasoning to search based on the dialogue (on the assistant side) and decide whether to accept an itinerary based on the scores (on the user side) and implemented a ReAct-style prompting approach (Yao et al., 2023). To do so, we augment the few-shot example dialogues in the user and assistant prompts with `[think]` steps ("`I am losing the most points from the travel time between events. I should reject the proposal...`"), which demonstrate how the agent can reason. For `Mediation`, to handle the multi-party dialogue, we adopt a simple turn-taking strategy where we iterate round-robin through all agents; on the assistant's turn, it is prompted with `You to` and chooses which user to send the message to by generating either `0` or `1`.

6 EVALUATION

In this section, we evaluate how well prompted present-day LLMs can collaborate with humans as a baseline for our task. First, we directly compare the performance of LM assistants with human assistants at assisting human users. While human assistance is the ultimate goal, human-LM evaluation is expensive and frustrating for human users, given the quality of current models, leading us to introduce two automatic evaluation settings for our benchmark to ease future evaluation and provide additional insights into model behavior: self-play and prompted self-play.

6.1 HUMAN-LM EVALUATION

First, we evaluate whether current baseline prompted LLMs can serve as effective decision-making assistants. We recruited 13 participants (a mixture of undergraduates, graduate students, and contractors) and collected a total of 77 dialogues between these participants and GPT-3, prompted with the information for the assistant role. In Figure 3, we show human-human and human-LM normalized rewards against the number of words in the dialogue. We also show the performance of a naive rule-based baseline that selects a random proposal from the set of all possible proposals.

We observed that human-LM dialogues achieved lower scores, despite being longer than human-human dialogues. Qualitatively, participants had a frustrating experience with the LM assistant. In initial trials, we observed that the LM assistant would often get "stuck" making similar proposals repeatedly, leading the dialogue to fail to make progress. In these cases, users were instructed to accept the best proposal they could get, but dialogues likely could have been much longer. We discuss particular failure modes of LM assistants further in §7. Overall, these results suggest that present-day LLMs are far from serving as useful assistants, despite the appearance of helpfulness.

6.2 SELF-PLAY

Since human evaluation is expensive and frustrating, we evaluate whether models can collaborate with each other in self-play as a cheaper proxy for the benchmark, prompting another model to play the role of the user. We prompt models with the same randomly generated task instances as the human-human dialogues in the evaluation dataset to reduce variance, although future agents can also generally be evaluated on new random instances generated from the environment. In Figure 3, we see that self-play exhibits a similar trend to human-LM play: models achieve lower rewards and longer dialogues than both human-human and human-LM pairs, suggesting that self-play is a reasonable proxy for human evaluation. We note that self-play is a more difficult setting than human-LM play, as models also have to serve as cooperative *users*. The performance drop compared to human-LM pairs provide a sense for how much human-LM performance could be attributed to human partners, who could compensate for model failures e.g. by taking initiative to share relevant information or keeping the dialogue on track to better solutions.

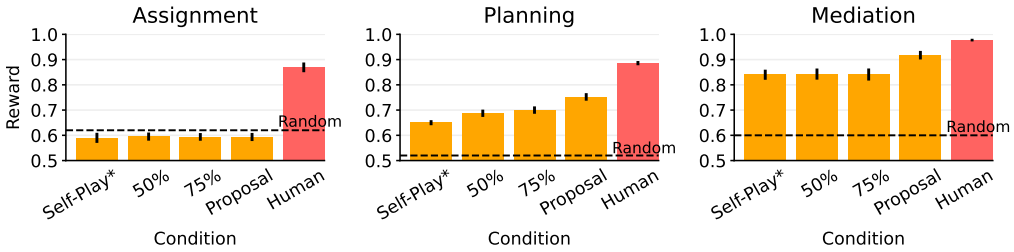

Figure 4: Prompted self-play results for all three tasks, compared to human results. For each setting, we initialize dialogues with 50% and 75% of a corresponding human game and let GPT-3 complete the dialogue. In the *proposal* setting, we prompt the model with an entire human dialogue except for the final proposal and force the model to end the game. The average score of a randomly selected proposal is shown for each task as a dashed line. (*) For reference, we also show the mean score of models in unrestricted self-play; this differs from a 0% PSP condition, because the PSP conditions bias the models to stop when the dialogue reaches the corresponding human-human dialogue length.

6.3 Prompted Self-Play

As a more nuanced proxy for human evaluation, we also propose a new mode of automatic evaluation known as *prompted self-play* (PSP), in which a given prefix of a human-human dialogue is completed with model-model play. PSP provides a more fine-grained picture of model capabilities by initializing models with a human dialogue that is already "on-track," containing information that the human-human pair has talked about already. This makes it easier to find good solutions *if* models are able to understand and reason over that information to make a proposal. Additionally, models should be able to reason over what commitments are established or information is known by the other agent to decide how to proceed from the prefix. For example, models ought to avoid asking about information already implied by previous utterances—which, in PSP, include real human utterances. Finally, prompting in this way encourages models to complete dialogues "in the style" of the human-human pair in the prefix. As a result, PSP both tests whether models can flexibly continue dialogues demonstrating different strategies (e.g. with one agent taking most of the initiative), and whether assistants can collaborate with a diverse range of humans, similar to population play and fictitious self-play evaluation (Jaderberg et al., 2019; Strouse et al., 2021).

Given a human-human dialogue from our dataset, we test how models perform if they are provided with 50% of the dialogue, 75% of the dialogue, and everything except the final proposal, and then continue the dialogue with self-play. We bias models to output dialogues that are approximately the same length as the corresponding human-human dialogue by prompting them to make their final proposal once the number of words in the dialogue exceeds the number of words in the human dialogue minus 25. Figure 4 shows average PSP performance for each task. In `Planning`, models perform better with additional human data in the prompt, suggesting that they are at least partially capable of integrating information from the human-human prefix. However, there is still a substantial gap between the *proposal* condition and human-human dialogue scores, indicating that models struggle to perform the final optimization step of choosing the best solution given the entire dialogue history. Meanwhile, in `Assignment`, models fail across all PSP conditions; this occurs because the final step of the reviewer matching task involves integrating the discussed values to compute a bipartite matching, which is difficult for models. Finally, in `Mediation`, models score well above a random baseline in all PSP conditions but do not perform better with additional human-human dialogue context, suggesting that they can meaningfully communicate about the task but don't make the optimal final proposal. In the future, tool use could potentially greatly improve performance on this task, particularly with tools that can specifically handle the optimization part of the problem.

7 Analysis

In order to quantify the strategies that humans and agents use in our tasks, we annotate individual messages in human-human and human-LM dialogues, categorizing them into dialogue acts. Quantitative analysis of dialogue acts over time are shown in Appendix D. Surprisingly, while LM assistants underperform human assistants on task performance, we observed no major differences between

the types of messages used in human-human and human-LM dialogues, suggesting that LMs are capable at imitating dialogues at least at a superficial level. To investigate why human-LM dialogues underperform, we turn to qualitative analysis, observing several classes of failure modes.

Lack of Goal-Directed Behavior Decision-oriented dialogues require models to explicitly optimize a decision objective. Critically, this requires *planning*, e.g. asking questions that will lead to discussion of decision-relevant information, or making proposals as a mechanism for gathering information. We observed that models do ask questions, but tend to ask general ones such as "Do you have any other preferences?" and sometimes slightly more specific ones such as "Do you have a price point?", but the questions are not *goal-directed* in eliciting decision-critical information. Models will also make iterative proposals, but the proposals only superficially build on each other (e.g. adding events one-by-one, and then concluding), often not improving in score. This led AI assistants to be much less efficient in their dialogues (longer, yet lower-scoring) than human assistants, who in contrast, ask questions and make proposals that help them narrow down the search space. This is unsurprising given that present-day models are not explicitly trained to optimize for task objectives beyond following the initial task instruction.

Failures of Reasoning and Grounding On `Planning`, we observed that the model would make tool queries as prompted to do so, but fail to reason over the outputs of the tool (e.g., searching for museums when the user asked to visit a museum and then outputting a proposal consisting of the search results and nothing else). Models also fail to do the optimization step of the proposal (as supported by our PSP results): proposals are often only slightly better than random, and do not improve drastically over the course of the dialogue. Additionally, models would often fail to ground the information they were given, e.g., outputting hallucinated flights on `Mediation`.

Uncooperativeness Human players were often frustrated that LM assistants were uncooperative. For instance, they would fail to fulfill requests like "please add ... to the itinerary" or would ignore information provided by the user such as "I cannot make any flights on Friday," even when human players would repeatedly send these messages. LM assistants also exhibited a failure to understand *joint commitment* by verbally committing to one course of action then making a different proposal entirely. `Mediation` was particularly challenging due to the multi-party dialogue—here, the LM failed to manage the coordination amongst multiple players, sometimes making a proposal after eliciting preferences from one player without consulting the other player.

Beyond achieving a basic level of cooperation, we would hope that future LMs can exhibit more rich and adaptive behaviors. We show a human-human dialogue side-by-side with a self-play dialogue in Figure 2. We generally observe across the human dialogues that human-human pairs exhibit diverse strategies in (1) *user vs. assistant initiative*: in some dialogues, users are proactive in sharing relevant information, while in others assistants make directed queries to narrow down the set of proposals; and (2) *coordination strategies*: working incrementally from partial proposals, backtracking, and more. In contrast, self-play dialogues and utterances from the LM assistant tend to be repetitive.

8 Discussion & Conclusion

In this paper, we presented data, environments, and model baselines for a class of tasks we call *decision-oriented dialogues*. Across all task settings, current language models did not perform as well as humans, suggesting failures in their ability to communicate efficiently and reason in structured real-world optimization problems. Future work in this domain may seek to integrate tools and inference techniques which would allow language models to compute optimal decisions while maintaining their flexible communication and collaboration skills.

The ultimate goal of this line of work is to build general collaborative agents rather than those specialized to particular settings. As we develop more generally capable models, future work should evaluate whether models can *generalize* their collaborative capabilities to harder task instances and *transfer* them to related tasks. Many real-world problems may be much more complex and unstructured but involve decision-making nonetheless, ranging from choosing a gift to designing a website layout to making a life decision. We hope that our work is a step toward future assistants that can help us deliberate and make the best decisions in the range of problems we face every day.

REFERENCES

James F. Allen and George Ferguson. Human-machine collaborative planning. In *Proceedings of the 2002 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, Edinburgh, Scotland, 2002. International Joint Conferences on Artificial Intelligence Organization.

James F. Allen, Lenhart K. Schubert, George Ferguson, Peter Heeman, Chung Hee Hwang, Tsuneaki Kato, Marc Light, Nathaniel Martin, Bradford Miller, Massimo Poesio, et al. The TRAINS project: A case study in building a conversational planning agent. *Journal of Experimental & Theoretical Artificial Intelligence*, 7(1):7–48, 1995.

Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, Athul Paul Jacob, Mojtaba Komeili, Karthik Konath, Minae Kwon, Adam Lerer, Mike Lewis, Alexander H. Miller, Sasha Mitts, Adithya Renduchintala, Stephen Roller, Dirk Rowe, Weiyan Shi, Joe Spisak, Alexander Wei, David Wu, Hugh Zhang, and Markus Zijlstra. Human-level play in the game of Diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022. doi: 10.1126/science.ade9097. URL https://www.science.org/doi/abs/10.1126/science.ade9097.

Nolan Bard, Jakob N. Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H. Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, Iain Dunning, Shibl Mourad, Hugo Larochelle, Marc G. Bellemare, and Michael Bowling. The Hanabi challenge: A new frontier for AI research. *Artificial Intelligence*, 280:103216, 2020. ISSN 0004-3702. doi: 10.1016/j.artint.2019.103216. URL https://doi.org/10.1016/j.artint.2019.103216.

Daniel S. Bernstein, Shlomo Zilberstein, and Neil Immerman. The complexity of decentralized control of Markov decision processes. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, UAI'00, pp. 32–37, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1558607099.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym, 2016. URL https://arxiv.org/abs/1606.01540.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. MultiWOZ — A large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5016–5026, Brussels, Belgium, October–November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1547. URL https://aclanthology.org/D18-1547.

Giuseppe Carenini and Johanna D. Moore. Generating and evaluating evaluative arguments. *Artificial Intelligence*, 170(11):925–952, 2006. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2006.05.003. URL https://www.sciencedirect.com/science/article/pii/S000437020600066X.

Micah Carroll, Rohin Shah, Mark K. Ho, Tom Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca Dragan. On the utility of learning about humans for human-AI coordination. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/f5b1b89d98b7286673128a5fb112cb9a-Paper.pdf.

Laurent Charlin and Richard S. Zemel. The Toronto paper matching system: An automated paper-reviewer assignment system. In *Proceedings of the ICML Workshop on Peer Reviewing and Publishing Models (PEER)*, 2013. URL http://www.cs.toronto.edu/~lcharlin/papers/tpms.pdf.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. PaLM: Scaling language modeling with pathways. *Computing Research Repository (CoRR)*, arXiv:2204.02311, 2022. URL https://arxiv.org/abs/2204.02311.

Allan Dafoe, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R. McKee, Joel Z. Leibo, K. Larson, and Thore Graepel. Open problems in cooperative AI. *Computing Research Repository (CoRR)*, arXiv:2012.08630, 2020. URL https://arxiv.org/abs/2012.08630.

Daniel Fried, Nicholas Tomlin, Jennifer Hu, Roma Patel, and Aida Nematzadeh. Pragmatics in grounded language learning: Phenomena, tasks, and modeling approaches. *Computing Research Repository (CoRR)*, arXiv:2211.08371, 2022. URL https://arxiv.org/abs/2211.08371.

Dylan Hadfield-Menell, Stuart J. Russell, Pieter Abbeel, and Anca Dragan. Cooperative inverse reinforcement learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/c3395dd46c34fa7fd8d729d8cf88b7a8-Paper.pdf.

He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL)*, pp. 1766–1776, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1162. URL https://aclanthology.org/P17-1162.

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. Decoupling strategy and generation in negotiation dialogues. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2333–2343, Brussels, Belgium, October–November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1256. URL https://aclanthology.org/D18-1256.

Johannes Heinrich, Marc Lanctot, and David Silver. Fictitious self-play in extensive-form games. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37 of *Proceedings of Machine Learning Research*, pp. 805–813, Lille, France, July 2015. URL https://proceedings.mlr.press/v37/heinrich15.html.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. *Computing Research Repository (CoRR)*, arXiv:2103.03874, 2021. URL https://arxiv.org/abs/2103.03874.

Eric Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 159–166, 1999.

Max Jaderberg, Wojciech M. Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castañeda, Charles Beattie, Neil C. Rabinowitz, Ari S. Morcos, Avraham Ruderman, Nicolas Sonnerat, Tim Green, Louise Deason, Joel Z. Leibo, David Silver, Demis Hassabis, Koray Kavukcuoglu, and Thore Graepel. Human-level performance in 3D multiplayer games with population-based reinforcement learning. *Science*, 364(6443):859–865, 2019. doi: 10.1126/science.aau6249. URL https://www.science.org/doi/abs/10.1126/science.aau6249.

Hong Jun Jeon, Smitha Milli, and Anca Dragan. Reward-rational (implicit) choice: A unifying formalism for reward learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 4415–4426. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/2f10c1578a0706e06b6d7db6f0b4a6af-Paper.pdf.

Mike Lewis, Denis Yarats, Yann Dauphin, Devi Parikh, and Dhruv Batra. Deal or no deal? end-to-end learning of negotiation dialogues. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2443–2453, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1259. URL https://aclanthology.org/D17-1259.

Belinda Z. Li, Maxwell Nye, and Jacob Andreas. Implicit representations of meaning in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (ACL-IJCNLP)*, pp. 1813–1827, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.143. URL https://aclanthology.org/2021.acl-long.143.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1192–1202, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1127. URL https://aclanthology.org/D16-1127.

Jessy Lin, Daniel Fried, Dan Klein, and Anca Dragan. Inferring rewards from language in context. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL)*, pp. 8546–8560, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.585. URL https://aclanthology.org/2022.acl-long.585.

David G. Novick and Stephen Sutton. What is mixed-initiative interaction? In *Proceedings of the AAAI Spring Symposium on Computational Models for Mixed Initiative Interaction*, volume 2, pp. 12, 1997.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. Show your work: Scratchpads for intermediate computation with language models. *Computing Research Repository (CoRR)*, arXiv:2112.00114, 2021. URL https://arxiv.org/abs/2112.00114.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Computing Research Repository (CoRR)*, arXiv:2203.02155, 2022. URL https://arxiv.org/abs/2203.02155.

Christopher Potts. Goal-driven answers in the Cards dialogue corpus. In *Proceedings of the 30th West Coast Conference on Formal Linguistics (WCCFL)*, pp. 1–20. Cascadilla Proceedings Project, 2012.

Dorsa Sadigh, Shankar Sastry, Sanjit A. Seshia, and Anca D. Dragan. Planning for autonomous cars that leverage effects on human actions. In *Proceedings of Robotics: Science and Systems (RSS)*, Ann Arbor, Michigan, June 2016. doi: 10.15607/RSS.2016.XII.029. URL https://doi.org/10.15607/RSS.2016.XII.029.

Maarten Sap, Ronan LeBras, Daniel Fried, and Yejin Choi. Neural theory-of-mind? On the limits of social intelligence in large LMs. *Computing Research Repository (CoRR)*, arXiv:2210.13312, 2022. URL https://arxiv.org/abs/2210.13312.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Computing Research Repository (CoRR)*, arXiv:2302.04761, 2023. URL https://arxiv.org/abs/2302.04761.

David Schlangen. Grounded agreement games: Emphasizing conversational grounding in visual dialogue settings. *Computing Research Repository (CoRR)*, arXiv:1908.11279, 2019. URL http://arxiv.org/abs/1908.11279.

Semantic Machines, Jacob Andreas, John Bufe, David Burkett, Charles Chen, Josh Clausman, Jean Crawford, Kate Crim, Jordan DeLoach, Leah Dorner, Jason Eisner, Hao Fang, Alan Guo, David Hall, Kristin Hayes, Kellie Hill, Diana Ho, Wendy Iwaszuk, Smriti Jha, Dan Klein, Jayant Krishnamurthy, Theo Lanman, Percy Liang, Christopher H. Lin, Ilya Lintsbakh, Andy McGovern, Aleksandr Nisnevich, Adam Pauls, Dmitrij Petters, Brent Read, Dan Roth, Subhro Roy, Jesse Rusak, Beth Short, Div Slomin, Ben Snyder, Stephon Striplin, Yu Su, Zachary Tellman, Sam Thomson, Andrei Vorobev, Izabela Witoszko, Jason Wolfe, Abby Wray, Yuchen Zhang, and Alexander Zotov. Task-oriented dialogue as dataflow synthesis. *Transactions of the Association for Computational Linguistics (TACL)*, 8:556–571, September 2020. ISSN 2307-387X. doi: 10.1162/tacl_a_00333. URL https://doi.org/10.1162/tacl_a_00333.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3): 339–374, 2000. URL https://aclanthology.org/J00-3003.

D. J. Strouse, Kevin McKee, Matt Botvinick, Edward Hughes, and Richard Everett. Collaborating with humans without human data. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 14502–14515. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/797134c3e42371bb4979a462eb2f042a-Paper.pdf.

Alane Suhr, Claudia Yan, Jack Schluger, Stanley Yu, Hadi Khader, Marwa Mouallem, Iris Zhang, and Yoav Artzi. Executing instructions in situated collaborative interactions. 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. 2119–2130, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1218. URL https://aclanthology.org/D19-1218.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMA: Open and efficient foundation language models. *Computing Research Repository (CoRR)*, arXiv:2302.13971, 2023. URL https://arxiv.org/abs/2302.13971.

Takuma Udagawa and Akiko Aizawa. A natural language corpus of common grounding under continuous and partially-observable context. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 33(01):7120–7127, July 2019. doi: 10.1609/aaai.v33i01.33017120. URL https://ojs.aaai.org/index.php/AAAI/article/view/4694.

Adam Vogel, Max Bodoia, Christopher Potts, and Daniel Jurafsky. Emergence of Gricean maxims from multi-agent decision theory. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 1072–1081, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL https://aclanthology.org/N13-1127.

Douglas Walton and Erik C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. SUNY Press, 1995.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *Computing Research Repository (CoRR)*, arXiv:2201.11903, 2022. URL https://arxiv.org/abs/2201.11903.

Wei Wei, Quoc Le, Andrew Dai, and Jia Li. AirDialogue: An environment for goal-oriented dialogue research. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3844–3854, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1419. URL https://aclanthology.org/D18-1419.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

A Related Work

Task-Oriented Dialogue Our work may be viewed as an extension of task-oriented dialogue, where a system must assist a user with accomplishing a goal, such as hotel booking or calendar scheduling (Budzianowski et al., 2018; Wei et al., 2018; Semantic Machines et al., 2020). Most task-oriented dialogue settings evaluate systems with coarse metrics such as success rate (e.g. at returning hotel information requested by a user) or word overlap with human-human dialogues. In contrast, our tasks are grounded in underlying optimization problems, where the quality of the final solution provides a richer measure of communicative success. Additionally, agents must *take initiative* to share and query information, similar to early work on task-oriented dialogue in mixed-initiative settings (Novick & Sutton, 1997; Horvitz, 1999) such as TRAINS (Allen et al., 1995) and TRIPS (Allen & Ferguson, 2002), in which users had to collaborate with a computer agent in order to solve planning problems.

Grounded & Goal-Directed Dialogue Many prior works have studied grounded and goal-directed dialogue more broadly, where agents use language to communicate and achieve goals, often in a setting that involves multimodal, situated, or external (non-linguistic) knowledge. Examples of such tasks include Cards (Potts, 2012; Vogel et al., 2013), CerealBar (Suhr et al., 2019), MutualFriends (He et al., 2017), and OneCommon (Udagawa & Aizawa, 2019), as well as partially-cooperative negotiation dialogue tasks such as Deal or No Deal (Lewis et al., 2017) and Craigslist Bargaining (He et al., 2018). In many of these tasks, including ours, the nature of the multi-agent collaboration requires that agents not only find the optimal solution, but also reach mutual understanding (a setting termed "grounded agreement games"; Schlangen (2019)), eliciting rich coordination and communication strategies in language. Other work has studied how agents can explicitly model user preferences to more effectively persuade or argue that a course of action is desirable (Carenini & Moore, 2006). Decision-oriented dialogue shares elements with many of these tasks, with a focus on fully-cooperative problems in real-world decision domains and a formalism to characterize the underlying inference problem in these settings.

Large Language Models Our goal of building task-general dialogue agents motivates the use of large language models (LLMs) such as GPT-3 (Brown et al., 2020; Ouyang et al., 2022), PaLM (Chowdhery et al., 2022), or LLaMA (Touvron et al., 2023). Current-era language models are known to struggle with aspects of our tasks, such as mathematical reasoning (Hendrycks et al., 2021), explicit state tracking (Li et al., 2021), pragmatics (Fried et al., 2022), and theory of mind (Sap et al., 2022). However, recent work in scratchpad prompting (Nye et al., 2021), chain-of-thought reasoning (Wei et al., 2022), and external tool use (Schick et al., 2023) has sought to address these problems. We build baseline models with similar approaches in our setting. While LLMs can perform reasonably well in some of our settings, we show that they cannot consistently handle dialogues with complex decision problems as well as humans.

Human–AI Collaboration Our task may also be viewed as a cooperative multi-agent setting (Dafoe et al., 2020). Research in human-AI collaboration and multi-agent reinforcement learning has also formalized tasks that require collaborating strategically with other agents on a shared goal, through tasks such as Overcooked (Carroll et al., 2019), Hanabi (Bard et al., 2020), and Diplomacy (Bakhtin et al., 2022). Our evaluation methodology is adapted from these tasks, where methods like population play and fictitious self-play are often used as proxies for human evaluation in addition to self-play (Heinrich et al., 2015; Strouse et al., 2021). In human–AI collaboration, cooperative tasks have been formulated in game-theoretic terms where agents use signals from the user such as demonstrations, feedback, or language (Jeon et al., 2020; Lin et al., 2022) to explicitly optimize for assistive behavior (Hadfield-Menell et al., 2016; Sadigh et al., 2016). In our work, we are similarly interested in formalizing settings where agents should explicitly optimize for human assistance in the course of dialogue.

B Environment Details

B.1 Formalization

Here, we describe how each task is formalized as an instance of a decision-oriented dialogue problem.

Assignment We represent W as a bipartite graph and restrict valid proposals $W' \subseteq W$ to be bipartite matchings. Edge weights $w(e_{ij})$ represent reviewer-paper affinities, and each agent observes some subset of these weights. Agents have symmetric information and roles in this task: their observations are drawn from the same distribution, and either agent can propose a decision.[3]

Planning We formalize this task by constructing W as a fully-connected graph over the locations, where edge weights represent travel times (and the preference over edge weights is negative). The user has preferences θ about which sites to visit, a budget, and a preference for reducing travel time. Meanwhile, the assistant has access to a database of sites, along with information about their cost, location, and amenities (e.g., outdoor seating). Unlike reviewer matching, this task exhibits asymmetry of information: the assistant has information about vertex features and edge weights, while the user only has information about their own preference vector θ. Additionally, only the assistant can make proposals that the user must accept or reject. Due to the budget constraint, the prescribed itinerary length, and the preference to minimize travel, this domain involves aspects of the knapsack problem, subset-selection problems, and the traveling salesman problem.

Mediation In the underlying optimization problem, the world state W can be modeled as a complete n-partite graph, where the vertices associated with each user are their flight options. Any two flights for different users are connected by an edge, whose weight indicates how compatible the flights are (i.e., whether they arrive at similar times). Vertex weights are derived from the users' calendars, with important meetings creating a preference against flights (vertices) that conflict with them. The goal is to select a flight for each user so that the induced subgraph W' (with n vertices and $\binom{n}{2}$ edges) has high total weight. This task has asymmetric roles and information.

B.2 Procedural Generation Details

Here, we describe how games are procedurally generated, omitting minor details that we implement for task realism. To fully reproduce our environments, please see our code release.

Assignment To create an environment instance, each cell of the $k \times k$ table of reviewer-paper affinity scores is sampled from Uniform$[0, 100]$ (with $k = 8$ in our experiments). To ensure that communication is necessary to do well, we reject a random game unless the optimal score with the agents' pooled knowledge is ≥ 1.25 times as good as the score that either player would achieve with their own information if they replace unknown cells with the average value (50). We scale values by a random scalar sampled from Uniform$[1, 10]$.

Planning To generate contexts for the dialogue, we create a seed list of 39 site names and locations. Each site is one of the following categories: restaurants, bars, cafes, sights (museums and landmarks), outdoor (parks), or shopping.

To create an environment instance, we randomly shuffle the locations of the sites and randomize their features. Each site has five nonzero random features, out of the following list (some of which only apply to some categories):

- Rating (categorical)
- Has parking (bool)

[3] There are many ways we could have made the task more realistic. Rather than reveal each score either perfectly or not at all, we could reveal some amount of noisy evidence about the score. Alternatively, each score could be a function of underlying features—for example, the dot product of the paper's topic vector and the reviewer's topical-expertise vector. We could then selectively reveal evidence about these features—"Alice is an expert on Botany"—rather than about edge weights.

- Has takeout (bool) [restaurants only]
- Touristy (bool)
- Cuisine (categorical) [restaurants only]
- Good for kids (bool) [restaurant, cafe, museum, landmark, park, shop only]
- Accepts reservations (bool) [restaurants only]
- Open late (bool)
- Good for groups (bool)
- Ambience (categorical) [restaurant, cafe, bar]
- Outdoor seating (bool) [restaurant, cafe, bar]
- Vegetarian options (bool) [restaurant, cafe]
- Vegan options (bool) [restaurant, cafe]
- Live music (bool) [restaurant, bar]
- Has Wi-Fi (bool) [cafe]
- Alcohol type (categorical) [bar]
- Viewpoint (bool) [park]

We procedurally generate preferences from the user from the following types:

- Feature: a preference over the value of one of the features above
- Want to go: a preference to go to a specific site or set of sites
- Price: a preference to keep the budget less than some fixed amount
- At least one site of type: a preference to go to at least one site of some type (e.g., to visit at least one museum)
- Distance: a (negative) preference per unit traveled between sites

Each of these preferences is parameterized and randomized on every environment instance. Every user has a price and distance preference; the other preferences are sampled with some probability up to a total of P preferences ($P = 10$ in our experiments). We specifically exclude preference configurations that are un-intuitive (e.g., a preference for places that do not have takeout). We template natural language descriptions for each preference to present to the user.

Mediation To create an environment instance, we generate a random calendar for each user. For each 30-min slot between 9am–8pm during a 3-day period, if the slot is still free, we add an event with probability $p_{\text{event}} = 0.35$, selecting the event duration uniformly at random from {30 min, 60 min, 2 hr, 4 hr}. $f_{\text{shared}} = 0.75$ of these events are selected to be shared events that both the assistant and user can see; the remainder are private events that only the user can see. The importance of each event is sampled from Uniform$[1, 10]$.

We generate a set of $F = 30$ flights for each user with a random start time in the 3-day period, sampling a duration (in hours) from Uniform$[1, 10]$. Flight prices for each user i are sampled from $\max(50, \mathcal{N}(\mu_i, \sigma_i))$ to ensure that flight prices a user sees are realistically around the same value, and the parameters of the distribution $\mu = \sigma$ are sampled from Uniform$[50, 1000]$. We generate a price preference weight $\theta_{\text{price}} \sim$ Uniform$[-20, -1]$ and preference per 3-hour difference in arrival between the two users' flights $\theta_{\text{arrival}} \sim$ Uniform$[-10, -1]$ (for every 3 hour difference between their flight times, deduct θ_{arrival}).

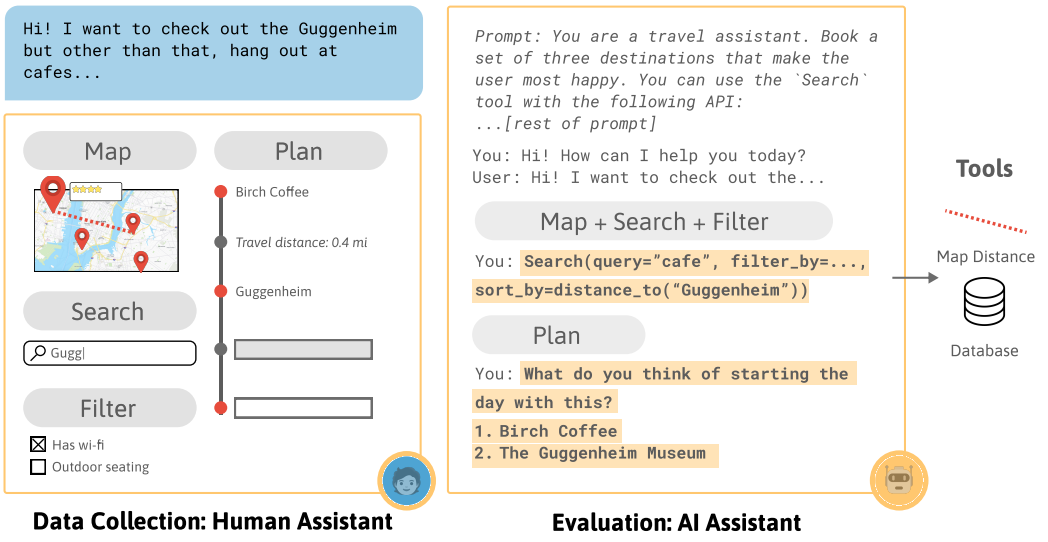

Figure 5: Data collection and evaluation frameworks. To collect human-human dialogues, we built web interfaces that allow humans to play either the User or Assistant role for each task. When evaluating how well an LLM can act as an assistant, we linearize information from the web interface into a text prompt and provide additional tools that let the language model access information that cannot fit within its context window. This figure shows just the Assistant role, for one task.

C Data Collection Details & Statistics

Human players from Mechanical Turk were vetted via a pre-qualification survey. Data collection was run in multiple dyads, with cooperative players from each dyad (as judged manually) being invited to participate in followup rounds of data collection. Workers are bonused up to $2.00 in tiers by how close they get to the best possible proposal. In Table 1, we show the data statistics for human-human dialogues. In Figures 6–8, we show example dialogues for each task. As depicted in Figure 5 for `Planning`, humans (left) receive the same information as language models, but in a graphical interface instead of in text.

	Dialogues	Messages (μ)	Words (μ)	Proposals (μ)	Time (μ)
Assignment	134	18.4 ± 1.1	169.3 ± 10.9	1.7 ± 0.1	8m 9s
Planning	114	9.0 ± 0.4	141.9 ± 6.5	3.0 ± 0.1	10m 56s
Mediation	162	10.9 ± 0.5	119.0 ± 5.7	2.8 ± 0.2	7m 15s
All Domains	409	12.8 ± 0.5	141.8 ± 4.7	2.5 ± 0.1	8m 19s

Table 1: Data statistics for human-human dialogues. We collect a total of 409 dialogues, resulting in 5253 messages and 58K words across domains. Dialogues for each setting are roughly the same number of words on average.

Assignment

A: I want to make a case for Joseph being Llama
B: Hey there! That sounds good to me, my information is limited on LLama. I would like to propose that we assign the following: / Daniel - Electra / Noah - BLEU / Sofia - SWAG (most confident) / I also see Sofia being solid at some other options too if we should decide we need her to be flexible.
A: Okay lets set that up for now / who do we have for Glove / Ethan is pretty solid
B: Lets stick with Ethan for now then.
A: he does better at Quac however / unless you have a good Quac option / Ethan can Glove or Quac
B: The best person I see for Quac is Noah, but im much more confident having him on BLEU. He is middling at quac / we can put Ethan on Quac
A: Ethan Quac selected for now
B: Sofia is above average at Glove, its just that I have supreme confidence in her on SWAG.
A: okay looks like we dont have much for Glove so we go back / I have no info on Glue for any unselected person
B: I dont have any good options for Quac besides going with your information on Ethan, what about you?
A: Ethan Quac for now / next best is sofia and she is only slightly worse
B: Literally the only info I have for Glue is that Ethan is not so good at it
A: Joseph is good at it / only info I have that you dont
B: Lets lock in Joseph for now on it, I see Joseph being not so good at anything on my end. / oh wait, you said Joseph on LLama
A: yes / thats ideally super lock
B: lets go back to that / do you have any good options for swag besides sofia?
A: no they are all bad
B: trying to see if we might want to move her to roberta since I only see bad options for roberta. / Ok then maybe we will have to take a gamble with roberta / Going to throw a proposal together, feel free to reject once youve absorbed that info.
A: do you have any decent options for Andrei Daniel or Morgan / Something mid that isn't taken / I dont have anything like that for them
B: Daniel is locked on Electra, but Morgan and Andrei I only see 4/10 levels of confidence. / Do you see anything for Ava? / Andrei could take llama if he really needed to
A: for the 3 empty, no / and she is bad at everything taken / Is it a gamble on the last 3
B: do you see anything shes good at thats not taken? Just curious, but most likely we have to gamble with her
A: I dont have anything good at them / Yeah the 3 with 0 selection are all red only for me
B: Ok here is the gamble version of my proposal haha.
B: `BLEU: Noah Wilson, Electra: Daniel Nguyen, GLUE: Andrei Petrov, GloVe: Ava Li, LLaMA: Joseph Santos, QuAC: Ethan Smith, RoBERTa: Morgan Reed, SWAG: Sofia Patel`
A: `accept`

Figure 6: Example human-human dialogue for `Assignment`. Forward slashes denote the boundary between multiple messages sent sequentially without a response from the other player.

Planning

User: Greetings!
Agent: Hello! Is there anything in particular that you would like to see in the city? Museums, food options, tourist attractions?
U: Yes, I would love to go to a concert at an amazing venue in town.
A: I don't have any concerts but there is live music at the Dockside Grill. What is your budget for your trip?
U: I'd like to keep it below $40.
A: The Dive has live music for $30. Does this look like some place you would like to go?
U: Yes.
A: Ok, we are set on The Dive. With $10 left in budget, what kind of food would you be interested in?
U: Maybe a food truck outside the Mall?
A: `[The Dive, Saul's]`
A: With your budget, the nearest food place around $10 is Sauls but it is a far. There is a closer option with The Dockside Grill but it is $20 so it will be slightly over budget. Do you prefer either?
U: Sauls. And let's find an outdoor attraction I can take kids to, like a park.
A: `[The Dive, Saul's, Garden of Wonders]`
A: How does the Garden of Wonders sound?
U: `accept`

Figure 7: Example human-human dialogue for `Planning`.

Mediation

Room 1 (User 1, Assistant)	**Room 2 (User 2, Assistant)**
U1: Hello, I've got a busy schedule but need a flight booked. I'm thinking of something Friday evening after 5 PM, but may be able to flex depending on my meetings	
A: There is a slight overlap but we have a flight on 6/2 4:38 PM — 6:38 PM for 1367$	
	U2: Hi, I'd like to get a flight, preferably something that doesn't conflict with my important meetings. Is there any redeye flights on Friday, that might work well for me.
	A: Yes, there is a flight on 6/2 5:28 PM — 2:28 AM that overlaps with your meeting from 2:30 to 6:30 for 629$
U1: That does overlap with a very important meeting, so I wouldn't want to miss any of that. Could we explore other options but I may reconsider if it means I can arrive at the same time as my friend :)	
	U2: That's pretty pricey, maybe I could do a redeye on Wednesday and just miss some of my meeting that evening?
	A: Would 6/1 6:49 PM — 3:49 AM for 50$ work? It would overlap with an 8 to 10 PM meeting.
	U2: That meeting is pretty important, but if it gets me in close to my friend and is that cheap it could be worth it. Go ahead and book it.
A: Are you open to 6/1 7:50 PM — 9:50 PM for $50, it would slightly overlap with the 7:30 PM meeting however it is closer to your friend.	
U1: I think I may be willing to give up that meeting	
A: `Southwest | $50 | [6/01 07:50 PM] - [6/01 09:50 PM]`	**A:** `Alaska | $50 | [6/01 06:49 PM] - [6/02 03:49 AM]`
U1: `accept`	**U2:** `accept`

Figure 8: Example human-human dialogue for `Mediation`.

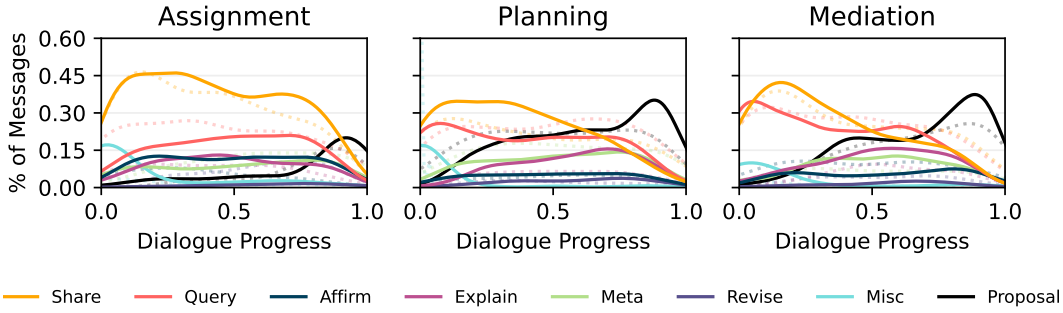

Figure 9: Kernel density estimates of message types in human-human (solid) and human-LM (dotted) dialogues plotted against their position within a dialogue. Message types were annotated using few-shot prompting with GPT-4 and validated by manual human annotation.

D Quantitative Dialogue Act Analysis

Humans may use a wide range of communicative strategies to negotiate with one another, optimize for their goals, and make decisions (Walton & Krabbe, 1995). In order to quantify the strategies that may be useful in our tasks, we used GPT-4 to annotate human-human and human-LM dialogues at the level of individual messages. Based on manual inspection of a small set of games, we devised a list of message types: (1) *share*, in which agents provide information about their preferences; (2) *query*, in which agents ask each other for information; (3) *affirm*, in which agents agree with each other and/or ground incoming messages; (4) *explain*, in which agents provide justification for a previous message or action; (5) *meta*, in which agents engage in discussion about high-level strategies or meta-game details; (6) *revise*, in which agents correct earlier statements; (7) *miscellany*, which includes other messages such as greetings; and (8) *proposal*, which denotes a formal proposed decision. These categories were roughly based on standard course-grained dialogue act taxonomies (e.g., Stolcke et al., 2000), which often contain statements, queries, revisions, agreements, and a miscellany category; we then added types such as *meta* based on the idiosyncrasies of our problem domain.[4] Each message may have multiple message types. We prompted GPT-4 to generate annotations for each message using two hand-annotated example dialogues.[5]

We provide a breakdown of message types over the time-course of dialogues in Figure 9. As expected, many interactions begin with greetings, which is evidenced by a spike in the *miscellany* category at the beginning of all three plots; meanwhile, complete dialogues end in *proposal* actions. Most dialogues are focused on exchanging information: of the message types, we find that agents most commonly *share* or *query* for information. In the `Assignment` task, agents send twice as many *share* messages as any other type of message, often sending information about individual cells in their observed tables. One common strategy involves both players sharing all observed information and then making a decision at the end of the game. This approach is most tractable in `Assignment`, where players have a relatively small observation space. However, this strategy leads to exceptionally long dialogues, even in `Assignment`, and is not the most common approach. Meanwhile, in the `Planning` and `Mediation` tasks, which have asymmetric information and roles, agents are more likely to *query* for information or engage in *meta*-game discussion in order to learn what information the other agent can see.

[4] *Meta* messages reference the task but don't provide information about the underlying graph, e.g., "I have sent a proposal" or "Hello! I can definitely help you find a cheap flight." *Explain* messages justify some previous or future action, e.g., "I think a museum would be great for the kids" after sending a proposal that includes a museum. *Proposals* are task-specific formal messages, e.g., `[Mad Seoul, Riverside Trail, Garden of Wonders]` in `Planning`.

[5] We performed a manual human validation on 106 messages (across six dialogues) and found that human labels matched GPT-generated labels on 88% of messages. On the 13 instances where human labels differed, we found 7 of the GPT-generated labels to be reasonable and correct alternatives.