A POMDP Dialogue Policy with 3-way Grounding and Adaptive Sensing for Learning through Communication

Maryam Zare and **Alan R. Wagner** and **Rebecca Jane Passonneau**
Pennsylvania State University, University Park
{muz50, azw78, rjp49}@psu.edu

Abstract

Agents to assist with rescue, surgery, and similar activities could collaborate better with humans if they could learn new strategic behaviors through communication. We introduce a novel POMDP dialogue policy for learning from people. The policy has 3-way grounding of language in the shared physical context, the dialogue context, and persistent knowledge. It can learn distinct but related games, and can continue learning across dialogues for complex games. A novel sensing component supports adaptation to information-sharing differences across people. The single policy performs better than oracle policies customized to specific games and information behavior.

(a) Pepper

(b) A Quarto win

Pepper: Where else can I put these pieces and still come out with a win?
- Informative Answer: You can lay those pieces in any of the three remaining empty rows and still win.
- Uninformative Answer: It'll be a win if they are put down on the first row.

Figure 1: A question about Quarto produced by our dialogue policy. Two answers taken from a corpus we used illustrate a more versus less informative answer.

1 Introduction

Agents that drive cars, assist in surgery, or support elder care could more easily collaborate with us if they could ask us questions to learn to operate a new type of vehicle, perform a related type of surgery, or help with a different daily life activity (Higgins et al., 2021). We present a novel POMDP policy with the following distinctive communicative abilities. It can apply general communicative skills to similar but different learning goals, pursue more complex learning objectives across multiple dialogues, and adapt to human variation in information sharing. To motivate the latter, we also show that human subjects vary in the information they share, with corresponding differences in the optimal questioning strategy.

We build on previous work in statistical POMDP policies for dialogue management in several ways (Wen et al., 2017; Young et al., 2013). As in most such systems, natural language utterances are mapped to a domain-dependent semantic representation. Similarly, we train offline using a simulator instead of a human dialogue partner, common since (Schatzmann et al., 2007), and we update the POMDP state representation using belief state tracking (Gao et al., 2019). However, we make significant extensions: 1) context-specific communicative actions are generated using a compositional meaning representation (MR); 2) each communicative action has a parent action for choosing the context, and a child action for what to say in that context, controlled by a hierarchical policy; 3) communicative actions have three-way grounding in the shared physical context, the dialogue context, and the agent's persistent knowledge; 4) learning builds upon generic knowledge by specializing to specific learning tasks; and 5) a sensing component tracks the flow of information, or *information dynamics*, between the agent and interlocutor. We refer to the adaptive policy with three-way grounding as 3GA. The following section is an extended example that illustrates 1) through 4). Here we illustrate 5) and give full explanation in section 4.

Grice proposed that co-operative conversationalists are as informative as possible (Grice, 1975). We find, however, that people vary in informativeness, and the 3GA policy adapts its questioning strategy accordingly. Figure 1 illustrates different answers to the same question from the 3GA policy,

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 6796 - 6809

given a demonstrated way to win the board game Quarto.[1] The first answer is highly informative while the second is less so. The policy generates more open-ended questions, more *yes-no* questions, or more requests for board demonstrations, in proportion to the informativeness of respondents.

We train the POMDP on three n-in-a-row games, and assess performance by how much game knowledge the agent acquires. First, we measure knowledge acquisition in dialogues with a simulator, where we control for information quantity in the simulator's answers to questions. 3GA performs better than oracle policies trained for each specific condition of game and information quantity, and generalizes to unseen games. Second, an exploratory study confirms that people vary in informativeness, and shows that 3GA continues to learn a complex game in a succession of dialogues.

2 Example Dialogue

Games are a common test bed for agents that learn through communication (Kaiser, 2012; Kirk and Laird, 2019; Zare et al., 2020; Ayub and Wagner, 2018). As in Zare et al. (2020); Kirk and Laird (2019), we adopt a formal meaning representation (MR) to express the agent's communicative intentions, and its interpretations of utterances. Further, we use extensive-form game trees to represent the agent's game knowledge, cf. (Zare et al., 2020; Ayub and Wagner, 2018). In a game tree, nodes are game states, and edges at a given level of the tree are the actions available to the player (odd levels) or the opponent (even levels). Leaves represent terminal states of a win, loss or draw for the player. Game search algorithms can then be applied to the declarative knowledge in the agent's game tree, for the agent to choose moves while playing the game (cf. (Zare et al., 2020)). To support the ability to learn different games, we create a generic representation for two-person zero-sum games. Some of the questions 3GA generates aim to specialize this generic knowledge for a specific game like Quarto. In contrast to previous work on learning through communication, we use a statistical POMDP for dialogue management. The trained policy chooses what question to ask in each dialogue state, and context-specific questions are generated on the fly. Figure 2 illustrates how responses to the agent's questions are added to an evolving game tree.

[1] An n-in-a-row game, it is played on a 4×4 grid with pieces differentiated by height, shape, color and hollowness.

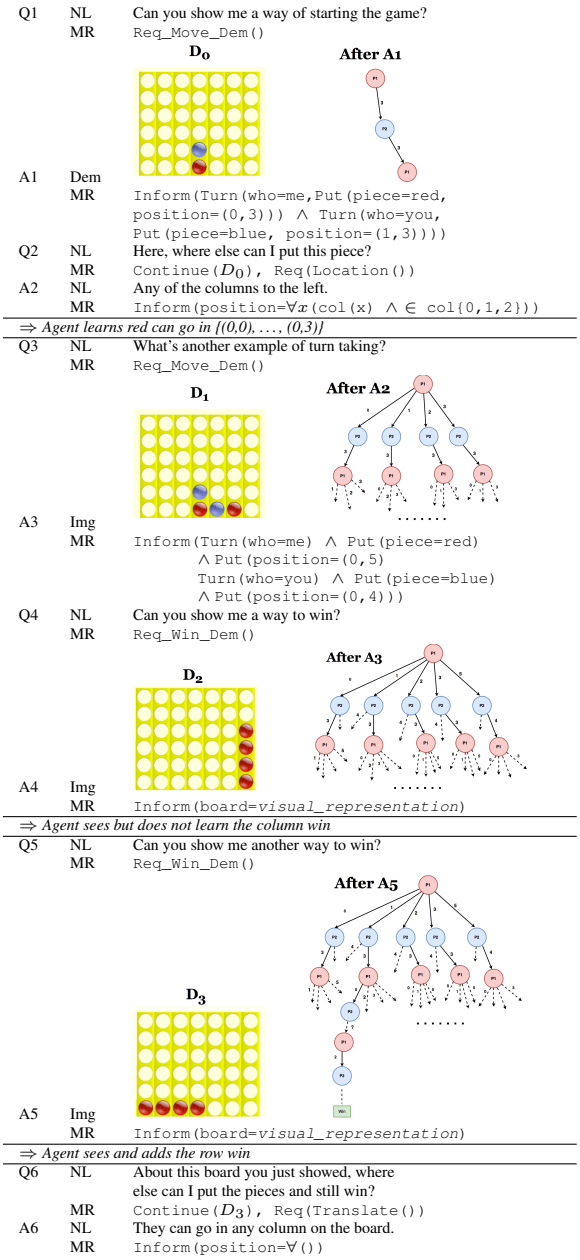

Figure 2: An example dialogue with questions (Q) from the agent and answers (A) from the interlocutor. Knowledge of each move in a win sequence is a precondition for adding the path to the tree; cf. tree AFTER A3 vs. tree AFTER A4.

To illustrate 3-way grounding of utterances to a representation of the physical world, to abstract game knowledge, and to the discourse context itself, Figure 2 shows part of a dialogue to learn Connect Four.[2] Text-based, multi-modal dialogues take place in a graphical user interface. Natural language questions and responses are shown here with corresponding meaning representations (MRs). Utterances are grounded in the physical world of game

[2] See Appendix A for descriptions of the games we use.

boards, pieces and moves via elements of the MR, as in `Put(piece=red,position=(0,3))` from A1. They are also grounded by initializing a new game tree at the beginning of a dialogue, as in the tree labeled AFTER A1, or by adding to it. Finally, each utterance is in reference to a specific game board demonstration, or local context.

In Q1, the agent asks how the game starts, and the response is a visual demonstration. The game tree labeled AFTER A1 shows a root node representing an empty board state, an edge (move) to a successor game state where the player (red) has put a piece in column 3 (numbering starts in 0), and a subsequent edge to a state where the opponent (blue) added another piece to column 3. As the dialogue progresses, the agent grounds new ways to make moves or to win. If it has sufficient grounding (knowledge about moves), the agent adds paths to the tree that terminate in wins. In Q4, the agent asks for a way to win, and is shown a board with four discs in the last column. Because the individual moves have not been grounded as game knowledge, the agent's belief state (not shown) is updated, but not its game tree knowledge. In contrast, after a different win demonstration in A5, the agent can add a complete path to a win state (green box in tree), because in A1-A2 each move in that path was grounded. At the dialogue's end, the size of the game tree is a measure of the knowledge gained.

3 Learning and Remembering Games

For the agent to learn different n-in-a-row games, we formalized generic properties of two-person zero-sum games. Game trees for different games have different game states and actions from each state. In Connect Four, for example, a player has 7 opening moves, while in our modified version of Gobblet, with 4 game piece sizes and a 4×4 grid, there are 64 opening moves. To learn a new game from scratch, the agent must learn the new state space, action set and successor function, or it will not be able to retain persistent knowledge.

The agent's initial game knowledge consists of an unspecified start state G_0, a place-holder for the set of game states Γ, a place-holder for the set of actions (game moves) M, and a procedure to build a `successor` function that defines the successor states γ_{i+1} given a move $\mu_i \in M$ taken in a state $\gamma_i \in \Gamma$. During a dialogue, the agent's choice to ask about a new move versus about a new win path is determined by the trained dialogue policy. For a given game, after the agent has learned at least one game state and some actions available in that state, it can start constructing a game tree for that game.

We use the term information for facts presented to the agent through demonstration or language, and knowledge for incorporation of information into a persistent knowledge representation that can be used for reasoning. Persistent knowledge store is useful for learning games that are too complex to learn in one short dialogue. It is also useful to be able to turn to a new dialogue partner if the current one is uninformative. The difference in an update to the agent's beliefs versus its knowledge is explained in section 4.

4 3GA POMDP with Adaptive Sensing

A Markov Decision Process (MDP) formalizes an agent's step-by-step decision making to reach a goal in an uncertain environment as a tuple $< S, A, T, R, s_0 >$, where S is the set of **states**, A is the set of the agent's **actions**, T is the **transition model** consisting of a probability distribution over successor states s_{i+1} given an action a_i taken in s_i, R is a **reward function** for each action, and s_0 is the **initial state**. An MDP dialogue agent's communicative actions are chosen by a policy π that maps dialogue states to the actions that maximize the discounted, cumulative reward over time. In a Partially Observable MDP (POMDP), states are not fully observed. Here, as in previous statistical POMDP dialogue management, S consists of the agent's belief states that represent the agent's uncertain interpretations of the interlocutor's utterances, A is the agent's communicative actions, and the reward R is a trade-off between a small cost per turn and metrics that encourage the agent to achieve its dialogue goal. The remainder of this section describes the following features in turn[3]:

1. A hierarchical policy for context switching;
2. Compositional communicative actions that mirror the policy structure;
3. Three-way grounding;
4. Distinct updates to beliefs versus knowledge;
5. Information dynamics to support an adaptive dialogue strategy.

The hierarchical policy creates nested communicative actions consisting of a parent action followed by a child action. The parent action, determined by a parent policy π_P, is a decision to

[3]Github code: https://github.com/mry94/SPACe

continue the local context initiated with demonstration D_j, to resume a previous context associated to $D_i, (i < j)$; or to request a new demonstration D_{j+1}. Figure 2 illustrated requests for demonstrations of new moves in Q1 and Q3, and for new ways to win in Q4 and Q5. New demonstrations are selected to be more or less informative, depending on the simulator setting, or the sensed informativeness of the human interlocutor (more below). Questions Q2 and Q6 continue the current demonstration (`Continue(`D_i`)`). As noted in the description of three-way grounding, belief states are co-indexed with the demonstrations that spawn each local context. Game board images are saved in a database, so when π_G chooses `Resume(`D_i`)`, the image for D_i can be re-displayed in the GUI. The child policy π_C chooses a question type, which completes the compound decision to ask a specific question about the selected context.

Compositional communicative actions are expressed using an MR that is a variant of first order logic, with predicate-argument structure, quantifiers, and question operators that are functions from contexts to specific questions (see complete specification in Appendix B). The previous paragraph illustrated the parent predicates that choose the context, which is passed as an implicit argument of question operators chosen by the child policy. For example, in Q2 the full MR first specifies the context as D_0, to be an argument of the entire question function, such that the embedded predicate `Req(Location())` constitutes a context-specific request for a new location on D_0. The local context-specific MR elements fall into two categories: open-ended (*wh-*) questions, and *yes/no* questions. There are five *wh-* question types (e.g., about locations of game pieces, or their properties, such as shape and color): three about moves and two about win conditions. Eleven *yes/no* question types consist of two about moves and nine about win conditions. The *yes/no* question types are more numerous because they are more specific.

The MRs are converted to English text using a natural language generation (NLG) sequence-to-sequence RNN with two hidden layers and Bahdanau attention (Bahdanau et al., 2015). A similar model converts English responses to MRs. The RNNs were trained for 15 epochs on the dialogue corpus from (Zare et al., 2020) (13K turn exchanges).[4] To ensure the agent never receives invalid answers due to inaccurate NLU, we perform automated verification on NLU output.

Invalid answers to *yes/no* questions are converted to `Unknown()`, and inconsistent information from answers to *wh-* questions is removed (e.g., non-existent board locations). Replacing this step with clarification questions is left for future work.

Three-way grounding builds on two kinds of grounding previously utilized in human-machine interaction. Symbol grounding is the relation between symbols and a representation of the physical world of objects and actions, e.g., (Pillai et al., 2021). Communicative grounding involves how information is presented and tracked during dialogue (Skantze, 2007). We include grounding of symbols in a knowledge store for representing and reasoning about knowledge, as in dialogue for learning board games (Kirk and Laird, 2019). What distinguishes our three-way grounding is that the 3GA MR serves as an interface to connect all communicative actions produced and generated by the agent concurrently to the real world, to an abstract knowledge store and reasoner, and to the evolving but temporary dialogue context.

Belief and knowledge updates take place after each turn exchange. In POMDP dialogue policies, belief state updates track the information that has been communicated to the system or agent (Young et al., 2013). In 3GA, a distinct belief vector B_i is updated for each local context D_i with information given in answers to questions that reference the current context. Each B_i has J subvectors v_{ij} representing the current state of belief about the jth predicate of the MR subset that denotes properties of game pieces or boards. For example, after A2 in Figure 2, the belief subvector for the `Location` predicate, which is length 42 for the Connect Four 6×7 rack, is updated to represent the belief that pieces can go in locations $\{0, 1, 2\}$.

Each subvector v_{ij} of the corresponding belief state B_i is updated after interpreting a verbal response or new demonstration using the method from (Wang and Lemon, 2013). With positive answers to *yes/no* questions or answers with new information, updates use equation (1), and use equation (2) for negative responses:

$$P_{v_t} = 1 - (1 - P_{v_{t-1}})(1 - P_{u_t}) \quad (1)$$

$$P_{v_t} = \big(1 - P_{v_{t-1}}\big)\big(1 - P_{u_t}\big) \quad (2)$$

[4] A Quarto corpus provided by the authors, but applicable to all our games. Training used the Adam optimizer (Kingma and Ba, 2015), with embedding size 256, 127 RNN cells, and batch size 128.

where P_{u_t} is the probability from the NLU module of the interpretation of utterance u at time t.

The game tree update is very similar to that used in (Zare et al., 2020): new edges or win paths in the belief state are added to the game tree. The game tree updates determine the reward function R for calculation of the cumulative reward as the discounted sum of rewards over time (Bellman equation). R is a weighted sum of the number of new moves, new win conditions, the strategic value (SV) of the new game knowledge, and a turn cost:

$$R = E \times \alpha_1 + W \times \alpha_2 + SV + C \quad (3)$$

where E is the number of new edges, W is the number of new paths to a terminal win, C is the turn exchange cost, and α_1 and α_2 are weights.[5] E and W balance learning new moves versus new ways to win, SV rewards some win paths over others, and C encourages dialogue efficiency. Dialogues end when the cumulative cost outweighs the gains.

Information dynamics senses the information synergy of the dialogue by monitoring *what types of question* the agent chooses, and *how much information* the interlocutor provides. As a result, the agent can adapt its strategy at each turn exchange. Previous work on MDP dialogue policies for learning games found that policies differed when trained under different fixed conditions of the amount of information a simulator provided in response to questions (Zare et al., 2020). Given a simulator that answered all questions completely, the trained MDP produced relatively more *wh-* questions. Training with a moderately informative simulator led to relatively more *yes/no* questions. At test time, the agent learned more in a dialogue if the policy matched the informativeness of the interlocutor. MDP policies, however, are not realistic given imperfect NLU, and it is not practical to depend upon an expectation of the interlocutor's informativeness. Information dynamics leads to a single adaptive policy.

Information dynamics consists of two measures updated after each turn exchange: Self Information Dynamics (SID) and Partner Information Dynamics (PID). SID stores frequency counts for the 16 categories of questions, and the number of times the agent switches between different dialogue contexts. For responses to *yes/no* questions, PID increments the counts for the three types of responses (affirmative, negative, and non-answers), and updates the probability of getting a non-answer (e.g., *I don't know*). For *"wh-"* questions, PID updates a list of the number of different win paths provided by the dialogue partner, along with the standard deviation of the list. This gives the agent an expectation of the informativeness and consistency of answers.

[5] We found the best performance with $\alpha_1 = 0.2$, $\alpha_2 = 0.25$, and $C = 2.0$. We adopt SV from (Zare et al., 2020).

5 Training 3GA

To train 3GA, we expose it to different games and different information dynamics, using the Gaussian process, Q-learning from Gašić and Young (2013). Their model had relatively few hyper-parameters, converged quickly to a local optimum ($<$ 20k epochs), and produced good results. We trained for 30k epochs, with hyper-parameters σ (noise residual) = 2, η = 3, υ = 0.15, and a polynomial kernel function. Each training epoch (one dialogue) randomly selected a game and simulator informativeness. Three games of increasing complexity were used: Connect Four, Gobblet and Quarto. Simulator informativeness had five levels: 20%, 40%, 50%, 60%, 80% or 100%.

The simulator picks answers to questions from a database that matches questions to response MRs, which the NLG module converts to English. There are different sets of MR responses for each of the 16 question categories (see Appendix B). We systematically control the simulator to have informativeness $0\% \leq x \leq 100\%$ for verbal and visual responses. For *yes/no* questions, the simulator responds `unknown` with a probability $(100 - x)\%$. For *wh-* questions, it provides $x\%$ of a complete answer. For example, A2 of Figure 2 provides all other locations to the left of column 3 where a first disc could be placed, but omits the columns to the right, thus provides only 50% of a complete answer to the question in Q2. For questions that request a new board demonstration, the simulator uses a database that pairs images with a formal representation. Demonstrations of win conditions are chosen from the top (100-x)% of an ascending sort of all win paths by the number of times it has already been presented in a board demonstration.

To compare how the policy converges across the three games, we sort the training epochs by game. The policy converges quickly for Connect Four, less quickly for Gobblet, and incompletely for Quarto. We tried weighting the random game selection by game complexity, but this degraded performance. Average dialogue length ranged from about a dozen turn exchanges for Connect Four to

IL	Base.	noPID	noSID	3GA	Oracle
Connect Four (%)					
1.0	49±36	65±2	63±7	**98±3**	**98±10**
0.5	6±5	27±9	29±11	**37±9**	**38±9**
Gobblet (%)					
1.0	23±32	47±5	40±8	**84±8**	**97±11**
0.5	20±26	21±16	22±10	**25±3**	**25±1**
Quarto (%)					
1.0	3±5	20±5	23±8	**35±11**	**30±14**
0.5	0±0	4±2	3±4	***8±2**	2±0

Table 1: Average percentage of learned win conditions across 100 trials, for 3 games $\times$ 2 informativeness levels (IL).

$\geq$ 30 for Quarto. We capped Quarto dialogues at 25, for more reasonable dialogues with humans.

6 Simulation Experiments

To test the 3GA policy in controlled conditions of informativeness, we used the simulator. In experiment 1, we examined how well the agent learned a game seen in training, under different conditions of informativeness. In experiment 2, we tested 3GA on unseen games, with no change in the training or policy. We measure the agent's game knowledge after a dialogue as the proportion of learned win paths to total win paths in a given game. Note that minimax search on zero-sum game trees is known to lead to optimal play. Previous work has shown that an agent that learns a game tree by asking questions can play against humans, and plays better when it acquired a more complete game tree (Zare et al., 2020). We compare the behavior of 3GA with a baseline, and with an oracle for each condition of game and informativeness. The baseline is identical to 3GA but omits SID_PID; the training is the same. Each of the 18 oracle policies is the baseline policy, trained only a single game and informativeness level, and tested for that condition. We also report two ablations of 3GA: noSID, and noPID.

6.1 Experiment 1: Testing on Seen Games

Our first experiment asks: *How well does 3GA perform in a single dialogue to learn a game seen in training, and how does information dynamics affect the quantity of knowledge gained?* We tested 3GA in all 3 $\times$ 6 conditions of game by informativeness. For each condition, we averaged the proportion of total learned win paths across 100 dialogues. Table 1 reports the average percentage of a complete game tree acquired in 100 trials, for three games and two informativeness levels (IL): maximum (1.0) and middle (0.5). The trend is the same across all six ILs: 3GA always learns more than the baseline (no SID_PID) and either ablated model. It learns as much (within the confidence interval) as the corresponding oracle policy for each game and informativeness, *or more*. For Quarto in the 50% informativeness condition, 3GA outperformed the oracle (8±2% vs. 2±0%). We observed that 3GA exploits information dynamics in an unintended way, producing questions that are semantically more diverse. Unlike the other games, Quarto pieces have many properties, and 3GA asks more diverse questions about game piece properties than the oracle policy. The oracle policy with IL=1 did not learn much about the game piece properties, and how they contribute to win conditions.

To provide further insight into 3GA, Table 2 reports the number of turn exchanges spent learning game moves versus complete win conditions. Recall that learning moves is a precondition to initializing or updating a game tree. With a fully informative simulator (IL=1.0), most questions are about win conditions. With lower IL, most questions are about game moves. This explains why the agent learns far fewer win conditions for Quarto or Gobblet from the less informative partner (IL=0.5) in Table 1. IL also affects dialogue length, with shorter dialogues as IL decreases. Dialogue lengths for the oracle policies and 3GA were very similar. Dialogue length for the baseline was consistently around 15 turn exchanges in all conditions.

Table 1 shows that there is still a lot to learn about Quarto after the first dialogue. Therefore, we investigated how well the agent can learn Quarto in two dialogues, and how it adapts when the simulator IL differs in the two dialogues. We tested a variant of 3GA trained specifically for Quarto, omitting the generic game knowledge.

In the first dialogue, the agent starts with a blank slate, then carries over the game and experiential knowledge of information dynamics it acquired in dialogue one as the start state for dialogue two. We evaluated the agent under three conditions across a succession of dialogues as shown in Table 3. The three conditions of simulator IL values in the two dialogues were: 1) 0.2, 1.0; 2) 1.0, 0.20; and 3) 1.0, 1.0. We examined the average game knowledge acquired over 100 trials for each condition. We also report an interesting phenomenon we observed that we refer to as *retries*.

Because the policy generates context-specific questions on the fly, based on a repertoire of question types it knows how to generate, the policy

IL	Connect Four			Gobblet			Quarto		
	Moves	Paths	Length	Moves	Paths	Length	Moves	Paths	Length
1.0	20±3%	67±7%	16.08±2	17±0%	74±2%	22.9±0.2	17+-0%	73±0%	22.9+-0.4
0.5	49±0%	35±0%	14.±3.1	32±0%	55±0%	16.3±4.4	81±0%	10±0%	17.9±3.3

Table 2: Average percentage of turn exchanges spent on learning about the game moves (Moves) versus win conditions (Paths), along with the average dialogue length (in turn exchanges) for 3 games × 2 informativeness levels (IL) (across 100 trials). The first two columns do not add up to 100%, as two turn exchanges of the dialogues are for greetings and goodbyes.

Cond.	Question Distribution		Knowledge Acquired	
	retried questions	new questions	retried questions	new questions
0.2, 1.0	**16%**±3%	84%±5%	**18%**±3%	82%±4%
1.0, 0.2	3%±2%	97%±1%	0%±1%	100%±1%
1.0, 1.0	4%±2%	96%±2%	1%±1%	99%±1%

Table 3: Question distribution and gained knowledge for Quarto in dialogue two, for new and retried questions.

has no way to identify whether it repeats the same question. On the one hand, the reward structure discourages a repeat of the same question in the same dialogue: there would be no expectation of improving the total reward. In a new dialogue, however, a retried question could be advantageous if the simulator in the second dialogue is more informative than was the case in the first dialogue. We were able to identify retries by relying on a feature of the agent that supports context switching. The agent can switch contexts by verbally referencing a previously seen board demonstration, by re-displaying the image of a previous board to the dialogue partner, or both. Because the agent's persistent memory includes a database of the board configurations it has seen, it can not only ask about a game board demonstration seen in a previous dialogue, it can repeat the exact question it used earlier.

Table 3 shows that in a succession of two dialogues, the agent adapts to its partner equally well, regardless of condition and order of the dialogue. Further, in the 0.2, 1.0 condition (a very uninformative partner followed by a very informative partner), the agent retried questions (16%± 3%) of the time, and to good effect: this contributes a corresponding proportion of its newly acquired knowledge (18% ± 3%). In contrast, it rarely retries questions when there could be no expectation of new knowledge.

6.2 Experiment 2: Testing on Unseen Games

Experiment 2 asks: *What can 3GA learn about a game not seen in training, and how does information dynamics affect performance?* To test the agent's communicative skills on unseen games, we picked a simple n-in-a-row game, Tic-Tac-Toe, and a complex one, Gomoku. We show that to a limited degree, 3GA's communicative skills generalize to unseen situations. We also show the large impact of the informativeness level (IL).

IL	Tic-Tac-Toe		Gomoku	
	Baseline	3GA	Baseline	3GA
1.0	38±10%	**50±26%**	8±3%	**12±15%**
0.5	2%±4%	3±11%	0±0%	1±6%
0.2	0±0%	0±0%	0±0%	0±0%

Table 4: Average percentage of learned win conditions in 100 trials, for 2 unseen games × 3 informativeness levels (IL)

Table 4 shows how much of the two unseen games the agent learns from 100%, 50%, and 20% informative partners, compared with the baseline policy used in Experiment 1. Much more of the game is learned when IL=1.0, and much more for the simpler game. The trend here is consistent with Table 1: with decreasing IL, there is a decrease in knowledge acquired through dialogue.

There are also differences across conditions between the information presented to the agent, versus what it learned. With IL=1.0, the agent added most of the win conditions it was told about to its game tree, meaning the game moves have been grounded. With lower IL, however, the agent grounded fewer win conditions that were presented to the agent (for details, see Appendix C). Further, as shown in Table 2), the agent had longer dialogues for more complex games and for higher IL. With lower IL, more questions were about game moves (see Table 9 in Appendix C). In sum, 3GA learned nearly 50% of the simpler game with IL=1.0, and 12% for a very complex game. Improvements on unseen games would likely require extensions to generic game knowledge and richer question types.

6.3 Discussion

In simulation with three games, the single 3GA policy learns far more of the game tree for in a single

dialogue than the baseline, and in all 18 conditions, as much or more than oracle policies trained for a specific game and IL. Because Quarto pieces have so many properties, only about 50% of the complete game tree is learned in five dialogues with IL=1.0, but future work could investigate a way to explicitly reward continued learning. Further, the single 3GA policy exhibits strategy adaption in its choice of questions. For example, a comparison of IL=1.0 and IL=0.5 for Quarto shows three times the number of open-ended questions, and 56% the number of requests for new board demonstrations.

Cond.	Part 1	Part 2	Total
Lo-Md: 3	10.0± 1.2%	8.3± 2.3%	18.3± 0.2%
Lo-Hi: 2	10.0± 4.5%	15± 1.5%	25± 1.4%
Md-Lo: 1	16.0± 0%	5.0± 0%	21.0± 0%
Md-Md: 7	15.0± 2.4%	20± 3.7%	35± 3.5%
Md-Hi: 10	9.6± 2.1%	9.1± 1.2%	18.7± 2.4%
Hi-Md: 6	25.5± 4.3%	11.3± 3.4%	36.8± 6.1%
Hi-Hi: 17	22.0± 1.6%	10± 2.5%	32± 2.3%

Table 5: Col. one shows informativeness (Lo, Md, Hi) of subjects in part 1 and 2 dialogues, and the count (Md-Lo: 1 indicates one example in the condition of a medium informative subject in part one and a low informative one in part two). The Part 1 and Part 2 cols. show the percentage of the game tree acquired in a dialogue, with the sum in the last col.

7 Exploratory Human Study

Our exploratory human study posed three questions: how well does 3GA communicate with people, do people vary in informativeness, and can 3GA acquire knowledge across dialogues. We used Quarto, given that less of its game tree is learned in one dialogue. We tested a variant of 3GA trained specifically for Quarto, omitting the generic game knowledge. Participants communicated easily with the agent, had different informativeness from each other and themselves, and 3GA could continue to learn in a second dialogue.

Twenty-three college students (15 native speakers of English, 8 fluent non-native) participated in a two-part dialogue collection. In part one, eight native speakers had two dialogues each where the agent started out with no game tree (N=16). In part two, all twenty-three subjects each had two dialogues where the agent's knowledge state was a randomly selected end state of one of the part one dialogues (N=46). Participants communicated with the agent via text through a Graphical User Interface (GUI) that displayed the agent's questions, with a text box for subjects to enter responses. The GUI displayed a visual representation of the current game board. Instead of allowing subjects to select demonstrations, they were selected as in simulated dialogues, using the current estimate of the subject's informativeness.

Participants were asked seven questions about whether the language seemed natural or artificial, whether they could answer better if they knew the game better, what aspects of the dialogue they found interesting, and how willing they would be to have another dialogue with this agent (details in Appendix D). All participants said most of the questions were like those a human would ask. Six said they might have provided incorrect answers because they did not know Quarto well. Three mentioned they found it interesting the agent could generalize by asking about re-configurations of the board. Twenty subjects said they would be willing to have more dialogues, and three were neutral. Most subjects volunteered positive comments on the agent's ability to resume discussion of a previous demonstration within a dialogue, and to pick up on its knowledge from previous conversations. Some of their comments appear in Figure 3.

To assess human informativeness, we computed an informativeness score for each question-answer pair in a dialogue. The score for *"wh-"* questions measures the percentage of win conditions in the subject's response, compared to a fully informative answer. For *"yes/no"* questions, a non-response is uninformative (e.g., "I don't know"). The informativeness score is ratio of the positive or negative answers to the total number of *"yes/no"* questions in the dialogue. A subject's informativeness in a dialogue is average informativeness for all questions.

Based on average informativeness as calculated above, part one dialogues were binned into three groups of low ($0\% - 33\%$; n=1), medium ($33\% - 66\%$; n=6) and high ($66\% - 100\%$, n=9) informativeness. Part two dialogues had 1 low, 16 medium, and 29 high. Individuals' informativeness varied. For example, the one subject with an uninformative dialogue in part one had a part two dialogue in the medium group and one in the high group.

The combinations of part one and part two di-

1	It's interesting how the agent seems to pull in old information as if it has a memory.
2	This agent asked questions that seemed human since it claimed to know boards from past conversations.
3	This dialog seemed a lot cleaner and that the agent knew more about the game and how to communicate.

Figure 3: Comments from three subjects.

alogues yielded the seven conditions in Table 5, represented as a hyphenated code (Hi for high, Md for medium, Lo for low) followed by a count for that condition. The variations in informativeness within and across subjects motivates a policy that can adapt its questioning strategy to match the interlocutor's informativeness. Table 5 shows that in part one dialogues with a Hi informative subject, 3GA acquired 22-25±3% of the Quarto game tree. This is within the range of 35±11% on average from one dialogue with the 100% simulator, with unvarying informativeness. Average NLU accuracy on the 62 dialogues was 84±7.8%, and average NLG quality (using a 5-point scale, where 5 is best) was 4.6±0.3.

8 Related Work

Hierarchical POMDP dialogue policies have been applied to switching between domains (Budzianowski et al., 2017), or between different belief states for communicative actions versus slot values (Casanueva et al., 2020). Entities versus relations among them are represented independently in the belief state in (Ultes et al., 2018). In our work, the hierarchical policy is for context switching, and mirrored in the belief state.

Work on learning through communication has often addressed natural language instructions for procedural tasks such as "forwarding an email" (Azaria et al., 2016), "grabbing an object from the desk" (Thomason et al., 2015), or "folding cloth" (Chai et al., 2018). Our work is for learning game-theoretic decision making, rather than procedures.

Extensive work has been done on learning to ground language, including words for objects or actions (Lindes et al., 2017; Goldwasser and Roth, 2014; Matuszek et al., 2012; Wang et al., 2016). Our work grounds not only has 3-way grounding,it grounds generic knowledge to specific games.

9 Limitations

The 3GA policy presented here addresses the question of whether reinforcement learning can be applied to learn a single policy for learning a variety of two-person zero-sum games through dialogues with different individuals who differ in how much information they provide. The specific limitations of this work include the difficulty of generalizing to games not seen in training, the restriction to agent initiative dialogue strategies, and the lack of communicative actions to handle poorly understood responses from the human interlocutor.

We speculate the agent is unable to learn much about the structure of the game tree for unseen games. For the agent to do so this might require an ability to reason about similarities among different games. We found that in unseen games, 3GA is more sensitive to informativeness of the dialogue partner, and to game complexity, in comparison to the games seen in training.

The 3GA policy is agent-initiative, which is a natural starting point for developing dialogue strategies to learn through communication: the agent learns to request information it does not already have. The three existing components that would need the most development to support mixed initiative are the natural language understanding (NLU) capabilities, the reward function, and the calculation of information dynamics.

Regarding the lack of clarification actions, our early probes indicated this would not greatly improve the user experience, and the questionnaire results confirmed that subjects found the agent easy to communicate with. To add this capability, communicative actions to request a restatement, or to state a lack of understanding, should be added to the agent's repertoire. As the NLU confidence is already part of the belief state, it is likely the policy could learn to execute clarification requests.

A final limitation is that we have not yet looked at extending this work to learning other kinds of real-world activities. Extending this work for new domains would require the means to store and reason over knowledge, extensions to the MR, and training data for the natural language models.

10 Conclusion

The 3GA policy applies to **different games**, supports **sustained knowledge acquisition across multiple dialogues**, and **accommodates differences in informativeness** of interlocutors. We have shown that people do vary in their informativeness. Information dynamics, which is inspired by interpersonal synergy (Fusaroli and Tylen, 2016), supports seamless transfer of a policy trained in simulation for immediate use with people, similar to use of sensing in robotics to close the so-called reality gap (Abraham et al., 2020; Peng et al., 2018; Bousmalis et al., 2018). It also leads to more semantic diversity of questions for the most complex game. Finally, the agent can learn in a limited way about a game not seen during training.

References

Ian Abraham, Ankur Handa, Nathan Ratliff, Kendall Lowrey, Todd D Murphey, and Dieter Fox. 2020. Model-based generalization under parameter uncertainty using path integral control. *IEEE Robotics and Automation Letters*, 5(2):2864–2871.

Ali Ayub and Alan R Wagner. 2018. Learning to win games in a few examples: Using game-theory and demonstrations to learn the win conditions of a Connect Four game. In *Social Robotics*, pages 349–358, Qingdao, China. Springer International Publishing.

Amos Azaria, Jayant Krishnamurthy, and Tom Mitchell. 2016. Instructable intelligent personal agent. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.1.

Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *(International Conference on Learning Representations ICLR)*, San Diego, CA, USA. ICLR.

Konstantinos Bousmalis, Alex Irpan, Paul Wohlhart, Yunfei Bai, Matthew Kelcey, Mrinal Kalakrishnan, Laura Downs, Julian Ibarz, Peter Pastor, Kurt Konolige, et al. 2018. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4243–4250, Brisbane Convention and Exhibition Centre, Brisbane, Australia. IEEE, IEEE.

Paweł Budzianowski, Stefan Ultes, Pei-Hao Su, Nikola Mrkšić, Tsung-Hsien Wen, Iñigo Casanueva, Lina M. Rojas-Barahona, and Milica Gašić. 2017. Sub-domain modelling for dialogue management with hierarchical reinforcement learning. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 86–92, Saarbrücken, Germany. Association for Computational Linguistics.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

Joyce Y. Chai, Qiaozi Gao, Lanbo She, Shaohua Yang, Sari Saba-Sadiya, and Guangyue Xu. 2018. Language to action: towards interactive task learning with physical agents. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 2–9, Stockholm, Sweden. International Joint Conferences on Artificial Intelligence Organization.

Riccardo Fusaroli and Kristian Tylen. 2016. Investigating Conversational Dynamics: Interactive Alignment, Interpersonal Synergy, and Collective Task Performance. *Cognitive Science*, 40:145–171.

Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tur. 2019. Dialog state tracking: A neural reading comprehension approach. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 264–273, Stockholm, Sweden. Association for Computational Linguistics.

Milica Gašić and Steve Young. 2013. Gaussian processes for POMDP-based dialogue manager optimization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(1):28–40.

Dan Goldwasser and Dan Roth. 2014. Learning from natural instructions. *Machine learning*, 94(2):205–232.

H. P. Grice. 1975. Logic and conversation. In *Syntax and semantics 3: Speech acts*, pages 41–58. Academic Press.

Padraig Higgins, Gaoussou Youssouf Kebe, Kasra Darvish, Don Engel, Francis Ferraro, Cynthia Matuszek, et al. 2021. Towards making virtual human-robot interaction a reality. In *Human-Robot Interaction (HRI) 3rd International Workshop on Virtual, Augmented, and Mixed-Reality for Human-Robot Interactions (VAM-HRI)*.

Lukasz Kaiser. 2012. Learning games from videos guided by descriptive complexity. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 963–69, Toronto, Ontario, Canada. MIT Press.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

James R. Kirk and John E. Laird. 2019. Learning hierarchical symbolic representations to support interactive task learning and knowledge transfer. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 6095–6102, Macao, China. International Joint Conferences on Artificial Intelligence Organization.

Peter Lindes, Aaron Mininger, James R Kirk, and John E Laird. 2017. Grounding language for interactive task learning. In *Proceedings of the First Workshop on Language Grounding for Robotics*, pages 1–9.

Cynthia Matuszek, Nicholas FitzGerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. 2012. A joint model of language and perception for grounded attribute learning. In *Proceedings of the 29th International Conference on Machine Learning*, ICML'12, page 1435–1442, Madison, WI, USA. Omnipress.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. 2018. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1–8, Brisbane, Australia. IEEE.

Nisha Pillai, Cynthia Matuszek, and Francis Ferraro. 2021. Neural variational learning for grounded language acquisition. In *Proc. of the IEEE International Conference on Robot & Human Interactive Communication (Ro-Man)*, British Columbia, Canada.

Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-based user simulation for bootstrapping a POMDP dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152, Rochester, New York. Association for Computational Linguistics.

Gabriel Skantze. 2007. Making grounding decisions: Data-driven estimation of dialogue costs and confidence thresholds. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, pages 206–210, Antwerp, Belgium. Association for Computational Linguistics.

Jesse Thomason, Shiqi Zhang, Raymond J Mooney, and Peter Stone. 2015. Learning to interpret natural language commands through human-robot dialog. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Stefan Ultes, Paweł Budzianowski, Iñigo Casanueva, Lina M. Rojas-Barahona, Bo-Hsiang Tseng, Yen-Chen Wu, Steve Young, and Milica Gašić. 2018. Addressing objects and their relations: The conversational entity dialogue model. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 273–283, Melbourne, Australia. Association for Computational Linguistics.

Maaike van den Haak, Menno De Jong, and Peter Jan Schellens. 2003. Retrospective vs. concurrent think-aloud protocols: Testing the usability of an online library catalogue. *Behaviour & Information Technology*, 22(5):339–351.

Sida I. Wang, Percy Liang, and Christopher D. Manning. 2016. Learning language games through interaction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2368–2378, Berlin, Germany. Association for Computational Linguistics.

Zhuoran Wang and Oliver Lemon. 2013. A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In *Proceedings of the SIGDIAL 2013 Conference*, pages 423–432, Metz, France. Association for Computational Linguistics.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449, Valencia, Spain. Association for Computational Linguistics.

S. Young, M. Gašić, B. Thomson, and J. D. Williams. 2013. POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Maryam Zare, Ali Ayub, Aishan Liu, Sweekar Sudhakara, Alan Wagner, and Rebecca J. Passonneau. 2020. Dialogue policies for learning board games through multimodal communication. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 339–351, 1st virtual meeting. Association for Computational Linguistics.

A Descriptions of Board Games

The 3GA agent was tested on five two-player, zero-sum board games, regarding its ability to engage in dialogues for learning through communication. All were n-in-a-row games, meaning where a win consists of n pieces in a straight line. The five games were Connect Four, Gobblet, Gomoku, Quarto, and Tic-Tac-Toe. Each is described below.

Connect Four has a vertical game board, or rack, of seven columns with slots for six game discs per column (size $6 \times 7 = 42$). At each game state, a player has at most seven possible actions. Once an action is taken, it cannot be taken again. A player wins when four of her discs are adjacent in a column, row, or on a diagonal. Any slot filled by a disc is no longer available for play. When both players have filled the board with no winning sequence, the game ends as a draw in a terminal state (game tree leaf). Finally, if a player makes 4-in-a row, the game ends and the game tree path ends in a terminal win.

Gobblet is played on a 4×4 horizontal board, where each player has twelve game pieces of one color in four sizes, three of each size. Pieces can be placed over (gobble) smaller pieces. At the root node, there are $4\ rows \times 4\ cols \times 4\ sizes = 64$ possible moves. After the first player makes a move, the second player can place a piece of a given size on any of the 15 remaining positions on the board ($4 \times 15 = 60$), or gobble the first player's piece, for a maximum of 63 possible moves, and once a piece has been played, it cannot be moved. (These constraints modify the original game rules, in which gobbling can only be carried out by a piece already in play.) Play ends in a draw when no moves are available, and ends in a win for any player who has four visible pieces in a row.

Gomoku is played with 181 black and 180 white game pieces (stones) on a Go board, with each player choosing a color. It can be played using a board with 15×15 lines (instead of cells) or a 19×19 board (lines instead of cells). Players take turns placing a piece of their color on an empty corner, with black playing first. We use a modified version where corners on the outer lines can be played. The winner is the first player to form an unbroken chain of five pieces horizontally, vertically, or diagonally. When the game begins, black has 225 available actions. The second player can choose one out of the 224 remaining board positions. Similar to Connect Four, at each state of the game, the number of moves is reduced by one, and the game tree terminates when the board is filled (draw), or a win or loss is reached.

Quarto has a 4×4 board and 16 game pieces, evenly divided into two colors, two heights, two shapes, and whether they are solid or hollow. In each turn, the piece that the current player will place on the board is chosen by the opponent from the unplayed pieces. Four in a row wins if the four pieces share at least one property. At the start of the game, when the first player is given a piece by the opponent, there are 16 board locations for that piece. After each move, the current player selects a piece from those that have not yet been played and gives it to the next player to place on the board in an open position. The game ends in a draw if all pieces have been played with no winning sequence, else in a win for the player who gets four in a row that all have the same color, height, shape, or hollowness.

Tic-Tac-Toe is played on a 3×3 grid where two players take turns placing their mark (X or O) in an open square. The player who succeeds in placing three marks in a diagonal, horizontal, or vertical line is the winner. Compared to the other four games tested here, Tic-Tac-Toe has a smaller game tree. At the start of the game, the first player has 9 possible moves. After the opening move, the second player has 8 possible moves: the remaining unmarked squares. After each move, the number of moves in the next game state reduces by one, until a draw, or a win for one player.

B Meaning Representation (MR)

The meaning representation (MR) for generating questions about games, or for understanding responses, is a variant of first order logic, where question type operators (e.g., `Inform`) are functions that take an attentional state (beliefs associated with a specific game board demonstration or answers to questions about it) as an argument, and whose values are specific questions formulated in reference to that context. The MRs for the agent's questions are hierarchical. The parent operators choose whether to continue the current context, resume a previously suspended one, or request a new game board demonstration (e.g. a new win condition (`Req_Win_Dem()`), or a new game move demonstration (`Req_Move_Dem()`)). The child operators produce context specific questions of different types. Table 6 lists the communicative action

Action Type	Meaning
Parent Policy	
Req_Move_Dem()*	Request the demonstration of a new game move.
Req_Win_Dem()	Request the demonstration of a new win condition.
Continue(D_i)	Continue discussion of current demonstration D_i.
Resume(D_i)	Resume discussion of a previous demonstration D_i.
Start(), Finish()	Greetings and Goodbyes.
Child Policy	
Conf(ShiftBoard)	Is demonstration D still a win after ShiftBoard? (ShiftBoard can be Rotate or Translate.)
Conf(Property())	Does Property() cause demonstration D to be a win? (Property can be shape, color, height, hollowness, or size.)
Conf(ChangeDisk())	Is demonstration D still a win after ChangeDisks? (ChangeDisk can be RemoveDisk or AddDisk.)
Req(ShiftBoard())	What ShiftBoard operations on demonstration D will also be a way to win?
Req_Other()	Can the other player undo D?
Req(Location())*	What locations other than demonstration D are legal moves?
Conf(Location())*	Is Location() also valid for the game piece shown in demonstration D?
Req(Piece())*	What other game pieces can be played instead of the one shown in demonstration D?
Conf(Piece())*	Can Piece() be used instead of the game piece used in demonstration D?

Table 6: Communicative action operators for the agent. Asterisks mark those used to learn about game moves.

Action Type	Meaning
Inform()	Some or all of the requested information has been provided.
Affirm()	Positive answer to a yes/no question.
Negate()	Negative answer to a yes/no question.
Unknown()	Non-answer to a question.
Start(), Finish()	Greetings and closings

Table 7: Partner's Communicative Actions

types for asking questions about games, and Table 7 shows those for interpreting answers.

When a dialogue about a game begins, the agent can ask about individual moves or game pieces and about ways to win, where a win condition is a sequence of n pieces in a straight line, possibly with constraints on the shape, height, color, etc. Using communicative actions that ask about game moves or pieces, the agent learns the specific set of actions available in a particular game, and therefore, how to add actions and states to a specific game tree. Using communicative actions that ask about win conditions, the agent learns paths from the root to a leaf in the game tree.

Communicative Action Predicates	
Predicate	**Range**
Game Piece	
Shape*	Round, Square
Height*	Tall, Short
Color*	Red, Blue
Hollowness*	Hollow, Solid
Size$^+$	Small, Medium1, Medium2, Large
Game Board	
Columns	$\{c \mid c \in \#BoardColumns\}$
Rows	$\{r \mid r \in \#BoardRows\}$
Angles	$\{0, 45, 90, 135\}$
Board Position	$\{(x, y) \mid x \in \#Rows, y \in \#Columns\}$ (e.g. 42 places for Connect Four, 16 places for Quarto, etc.)

Table 8: Predicates that serve as arguments to the communicative action types shown in Table 6. Asterisks mark those used only in Quarto, and plus signs mark those only used in Gobblet.

IL	Tic-Tac-Toe		
	Moves	Paths	Length
100%	10±5%	76±8%	12.5±1.1
50%	75±7%	2±3%	12.35±0.9
20%	83±7%	1±7%	12.31±0.7
	Gomoku		
100%	21±8%	69±9%	21.7±2.7
50%	47±18%	44±18%	16.3±4.4
20%	87±15%	3±3%	13.9±3.3

Table 9: Average percentage of turn exchanges spent asking about game moves (Moves) versus win conditions (Paths), along with average dialogue length, for two unseen games × three informativeness levels (across 100 dialogues per condition). The proportions sum to less than 100%; the remainder pertain to turn exchanges on greetings and closings.

Table 8 shows the predicates for the context-specific components of the agent's communicative actions. Each predicate has a corresponding vector in the belief state, as described in the paper.

C Analysis of Questions in Dialogues about Unseen Games

Table 9 reports the proportion of time the agent spends learning about game moves versus win conditions for the two unseen games Tic-Tac-Toe and Gomoku, across 100 dialogues per condition. The results are comparable to those in Table 2. The agent has longer dialogues for the more complex game, and with more informative responses. As informativeness decreases, the agent asks relatively more questions about game moves.

D Information about Human Study Setup

We recruited twenty three college students from the same university (20 males and 3 females). Subjects were told that we were interested in the overall conversational ability of the agent, the experience of a person whose role in the conversation is to answer the agent's questions, and what sorts of next-step improvements we might be able to make. They completed the questionnaire in Figure 4, after a session with two dialogues. We used a retrospective think-aloud format in which participants entered their reflections after each dialogue, to minimize the cognitive overload of a concurrent think-aloud (van den Haak et al., 2003). Additionally they were asked to report their thoughts about any questions they found surprising/odd/unnatural/smart, or to share any other comments.

Participants engaged in text-based dialogues with the agent using the graphical user interface shown in Figure 5. A button in the interface allowed participants to mark points in the dialogue that they could later review and optionally comment on. After each dialogue, they immediately reviewed the transcript to enter positive, negative or neutral comments about their experience. After they finished their two dialogues, we asked them to fill out a questionnaire to better understand their experience with the agent. Subjects answered three questions about their knowledge of Quarto, their experience interacting with artificial agents, and their experience with agents that maintain a dialogue context.

1. How often were the questions like those a human might have asked?
2. How often were the questions like those an artificial type of "decision maker"?
3. How often did you feel you could understand the agent's question better if you knew the game better yourself?
4. Did you feel like you helped the agent learn the game?
5. What aspects of the dialogues did you find interesting, if any?
6. How willing would you be to have another dialogue about a game with this agent?
7. You are among a very small group of people who has had a "dialogue" with an agent that learned how to do something new from you through communication. The final optional part of this study is for you to tell us if you have any additional comments.

Figure 4: The complete list of questions asked from subjects at the end of part 1 of the study. Part 2 of the study used questions 4 and 7.

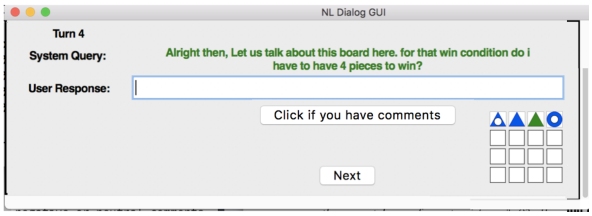

Figure 5: The GUI shows the agent's question, a box for subject's response, a 2D representation of the board, and a button to bookmark a turn, for later review.)