Adapting Language Complexity for AI-Based Assistance

Michelle Zhao
mzhao2@andrew.cmu.edu
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA

Reid Simmons
rsimmons@andrew.cmu.edu
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA

Henny Admoni
henny@cmu.edu
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA

ABSTRACT

AI systems designed for personalized coaching and instruction can leverage repeated interactions to improve upon the assistance they provide to the learners. There exist different levels of complexity at which instructions can be presented. For example, "go to my office" and "walk forward 5 steps" are two navigation commands with different complexities and knowledge prerequisites. The appropriate complexity level of instruction varies as the student knowledge and cognitive state, task, and environment change. By maintaining a model of student understanding, an AI assistant can adapt its teaching strategy to the capabilities of human partners. An accurate assessment of the student's ability is thus critical for dependable and effective human-AI coaching. We present a closed-loop interaction framework that adapts the level of information complexity based on the human partner's observable cognitive understanding. This work-in-progress investigates how knowledge and preparation impact the suitability of different complexity levels, motivating dynamic interaction.

CCS CONCEPTS

• **Human-centered computing** → **Interaction design process and methods**; **Activity centered design**.

KEYWORDS

adaptive instruction, intent prediction, navigation, coaching

1 INTRODUCTION

In order to effectively assist humans, an AI agent should be able to (1) evaluate human behavior, and (2) plan actions according to its understanding of the human partner [3, 14]. Specifically, we aim to design effective human-AI instruction. Human teachers and coaches must infer and assess their students' limitations and mastery in order to adapt content and style in response. Similarly, an AI instructor must not only track a student's actions, but also be able to infer their intentions, cognitive state, and level of understanding. By iteratively improving upon its model of the human, an AI instructor can better adapt its interactions to the needs of the human.

The objective of matching human skill to method of assistance is a concept in game design, which aims to maintain equilibrium between stress, arousal, and performance [11, 17]. External assistance techniques at an appropriate difficulty level can influence a player's performance and promote cognitive flow [17]. In coaching, balancing instruction method and player preparedness manages stress and engagement. For example, a coach who commands a beginner player with esoteric terms may confuse, not assist, the player. Similarly, a coach who uses far too basic instructions towards an experienced player may frustrate and disengage the player [1].

Thus, the primary research question is *How can assistance be adapted to the knowledge and cognitive states of humans?* Without explicitly querying a human, can AI systems infer what level of interaction complexity is appropriate? Our experiments next investigate how humans with varying knowledge interact with this adaptive system.

In this work, we use an AI-guided search-and-rescue task. Search-and-rescue tasks are especially difficult when the rescuer has a limited or obscured field of view and does not know where in the building the injured victims are located. Thus, the primary task is navigation through an uncertain environment. The AI guide, having knowledge of the building layout and victim locations, must help the human rescuer navigate to all victims throughout the building as quickly as possible. Navigation instructions can be provided at varying levels of complexity. Higher complexity instructions require more prerequisite information. At the highest level of complexity, the AI coach may instruct the rescuer, "Go to Destination C." This requires the rescuer to have stored knowledge of each destination location. At the lowest level, the command, "take 1 step forward" does not require any prerequisite information.

In this work, we introduce a closed-loop, interaction framework for an AI instructor to guide human students. The interaction is comprised of (1) an EVALUATE step, where the AI infers the intent and understanding of the human student, and (2) an ADAPT step, in which the AI adapts the complexity level of instructions presented to the student to better suit their cognitive understanding of the task and directions at hand.

The adaptation process is the outcome of repeated interactions between student and instructor. Gaines [6] describes adaptive instruction as consisting of three elements: (1) evaluation of learning outcome, (2) adaptive logic for dynamically changing the learning process, and (3) an adaptive variable that changes the training task or environment.

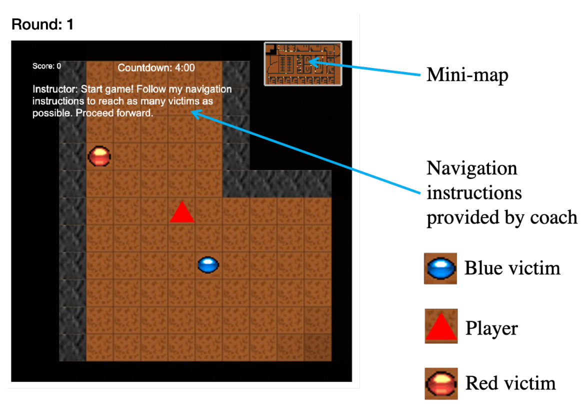

Figure 1: 2D gridworld view of search and rescue task provided to participants.

Evaluating the human learner's performance and surveying their experience allows us to measure the efficacy of various instruction complexity levels for a particular learner. The adaptive logic for dynamically changing the learning process occurs based on the learner's perceived understanding. We observe the learner's behavior to measure their understanding. The adaptive variable that changes throughout the interaction is the complexity level of navigation instruction provided to the rescuer (Section 2.2).

2 PRELIMINARIES

2.1 Task Scenario

We construct a damaged office building in a 2D gridworld environment as the task scenario for a single human rescuer (see Figure 2). There are 20 injured victims inside of the building who need to be found and rescued. Of these, 7 victims are severely injured (denoted in red), and will expire if not treated in time. The rest are moderately injured (denoted in blue), and will persist the duration of the 4-minute game. The AI guide will provide on-screen navigation instructions throughout the game to help the player navigate to all victims in time (see Figure 1). The task of the human player is to follow the coach's instructions to save all victims.

2.2 Instruction Complexity Levels

2.2.1 Human Level Assignment. The AI instructor maintains a single-parameter model of the rescuer. The rescuer is assigned a specific instruction complexity level β, which determines the complexity of text-based navigation instructions at which the coach will interact with the rescuer [5, 19, 20]. Instruction complexity is defined by the amount of prerequisite knowledge required to understand the instruction. There are 3 total levels: Level 1 ($\mathcal{L}_1$): LOW, Level 2 ($\mathcal{L}_2$): MID, Level 3 ($\mathcal{L}_3$): HIGH.

2.2.2 Definition. Each level is defined by a vocabulary corpus V_i, where i indicates the level. Each corpus is comprised of a set of actions and objects, $V_i = (A_i, O_i)$. An instruction is an (action, object) tuple. `ROOMS` is the set of room names in the building map.

A_3 = {Go to, Triage}
O_3 = `ROOMS` ∪ {Red Victim, Blue Victim}

A_2 = {Turn, Proceed, Enter, Exit, Approach and Save}
$O_2 = \mathbb{N} \times$ {Room, Hallway, Intersection, left, right, red victim blue victim}

A_1 = {Walk forward, Turn, Stop, Utilize}
$O_1 = \mathbb{N} \times$ {Steps, Left, Right, Medical Equipment]}

At $\mathcal{L}_3$, the AI guide instructs the rescuer at the highest level of prerequisite knowledge. Examples of Level 3 instruction tuples are (Go to, Room 109) or (Triage, Blue Victim). The instructor can only tell the rescuer to go to a particular room, whose name and location the rescuer must know. While this complexity level requires greater prior knowledge, it trades off cognitive load, by being brief and more straightforward for knowledgeable rescuers.

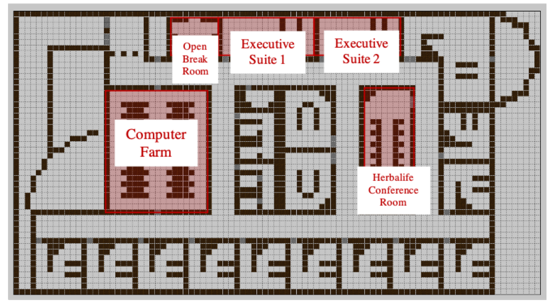

Figure 2: Map of 2D building blueprint, containing a subset of room names.

$\mathcal{L}_2$ instructions require minimal knowledge. Level 2 commands require the rescuer to have baseline knowledge of how doors and intersections appear in the environment. Examples include (Proceed to, 2nd door on right), and (Turn, Right).

Examples of Level 1 instructions are (Walk forward, 5 Steps), (Turn, Right), and (Utilize, Medical Equipment). This command type requires no knowledge of the building blueprint, because the instruction corpus is restricted to motion primitives. Each instruction level can be generated from a transformation of a neighboring level, that either applies higher or lower complexity.

3 INTERACTION FRAMEWORK

In this section, we present a two-step, EVALUATE-and-ADAPT interaction framework for adaptive coaching based on complexity level switching (see Figure 3), similar to [14]. The human rescuer's role is an instruction-following task, which ensures compliance to the provided instructions. At time $t = 0$, the AI guide gives the rescuer (human) an instruction at a default complexity level, β_0. The instruction directs the rescuer to the first victim. In the EVALUATE step, the instructor observes the actions of the rescuer and predicts whether they are headed to the correct victim (goal). Assuming the human complies with all instructions, goal prediction serves as a proxy for comprehension at the current complexity level β_0, because failing to reach a goal must thus be attributed to inability. The instructor computes the probability that the rescuer is headed to the intended victim g_{intended} given their current trajectory $\mathbb{P}(g_{\text{intended}}|\text{Trajectory})$.

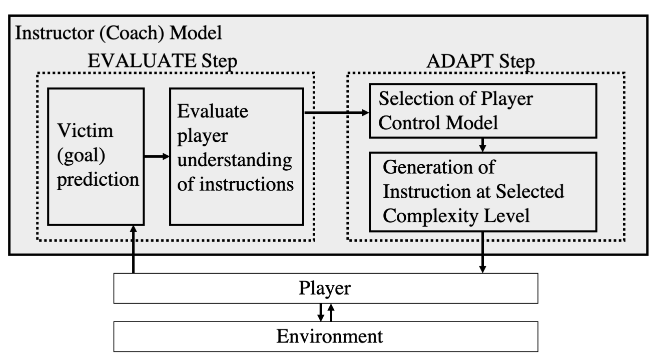

Figure 3: Instruction loop. First, the AI performs an EVALUATE step, followed by an ADAPT step, generating the next instruction at the appropriate level for the human rescuer.

A high $\mathbb{P}(g_{\text{intended}}|\text{Trajectory})$ indicates the human understands the instructions provided at the current level, and vice versa.

In the ADAPT step, the AI adjusts its model of the rescuer. The instructor makes a threshold-based decision on whether or not the level of instruction must change in order to better fit the human rescuer's comprehension and knowledge. If $\mathbb{P}(g_{\text{intended}}|\text{Trajectory})$ falls below some threshold, indicating the rescuer is unable to understand instructions at the current complexity, the AI instructor decreases the rescuer's assigned level for the next time step, β_{t+1}, and thus provides a more basic instruction in the next interaction. A high $\mathbb{P}(g_{\text{intended}}|\text{Trajectory})$ value indicates the rescuer understands the instruction complexity level well and is headed to the correct victim. The instructor may increase the rescuer's β_{t+1} parameter, provide a higher-level next instruction, and EVALUATE again. This process occurs for as many interactions as necessary to guide the rescuer throughout the episode.

3.1 EVALUATE Step

In the EVALUATE step at time t, the instructor's objective is to determine whether or not the human rescuer understands the instructions provided at the current assigned complexity level β_t. Identifying situations where the rescuer lacks the ability or prerequisite knowledge needed to follow particular directions informs the AI to adjust the complexity of instruction accordingly. Victim goal prediction [2, 8, 16] serves as a proxy and measure for human understanding.

3.1.1 Bayesian Goal Prediction. The set of goals is the set of victims $G = \{g_1, ..., g_{20}\}$. We aim to compute $\mathbb{P}(g^*|\xi_{0:t})$ the probability that the rescuer is headed to the instructed victim g^* given their current trajectory $\xi_{0:t} = \{x_1, ..., x_t\}$, where x_t is the position at time t. We utilize the Bayesian formulation:

$$\mathbb{P}(g^*|\xi_{0:t}) = \mathbb{P}(g^*|x_0, ..., x_t) = \frac{\mathbb{P}(x_0, ..., x_t|g^*)\mathbb{P}(g^*)}{\mathbb{P}(x_0, ..., x_t)}. \quad (1)$$

We define a term: the initial compliance prior $\mathbb{P}(g^*)$, parameterized by α, to be a static prior that represents the baseline compliance of the human rescuer. Given the rescuer has not taken any steps after being given an instruction, $\mathbb{P}(g^*)$ is the probability that the rescuer is headed towards the intended goal g^*. $\mathbb{P}(g^*)$ is a static prior because the instructor model must maintain over long trajectories the notion that the rescuer is still most likely to go to the victim (goal) that he or she was instructed to go to. $\mathbb{P}(g^*;\alpha) = \alpha$. Let G_R be the set of victims remaining to be saved. For the other remaining non-intended victims $\{g_i \in G_R : g_i \neq g^*\}$, $\mathbb{P}(g_i;\alpha) = \frac{1-\alpha}{N-1}$.

To get the likelihood of the trajectory $\xi_{0:t}$ given goal g^*, (data given model), we assume Markov independence of position at each state. To compute $\mathbb{P}(x_0, ..., x_t)$, marginalize over the victims remaining.

$$\mathbb{P}(g^*|\xi_{0:t}) = \frac{\mathbb{P}(x_t|g^*, x_{t-1})\mathbb{P}(g^*;\alpha)\mathbb{P}(g^*|\xi_{0:t-1})}{\sum_{g_j \in G_R} \mathbb{P}(x_t|g_j, x_{t-1})\mathbb{P}(g_j;\alpha)\mathbb{P}(g_j|\xi_{0:t-1})} \quad (2)$$

Equation 2 thus serves as the inference update. $\mathbb{P}(x_0|g^*) = 1$, because the starting position is guaranteed to be visited. In order to compute the probability of taking each step $\mathbb{P}(x_i|x_{i-1}, g_j)$, we run for each remaining victim, a flood-fill over the entire map originating from the victim locations. The flood-fill value for victim g_j informs us of the proximity in number-of-steps-remaining to reach victim g_j from all map coordinate locations (x, y). When the rescuer takes a single step, decreasing g^* flood-fill values indicate approaching the intended victim, and increasing values indicate moving away from the victim.

It is only fair to allow the rescuer to make mistakes when traversing an uncertain environment. The rescuer error allowance is defined by ϵ. ϵ is a step-wise function of proximity to goal, where $D(x_t, g^*)$ is the taxicab distance from current position to instructed victim. The closer the rescuer gets to the victim, the more likely it becomes that the rescuer is headed directly towards the precise location of the victim. Hyperparameter ϵ values can be adjusted based on the task. For the search-and-rescue, we tune ϵ and set the below thresholds based on sample trials.

$$\epsilon = \begin{cases} 0.1, & \text{if } D(x_t, g^*) \leq 10 \\ 0.4 & \text{otherwise} \end{cases} \quad (3)$$

The probability that the rescuer takes one step away from the instructed victim $\mathbb{P}(x_t|x_{t-1}, g^*) = \epsilon$, since this would occur if the rescuer makes a single-step mistake with respect to the g^*. $\mathbb{P}(x_t|x_{t-1}, g^*) = 1 - \epsilon$ if the rescuer moves closer to the victim. $\mathbb{P}(x_t|x_{t-1}, g^*)$ is normalized over the probabilities of each possible step from position x_{t-1}. Let $Reachable(x_{t-1})$ be the set of 1-step reachable locations from x_{t-1}, and $C_{g_j}(x_{t-1})$ is the distance between position x_{t-1} and victim g_j. The one-step likelihood given goal g_j becomes

$$\mathbb{P}(x_t|x_{t-1}, g_j) = \begin{cases} \frac{1-\epsilon}{Z}, & \text{if } C(x_t) \leq C(x_{t-1}) \\ \frac{\epsilon}{Z}, & \text{if } C(x_t) > C(x_{t-1}) \end{cases} \quad (4)$$

$$Z = \sum_{x_i \in Reachable(x_{t-1})} \mathbb{P}(x_i|x_{t-1}, g_j)$$

3.2 ADAPT Step

In order to generate navigation instructions to all victims under the 4-minute time constraint, the AI guide first plans a path using A^* from the rescuer's starting location to the locations of all victims, constrained by the expiration times of each victim type.

In the ADAPT step, the threshold-based decision adapts instruction to the rescuer's level of understanding. Define two hyperparameters: a threshold τ_G for leveling-up, and a threshold τ_B for leveling-down. If $\mathbb{P}(g^*|\xi_{0:t}) > \tau_G$, the instructor increments the level of instruction. If $\mathbb{P}(g^*|\xi_{0:t}) < \tau_B$, then decrement the instruction level. Otherwise, the rescuer level remains as is (see Figure 4). Algorithm 1 defines the interaction loop algorithm for adaptive navigation assistance (ANA).

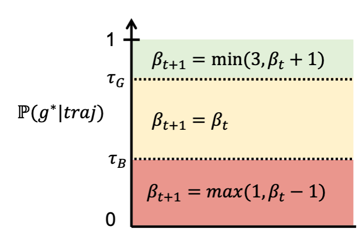

Figure 4: Threshold-based level switching mechanism

Algorithm 1: Adaptive Navigation Assistance (ANA)

Result: $\{\mathcal{L}_{t+1}\}$

$\xi_{0:t}, g_t^*, C_{g_t^*}, \mathcal{L}_t \leftarrow$ **Input:** trajectory, current goal, floodfill values of current goal, current level

$\tau_B, \tau_G \leftarrow$ **Input:** Level-Decrement threshold, Level-Increment threshold

$Levels = \{\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3\}$

Compute $\mathbb{P}(g_t^*|\xi_{0:t})$

if $\mathbb{P}(g_t^*|\xi_{0:t}) < \tau_B$ **then**

  $\mathcal{L}_{t+1} \leftarrow max(1, \mathcal{L}_t - 1)$

end

if $\mathbb{P}(g_t^*|\xi_{0:t}) > \tau_G$ **then**

  $\mathcal{L}_{t+1} \leftarrow min(3, \mathcal{L}_t + 1)$

else

  $\mathcal{L}_{t+1} \leftarrow \mathcal{L}_t$

end

4 EXPERIMENTAL DESIGN

We design a user study to answer the following questions

Q1: How does a rescuer's preparedness (amount of pretraining and prerequisite knowledge) for a task affect their preference and performance when instructed at different levels of complexity?

Q2: What effect does dynamic versus static assistance have on rescuer performance and preference?

4.1 Study Design

To investigate these questions, we define a mixed-design study, where **Q1** leverages a between-subjects design, and **Q2** uses a within-subjects comparison. The between-subjects independent variable, denoted IV_1, is *preparedness level*. The within-subjects independent variable, IV_2, is *adaptivity of instructor*.

4.1.1 Procedure. Participants will play a 2D gridworld search-and-rescue game for two rounds. One of the trials will be guided by a static instructor, and the other will be with an adaptive instructor. The static instructor can only provide instruction at a fixed complexity level, which will be selected randomly from the 3 levels $\{\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3\}$. The adaptive instructor will use ANA to dynamically adjust the instruction complexity to the knowledge and understanding of the human rescuer. Each trial is divided into three phases: Train, Pre-test, and Test (see Figure 5).

Training Phase In the pretraining phase, the human participant is provided a limited time frame to memorize the map and room names to the best of his/her ability. The rescuer is assigned to a pretraining group drawn from the set [1 minute, 5 minutes], in order to ensure a wide, near-uniform spread of participant preparedness. The pretraining amount will be held constant over both trials.

Pre-Test Phase In the pre-test phase, the participant is quizzed on the room names. Their score on this room-labeling pre-test defines their rescuer-preparedness metric.

Test Phase In the test phase, the participant plays the game to rescue as many victims as possible while following the directions of an AI instructor.

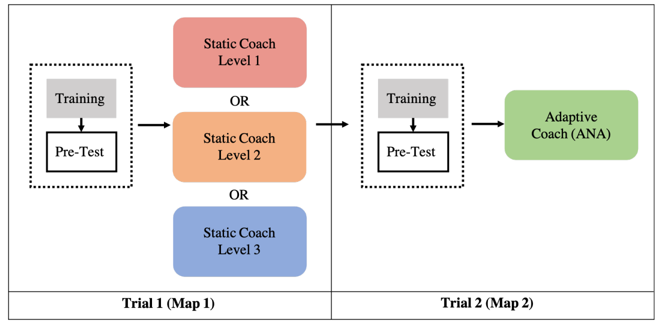

Figure 5: User study design.

The instructor order is randomized to counterbalance training effects. After participants have worked with both instructors, they will be asked a series of questions comparing the two, using Likert and interval scale ratings.

4.2 Hypotheses

We anticipate positive correlation between preparedness and preferred complexity level. We also predict an partiality for the adaptive over static coach.

H1: Higher participant preparedness will induce preference for higher levels of instruction complexity.

H2: Participants with high preparedness scores will perform better at higher levels of complexity.

H3: Participants will prefer the adaptive instruction over static instruction, especially at low preparedness levels.

5 CONCLUSION

In this work, we demonstrate an interactive EVALUATE-and-ADAPT strategy for adaptive coaching for human learners. The adaptive strategy generalizes to other tasks in addition to navigation. For example, performing emergency medical procedures is complex, and very difficult for nonprofessional bystanders. An AI instructor for such situations must perform complexity level adjustment on-the-fly for first-responders of various skill levels.

One important future direction of this work is developing modular adaptive advising systems. The interaction framework we present manipulates information complexity. In future work, we plan to investigate abstraction levels across further aspects of learning in team tasks. This system can be employed in more complex teaching systems, and extended to manipulate other variables of machine social intelligence.

ACKNOWLEDGMENTS

This research is based upon work supported by the Defense Advanced Research Projects Agency (DARPA), award number: FP00002636. Any opinions, findings and conclusions or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA).

REFERENCES

[1] Andy Abraham and Dave Collins. 2011. *Effective Skill Development: How Should Athletes' Skills Be Developed?* 207–230. https://doi.org/10.1016/B978-0-443-06734-1.00015-8

[2] G. Best and R. Fitch. 2015. Bayesian intention inference for trajectory prediction with an unknown goal destination. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 5817–5823. https://doi.org/10.1109/IROS.2015.7354203

[3] Andrew Bimba, Norisma Idris, Ahmed Al-Hunaiyyan, Rohana Mahmud, and Liyana Shuib. 2017. Adaptive feedback in computer-based learning environments: a review. *Adaptive Behavior* 25 (08 2017), 105971231772759. https://doi.org/10.1177/1059712317727590

[4] Micah Carroll, Rohin Shah, Mark K. Ho, Thomas L. Griffiths, Sanjit A. Seshia, Pieter Abbeel, and Anca D. Dragan. 2019. On the Utility of Learning about Humans for Human-AI Coordination. *CoRR* abs/1910.05789 (2019). arXiv:1910.05789 http://arxiv.org/abs/1910.05789

[5] Matt Duckham, Stephan Winter, and Michelle Robinson. 2010. Including landmarks in routing instructions. *J. Location Based Services* 4 (03 2010), 28–52. https://doi.org/10.1080/17489721003785602

[6] Brian Gaines. 1972. The learning of perceptual-motor skills by men and machines and its relationship to training. *Instructional Science* 1 (10 1972), 263–312. https://doi.org/10.1007/BF00055704

[7] Peter Hase and Mohit Bansal. 2020. Evaluating Explainable AI: Which Algorithmic Explanations Help Users Predict Model Behavior? arXiv:2005.01831 [cs.CL]

[8] Hong Jun Jeon, Smitha Milli, and Anca D. Dragan. 2020. Reward-rational (implicit) choice: A unifying formalism for reward learning. arXiv:2002.04833 [cs.LG]

[9] Minae Kwon, M. Li, Alexandre Bucquet, and D. Sadigh. 2019. Influencing Leading and Following in Human-Robot Teams. In *Robotics: Science and Systems*.

[10] Shen Li, Rosario Scalise, Henny Admoni, Siddhartha Srinivasa, and Stephanie Rosenthal. 2017. Evaluating critical points in trajectories. 1357–1364. https://doi.org/10.1109/ROMAN.2017.8172481

[11] Kent Lyons. 2016. Visual Parameters Impacting Reaction Times on Smartwatches. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Florence, Italy) *(MobileHCI '16)*. Association for Computing Machinery, New York, NY, USA, 190–194. https://doi.org/10.1145/2935334.2935344

[12] Mathew Monfort, Anqi Liu, and Brian D. Ziebart. 2015. Intent Prediction and Trajectory Forecasting via Predictive Inverse Linear-Quadratic Regulation. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (Austin, Texas) *(AAAI'15)*. AAAI Press, 3672–3678.

[13] Stefanos Nikolaidis, Keren Gu, Ramya Ramakrishnan, and Julie A. Shah. 2014. Efficient Model Learning for Human-Robot Collaborative Tasks. *CoRR* abs/1405.6341 (2014). arXiv:1405.6341 http://arxiv.org/abs/1405.6341

[14] Xavier Puig, Tianmin Shu, Shuang Li, Zilin Wang, Joshua B. Tenenbaum, Sanja Fidler, and Antonio Torralba. 2020. Watch-And-Help: A Challenge for Social Perception and Human-AI Collaboration. arXiv:2010.09890 [cs.AI]

[15] David V. Pynadath, Paul S. Rosenbloom, and Stacy C. Marsella. 2014. Reinforcement Learning for Adaptive Theory of Mind in the Sigma Cognitive Architecture. In *Artificial General Intelligence*, Ben Goertzel, Laurent Orseau, and Javier Snaider (Eds.). Springer International Publishing, Cham, 143–154.

[16] Siddharth Reddy, Anca D. Dragan, and Sergey Levine. 2018. Where Do You Think You're Going?: Inferring Beliefs about Dynamics from Behavior. *CoRR* abs/1805.08010 (2018). arXiv:1805.08010 http://arxiv.org/abs/1805.08010

[17] Jawad Jandali Refai, Scott Bateman, and Michael W. Fleming. 2020. External Assistance Techniques That Target Core Game Tasks for Balancing Game Difficulty. *Frontiers in Computer Science* 2 (2020), 17. https://doi.org/10.3389/fcomp.2020.00017

[18] Jan Roessingh, Gerald Poppinga, Joost van Oijen, and Armon Toubman. 2019. *Application of Artificial Intelligence to Adaptive Instruction - Combining the Concepts*. 542–556. https://doi.org/10.1007/978-3-030-22341-0_43

[19] Rosario Scalise, Shen Li, Henny Admoni, Stephanie Rosenthal, and Siddhartha S Srinivasa. 2018. Natural language instructions for human–robot collaborative manipulation. *The International Journal of Robotics Research* 37, 6 (2018), 558–565. https://doi.org/10.1177/0278364918760992 arXiv:https://doi.org/10.1177/0278364918760992

[20] Raphael Schumann and Stefan Riezler. 2020. Generating Landmark Navigation Instructions from Maps as a Graph-to-Text Problem. arXiv:2012.15329 [cs.CL]

[21] Mark Woodward, Chelsea Finn, and Karol Hausman. 2019. Learning to Interactively Learn and Assist. *CoRR* abs/1906.10187 (2019). arXiv:1906.10187 http://arxiv.org/abs/1906.10187

[22] Annie Xie, Dylan P. Losey, Ryan Tolsma, Chelsea Finn, and Dorsa Sadigh. 2020. Learning Latent Representations to Influence Multi-Agent Interaction. arXiv:2011.06619 [cs.RO]