Too many cooks: Coordinating multi-agent collaboration through inverse planning

Too many cooks: Coordinating multi-agent collaboration through inverse planning

Sarah A. Wu* MIT sarahawu@mit.edu

Rose E. Wang* MIT rewang@mit.edu

James A. Evans UChicago jevans@uchicago.edu

Joshua B. Tenenbaum MIT jbt@mit.edu

David C. Parkes Harvard parkes@eecs.harvard.edu

Max Kleiman-Weiner Harvard, MIT & Diffeo maxkleimanweiner@fas.harvard.edu

Abstract

Collaboration requires agents to coordinate their behavior on the fly, sometimes cooperating to solve a single task together and other times dividing it up into sub-tasks to work on in parallel. Underlying the human ability to collaborate is theory-of-mind, the ability to infer the hidden mental states that drive others to act. Here, we develop Bayesian Delegation, a decentralized multi-agent learning mechanism with these abilities. Bayesian Delegation enables agents to rapidly infer the hidden intentions of others by inverse planning. These inferences enable agents to flexibly decide in the absence of communication when to cooperate on the same sub-task and when to work on different sub-tasks in parallel. We test this model in a suite of multi-agent Markov decision processes inspired by cooking problems. To succeed, agents must coordinate both their high-level plans (e.g., what sub-task they should work on) and their low-level actions (e.g., avoiding collisions). Bayesian Delegation bridges these two levels and rapidly aligns agents' beliefs about who should work on what. Finally, we tested Bayesian Delegation in a behavioral experiment where participants made sub-task inferences from sparse observations of cooperative behavior. Bayesian Delegation outperformed heuristic models and was closely aligned with human judgments.

Keywords: coordination; social learning; inverse planning; Bayesian inference

Introduction

Working together enables a group of agents to achieve together what no individual could achieve on their own (Tomasello, 2014; Henrich, 2015). However, collaboration is challenging as it requires agents to coordinate their behaviors. In the absence of prior experience, social roles, and norms, we still find ways to negotiate our joint behavior in any given moment to work together with efficiency (Tomasello, Carpenter, Call, Behne, and Moll, 2005; Misyak, Melkonyan, Zeitoun, and Chater, 2014). Whether we're writing a scientific manuscript with collaborators or preparing a meal with friends, core questions we ask ourselves are: how can I help out the group? What should I work on next, and with whom should I do it with? Coordination unfolds over many timescales and these commonsense abilities are at the core of human social intelligence. In order to build social machines we must engineer AI systems that can coordinate with us and with each other as rapidly and as flexibly as people do (Lake, Ullman, Tenenbaum, and Gershman, 2017).

Central to this challenge is that agents' reasoning about what they should do in a multi-agent context requires knowledge about the future actions and intentions of others. When agents, like people, make independent decisions, these intentions are unobserved. Actions can reveal information about intentions, but predicting them is difficult because of uncertainty and ambiguity – multiple intentions can produce the same action. In humans, the ability to understand intentions from actions is called theory-of-mind (ToM). Humans rely on this ability to cooperate in coordinated ways, even in novel situations (Tomasello et al., 2005; Shum, Kleiman-Weiner, Littman, and Tenenbaum, 2019). We aim to build agents that have these kinds of abilities and show that they are powerful building blocks for coordinated cooperation.

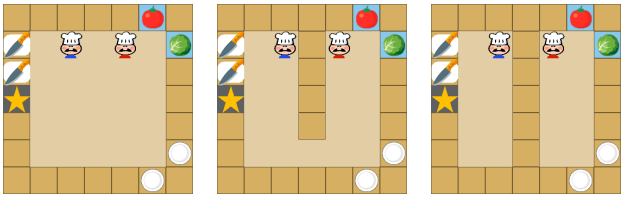
In this work, we study these abilities in the context of multiple agents cooking a meal together, inspired by the video game *Overcooked* (Ghost Town Games, 2016). These problems have hierarchically organized sub-tasks and share many features with other object-oriented tasks such as construction and assembly. Cooking tasks are challenging because of the sheer variation that is present: no kitchen or recipe is exactly alike, so successful collaboration requires flexible and abstract mechanisms for coordination. These tasks highlight three coordination challenges that any decentralized multi-agent system must grapple with: (A) Divide and conquer: agents should work in parallel when sub-tasks can be carried out individually, (B) Cooperation: agents should work together on the same sub-task when most efficient or necessary, (C) Spatio-temporal movement: agents should avoid getting in each other's way at any time.

To illustrate, imagine the process required to make a simple salad: first chopping both tomato and lettuce and then assembling them together on a plate. Two people might collaborate by first dividing the sub-tasks up: one person chops the tomato and the other chops the lettuce. This doubles the efficiency of the pair by completing sub-tasks in parallel (challenge A). On the other hand, some sub-tasks may require multiple to work together. If only one person can use the knife and only the other can reach the tomatoes, then they must cooperate to chop the tomato (challenge B). In all cases, agents must coordinate their low-level actions in space and time to avoid interfering with others and be mutually responsive (challenge C).

Contributions

We develop *Bayesian Delegation*, a learning mechanism that enables agents to rapidly infer the sub-tasks of other agents

*indicates equal contribution



(a) (b) (c)

Figure 1: The Overcooked environments. The movable objects start in the same locations; only the counters differ. Different recipes are possible in each environment, allowing for variation in high-level goals while keeping the low-level navigation challenges fixed. In (a) *Open-Divider*, agents can move between both sides of the kitchen. In (b) *Partial-Divider*, agents can only pass through a narrow bottleneck. In (c) *Full-Divider*, agents are confined to one half of the space.

based on theory-of-mind. It uses Bayesian inference to capture uncertainty over other agents' intentions and computes the likelihood of different intentions by inverse planning. We quantitatively study the behavior of this mechanism in a suit of 2D grid worlds. When Bayesian Delegation is used in conjunction with a model-based reinforcement learner, it achieves ad-hoc decentralized coordination across sub-tasks and overcomes the coordination challenges highlighted above. Finally, behavioral experiments show that the inferences made by our model closely align with those made by humans.

Related Work

Our work builds on a long history of using cooking tasks for evaluating multi-agent and AI systems and coordinating multi-agent plans across hierarchies of sub-tasks (Grosz and Kraus, 1996; Cohen and Levesque, 1991; Tambe, 1997). Most recently, *Overcooked* has inspired work multi-agent deep learning where agents are trained using self-play and human data (Carroll, Shah, Ho, Griffiths, Seshia, Abbeel, and Dragan, 2019). This approach requires large amounts of training experience for a specific environment, and studies only one task structure. In contrast, the abstractions in our agents generalize to multiple tasks and environments including those that they have no prior experience with.

Bayesian Delegation is inspired by the cognitive science of how people coordinate their cooperation in the absence of communication (Kleiman-Weiner, Ho, Austerweil, Littman, and Tenenbaum, 2016). Our approach most closely follows from recent work on Bayesian theory-of-mind (ToM) (Ramırez and Geffner, 2011; Nakahashi, Baker, and Tenenbaum, 2016; Baker, Jara-Ettinger, Saxe, and Tenenbaum, 2017) and learning statistical models of others (Barrett, Stone, Kraus, and Rosenfeld, 2012; Melo and Sardinha, 2016). Our hierarchical planning architecture also builds on previous work linking low-level navigation to high-level sub-tasks (Amato, Konidaris, Kaelbling, and How, 2019).

Our approach is fully decentralized in both task delegation and action coordination, thus contrasting with existing schemes for allocating tasks such as a centralized auctioneers (Brunet, Choi, and How, 2008; McIntire, Nunes, and Gini, 2016) or consensus approaches (see (Brunet et al., 2008) for a review). It draws from other decentralized multi-agent planning approaches in which agents aggregate the effects of others and best respond (Claes, Robbel, Oliehoek, Tuyls, Hennes, and Van der Hoek, 2015). These prior works focus on *spatially distributed* tasks, however, which addresses challenges A and C but not B. We extend them by considering effects over unobserved sub-tasks instead of being limited to observable features (i.e. location) and generalizing to object-oriented formalism which increases planning complexity.

Multi-Agent MDPs with Sub-Tasks

A multi-agent Markov decision process (MMDP) with sub-tasks is described as a tuple $\langle n, \mathcal{S}, \mathcal{A}_{1\ldots n}, T, R, \gamma, \mathcal{T} \rangle$ where n is the number of agents (Boutilier, 1996). $s \in \mathcal{S}$ are object-oriented states specified by the locations, status and type of each object and agent in the environment (Diuk, Cohen, and Littman, 2008). The environment state is fully observable to all agents. $\mathcal{A}_{1\ldots n}$ is the joint action space with $a_i \in \mathcal{A}_i$ being the set of actions available to agent i; each agent chooses its own actions independently. $T(s, a_{1\ldots n}, s')$ is the transition function which describes the probability of transitioning from state s to s' after all agents act $a_{1\ldots n}$. $R(s, a_{1\ldots n})$ is the reward function shared by all agents and γ is the discount factor. Each agent aims to find a policy $\pi_i(s)$ that maximizes expected discounted reward. Although agents fully observe the state of the environment, they do not observe the policies $\pi_{-i}(s)$ ($-i$ refers to all other agents except i) or any other internal representations of others.

Unlike traditional MMDPs, the environments we study have a partially ordered set of sub-tasks $\mathcal{T} = \{\mathcal{T}_0 \ldots \mathcal{T}_{|\mathcal{T}|}\}$. Each sub-task $\mathcal{T}_i$ has preconditions that specify when a sub-task can be started, and postconditions that specify when it is completed. They provide structure when R is very sparse and are also targets of high-level coordination between agents. In this work, all sub-tasks can be expressed as `Merge(X,Y)`, that is, to bring `X` and `Y` into the same location. Importantly, `Merge` does not specify how the merge should happen or who should do the sub-task. In the cooking environments we study here, the partial order of sub-tasks refers to a "recipe" (see Figure 2 for recipes and their partial orders of sub-tasks).

Coordination Test Suite in Overcooked

We now describe the environment test suite for evaluating the multi-agent models. Each environment is a 2D grid-world kitchen containing various objects and posing different coordination challenges, as shown in Figure 1. Agents can move north, south, east, west or stay still. All agents move simultaneously. They cannot move through each other, into the same space, or through counters. If they try to do so, they remain in place instead. The environment terminates after either the agents deliver the finished recipe to the star square

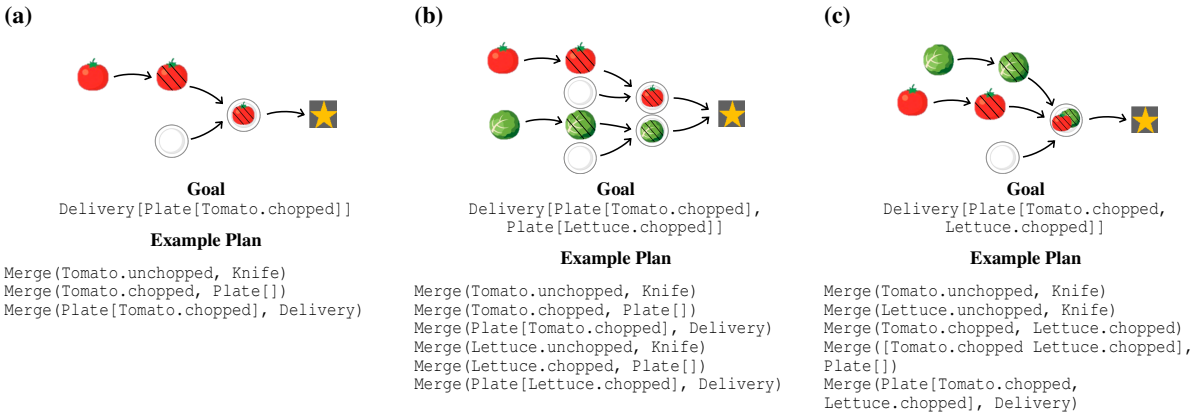


Figure 2: Recipes and example partial orderings. All sub-tasks are expressed in the `Merge` operator. In (a) *Tomato*, the task is to take an unchopped tomato and then chop, plate, and deliver it. In (b) *Tomato+Lettuce*, the task builds on *Tomato* and also requires chopping, plating, and delivering a piece of lettuce. In (c) *Salad*, the two chopped foods are combined on a single plate and delivered. The example plans show one possible ordering for completing the recipe, but other permutations are possible.

or 100 time steps elapse. The kitchens are built from counters that may contain both moveable food and dishware (e.g., tomatoes, lettuce, plates) and immoveable stations (e.g. knife stations). Agents may interact with objects by picking them up, putting them down, or transforming foods into a chopped state at a knife station. Agents can only carry one object at a time and cannot directly pass to each other.

This test suite allows us to evaluate models based on the coordination challenges raised in the introduction. The recipes assess rapid convergence–for instance, *Salad* can be assembled in multiple ways and agents' plans must align–and the spatial layouts provide opportunities for multiple agents to work together advantageously and/or avoid navigational obstacles. Thus, these environments enable us to study multi-agent coordination across levels of hierarchical planning.

Computational Model

We introduce a novel learning algorithm for multi-agent coordination based on probabilistic inference over sub-tasks called *Bayesian Delegation*. A high-level planner decides which sub-task should be done next, Bayesian Delegation models latent intentions in order to dynamically decide whether to divide-and-conquer or to cooperate, and a low-level planner finds approximately optimal policies for each sub-task. Note that planning is decentralized at both levels, i.e. agents plan and learn for themselves without sharing information with each other.

High-Level Planning (Sub-Task)

Inferring the sub-tasks others are working on enables each agent to efficiently select a single sub-task when multiple are possible. Bayesian Delegation works by having an agent maintain and update a belief state over the possible sub-tasks that all agents (including itself) are likely working on based on a history of observations.

Formally, Bayesian Delegation maintains a probability distribution over task allocations **ta**. For example, if there are two sub-tasks ($[\mathcal{T}_1, \mathcal{T}_2]$) and two agents ($[i, j]$), then $\mathbf{ta} = [(i : \mathcal{T}_1, j : \mathcal{T}_2), (i : \mathcal{T}_2, j : \mathcal{T}_1), (i : \mathcal{T}_1, j : \mathcal{T}_1), (i : \mathcal{T}_2, j : \mathcal{T}_2)]$ where $i : \mathcal{T}_1$ means that agent i is "delegated" to sub-task $\mathcal{T}_1$. Thus, **ta** includes the possibility of divide and conquer (separate sub-tasks) and cooperation (shared sub-tasks). If all agents pick the same $ta \in \mathbf{ta}$, then they will easily coordinate. However, in our environments, agents cannot communicate before or during execution, so they maintain uncertainty about which ta the group is coordinating on, $P(ta)$.

At every time step, each agent selects the most likely allocation $ta^* = \arg\max_{ta} P(ta|H_{0:T})$, where $P(ta|H_{0:T})$ is the posterior over ta after having observed a history of actions $H_{0:T} = [(s_0, \mathbf{a_0}), \ldots (s_T, \mathbf{a}_T)]$ over the first T timesteps, and $\mathbf{a}_t$ is all agents' actions at time step t. The agent then plans the next best action according to ta^* using the low-level planner. This posterior is computed at time step T according to Bayesian inference:

$$P(ta|H_{0:T}) \propto P(ta)P(H_{0:T}|ta) \quad (1)$$
$$= P(ta)\prod_{t=0}^{T} P(\mathbf{a_t}|s_t, ta)$$

where $P(ta)$ is the prior over ta and $P(\mathbf{a_t}|s_t, ta)$ is the likelihood of actions at time step t for all agents. The priors are initialized to the inverse distance between the two objects specified in the `Merge` operator. Note that these belief updates do not explicitly consider what each agent knows about their own sub-tasks at time $T-1$, but rather what is known by all, i.e., to a third-party observer (Nagel, 1986). The likelihood of a given ta is the likelihood that each agent is following their assigned task in that ta. It is computed as:

$$P(\mathbf{a_t}|s_t, ta) \propto \prod_{i:\mathcal{T} \in ta} exp(\beta * Q^*_{\mathcal{T}_i}(s, a_i)) \quad (2)$$

where $Q^*_{\mathcal{T}_i}(s, a_i)$, is the expected future reward of a towards the completion of sub-task $\mathcal{T}_i$ for agent i. This is computed

by low-level planning (see below) with a soft-max to account for non-optimal and variable behavior. β controls the degree to which an agent believes other agents are optimizing. When $\beta \to 0$, the agent believes others are acting randomly. When $\beta \to \infty$, the agent believes others are perfectly maximizing. Since the likelihood is computed by planning, this approach to posterior inference is called inverse planning. Note that even though agents see the same history of states and actions, their Bayesian updates are not necessarily the same because updates come from $Q_{\mathcal{T}_i}$, which is affected by randomness in exploration and the computational limit set on exploration.

Low-Level Planning (Action)

Low-level planning grounds sub-tasks into context-sensitive actions and provides the critical likelihood for Bayesian Delegation (see Equation 1). Low-level planning takes the *ta* selected by Bayesian Delegation and the next best action while modeling the movements of other agents. In this work, we use bounded real-time dynamic programming (BRTDP) extended to a multi-agent setting (McMahan, Likhachev, and Gordon, 2005). We use BRTDP to express $V^b_{\mathcal{T}_i}(s) = \min_{a \in \mathcal{A}_i} Q^b_{\mathcal{T}_i}(s,a)$ and $Q^b_{\mathcal{T}_i}(s,a) = C_{\mathcal{T}_i}(s,a) + \sum_{s' \in S} T(s'|s,a) V^b_{\mathcal{T}_i}(s')$ in terms of cost, where C is cost and $b = [l,u]$ is the lower and upper bound respectively. Each timestep is penalized by 1 and movement (as opposed to staying still) by an additional 0.1. This cost structure incentivizes efficiency. For details on how BRTDP updates on V and Q, see (McMahan et al., 2005).

Agents can use ta^* from the high-level planner to address two types of low-level coordination problems in relation to others: (1) avoiding collisions while working on distinct sub-tasks, and (2) cooperating as necessary to solve a shared sub-task. Each *ta* provides each agent i a hypothesis about the sub-tasks carried out by others, $\mathcal{T}_{-i}$. When $\mathcal{T}_i \neq \mathcal{T}_{-i}$, agent i addresses the first coordination problem: it creates simple models of the others performing $\mathcal{T}_{-i}$ and best responds. This process finds level-0 policies of other agents, $\pi^0_{\mathcal{T}_{-i}}(s)$, which assume that all other agents except for $-i$ are static. These level-0 models are used to reduce the multi-agent transition function to a single agent problem T' where the transitions of the other agents are assumed to follow the level-0 policies, $T'(s'|s,a_{-i}) = \sum_{a_i} T(s'|s,a_{-i},a_i) \prod_{A \in -i} \pi^0_{\mathcal{T}_A}(s)$. Running BRTDP on this transformed environment finds an approximately optimal level-1 policy $\pi^1_{\mathcal{T}_i}(s)$ for agent i that best responds to the level-0 models of the other agents.

When $\mathcal{T}_i = \mathcal{T}_{-i}$, agent i address the second coordination problem, i.e. it attempts to work together on the same sub-task with another agent. The agent simulates a fictitious centralized planner that controls the actions of all agents working together on the same sub-task (Kleiman-Weiner et al., 2016). This transforms the action space: if both i and j are working on $\mathcal{T}_i$, $\mathcal{A}' = a_i \times a_j$. Joint policies $\pi^J_{\mathcal{T}_i}(s)$ can similarly be found by single-agent planners such as BRTDP. An agent can find its own role by playing the action assigned to it under $\pi^J_{\mathcal{T}_i}(s)$. Cooperative planning enables emergent decentralized cooperative behavior—agents pass objects across the counters when efficient even though there was nothing about passing encoded into the environment and different agents may find different policies.

Computational Experiments

We conduct a series of computational simulations to test our agents' abilities to generate coordinated behavior in our suite of cooking environments. Our aim is to investigate how Bayesian Delegation combined with joint planning is important for successful coordination. We evaluate our model alongside agent "lesions" that use simplified representations or alternative planning strategies. All tested models take advantage of the sub-task structure because end-to-end optimization of the full recipe never succeeded under our computational budget.

Experimental Setup

We first describe the lesioned agents. The first alternative agent lesions Bayesian Delegation (`NBD`). `NBD` starts with the same priors over sub-tasks as the full agent but these priors are not updated during interaction. This allows us to investigate the importance of dynamically updating beliefs throughout the interaction. The second alternative model lesions joint planning (`NJP`). `NJP` has Bayesian Delegation to update beliefs, but it does not have the capacity to plan cooperatively with another agent on the same sub-task. A third alternative model lesions both Bayesian Delegation and joint planning (`NBD+NJP`). `NBD+NJP` selects the best available sub-task greedily without considering the sub-tasks the other agents are working on.

When any of the agents did not have a valid sub-task, they were programmed to take an action selected uniformly at random with probability 0.5 and to stay in place otherwise. This greatly improves the performance of the lesioned agents since without this noise, they often get stuck and block each other from completing the recipe. It has no effect on the full agent.

To highlight the differences between our model and the alternatives, let us consider the example from the High-Level Planning section with two possible sub-tasks ($[\mathcal{T}_1, \mathcal{T}_2]$) and two agents ($[i,j]$). Our model would propose $\mathbf{ta} = [(i:\mathcal{T}_1, j:\mathcal{T}_2), (i:\mathcal{T}_2, j:\mathcal{T}_1), (i:\mathcal{T}_1, j:\mathcal{T}_1), (i:\mathcal{T}_2, j:\mathcal{T}_2)]$ where $i:\mathcal{T}_1$ means that agent i is assigned to sub-task $\mathcal{T}_1$. `NBD` proposes the same **ta**, but never updates its beliefs. `NJP` does not allow for shared sub-tasks, and thus reduces its set to $\mathbf{ta} = [(i:\mathcal{T}_1, j:\mathcal{T}_2), (i:\mathcal{T}_2, j:\mathcal{T}_1)]$. Lastly, `NBD+NJP` makes inferences and plans for itself so each agent i proposes $\mathbf{ta} = [(i:\mathcal{T}_1), (i:\mathcal{T}_2)]$. Note that j does not appear here because this lesion does not make inferences over other agents.

All simulations are replicated with the same 10 random seeds. Agents were evaluated in 9 combinations of kitchens and recipes (3 kitchens x 3 recipes). We measure performance by the time to complete the full recipe. BRTDP was run until the bounds converged ($\alpha = 0.01, \tau = 2$ see (McMahan et al., 2005) for usage) or for a maximum of 100 trajectories each with up to 75 roll-outs for all models. The softmax during inference used $\beta = 0.9$.

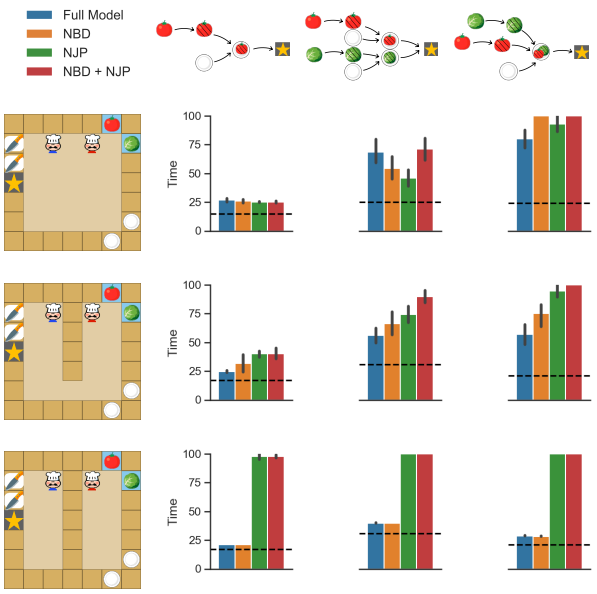


Figure 3: Number of timesteps needed to complete each kitchen-recipe composition (lower is better). The row shows the kitchen and the column shows the recipe. The full model is compared to three alternatives described in the text. The dashed lines show the optimal performance given no uncertainty in planning. Error bars are the standard error of the mean.

Results

Figure 3 shows the empirical results for recipe completion time, where our agent outperforms the alternatives on most compositions except for *Tomato+Lettuce* in *Open-Divider*. All four agents are comparable when given the recipe *Tomato* in *Open-Divider*. However, when faced with more complex situations, the lesioned agents failed to complete some of the tasks. For example, without the ability to do joint planning, `NJP` and `NBD+NJP` fail in *Full-Divider* because they cannot coordinate low-level actions to complete the sub-tasks, all of which require ad-hoc cooperation to pass objects across the counters. Here, `NBD` is comparable to the full agent, because inference does not contribute much when spatial constraints necessitate joint planning for all sub-tasks.

Without the ability to reason about other agents, `NBD` and `NBD+NJP` often fail to coordinate their high-level goals. This is particularly clear for the *Salad* recipe which requires merging three distinct objects. The lesioned agents frequently get stuck in cycles in which both agents are holding objects that must be merged. They cannot converge on a sequence of actions in which one agent puts their object down in order for the other to pick it up and use it. This highlights one key purpose of Bayesian Delegation – it can break symmetries by enabling actions that involve yielding to others so long as they make net progress towards the completion of one of the sub-tasks. An agent that only considers its own intentions is highly unlikely to coordinate in this way. With Bayesian Delegation, agents can infer another agent's intentions and yield to them. This results in an agent giving up its own task to complement its team when needed.

In the *Partial-Divider* environments, `NBD` outperforms `NJP`, which both take longer than the full agent but are faster than the greedy agent `NBD+NJP`. This suggests that in settings in which cooperation on sub-tasks is not necessary but is helpful, both joint planning and theory-of-mind inference abilities are important for successful coordination. Joint planning may be more important in these specific contexts since it opens up possibilities for sub-tasks to be completed in a more efficient manner.

Interestingly, *Open-Divider* poses the most difficulty for our agent. This environment is the most unconstrained, so agents' actions become more ambiguous and they cannot easily use the counters to coordinate their roles. For instance, in Figure 1a, the blue agent might move down in order to either grab a plate, move around the red agent towards the food items, or simply move out of the red agent's way. The full agent performs poorly on *Tomato-Lettuce*, which has the most sub-task combinations, possibly because the model must maintain beliefs over all sub-task allocations in spite of the ambiguity of actions. On the other hand, it outperforms alternatives for *Salad*, which has fewer sub-tasks but has the added complexity of merging three objects together. Compared to `NBD`, our full model dynamically changes to new tasks when other agents are close to completing their tasks. This feature prevents multiple agents from working on the same task in uncoordinated ways.

Human Experiments

We hypothesize that in addition to enabling better coordinated performance, Bayesian Delegation is human-like in its ability to rapidly infer the sub-tasks others are working on. To test this hypothesis, we designed a novel behavioral experiment where human participants observed the behavior of other agents and then made inferences about the sub-tasks each agent was carrying out. We compared the human inferences to those made by Bayesian Delegation as well as heuristics.

Experiment Setup

We designed six scenes of two agents working together in the same scenarios as in the previous experiment (Figure 4). These scenes include a variety of coordinated plans such as instances of clear task allocation (both joint planning and divide-and-conquer) and of ambiguous plans where the agent intentions become more clear over time as the interaction continues. Participants ($n = 45$) recruited through Amazon Mechanical Turk watched the agents interact and at different time points throughout the experiment made judgments about which sub-tasks each agent was working on. Participants made judgments on a continuous slider [0, 1] with endpoints labeled "not likely at all" to "certainly". Beliefs were normalized into a probability distribution for each subject and then averaged across all subjects. These averages were compared

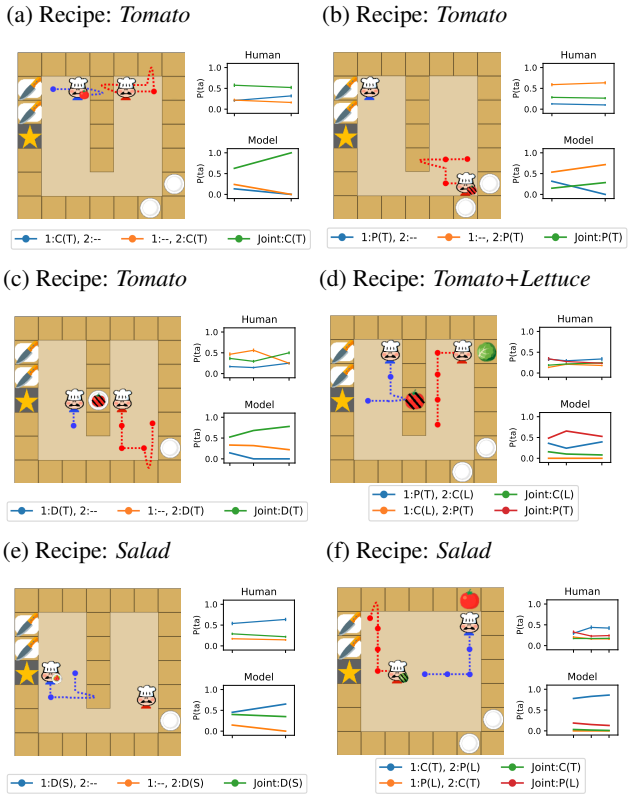


Figure 4: Scenarios and results for the human behavioral experiment. Each agent's past trajectory is illustrated by a dotted path, with sharp curves into counters representing picking up or putting down an object. To the right of each trajectory are the inferences made by the model and the average participant inference. The legend notes the possible task allocations of agents (1 or 2) working individually or together (Joint): C = chop, P = plate, D = deliver, T = tomato, L = lettuce, and S = salad. E.g., 1:C(T) refers to Agent 1 chopping the tomato. Error bars are the standard error of the mean.

to the beliefs formed by our model ($P(ta|H)$) after observing the same trajectory H. Each participant made 51 judgments.

Results

Figure 4 shows participant and model inferences for each scenario at each time step. Overall, there is a close correspondence between model and human judgments. Figure 5 quantifies this comparison. The inferences made by Bayesian Delegation correlate closely with the human judgments ($R = 0.732$). We also compared the full model to two lesions. Neither lesion does Bayesian updating based on the observed agent interaction. The first lesion uses a spatial prior but does not use the likelihood to update beliefs over the course of the trajectory. The second uses a uniform prior and does not update. Figure 5 shows that these lesions are less aligned with the human judgments illustrating the importance of updating beliefs over time for accurately modeling human judgments of coordinated interactions.

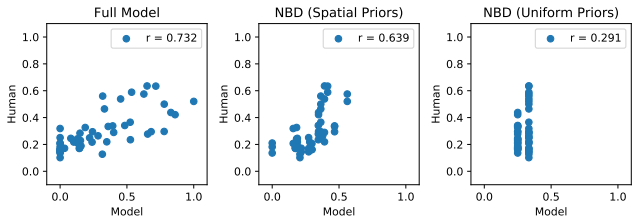


Figure 5: Correlation between human and model inferences. (left) full model, NBD lesions with (middle) spatial or (right) uniform priors.

Discussion

We developed a new set of multi-agent, object-oriented cooking challenges that require coordinated strategies to successfully complete. We developed an abstract approach inspired by and consistent with human theory-of-mind: Bayesian Delegation, which rises to these challenges and learns to coordinate in a decentralized way. Bayesian Delegation enables coordination through two key mechanisms: (1) Theory-of-mind through inverse planning that enables agents to rapidly infer the sub-tasks of others. With this, each agent dynamically aligns its beliefs about who is doing what, determining when it should help another agent on the same sub-task and when it should work in parallel on other sub-tasks. (2) Joint planning that enables agents to mesh their intentions and find coordinated low-level policies to complete sub-tasks in ways that neither agent could achieve on their own.

While Bayesian delegation solves some aspects of commonsense coordination, there are still limitations which we hope to address in future work. One challenge is that when agents jointly plan for a single sub-task, they currently have no way of knowing when they have completed their individual "part" of the joint effort. For instance, in the case where one agent needs to pass lettuce and tomato across the divider for the other to chop it, after dropping off the lettuce, the first agent is currently unable to reason that it has fulfilled its role in that joint plan and can move on, i.e., that the rest of the sub-task depends only on the actions of the other agent. If agents were able to recognize when their sub-tasks were finished with respect to themselves, then they would be able to coordinate even more efficiently and flexibly. This opens the possibility of looking ahead to future sub-tasks that will need to be done even before their preconditions are satisfied. For example, once an agent passes off a tomato to another to chop, the first agent can go and get a plate in anticipation of also passing that over even before the chopping has begun

Future work will also look at models that allow for agents to form longer term collaborations that persist beyond a single short interaction such as roles and norms. Such representations are essential for building AI agents that can partner with human teams and with each other.

Acknowledgments

We thank Alex Peysakhovich, Barbara Grosz, DJ Strouse, Leslie Kaelbling, Micah Carroll, and Natasha Jaques for insightful ideas and comments. This work was funded by the Harvard Data Science Initiate, Harvard CRCS, Templeton World Charity Foundation, The Future of Life Institute, DARPA Ground Truth, and The Center for Brains, Minds and Machines (NSF STC award CCF-1231216).

References

Christopher Amato, George Konidaris, Leslie Pack Kaelbling, and Jonathan P. How. 2019. Modeling and Planning with Macro-Actions in Decentralized POMDPs. *Journal of Artificial Intelligence Research* 64 (2019), 817–859.

Chris L Baker, Julian Jara-Ettinger, Rebecca Saxe, and Joshua B Tenenbaum. 2017. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour* 1 (2017), 0064.

Samuel Barrett, Peter Stone, Sarit Kraus, and Avi Rosenfeld. 2012. Learning teammate models for ad hoc teamwork. In *AAMAS Adaptive Learning Agents (ALA) Workshop*. 57–63.

Craig Boutilier. 1996. Planning, learning and coordination in multiagent decision processes. In *Proceedings of the 6th conference on Theoretical aspects of rationality and knowledge*. Morgan Kaufmann Publishers Inc., 195–210.

Luc Brunet, Han-Lim Choi, and Jonathan How. 2008. Consensus-based auction approaches for decentralized task assignment. In *AIAA guidance, navigation and control conference and exhibit*. 6839.

Micah Carroll, Rohin Shah, Mark Ho, Thomas Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca Dragan. 2019. On the Utility of Learning about Humans for Human-AI Coordination. In *Advances in Neural Information Processing Systems*.

Daniel Claes, Philipp Robbel, Frans A Oliehoek, Karl Tuyls, Daniel Hennes, and Wiebe Van der Hoek. 2015. Effective approximations for multi-robot coordination in spatially distributed tasks. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 881–890.

Philip R Cohen and Hector J Levesque. 1991. Teamwork. *Noûs* 25, 4 (1991), 487–512.

Carlos Diuk, Andre Cohen, and Michael L Littman. 2008. An object-oriented representation for efficient reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*. ACM, 240–247.

Ghost Town Games. 2016. Overcooked. (2016).

Barbara J Grosz and Sarit Kraus. 1996. Collaborative plans for complex group action. *Artificial Intelligence* 86, 2 (1996), 269–357.

Joseph Henrich. 2015. *The secret of our success: how culture is driving human evolution, domesticating our species, and making us smarter*. Princeton University Press.

Max Kleiman-Weiner, Mark K Ho, Joseph L Austerweil, Michael L Littman, and Joshua B Tenenbaum. 2016. Coordinate to cooperate or compete: abstract goals and joint intentions in social interaction. In *Proceedings of the 38th Annual Conference of the Cognitive Science Society*.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. 2017. Building machines that learn and think like people. *Behavioral and Brain Sciences* 40 (2017).

Mitchell McIntire, Ernesto Nunes, and Maria Gini. 2016. Iterated multi-robot auctions for precedence-constrained task scheduling. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. 1078–1086.

H Brendan McMahan, Maxim Likhachev, and Geoffrey J Gordon. 2005. Bounded real-time dynamic programming: RTDP with monotone upper bounds and performance guarantees. In *Proceedings of the 22nd international conference on Machine learning*. ACM, 569–576.

Francisco S Melo and Alberto Sardinha. 2016. Ad hoc teamwork by learning teammates task. *Autonomous Agents and Multi-Agent Systems* 30, 2 (2016), 175–219.

Jennifer B Misyak, Tigran Melkonyan, Hossam Zeitoun, and Nick Chater. 2014. Unwritten rules: virtual bargaining underpins social interaction, culture, and society. *Trends in cognitive sciences* (2014).

Thomas Nagel. 1986. *The view from nowhere*. Oxford University Press.

Ryo Nakahashi, Chris L Baker, and Joshua B Tenenbaum. 2016. Modeling Human Understanding of Complex Intentional Action with a Bayesian Nonparametric Subgoal Model.. In *AAAI*. 3754–3760.

Miquel Ramırez and Hector Geffner. 2011. Goal recognition over POMDPs: Inferring the intention of a POMDP agent. In *IJCAI*. IJCAI/AAAI, 2009–2014.

Michael Shum, Max Kleiman-Weiner, Michael L Littman, and Joshua B Tenenbaum. 2019. Theory of Minds: Understanding Behavior in Groups Through Inverse Planning. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*.

Milind Tambe. 1997. Towards flexible teamwork. *Journal of artificial intelligence research* 7 (1997), 83–124.

Michael Tomasello. 2014. *A natural history of human thinking*. Harvard University Press.

Michael Tomasello, Malinda Carpenter, Josep Call, Tanya Behne, and Henrike Moll. 2005. Understanding and sharing intentions: The origins of cultural cognition. *Behavioral and brain sciences* 28, 05 (2005), 675–691.