Northumbria Research Link

Diversifying Agent's Behaviors in Interactive Decision Models

Yinghui Pan[1] | Hanyi Zhang[1] | Yifeng Zeng[2] |
Biyang Ma[2] | Jing Tang[2] | Zhong Ming[1]

[1]College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, 510810, China

[2]Department of Computer and Information Sciences, Northumbria University, Newcastle, NE7 7UX, UK

Correspondence
Yifeng Zeng, Department of Computer and Information Sciences, Northumbria University, Newcastle, NE7 7UX, UK
Ming Zhong, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, 510810, China
Email: yifeng.zeng@northumbria.ac.uk;

Modelling other agents' behaviors plays an important role in decision models for interactions among multiple agents. To optimise its own decisions, a subject agent needs to model what other agents act simultaneously in an uncertain environment. However, modelling insufficiency occurs when the agents are competitive and the subject agent can not get full knowledge about other agents. Even when the agents are collaborative, they may not share their true behaviors due to their privacy concerns. Most of the recent research still assumes that the agents have common knowledge about their environments and a subject agent has the true behavior of other agents in its mind. Consequently, the resulting techniques are not applicable in many practical problem domains. In this article, we investigate into diversifying behaviors of other agents in the subject agent's decision model prior to their interactions. The challenges lie in generating and measuring new behaviors of other agents. Starting with prior knowledge about other agents' behaviors, we use a linear reduction technique to extract representative behavioral features from the known behaviors. We subsequently generate their new behaviors by expanding the features and propose two diversity measurements to select top-K behaviors. We demonstrate the performance of the new techniques in two well-studied problem domains. The

top-K behavior selection embarks the study of unknown behaviors in multiagent decision making and inspires investigation of diversifying agents' behaviors in competitive agent interactions. This research will contribute to intelligent systems dealing with unknown unknowns in an open artificial intelligence world.

1 | INTRODUCTION

Understanding interactive behaviors facilitates the development of intelligent systems that involves interactions between either multiple agents or agents-and-humans. From the viewpoint of a subject agent, it expects to optimise its own decisions given what it observes from an environment shared by other agents who act in a similar way. However, the decision optimisation becomes difficult when the subject agent can not get full knowledge about the other agents. This often occurs in a setting of competitive agents where the subject agent can not obtain sufficient knowledge about others in the decision making process. For example, a ground force schedules its routine patrol based on the criminal activities without fully knowing what the criminals will precisely react. Even in a collaborative agent environment, the agents may not be willing to share their information with each other due to their privacy concerns. Hence, modelling other agents becomes very important in understanding behaviors so as to optimize agents' interaction and has attracted growing interests in the fields of artificial intelligence, decision science and general intelligent systems [1, 2].

There has seen various types of modelling languages to represent decision making, behavior reasoning and learning problems in different types of environments, e.g. from a classical model of fictitious play [3] to a probabilistic deterministic finite-state automaton for modelling stochastic actions in partially observable stochastic game (POSG) [4]. More sophisticated ones include recursive modelling methods that follow a nested reasoning form of what does agent A think that agent B thinks that agent A thinks (and so on), e.g. interactive dynamic influence diagrams (I-DIDs) [5, 6, 7] and interactive partially observable Markov decision processes (I-POMDPs) [8], and even more rigorous planning systems based on epistemic logic [9]. Recently, Ma *et al.* [10] used knowledge graph to model opponents' behaviors and inferred agents' intentions accordingly. In particular, a series of probabilistic graphical models have been proposed to solve multiagent decision making problems. Suryadi and Gmytrasiewicz [11] used influence diagrams to model other agents, but did not provide a mechanism to update the models upon a subject agent's observations. Koller and Milch [12] proposed multiagent influence diagrams (MAID) to compute Nash equilibrium strategies for all agents involved in the interaction while Gal and Pfeffer [13] developed networks of influence diagrams (NID) for recursively modelling other agents. Both MAID and NID formalisms focus on a static, single-shot interaction. In contrast, I-DIDs offer solutions over extended time interactions, where agents act and update their beliefs over others' models which are themselves dynamic. In this work, we focus on decision problems with a limited number of planning horizons in a multiagent setting.

In Figure 1, we show one example of interactions involving two agents (i and j) in POSG [14]. Both agents act when they receive observations from the environment and are awarded according to impact of their actions on the environmental states. From the viewpoint of agent i, it needs to predict behaviors of the other agent j which are not
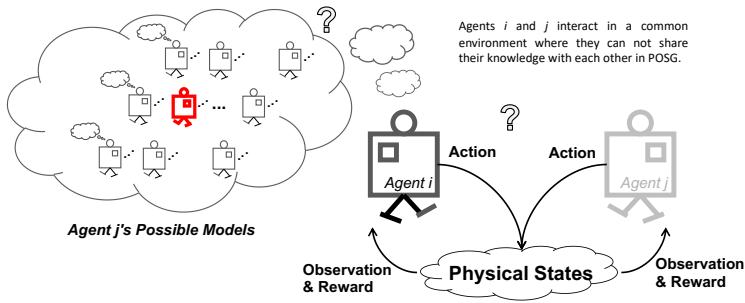
FIGURE 1 Agent i optimises its actions by solving the hypothesised models of the other agent j. In this case, the true model of agent j (the frame colored by red) is contained in the set of candidate models. However, it often occurs that agent i does not hold the true model of agent j due to the modelling insufficiency.

known prior to their interactions. Hence, agent i has to hypothesise a large number of agent j's possible models by solving which agent i can predict j's behaviors. Unfortunately, the set of candidate models may not contain the true model of agent j due to the modelling insufficiency. This leads to the challenge of optimising the subject agent i's decisions under the uncertainty of other agents' behaviors.

Solving decision problems without sufficient prior knowledge, also termed as modelling insufficiency, has been a long-standing issue in the field of uncertainty in artificial intelligence (UAI) [15] and opens a new challenge to deploy intelligent systems in an open AI world particularly through decision-making technologies [16, 17, 18]. Due to unknown behaviors (true models) of other agents, most of the research hypothesizes a large number of models for the other agents based on which a subject agent adapts its decisions even when their behaviors change over time (sometimes is referred as a dynamic opponent) [19]. This leads to a significantly increasing complexity in solving other agents' models. Hence, a large amount of research has been invested into reducing the model space of the other agents. For example, the concept of minimal mental models was used to compress candidate models of other agents so as to reduce the computational complexity due to introducing redundant potential models [20]. The behavioral equivalence principle becomes a commonly used technique to group candidate models that exhibit identical behaviors of other agents [21, 22]. Similarly, a value equivalence was used by Conroy [23] to cluster the models that have similar influence on a subject agent's expected rewards, which leads to more compressed model space. This line of work on compressing model space is also due to limited prior knowledge on building a good set of candidate behavior models. Hence, generating a good set of initial models becomes important since it may provide a good chance of containing the true models and the model compression techniques are not strictly required.

In this article, we consider modelling insufficiency in a well-known interactive multiagent decision model, e.g. interactive dynamic influence diagrams (I-DIDs) and interactive partially observable Markov decision processes (I-POMDPs), in a POSG setting [7, 19]. As demonstrated in the previous work, I-DIDs have shown both representation and computational advantages compared to I-POMDPs [24, 7]. Hence we develop I-DID-based solutions to the modelling insufficiency issue in this work and the solutions can also be generalized to other multiagent decision models. An I-DID model extends a single-agent dynamic influence diagram [25] to represent how a subject agent solves a sequential decision problem involving other agents in a common environment. In a POSG setting, both agents act simultaneously and they can not directly see what the others act and can only reason with their actions given what they receive from the environment. In addition, their observations do not fully reflect the environmental states in a deterministic manner, but with a probabilistic distribution.

A main component of a I-DID model is a subject agent i's dynamic influence diagram based on which the other agent j's behaviors are modelled and embedded into the decision model. Modelling insufficiency occurs where agent j's behaviors are not fully represented in an agent i's I-DID model. For example, if some actions of agent j are not modelled, agent i may not receive consistent observations in their real-time interactions. This is because agent j acts according to its own optimal decisions and this true behavior is not in agent i's mind when agent i builds its I-DID model. The issue of inconsistent observations was also identified in a general probabilistic graphical decision model [26]. Recently, Pan *et al.* [27] learned models of other agents from historical data of agents' interactions, but did not provide new models from the learned models. In this article, we aim to generate new models of other agents given limited knowledge about their behaviors and the new behaviors may not be seen in the previous interactions. To the best of our knowledge, our work is the first attempt to solve the modelling insufficiency issue in I-DIDs.

Given limited amount of prior knowledge about agent j's behaviors, we aim to develop a set of its new behaviors that are to be modelled in agent i's I-DID models [1]. The new behavior set expects to contain agent j's true behaviors in a large probability so as to increase agent i's decision quality. Since the prior knowledge provides essential information about the other agent, we first use a linear reduction technique to elicit behavioral features from the known behaviors. The features could reflect representative patterns of what the other agent acts given its beliefs about the environment. The reduction differs from the previous work on reducing agent j's behavior space since it is to find representative behavior patterns and does not aim to compress the behavior space [7, 28, 29].

Subsequently, we randomize a set of new behaviors for the other agent based on the behavioral features. The randomization is conducted by sampling the action that achieves the best reward for the other agent. By doing so, we can obtain a large set of candidate behaviors for the other agent. The issue lies in selecting a set of K behaviors, namely top-K behaviors, that hold a good chance of including the true behaviors of the other agent. We propose two new measurements that quantify the diversity of a set of behaviors and are used to optimise the top-K behavior selection. The first diversity measurement considers the difference between specific actions upon observations at each time step while the second one contains an extra factor of differentiating general behavior patterns over time. Intuitively, a set of diverse behaviors have a large chance of containing the true behaviors of the other agent which are not known by a subject agent prior to their interactions. Once we have the top-K behaviors, we can represent them in the subject agent's I-DID model and solve the decision model, which results in an optimal policy for the subject agent. Finally, we conduct comprehensive experiments in two well-known problem domains and show the performance of the new techniques. This work is one of very few attempts on developing behavior diversity in I-DID solutions and will inspire more interesting work in studying agent behaviors in intelligent systems. We summarize the main contributions of this work below.

- We develop a linear reduction technique to extract representative behaviors from known behaviors of other agents;
- We propose two quantitative measurements for evaluating the behavior diversity in the top-K behavior selection;
- We develop the top-K behavior selection procedure including sampling new behaviors through the new diversity measurements, and demonstrate its empirical performance in two problem domains.

We organise this article as follows. We present the background knowledge of I-DID models in Section 2. In Section 3, we propose the new methods in generating a set of new behaviors through top-K behavior selection. We demonstrate the empirical performance of our proposal in Section 4 and conclude this work with discussions in

---

[1] To facilitate the presentation, we elaborate the development using two agents (agents i and j) in the following discussion; however, the techniques are general for a multiagent setting.

Section 5.

2 | BACKGROUND KNOWLEDGE ON I-DIDS

As we will develop methods to generate new behaviors for other agents in I-DID models, we proceed to provide background knowledge on I-DIDs. By extending influence diagrams [30] that are commonly used to represent a single-agent decision making problem, I-DIDs are a general probabilistic graphical representation for solving interactive multiagent decision making problems under uncertainty [6]. From the viewpoint of a subject agent, an I-DID model represents how it solves the decision problems when considering other agents' behaviors that impact their decision outcomes. It integrates multiagent game theory into individual decision making frameworks. Details about I-DIDs can be found in the previous research [7].

Figure 2(a) shows a dynamic influence diagram (DID - a dynamic version of influence diagrams over time) for a single agent who plans its decisions over three time steps. In the DID model, a chance node (denoted by a circle/oval shape) represents environmental states (S) and observations (O) received by the agent, a decision node (denoted by a rectangular shape) represents the agent's decisions (A) and a utility node (denoted by a diamond shape) models the agent's rewards (R, also called as a utility function) upon the states and decisions. The agent decision (A) is directly influenced by what it observes (O) from the environment while the environmental states (S) are not fully observable. To quantify the strength of variables' relations, we associate the arcs with conditional probabilities ($Pr(\cdot|\cdot)$) while the arcs into the utility node models actual reward/utility values. For example, $Pr(S^3|S^2, A^2)$ is a transition function from S^2 to S^3 given the effect of the action A^2 at time $t = 2$. $Pr(O^3|S^3, A^2)$ is an observation function that models the probability of how the observations reflect the true environmental states.
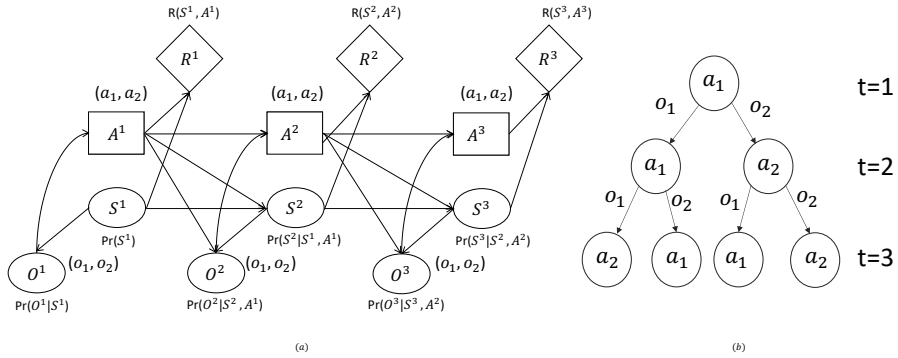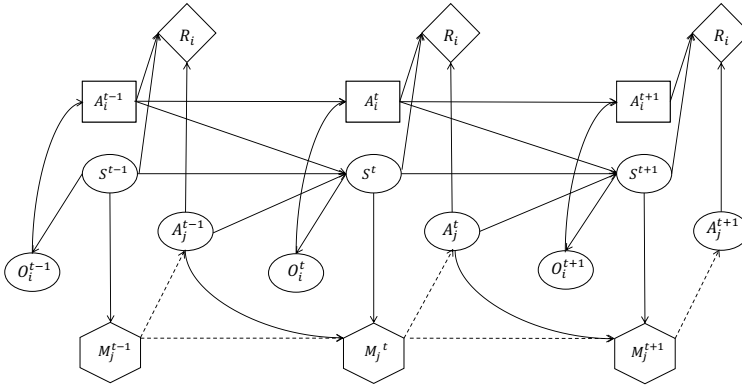

FIGURE 2 A dynamic influence diagram and its solutions: (a)the dynamic influence diagram with three time steps; and (b) its solution is represented as a policy tree.

Once we have the parameter specification, e.g. the transition, observation and utility functions, in a dynamic influence diagram, we can solve the model through well-developed inference algorithms [30] and obtain an optimal plan for the subject agent. The plan prescribes how the agent shall act upon its observations at each time step and is often represented by a policy tree. For example, Fig. 2(b) shows one policy tree as the optimal plan given the dynamic influence diagram in Fig. 2(a). The agent takes action a_1 at the first time step t=1, and repeats the action if it receives observation o_1; otherwise, it takes a_2 at t=2. As a solution to a dynamic influence diagram, a policy tree represents

the optimal plan according to which a subject agent behaves over times. In general, the policy tree is termed as a behavioral model - optimal decision outcomes from a decision model.

An I-DID model extends dynamic influence diagrams for multiple agents by introducing the model node M_j (the hexagon node) in the model, as shown in Fig. 3. The model node contains possible models of the other agent j each of which is either a decision model, e.g. dynamic influence diagrams, or a behavioral model. Once the subject agent expands the model node with possible models of other agents, it can solve the I-DID model through conventional dynamic influence diagram algorithms therefore resulting in an optimal plan for the subject agent. As the true model is not known by the subject agent i, the number of candidate models contained in the model node tends to rather large. In theory, it requires an infinite number of j's candidate models so that a subject agent can sufficiently predict j's behaviors in order to optimise its decisions. The computational limit prevents a large model space in the model node within an I-DID model. Hence, there is a significant amount of research on dealing with the compression of the model space in the I-DID [22, 27].

FIGURE 3 By extending dynamic influence diagrams, the I-DID model represents how the subject agent i optimises its decisions by modelling the other agent j's behaviors over three time steps.

Most of the current I-DID research still assumes that the true behavior is within the model node (M_j) and does not handle unknown behaviors of other agents that are not contained in the model node [22, 27, 31]. One barrier to deal with the case of unknown behaviors is partially due to the incapability of generating a good set of candidate behaviors for other agents. The current work always stands on the pre-defined models that lead to a set of monotonic behaviors for other agents. Consequently, there is a rather large chance that the true behaviors slip from the model node in a subject agent's I-DID model. In this article, we will propose a novel model generation approach to modelling unknown behaviors in an I-DID model and focus on measure the quality of the new set of generated behaviors.

3 | TOP-κ BEHAVIOR SELECTION

Intuitively, a large set of new behaviors have the potential of increasing the chance to include the true behaviors of other agents. However, due to the computational constraints, solving I-DID models becomes infeasible when the set is too large as shown in the previous I-DID research [7]. Hence, the set of new behaviors expect to have a good diversity while keeping a reasonable size. In addition, the new behaviors can not be fully randomised since we have prior knowledge (although it is limited) about other agents' behaviors. Accordingly, we proceed to exploit the

knowledge by eliciting representative behaviors from the known behaviors, based on which the set of new behaviors are generated and measured in terms of their diversity.

Figure 4 describes our main methods in generating top-K behaviors for other agents. Based on the known behaviors of the size M, we adapt a linear reduction technique to extract m behavior sequences that represent important features of all the M behaviors (in Section 3.1). Subsequently, we propose two measurements to quantify the diversity of the behavior set and choose the top-K behaviors from the sampled behaviors (in Section 3.2). In addition, we analyse the property of representative behavior sequences and the complexity of the top-K selection algorithm.
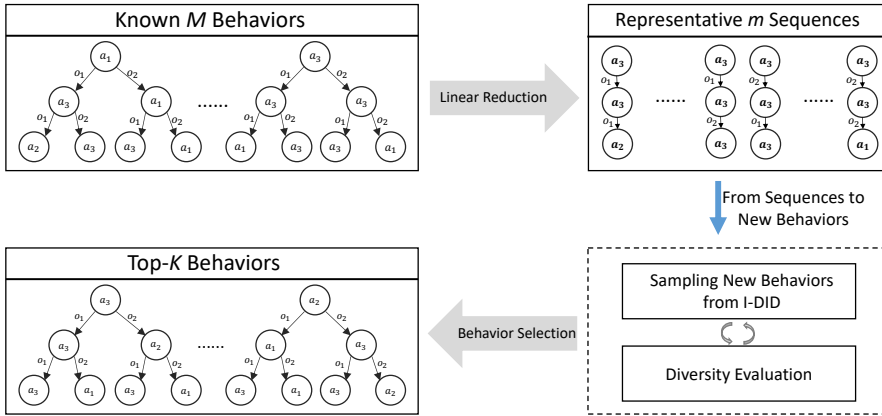
FIGURE 4 The main components include extracting representative behavior sequences and generating new behaviors for other agents in the top-K behavior selection.

3.1 | Eliciting Representative Behaviors

An agent's behavior prescribes what the agent shall do given its observations in an environment. By solving a decision model, e.g. dynamic influence diagrams in Fig. 2(a), we can obtain the agent's optimal policy that contains a set of actions given various observations received by the agent over time. In general, the policy can be represented in a tree structure as shown in Fig. 2(b). Each branch of the policy tree is a behavior sequence specifying the agent's optimal action given a possible observation at each time step. We formally define a behavior sequence below.

Definition 1 (Behavior Sequence) *For agent j, a behavior sequence, $h_T^j = (a_1, o_2, a_2, \cdots, o_{T-1}, a_T)$ where a_t ($\in A$) and o_t ($\in \Omega$), is a set of alternating actions and observations over its planning horizon T.*

Subsequently, we can define a policy tree that is composed of a set of behavior sequences.

Definition 2 (Policy Tree) *A policy tree for agent j is a set of behavior sequences, $H_T^j = \bigcup h_T^j$, that are organized as a tree structure with the depth T where actions are in the nodes while observations are attached to the branches in the tree.*

Following a policy tree, an agent executes a behavior sequence when it receives a particular observation from an environment at each time step. In the policy tree example in Fig. 2(b), the agent takes the action a_1 at the first time step and executes the actions a_2 and a_1 given its observations o_2 and o_1 respectively at the second and third time steps.

A policy tree represents the agent's behavior that can be obtained by solving its decision model. Variations of parameters, e.g. probability distributions in a dynamic influence diagram, in the agent's decision model could lead to different behaviors. From the viewpoint of a subject agent i, what does matter is the behaviors exhibited by the other agent j, not how agent j optimises its behaviors through a decision model. Hence, in this article, we will focus on j's behavior representation and assume that agent i knows a set of M behaviors for the other agent j. The question remains on generating a new set of behaviors ($\geq M$ including the known behaviors) for agent j based on the known M behaviors. The set of new behaviors are expected to include potentially true behaviors of the other agent j.

The known M behaviors represent basic types of how agent j behaves according to agent i's prior knowledge about agent j. They can serve as salient features to expand new behaviors of agent j. We proceed to extract a set of behavior sequences from the known agent j's behaviors based on which its new behaviors are to be generated. The behavior sequences are representative of agent j behaviors that are known to agent i.
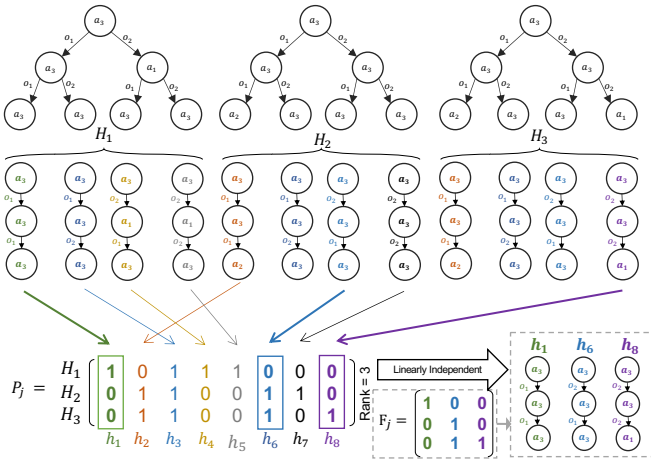
FIGURE 5 A set of behavior sequences, $F_j=(h_1, h_6, h_8)$, are selected from known behavior matrix P_j for agent j. They represent important features of the agent j's known behaviors.

Instead of randomly selecting m behavior sequences, $F_j=\{h_1, \cdots, h_m\}$, from M behaviors, $\mathcal{H}=\{H_1, \cdots, H_M\}$ [2], we use a linear reduction method to extract the sequences that have most sufficient features from the known behaviors. We compose a behavior matrix, namely P, where each row is a policy tree H_m and each column is one behavior sequence seen in the policy trees. Since some behavior sequences could appear in different policy trees, the column dimension is smaller than $M \times |\Omega|^{T-1}$, but much larger than the row dimension M particularly for a large planning horizon T. The matrix element $P(H_i, h_j)$ is 1 if the sequence h_j appears in the behavior H_i; otherwise, $P(H_i, h_j)$ = 0. Hence, the matrix P may contain some linearly dependent behavior sequences so that its columns could be reduced into a set of representative sequences F. To extract the linearly independent sequences F, we use a *Gaussian* elimination method to find the pivot columns in the large matrix P. The extraction process can be conducted in a polynomial time [32]. In principle, we find one pivot matrix F_j (corresponding to the linearly independent sequences F [3]) and the other matrix U_j to ensure: $P_j=F_j \times U_j$.

---

[2] We use the subscript of H according to the context. Here it numbers the policy tree in the set while it may refer to the length of a policy tree as defined in Def. 2. It shall be self-evident in the context.

[3] To simplify the presentation, we do not differentiate the two terms between a pivot *matrix* and *linearly independent sequences*.

In Fig. 5, we elaborate the extraction of behavior sequences from three behavior models $\mathcal{H} = \{H_1, H_2, H_3\}$. The behavior matrix P_j has the rank of three so that we could obtain the pivot matrix $F_j = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$ that returns three linearly independent sequences, e.g. $F_j = (h_1, h_6, h_8)$. The selected sequences provide sufficient features, also called as behavioral features, that represent the three known behavior models.

We shall note that selecting the matrix-based behavior sequences is different from the research on compressing model or behavioral space [20, 7]. The previous compression method focuses on reducing the entire behavior space while maintaining a complete policy tree. Our work is to select a set of representative behavior sequences that are often a partial policy tree. The selected sequences provide important behavioral features in the expansion of new behaviors for other agents.

3.2 | Measuring Behavior Diversity

Given the representative behavior sequences F_j, we aim to generate new policy trees that contain the sequences corresponding to a set of branches, and select the top-K policy trees by measuring their diversity. Algorithm 1 presents the procedure of top-K policy tree generation.

Algorithm 1 Generating Top-K Policy Trees $\mathcal{H}_K$

1: **function** TopK(F_j, $\mathcal{H} = \{H_1, \cdots, H_M\}$ and DID of Agent j)
2:     Get a set of behavior sequences F_j from $\mathcal{H}$
3:     Convert agent j's DID into dynamic Bayesian networks B_j
4:     Instantiate B_j with F_j
5:     Initiate $\mathcal{H}_K = \{H_1, \cdots, H_M\}$
6:     **repeat**
7:         Sample a full policy tree H_T from the B_j
8:         $\mathcal{H}_K = \mathcal{H}_K \cup H_T$
9:         Recompute the set diversity $Div(\mathcal{H}_K)$
10:    **until** $Div(\mathcal{H}_K)$ does not change
11:    **return** $\mathcal{H}_K$

To sample a full policy tree, we first convert agent j's dynamic influence diagram (DID) into its counterpart of Bayesian networks B_j and instantiate the networks using the known behavior sequences F_j (line 3-4). We convert decision nodes A into chance nodes where we can instantiate their states with the actions a from the behavior sequences F_j. The utility nodes are converted into chance nodes where the utility values are normalized into probabilities in the nodes. For the chance nodes O, we instantiate their states with the observations o from F_j.

We randomise a probability distribution in the initial beliefs S^1 in B_j and calculate the probability distributions for the chance nodes O^t and A^t at each time step. Given the probability $Pr(O^t|S^1, F_j)$, we sample a possible observation o_t to be added into a new tree H_T. To decide the action a_t given o_t, we choose the best one that results in the largest utility value $R(S^t, A^t)$. By sampling the actions and observations over the planning horizon T, we can compose a set of behavior sequences therefore composing a new policy tree H_T (line 7). In other words, we complete a full policy tree by filling in the rest of its branches based on the known sequences. We repeat the adding of new policy trees until the diversity of the new set of K policy trees does not increase (line 8-9). We may terminate the process once a big

K is reached. This is to ensure that the I-DID models with the top-K behaviors can be solved within a computational limit.

What remains is to compute the diversity of the behavior set $Div(\mathcal{H}_K)$. We need to measure how different the K behavior trees would be in terms of their actions given specific observations over T time steps. We propose two diversity measurements for this purpose. The first measurement of diversity considers difference among behavior sequences in a vertical manner: the sequences or paths (with one specific observation at one time step) in the tree are examined separately along the depth. It, called as MDP (measurement of diversity over paths), mainly measures the diversity of sequences in the tree. The second diversity measurement extends MDP with the extra consideration of behaviors in a horizontal way: the sequences (with all possible observations at one time step) are compared along the width. It, also named as MDF (measurement of diversity with frames), measures the frames of a policy tree on the top of MDP.

In the first measurement, we retrieve all the different behavior sequences from the policy trees $\mathcal{H}_K$. For every sequence h_T, we aggregate all sub-sequences h_t with the length t ($\in [1, T]$). Since early actions of agent j have immediate impacts on agents' interactions, the short sequences contribute more into the diversity. Hence, we weight the sequence h_t using the factor of $\frac{1}{|\Omega|^{t-1}}$. Formally, we define the MDP diversity of K policy trees in Eq. 1.

$$Div(\mathcal{H}_K)_{MDP} = \sum_{t=1}^{T} \frac{Diff(h_t)}{|\Omega_j|^{t-1}} \tag{1}$$

where $Diff(h_t)$ is the number of different sequences h_t in $\mathcal{H}_K$ and $|\Omega_j|$ is the number of agent j's observations.

The first measurement considers a single action at one time step (within individual sequences) and may lose a general picture of what agent j behaves given different observations at one time step. To capture the frame of general behaviors, we add the diversity of sub-trees (H_t as a part of each policy tree in $\mathcal{H}_K$) of different depths into MDP. This leads to the second diversity measurement MDF in Eq. 2.

$$Div(\mathcal{H}_K)_{MDF} = \sum_{t=1}^{T} \frac{Diff(h_t) + Diff(H_t)}{|\Omega_j|^{t-1}} \tag{2}$$

where $Diff(h_t)$ and $Diff(H_t)$ are the numbers of different sequences h_t and sub-trees (frames) H_t respectively in $\mathcal{H}_K$, and $|\Omega_j|$ is the number of agent j's observations.

In Fig. 6, we elaborate the two diversity measurements (MDP and MDF) in one specific example. Given four behavior trees, e.g. $\mathcal{H}=\{H_1, \cdots, H_4\}$, we could obtain 2 different sequences with the length of one time step, 5 different sequences with the length of two time steps and 12 different sequences with the length of three time steps, as shown in the left panel in Fig. 6. In contrast, there exist: 2 different frames with one time step, 3 different frames with two time steps and 4 different frames with three time steps in the right panel. Following the diversity calculation in Eqs. 1 and 2, we can get the diversity values for the four policy trees using the two measurements, $Div(\mathcal{H}_K)_{MDP}$=7.5 and $Div(\mathcal{H}_K)_{MDF}$=12.0, respectively.

As shown in Eqs. 1 and 2, MDF has a larger diversity value than MDP since it includes more factors in the computation and could differentiate more behaviors. We will investigate how this extra information will add value into decision quality for a subject agent in the experiments. In terms of time complexity, MDP needs to compare all possible behavior sequences (with length from 1 to T) for at most N potential trees (generated in the sampling process) resulting in $\frac{NT(1+T)}{2}$ comparison operations while MDF requires extra NT operations in the frame comparison. Since the com-
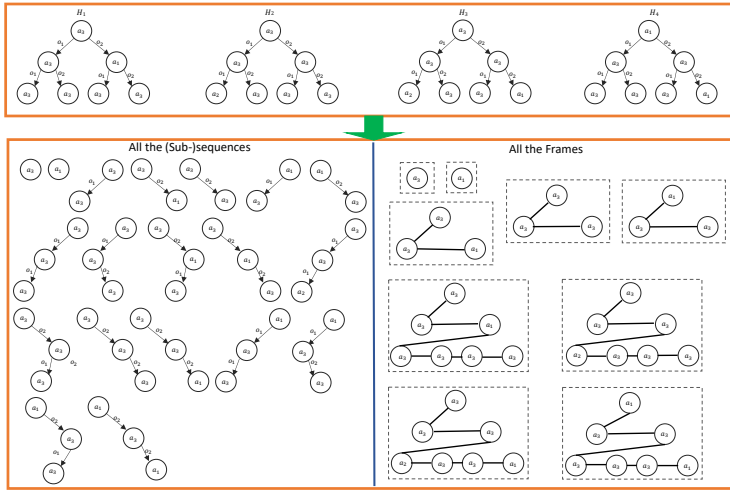
FIGURE 6 By aggregating all the (sub)-sequences and frames from the known policy trees $\mathcal{H}=\{H_1, \cdots, H_4\}$, the two diversity measurements (MDP and MDF) can be calculated.

parison is merely to check the actional equivalence, there is no significant difference between the time complexities of the two diversity measurements.

Once we have the diversity measurements, we calculate the set diversity ($Div(\mathcal{H}_K)$ in line 9-10) using either $Div(\mathcal{H}_K)_{MDP}$ or $Div(\mathcal{H}_K)_{MDF}$ in the the top-K behavior selection in Alg. 1. Notice that we decide whether a new policy tree is to be included in the top-K set while generating one from the sampling process (line 7). We do not generate all the policy trees and select K from them. As the number of possible new trees could be infinite, it is still an open issue about the boundary of agents' *sound* behaviors given prior knowledge.

4 | EXPERIMENTAL STUDY

We implemented the proposed approach (as described in Alg. 1) of generating new behaviors for other agents by first eliciting linearly independent behavior sequences and then selecting the top-K behaviors through the two behavior diversity measurements. We integrated this new approach into the I-DID solutions and investigated its performance in an empirical way. We used two well-known problem domains in our experiments. One is the multiagent tiger problem while the other is the multiagent unmanned aerial vehicle (UAV) problem [7]. All the implementations and tests were conducted in Window 10 with the setting of CPU (11th Gen Intel Core i7-6700 @ 3.40GHz 4-core) and 24GB RAM.

For each problem domain involving two agents, we built an I-DID model for the subject agent i who hypothesises a number of behavioral models of the other agent j in the model node M_j of Fig. 3. As the new approach provides a new way of solving I-DID models by providing more diverse behaviors to the other agent j, we compared three I-DID algorithms in the experiments. One is the state-of-art I-DID algorithm (IDID) that expands the model node only using the known models M. However, it assumes that the true behaviors of agent j are in the model node. The other two algorithms, namely IDID-MDP and IDID-MDF, use the diversity measurements of MDP and MDF to select top-K behaviors respectively for the other agent (in Alg. 1), and expand the model node in the I-DID models. The three

algorithms (IDID, IDID-MDP and IDID-MDF) adopt the same exact algorithm to solve the I-DID models [24] once the model nodes are expanded with different candidate models of the other agent j.

We evaluate the algorithms in terms of the average rewards that agent i receives when it interacts with agent j. We randomly select one behavioral model of agent j as its true model that is either from the M known models or a randomised model from K models (generated by either the IDID-MDP or IDID-MDF algorithm). Subsequently, agent i executes its I-DID optimal policies while agent j follows the selected behaviors in their interactions. For every interaction, agent i receives actual rewards given the outcomes of their actions for each time step and accumulates the rewards over the entire planning horizon T. We let both agents interact for 50 rounds and compute the average rewards for agent i accordingly. In addition, we investigate the impact of the diversity values for the two diversity measurements in the experiments.

4.1 | Multiagent Tiger Problems
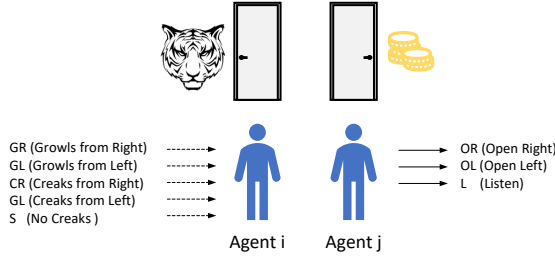
FIGURE 7 A two-agent tiger problem where agent i models agent j's behaviors in order to optimise its own decisions over times. The problem specification follows: $|S|=2, |A_i|=|A_j|=3, |\Omega_i|=6$, and $|\Omega_j|=2$.

A multi-agent tiger problem is well studied in multi-agent planning research and has become a benchmark for evaluating agent planning models [14, 19]. We consider the two-agent version of this problem in Fig. 7. Both agent i and agent j need to decide either Open the Right/Left-hand side of door (OR or OL) or Listen (L) when they are uncertain of a tiger's location (behind a door). If both open the door behind which a pot of gold exists, they share the gold; otherwise, they will be eaten by the tiger if only one of them faces the tiger. Their decisions are based on what they can observe, e.g. tiger's growls or creaks from either door. From the viewpoint of agent i, it needs to predict what agent j does simultaneously in order to optimise its own decisions. We built an I-DID model for agent i and varied the agent j's model space M_j in the I-DID model.
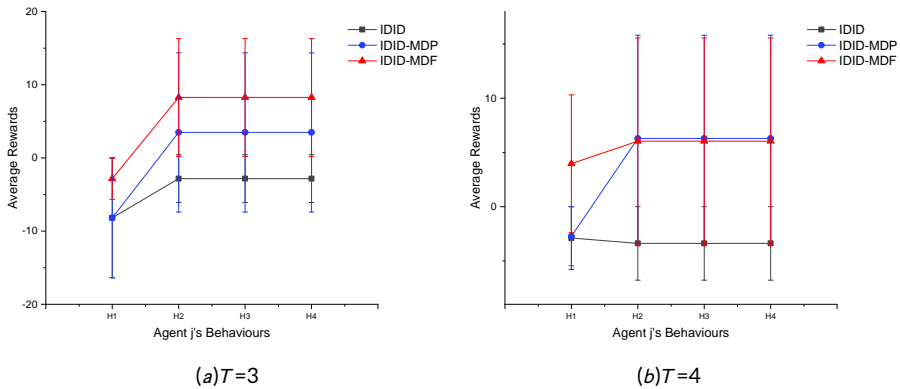
Figure 8 shows the average rewards received by agent i when it runs the I-DID models with different planning horizons ($T=3$ and 4). For both the models, we have six initial models ($M=6$) for agent j. However, the model with $T=4$ selects four new models ($K=4$) when the model with $T=3$ adds only three new models ($K=3$) using both MDF and MDP measurements in the top-K model selection. The selection is reasonable since the model with a large planning horizon often has more different behaviors. In almost all the cases, the IDID-MDF algorithm achieves better performance (when agent i receives larger rewards) than the other two algorithms. In some cases, e.g. H_6 in Fig. 8(a) and H_5 and H_6 in Fig. 8(b), IDID-MDP does not perform well, which is probably due to the fact that local behaviors in those particular models are not caught by the MDP measurement. Consequently, agent i can not react well when agent j executes those actions. In addition, we observe that IDID-MDF has better reliability than IDID-MDP in terms of small variances (the vertical candle bars). This is partially attributed to the merit that the MDF measurement considers

a general behavioral pattern when selecting the models.
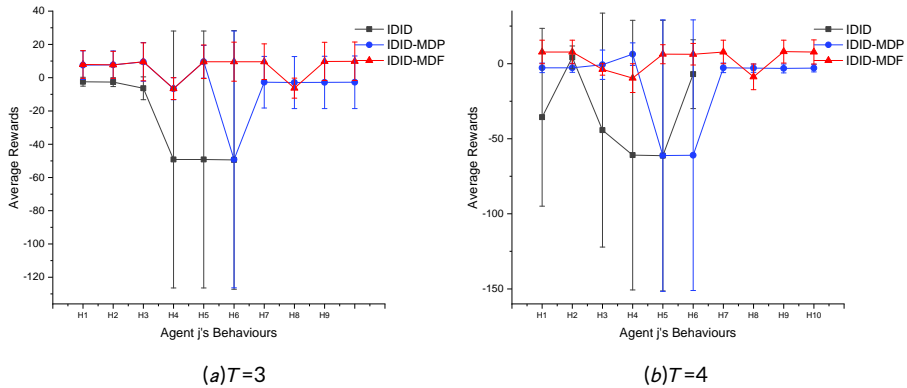
FIGURE 8 Average rewards received by the subject agent i when it uses j's behaviors models in the I-DID models of the planning horizons (*a*) T=3 and (*b*) T=4.

Subsequently, we ran the I-DID models with K=4 given four initial (M=4) models of agents. Hence, the model size in the models are equal to eight. In the first set of experiment, we had the true model j selected from the eight models (as the experiments in Fig. 8). Fig. 9 repeats the similar performance pattern for four models randomly selected in the experiments. Both IDID-MDF and IDID-MDP methods perform consistently well in all the cases except the case of H_1 picked from the model set. We investigated the behaviors in the model and found that the behaviors are rather monotonic therefore being mis-representative by the selection based on the MDP measurement.

FIGURE 9 Average rewards received by the subject agent i when the true model of agent j is randomly selected from one of the candidate models within the mode node in the I-DID models with (*a*) T=3 and (*b*) T=4.

In contrast, in the second set of experiments, we randomly generate a true model for agent j (based on the initial models). The model is not identical to any model in the model node expanded by the three algorithms. Hence agent i interacts with agent j whose behaviors are not completely in agent i's model space. In the IDID implementation, we let agent i act randomly if it can not recognize agent j's actions based on its I-DID model. This is to test the model capability in dealing with unknown behaviors of agent j. Fig. 10 demonstrates that the IDID-MDF algorithm

performs better than both IDID and IDID-MDP. This is benefited from the fact that IDID-MDF includes more diverse behaviors so that it can capture some random models (could potentially be true models) in a good manner. In the particular model H_3 in Fig. 10(b), all the methods exhibit a dip in the rewards since the behaviors in the model are rather random and the methods have not contained all the sequences in the model. The IDID-MDP and IDID methods have similar performance since both of them can not find the agent j's actions in the selected models, which leads to poor reactions resulting in low rewards in the simulations.

$(a)T=3$ $(b)T=4$
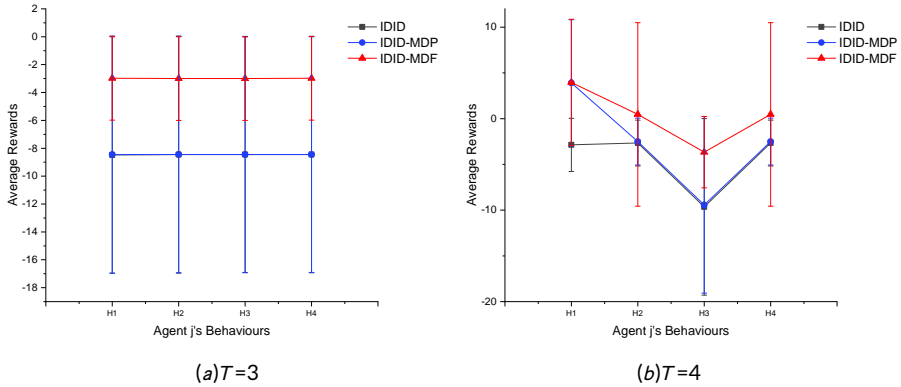
FIGURE 10 Average rewards received by the subject agent i when the true model of agent j is randomly generated (not being identical to any model generated by IDID-MDP or IDID-MDF) in the I-DID models (a) $T=3$ and (b) $T=4$.

Finally, we empirically investigate the relations between the model diversity and the average rewards. For each I-DID model with the same planning horizon (either $T=3$ or $T=4$), we run various settings of M and K, and calculate the diversity of the resulting candidate models in the I-DID models as well as the average rewards received by agent i in the interactions. Figure 11 shows positive correlations between the diversity values and the average rewards. The MDF values are generally larger than the MDP values as indicated in Eq. 1 and 2. Agent i obtains better rewards when the I-DID models have larger diversity values using both the MDP and MDF measurements. This verifies our motivation of improving agents' decision quality through diversifying model selection.


(a) MDP Performance   (b) MDF Performance
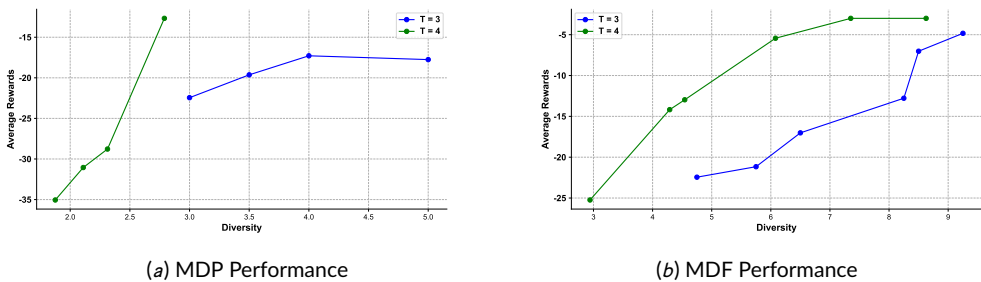
FIGURE 11 Relations between the diversity values [of (a) MDP and (b) MDF] and the average rewards received by agent i using the I-DID models with $T=3$ and $T=4$.
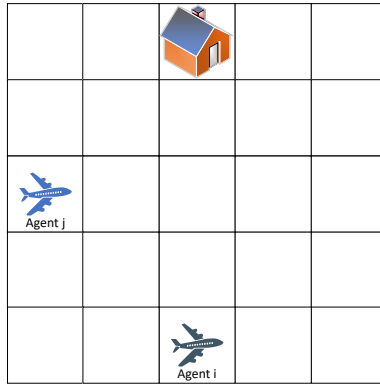
FIGURE 12 A two-agent UAV problem where agent *i* intends to capture agent *j* before agent *j* reaches the safe house. The problem has the specification of $|S_i|$=81, $|S_j|$=25, $|A_i|$=$|A_j|$=5, and $|\Omega_i|$=$|\Omega_j|$=4.

4.2 | Multiagent UAV Problems

The multi-agent UAV problem is the largest problem domain in testing I-DID algorithms [7, 19]. As shown in Fig. 12, both UAVs have the options of either moving in four directions or staying at their original positions. They do not know exact positions of their own and others, but can receive the signals of relative positions from each other. Since the two UAVs act simultaneously, one UAV needs to have a good estimation of what the other behaves so as to achieve its own goal. In our experiments, we let agent *i* act as a chaser UAV who is planning to intercept a fugitive UAV *j* on its way to the safe house. The chaser agent *i* gets rewarded once it captures the fugitive agent *j* before the agent *j* reaches the safe house. We build the I-DID models for the chaser agent *i* who models the fugitive agent *j*'s behaviors through the IDID, IDID-MDP and IDID-MDF algorithms.
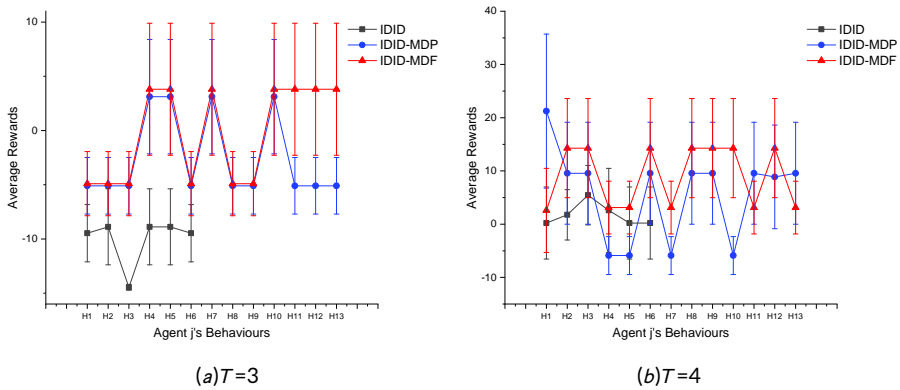
FIGURE 13 Average rewards received by the chaser agent *i* when it plans to capture the fugitive agent *j* through solving the I-DID models of (*a*) *T*=3 and (*b*) *T*=4.

In Fig. 13, we report the average rewards of agent *i* in the setting of *M*=6 for both the I-DID models of *T*=3 and *T*=4. The IDID-MDF algorithm exhibits better performance, compared to the IDID and IDID-MDP algorithms, similarly as shown in the multiagent problem domain. In most of the cases, both the IDID-MDP and IDID-MDF algorithms

outperform the IDID algorithm particularly in the I-DID models with $T=3$ and $T=4$. Surprisingly, we notice that the IDID algorithm does not perform good particularly in the I-DID models of $T=3$. For the short planning horizon ($T=3$), the chaser agent i does not have enough times to gather sufficient information as so to reduce the uncertainty of agent j's behaviors, therefore failing in capturing the fugitive agent j in most the cases. With extra models introduced by IDID-MDP and IDID-MDF, the chaser agent i gains more knowledge about the fugitive agent j's behaviors and achieves better rewards. The IDID-MDF algorithm performs slightly better than the IDID-MDP algorithm in the I-DID models of $T=4$. Capturing general patterns (in IDID-MDF) provides more informative behaviors when the fugitive agent j plans a long way to the safe house. In the large planning horizon ($T=4$), we notice that IDID-MDP does not perform well particularly for the agent j's models (H_4-H_{10}). The agent j's behaviors exhibited by those models actually lead to very straight routes to the safe house. The behaviors are not well represented by the models selected by IDID-MDP since MDP prefers to select models that have different routes and loses a general navigation tendency in the battlefield.

$(a)T=3$ $(b)T=4$
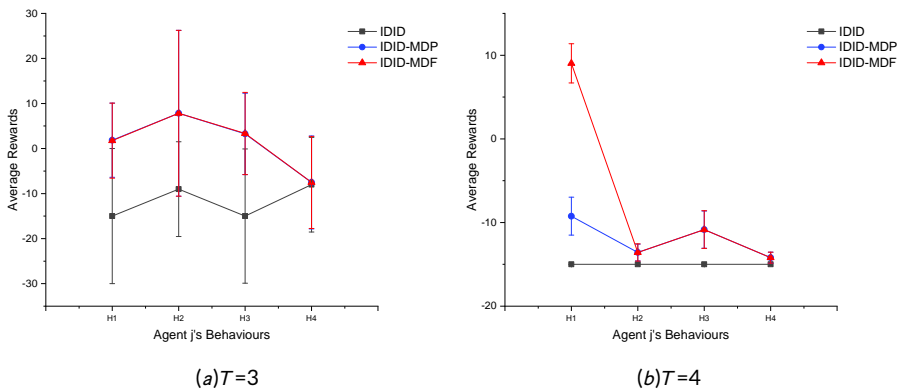
FIGURE 14 Average rewards received by the chaser agent i when it knows the true model of the fugitive agent j within the candidate model set in the I-DID models with (a) $T=3$ and (b) $T=4$.
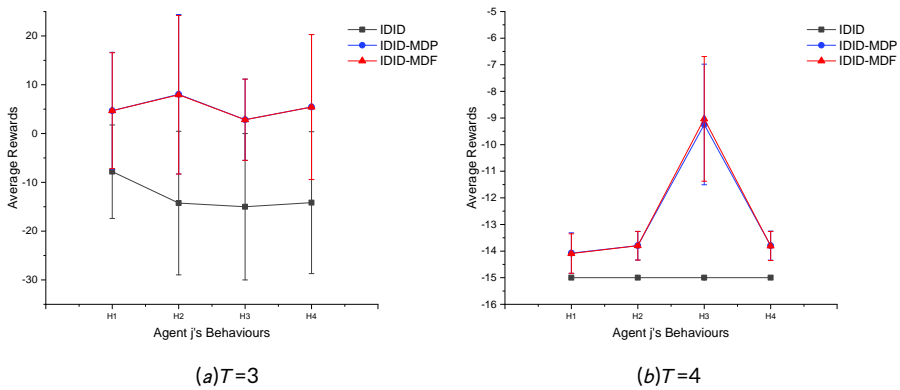

$(a)T=3$ $(b)T=4$

FIGURE 15 Average rewards received by the chaser agent i when it faces random behaviors of the fugitive agent j in the I-DID models (a) $T=3$ and (b) $T=4$.

We proceed to examine the performance of the IDID, IDID-MDP and IDID-MDF algorithms when (*a*) the chaser agent *i* knows the true model of the fugitive agent *j* in Fig. 14 and (*b*) the chaser agent *i* interacts with the fugitive agent *j* whose behaviors are not in the chaser's mind in Fig. 15. Both the IDID-MDP and IDID-MDF algorithms perform better than IDID, which is consistently with what we observed in the multiagent tiger problem domain. In almost all the cases, IDID does not perform well particularly when the agent *j*'s true models are randomly selected. This is expected since the I-DID algorithm does not consider unknown behaviors in the modelling process. Thus, the chaser agent *i* can only act randomly when it observes the unexpected behaviors of the fugitive agent *j*. The IDID-MDF algorithm performs similarly to the IDID-MDP algorithm in this problem domain. Since the fugitive agent *j* has many paths leading to the safe house, the limited number of models do not provide general patterns, which compromises the benefits of IDID-MDF. When we increase the number of models, as shown in Fig. 13, the IDID-MDF algorithm does show better performance. Hence, the IDID-MDF algorithm is suggested to be well used in complex behaviors of other agents.

Additionally, we show the increasing rewards for the chaser agent *i* when it holds a more diverse set of candidate models for the fugitive agent *j* in Fig. 16. We observe that MDP has a small dip of the average rewards when the diversity is increased. In contrast, the rewards increase monotonically with the increasing values of the MDF diversity. We notice that the factor of capturing general behavior patterns does not contribute much into the MDF measurement, as shown the diversity values in the *x*-axis of the figure. This indicates that both the IDID-MDP and IDID-MDF algorithms may select the same models for the fugitive agent *j*. Thus, the two algorithms provide similar performance in the aforementioned experiments.

(*a*) MDP Performance   (*b*) MDF Performance
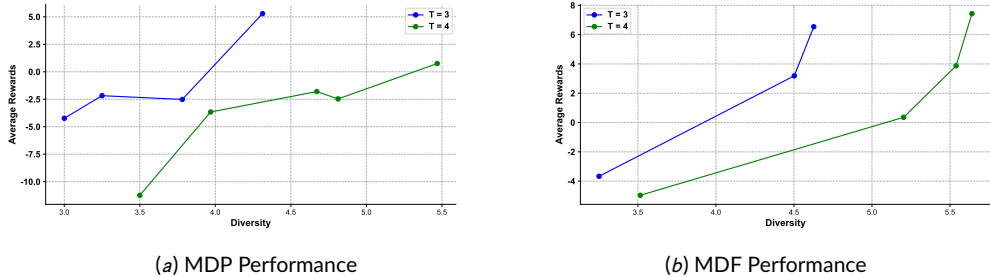
FIGURE 16   The chaser agent *i* gets better rewards when the diversity of the fugitive agent *j*'s models (measured by (*a*) MDP and (*b*) MDF) is increased in the I-DID models with *T*=3 and *T*=4.

In summary, both the IDID-MDF and IDID-MDP algorithms perform as we expect in the experiments, and the IDID-MDF algorithm would be a better choice when more complex behaviors are involved in a problem domain. Furthermore, we show the comparative efficiency for the IDID-MDP and IDID-MDF algorithms in Table 1. For every set of initial models (*M*) of agent *j*, we compare their running times including generating new models and then selecting the top-*K* models from the models in the Alg. 1. As expected, the larger planning horizon (*T*=4) requires more complex I-DID models therefore consuming more times in sampling and comparing the behaviors. Since the UAV problem domain has much larger state, action and observation space, the corresponding I-DID models and agents' behaviors become more complex. Consequently, both the MDP and MDF methods demand more computational times. Because MDF considers the extra factor of behavioral patterns in selecting the models, it can differentiate more behaviors therefore resulting in more new models in the model generation. Apparently, MDF needs more times in comparing more models. However, its efficiency is not significantly compromised as shown in the table - not more than double

amount of times as spent by MDP in most the experiments.

TABLE 1 Running times spent by the MDP and MDF methods on generating and selecting the top-K models in Alg. 1.

Times (s)	$T=3$				$T=4$			
	Tiger		UAV		Tiger		UAV	
M	MDP	MDF	MDP	MDF	MDP	MDF	MDP	MDF
3	0.35	0.97	2.37	6.86	0.41	1.80	2.75	6.96
4	1.66	1.65	2.94	9.26	1.79	2.25	4.02	9.52
5	1.98	4.49	10.43	10.54	2.14	11.29	24.20	10.42
6	41.56	76.32	17.14	10.46	45.16	100.74	18.92	22.30
7	328.43	442.94	42.76	81.96	330.31	573.85	46.46	97.40

5 | CONCLUSION

Exploring unknown behaviors of other agents has become an important research issue in the research of multiagent systems. In particular, in partially observable stochastic game (POSG), a subject agent needs to predict actions of other agents in order to optimise its own decisions. In this article, we investigate a novel way of generating new behaviors based on known behaviors of other agents in a general multiagent decision model - interactive dynamic influence diagrams (I-DIDs). This work is the first attempt on studying the behavior diversity in I-DID solutions, which is significantly different from the previous I-DID research on compressing other agents' models in I-DIDs. Given known behaviors of other agents, we use a linear reduction technique to generate a set of representative behavior sequences. The resulting sequences are linearly independent and provide basic knowledge to generating new behaviors for other agents. Subsequently, we sample a new set of behaviors, represented by policy trees, through applying conventional inference techniques in the I-DID model. We propose two diversity measurements to select top-K policy trees that maximise the diversity of the selected behaviors. We conduct experiments to investigate the performance of the new I-DID solutions on dealing with unknown behaviors of other agents in two problem domains. Although the new method of top-K behavior selection is implemented in I-DIDs, it can also be generalized in other models, like I-POMDPs, by adapting the sampling method to generate a set of new behaviors. This work also implies that diversifying behaviors can benefit the exploration of unknown behaviors in agents' interactions and contribute to the research on unknown unknowns [33].

As observed in the experimental study, it seems that the two diversity measurements are not one-size-fits-all. In addition, sampling new behaviors from decision models requires sophisticated techniques to ensure their soundness. However, the study lights up critical thoughts about behavior diversity and its impact on a subject agent's decision quality. Thoughts are also captured by the recent I-DID study although the work still focuses on known behaviors of other agents [29]. Hence, this work will open a new field on developing I-DID solutions based on diversifying other agents' behaviors. A couple of new research lines would be conducted in the future. For example, a new method of

diversifying behaviors with the consideration of a subject agent's decision quality would be immediate improvement of the current measurements. Of course, the challenge lies in measuring the decision quality since the subject agent has not yet completed the modelling process at this stage. Another way of developing new diversity measurements could be conducted in an online manner when the subject agent receives more knowledge about true behaviors of other agents in real-time interactions. The diversity needs to be adjusted in a dynamic way when a specific type of behavior is discovered in the interactions. This work can also contribute to the recent development of solving unknown weights in group decision making problems [34, 35]. It will benefit the recent research on intelligent systems when the systems are to be deployed in an open world and can not fully model a new environment prior to their applications.

Acknowledgements

references

[1] Andersen PA, Goodwin M, Granmo OC. Towards safe reinforcement-learning in industrial grid-warehousing. Information Sciences 2020;537:467 – 484.

[2] Albrecht SV, Stone P, Wellman MP. Special issue on autonomous agents modelling other agents: Guest editorial. Artificial Intelligence 2020;285:103292.

[3] Brown GW. Iterative Solution of Games by Fictitious Play. New York, USA: Wiley; 1951.

[4] Alessandro P, Piotr G. Interactive POMDPs with Finite-state Models of Other Agents. Autonomous Agents and Multi-Agent Systems 2017 Jul;31(4):861–904.

[5] Doshi P, Zeng Y, Chen Q. Graphical models for online solutions to interactive POMDPs. In: Durfee EH, Yokoo M, Huhns MN, Shehory O, editors. Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS);. p. 217–224.

[6] Doshi P, Zeng Y, Chen Q. Graphical Models for Interactive POMDPs: Representations and Solutions. Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS) 2009;18(3):376–416.

[7] Zeng Y, Doshi P. Exploiting Model Equivalences for Solving Interactive Dynamic Influence Diagrams. J Artif Intell Res 2012;43:211–255.

[8] Gmytrasiewicz P, Doshi P. A Framework for Sequential Planning in Multiagent Settings. Journal of Artificial Intelligence Research (JAIR) 2005;24:49–79.

[9] Bolander T, Andersen MB. Epistemic planning for single- and multi-agent systems. Journal of Applied Non-Classical Logics 2011;21(1):9–33.

[10] Ma Y, Shen M, Zhao Y, Li Z, Tong X, Zhang Q, et al. Opponent portrait for multiagent reinforcement learning in competitive environment. International Journal of Intelligent Systems 2021;36:7461-7474.

[11] Suryadi D, Gmytrasiewicz PJ. Learning models of other agents using influence diagrams. In: International Conference on User Modeling Springer; 1999. p. 223–232.

[12] Koller D, Milch B. Multi-agent influence diagrams for representing and solving games. Games and Economic Behavior 2003;45(1):181–221.

[13] Gal Y, Pfeffer A. Networks of Influence Diagrams: A Formalism for Representing Agents' Beliefs and Decision-Making Processes. Journal of Artificial Intelligence Research 2008 sep;33(1):109–147.

[14] Bernstein D, Givan R, Immerman N, Zilberstein S. The Complexity of Decentralized Control of Markov Decision Processes. Mathematics of Operations Research 2002 12;27:637–842.

[15] Poh KL, Fehling MR, Horvitz EJ. Dynamic Construction and Refinement of Utility-Based Categorization Models. IEEE Trans Syst Man Cybern Syst 1994;24(11):1653–1663.

[16] Albrecht SV, Stone P. Autonomous agents modelling other agents: A comprehensive survey and open problems. Artif Intell 2018;258:66–95.

[17] Abhijit S, Debjit D, Samarjit K. Some new hybrid hesitant fuzzy weighted aggregation operators based on Archimedean and Dombi operations for multi-attribute decision making. Neural Computing and Applications 2021;33(14):8753–8776.

[18] Rani P, Mishra AR, Saha A, Hezam IM, Pamucar D. Fermatean fuzzy Heronian mean operators and MEREC-based additive ratio assessment method: An application to food waste treatment technology selection. International Journal of Intelligent Systems 2022;37(3):2612–2647.

[19] Doshi P, Gmytrasiewicz PJ, Durfee EH. Recursively modeling other agents for decision making: A research perspective. Artif Intell 2020;279.

[20] Pynadath DV, Marsella S. Minimal Mental Models. In: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence; 2007. p. 1038–1044.

[21] Rathnasabapathy B, Doshi P, Gmytrasiewicz PJ. Exact solutions of interactive POMDPs using behavioral equivalence. In: The Fifth International Joint Conference on Autonomous Agents and Multiagent Systems ACM; 2006. p. 1025–1032.

[22] Zeng Y, Doshi P, Chen Y, Pan Y, Mao H, Chandrasekaran M. Approximating behavioral equivalence for scaling solutions of I-DIDs. Knowledge Information Systems 2016;49(2):511–552.

[23] Conroy R, Zeng Y, Cavazza M, Tang J, Pan Y. A Value Equivalence Approach for Solving Interactive Dynamic Influence Diagrams. In: Proceedings of the 15th International Conference on Autonomous Agents& Multiagent Systems (AAMAS); 2016. p. 1162–1170.

[24] Doshi P, Zeng Y, Chen Q. Graphical models for interactive POMDPs: representations and solutions. Auton Agents Multi Agent Syst 2009;18(3):376–416.

[25] Tatman JA, Shachter RD. Dynamic programming and influence diagrams. IEEE Transactions on Systems, Man, and Cybernetics 1990;20(2):365–379.

[26] Søndberg-Jeppesen N, Jensen FV, Zeng Y. Opponent modeling in a PGM framework. In: Proceedings of International conference on Autonomous Agents and Multi-Agent Systems (AAMAS); 2013. p. 1149–1150.

[27] Pan Y, Tang J, Ma B, Zeng Y, Ming Z. Toward data-driven solutions to interactive dynamic influence diagrams. Knowl Inf Syst 2021;63(9):2431–2453.

[28] Conroy R, Zeng Y, Tang J. Approximating Value Equivalence in Interactive Dynamic Influence Diagrams Using Behavioral Coverage. In: Kambhampati S, editor. Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI); 2016. p. 201–207.

[29] Pan Y, Ma B, Tang J, Zeng Y. Behavioral model summarisation for other agents under uncertainty. Information Sciences 2022;582:495–508.

[30] Jensen FV, Nielsen TD. Bayesian Networks and Decision Graphs. 2nd ed. Springer; 2007.

[31] Zeng Y, Ran Q, Ma B, Pan Y. Modelling other agents through evolutionary behaviours. Memetic Computing 2022;14(1):19–30.

[32] Cunningham JP, Ghahramani Z. Linear Dimensionality Reduction: Survey, Insights, and Generalizations. Journal of Machine Learning Research 2015;16(89):2859–2900.

[33] Lakkaraju H, Kamar E, Caruana R, Horvitz E. Identifying unknown unknowns in the open world: Representations and policies for guided exploration. In: AAAI, vol. 31; 2017. .

[34] Abhijit S, Tapan S, Ronald Y. Hybridizations of generalized Dombi operators and Bonferroni mean operators under dual probabilistic linguistic environment for group decision-making. International Journal of Intelligent Systems 2021;36(11):6645–6679.

[35] Saha A, Garg H, Dutta D. Probabilistic linguistic q-rung orthopair fuzzy Generalized Dombi and Bonferroni mean operators for group decision-making with unknown weights of experts. International Journal of Intelligent Systems 2021;36(12):7770–7804.