A sufficient statistic for influence in structured multiagent environments

Oliehoek, Frans A.; Witwicki, Stefan; Kaelbling, Leslie P.

DOI
10.1613/JAIR.1.12136

Publication date
2021

Document Version
Final published version

Published in
Journal of Artificial Intelligence Research

Journal of Artificial Intelligence Research 70 (2021) 789-870 Submitted 05/2020; published 02/2021

A Sufficient Statistic for Influence in Structured Multiagent Environments

Frans A. Oliehoek F.A.OLIEHOEK@TUDELFT.NL
Department of Intelligent Systems
Delft University of Technology
Mourik Broekmanweg 6
Delft, 2628 XE, The Netherlands

Stefan Witwicki STEFAN.WITWICKI@NISSAN-USA.COM
Alliance Innovation Lab Silicon Valley
Nissan Technical Center North America
3400 Central Expressway
Santa Clara, CA 95051, USA

Leslie P. Kaelbling LPK@CSAIL.MIT.EDU
CSAIL
Massachusetts Institute of Technology
32 Vassar Street
Cambridge, MA 02139, USA

Abstract

Making decisions in complex environments is a key challenge in artificial intelligence (AI). Situations involving multiple decision makers are particularly complex, leading to computational intractability of principled solution methods. A body of work in AI has tried to mitigate this problem by trying to distill interaction to its essence: how does the policy of one agent *influence* another agent? If we can find more compact representations of such influence, this can help us deal with the complexity, for instance by searching the space of influences rather than the space of policies. However, so far these notions of influence have been restricted in their applicability to special cases of interaction. In this paper we formalize *influence-based abstraction (IBA)*, which facilitates the elimination of latent state factors *without any loss in value*, for a very general class of problems described as factored partially observable stochastic games (fPOSGs). On the one hand, this generalizes existing descriptions of influence, and thus can serve as the foundation for improvements in scalability and other insights in decision making in complex multiagent settings. On the other hand, since the presence of other agents can be seen as a generalization of single agent settings, our formulation of IBA also provides a sufficient statistic for decision making under abstraction for a single agent. We also give a detailed discussion of the relations to such previous works, identifying new insights and interpretations of these approaches. In these ways, this paper deepens our understanding of abstraction in a wide range of sequential decision making settings, providing the basis for new approaches and algorithms for a large class of problems.

1. Introduction

One of the important ideas in the development of algorithms for multiagent systems (MASs) is the identification of compressed representations of the information that is relevant for an

agent (Becker, Zilberstein, Lesser, & Goldman, 2003; Becker, Zilberstein, & Lesser, 2004; Varakantham, young Kwak, Taylor, Marecki, Scerri, & Tambe, 2009; Petrik & Zilberstein, 2009; Witwicki & Durfee, 2010a, 2010b, 2011; Velagapudi, Varakantham, Scerri, & Sycara, 2011; Witwicki, 2011; Witwicki, Oliehoek, & Kaelbling, 2012; Oliehoek, Witwicki, & Kaelbling, 2012; Hernandez-Leal, Kaisers, Baarslag, & Munoz de Cote, 2017; Bazinin & Shani, 2018). For instance, when a cook and a waiter collaborate, the waiter might not need to know all details of how the cook prepares the food; it may be sufficient if he/she has an understanding of the time that it will take.

In this paper we investigate abstractions that aim at decomposing structured MASs into a set of smaller interacting problems (Oliehoek et al., 2012; Witwicki & Durfee, 2010b). In particular, we describe in detail the concept of *influence-based abstraction (IBA)*, which facilitates the abstraction of latent state variables without sacrificing task performance. It constructs a smaller, local model for one of the agents given the policies of the other agents. IBA consists of two steps: first, we compute a so-called *influence point*—a more abstract representation of how an agent's local problem is affected by other agents and external (i.e., non-local) parts of the problem—, second, this influence is used to construct the smaller *influence-augmented local model (IALM).* This IALM can subsequently be used to compute a best response.

IBA does not only give a new perspective on best-response computations themselves, but this new perspective also has broader implications. For instance, it forms the basis of *influence search* (Becker et al., 2003; Witwicki & Durfee, 2010b; Witwicki et al., 2012; Bazinin & Shani, 2018), which can provide significant speedup for multiagent planning by searching the space of *joint influences* rather than the potentially much bigger space of joint policies. It also can underpin guarantees on the quality of heuristic solutions, by considering *optimistic* influences (Oliehoek, Spaan, & Witwicki, 2015a), or approximate influences (Congeduti, Mey, & Oliehoek, 2020). While in this article, we assume that the model (which can be seen as a specific type of dynamic Bayesian network) is known in advance, future work could consider learning such representations. Moreover, IBA can serve as inspiration, in the context of deep reinforcement learning, for neural network architectures that compute approximate versions of influence, which can improve learning, both in terms of speed as well as performance (Suau de Castro, Congeduti, Starre, Czechowski, & Oliehoek, 2019b).

This article gives a formal definition of influence that can be used to perform IBA for general factored partially observable stochastic games (fPOSGs) (Hansen, Bernstein, & Zilberstein, 2004; Boutilier, Dean, & Hanks, 1999), and proves that an IALM constructed using this definition of influence in fact allows computation of an *exact* best-response. In other words, it shows that this description of influence is a *sufficient statistic* of the policy of the other agents: it is sufficient to predict observations and rewards and to thereby optimize value. This article extends our previous paper (Oliehoek et al., 2012) in the following ways:

1. it provides a complete proof of the claimed exactness of IBA;
2. it elaborates on a number of technical subtleties, such as dealing with multiple sources of influence, and specifying initial beliefs in the IALM;
3. it provides an extension of IBA and corresponding proofs to fPOSGs with intra-stage dependencies, which are critical for the expressiveness of the formalism (cf. Section 4.1.1);

4. it provides additional illustration and explanation, making the concept of IBA more accessible;
5. it deepens the discussion of the relation to special cases of fPOSGs, and more explicitly identifies ways in which future work can improve scalability of these sub-classes;
6. it provides a much more extensive discussion of related work, including the more recent work on deep reinforcement learning (RL). Specifically, by building on the theoretical results provided in this paper, it generates insights into the nature of the 'approximate value factorization' assumption which has been successfully exploited by a popular class of deep RL methods.

Additionally, in Section 3 we make a simple (but, in the context of IBA, novel) observation: the presence of other agents can be seen as a generalization of single agent settings, which directly implies that *our formulation of IBA also provides a sufficient statistic for decision making under abstraction for a single agent.* While there is a multitude of performance loss bounds available for abstractions, e.g., see Dearden and Boutilier (1997), Dean, Givan, and Leach (1997), Givan, Dean, and Greig (2003), Iyengar (2005), Li, Walsh, and Littman (2006), Petrik and Subramanian (2014), Abel, Hershkowitz, and Littman (2016), these are usually based on assumed quality bounds on the transition probabilities and rewards of the abstracted model (see Section 8.3 for more details). In contrast, our work here shows how an abstracted model can preserve exact transition and reward predictions, by 'remembering' appropriate elements of the local history. In the words of McCallum (1995), we detail an approach to *perfectly* "uncover [...] hidden state" in abstractions for a large class of structured problems.

As such, the contributions of this paper are of a theoretical nature: they provide a principled understanding of lossless abstractions in structured (multiagent) decision problems by providing a formal framework that gives a unified perspective on previous work, while at the same time providing new insights and extending the scope of applicability. The main technical result is the proof of sufficiency given in Section 6: the smaller influence-augmented local model produced by IBA can be used instead of the original larger model *without any loss* in solution quality (i.e., value). The proof is not only a certification of the theory, it also serves a practical purpose: it isolates the core technical property that needs to hold for sufficiency, thus providing 1) insight into *how* abstraction of latent state factors affects value, 2) a derivation that can be used to obtain a simplification of influence in simpler cases, and 3) a recipe of how to prove similar results for more complex settings.

This paper is organized as follows: First, Section 2 provides the necessary background by introducing single and multiagent models for decision making. Section 3 introduces the concept of computing best responses (using global value functions) to the policies of other agents and the concept of 'local form models' which formalizes a desired abstraction for an agent. Next, in Section 4, we bring these concepts together: we show how an agent can locally compute a best-response (compute a local value function) provided it is given an influence point. Section 5 extends this framework to problems with intra-stage dependencies. Section 6 then presents the main proof of sufficiency of our influence points, i.e., it shows that they provide sufficient information to compute optimal policies without any loss in value. Section 7 discusses reinterpretations of previous work on forms of influence-

based abstraction in our more general framework, while Section 8 details the relations to other related work. Finally, Section 9 concludes.

2. Background

Here we concisely provide background on some of the models that we use. The main purpose is to introduce the notation formally. For an extensive introduction to *partially observable Markov decision processes (POMDPs)* we refer to Kaelbling, Littman, and Cassandra (1998) and Spaan (2012), for an introduction to multiagent variants see Seuken and Zilberstein (2008), Oliehoek (2012) and Oliehoek and Amato (2016).

Unavoidably, this manuscript contains a fair amount of terminology and mathematical notations. To aid the reader we have included a list of acronyms (Appendix B) and a list of recurring notation (Appendix C).

2.1 Single-Agent Models: POMDPs

Partially observable Markov decision processes, or POMDPs, provide a formal framework for the interaction of an agent with a stochastic, partially observable environment. That is, they provide an agent with the capabilities to reason about both action uncertainty as well as state uncertainty.

2.1.1 Model

A POMDP is a discrete time model, in which the agent selects an action at every time step or *stage.* It extends the regular *Markov decision process (MDP)* (Puterman, 1994) to settings in which the state of the environment cannot be observed. It can be formally defined as follows.

Definition 1 (POMDP)**.** A *partially observable Markov decision process (POMDP)* is defined as a tuple $\mathcal{M}^{POMDP} = \langle \mathcal{S}, \mathcal{A}, T, R, \mathcal{O}, O, H, b^0 \rangle$ with the following components:

- $\mathcal{S}$ is a (finite) set of states s. The state at some stage t is denoted s^t;
- $\mathcal{A}$ is the (finite) set of actions a;
- T is the transition probability function, that specifies $T(s'|s,a) = \Pr(s^{t+1} = s' \mid s^t = s,\, a^t = a)$, the probability of a next state s' given a current state s and action a. This directly demonstrates the primed shorthand notation we will occasionally use;
- R is the immediate reward function $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$. With $R(s,a,s')$ we denote the reward specified for a particular transition s,a,s';
- $\mathcal{O}$ is the set of observations;
- O is the observation probability function, which specifies $O(o|a,s') = \Pr(o^{t+1} = o \mid a^t = a, s^{t+1} = s')$, the probability of a particular observation o after a and resulting state s';
- H is the horizon of the problem as mentioned above;
- $b^0 \in \Delta(\mathcal{S})$, is the initial state distribution at time $t = 0$.[1]

1. $\Delta(\cdot)$ denotes the set of probability distributions over $(\cdot)$.

In many cases, the set of states is huge, and states can be thought of as composed of values assigned to different variables:

Definition 2 (Factored POMDP)**.** In a *factored POMDP (fPOMDP)*, the state space $\mathcal{S}$ is spanned by a set $\mathcal{F} = \left\{F^1, \ldots, F^{|\mathcal{F}|}\right\}$ of state variables F^k (that are also called *factors*). Each of these can take values from its domain $\mathcal{F}^k$, such that the set of states is defined as $\mathcal{S} = \mathcal{F}^1 \times \cdots \times \mathcal{F}^{|\mathcal{F}|}$.

The merit of such a factored POMDP is that, by making the structure of the problem (i.e., how different factors influence each other) explicit, the model can be much more compact. In particular, the initial state distribution can be compactly represented as a *Bayesian network* (Pearl, 1988; Bishop, 2006; Koller & Friedman, 2009), and the transition and reward model can be specified compactly using a *two-stage dynamic Bayesian network (2DBN)* (Boutilier et al., 1999), and a similar approach can be taken for the observation model (Poupart, 2005). (An example of a 2DBN will be discussed in Figure 2 on page 796.)

The fPOMDP model is closely related to the framework of *influence diagrams* (Howard & Matheson, 1984; Tatman, 1990). In fact, by unrolling (over time) the 2DBN we create an influence diagram. We point out, however, that our notion of *influence* (i.e., the influence point and the resulting influence-based abstraction we will detail in Section 4) is novel; it has not been considered in influence diagrams.

2.1.2 Beliefs

In contrast to regular MDPs, in a POMDP the agent cannot observe the state; it only observes the observations. However, the observations are not a Markovian signal: i.e., the last observation o^t made by the agent does not provide the same amount of information (to predict the rewards and the future of the process) as the *action-observation history (AOH)*, the entire history of actions and observations $\vec{h}^t = \left(a^0, o^1, \ldots, a^{t-1}, o^t\right)$. This means that in general the agent needs to select its actions based on $\vec{h}^t$ in order to achieve optimal performance.

Luckily, for a POMDP this history can be summarized compactly as a *belief,* which is defined as the posterior probability distribution over states given the history:

$$b(s) \triangleq \Pr(s|b^0, \vec{h}^t).$$

The belief does not only summarize the history, it does so in a lossless way. That is, a belief is a *sufficient statistic* for optimal decision making (Bertsekas, 2005); it allows an agent to reach the same performance as an agent that would act optimally based on the AOH $\vec{h}^t$.

This belief can be recursively computed, which means that an agent can update its belief as it interacts with its environment. We write $b' = BU(b,a,o)$, where $BU(b,a,o)$ is the belief update operator that, given a previous belief b taken action a and received observation o, produces the next belief:

$$\forall s' \qquad BU(b,a,o)(s') = \frac{1}{\Pr(o|b,a)} \Pr(o|a,s') \sum_s \Pr(s'|s,a) b(s). \tag{2.1}$$

Here, $\Pr(o|b,a)$ is a normalization constant:

$$\Pr(o|b,a) = \mathbf{E}_{s \sim b, s' \sim T(s,a,\cdot)} \left[O(a,s',o)\right] = \sum_{s'} \Pr(o|a,s') \sum_s \Pr(s'|s,a) b(s).$$

2.1.3 Policies and Value Functions

In a POMDP, the agent employs a *policy,* π, to interact with its environment. Such a policy is a (deterministic) mapping from beliefs to actions. Note that, given the initial belief b^0, such a policy will specify an action for each observation history.[2]

The goal of the decision maker, or agent, in the POMDP is to choose a policy π that maximizes the expected (discounted) cumulative reward:

$$\mathbf{E}\left[\sum_{t=0}^{H-1}\gamma^t R(s^t,a^t,s^{t+1})|b^0,\pi\right], \tag{2.2}$$

here

- H is the horizon, i.e., the number of time steps, or *stages,* for which we want to plan,
- the expectation is over sequences of states and observations induced by the policy π,
- $\gamma \in [0,1]$ is the discount factor.

In this work, we focus on the finite-horizon case, in which it is typical (but not necessary) to assume $\gamma = 1$.

For a finite-horizon POMDP, the optimal (action-)value function for stage t can be expressed as

$$Q^t(b,a) = \begin{cases} R(b,a), & t = H-1 \\ R(b,a) + \gamma \sum_o \Pr(o|b,a) V^{t+1}(BU(b,a,o)), & \text{otherwise} \end{cases} \tag{2.3}$$

where $V^{t+1}(b') = \max_{a'} Q^{t+1}(b',a')$ is the value of acting optimally in the next time step and $R(b,a)$ is the expected immediate reward:

$$R(b,a) = \mathbf{E}_{s\sim b, s'\sim T(\cdot|s,a)}\left[R(s,a,s')\right] = \sum_s b(s) \sum_{s'} \Pr(s'|s,a) R(s,a,s'). \tag{2.4}$$

2.2 Multiagent Models: POSGs

The POMDP model can be extended to include multiple self-interested agents as follows.

Definition 3 (POSG)**.** A *partially observable stochastic game (POSG)* is defined as a tuple $\mathcal{M}^{POSG} = \langle \mathcal{D},\mathcal{S},\mathcal{A},T,\mathcal{R},\mathcal{O},O,H,b^0 \rangle$ with the following components:

- $\mathcal{D} = \{1,\ldots,n\}$ is the set of n agents.
- $\mathcal{S}$ is a (finite) set of states.
- $\mathcal{A} = \mathcal{A}_1 \times \ldots \mathcal{A}_n$ is the set of *joint* actions $a = \langle a_1,\ldots,a_n \rangle$, with $\mathcal{A}_i$ the set of individual actions for agent i.
- T is the transition probability function, that now depends on joint actions: $T(s^{t+1}|s^t,a^t) = \Pr(s^{t+1}|s^t,a^t)$.

2. This can be seen as follows: for b^0 the policy specifies an action, a^0, then given o^1 we can compute b^1 which we can use to look up a^1, etc.

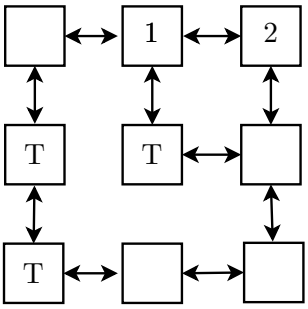


Figure 1: A possible instantiation of the House Search problem: 1, 2 represent the starting locations of the agents, while 'T' encodes the possible locations of the target.

- $\mathcal{R} = \langle R_1, \ldots R_n \rangle$ is the collection of immediate reward functions (one for each agent). Each $R_i : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ maps from states, joint actions and next states to an immediate reward for agent i.
- $\mathcal{O} = \mathcal{O}_1 \times \cdots \times \mathcal{O}_n$ is the set of joint observations $o = \langle o_1, \ldots, o_n \rangle$, with $\mathcal{O}_i$ the set of individual observations for agent i,
- O is the observation probability function, which specifies $\Pr(o|a,s')$, the probability of a particular joint observation o after a and resulting state s'.
- H is the horizon of the problem as mentioned above.
- $b^0 \in \Delta(\mathcal{S})$, is the initial state distribution at time $t = 0$.

Since in a POSG each agent has its own goal, there no longer is a definition of optimality. Instead it is customary to focus on game-theoretic solution concepts (Hansen et al., 2004). Such solutions, e.g., Nash equilibria, typically specify a tuple of policies $\pi = \langle \pi_1, \ldots, \pi_n \rangle$, one for each agent, that are in equilibrium. In general, we will refer to a tuple of policies π as a *joint policy.*

Of course, it is also possible to consider cooperative teams of agents. In this case, we align the goals of the agents by giving them the same reward function:

Definition 4 (Dec-POMDP)**.** A *decentralized partially observable Markov decision process (Dec-POMDP)* is a POSG where all agents share the same reward function: $\forall_{i,j}\ R_i = R_j$.

Since interests are aligned, in a Dec-POMDP we can speak about optimality. Moreover, there is guaranteed to be at least one *deterministic* joint policy that is optimal (Oliehoek, Spaan, & Vlassis, 2008a). As was the case for POMDPs, we can also consider variants of the multiagent models with factored state spaces. We will refer to these as *factored POSGs (fPOSGs)* and *factored Dec-POMDPs (fDec-POMDPs)* (Oliehoek, Spaan, Whiteson, & Vlassis, 2008b).[3]

As an example, we consider the House Search problem (Oliehoek, Witwicki, & Kaelbling, 2011), in which a team of robots must find a target (say a remote control) in a house

3. More recently, researchers have also investigated deterministic and non-deterministic versions, called (factored) qualitative Dec-POMDP (Brafman, Shani, & Zilberstein, 2013). We will not particularly target this special case in this paper, but note that ideas of influence search can be exploited in this context too (Bazinin & Shani, 2018).

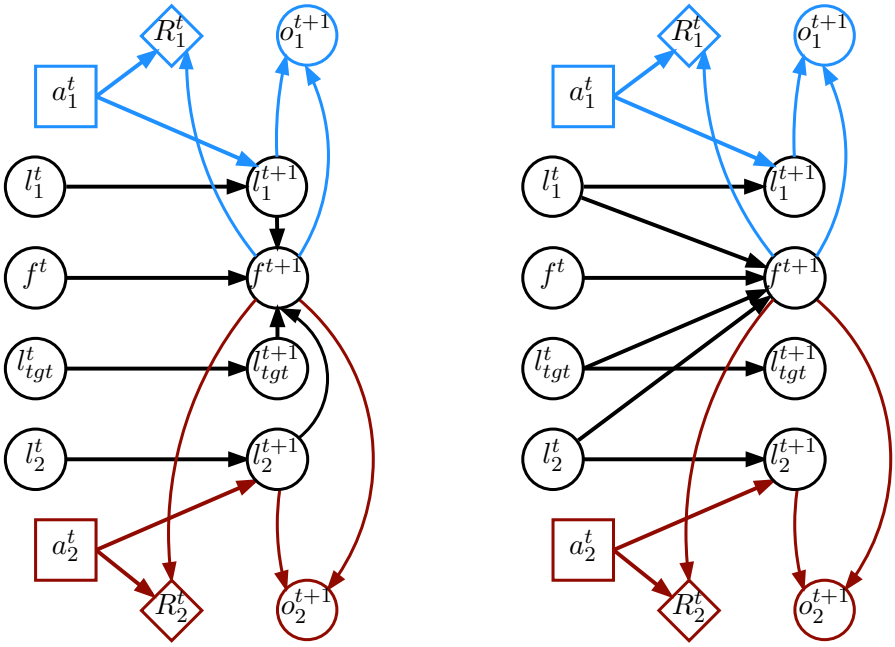
with multiple rooms. This task is representative of an important class of problems in which a team of agents needs to locate objects or targets. In House Search the assumption is that a prior probability distribution over the location of the target is available and that the target is stationary or moves in a manner that does not depend on the strategy used by the searching agents.

Example. The House Search environment can be represented by a graph, as illustrated in Figure 1 for the case of two agents. At every time step each agent can stay in the current room or move to a next one. The location of an agent i at time step t is denoted l_i^t and that of the target is denoted l_{tgt}^t. In general, the target could move with probabilities $p(l_{tgt}^{'}|l_{tgt})$. The actions (movements) of each agent have a specific cost $c_i(l_i,a_i)$ (e.g., the energy consumed by navigating to a next room) and can fail; we allow for stochastic transitions $p(l_i'|l_i,a_i)$. Also, each robot might receive a penalty c_{time} for every time step that the target is not found yet. When a robot is in (or near) the same node as the target, there is a probability of detecting the target $p(detect_i|l_{tgt},l_i)$, which will be modeled by a Boolean state variable 'target found' f^t, which both agents can observe (thus modeling a communication channel which the agents can only use to inform each other of detection). When the target is detected, the agents also receive a reward r_{detect}. Given the prior distribution and model of target behavior, the goal is to optimize the sum (over time) of rewards, thus trading off movement cost and probability of detecting the target as soon as possible. In this paper, we focus on the local perspective of a protagonist agent and therefore will assume that each agent has its individual rewards (so the POSG setting).[4]

Figure 2a demonstrates how a two-stage dynamic Bayesian network (2DBN) can be used to compactly represent the transition, observation, and reward model (Boutilier et al., 1999).[5] For instance, for each state variable at a state $t+1$, the 2DBN shows which other entities (state factors and actions) influence it. The figure illustrates that most dependencies are *across-stage* (e.g., l_2^t influences l_2^{t+1}) but that it is also possible to have *intra-stage dependencies (ISDs).* For instance, whether the target will be detected at stage $t+1$ depends on l_2^{t+1} not on l_2^t. The representation of the transition model is compact since it can be represented as a product of *conditional probability tables (CPTs)*, each of which are exponential only in the number of incoming dependencies. So as long as the number of incoming connections is limited, the transition probabilities can be represented compactly. Figure 2a also shows that this type of representation can also be employed for observation probabilities, as well as rewards.

Since ISDs complicate the notation and definition of influence, we also consider a version of the problem that has no intra-stage connections, shown in Figure 2b. For rewards and observations, intra-stage connections are still allowed. (In fact, since the observation probabilities in the standard POMDP definition depend on the next state s', there is no way of representing them without intra-stage connections). Note that this is a slightly different problem than the problem represented in Figure 2a: in the problem without ISDs the agents have a chance of detecting the target at stage $t+1$ if they are co-located with the target at

4. In previous work, the house search problem was treated as a Dec-POMDP by defining the team reward as the sum of the individual rewards (Oliehoek et al., 2011).
5. More formally, since we include actions (decisions) and rewards (utilities), diagrams like this are a type of influence diagram or decision network. However, not to introduce further terminology, we will refer to them simply as 2DBN.



(a) With intra-stage connections.

(b) Without intra-stage connections.

Figure 2: Factored representation of the House Search problem. Actions, observations and rewards of the first agent are in light blue, while those of agent 2 are in dark red. State variables are in black. We use standard shapes for influence diagrams: rectangles for actions, circles for random variables, and diamonds for rewards (e.g., Russell & Norvig, 2009) .

stage t, which means that there is a one-step delay incurred before they receive the reward. This illustrates the fact the ISDs do allow for a more expressive model, and that therefore developing theory that support such connections is an important goal.

To facilitate easier exposition, in Section 4 we will first introduce the concept of influence-based abstraction without ISDs. These will be considered in Section 5. Before we can jump to the topic of influence-based abstraction, however, we will need to discuss decision problems from a local perspective, in Section 3, which covers problems with ISDs.

3. Best Responses and Local-Form Models

In contrast to the typical solutions to POSGs and Dec-POMDPs, which try and identify a joint policy as the solution, this paper focuses on the local perspective of an individual agent. From this perspective, the agent's goal is to compute a best-response to the policies used by other agents. That is, given a multiagent model with state uncertainty (either a POSG or Dec-POMDP) and given some policy for the other agents $\pi_{-i} = \langle \pi_1, \ldots, \pi_{i-1}, \pi_{i+1}, \ldots, \pi_n \rangle$, we want to compute the best response π_i^{BR} for agent i. Such best-response computation is obviously important for self-interested agents (i.e., in POSGs), but is also an important component in many Dec-POMDP solution methods (Nair, Tambe, Yokoo, Pynadath, &

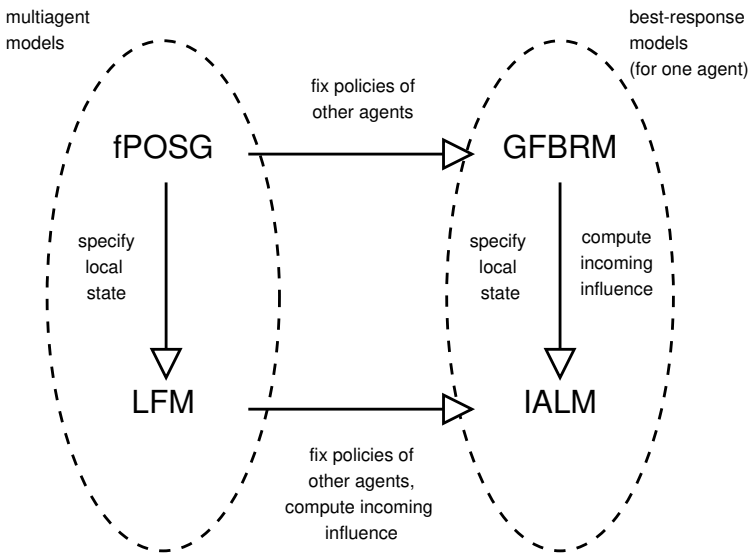


Figure 3: Overview of various models used.

Marsella, 2003; Nair, Varakantham, Tambe, & Yokoo, 2005; Kim, Nair, Varakantham, Tambe, & Yokoo, 2006; Pajarinen & Peltonen, 2011; Lauri, Pajarinen, & Peters, 2020). Also, let us point out that we make no restrictions on the policies employed by the other agents: they are general mappings from the action-observation histories $\vec{h}_j^t$ to probability distributions over actions. For instance, their policies could be learning algorithms such as Q-learning. As such, the setting we consider is very general.

As illustrated in Figure 3, we will consider a number of different types of models in this paper. The starting point is given by the fPOSG or a special case thereof (e.g., a Dec-POMDP). We refer those models as global-form models. For such models, it is possible to directly compute a best-response by fixing the policies of the other agents. We refer to the resulting POMDP as a global-form best-response model (GFBRM); these models will be introduced next. Subsequently, we will introduce *local-form models (LFMs),* which restrict the state factors that each agent primarily cares about. That is, an agent in an LFM only reasons about a subset of factors. This will then form the basis for computing best-responses in such a local model, called *influence-augmented local model (IALM),* which will be enabled by influence-based abstraction introduced in Section 4.

3.1 Global-Form Best-Response Model

In this section we define a Global-Form Best-Response Model (GFBRM) that an agent can use in order to compute a best-response in a general POSG. We first define this model and then talk about value functions for this model.[6]

Specification of the Model The basic idea of defining a best-response model is shown in Figure 4. By fixing π_{-i}, the policies of the other agents, all the choice nodes are turned

6. Our formulation here is closely related to the way best-responses are computed in DP-JESP (Nair et al., 2003): essentially our representation here is a reformulation that makes explicit the fact that fixing the policies of other agents leads to a single-agent POMDP model.

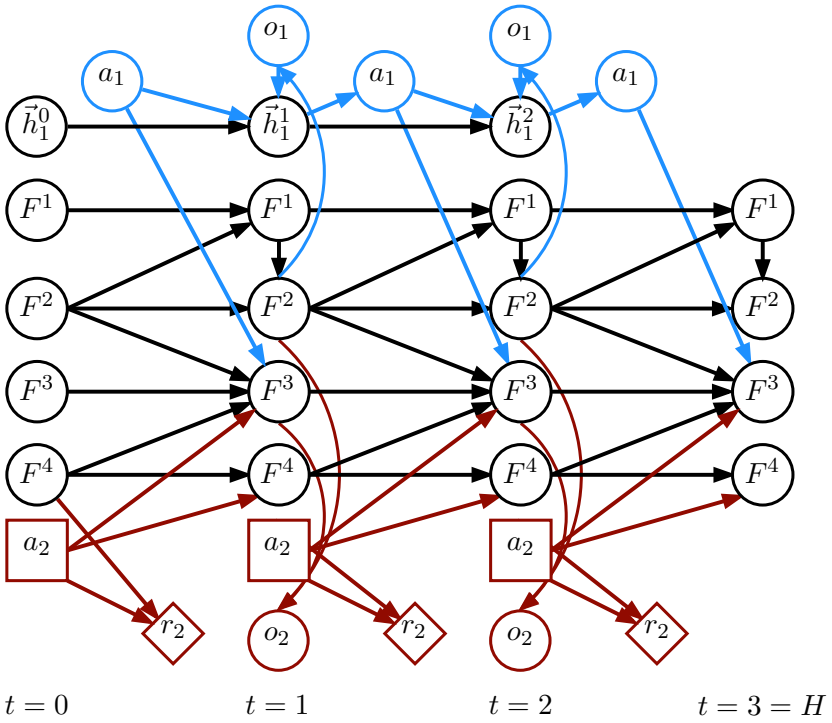


Figure 4: A hypothetical global-form best-response model for agent 2, unrolled over time. This model has a number of state factors F^k. In addition, the action-observation history $\vec{h}_1^t$ (or, more general, internal state) of agent 1 can be interpreted as a state factor in this model.

into random variables that now depend on the AOHs that those agents observed (Nair et al., 2003). So the key construct here is that the AOH of the other agent(s) is made part of the hidden state (often termed latent state factors) of the best-response model. This can be formalized as follows.

Definition 5 (Global-Form Best-Response Model)**.** Let $\mathcal{M}^{POSG} = \langle \mathcal{D}, \mathcal{S}, \mathcal{A}, T, \mathcal{R}, \mathcal{O}, O, H, b^0 \rangle$ be a (f)POSG and let π_{-i} be a profile of policies for all agents but i. We say that the POMDP $\mathcal{M}_i^{GFBR}(\mathcal{M}^{POSG}, \pi_{-i}) = \langle \bar{\mathcal{S}}_i, \mathcal{A}_i, \bar{T}_i, \bar{R}_i, \mathcal{O}_i, \bar{O}_i, H, \bar{b}_i{}^0 \rangle$ is a *Global-Form Best-Response Model (GFBRM)* for agent i, where

- $\bar{\mathcal{S}}_i$ is the set of augmented states $\bar{s}_i^t = \langle s, \vec{h}_{-i}^t \rangle$ that specify an underlying state of the POSG as well as an AOH history for all the other agents.
- $\mathcal{A}_i, \mathcal{O}_i$ are the (unmodified) sets of actions and observations for agent i.

- The transitions

$$\begin{aligned}
\bar{T}_i(\bar{s}_i^{t+1}|\bar{s}_i^t,a_i^t) &= \bar{T}_i(\langle s^{t+1},\vec{h}_{-i}^{t+1}\rangle|\langle s^t,\vec{h}_{-i}^t\rangle,a_i^t) \\
&= \bar{T}_i(\langle s^{t+1},\left(\vec{h}_{-i}^t,a_{-i}^t,o_{-i}^{t+1}\right)\rangle|\langle s^t,\vec{h}_{-i}^t\rangle,a_i^t) \\
&= \Pr(o_{-i}^{t+1},s^{t+1},a_{-i}^t|s^t,a_i^t,\vec{h}_{-i}^t) \\
&= \Pr(o_{-i}^{t+1}|a_i^t,a_{-i}^t,s^{t+1})\Pr(s^{t+1}|s^t,a_i^t,a_{-i}^t)\Pr(a_{-i}^t|\vec{h}_{-i}^t) \\
&= \left[\sum_{o_i^{t+1}} O(o^{t+1}|a^t,s^{t+1})\right] T(s^{t+1}|s^t,a^t)\pi_{-i}(a_{-i}^t|\vec{h}_{-i}^t) \qquad (3.1)
\end{aligned}$$

  with $\pi_{-i}(a_{-i}^t|\vec{h}_{-i}^t) = \prod_{j\neq i}\pi_j(a_j^t|\vec{h}_j^t)$ the probability of a_{-i}^t given $\vec{h}_{-i}^t$ according to π_{-i}.
- The observations

$$\begin{aligned}
\bar{O}_i(o_i^{t+1}|a_i^t,\bar{s}_i^{t+1}) &= \bar{O}_i(o_i^{t+1}|a_i^t,\langle s^{t+1},\left(\vec{h}_{-i}^t,a_{-i}^t,o_{-i}^{t+1}\right)\rangle) \\
&= \Pr(o_i^{t+1}|a_i^t,a_{-i}^t,s^{t+1},o_{-i}^{t+1}) \\
&= \frac{\Pr(o_i^{t+1},o_{-i}^{t+1}|a_i^t,a_{-i}^t,s^{t+1})}{\Pr(o_{-i}^{t+1}|a_i^t,a_{-i}^t,s^{t+1})} \\
&= \frac{O(o^{t+1}|a^t,s^{t+1})}{\sum_{o_i^{t+1}} O(o^{t+1}|a^t,s^{t+1})} \qquad (3.2)
\end{aligned}$$

  (Note that $o^{t+1} = \langle o_i^{t+1},o_{-i}^{t+1}\rangle$, such that the summation is over the component of o^{t+1} corresponding to agent i).
- $\bar{R}_i$ is the augmented reward model

$$\begin{aligned}
\bar{R}_i(\bar{s}_i^t,a_i^t,\bar{s}_i^{t+1}) &= \bar{R}_i(\langle s^t,\vec{h}_{-i}^t\rangle,a_i^t,\langle s^{t+1},\vec{h}_{-i}^{t+1}=\left(\vec{h}_{-i}^t,a_{-i}^t,o_{-i}^{t+1}\right)\rangle) \\
&= R_i(s^t,a_i^t,a_{-i}^t,s^{t+1}) \qquad (3.3)
\end{aligned}$$

  Note that a_{-i}^t is specified by $\bar{s}_i^{t+1}$.
- H is the (unmodified) horizon.
- $\bar{b_i}^0$ is the initial belief.

A GFBRM is a POMDP, which means that an agent can track a belief, which is now a distribution over *augmented states* $\bar{s}_i = \langle s^t,\vec{h}_{-i}^t\rangle$, as usual. We will refer to such beliefs as *global-form beliefs,* denoted b_i^g. The initial global-form belief follows directly from the initial belief of the POSG. Since at the first stage, the history of the other agents is the empty history (), it is trivially constructed from b^0:

$$\forall_s\ b_i^{g,0}(\langle s,()\rangle) \triangleq b^0(s).$$

Note that the description of the GFBRM depends rather crucially on the fact that we choose AOHs for the representation of the internal state of the other agent(s). That is, we assume that the policies of the other agent(s) are based on their AOHs. While this is a

very general model, other models of other agents with a more limited description of internal state (e.g., finite state controllers) can be useful too. For such more compact descriptions, however, it is not always possible to construct a POMDP model with an independent transition and observation model. Instead, one may need to replace $\bar{T}$,$\bar{O}$ by a combined 'dynamics function' $\bar{D}$ that specifies $\bar{D}(\bar{s}_i^{t+1},o_i^{t+1}|\bar{s}_i^t,a_i^t)$. For more details see Oliehoek and Amato (2014).[7]

Value Function Since a GFBRM is just a POMDP, all POMDP theory and solution methods apply. E.g., the optimal (action-)value function is given by:

$$Q_i^t(b_i^g,a_i^t) = R_i(b_i^g,a_i^t) + \gamma \sum_{o_i^{t+1}} \Pr(o_i^{t+1}|b_i^g,a_i^t) V_i^{t+1}(BU(b_i^g,a_i^t,o_i^{t+1})) \tag{3.4}$$

where

$$\begin{aligned} R_i(b_i^g,a_i^t) &= \mathbf{E}_{\bar{s}_i^t\sim b_i^g,\bar{s}_i^{t+1}\sim\bar{T}_i(\bar{s}_i^t,a_i^t,\cdot)}\left[\bar{R}_i(\bar{s}_i^t,a_i^t,\bar{s}_i^{t+1})\right] \\ &= \sum_{s^t}\sum_{s^{t+1}}\sum_{a_{-i}} \Pr(s^{t+1}|s^t,a) R_i(s^t,a,s^{t+1}) \sum_{\vec{h}_{-i}^t} \Pr(a_{-i}|\vec{h}_{-i}^t) b_i^g(s^t,\vec{h}_{-i}^t) \end{aligned} \tag{3.5}$$

(see Appendix A.1.1) and

$$\begin{aligned} \Pr(o_i^{t+1}|b_i^g,a_i^t) &= \mathbf{E}_{\bar{s}_i^t\sim b_i^g,\bar{s}_i^{t+1}\sim\bar{T}_i(\bar{s}_i^t,a_i^t,\cdot)}\left[\bar{O}_i(o_i^{t+1}|a_i^t,\bar{s}_i^{t+1})\right] \\ &= \sum_{s^t}\sum_{s^{t+1}}\sum_{a_{-i}}\sum_{o_{-i}^{t+1}} \Pr(s^{t+1}|s^t,a)\Pr(o^{t+1}|a,s^{t+1}) \sum_{\vec{h}_{-i}^t} \Pr(a_{-i}|\vec{h}_{-i}^t,\pi_{-i}) b_i^g(s^t,\vec{h}_{-i}^t) \end{aligned} \tag{3.6}$$

(see Appendix A.1.2.)

Solution of the GFBRM gives the best-response value for agent i:

$$V_i(\pi_{-i}) \triangleq V_i^0(b_i^{g,0}). \tag{3.7}$$

3.2 Local-Form Model

GFBRMs allow an agent i to compute a best-response policy against the fixed policies π_{-i} of the other agents. A difficulty here is that agent i needs to reason about many state factors as well as the internal state (the action-observation history) of the other agents. That is, drawing an analogy to human interactions, it is like in a simple collaborative task (e.g., carrying a table), we would need to reason over the inner working of our collaborator's

7. Essentially in such a setting we have that augmented states are tuples of nominal states and internal states of other agents $\bar{s}_i^t = \langle s^t,I_{-i}^t\rangle$. The internal states of the other agent are updated based upon the taken actions and observations, but do not store those actions and observations. This means that, in general, $\bar{D}$ is specified as a marginal:

$$\begin{aligned} \bar{D}(\bar{s}_i^{t+1},o_i^{t+1}|\bar{s}_i^t,a_i^t) &= \Pr(\langle s^{t+1},I_{-i}^{t+1}\rangle,o_i^{t+1}|\langle s^t,I_{-i}^t\rangle,a_i^t) \\ &= \sum_{a_{-i}^t,o_{-i}^{t+1}} \Pr(I_{-i}^{t+1}|I_{-i}^t,a_{-i}^t,o_{-i}^{t+1}) O(o^{t+1}|a^t,s^{t+1}) T(s^{t+1}|s^t,a^t) \pi_{-i}(a_{-i}^t|I_{-i}^t) \end{aligned}$$

and it is not possible to decompose it into a separate transition and observation function.

brain, as well as over the sequence of images that he or she perceives. Clearly, such an approach is infeasible in general. To make a step in the direction to overcome this problem, here we introduce *local-form models (LFMs)* which restrict the set of state factors that each agent primarily cares about, and eliminates the dependence on the AOH of other agents.

Local States An LFM augments an fPOSG with a function that provides a description of each agent's *local state*, i.e., the set of variables that each agent will model as part of its local problem.[8] Local state descriptions comprise potentially overlapping subsets of state factors that will allow us to decompose an agent's best-response computation from the global state. We start with some definitions.

Definition 6 (Local state function)**.** The *local state function* $S : \mathcal{D} \to 2^{\mathcal{F}}$ maps from agents to subsets of state factors $S(i) \subseteq \mathcal{F}$.

The local state function defines the local state space of each agent. In particular, we say that a state factor $F \in \mathcal{F}$ is *modeled* by an agent i if it is part of its local state space: $F \in S(i)$.

Definition 7 (Local state space)**.** The *local state space* of agent i is defined as the Cartesian product of the values that its modeled state factors can take:

$$\mathcal{X}_i \triangleq \prod_{k \text{ s.t.} F^k \in S(i)} \mathcal{F}^k \tag{3.8}$$

(remember that $\mathcal{F}$ is the set of state factors, while $\mathcal{F}^k$ is the set of values that the k-th state factor F^k can take).

Definition 8 (Observation-relevant factor)**.** We say that a state factor F is *observation-relevant* for an agent i, denoted $\text{ORel}_i(F)$, if it affects the probability of the agent's observation. That is, when in the 2DBN there is a link from F^t to o_i^t (i.e., F is a parent of o_i^t).

Definition 9 (Reward-relevant factor)**.** Similarly, a state factor F is *reward-relevant* for an agent i, $\text{RRel}_i(F)$ if it affects the agent's rewards, i.e., if F^t or F^{t+1} is a parent of R_i^t.

We can now define the local-form model.

Definition 10 (Local-form model)**.** A *local-form POSG,* also referred to as *local-form model (LFM),* is a pair $\mathcal{M}^{LFM} = \langle \mathcal{M}, S \rangle$, where $\mathcal{M}$ is an fPOSG and S is a local state function such that, for all agents:

1. All observation-relevant factors are in the local state: $\forall_i \forall_F \text{ ORel}_i(F) \implies F \in S(i)$.
2. All reward-relevant factors are in the local state: $\forall_i \forall_F \text{ RRel}_i(F) \implies F \in S(i)$.

8. Note that the word 'local' does not need to imply any form of spatial proximity. For instance, in House Search the agent might model its own location (which is spatial), and whether the target has been found (not spatial).

Modeled and Non-modeled Factors The basic idea behind the definition of the local-form model is to avoid reasoning over the subset of variables from the global-form model that are superfluous when it comes to computing the best response. Therefore, these non-modeled factors can be abstracted away. The requirements on observation- and reward-relevant factors make certain that the observation probabilities and rewards are still specified in this abstracted model. Note also that this means that we will only be able to abstract away (latent) state variables, not observation variables themselves. We will show that such latent factor abstraction can, in principle, be performed without loss in value. This certainly would not be the case for abstracting away observation variables: in general this would lead to a loss of information and a corresponding drop in achievable value (Oliehoek et al., 2008a).

The focus in this text is on the best-response perspective for one agent i. This allows us to divide the set of state factors in ones modeled by agent i's local problem (indicated with x) and ones that are not modeled (indicated with y).[9] To reduce the notational load, we will no longer distinguish between a factor (F^k above) and its values ($\mathcal{F}^k$ above). In particular, we will simply write

- x^k (an instantiation of) a modeled factor (with index k),
- x_i (an instantiation of) all modeled factors of agent i,
- y^k (an instantiation of) a non-modeled factor (with index k),
- y_i (an instantiation of) all non-modeled factors of agent i,

such that $s^t = \langle x_i^t, y_i^t \rangle$. We stress that 'modeled' is different from 'observed'. In particular, our aim is to construct a smaller POMDP with fewer (modeled) factors, but those factors may not be observable. In fact, all state factors x^k (and of course also y^k) are expressed as latent variables. When an agent can somehow (noisily) perceive information about x^k, this should be modeled by the observation function: there should be an arrow from such factors to the observation o_i of the agent and the CPT of o_i should appropriately express the observability of factor. Note that by construction of the LFM (cf. Definition 10), no such dependencies may exist from a y^k to o_i. In general, the observation o_i may itself consists of multiple observation factors, but we will not consider this in this paper.

Transition Probabilities In an LFM, the probability of the next local state is the marginal of the entire state:

$$\Pr(x_i^{t+1} | s^t, a_i, a_{-i}) = \sum_{y_i^{t+1}} \Pr(x_i^{t+1}, y_i^{t+1} | s^t, a_i, a_{-i}) \tag{3.9}$$

In an LFM, just as in a normal fPOSG, the flat transition probabilities on the right hand side of this equation are given by the product of the CPTs. However, from the perspective of an agent i we can now group these CPTs in three different categories: 1) those corresponding to modeled factors that are only affected by other factors and actions

9. More generally, from the perspective of agent i, S partitions the modeled factors $S(i)$ in two sets: a set of *private* factors that it models but other agents do not, and a set of *mutually-modeled factors* (MMFs) that are modeled by agent i as well as some other agent j. This distinction plays a crucial role in influence search for TD-POMDPs (Witwicki & Durfee, 2010a), but is less important for computing best-responses as considered in this document.

that are modeled, 2) those corresponding to modeled factors that are affected by at least one factor or action of the external problem, and 3) those corresponding to non-modeled factors. We will refer to the state factors corresponding to these as:

1. *Only-locally-affected factors (OLAFs)* $\mathring{x}^k$. These can have incoming arrows from all modeled factors x_i^t at the previous stage, and from all modeled factors x_i^{t+1} intra-stage (but, obviously, excluding $\mathring{x}^{k,t+1}$ itself, and respecting a non-cyclic structure as any 2DBN).
2. *Non-locally-affected factors (NLAFs)* $\tilde{x}^k$. These are affected by at least one non-modeled (intra-stage or previous-stage) factor or action of another agent.
3. *Non-modeled factors (NMFs)* y^k.

(Note that the oversets on x were chosen to resemble 'o' and 'n' for OLAF and NLAF respectively). These three types of factors are illustrated in Figure 5a, which shows a hypothetical local-form model. Using the introduced notation, we can write the transition probabilities as:

$$\begin{aligned}\Pr(s^{t+1}|s^t,a_i,a_{-i}) &= \left[\Pr(\mathring{x}_i^{t+1}|\dots)\Pr(\tilde{x}_i^{t+1}|\dots)\Pr(y_i^{t+1}|\dots)\right] \\ &= \Pr(\mathring{x}_i^{t+1}|x_i^t,\tilde{x}_i^{t+1},a_i)\Pr(\tilde{x}_i^{t+1}|x_i^t,\mathring{x}_i^{t+1},y_i^t,y_i^{t+1},a_i,a_{-i})\Pr(y_i^{t+1}|x_i^t,x_i^{t+1},y_i^t,a_i,a_{-i}) \quad (3.10)\end{aligned}$$

with

- $\Pr(\mathring{x}_i^{t+1}|x_i^t,\tilde{x}_i^{t+1},a_i)$ representing a product of CPTs of OLAFs $\mathring{x}^k$:

$$\Pr(\mathring{x}_i^{t+1}|x_i^t,\tilde{x}_i^{t+1},a_i) = \prod_{k\in OLAF(i)} \Pr(\mathring{x}^{k,t+1}|x_i^t,x_i^{t+1},a_i) \quad (3.11)$$

  Note that although such individual factors $\mathring{x}^{k,t+1}$ can have intra-stage dependencies on other OLAFs $\mathring{x}^{l,t+1}$ (i.e., $\mathring{x}^{k,t+1}$ can depend on x_i^{t+1} which can include other OLAFs $\mathring{x}^{l,t+1}$), the product term $\Pr(\mathring{x}_i^{t+1}|x_i^t,\tilde{x}_i^{t+1},a_i)$ itself can only have intra-stage dependencies on $\tilde{x}_i^{t+1}$. [10]

- $\Pr(\tilde{x}_i^{t+1}|x_i^t,\mathring{x}_i^{t+1},y_i^t,y_i^{t+1},a_i,a_{-i})$ the product of NLAF probabilities:

$$\Pr(\tilde{x}_i^{t+1}|x_i^t,\mathring{x}_i^{t+1},y_i^t,y_i^{t+1},a_i,a_{-i}) = \prod_{k\in NLAF(i)} \Pr(\tilde{x}^{k,t+1}|x_i^t,x_i^{t+1},y_i^t,y_i^{t+1},a_i,a_{-i}) \quad (3.12)$$

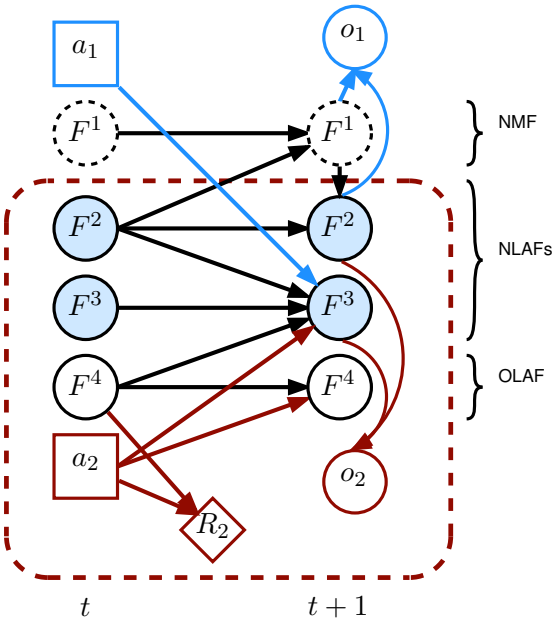
- $\Pr(y_i^{t+1}|x_i^t,x_i^{t+1},y_i^t,a_i,a_{-i})$ the product of probabilities of the NMFs y^k:

$$\Pr(y_i^{t+1}|x_i^t,x_i^{t+1},y_i^t,a_i,a_{-i}) = \prod_{k\in NMF(i)} \Pr(y^{k,t+1}|x_i^t,x_i^{t+1},y_i^t,y_i^{t+1},a_i,a_{-i}) \quad (3.13)$$

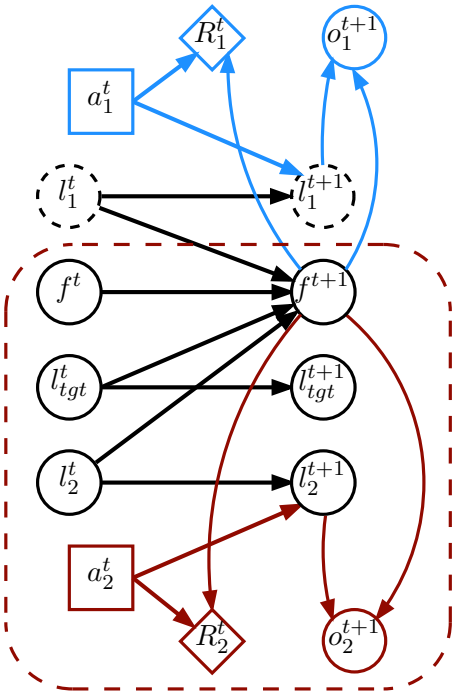
10. Note that the intra-stage OLAFs $\mathring{x}_i^{t+1}$ will not appear in the conditioning set ('behind the pipe') as they have all been multiplied in (they are 'before the pipe'). Since the 2DBN is non-cyclical per definition, this does not present any problems. A more explicit way of writing this is as follows. In general the OLAFs can now depend on some NLAFs $\tilde{x}_i^{ISD,t+1}$ that act as intra-stage dependencies:

$$\Pr(\mathring{x}_i^{t+1}|x_i^t,\tilde{x}_i^{ISD,t+1},a_i^t) \triangleq \prod_{k\in OLAF(i)} \Pr(\mathring{x}^{k,t+1}|x_i^t,a_i^t,x^{ISD(k),t+1})$$

  with $x^{ISD(k),t+1}$ denoting the intra-stage parents of $\mathring{x}^{k,t+1}$. To reduce the notational burden, however, we will use the shorthands from (3.11).



(a) Illustration of an abstract local-form model for agent 2. Factors can be divided into non-modeled factors (F^1), non-locally-affected factors (F^2, F^3, shaded in this figure), and locally-affected factors (F^4). Also note that F^4 is reward-relevant, while F^2 and F^3 are observation-relevant factors.



(b) Local-form model for agent 2 in the house search problem without intra-stage dependencies. The 'found' variable f is the only NLAF since it is affected by NMF l_1^t.

Figure 5: Local-form models.

Value Function An LFM contains an fPOSG and as such best-response values for an agent i can be defined using the techniques discussed above in Section 3.1. In particular, we can just ignore the local state function and apply the definition of Q-value (3.4) with the previously stated definitions of $R_i(b_i^g,a_i)$ (3.5) and $\Pr(o_i^{t+1}|b_i^g,a_i)$ (3.6).

Clearly, however, we would like to now rewrite the value function in a way that represents the local structure imposed by the LFM requirements and exploits this for computational benefits. The former is possible: for an LFM, we can indeed derive a expression for $R_i(b_i^g,a_i)$ that is more local (see Appendix A.2.1).

$$R_i(b_i^g,a_i^t) = \sum_{x_i^t} \sum_{x_i^{t+1}} R_i(x_i^t,a_i^t,x_i^{t+1}) \Pr(x_i^t,x_i^{t+1}|b_i^g,a_i^t,\pi_{-i}), \tag{3.14}$$

where (remember $s^t = \langle x_i^t,y_i^t \rangle$)

$$\Pr(x_i^t,x_i^{t+1}|b_i^g,a_i^t,\pi_{-i}) \triangleq \sum_{y_i^t} \sum_{a_{-i}^t} \Pr(x_i^{t+1}|s^t,a_i^t,a_{-i}^t) \sum_{\vec{h}_{-i}^t} \Pr(a_{-i}^t|\vec{h}_{-i}^t,\pi_{-i}) b_i^g(s^t,\vec{h}_{-i}^t). \tag{3.15}$$

And, similarly, we can find a new, local, expression for the observation probability (Appendix A.2.2):

$$\Pr(o_i^{t+1}|b_i^g,a_i^t) = \sum_{x_i^{t+1}} \Pr(o_i^{t+1}|a_i^t,x_i^{t+1}) \Pr(x_i^{t+1}|b_i^g,a_i^t,\pi_{-i}) \tag{3.16}$$

where

$$\Pr(x_i^{t+1}|b_i^g,a_i^t) \triangleq \sum_{s^t} \sum_{a_{-i}^t} \Pr(x_i^{t+1}|s^t,a_i^t,a_{-i}^t) \sum_{\vec{h}_{-i}^t} \Pr(a_{-i}^t|\vec{h}_{-i}^t \pi_{-i}) b_i^g(s^t,\vec{h}_{-i}^t). \tag{3.17}$$

These new definitions of $R_i(b_i^g,a_i)$, $\Pr(o_i^{t+1}|b_i^g,a_i)$ can be used directly in conjunction with the definition of Q-value (3.4).

However, even though these definitions (3.14) and (3.16) are local, they still depend on the global-form belief and this must perform summations over full states s^t and histories of other agents $\vec{h}_{-i}^t$ via (3.15) and (3.17), rendering them intractable for larger problems. In the next section, we will investigate formulations that are based on more local beliefs to try and overcome this computational hurdle. Before jumping to this, we first state an observation:

Observation. The presented definition of an LFM with multiple agents is a strict generalization of a single agent problem.

While this is a simple observation, the upshot of this is that the theory of influence-based abstraction that we will introduce in the remainder of this paper also directly applies to single-agent settings.[11] Specifically, the formulas and results we will derive have more specific forms for the single-agent case. We discuss relations to abstraction methods for single-agent settings in Section 8.3.

11. We acknowledge Craig Boutilier, for pointing this out.

4. Influence-Based Abstraction

In the previous section we introduced the GFBRM, which could be used to compute a best response against a fixed policy of other agents. This model gives a straightforward way of formulating the problem of computing a best-response. However, it is specified over the global state and internal state of other agents (i.e., their AOHs), which means that solving this model is computationally intractable.

To provide a more localized perspective, the local-form POSG defines for each agent a subset of factors that it should be concerned with. However, even if the policies of the other agents are fixed, it is not clear how an agent i can restrict its reasoning to its local state x_i: the non-modeled factors will still affect the local state transitions. Intuitively, we need to capture the *influence* that the non-modeled part of the problem exerts on the modeled part.

In this section, we formalize this intuition. In particular, we treat an LFM from the perspective of one agent and consider how that agent is affected by the other agents and can compute a best response against that 'incoming' influence.[12]

In an attempt to avoid notation overload, we first present a formulation without considering intra-stage connections. The general formulation that can deal with such connections is given in Section 5.

4.1 Definition of Influence

As discussed in Section 3.1, when the other agents are following a fixed policy, they can be regarded as part of the environment. The resulting decision problem can be represented by the complete unrolled DBN, as we saw in Figure 4 on page 799. In this figure, a node F^t is a different node than F^{t+1} and an edge at (emerging from) stage t is a different from the edge at $t+1$ that corresponds to the same edge in the 2DBN. Given this uniqueness of nodes and edges, we can define the 'influence' as follows.

4.1.1 Influence Links, Sources and Destinations

Intuitively, the influence of other agents is the effect of those edges leading into the agent's local problem. We say that every directed edge from outside the local model (e.g., from an NMF or action of another agent) to inside the local model (e.g., to a modeled state factor, observation variable, or reward), is an *influence link* $\langle u^t, v^t \rangle$, where u^t is called the *influence source* and v^t is the *influence destination*. In this section, we will assume that influence links traverse a stage of the process (i.e., that the influence source for a destination v^t lies in the stage $t-1$), but since we will also consider intra-stage influence links at a later point in this document, to keep notation consistent, we label an entire influence link with the stage-index of its destination.

For example, let us consider the House Search problem's LFM shown in Figure 5b on page 805. It shows that the link from l_1^t, the location of agent 1, to the 'target found' variable f^{t+1} is an influence link, such that we would write the link as $\left\langle u^{t+1} = l_1^t, v^{t+1} = f^{t+1} \right\rangle$, similarly $\left\langle u^t = l_1^{t-1}, v^t = f^t \right\rangle$ would denote the influence link in the preceding time step.

12. An agent also exerts 'outgoing' influence on other agents, but this is irrelevant for best response computation.

Assuming no intra-stage influence links, an influence source u^t can be either an action a_j^{t-1} or non-modeled state factor y^{t-1}. We write $u^t_{\to i} = \langle y_u^{t-1}, a_u^{t-1} \rangle$ for an instantiation of all influence sources exerting influence on agent i at stage t. That is, in the case of multiple influence links pointing to modeled factors in stage t, y_u^{t-1} denotes the (value of) influence sources that are state factors, while a_u^{t-1} corresponds to those influence sources that are actions. For instance, in our House Search example, $y_u^{t-1} = \{l_1^{t-1}\}$, while $a_u^{t-1} = \emptyset$ since there are no actions that are influence sources. We write $\vec{h}_u^{t-1}$ for the AOHs of those other agents whose action is an influence source (i.e., $\vec{h}_u^{t-1}$ and a_u^{t-1} involve the same agents) .

In general, an influence destination can be either a (per definition non-locally-affected) modeled factor $\tilde{x}^t$, an observation variable o_i^t, or a local reward node R_i^t. But Definition (10) requires reward- or observation-relevant factors to be included in the local state; effectively we restrict ourselves to the setting where the influence destination is an NLAF. This restriction is without loss in generality: because we will introduce (in Section 5) the machinery to deal with *intra-stage* influence links, influences on observations and rewards can easily be dealt with by introducing a 'dummy' NLAF that acts as a proxy for the observation or reward.[13] A similar construction can be used to deal with settings where actions of other agents would directly influence the observations or rewards of the agent under concern. As such, the capability of dealing with such intra-stage dependencies is critical for the applicability of the theory of influence-based abstraction.

4.1.2 Sufficient Information to Predict Influences: D-Separating Sets

If agent i would in advance know the value of its influence sources at different time steps, it could easily compute its best response by making use of only this knowledge and its local model. For instance, if in the House Search example of Figure 5b on page 805, we would in advance perfectly know the location of agent 1 at each timestep and thus know the sequence of values for l_1^t, we could decouple the local problem by just looking at the appropriate slices of the CPT of f^t.

Of course, this is in general not possible, since the influence sources are random variables. However, the influence exerted on agent i can be captured if we know the probability distribution over their values. That is, in order to predict the probability of some $\tilde{x}_i^{t+1}$ (i.e., an influence destination) agent i only cares about the following marginal probability

$$\sum_{u^{t+1}_{\to i}} \Pr(\tilde{x}_i^{t+1} | x_i^t, a_i^t, u^{t+1}_{\to i}) \Pr(u^{t+1}_{\to i} | \ldots), \tag{4.1}$$

where the dots $(\ldots)$ indicate any information that agent i needs to predict the probability of the values of the influence sources as accurately as possible. Moreover, since these probabilities will be used to plan a best response, correlations between influence sources and local states are important. This unfortunately means that in general, we might need to condition $\Pr(u^{t+1}_{\to i} | ...)$ on the entire history of actions, observations and and local states.

Fortunately, it turns out that in many cases we can find substantially more compact representations of the conditional probability of $u^{t+1}_{\to i}$, by making use of the concept of *d-separation* in graphical models (Bishop, 2006; Koller & Friedman, 2009). In particular,

13. E.g., to deal with an observation destination, we can transform the observation o_i to a state factor F^o and introduce a new observation variable that has a deterministic CPT depending only on F^o.

when two nodes A,B in a Bayesian network are d-separated given some of subsets D of evidence nodes, then A and B are conditionally independent given D, which means that $\Pr(A|D,B) = \Pr(A|D)$ and vice versa. Whether nodes are d-separated can be easily checked, by applying a small set of rules on the graph (Bishop, 2006, chapter 8).

Now, we can define the influence as a conditional probability distribution over $u_{\rightarrow i}^{t+1}$, given a d-separating set. Specifically, let D_i^{t+1} be a subset of variables (possibly including state factors and actions) in the local problem of agent i at stages $0,\ldots,t$,

Definition 11 (D-separating set)**.** D_i^{t+1} is a *d-separating set for agent i's influence at stage* $t+1$ if and only if it d-separates $y_u^t,\vec{h}_u^t$ from $x_i^t,\vec{h}_i^t$. That is, if:

$$\forall_{y_u^t,\vec{h}_u^t} \qquad \Pr(y_u^t,\vec{h}_u^t|x_i^t,\vec{h}_i^t,D_i^{t+1},b^0,\pi_{-i}) = \Pr(y_u^t,\vec{h}_u^t|D_i^{t+1},b^0,\pi_{-i}). \tag{4.2}$$

This definition implies that remembering more than D_i^{t+1} is not useful for predicting $y_u^t,\vec{h}_u^t$ and hence for predicting $u_{\rightarrow i}^{t+1} = \langle y_u^t,a_u^t\rangle$. Given their policies, the actions of other agents only depend on their AOHs. We note that when the other agents use simpler (e.g., memoryless) policies, one might not need to predict the full action observation history for agents whose actions are influence sources. Instead we will only need to predict relevant part, denoted $\rho(\vec{h}_u^t)$. Similarly, there might be a sufficient statistic σ that summarizes D_i^{t+1} and still is enough to provide the conditional independence. In such case we would only need

$$\forall_{y_u^t,\vec{h}_u^t} \qquad \Pr(y_u^t,\rho(\vec{h}_u^t)|x_i^t,\vec{h}_i^t,\sigma(D_i^{t+1}),b^0,\pi_{-i}) = \Pr(y_u^t,\rho(\vec{h}_u^t)|\sigma(D_i^{t+1}),b^0,\pi_{-i}). \tag{4.3}$$

To avoid a further burden on notation, we will not explicitly consider these special cases, and in our description assume that we condition on the values of the variables in the d-separating set. However, we will see examples of such more compact description of the information needed to predict the influence sources.

Deciding on D_i^{t+1} needs to be done in advance to compute the influence. When the d-separating set is compressed, $\sigma(D_i^{t+1})$, this will typically involve input by the human designer. However, we note that efficient algorithms are known to compute a minimal d-separating set (Acid & De Campos, 1996; Tian, Paz, & Pearl, 1998; van der Zander & Liśkiewicz, 2020) in cases where this would be infeasible to do by hand.

Example. Figure 6 illustrates a d-separating set D_i^3 for agent $i = 2$ in House Search. It shows that, in order to accurately compute the probability of influence source l_1^2, agent 2 needs to condition on $f^{0:2}$, the history of the found variable, as well as the histories of the location of the target l_{tgt} and its own location l_2. This dependence on the history in general leads to large conditioning sets, but in many cases the history can be represented more compactly. For instance, in House Search the 'found' variable can only switch on (not off) which means that its history $f^{0:t}$ can be summarized compactly. In cases where the target is static the same holds for $l_{tgt}^{0:t}$.

Example. Figure 7 describes a variant of the planetary exploration domain (Witwicki & Durfee, 2010b). Here agent 2 is a mars rover which is tasked with navigating to some goal. Agent 1 is a satellite which can aid the rover by planning a path, but this will use up computational resources and battery power modeled by bt_1^t (which it may want to use to

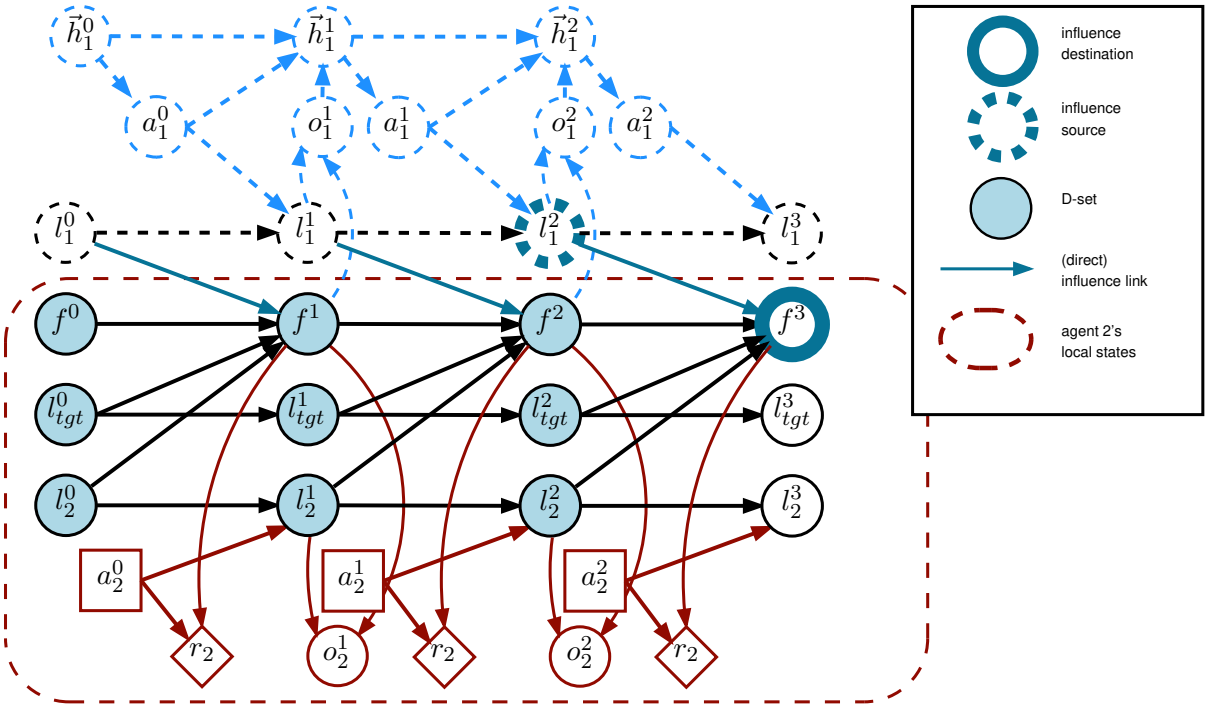


Figure 6: Illustration of the incoming influence on protagonist agent $i = 2$ in House Search at stage $t = 3$. f^3 is the only influence destination, with influence source $y_u^2 = l_1^2$ (i.e., $u^3_{\rightarrow i} = \langle l_1^2 \rangle$). The shaded nodes indicate the d-separating set D_i^3, which, in accordance with (4.2), d-separates the influence source l_1^2, from agent 2's AOH $\vec{h}_i^t$ and possibly remaining other local variables x_i^t (in this case there are no such variables, but one could imagine adding a battery life variable for agent 2).

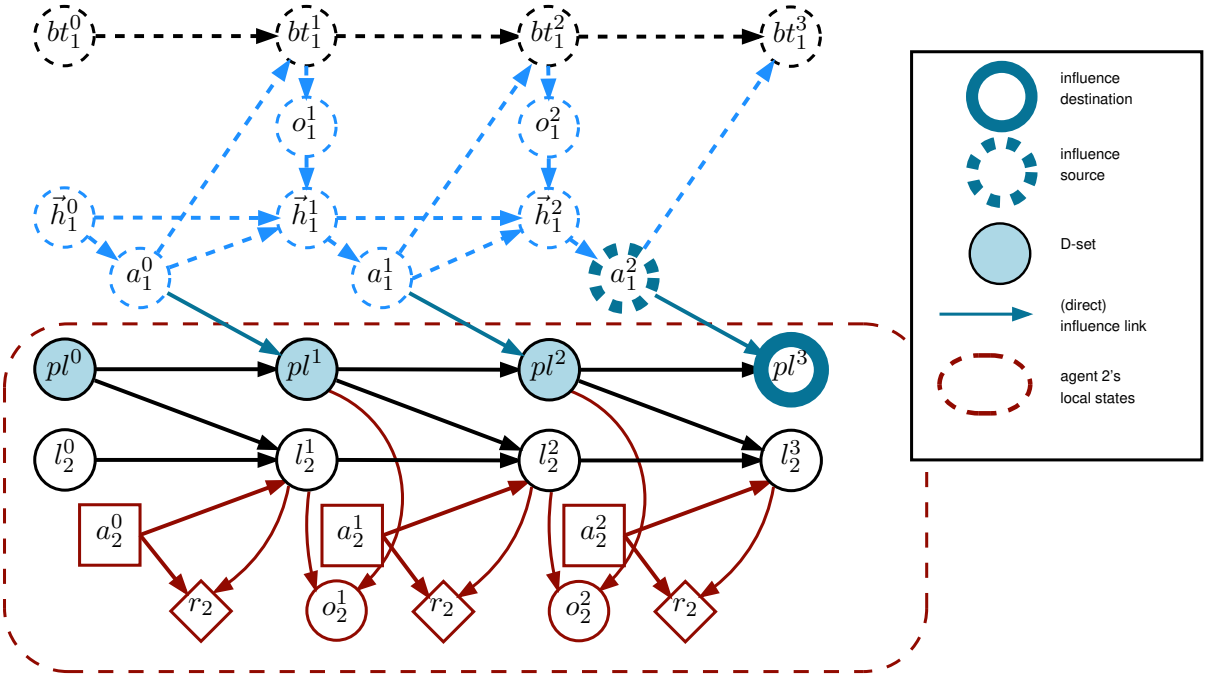


Figure 7: Illustration of the influence experienced by the mars rover (agent $i = 2$) at stage $t = 3$ in the PLANETARY EXPLORATION domain. If the satellite (agent 1) computes and transmits a plan (*pl*), the rover can more effectively navigate from that point onward.

support other rovers too, for instance). In the figure this is illustrated by the fact that the action of agent 1 $a_1 \in \{NOOP,PLAN\}$ (which now is the influence source) determines if there is a plan available for agent 2, modeled by a binary variable *pl* (which is the influence destination). In this example, the d-separating set only contains this variable *pl*. Again its history can be compactly summarized: as having the plan can only turn true, we can just store the time (if any) at which *pl* was switched to true.

4.1.3 The Influence Exerted on Agent i

Given the above machinery, we can now state our definition of influence:

Definition 12 (Incoming Influence)**.** The *incoming influence* at stage $t+1$, denoted $I^{t+1}_{\rightarrow i}(\pi_{-i})$, is a conditional probability distribution over values of the influence sources $u^{t+1}_{\rightarrow i}$:

$$I(u^{t+1}_{\rightarrow i}|D^{t+1}_i) \triangleq \sum_{\vec{h}^t_u} \Pr(a^t_u|\vec{h}^t_u) \Pr(y^t_u,\vec{h}^t_u|D^{t+1}_i,b^0,\pi_{-i}). \tag{4.4}$$

In order to predict a^t_u (the 'influence source actions') we need to predict the action-observation histories of the corresponding agents $\vec{h}^t_u$, but otherwise these histories are not needed and can thus be marginalized out. Note that, to reduce notational burden we drop arguments that can be inferred, such as b^0,π_{-i}. That is, $I(u^{t+1}_{\rightarrow i}|D^{t+1}_i)$ is shorthand for $I^{t+1}_{\rightarrow i}(u^{t+1}_{\rightarrow i}|D^{t+1}_i,b^0,\pi_{-i})$. In cases where we want to refer to this distribution as a whole, we will write $I^{t+1}_{\rightarrow i}(\pi_{-i})$, or use the shorthand $I^{t+1}_{\rightarrow i}$.

We will also say that this is the influence *exerted* on agent i at stage t or *experienced* by agent i at stage $t+1$. So far, these notions coincide, but when we consider intra-stage connections in the next section, we will discriminate between these concepts.

Finally, we are in position to specify the complete influence on agent i:

Definition 13 (incoming influence point)**.** An *incoming influence point* $I_{\to i}(\pi_{-i})$ for agent i, specifies the incoming influences for all stages $I_{\to i}(\pi_{-i}) = \big(I^1_{\to i}(\pi_{-i}), \ldots, I^H_{\to i}(\pi_{-i})\big)$.

As we will see in the remainder of this paper, an influence point contains all the information about the non-modeled part of the problem that agent i needs to compute a best response 'locally', i.e., only using its local model and that influence point. This can bring computational benefits for instance when there would be changes in the local model that require repeatedly performing planning, or in cases where the influence point can be computed easily. This form of influence-based abstraction, however, is not providing a free lunch (Wolpert & Macready, 1995): in general computing the incoming influences (4.4) for the different stages comprise a set of challenging inference problems. Fortunately, traction can still be gained in many special cases of problems identified in past work (Becker et al., 2003, 2004; Varakantham et al., 2009; Petrik & Zilberstein, 2009; Witwicki & Durfee, 2010a, 2010b, 2011; Velagapudi et al., 2011; Witwicki, 2011; Witwicki et al., 2012; Oliehoek et al., 2012), and IBA gives a unified perspective on these. Moreover, it can be used as tool to identify further special cases that allow for efficient solution, such as the class of ND-POMDPs discussed in Section 7. Given the potential benefits of using influence representations (Witwicki et al., 2012), such future search for special cases of problems that allow for compact influence specifications together with the inference algorithms that efficiently compute these is an important line of research. Our definition of influence in this section provides the general framework in which these special cases should be sought.

4.2 The Influence-Augmented Local Model (IALM)

Given the above definition of influence, we can now define a smaller *local* model for our protagonist agent i. The main idea is that given an incoming influence point, agent i no longer needs to reason over the non-modeled part of the problem. Instead, it can use the influence to compute marginal probabilities as expressed by (4.1), and this will allow it to compute an exact best-response.

In this section, we will first investigate a single NLAF and how the influence on it can be incorporated. Then we move to talk about the case where multiple variables in the local state $S(i)$ are non-locally affected. Then we proceed to the formal definition of the IALM, and how it can be solved.

4.2.1 Induced CPTs

In the case of a single influence destination, we can interpret (4.1) as constructing a new 'influence-induced' CPT:

Definition 14 (Induced CPT)**.** Let $\tilde{x}^{t+1}$ be an influence destination, and u^{t+1} (the instantiation of) the corresponding influence sources. Given the influence $I^{t+1}_{\to i}(\pi_{-i})$, and its

d-separating set D_i^{t+1}, we define the *induced CPT* for $\tilde{x}^{t+1}$ as the CPT that has probabilities:

$$p_{I_{\rightarrow i}^{t+1}}(\tilde{x}^{t+1}|x_i^t,D_i^{t+1},a_i) = \sum_{u_{\rightarrow i}^{t+1}=\langle y_u^t,a_u\rangle} \Pr(\tilde{x}^{t+1}|x_i^t,a_i,u_{\rightarrow i}^{t+1})I(u_{\rightarrow i}^{t+1}|D_i^{t+1}) \tag{4.5}$$

It is important to note that an induced CPT is specified purely in *local* terms, i.e., making use of variables that are modeled by our protagonist agent i. Therefore, the basic idea is that we can now define a smaller *local* model—which we will call the *Influence-Augmented Local Model (IALM)*—by replacing the CPTs of influence destinations (i.e., NLAFs) by induced CPTs.

4.2.2 Dealing With Multiple NLAFs

In case that there are multiple NLAFs, i.e., multiple variables $\tilde{x}^{t+1}$ in the local state space $S(i)$ that are affected non-locally at the same stage $t+1$, the story is slightly more involved, since we need to deal with their correlations.

Ideally, we would want to treat induced CPTs in the same way as normal CPTs; that is, we would represent the joint probability of NLAFs as a the product of induced CPTs:

$$\Pr(\tilde{x}_i^{t+1}|x_i^t,D_i^{t+1},a_i,I_{\rightarrow i}^{t+1}) = \prod_{k\in NLAF(i)} p_{I_{\rightarrow i}^{t+1}}(\tilde{x}^{k,t+1}|x_i^t,D_i^{t+1},a_i). \tag{4.6}$$

However, in general this is not possible since the different $\tilde{x}^{k,t+1}$ are correlated via any common influence sources. That is, in general the probability is given by:

$$\Pr(\tilde{x}_i^{t+1}|x_i^t,D_i^{t+1},a_i,I_{\rightarrow i}^{t+1}) = \sum_{u_{\rightarrow i}^{t+1}=\langle y_u^t,a_u\rangle} I(u_{\rightarrow i}^{t+1}|D_i^{t+1}) \prod_{k\in NLAF(i)} \Pr(\tilde{x}^{k,t+1}|x_i^t,a_i,u_{\rightarrow i}^{t+1}) \tag{4.7}$$

Of course, in certain cases a factorization as induced CPTs is possible. The above equations directly make clear when this is the case.

Proposition 1. *If each NLAF $\tilde{x}^{k,t+1}$ has its own influence sources $u^{k,t+1}$ (and these do not overlap), and if these sources are conditionally independent given D_i^{t+1}:*

$$I(u_{\rightarrow i}^{t+1}|D_i^{t+1}) = \prod_{k\in NLAF(i)} I(u^{k,t+1}|D_i^{t+1}),$$

then the joint probability of NLAFs factorizes as the product of induced CPTs as shown in (4.6).

Proof. Under stated conditions, we can rewrite as follows:[14]

$$
\begin{aligned}
(4.7) &= \sum_{u_{\rightarrow i}^{t+1}=\langle \dots,u^{k,t+1},\dots\rangle} I(u_{\rightarrow i}^{t+1}|D_i^{t+1}) \prod_{k\in NLAF(i)} \Pr(\tilde{x}^{k,t+1}|x_i^t,a_i,u^{k,t+1}) \\
&= \sum_{u_{\rightarrow i}^{t+1}=\langle \dots,u^{k,t+1},\dots\rangle} \left[\prod_{k\in NLAF(i)} I(u^{k,t+1}|D_i^{t+1})\right] \prod_{k\in NLAF(i)} \Pr(\tilde{x}^{k,t+1}|x_i^t,a_i,u^{k,t+1}) \\
&= \sum_{u_{\rightarrow i}^{t+1}=\langle \dots,u^{k,t+1},\dots\rangle} \prod_{k\in NLAF(i)} I(u^{k,t+1}|D_i^{t+1}) \Pr(\tilde{x}^{k,t+1}|x_i^t,a_i,u^{k,t+1}) \\
&= \prod_{k\in NLAF(i)} \sum_{u^{k,t+1}} I(u^{k,t+1}|D_i^{t+1}) \Pr(\tilde{x}^{k,t+1}|x_i^t,a_i,u^{k,t+1}) = (4.6)
\end{aligned}
$$

□

4.2.3 The IALM: A Formal Model to Incorporate Influence

Here we formally define the IALM, which is a non-stationary POMDP, since at every stage the influence destinations can be influenced in a different manner.

Definition 15 (IALM)**.** Given an LFM, $\mathcal{M}^{LFM}$, and profile of policies for other agents π_{-i}, an *Influence-Augmented Local Model (IALM)* for agent i is a POMDP $\mathcal{M}_i^{IALM}(\mathcal{M}^{LFM},\pi_{-i}) = \left\langle \bar{\mathcal{S}},\mathcal{A}_i,\bar{T}_i,\bar{R}_i,\mathcal{O}_i,\bar{O}_i,H,b_i^{l,0}\right\rangle$, where

- $\bar{\mathcal{S}}$ is the set of augmented states $\bar{s}_i^t = \langle x_i^t,D_i^{t+1}\rangle$ that specify an underlying local state of the POSG, as well as the d-separating set D_i^{t+1} for the next-stage influences. Note that D_i^{t+1} typically needs to include certain state factors for stage t, such that x_i^t and D_i^{t+1} both will specify such variables. This is no problem, as long as they specify consistent assignments; we define $\bar{\mathcal{S}}$ to be the set of states that are consistent.
- $\mathcal{A}_i$,$\mathcal{O}_i$ are the (unmodified) sets of actions and observations for agent i.
- The transition function $\bar{T}_i(\bar{s}_i^{t+1}|\bar{s}_i^t,a_i^t)$ which we will discuss in detail shortly.
- The observation function $\bar{O}_i(o_i^{t+1}|a_i^t,\bar{s}_i^{t+1}) = O(o_i^{t+1}|a_i^t,x_i^{t+1})$, since agent i's observations only depend on its local state (cf. Definition 10, property 1).
- The reward function $\bar{R}_i(\bar{s}_i^t,a_i^t,\bar{s}_i^{t+1}) = R_i(x_i^t,a_i^t,x_i^{t+1})$, since agent i's rewards only depend on its local state (cf. Definition 10, property 2).
- H is the unmodified horizon.
- $b_i^{l,0}$ is the initial state distribution, which is a *local-form belief.* It is a distribution over augmented states $\bar{s}_i^0 = \langle x_i^0,D_i^1\rangle$. Since for the first stage D_i^1 can only contain elements from x_i^0, it can trivially be constructed from a probability distribution over x_i^0, and such a distribution can be constructed from b^0, as we discuss in a bit more detail below.

In defining $\bar{T}_i$ and $b_i^{l,0}$, a few subtleties arise that we now discuss.

14. For the last step of this proof, to see why we can swap summation and product, note that the term on the third line has the form $\sum_{\langle a_1,\dots a_k,\dots,a_K\rangle} \prod_{k=1}^K a_k b_k$. We take $K=2$ for the example and get:

$$\sum_{\langle a_1 a_2\rangle} a_1 b_1 a_2 b_2 = \sum_{a_1} a_1 b_1 \sum_{a_2} a_2 b_2 = \left[\sum_{a_1} a_1 b_1\right]\left[\sum_{a_2} a_2 b_2\right] = \prod_{k=1}^K \sum_{a_k} a_k b_k.$$

Transition Probabilities Clearly, the IALM's transition probabilities should express

$$\bar{T}_i(\bar{s}_i^{t+1}|\bar{s}_i^t,a_i^t) \triangleq \Pr(\langle x_i^{t+1},D_i^{t+2}\rangle|\langle x_i^t,D_i^{t+1}\rangle,a_i^t,I_{\rightarrow i}^{t+1}). \tag{4.8}$$

For such probabilities to be specified, we need some further requirements on the d-separating sets. In particular, we require that (the instantiation of) D_i^{t+2} is fully specified by x_i^t,a_i^t,x_i^{t+1} and D_i^{t+1}.

Definition 16 (d-set update function)**.** The d-set update function is a function d that takes the previous-stage d-separating set and the latest transition, and that returns the next d-separating set:

$$D_i^{t+2} = d(x_i^t,a_i^t,x_i^{t+1},D_i^{t+1}).$$

In other words: d 'selects' the variables from $x_i^t,a_i^t,x_i^{t+1},D_i^{t+1}$ such that it forms the next d-separating set.[15]

Given a d-set update function we can write:

$$\Pr(\langle x_i^{t+1},D_i^{t+2}\rangle|\langle x_i^t,D_i^{t+1}\rangle,a_i^t,I_{\rightarrow i}^{t+1}) = \Pr(x_i^{t+1}|\langle x_i^t,D_i^{t+1}\rangle,a_i^t,I_{\rightarrow i}^{t+1})\mathbf{1}_{\{D_i^{t+2},d(x_i^t,a_i^t,x_i^{t+1},D_i^{t+1})\}},$$

where $\mathbf{1}_{\{\cdot,\cdot\}}$ denotes the Kronecker delta function.

A typical way to fulfill the requirement that D_i^{t+2} is fully specified by x_i^t,a_i^t,x_i^{t+1} and D_i^{t+1} is to assume that the d-separating sets for all stages are chosen as the history of the same subset $D_i \subseteq S(i)$ of modeled features.

Example. Looking at Figure 6 on page 810, the d-separating set D_2^3 for predicting f^3 is given by the history of the 'found', 'location of target' and 'location of agent 2' variables. So we can write $D_2 = \{f,l_{tgt},l_2\}$, and define D_2^3 to be its history at stage $t = 2$: $D_2^3 = \vec{D}_2^2$.

The probabilities $\Pr(x_i^{t+1}|\langle x_i^t,D_i^{t+1}\rangle,a_i^t)$ are now factored as the product of the CPTs of the OLAFs and the induced probabilities for the NLAFs:

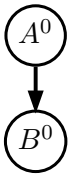
$$\begin{aligned}\bar{T}_i(\bar{s}_i^{t+1}|\bar{s}_i^t,a_i^t) &\triangleq \Pr(x_i^{t+1}|\langle x_i^t,D_i^{t+1}\rangle,a_i^t,I_{\rightarrow i}^{t+1})\mathbf{1}_{\{D_i^{t+2},d(x_i^t,a_i^t,x_i^{t+1},D_i^{t+1})\}},\\ &= \Pr(\tilde{x}_i^{t+1}|\langle x_i^t,D_i^{t+1}\rangle,a_i^t,I_{\rightarrow i}^{t+1})\Pr(\mathring{x}_i^{t+1}|x_i^t,\tilde{x}_i^{t+1},a_i)\mathbf{1}_{\{D_i^{t+2},d(x_i^t,a_i^t,x_i^{t+1},D_i^{t+1})\}}.\end{aligned} \tag{4.9}$$

Here the first term is given by (4.7) and the second term is given by (3.11).[16]

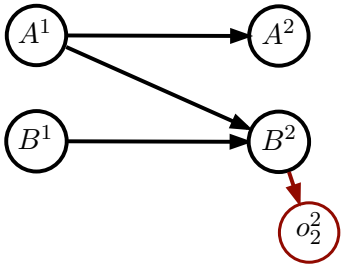
Initial Local State Distribution Here we discuss some of the issues involved in defining the initial belief in the IALM. Note that in a factored models such as fPOSGs, the initial state distribution b^0 is specified as a Bayesian network G^0. Together with the 2DBN, $G^{\rightarrow}$ (which in fact is a conditional probability distribution), it can form the unrolled DBN $G = \text{unroll}(G^0,G^{\rightarrow})$ which specifies the joint distribution over all the state variables, as is

15. Note, that if further compression by means of a statistic σ is employed (cf. the discussion under Definition 11) than the update function should work on these statistics $\sigma(D_i^{t+2}) = d(x_i^t,a_i^t,x_i^{t+1},\sigma(D_i^{t+1}))$.

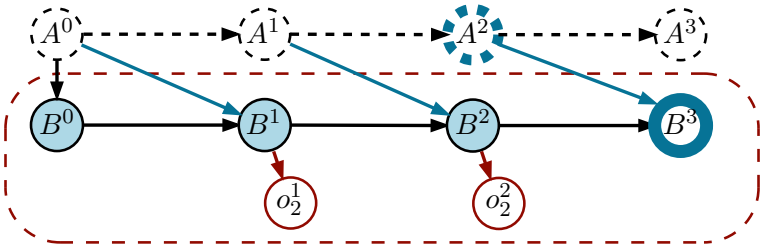
16. Note that, even though we have not dealt with intra-stage dependencies (ISDs) in the description of influences in this section, we refer back to the term $\Pr(\mathring{x}_i^{t+1}|x_i^t,\tilde{x}_i^{t+1},a_i)$ from section 3 which does allow for ISDs from NLAFs to OLAFs. This will allow us to make only minimal changes to the definition of $\bar{T}_i$ when we do deal with ISDs in Section 5.



(a) The Bayesian network G^0 representing the initial belief.



(b) The 2DBN (a conditional Bayesian network) $G^{\rightarrow}$ representing the transition and observation probabilities.



(c) The unrolled network $G = \text{unroll}(G^0, G^{\rightarrow})$. To convert it to an IALM, the local-form initial belief $b_i^{l,0}(B^0)$ and incoming influences $I_{\rightarrow i}^{t+1}(A^t|B^0,\ldots,B^t)$ need to be computed via inference. See text for further explanation.

Figure 8: Construction of the IALM.

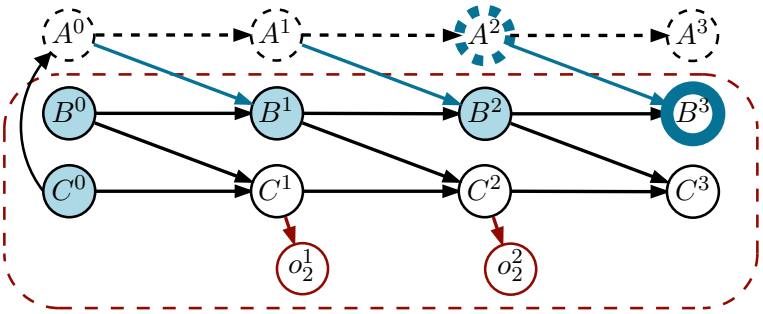


Figure 9: Impact of initial belief connectivity on the d-separating set of the IALM.

illustrated in Figure 8. Note that the figure gives a simplified representation not involving any actions.

Now we will discuss how to specify the initial belief $b_i^{l,0}(x_i^0)$ of the IALM. The basic idea is to simply restrict G^0 to those variables in the set $S(i)$ of agent i's local state variables. However, this can lead to problems when there are arrows in G^0 pointing from variables not included in $S(i)$ to variables included in $S(i)$. For instance, in Figure 8, the initial belief is factored: $b^0(s) = \Pr(A^0)\Pr(B^0|A^0)$. The initial local-form belief, however, should only be specified over B^0. The solution is to marginalize out the dependencies:

$$b_i^{l,0}(B^0) = \sum_{A^0} \Pr(A^0)\Pr(B^0|A^0).$$

This is also gives the general recipe for any other problem: construction of $b_i^{l,0}$ from b^0 is a marginal inference task. Certainly, for certain complex problems this could be intractable, but the hope is that for many real-world problems the prior b^0 is sufficiently sparsely structured for this not to be an issue. Also, any of the vast number of (exact or approximate) inference methods developed in the last decades can be used (Koller & Friedman, 2009; Boyen & Koller, 1998; Jordan, Ghahramani, Jaakkola, & Saul, 1999; Murphy, 2002; Wainwright, Jordan, et al., 2008).

Impact of Correlations of Initial State Factors on the D-separating Set Note that the correlation of the initial state distribution can affect d-separation and therefore what variables need to be included in the d-separating set D_i^t. For instance, if in the above example there additionally is a state factor C, which is not connected to A or B in the 2DBN $G^{\rightarrow}$, but which is a parent of A in G^0, we get the unrolled DBN as shown in Figure 9.

Now, to define the IALM, we will need the induced probability of B^3, which according to (4.7) can be written as

$$\Pr(B^3|\left\langle B^2,D_i^3\right\rangle,I_{\rightarrow i}^3) = \sum_{A^2} I(A^2|D_i^3)\Pr(B^3|B^2,A^2).$$

Therefore D_i^3 needs to contain any variables that can be used to better predict A^2 (more formally, any variables that d-separate A^2 from $\vec{h}_i^t$ and any remaining variables x_i^t, cf. Definition 11). However, looking at the figure, we see that this means that C^0 needs to be included in the d-set.

At the same time, however, we see that we do not need to condition on the entire history $\vec{C}^t$. This may appear counter intuitive, since observations at later time steps (e.g, o_i^1 and o_i^2) certainly provide information about A^2, while they also depend on $\vec{C}^t$. But this is precisely the point: by including C^0 in the d-separating set, it becomes part of the hidden state—e.g., for $t = 2$ we have $\bar{s}_i^2 = \langle x_i^2, D_i^3 \rangle = \langle \langle C^2 \rangle, \langle B^0, B^1, B^2, C^0 \rangle \rangle$—and those later observations certainly provide information as to what that hidden state is.

4.3 Planning in an IALM

Here we look at how we can plan using an IALM. It turns out that this is surprisingly simple, since an IALM *is a* (special case of) POMDP.

Observation. An influence-augmented local model is a POMDP.

Proof. This can simply be verified by comparing Definition 15 to the definition of a POMDP (Definition 1). □

This means that belief updates and definition of value functions follow as usual. For completeness and future reference, we write these out in detail below.

4.3.1 Local-form Belief Update

As implied by Definition 15, in an IALM, an agent uses a *local-form belief*:

Definition 17 (local-form belief)**.** A *local-form belief* $b_i^{l,t}$ for an IALM constructed for agent i is the posterior probability distribution over augmented states $\bar{s}_i^t = \langle x_i^t, D_i^{t+1} \rangle$.

The belief update for such a local-form belief is as in a regular POMDP, cf. (2.1):

$$BU(b_i^l, a_i^t, o_i^{t+1})(\bar{s}_i^{t+1}) = \frac{1}{\Pr(o_i^{t+1}|b_i^l, a_i^t)} \bar{O}_i(o_i^{t+1}|a_i^t, \bar{s}_i^{t+1}) \sum_{\bar{s}_i^t} \bar{T}_i(\bar{s}_i^{t+1}|\bar{s}_i^t, a_i^t) b_i^l(\bar{s}_i^t) =$$

$$\frac{O(o_i^{t+1}|a_i^t, x_i^{t+1})}{\Pr(o_i^{t+1}|b_i^l, a_i^t)} \sum_{x_i^t, D_i^{t+1}} \Pr(x_i^{t+1}|\langle x_i^t, D_i^{t+1} \rangle, a_i^t, I_{\rightarrow i}^{t+1}) \mathbf{1}_{\{D_i^{t+2}, d(x_i^t, a_i^t, x_i^{t+1}, D_i^{t+1})\}} b_i^l(x_i^t, D_i^{t+1}) \tag{4.10}$$

The expected observation probability (the normalization factor) in this case can be shown (see the derivation in Appendix A.3.1) to satisfy

$$\begin{aligned} \Pr(o_i^{t+1}|b_i^l, a_i^t) &= \mathbf{E}_{\bar{s}_i^t \sim b_i^l, \bar{s}_i^{t+1} \sim \bar{T}(\bar{s}_i^t, a_i^t, \cdot)} \left[\bar{O}(o_i^{t+1}|a_i^t, \bar{s}_i^{t+1}) \right] \\ &= \mathbf{E}_{\langle x_i^t, D_i^{t+1} \rangle \sim b_i^l, \langle x_i^{t+1}, D_i^{t+2} \rangle \sim \bar{T}(\langle x_i^t, D_i^{t+1} \rangle, a_i^t, \cdot)} \left[O(o_i^{t+1}|a_i^t, x_i^{t+1}) \right] \\ &= \sum_{x_i^{t+1}} O(o_i^{t+1}|a_i^t, x_i^{t+1}) \Pr(x_i^{t+1}|b_i^l, a_i^t, I_{\rightarrow i}^{t+1}), \end{aligned} \tag{4.11}$$

with

$$\Pr(x_i^{t+1}|b_i^l, a_i^t, I_{\rightarrow i}^{t+1}) \triangleq \sum_{x_i^t, D_i^{t+1}} \Pr(x_i^{t+1}|\langle x_i^t, D_i^{t+1} \rangle, a_i^t, I_{\rightarrow i}^{t+1}) b_i^l(x_i^t, D_i^{t+1}). \tag{4.12}$$

4.3.2 IALM Value

Putting everything together, we can show that for an IALM, the value function is similar to the normal POMDP value function:

Proposition 2 (IALM value function)**.** *The value function is given by*

$$Q_i^t(b_i^l,a_i^t) = R_i(b_i^l,a_i^t) + \gamma \sum_{o_i^{t+1}} \Pr(o_i^{t+1}|b_i^l,a_i^t) V_i^{t+1}(BU(b_i^l,a_i^t,o_i^{t+1})), \tag{4.13}$$

$$V_i^{t+1}(b_i^l) = \max_{a_i} Q_i^{t+1}(b_i^l,a_i), \tag{4.14}$$

where

$$\begin{aligned} R_i(b_i^l,a_i^t) &= \mathbf{E}_{\bar{s}_i^t \sim b_i^l, \bar{s}_i^{t+1} \sim \bar{T}(\bar{s}_i^t,a_i^t,\cdot)} \left[\bar{R}_i(\bar{s}_i^t,a_i^t,\bar{s}_i^{t+1})\right] \\ &= \sum_{x_i^t} \sum_{x_i^{t+1}} R_i(x_i^t,a_i^t,x_i^{t+1}) \Pr(x_i^t,x_i^{t+1}|b_i^l,a_i^t,I_{\to i}^{t+1}) \end{aligned} \tag{4.15}$$

with

$$\Pr(x_i^t,x_i^{t+1}|b_i^l,a_i^t,I_{\to i}^{t+1}) \triangleq \sum_{D_i^{t+1}} \Pr(x_i^{t+1}|\langle x_i^t,D_i^{t+1}\rangle,a_i^t,I_{\to i}^{t+1}) b_i^l(x_i^t,D_i^{t+1}). \tag{4.16}$$

Proof. This follows from the value function of regular POMDPs together with the derivations of $R_i(b_i^l,a_i^t)$ and $\Pr(x_i^t,x_i^{t+1}|b_i^l,a_i^t,I_{\to i}^{t+1})$ in Appendix A.3.2. □

The solution of the IALM gives the influence-based best-response value, defined as the value of the initial local-form belief:

$$V_i(I_{\to i}(\pi_{-i})) \triangleq V_i^0(b_i^{l,0}). \tag{4.17}$$

4.4 IBA by Example

To further clarify the process of influence-based abstraction, and provide some intuition of the potential computational savings and in which cases they could arise, we discuss two examples in some more detail.

The Planetary Exploration Domain. First, let us consider the planetary rover domain illustrated in Figure 7. We will give a characterization of this problem in terms of number of states, for both the global-form best-response model (GFBRM) and IALM, thus providing an analysis of the computational savings that can arise in this case.

We will use the following notations and assumptions:

- L is the number of locations
- $pl \in \{yes,no\}$ indicates if a plan has been sent to the rover.
- B is the number of private states bt_1 of the satellite (e.g., number of battery levels).
- $|\mathcal{O}_1|$ is the size of the observation set of agent 1. In case that $o_1 = bt_1$, i.e., the satellite can perfectly observe its battery level, we would have $|\mathcal{O}_1| = B$.

- $a_1 \in \{NOOP, PLAN\}$ is the action of the satellite.

So now, in the GFBRM model, the augmented state for the rover (who is agent 2) is $\bar{s}_2^t = \langle s^t, \vec{h}_1^t \rangle = \langle bt_1^t, pl^t, l_2^t, \vec{h}_1^t \rangle$. Given the above assumptions, the number of AOHs for the satellite at stage t is $|\vec{\mathcal{H}}_1^t| = (2|\mathcal{O}_1|)^t$. And therefore the state space of the GFBRM is of size $\left|\bar{\mathcal{S}}_i^t\right| = 2LB \cdot (2|\mathcal{O}_1|)^t$.

In contrast, in an IALM, we have states $\bar{s}_2^t = \langle x_2^t, D_2^{t+1} \rangle = \left\langle \left\langle pl^t, l_2^t \right\rangle, pl^{0:t} \right\rangle = \left\langle l_2^t, pl^{0:t} \right\rangle$, meaning that the size of the state space in the IALM is $L \cdot 2^{t+1}$, which is strictly smaller than the GFBRM model.

Moreover, we can exploit that fact that pl can only turn on, meaning that $pl^{0:t}$ has only $t+2$ possible values: it got turned on on one of the stages $0 \ldots t$ or it has not yet been turned on ("$NotYet$"). We refer to this re-coded variable as $PlanIssueTime^t = \sigma(pl^{0:t})$ such that we can write $\bar{s}_i^t = \left\langle l_2, PlanIssueTime^t \right\rangle$. This means that the number of IALM states at stage t in the planetary exploration problem can be further reduced to $L(t+2)$. This suggests that the IALM will be much cheaper to solve for larger horizons: it scales linearly with t, whereas the GFBRM scales exponentially with t.

However, our discussion so far has excluded the time it takes to construct the IALM. Specifically, to compute the transition probabilities for every stage t, we will need to compute the incoming influence:

$$I(u_{\rightarrow 1}^{t+1} | D_1^{t+1}) = \Pr(a_1^t | PlanIssueTime^t).$$

In general, we would need to compute this for all possible instantiations of D_2^{t+1}. However, in this case, the action of the satellite agent is only relevant in when $PlanIssueTime^t = NotYet$, which mean that we only need to compute the probability $\Pr(a_1^t | PlanIssueTime^t = NotYet)$. Applying (4.4) yields (we leave b^0, π_{-i} implicit):

$$\Pr(a_1^t | PlanIssueTime^t) = \sum_{\vec{h}_1^t} \pi_1(a_1^t | \vec{h}_1^t) \Pr(\vec{h}_1^t | PlanIssueTime^t),$$

which shows that if we have $\Pr(\vec{h}_1^t | PlanIssueTime^t)$ for each stage t, we can directly derive $\Pr(a_1^t | PlanIssueTime^t)$. The main issue therefore is to compute $\Pr(\vec{h}_1^t | PlanIssueTime^t)$ for all $t = 1 \ldots h-1$. This is essentially a belief tracking problem. Specifically, we can model this as a special type of hidden Markov model where the hidden state is $\langle bt_1^t, \vec{h}_1^t \rangle$, and our observations are pl^t. The number of such states at stage t is $B \cdot (2|\mathcal{O}_1|)^t$ and that also is the dominant term in the complexity.

This shows that in this example, computing a best-response using a GFBRM requires to solve a POMDP with $\left|\mathcal{S}^{GFBR}\right| = 2LB \cdot (2|\mathcal{O}_1|)^t$, while doing it using an IALM requires solving a POMDP with $\left|\mathcal{S}^{IALM}\right| = L(t+2)$ states and a construction cost of $O\left(B \cdot (2|\mathcal{O}_1|)^t\right)$. This means that particularly when also the number L of locations is considerable, we can tackle much larger problems, since we have isolated the exponential complexity of tracking $\vec{h}_1^t$ from the impact of the number of locations L. Similarly, in the case where B (in general the number of states induced by non-modeled factors) is very large, this cost now only appears in the IALM construction and is not multiplied with L.[17]

17. Of course, the reader could wonder in how far these terms are relevant, given the remaining exponential dependence on the horizon via the cost of tracking $\vec{h}_1^t$. To answer this, let us point out that this

The House Search Domain Next, we turn to the House Search problem, illustrated in Figure 6. In contrast to Planetary Exploration, House Search exhibits a more substantial d-separating set D_2^{t+1} and so we expect less computational savings. In fact, as we will see below, the IALM provides little to no computational benefit *except* in the face of additional assumptions on the structure of the problem.

Let us again define the sizes of the relevant quantities:

- L is the number of locations.
- $f \in \{yes,no\}$ indicates if the target has been found. Once a target has been found the location l_1 of agent 1 no longer has any effect.
- $|\mathcal{O}_1|$ is the size of the observation set of agent 1. In case that $o_1 = \langle l_1,f \rangle$, i.e., the agent can perfectly observe its location and if the target is found, we would have $|\mathcal{O}_1| = 2L$.
- $\mathcal{A}_1 = \mathcal{A}_2$ are the action sets that can allow the agents to move to adjacent rooms.

Repeating the analysis, we see that the GFBRM state is $\bar{s}_2^t = \langle s^t,\vec{h}_1^t \rangle = \left\langle \left\langle l_1^t,f^t,l_{tgt}^t,l_2^t \right\rangle ,\vec{h}_1^t \right\rangle$. Since $|\vec{\mathcal{H}}_1^t| = \left(|\mathcal{A}_1|\,|\mathcal{O}_1|\right)^t$, the state space of the GFBRM is of size $\left|\bar{\mathcal{S}}^t\right| = 2L^3\left(|\mathcal{A}_1|\,|\mathcal{O}_1|\right)^t$. If we assume the agent has 4 movement actions and can perfectly observe it location and if the target is found, as above, this becomes $2L^3\left(4 \cdot 2L\right)^t = 2L^3\left(8L\right)^t = 2^{3t+1}L^{t+3}$.

On the other hand, the IALM has states $\bar{s}_2^t = \langle x_2^t,D_2^{t+1} \rangle = \left\langle f^{0:t},l_{tgt}^{0:t},l_2^{0:t} \right\rangle$. Therefore, without further simplifications, the size of the state space at stage t in the IALM is $2^{t+1} \cdot L^{2(t+1)}$. In other words, even disregarding the construction costs of the IALM, this would only guaranteed to be smaller for time step $t = 1$, as illustrated in Table 1.

Simplifications are possible, however: as before, the 'found' variable f may only turn on which reduces the IALM state to $\bar{s}_2^t = \left\langle foundTime^t,l_{tgt}^{0:t},l_2^{0:t} \right\rangle$ and the state space size to $(t+2) \cdot L^{2(t+1)}$. When the target is stationary, this reduces to $(t+2) \cdot L \cdot L^{(t+1)}$. Also, it may not be realistic that all sequences of locations are realizable. Given a fixed start position and 4 deterministic movement actions, the number of realizable sequences of locations would be $O(4^t)$,which would lead to a 'simplified IALM' with state space of size $(t+2) \cdot L \cdot 4^t$. Exploiting the realizable location sequences of agent 1 to similarly reduce the state space of the GFBRM leads to $2 \cdot L^3 \cdot (t+2) \cdot 4^t$ states. As such, the IALM representation in terms of $\bar{s}_2^t = \langle x_2^t,D_2^{t+1} \rangle$ enables us to capture structure of the problem to significantly reduce its local state space.

Of course, we did not yet cover the cost of the construction cost of the IALM by computing the influence. Here, we sketch what is involved in the construction of the 'simplified IALM' we constructed. Specifically the influence specification now is:

$$I(u_{\rightarrow 1}^{t+1}|D_1^{t+1}) = \Pr(l_1^t|\left\langle FoundTime^t,l_{tgt},l_2^{0:t-1} \right\rangle).$$

exponential dependence is directly the consequence of the fact that in this paper we have not restricted the class of policies considered for other agents, but assumed these are general mappings from AOHs to actions. However, this problem is inherently complex. In fact, unless a particular compact description is available, the size of the policy of the satellite π_1, i.e., *the size of the input* (tabular representations of the π_1) of the best-response problem, is exponential in the horizon. However, in cases where the policy π_1 of the other agent has a compact representations (e.g., a finite state controller with K states) it may be possible to substantially reduce the cost of tracking the other agent's internal state (e.g., $\vec{h}_1^t$ is replaced by agent 1's controller node and tracking can be done in time $O(B \cdot K)$.

model	state space size	stage t 1	2	3	4
GFBRM	$2^{3t+1}L^{t+3}$	$16L^4$	$128L^5$	$1024L^6$	$8192L^7$
GFBRM simplified	$2 \cdot L^3 \cdot (t+2) \cdot 4^t$	$24L^3$	$128L^3$	$640L^3$	$3072L^3$
IALM naive	$2^{t+1} \cdot L^{2(t+1)}$	$4L^4$	$8L^6$	$16L^8$	$32L^{10}$
IALM simplified	$L \cdot (t+2) \cdot 4^t$	$12L$	$64L$	$320L$	$1536L$

Table 1: State space sizes for GFBRM and two versions of IALM models for the House Search problem.

As before, we only care about cases where $FoundTime^t = NotYet$ (since otherwise the location l_1^t is irrelevant). However, the different options for l_{tgt},$l_2^{0:t-1}$ should be evaluated, which means that we need to solve the inference problem for $O(L \cdot 4^{t-1})$ instantiations of the d-set. Each of these instantiations requires tracking hidden states of the form $\left\langle l_1^t, \vec{h}_1^t \right\rangle$, and there are $L\left(|\mathcal{A}_1||\mathcal{O}_1|\right)^t$ of them in general. For the simplified setting of deterministic moves and perfect observations by agent 1 this can be limited to $L \cdot t \cdot 4^t$ since in that case $\vec{h}_1^t = \left\langle f^{0:t}, l_1^{0:t} \right\rangle = \left\langle FoundTime^t, l_1^{0:t} \right\rangle$.

In summary, the simplified version of the House Search problem enables us to reduce the state space of the best-response model substantially, from $2^{3t+1}L^{t+3}$ to $L \cdot (t+2) \cdot 4^t$. However to construct the IALM state space at stage t still requires time of the order $(L \cdot 4^{t-1})(L \cdot t \cdot 4^t) = L^2 \cdot t \cdot 4^{2t-1}$.

4.5 More General Implications of IBA

In the above, we saw that IBA can lead to more efficient computation of exact best-responses in settings that have sufficient structure to exploit. However, our motivation for developing the theory presented in this paper is more general than this. Here we elaborate on the broader implications that we envision.

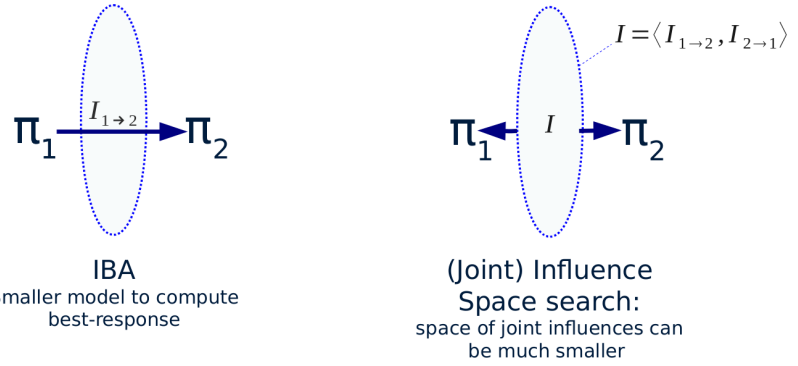


Figure 10: Illustration of the ideas of influence-based abstraction and influence search.

4.5.1 Influence Search

The ideas underlying influence-based abstraction were developed in the research community focusing on multiagent sequential decision making by people like Becker et al. (2003), Varakantham et al. (2009) and Petrik and Zilberstein (2009). The goal that these works pursued were not the computation of merely a best response, but of the optimal *joint* policy. Hence these works performed a type of *influence search* (Witwicki & Durfee, 2010b). The idea, illustrated in Figure 10, is that many policies of one agent, say agent 1, may correspond to the same influence $I_{1\to 2}$ on agent 2, which would mean that the set of such influences could be much smaller than the set of possible policies π_2. Therefore, if it is possible to search through the space of *joint influences* $I = \langle I_{1\to 2}, I_{2\to 1}\rangle$, this could be much more effective than searching the much larger space of joint policies $\pi = \langle \pi_1, \pi_2\rangle$. Specifically, Witwicki et al. (2012) showed orders of improvement in computational cost over joint policy-search approaches.

So far, however, these ideas have only been exploited in sub-classes of fDec-POMDPs (see also Section 7), and generalizing influence search to general fDec-POMDPs or even fPOSG (i.e., to find Nash equilibria) is still an open problem. Our definition of influence in Definition 12 can serve as a starting point for such extensions.

4.5.2 Approximate Influence Representations

The discussion in Section 4.4 demonstrated that, in cases with sufficient structure, the representation of the influence $I(u_{\to i}^{t+1}|D_i^{t+1})$ can be compact, leading to substantial benefits. However, at the same time it also showed that in general, without exploiting special properties of the domain, these representations can become very big and unwieldy due to the dependence on the history of a subset of variables. Large influence representations can not be exploited for more efficient best-response computations, and they also suggest that the number of possible influences will be large, possibly limiting the effectiveness of influence search.

However, even though exact representations of $I(u_{\to i}^{t+1}|D_i^{t+1})$ may be very large, it might be the case that approximate representations $\hat{I}(u_{\to i}^{t+1}|D_i^{t+1})$ can be compactly represented while still affording good performance; for the purposes of making good predictions it is usually not required to remember the full history (Littman, 1994; McCallum, 1995; Kaelbling et al., 1998; Meuleau, Peshkin, Kim, & Kaelbling, 1999). Moreover, learning such approximate influence points is a supervised learning problem (a specific type of sequence prediction problem), which means that we can directly build on recent advancements for such prediction problems, including work on *deep learning* (Schmidhuber, 2015; LeCun, Bengio, & Hinton, 2015). Of course, whether the successes from natural language processing (Vinyals, Toshev, Bengio, & Erhan, 2015; Young, Hazarika, Poria, & Cambria, 2018), machine translation (Cho, van Merriënboer, Gulcehre, Bahdanau, Bougares, Schwenk, & Bengio, 2014), speech recognition (Graves, Mohamed, & Hinton, 2013; Weninger, Erdogan, Watanabe, Vincent, Le Roux, Hershey, & Schuller, 2015) or biological sequence data (Jurtz, Johansen, Nielsen, Almagro Armenteros, Nielsen, Sønderby, Winther, & Sønderby, 2017) will transfer to the task of influence prediction remains to be investigated, but there already are some positive indications.

Specifically, some studies have shown that approximate representations of influence can enable further scalability in a variety of respects. For instance, Oliehoek, Whiteson, and Spaan (2013) introduced the idea of *transfer planning,* which defines a number of smaller source tasks, whose solution is transferred to the larger (involving more agents) target task. The definition of the source tasks ignores the actual influence of the rest of the system, and hence can be seen as a very naive special case of approximate influence-based abstraction: it assumes an arbitrary influence point for each sub-problem. Nevertheless, the authors empirically showed that this can lead to good performance in Dec-POMDPs with many agents. This was corroborated by Oliehoek et al. (2015a) who employed *optimistic influences* (also an approximate form of influence) to compute factored upper bounds on the Dec-POMDP value function. They demonstrated that in some cases the solution found by transfer planning for Dec-POMDPs with over 50 agents was essentially optimal. Recently, He, Suau, and Oliehoek (2020) demonstrated that, by using learned (recurrent neural network) representations of influence in online planning, it is possible to get better task performance when the time for action selection is limited. Of course, giving hard performance guarantees for such approaches is very difficult, but Congeduti et al. (2020) show that it is possible to derive performance loss bounds for approximate influence representations. Their analysis also suggests that typical machine learning approaches that minimize the cross-entropy loss may be well aligned with minimizing the performance loss.

As such, there is substantial evidence that approximate extension of the formal IBA framework presented in this paper can lead to various benefits. Related to this is the new perspective these approaches give on the systems they aim to control. For instance, both Oliehoek et al. (2015a) and He et al. (2020) experimented with forms of "influence strength" to better understand parametrized domains by looking at the impact on the resulting solution quality. Further formalization and refinements of such notions could lead to a better understanding of the application of decision making methods to complex domains.

4.5.3 Identifying Inductive Biases

Notions like influence strength can enable us to better understand the problems that we are trying to tackle, and the IBA perspective can generate more of such insights. For instance, the discussion on the impact of the initial state distribution on page 816 neatly exemplifies some different types of structure we can expect to encounter when dealing with abstraction in structured decision making processes.

Identification of such structure is important even for deep learning: even though the representations are learned automatically, no learning methods are effective without the appropriate inductive biases (Mitchell, 1980; Wolpert, 1996). For instance, convolutional neural networks are so successful for image processing because they exploit the fact that there is local and repeated structure in real-world images. In a similar way, in the discussion on the initial state distribution, we noticed that certain forms of structure, such as dependence on certain state factors at stage $t = 0$, might be common in sequential decision processes involving abstraction.

In fact, recent research provides clear evidence that structure as implied by influence-based abstraction can be effectively used to bias deep reinforcement learning (Suau de

Castro, Congeduti, Starre, Czechowski, & Oliehoek, 2019a). Specifically, that work shows that by equipping a policy and/or value network with a recurrent sub-network that is only fed with a subset of variables (roughly corresponding to the d-separating set) can lead to higher performance than feed-forward networks, while learning much faster than a full-sized recurrent neural network. Further connections to deep RL and multiagent RL approaches are discussed in Section 8.

5. IBA With Intra-Stage Dependencies

The previous section presented the framework of influence-based abstraction, which enables us to abstract away hidden state variables in so-called local-form models. We illustrated how this can lead to speeding up best-response computations and discussed more general implications of the theory. So far we assumed there are no intra-stage connections: all influence links span a time step. However, intra-stage dependencies (ISDs) can be useful to specify a more intuitive model, as we saw for House Search in Figure 2a. Additionally, there could be problems that only have a correct formulation using intra-stage dependencies: since intra-stage connections can model additional correlations, the transition functions $T(s^{t+1}|s^t,a^t)$ that can be represented without ISDs is a strict subset of those we can represent with ISDs. Simply removing ISDs from problems that need them is not possible, as it is not clear what probabilities the CPTs should specify for the remaining parents.

Moreover, intra-stage connections enable us to introduce 'dummy' variables, as discussed in Section 4.1.1. Without this capability, the requirement of including all observation-relevant and reward-relevant variables in the local state (cf. Definition 10) would limit the applicability of influence-based abstraction. For instance, imagine the setting where our agent's reward is directly affected by how many other agents take the some action a. Without intra-stage connections, we would be forced to model all the action variables of the other agents in the local state, making the local model intractable. However, using intra-stage connections, we can instead introduce a count variable that affects our reward, while we do not model (abstract away) all the individual actions of other agents. As such, the ability to use ISDs can allow us to effectively describe scenarios with anonymous interactions, such as mean-field games and others (Jovanovic & Rosenthal, 1988; Kizilkale & Caines, 2012; Varakantham, Adulyasak, & Jaillet, 2014; Robbel, Oliehoek, & Kochenderfer, 2016; Nguyen, Kumar, & Lau, 2017; Subramanian & Mahajan, 2019), in the IBA framework.

Therefore, this section extends our definition of influence to also be applicable for models that have such *intra-stage dependencies (ISDs)*.

5.1 Definition of Influence under ISDs

Here we extend IBA by adapting notions of influence sources, d-separating sets, and incoming influence points to properly take into account ISDs.

5.1.1 Intra-Stage Influence Sources

In settings with intra-stage dependencies, there is at least one non-modeled factor y^{t+1} that influences an NLAF $\tilde{x}^{t+1}$. If there are multiple such factors, we let y_u^{t+1} denote them. Therefore, in order to perform IBA in settings with ISDs, we will need to predict influence

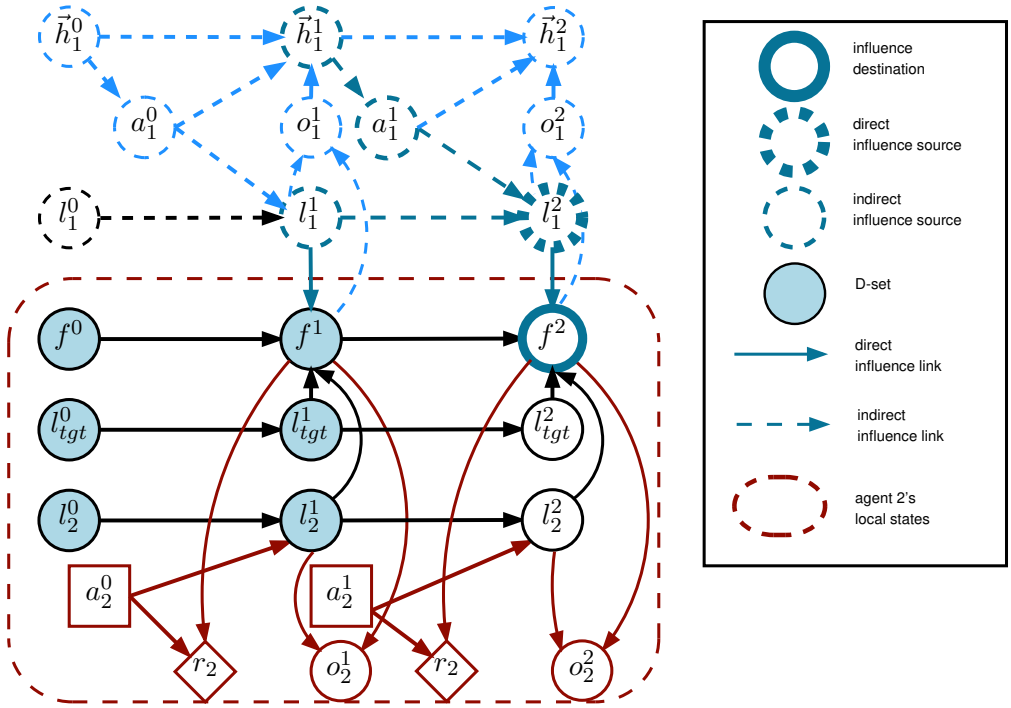


Figure 11: Illustration of the influence experienced by protagonist agent $i = 2$ in the intra-stage version of House Search at stage $t = 2$. f^2 is the influence destination, with *direct* influence source $y_u^2 = \langle l_1^2 \rangle$ (i.e., $u_{\rightarrow i}^2 = \langle l_1^2 \rangle$). Additionally, the figure highlights the *indirect* influence sources $y_v^1 = \langle l_1^1 \rangle$, $a_v^t = \langle a_1^1 \rangle$ and $\vec{h}_v^1 = \left\langle \vec{h}_1^1 \right\rangle$, which determine the influence that is exerted at stage $t = 1$. (Note that l_1^1 in fact is also a *direct* influence source for the influence *experienced at* stage $t = 1$.)

sources $u_{\rightarrow i}^{t+1} = \langle y_u^t, a_u^t, y_u^{t+1} \rangle$. In order to correctly deal with the intra-stage sources y_u^{t+1}, we will additionally need to consider those variables that influence *them*.

Indirect Sources In particular, we use 'v' as the symbol to denote such 'indirect' or 'second order' influences and will write $x_v^t, y_v^t, a_i^t, a_v^t, x_v^{t+1}$ and y_v^{t+1} for the possible[18] ancestors in the 2DBN of intra-stage sources y_u^{t+1}.

Example. Figure 11 illustrates the direct and indirect influence sources for House Search. In order to be able to make accurate predictions of the influence destination f^2, at stage $t = 1$ we should be able to predict l_1^1 (y_v^1) and a_1^1 (a_v^t) as accurately as possible. Given that we assume access to the policy of agent 1, we can equivalently predict $y_v^1 = l_1^1, \vec{h}_v^1 = \vec{h}_1^1$.

Now, in order to define the influence, we will need to consider the probability of such y_u^{t+1} given variables that we know how to predict at stage t. In general it is given by:

$$\Pr(y_u^{t+1} | x_v^t, y_v^t, a_i^t, a_v^t, x_v^{t+1}) = \sum_{y_v^{t+1}} \Pr(y_u^{t+1}, y_v^{t+1} | x_v^t, y_v^t, a_i^t, a_v^t, x_v^{t+1}), \tag{5.1}$$

18. Of course, in any given problem not all of these types of variables are relevant. For instance, if there is no action a_j^t of another agent j that would influence an ISD influence source, then a_v^t can be removed from the equations.

where:

- $\Pr(y_u^{t+1},y_v^{t+1}|x_v^t,y_v^t,a_i^t,a_v^t,x_v^{t+1})$ is the product of CPTs of (both direct and indirect) intra-stage sources—in Figure 11 this is simply $\Pr(l_1^2|l_1^1,a_1^1)$,
- x_v^t are those state factors at stage t ("in the left-hand slice of the 2DBN") that are modeled by agent i, and are ancestor to an intra-stage influence source of agent i at stage $t+1$ ("in the right-hand slice of the 2DBN")—in Figure 11 no such variables exist,
- y_v^t are those state factors in the left-hand slice of the 2DBN that are not modeled by agent i, but are ancestor to an influence destination of agent i—in Figure 11 this is l_1^1,
- x_v^{t+1}, y_v^{t+1} are the modeled, respectively non-modeled state factors at state $t+1$ that are ancestors to an intra-stage influence source—in Figure 11 no such variables exist,
- a_i^t might directly or indirectly affect an an intra-stage influence source, in which case it needs to be included in (5.1)—in Figure 11 this is not the case, and
- a_v^t are the actions of other agents that are ancestors of an intra-stage influence source—in Figure 11 this is a_1^1.

We will also write $\vec{h}_v^t$ for the AOHs of the agents v that correspond to a_v^t (i.e., those agents of which the action is an ancestor in the 2DBN of an influence destination of agent i).

All sources So far we have introduced notation using u for direct sources and using v for indirect sources. We will also want to consider the union of direct and indirect sources, and for these purposes we will write w. For example, we will write $a_w^t = \langle a_u^t,a_v^t \rangle$ for the actions of agents that either directly or indirectly influence an influence destination.

5.1.2 The D-Separating Set

We now build on this insight to define the d-separating set in problems with intra-stage dependencies:

Definition 18 (d-separating set)**.** The *d-separating set for agent i*, D_i, is a subset of variables (state factors and/or actions), such that the history of these variables d-separates $y_w^t,\vec{h}_w^t$ from $x_i^t,\vec{h}_i^t$. I.e., it is defined in such a way that

$$\forall_{y_w^t,\vec{h}_w^t} \qquad \Pr(y_w^t,\vec{h}_w^t|x_i^t,\vec{h}_i^t,D_i^{t+1},b^0,\pi_{-i}) = \Pr(y_w^t,\vec{h}_w^t|D_i^{t+1},b^0,\pi_{-i}). \tag{5.2}$$

As before this should be interpreted to mean: D_i^{t+1} d-separates $y_w^t,\vec{h}_w^t$ from those parts of $x_i^t,\vec{h}_i^t$ (i.e, of the local model) not contained in D_i^{t+1}.

Comparing Definition 18 with the earlier Definition 11, we see they are pleasingly similar; all that changed is that u's have been replaced with w's to now take into account the possibility of indirect sources.

5.1.3 Definition of Influence under ISDs

With this as background, we are now in position to define the concept of influence in all its generality:

Definition 19 (Experienced Influence under ISDs)**.** The *influence experienced by agent i at stage $t+1$* is a conditional probability distribution over the direct influence sources:

$$I(u_{\rightarrow i}^{t+1}|D_i^{t+1},x_v^t,a_i^t,x_v^{t+1}) \triangleq \Pr(\langle y_u^t,a_u^t,y_u^{t+1}\rangle\,|D_i^{t+1},x_v^t,a_i^t,x_v^{t+1},b^0,\pi_{-i})$$
$$= \sum_{\langle y_v^t,a_v^t,y_v^{t+1}\rangle} \Pr(y_u^{t+1},y_v^{t+1}|x_v^t,y_v^t,a_i^t,a_v^t,x_v^{t+1}) \sum_{\vec{h}_w^t} \pi_w(a_w^t|\vec{h}_w^t)\Pr(y_w^t,\vec{h}_w^t|D_i^{t+1},b^0,\pi_{-i}) \quad (5.3)$$

where

- u denote (direct) influence sources;
- v denote the (indirect) 'second order' sources;
- w (as above) denotes the union of u and v;
- $\Pr(y_u^{t+1},y_v^{t+1}|x_v^t,y_v^t,a_i^t,a_v^t,x_v^{t+1})$ is the term necessary to predict the intra-stage sources. It is a term that consists of the product of CPTs;
- $\pi_w(a_w^t|\vec{h}_w^t) = \prod_{i\in w}\pi_i(a_i^t|\vec{h}_i^t) = \pi_u(a_u^t|\vec{h}_u^t)\pi_v(a_v^t|\vec{h}_v^t)$ is the product of action probabilities according to the policies of the other agents that are relevant directly (the u) or indirectly for intra-stage sources (the v); and
- $\Pr(y_w^t,\vec{h}_w^t|D_i^{t+1},b^0,\pi_{-i}) = \Pr(y_u^t,y_v^t,\vec{h}_u^t,\vec{h}_v^t|D_i^{t+1},b^0,\pi_{-i})$ predicts the non-modeled factors that are relevant directly (the u) or indirectly for intra-stage sources (the v), as well as the histories for the relevant agents.

Tying back to the example of Figure 11, (5.3) reduces to

$$I(l_1^2|D_2^2) = \sum_{l_1^1,a_1^1}\Pr(l_1^2|l_1^1,a_1^1)\sum_{\vec{h}_1^1}\pi_1(a_1^1|\vec{h}_1^1)\Pr(l_1^1,\vec{h}_1^1|D_1^2,b^0,\pi_1).$$

We use $I_{\rightarrow i}^{t+1}(\pi_{-i})$ to denote the conditional distribution $I(\cdot|D_i^{t+1},x_i^t,a_i^t,x_i^{t+1})$.

We make a few observations:

- The term $\Pr(y_u^{t+1},y_v^{t+1}|x_v^t,y_v^t,a_i^t,a_v^t,x_v^{t+1})$ can be simplified as given by (5.1), but it is important to keep in mind that this resulting term requires actual inference and is not the product of CPTs anymore.
- Note that, in many cases, we will consider other agents that use deterministic policies, however, we chose to give the more general description that also allows for stochastic policies. In case of deterministic policies, the summation over a_v^t can be omitted, $a_{v/w}^t$ can be replaced by $\pi_{v/w}^t(\vec{o}_{v/w}^t)$, and $\vec{h}_w^t$ becomes $\vec{o}_w^t$ (Oliehoek, 2012).
- The dependence of $I(u_{\rightarrow i}^{t+1}|D_i^{t+1},x_v^t,a_i^t,x_v^{t+1})$ on a_i^t is only needed when a_i^t is an indirect source (i.e., it is an ancestor of y_u^{t+1} or y_v^{t+1}).

5.1.4 Exerted vs. Experienced Influence

Here we make a reinterpretation of the experienced influence at stage $t+1$ as the result of the influence exerted at stage t plus the effect of the intra-stage effects. While this does not fundamentally change anything about the definition of influence per Definition 19, it may

provide some insight on the nature with which influence manifests itself in settings with intra-stage connections, and provide guidance for possible implementations.

In particular, it is possible to define a distribution, only in terms of variables at stage t, which acts as a sufficient statistic to predict the intra-stage source. The intuition is that the *experienced influence,* can be thought of as being induced by the *exerted influence:*

- **Exerted Influence (at stage t):**

$$\Pr(y_w^t,a_w^t|D_i^{t+1},b^0,\pi_{-i}) = \Pr(y_u^t,y_v^t,a_u^t,a_v^t|D_i^{t+1},b^0,\pi_{-i}) \\ = \sum_{\vec{h}_w^t} \pi_w(a_w^t|\vec{h}_w^t)\Pr(y_w^t,\vec{h}_w^t|D_i^{t+1},b^0,\pi_{-i}). \quad (5.4)$$

- **Experienced Influence (at $t+1$):**

$$I(u_{\to i}^{t+1}|D_i^{t+1},x_v^t,a_i^t,x_v^{t+1}) = \Pr(y_u^t,a_u^t,y_u^{t+1}|D_i^{t+1},x_v^t,x_v^{t+1},b^0,\pi_{-i}) \\ = \sum_{\langle y_v^t,a_v^t,y_v^{t+1}\rangle} \Pr(y_u^{t+1},y_v^{t+1}|x_v^t,y_v^t,a_i^t,a_v^t,x_v^{t+1})\Pr(y_w^t,a_w^t|D_i^{t+1},b^0,\pi_{-i}). \quad (5.5)$$

This last equation (5.5) clearly demonstrates how the experienced influence is induced by the exerted influence. The notion of exerted influence (5.4) lays a clear link to IBA in settings without ISDs (cf. Equation 4.4) and is conceptually useful since it isolates which information needs to be retained for each stage t. As such, we expect that any practical implementations for computing the influence by means of filtering (belief tracking) (Russell & Norvig, 2009; Thrun, Burgard, & Fox, 2005) would use this as the primary quantity of interest.

5.2 Influence-Augmented Local Model (IALM)

Here we define the influence-augmented local model under intra-stage connections. Looking at Definition 15, we can conclude that the only changes that we need to make involve the transition function (4.9).

In particular, we need to deal with the fact our definition of influence (5.3) can now be of the more complex form $I(u_{\to i}^{t+1}|D_i^{t+1},x_v^t,a_i^t,x_v^{t+1})$, as given by (5.5). This means that the NLAF probability $\Pr(\tilde{x}_i^{t+1}|\langle x_i^t,D_i^{t+1}\rangle,a_i^t,I_{\to i}^{t+1})$ given by (4.7) must be updated to deal with this new form, and this in turn implies that the definition of $\bar{T}_i(\bar{s}_i^{t+1}|\bar{s}_i^t,a_i^t)$ per (4.9) needs to be updated too.

Let us start with the former. Like (3.12), this can now depend on ISDs from OLAFs $\mathring{x}_i^{t+1}$

$$\Pr(\tilde{x}_i^{t+1}|\langle x_i^t,D_i^{t+1}\rangle,\mathring{x}_i^{t+1},a_i^t,I_{\to i}^{t+1}) \triangleq \\ \sum_{u_{\to i}^{t+1}=\langle y_u^t,a_u^t,y_u^{t+1}\rangle} I(u_{\to i}^{t+1}|D_i^{t+1},x_v^t,a_i^t,x_v^{t+1})\Pr(\tilde{x}_i^{t+1}|x_i^t,\mathring{x}_i^{t+1},a_i^t,u_{\to i}^{t+1}), \quad (5.6)$$

with $\Pr(\tilde{x}_i^{t+1}|x_i^t,\mathring{x}_i^{t+1},a_i^t,u_{\to i}^{t+1})$ simply the product of CPTs of the NLAFs, as given by (3.12), but now restricted to only y_i^t,y_i^{t+1},a_{-i}^t that are influence sources.

We are now in the position to define the IALM under intra-stage dependencies:

Definition 20 (IALM)**.** Given an LFM with intra-stage dependencies, $\mathcal{M}^{LFM}$, and profile of policies for other agents π_{-i}, an *Influence-Augmented Local Model (IALM)* for agent i is a POMDP $\mathcal{M}_i^{IALM}(\mathcal{M}^{LFM},\pi_{-i}) = \left\langle \bar{\mathcal{S}},\mathcal{A}_i,\bar{T}_i,\bar{R}_i,\mathcal{O}_i,\bar{O}_i,H,b_i^{l,0} \right\rangle$, where

- $\bar{\mathcal{S}},\mathcal{A}_i,\bar{R}_i,\mathcal{O}_i,\bar{O}_i,H,b_i^{l,0}$ are identical to those in Definition 15,
- $\bar{T}_i$ is the transition function is defined as:

$$\begin{aligned}\bar{T}_i(\bar{s}_i^{t+1}|\bar{s}_i^t,a_i^t) &\triangleq \Pr(x_i^{t+1}|\langle x_i^t,D_i^{t+1}\rangle,a_i^t,I_{\to i}^{t+1})\mathbf{1}_{\{D_i^{t+2},d(x_i^t,a_i^t,x_i^{t+1},D_i^{t+1})\}} \\ &= \Pr(\tilde{x}_i^{t+1}|\langle x_i^t,D_i^{t+1}\rangle,\mathring{x}_i^{t+1},a_i^t,I_{\to i}^{t+1}) \Pr(\mathring{x}_i^{t+1}|x_i^t,\tilde{x}_i^{t+1},a_i^t)\mathbf{1}_{\{D_i^{t+2},d(x_i^t,a_i^t,x_i^{t+1},D_i^{t+1})\}}, \quad (5.7)\end{aligned}$$

with the first term is given by (5.6) and the second term is given by (3.11).

5.3 Planning in an IALM with ISDs

Since the only modifications that we needed to make to incorporate ISDs were in the transition function, the conclusions about how to plan in IALM made in Section 4.3 remain valid. In particular, the IALM is still a POMDP, with a well-defined belief-update function, and value functions. The solution of the IALM still gives the influence-based best-response value, defined in (4.17) as the value of the initial local-form belief: $V_i(I_{\to i}(\pi_{-i})) \triangleq V_i^0(b_i^{l,0})$.

6. Sufficiency of Influence-Based Abstraction

In this section, we will show that influence-based abstraction is *completely lossless.* By that we mean that an IALM constructed according to Definition 20 can be used to accurately predict rewards and observations, and thus to compute an exact, optimal (best-response) value.

The latter is our main result, Theorem 1, which shows that the optimal values for the GFBRM and the IALM are equal, thus establishing that one can use the IALM to plan (or learn) without any loss in value. In other words, it proves that the definition of influence constitutes a sufficient statistic for predicting the optimal value, and thus that the resulting IALM achieves a best-response against the policy π_{-i} that generated the influence $I_{\to i}(\pi_{-i})$.

Theorem 1. *For a finite-horizon POSG, the solution of the IALM for the incoming influence point $I_{\to i}(\pi_{-i})$ associated with any π_{-i} achieves the same value $V_i(I_{\to i}(\pi_{-i}))$, given by* (4.17)*, as the best-response value $V_i(\pi_{-i})$, given by* (3.7)*, computed against π_{-i} directly:*

$$\forall_{\pi_{-i}} \quad V_i(I_{\to i}(\pi_{-i})) = V_i(\pi_{-i}). \quad (6.1)$$

We note that this results also holds in the presence of intra-stage connections. To prove the result (in Section 6.4) we will show that the immediate reward terms and observation probabilities are equal (Section 6.3). In turn, to show this, we will need to show that transition probabilities are the same given a local-form belief and a global-form belief, which means that the local-form belief is a sufficient statistic to predict the next local state (Section 6.2). In order to allow the rewriting to take place, we first show how the global-form belief can be factorized.

We believe that this proof by itself is useful: it isolates the core technical property that needs to hold for sufficiency in Lemma 1 in Section 6.2. In this way it 1) conveys insight into the nature of how abstraction of latent state factors affects value, 2) provides a derivation that can be used to obtain simplifications of the definition of influence (Definition 19) in simpler cases, and 3) provides a recipe of how to prove similar results in problems which add even more complexities.

6.1 Factorization of the Global-Form Belief

In order to prove the equivalence of the GFBRM and the IALM, we will show that their value functions are the same. In order to do that, it will be necessary to decompose the global-form belief b_i^g in components.

To do that, we make use of the insight that, for any D_i^{t+1}, the law of total probability allows us to write

$$b_i^g(s^t,\vec{h}_{-i}^t) = \sum_{D_i^{t+1}} b_i(\langle x_i^t,y_i^t\rangle,\vec{h}_{-i}^t,D_i^{t+1}) = \sum_{D_i^{t+1}} b_i(x_i^t,D_i^{t+1})b_i(y_i^t,\vec{h}_{-i}^t|x_i^t,D_i^{t+1}). \tag{6.2}$$

(We drop the superscript 'g' because we are rewriting to something that we do not call global-form belief anymore.)
Also, it is important to remember that the belief is *defined* as

$$b_i^g(s^t,\vec{h}_{-i}^t) \triangleq \Pr(s^t,\vec{h}_{-i}^t|\vec{h}_i^t,b^0,\pi_{-i}),$$

which means that in (6.2), the definitions of the components are

$$b_i(x_i^t,D_i^{t+1}) \triangleq \Pr(x_i^t,D_i^{t+1}|\vec{h}_i^t,b^0,\pi_{-i}), \tag{6.3}$$

$$b_i(y_i^t,\vec{h}_{-i}^t|x_i^t,D_i^{t+1}) \triangleq \Pr(y_i^t,\vec{h}_{-i}^t|x_i^t,D_i^{t+1},\vec{h}_i^t,b^0,\pi_{-i}). \tag{6.4}$$

These equations further clarify how to think about inclusion of actions a_i and observations o_i inside the d-separating set D_i^{t+1}: the belief *per definition* conditions on the history of actions and observations, as such these can be included in D_i^{t+1} without further problems. In particular, suppose that a_i^k is part of d-separating set D_i^{t+1}, then this will lead to $\Pr(x_i^t,\langle\ldots a_i^k\ldots\rangle\,|\,\langle\ldots a_i^k\ldots\rangle,b^0,\pi_{-i})$ in (6.4). However, the interpretation is simply that this does not influence the probabilities, since $P(x|x)=1$. Similarly, it would lead to a term $\Pr(y_i^t,\vec{h}_{-i}^t|x_i^t,\langle\ldots a_i^k\ldots\rangle,\langle\ldots a_i^k\ldots\rangle,b^0,\pi_{-i})$ in (6.4). Again, this poses no problem, since $\Pr(y|x,x)=\Pr(y|x)$. However, let us repeat that we do need all observation relevant state factors in the local state: otherwise we cannot define the local observation model $\bar{O}_i$ and track the local-form belief $b_i(x_i^t,D_i^{t+1})$ (cf. Definition 10 and Definition 15).

6.2 Sufficiency for Prediction of Local State Transitions

In this section, we show that the influence together with the local-form belief is sufficient to predict local state transitions. We first prove the following lemma, that shows that pairwise marginal distributions over states are the same in the IALM and the GFBRM. This will then be used in other proofs.

Lemma 1. *The joint distribution over current local state and next local state induced by a local-form belief is identical to that of the global-form belief:*

$$\forall_{\vec{h}_i^t}\forall_{x_i^t,x_i^{t+1}} \qquad \Pr(x_i^t,x_i^{t+1}|b_i^g,a_i^t,\pi_{-i}) = \Pr(x_i^t,x_i^{t+1}|b_i^l,a_i^t,I_{\rightarrow i}^{t+1}), \tag{6.5}$$

where b_i^l, b_i^g denote the for the local-form and global-form beliefs induced by $\vec{h}_i^t$.

Proof. To improve readability we will omit some time indices that do not cause confusion. We assume arbitrary $\vec{h}_i^t,x_i^t,x_i^{t+1}$, and start with the left-hand side, which is given by (3.15):

$$\sum_{y_i^t}\sum_{a_{-i}} \Pr(x_i^{t+1}|s^t,a_i,a_{-i}) \sum_{\vec{h}_{-i}^t} \Pr(a_{-i}|\vec{h}_{-i}^t,\pi_{-i}) b_i^g(s^t,\vec{h}_{-i}^t)$$

$$=\{\text{via (6.2)}\}$$

$$\sum_{y_i^t}\sum_{a_{-i}} \Pr(x_i^{t+1}|s^t,a_i,a_{-i}) \sum_{\vec{h}_{-i}^t} \Pr(a_{-i}|\vec{h}_{-i}^t,\pi_{-i}) \left[\sum_{D_i^{t+1}} b_i(x_i^t,D_i^{t+1}) b_i(y_i^t,\vec{h}_{-i}^t|x_i^t,D_i^{t+1})\right] \tag{6.6}$$

$$=\{\text{via (3.9)}\}$$

$$\sum_{y_i^t}\sum_{a_{-i}} \left[\sum_{y_i^{t+1}} \Pr(y_i^{t+1},x_i^{t+1}|s^t,a_i,a_{-i})\right] \sum_{\vec{h}_{-i}^t} \Pr(a_{-i}|\vec{h}_{-i}^t,\pi_{-i}) \sum_{D_i^{t+1}} b_i(x_i^t,D_i^{t+1}) b_i(y_i^t,\vec{h}_{-i}^t|x_i^t,D_i^{t+1}) \tag{6.7}$$

$$=\{\text{via (3.10)}\}$$

$$\sum_{y_i^t}\sum_{a_{-i}}\sum_{y_i^{t+1}} \Pr(\mathring{x}_i^{t+1}|x_i^t,a_i,\tilde{x}_i^{t+1}) \Pr(\tilde{x}_i^{t+1}|x_i^t,\mathring{x}_i^{t+1},a_i,y_u^t,y_u^{t+1},a_u) \Pr(y_i^{t+1}|x_i^t,y_i^t,a_i,a_{-i},x_i^{t+1})$$

$$\sum_{\vec{h}_{-i}^t} \Pr(a_{-i}|\vec{h}_{-i}^t,\pi_{-i}) \sum_{D_i^{t+1}} b_i(x_i^t,D_i^{t+1}) b_i(y_i^t,\vec{h}_{-i}^t|x_i^t,D_i^{t+1}) \tag{6.8}$$

$$=\{\text{reordering terms}\}$$

$$\sum_{D_i^{t+1}} \Pr(\mathring{x}_i^{t+1}|x_i^t,a_i,\tilde{x}_i^{t+1}) b_i(x_i^t,D_i^{t+1})$$

$$\left[\sum_{a_{-i}}\sum_{\vec{h}_{-i}^t}\sum_{y_i^t}\sum_{y_i^{t+1}} \Pr(\tilde{x}_i^{t+1}|x_i^t,\mathring{x}_i^{t+1},a_i,y_u^t,y_u^{t+1},a_u) \Pr(y_i^{t+1}|x_i^t,y_i^t,a_i,a_{-i},x_i^{t+1}) \Pr(a_{-i}|\vec{h}_{-i}^t,\pi_{-i}) b_i(y_i^t,\vec{h}_{-i}^t|x_i^t,D_i^{t+1})\right] \tag{6.9}$$

This equation has grouped together all the probabilities that are affected by the non-local part of the problem in between the brackets. The terms before do not depend on the external part at all. We will now further investigate the externally influenced (bracketed) part:

$$\sum_{a_{-i}}\sum_{\vec{h}_{-i}^t}\sum_{y_i^t}\sum_{y_i^{t+1}} \Pr(\tilde{x}_i^{t+1}|x_i^t,\mathring{x}_i^{t+1},a_i,u_{\rightarrow i}^{t+1}) \Pr(y_i^{t+1}|x_i^t,y_i^t,a_i,a_{-i},x_i^{t+1}) \pi_{-i}(a_{-i}|\vec{h}_{-i}^t) b_i(y_i^t,\vec{h}_{-i}^t|x_i^t,D_i^{t+1}) \tag{6.10}$$

$$=\sum_{a_{-i}}\sum_{y_i^t}\sum_{y_i^{t+1}} \Pr(\tilde{x}_i^{t+1}|x_i^t,\mathring{x}_i^{t+1},a_i,u_{\rightarrow i}^{t+1}) \Pr(y_i^{t+1}|x_i^t,y_i^t,a_i,a_{-i},x_i^{t+1}) \sum_{\vec{h}_{-i}^t} \pi_{-i}(a_{-i}|\vec{h}_{-i}^t) b_i(y_i^t,\vec{h}_{-i}^t|x_i^t,D_i^{t+1}) \tag{6.11}$$

In this equation, not all non-modeled factors y_i^{t+1} are relevant: we can restrict to the intra-stage sources y_u^{t+1} and their intra-stage ancestors y_v^{t+1}, other factor's probabilities just sum to 1. This yields:

$$\sum_{a_{-i}}\sum_{y_i^t}\sum_{y_u^{t+1}}\Pr(\tilde{x}_i^{t+1}|x_i^t,\mathring{x}_i^{t+1},a_i,u_{\to i}^{t+1})\sum_{y_v^{t+1}}\Pr(y_u^{t+1},y_v^{t+1}|x_i^t,y_i^t,a_i,a_{-i},x_i^{t+1})\sum_{\vec{h}_{-i}^t}\pi_{-i}(a_{-i}|\vec{h}_{-i}^t)b_i(y_i^t,\vec{h}_{-i}^t|x_i^t,D_i^{t+1}) \tag{6.12}$$

={restricting to a_v,x_v^{t+1} that actually influence y_u^{t+1}. I.e., v denotes other 'second order' sources}

$$\sum_{a_{-i}}\sum_{y_i^t}\sum_{y_u^{t+1}}\Pr(\tilde{x}_i^{t+1}|x_i^t,\mathring{x}_i^{t+1},a_i,u_{\to i}^{t+1})\sum_{y_v^{t+1}}\Pr(y_u^{t+1},y_v^{t+1}|x_v^t,y_v^t,a_i,a_v,x_v^{t+1})\sum_{\vec{h}_{-i}^t}\pi_{-i}(a_{-i}|\vec{h}_{-i}^t)b_i(y_i^t,\vec{h}_{-i}^t|x_i^t,D_i^{t+1})$$

={pushing in summations, recall $u_{\to i}^{t+1}=\langle y_u^t,a_u,y_u^{t+1}\rangle$}

$$\sum_{u_{\to i}^{t+1}}\Pr(\tilde{x}_i^{t+1}|x_i^t,\mathring{x}_i^{t+1},a_i,u_{\to i}^{t+1})\sum_{a_v}\sum_{y_v^t}\sum_{y_v^{t+1}}\Pr(y_u^{t+1},y_v^{t+1}|x_v^t,y_v^t,a_i,a_v,x_v^{t+1})\sum_{\vec{h}_{-i}^t}\pi_{-i}(a_{-i}|\vec{h}_{-i}^t)b_i(y_i^t,\vec{h}_{-i}^t|x_i^t,D_i^{t+1}) \tag{6.13}$$

={marginalize out non-relevant terms}

$$\sum_{u_{\to i}^{t+1}}\Pr(\tilde{x}_i^{t+1}|x_i^t,\mathring{x}_i^{t+1},a_i,u_{\to i}^{t+1})\sum_{a_v}\sum_{y_v^t}\sum_{y_v^{t+1}}\Pr(y_u^{t+1},y_v^{t+1}|x_v^t,y_v^t,a_i,a_v,x_v^{t+1})$$
$$\sum_{\vec{h}_u^t}\sum_{\vec{h}_v^t}\pi_u(a_u^t|\vec{h}_u^t)\pi_v(a_v^t|\vec{h}_v^t)b_i(y_u^t,y_v^t,\vec{h}_u^t,\vec{h}_v^t|x_i^t,D_i^{t+1}) \tag{6.14}$$

={let $w=u\cup v$ denote the union of direct and indirect sources}

$$\sum_{u_{\to i}^{t+1}}\Pr(\tilde{x}_i^{t+1}|x_i^t,\mathring{x}_i^{t+1},a_i,u_{\to i}^{t+1})\sum_{a_v}\sum_{y_v^t}\sum_{y_v^{t+1}}\Pr(y_u^{t+1},y_v^{t+1}|x_v^t,y_v^t,a_i,a_v,x_v^{t+1})\sum_{\vec{h}_w^t}\pi_w(a_w|\vec{h}_w^t,\pi_w)b_i(y_w^t,\vec{h}_w^t|x_i^t,D_i^{t+1}) \tag{6.15}$$

={since $b_i(y_w^t,\vec{h}_w^t|x_i^t,D_i^{t+1})\triangleq\Pr(y_w^t,\vec{h}_w^t|x_i^t,D_i^{t+1},\vec{h}_i^t,b^0,\pi_{-i})\overset{\{\text{def. of d-set (5.2)}\}}{=}\Pr(y_w^t,\vec{h}_w^t|D_i^{t+1},b^0,\pi_{-i})$ }

$$\sum_{u_{\to i}^{t+1}}\Pr(\tilde{x}_i^{t+1}|x_i^t,\mathring{x}_i^{t+1},a_i,u_{\to i}^{t+1})\sum_{\left\langle y_v^t,a_v,y_v^{t+1}\right\rangle}\Pr(y_u^{t+1},y_v^{t+1}|x_v^t,y_v^t,a_i,a_v,x_v^{t+1})\sum_{\vec{h}_w^t}\pi_w(a_w|\vec{h}_w^t)\Pr(y_w^t,\vec{h}_w^t|D_i^{t+1},b^0,\pi_{-i}). \tag{6.16}$$

We can now apply the definition of influence (Definition 19 on page 826) to (6.16), which yields

$$=\sum_{u_{\to i}^{t+1}=\left\langle y_u^t,a_u,y_u^{t+1}\right\rangle}\Pr(\tilde{x}_i^{t+1}|x_i^t,\mathring{x}_i^{t+1},a_i,y_u^t,a_u,y_u^{t+1})I(u_{\to i}^{t+1}|D_i^{t+1},x_v^t,a_i,x_v^{t+1}), \tag{6.17}$$

which is the definition (5.6) of $\Pr(\tilde{x}_i^{t+1}|\langle x_i^t,D_i^{t+1}\rangle,\mathring{x}_i^{t+1},a_i,I_{\to i}^{t+1})$.

Substituting (6.17) back in (6.9) we get

$$\sum_{D_i^{t+1}}\Pr(\mathring{x}_i^{t+1}|x_i^t,x_i^{t+1},\tilde{x}_i^{t+1},a_i)b_i(x_i^t,D_i^{t+1})\left[\Pr(\tilde{x}_i^{t+1}|\langle x_i^t,D_i^{t+1}\rangle,\mathring{x}_i^{t+1},a_i,I_{\to i}^{t+1})\right]$$
$$=\{\text{via 5.7 }\}$$
$$\sum_{D_i^{t+1}}\Pr(x_i^{t+1}|x_i^t,D_i^{t+1},a_i,I_{\to i}^{t+1})b_i(x_i^t,D_i^{t+1})\overset{\{\text{via (4.16)}\}}{=}\Pr(x_i^t,x_i^{t+1}|b_i^l,a_i^t,I_{\to i}^{t+1}), \tag{6.18}$$

which concludes the proof. □

Lemma 2. *A local-form belief is a sufficient statistic for predicting the next local state. That is, when b_i^l, b_i^g denote the for the local-form and global-form beliefs induced by the same action-observation history $\vec{h}_i^t$, we have that:*

$$\forall_{\vec{h}_i^t} \forall_{x_i^{t+1}} \qquad \Pr(x_i^{t+1}|b_i^g,a_i^t,\pi_{-i}) = \Pr(x_i^{t+1}|b_i^l,a_i^t,I_{\to i}^{t+1}). \tag{6.19}$$

Proof. This follows directly from Lemma (1):

$$\begin{aligned}\Pr(x_i^{t+1}|b_i^l,a_i^t,I_{\to i}^{t+1}) &= \sum_{x_i^t} \Pr(x_i^t,x_i^{t+1}|b_i^l,a_i^t,I_{\to i}^{t+1}) \\ &= \sum_{x_i^t} \Pr(x_i^t,x_i^{t+1}|b_i^g,a_i^t,\pi_{-i}) = \Pr(x_i^{t+1}|b_i^g,a_i^t,\pi_{-i}). \quad \square\end{aligned}$$

6.3 Sufficiency for Predicting Rewards and Observations

Given that we established that local-form beliefs in an IALM are sufficient to predict local-state transitions, we can now also establish their sufficiency for predicting rewards and observations.

Lemma 3. *The local-form belief is a sufficient statistic to predict the immediate reward. That is*

$$\forall_{\vec{h}_i} \forall_{a_i^t} \qquad R_i(b_i^g,a_i^t) = R_i(b_i^l,a_i^t) \tag{6.20}$$

where b_i^l, b_i^g denote the for the local-form and global-form beliefs induced by $\vec{h}_i$.

Proof. When we compare equations (3.14) and (4.15), we see that this holds if $\Pr(x_i^t,x_i^{t+1}|b_i^g,a_i^t,\pi_{-i}) = \Pr(x_i^t,x_i^{t+1}|b_i^l,a_i^t,I_{\to i}^{t+1})$. This is precisely what Lemma 1 shows. $\square$

Lemma 4. *The local-form belief is a sufficient statistic for predicting the observation. That is:*

$$\forall_{\vec{h}_i} \forall_{a_i^t,o_i^{t+1}} \qquad \Pr(o_i^{t+1}|b_i^g,a_i^t) = \Pr(o_i^{t+1}|b_i^l,a_i^t), \tag{6.21}$$

where b_i^l, b_i^g denote the for the local-form and global-form beliefs induced by $\vec{h}_i$.

Proof. Comparing equations (3.16) and (4.11), we see that equality holds if $\Pr(x_i^{t+1}|b_i^g,a_i,\pi_{-i}) = \Pr(x_i^{t+1}|b_i^l,a_i,I_{\to i}^{t+1})$; this is exactly what Lemma 2 shows. $\square$

6.4 Proof of Theorem 1: Sufficiency for Predicting Optimal Value

Finally, we can prove that our definition of influence is sufficient to predicting the optimal best-response value. The values in (6.1) are defined as the value of the initial beliefs, cf. equations (4.17) and (3.7). Putting this all together, we need to show that

$$\forall_{\pi_{-i}} \quad V_i(I_{\to i}(\pi_{-i})) \triangleq V_i^0(b_i^{l,0}) = V_i^0(b_i^{g,0}) \triangleq V_i(\pi_{-i}). \tag{6.22}$$

The proof is by induction over the horizon, where the base case is given by the last stage.

Base Case. Assume an arbitrary last-stage AOH, $\vec{h}_i^{H-1}$, and let b_i^l, b_i^g denote the for the local-form and global-form beliefs induced by it. Their respective values are given by

$$V_i^t(b_i^g) = \max_{a_i} R_i(b_i^g,a_i),$$

$$V_i^t(b_i^l) = \max_{a_i} R_i(b_i^l,a_i).$$

So we need to show that the predicted immediate rewards are equal. This, however, is exactly what Lemma 3 shows.

Induction Step. The induction hypothesis is that, for stage $t+1$,

$$\forall_{\vec{h}_i^{t+1}} \qquad V_i^{t+1}(b_i^{l,t+1}) = V_i^{t+1}(b_i^{g,t+1}), \tag{6.23}$$

where we write $b_i^{l,t+1}, b_i^{g,t+1}$ are the local-form and global-form beliefs induced by $\vec{h}_i^{t+1}$.

Now we need to prove that $V_i^t(b_i^l) = V_i^t(b_i^g)$, for all $\vec{h}_i^t$. Since, per definition,

$$V_i^t(b_i^l) = \max_{a_i} Q_i^t(b_i^l,a_i),$$

$$V_i^t(b_i^g) = \max_{a_i} Q_i^t(b_i^g,a_i),$$

we will show this by proving that the Q-values are equal. Assume an arbitrary $\vec{h}_i^t$. Its Q-values, for all a_i, are given by (3.4):

$$Q_i^t(b_i^g,a_i) = R_i(b_i^g,a_i) + \gamma \sum_{o_i} \Pr(o_i|b_i^g,a_i) V_i^{t+1}(BU(b_i^g,a_i,o_i)) \tag{6.24}$$

By the induction hypothesis, we get

$$Q_i^t(b_i^g,a_i) = R_i(b_i^g,a_i) + \sum_{o_i} \Pr(o_i|b_i^g,a_i) V_i^{t+1}(BU(b_i^l,a_i,o_i)). \tag{6.25}$$

Note that $BU(b_i^g,a_i,o_i)$ and $BU(b_i^l,a_i,o_i)$ are the local-form and global-form beliefs induced by the same next-stage history $\vec{h}_i^{t+1} = (\vec{h}_i^t,a_i,o_i)$, and hence the induction hypothesis applies:

$$V_i^{t+1}(BU(b_i^g,a_i,o_i)) = V_i^{t+1}(BU(b_i^l,a_i,o_i)).$$

So, in order to show that (6.25) is equal to

$$Q_i^t(b_i^l,a_i) = R_i(b_i^l,a_i) + \sum_{o_i} \Pr(o_i|b_i^l,a_i) V_i^{t+1}(BU(b_i^l,a_i,o_i)) \tag{6.26}$$

we need to show equality for both the immediate rewards, $R_i(b_i^g,a_i) = R_i(b_i^l,a_i)$, and the observation probabilities, $\Pr(o_i|b_i^g,a_i) = \Pr(o_i|b_i^l,a_i)$. The former was shown in Lemma 3 and the latter was shown in Lemma 4. Hence, the Q-values are the same, hence the values are the same, which concludes the induction step. □

7. Tractable Influence Representations

There are a number of important problem classes and associated models developed in previous work that emphasize weakly coupled problem structure in more restrictive settings. We now reformulate these classes in the context of IBA, thus demonstrating how the theory presented in this paper unifies such previous work in a coherent graphical framework. All of the models that we review below are specialized instances of the factored Dec-POMDP (fDec-POMDP) model and since an fDec-POMDP is an fPOSG, our definition of influence is applicable to all of these models.

However, as we illuminate below, some sub-classes allow for particularly compact influence specifications that can be computed efficiently. Similar to the examples in Section 4.4, this makes clear how it is possible to compute best responses more effectively, and provides some intuition about how influence search approaches can enable speed-ups in these sub-classes.

We will also see how our unified perspective allows us to make novel observations about these previously defined classes that can lead to improvements and extensions. For instance, we will see that we can derive more compact forms of influence for the so-called EDI-Dec-MDP framework. In general we expect that the more compact the representation, the more efficiently these sub-classes can be solved. However, we note that, unlike (most of) the papers that introduced these sub-classes, in this paper we are not proposing an influence–search technique to solve the optimization for all agents. This is left for future work.

7.1 TD-POMDP

An earlier embodiment of influence abstraction (Witwicki & Durfee, 2010b; Witwicki, 2011; Witwicki et al., 2012) sought to exploit cooperative agents' weak coupling, showing that searching in the space of joint influences can provide significant speed-ups over searching the space of joint policies for a restrictive sub-class of fDec-POMDPs. The so-called *Transition-Decoupled POMDP (TD-POMDP)* (Witwicki, 2011) describes a local state for each agent that resembles our local form models. However, it also distinguishes so-called *mutually-modeled factors (MMFs)* common to more than one agent's local state. These MMFs have the same role as our non-locally affected factors (NLAFs), but impose additional restrictions (Witwicki, 2011, Section 3.4.3). Specifically, there are two important differences that make the TD-POMDP more restrictive than our local-form model:

1. The TD-POMDP does not allow intra-stage dependencies between private state variables and MMFs.
2. In a TD-POMDP each state factor can only be directly affected by (have an incoming edge from) the action (or private state variable) of just one agent.

These constraints effectively limit the representational power of the TD-POMDP to *non-concurrent* interactions. As an example, the planetary exploration domain from Figure 7 can be directly modeled as a TD-POMDP by making pl the single MMF in the model. In contrast, the House Search problem from Figure 2 cannot be modeled in the same way: the TD-POMDP version of this problem requires separating the 'found' variable into two

MMF variables: 'found by agent 1' and 'found by agent 2' thus increasing the size of the local problems (Witwicki et al., 2012).

Observation. The TD-POMDP model is a special case of the LFM, imposing restrictions that limit its modeling capabilities to a subset of those interactions representable as local-form POSGs: TD-POMDP $\subset$ LFM.

The TD-POMDP's formalization is less flexible than that proposed in this paper. In particular, it seems difficult to extend the TD-POMDP to deal with intra-stage connections, which we have argued in Sections 4.1.1 and 5 is important for expressiveness.

However, the authors derive that this representational restriction affords the TD-POMDP a particular form of influence, since the history of mutually-modeled factors is guaranteed to d-separate an agent's observations from all external factors (i.e., those outside of its local state). The form of influence that Witwicki and Durfee propose for TD-POMDPs actually corresponds to our notion of 'induced CPT' (cf. Section 4.2.1) or the marginal of their product (4.7). In many cases this allows for compact representations of the influence. Compact influence representations in turn appear to provide traction when it comes to computing solutions, as evidenced by the efficiency and scalability gains of influence-space search for TD-POMDPs (Witwicki, 2011; Witwicki et al., 2012).

7.2 TI-Dec-MDP

Another model, the Transition-Independent Dec-MDP (Becker et al., 2003), imposes other more stringent restrictions on the dependencies between agents' local models. In particular, an agent fully observes its private factors and there are no paths of dependence in the DBN connecting one agent's private factors, actions, and observation to those of another. This implies that the agents are *transition and observation independent*. Agents' local models are instead coupled through their rewards, which can depend on the *events* $e_i = \langle s_i^t, a_i^t, s_i^{t+1} \rangle$ that occur (at most once) within another agents' state space.

This class of problems includes, for instance, missions executed by Mars rovers during which they need to collect samples at various sites. In such settings, it is reasonable to assume that the rovers have their own routes and therefore will not affect each other's transitions. However, the utility of rover 1 taking a soil sample at a particular site might depend on what samples are taken by rover 2 at nearby sites.

For instance, imagine that the rovers pass at different sides of a canyon, taking a picture of this canyon provides some utility, but if both rovers take a picture from their side of the canyon (corresponding to the individual events e_i), this may enable a better 3D reconstruction, providing more value than just the sum of two individual pictures.

This can be easily captured in a factored representation as shown in Figure 12. It shows that the combined occurrence of both agents' events (as represented by Boolean variable E) leads to a change in the reward (split between the agents as soon as the event occurs). When the discount factor is 1 (as Becker et al., 2003 assume), the reward may as well be affected at the last time step as we have indicated. This leads to a very simple form of influence $I_{\rightarrow i}(e_j^{h-1})$ corresponding to the probability of e_j^{h-1} being true. This corresponds exactly to the characterization of 'parameter space' presented by Becker et al. in their development of the *coverage set algorithm (CSA)*.

Our characterization of the TI-Dec-MDP immediately leads to some new insights.

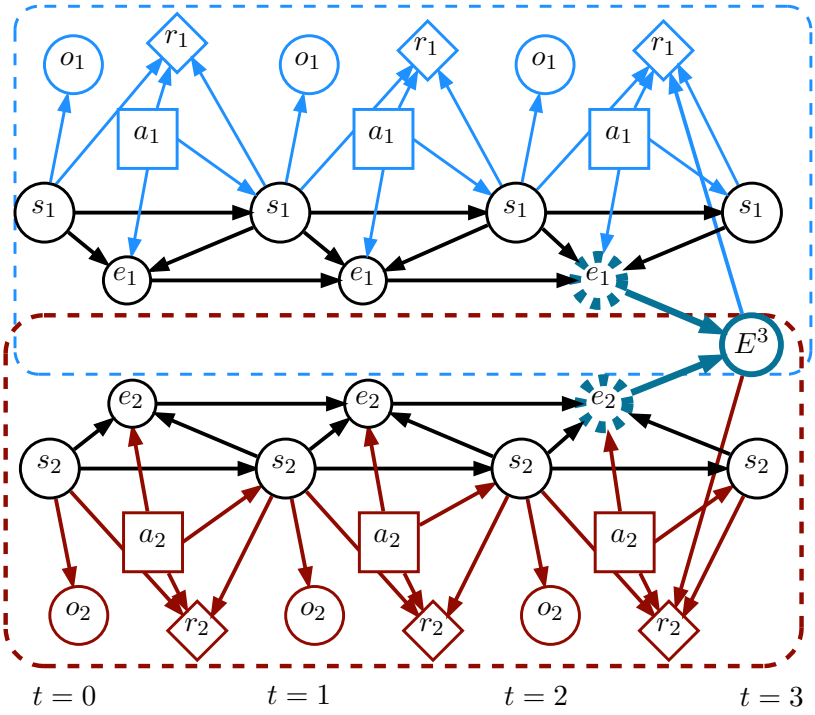


Figure 12: Local-form representation of TI-Dec-MDP: the LFMs of both agents (indicated by different color bounding boxes) are tied together by a influence source E that indicates if the joint event happened. To be able to predict the value of this influence source, the agent i will need to condition on their e_i variable.

Observation. While the TI-Dec-MDP framework is arguably more restrictive than the TD-POMDP, the graphical structure in Figure 12 makes clear that a TI-Dec-MDP is not a TD-POMDP: E^3 is affected from both sub-problems simultaneously.

Observation. The properties that 1) events cannot occur more than once; and 2) events are unobserved, allow for history-independent influence encoding in TI-Dec-MDPs.

Observation. CSA and closely-related TI-Dec-MDP algorithms (Petrik & Zilberstein, 2009) exploit structure that is also present in more general contexts, such as TI-Dec-POMDPs with partial observability of private factors.

That is, we make the observation that that CSA and its successors can actually be extended to more general problem whose joint value function is piecewise linear and convex in the influence parameters, such as settings where agents receive only partial observations of their local states.

7.3 Event-Driven Interactions

The TI-Dec-MDPs assumes that transitions are independent, but interactions are present in rewards. However, in many problems it may be the other way around: for instance, the rewards that a vacuum cleaner robot generates only depends on the amount of dirt it cleans

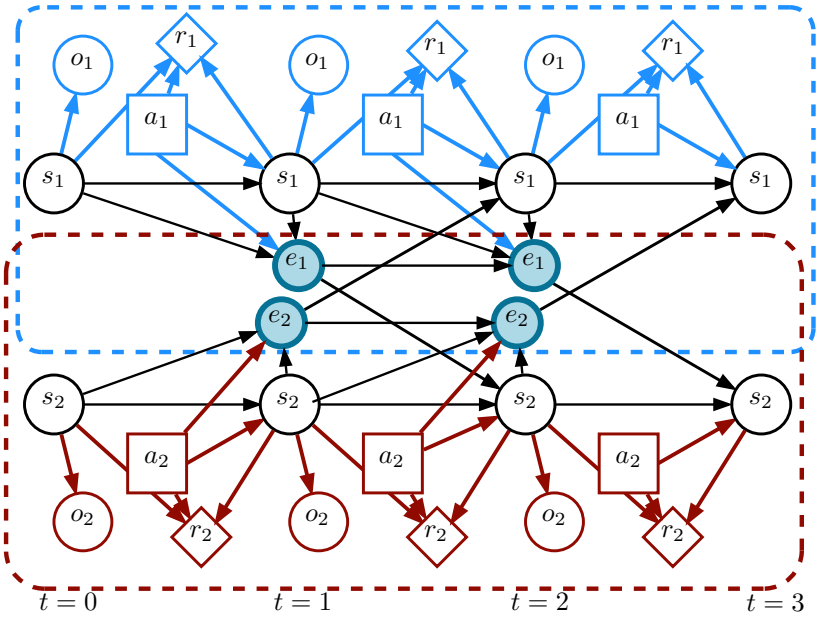


Figure 13: Local-form representation of the ED-Dec-MDP. The events e_1 act as influence destination for agent 2 and vice versa. To avoid clutter we do not indicate influence sources. The history of all e_i serves as a d-separating set for both agents.

up, but it cannot enter a dirty room until a general purpose house-hold robot opens the door.

To deal with such problems Becker et al. (2004) proposed the *Dec-MDP with Event-Driven Interaction (EDI-Dec-MDP)*, which provides an explicit representation for structured transition dependencies between two agents. Again, we will interpret this model in the IBA framework, as illustrated in Figure 13. This figure shows that, agent 1's transition probabilities may be affected by the prior occurrence of agent 2's event e_2 (and vice versa). In this case, the history of these event features are sufficient for d-separation, i.e., $D_1 = \{e_1, e_2\}$. This leads us to develop a new influence specification for this sub-class of problems:

Observation. The (induced-CPT form of) influence on EDI-Dec-MDP agent i, $I_{\rightarrow i}^{t+1}(\pi_j)$ can be defined as $I(s_j^t, a_j^t, s_j^{t+1} | \vec{e}_i^t, \vec{e}_j^t)$. Moreover, similar to what we saw in Section 4.4, the history $\vec{e}_i^t, \vec{e}_j^t$ can be represented compactly since events can only switch to true.

The marginal of product of induced CPTs $Pr(e_1^{t+1}, e_2^{t+1} | \vec{e}_1^t, \vec{e}_2^t)$ is similar to the parameters used by Becker *et al.* (in their application of CSA), but is slightly more compact, since it does not depend on private factors s_i^t, which our theory suggests to be unnecessary.

Having derived a more compact parameter form, we anticipate that this will translate directly into a more efficient application of CSA. We note that our reformulation of the TI-Dec-MDP and EDI-Dec-MDP also serve as influence specifications for the EDI-CR model (Mostafa & Lesser, 2009), developed to include both event-driven interactions and reward dependencies (as in the TI-Dec-MDP).

The *distributed POMDP with coordination locales (DPCL)* model (Varakantham et al., 2009) can also be reinterpreted using Figure 13. This model assumes all agents' observations are conditionally independent given the state, but that in some specific states, agents can affect each other's transitions or rewards. Looking at Figure 13, the events e_i precisely can model what Varakantham et al. refer to as *future-time coordination locales* ("situations where actions of one agent impact actions of others in the future"). Varakantham et al. also consider *same-time* coordination locales, which can model simultaneous effects such as robots failing to move when both try to move to the same grid cell. In Figure 13 this would be captured by adding arrows from $\vec{e}_i^t$ to s_i^{t+1} (or alternatively by introducing joint events E, as in Figure 12, at every time step). While these same-time coordination locales overcome the modeling requirements of non-concurrency as observed in TD-POMDPs and ED-MDPs, the solution method proposed by Varakantham et al. is heuristic. In fact, it is precisely our definition of influence presented in this paper that explains how to deal with such concurrent interaction in a principled fashion.

7.4 ND-POMDP

The *Network Distributed POMDP (ND-POMDP)* introduced by Nair et al. (2005) is another transition and observation independent model whose structure can easily be represented in our framework. It was motivated by problems like sensor networks for intrusion detection, where the sensors need to select actions to scan their local surroundings. Such actions do not affect the local state of other sensors, but combinations of actions of neighboring sensor nodes can lead to higher rewards (e.g., if two adjacent sensors scan the same area where an intruder is, there might be a higher detection probability).

The formalization is depicted in Figure 14. Each agent's observation can be affected by an unaffectable, mutually-modeled factor s_0 (e.g., the location of an intruder). The reward dependencies involving joint actions are captured with an unobservable variable z encoding the local state-action pair that in much the same way as did the TI-Dec-MDP's events. The difference is that these joint actions are not constrained to occur only once, and may affect the rewards at any time.

In general, an ND-POMDP can consist of multiple local neighborhoods, which can be modeled using a coordination (hyper-)graph (Guestrin, Koller, & Parr, 2002a; Nair et al., 2005; Kok & Vlassis, 2006). Agents correspond to nodes, while $\mathcal{E}$ is a set of (hyper-)edges corresponding to subsets, e, of agents. To encode the interactions between different subsets of agents $e \in \mathcal{E}$, one can introduce different variables z_e. Our reformulation presented here immediately leads to the first specification of influence that we are aware of for this problem class. Let us write $N(i)$ to denote the neighbors of agent i excluding agent i itself: $N(i) = \{j \in \mathcal{D} | \exists e \in \mathcal{E}\ i,j \in e\ \wedge\ i \neq j\}$.

Observation. The influence on ND-POMDP agent i, $I_{\rightarrow i}^{t+1}(\pi_{-i})$ can be defined as

$$I_{\rightarrow i}^{t+1}(s_{N(i)}^t, a_{N(i)}^t | \vec{s}_0^{\,t}) = \prod_{j \in N(i)} I_{j\rightarrow}^t(s_j^t, a_j^t | \vec{s}_0^{\,t}), \tag{7.1}$$

with $I_{j\rightarrow}^t(s_j^t, a_j^t | \vec{s}_0^{\,t}) = \Pr(s_j^t, a_j^t | \vec{s}_0^{\,t})$ the outgoing influence of agent j.

Like the other mentioned sub-classes, the ND-POMDP affords a compact influence encoding, suggesting that influence-based planning methods could gain traction if applied

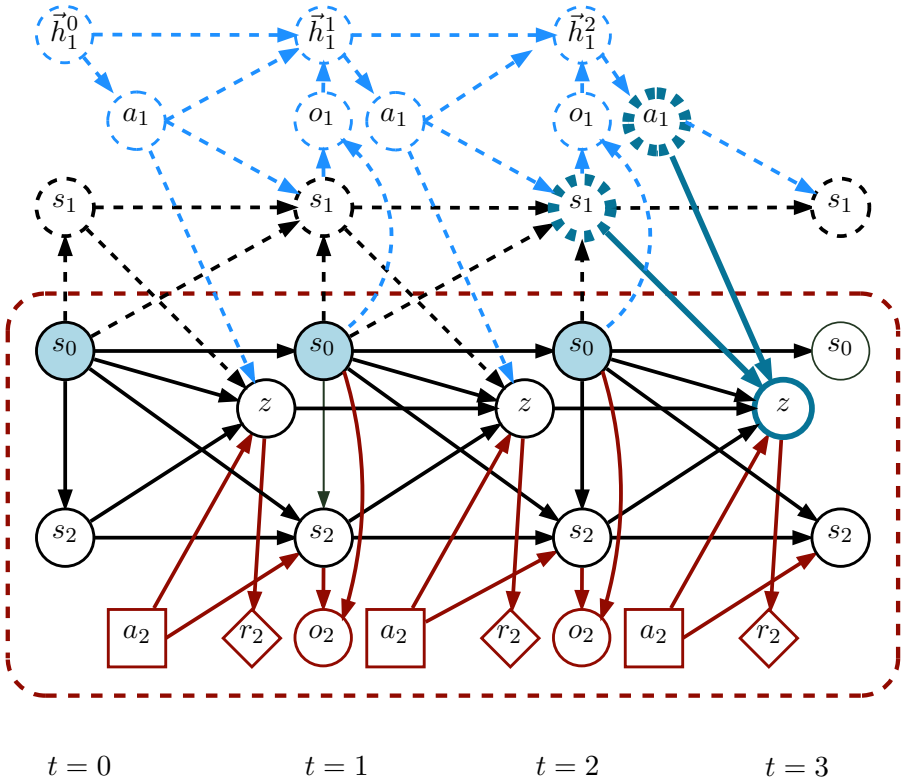


Figure 14: Local-form representation of agent 2 in a two-agent ND-POMDP. Highlighted are the influence sources for the "neighborhood state-action variable" z^3. (Time indices are omitted in the figure to avoid clutter, but can be inferred from the stages indicated at the bottom.) Note that because z itself nor any of its descendants are observable (rewards are not observed in Dec-POMDPs), it does not open a path of influence to s_2. Therefore only the history of s_0 needs to be encoded in the d-separating set.

here. Existing forms of influence search exploit the fact that one can enumerate the *joint influences,* which describe how agents influence each other (Witwicki & Durfee, 2010b; Witwicki et al., 2012). Due to the factorization of (7.1), the joint influence space here is a product space, which is easier to generate and search through. Moreover, it would be possible to exploit the graph structure of the ND-POMDP, similar to the approach by Witwicki (2011, Section 6.6): the space of joint influences can be decomposed as a factor graph over which one can optimize more effectively.

8. Related Work

Apart from the models and approaches reviewed in Section 7, there are important connections to be drawn with a large body of other work. Given the generality of the intuitive notion of 'influence' this should come as no surprise. Here we describe those relations to previous work and discuss further insights that our results provide. We sub-divide these related works in:

- work on locality of interaction and value factorization in multiagent systems,
- other decomposition-like approaches in multiagent systems, and
- more general forms of abstraction.

8.1 Locality of Interaction and Value Factorization

Past studies of factored Dec-POMDPs with factored value functions (Nair et al., 2005; Varakantham, Marecki, Yabu, Tambe, & Yokoo, 2007; Kumar, Zilberstein, & Toussaint, 2011; Witwicki & Durfee, 2011) have shown that gains in computational efficiency are possible when the value function can be expressed as the sum of a number of local components, each of which is specified over subsets of agents and state factors. In particular, the value in general Dec-POMDPs can be expressed as a function $V_\pi(s^t,\vec{o}^t)$ (e.g., see Oliehoek & Amato, 2016, chap. 3) of states and joint observation histories. Such a value function is said to be a *factored value function* if there is a set of components $e \in \mathcal{E}$ such that

$$V_\pi(s^t,\vec{o}^t) = \sum_{e\in\mathcal{E}} V_{\pi_e}(s_e^t,\vec{o}_e^t), \tag{8.1}$$

with π_e and $\vec{o}_e^t$ the policies respectively observation histories of the agents that participate in component e, and s_e^t the value of the state factors relevant for e. For such problems, it is easy to show that they possess *locality of interaction* (Nair et al., 2005): one can define a local neighborhood for each agent such that its actions will not impact the value beyond that neighborhood. This property allows one to reduce the problem to a form of (distributed) constraint optimization problem (e.g., see Oliehoek & Amato, 2016, chap. 8) .

However, for general factored Dec-POMDPs, the components e involve all agents and factors. I.e., they are *not* local (Oliehoek et al., 2008b; Oliehoek, 2010). This paper shows that even in the most general case, *it actually is possible to find local (i.e., restricted scope) components*, although this may be at the cost of introducing a dependence on the history of a subset of the local state factors (the d-separating set D_i). This means that it may be possible to extend the planning-as-inference method of (Kumar et al., 2011) to exploit structure in general fDec-POMDPs.[19] Researchers in the field of (deep) multiagent reinforcement learning, have tried to exploit such factorized structure approximately (Guestrin, Lagoudakis, & Parr, 2002b; Kok & Vlassis, 2006; Kuyer, Whiteson, Bakker, & Vlassis, 2008; Van der Pol & Oliehoek, 2016; Sunehag, Lever, Gruslys, Czarnecki, Zambaldi, Jaderberg, Lanctot, Sonnerat, Leibo, Tuyls, & Graepel, 2018; Rashid, Samvelyan, Schroeder de Witt, Farquhar, Foerster, & Whiteson, 2018; Castellini, Oliehoek, Savani, & Whiteson, 2019; Böhmer, Kurin, & Whiteson, 2019; Son, Kim, Kang, Hostallero, & Yi, 2019; Wang, Wang, Zheng, & Zhang, 2019), and our work brings deeper understanding of those approaches.

For instance, Sunehag et al. (2018) proposed a form of factored value functions (Guestrin et al., 2002a) making use of neural networks that can be understood better using the theory developed in this paper. Specifically they propose *value-decomposition networks*, a variant of *deep Q-networks (DQN)* introduced by Mnih, Kavukcuoglu, Silver, Rusu, Veness,

19. Note that, in general, IBA draws close connections to the paradigm of planning as inference (Toussaint, 2009); it performs inference to compute a compact local model; subsequently, inference (among other choices of solution methods) could be used to solve the IALM.

Bellemare, Graves, Riedmiller, Fidjeland, Ostrovski, Petersen, Beattie, Sadik, Antonoglou, King, Kumaran, Wierstra, Legg, and Hassabis (2015), that uses a Q-function

$$\tilde{Q}(\vec{h},a) = \sum_{i \in \mathcal{D}} \tilde{Q}_i(\vec{h}_i,a_i), \tag{8.2}$$

which is implemented in a single neural network with a linear layer at the end that performs this summation. They state that

> "the main assumption we make and exploit is that the joint action-value function for the system can be additively decomposed into value functions across agents"

and this assumption has been pointed out as a limitation in subsequent work (Rashid et al., 2018; Böhmer et al., 2019). This paper, however, demonstrates that *there is a very large class of problems for which this assumption (approximately) holds.* In particular, we show that for any factored Dec-POMDP for which we can create a set of local-form models (cf. Definition 10), we have that:

$$V_\pi(\vec{h}) = \sum_{i \in \mathcal{D}} V_i(b_i^l) = \sum_{i \in \mathcal{D}} \max_{a_i} Q_i(b_i^l,a_i), \tag{8.3}$$

where b_i^l is the local-form belief induced by $\vec{h}_i$ and the policies of the other agents π_{-i}. We also discuss that, by introducing dummy variables as required (cf. the end of Section 4.1.1), any factored Dec-POMDP can be re-coded as such set of local-from models.[20] As such, there is a very large class of problems for which "the system can be additively decomposed into value functions across agents". However, the devil is the details, we write "(approximately)" since the statement by Sunehag et al. (2018) is about Q not V. In particular, we have that

$$Q_\pi(\vec{h},a) \neq \sum_{i \in \mathcal{D}} Q_i(b_i^l,a_i) \tag{8.4}$$

since each term $Q_i(b_i^l,a_i)$ assumes that the other agents act according to π_{-i}, not according to a. This can explain the empirical improvements of methods that consider 'higher order approximations' with Q-components that involve subsets of agents (Oliehoek et al., 2013; Castellini et al., 2019; Böhmer et al., 2019).

We point out that this does not mean that an approximation $Q_\pi(\vec{h},a) \approx \sum_{i \in \mathcal{D}} Q_i(b_i^l,a_i)$ is senseless: in fact, we know that for the modified joint policy π', which is like π but does a instead of $\pi(\vec{h})$, the decomposition of $V_{\pi'}$ according to (8.3) also holds. As such, the question "how good of an approximation can we get with the individually factored Q-functions from (8.2)?" can be reinterpreted as a question of how the prediction of the components $Q_i(b_i^l,a_i)$ (which assume the others follow π_{-i}) generalize to "first action modified policies" π'. In other words, if for all such one-joint-action-modifications π' and their induced local form beliefs $b_i^{l\prime}$ we have that $Q_i(b_i^l,a_i) \approx Q_i(b_i^{l\prime},a_i)$, then we expect this approximation

20. Of course, depending on the problem, these components themselves might be small (need to involve only few state variables) or large. We cannot claim anything about the size of these components in general problems. We merely reason that they can in principle be constructed, which is sufficient to support our argument here.

to work well. Further formalizing the impact of such first-action-modifications may be a promising direction of research, and could lead to a novel notion of *influence strength* (Allen & Zilberstein, 2009; Oliehoek, Spaan, & Witwicki, 2015b) in multiagent domains.

We also remark that this analysis shows that, at least in cases that do not exhibit strict locality of interaction, value factorization inherently depends on the current policies of other agents, and hence implies an 'on-policy characteristic': when learning such factored value components we can only learn about the $Q_i(b_i^l,a_i)$ that are induced by those π_{-i} that are currently being followed by the other agents. Of course, agent i itself can still try to learn its approximation $\tilde{Q}_i(\vec{h}_i,a_i)$ with off-policy methods, but sudden large changes to the own policy may affect the ability of other agents to learn their local approximation. We speculate that in more tightly coupled problems on-policy methods with factorization may outperform off-policy ones.

8.2 More General Forms of Decomposition in MASs

In multiagent decision-making, there is a rich history of trying to leverage structured interactions. For instance, our approach resembles the distributed approximate planning method by Guestrin and Gordon (2002) in that both methods decompose an agent's decision model into internal and external parts. Our proposed abstraction, in addition to being sufficient for *optimal* decision-making, is more general in that it can deal with partial observability.

Allen and Zilberstein (2007, 2009) proposed a different formalization of 'influence' by building upon information-theoretical concepts (mutual information between individual actions and joint states/observations/rewards). They show how their notion of influence and influence gap (which measures differences between the influencing power of agents) can predict the difficulty of solving a problem. While conceptually closely related to our work, their proposed notion of influence does not seem to support doing abstraction in any non-trivial manner, and thus should be seen as a very different type of object than our influence point.

The work by Chitnis and Lozano-Pérez (2020) is close in spirit to IBA: they propose to form a local abstraction of a factored MDP that approximates the original model well. Their approach is to abstract away a subset of *exogenous variables* (Boutilier et al., 1999) and they propose a method to select this subset. However, exogenous variables are defined as variables that can influence our local model, but that *cannot be influenced by* the local model. This stands in stark contrast to the non-modeled variables in IBA which can be affected by the local model.

Another class of related work is that focusing on *anonymous interactions* such as mean field games, D-SPAIT, and Collective Dec-POMDPs (Jovanovic & Rosenthal, 1988; Kizilkale & Caines, 2012; Varakantham et al., 2014; Robbel et al., 2016; Nguyen et al., 2017; Subramanian & Mahajan, 2019). These models assume that the interactions between a large set of agents are governed by low dimensional statistics that capture how the rest of the population influences each individual. For instance, in disease propagation, only the *number* (not the identity) of people that are infected in one's neighborhood might matter (Robbel et al., 2016). As argued in the beginning of Section 5, the ability to include intra-stage connections into the IBA framework can enable us to model most (if not all) such problems in the IBA framework. So far, however, we have not yet identified how this

can lead to compact influence representations (as described in Section 7) or more efficient approaches to solving these games.

Bazinin and Shani (2018) investigate exploiting a heuristic form of influence in deterministic multiagent planning problems, formalized in the qualitative Dec-POMDP (Brafman et al., 2013) framework. Their approach plans "per agent": first each agent computes a plan assuming the other agents execute the actions that are most beneficial for it. This creates constraints (influences) for the other agents. Then the next agent gets to plan, subject to these constraints, and the process iterates.

In this work, we show how we can define a local-form model based upon a factored POSG model and a specification of a local state function S. We have not touched the question of how to define this local state function. E.g., in a multi-robot cleaning task, each agent potentially could clean every location, leading to local models as large as the original problem. To counter this, one can apply *organizations* (Carley & Gasser, 1999; Ferber, Gutknecht, & Michel, 2004; Vázquez-Salceda, Dignum, & Dignum, 2005) which effectively constrain which agents can address what parts of the problems. Sleight and Durfee (2012, 2015) investigate such organizations in a decision-theoretic context, also taking into account simpler, approximate, forms of influence. Our definition of influence is different as it critically depends on the d-separating set, which is not considered by Sleight and Durfee. Claes, Oliehoek, Baier, and Tuyls (2017) use heuristics from multi-robot task allocation (Gerkey & Mataric, 2003).

Other models (Spaan & Melo, 2008; Varakantham et al., 2009; Melo & Veloso, 2010, 2011) have allowed for approximate decoupled local planning by leveraging a form of context-specific independence, where agents only influence each other in certain states. An important direction of research is to also exploit this type of independence in LFMs. Similar ideas have been considered in the multiagent RL setting too (Melo & Veloso, 2009; De Hauwere, Vrancx, & Nowé, 2010). Structured interactions between agents are starting to be used for examining concepts like understanding by agents (Corona, Alaniz, & Akata, 2019).

Approaches that take the perspective of a protagonist agent, like the recursive modeling method (Gmytrasiewicz & Durfee, 1995, 2000), I-POMDPs (Gmytrasiewicz & Doshi, 2005), and work on ad-hoc teams (Stone, Kaminka, Kraus, & Rosenschein, 2010; Albrecht & Ramamoorthy, 2013) inherently provide a subjective perspective, which can include modeling other agents recursively, that is conceptually related to our notion of the local model. Although the formal definition of these models is different from the fPOSG, our definition of influence (and thus IBA) is readily applicable to factored-state versions of these models, and therefore IBA can be extended to such models. Specifically, influence-based abstraction is conceptually similar to existing approaches that exploit *behavioral equivalence* (Pynadath & Marsella, 2007; Rathnasabapathy, Doshi, & Gmytrasiewicz, 2006), but these approaches abstract classes of behaviors down to policies, whereas we abstract policies down to even more abstract influences. These relations are illustrated in Figure 15.

Also in multiagent learning, the idea that abstract representations of influence exist and can help in learning are starting to be considered (Hernandez-Leal et al., 2017). For instance, Claes, Robbel, Oliehoek, Hennes, Tuyls, and Van der Hoek (2015) investigated an approximate form of influence of team mates in collaborative spatial task allocation problems. Foerster, Nardelli, Farquhar, Afouras, Torr, Kohli, and Whiteson (2017) propose 'fingerprints' (episode indices) for when data was collected to capture non-stationary due

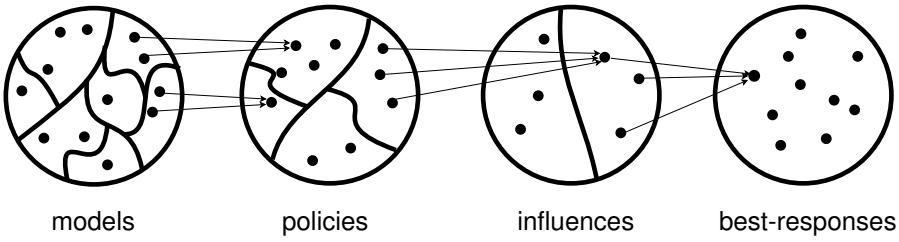


Figure 15: Many (e.g., I-POMDP) models for agent j may be *behaviorally equivalent,* i.e., map to the same policy π_j. In turn, many policies π_j can lead to the same influence $I_{\rightarrow i}$ on agent i. Finally, many influences may map to the same best-response π_i. (Note that only a small part of the space of π_i may be a best-response to some influence/policy/model.)

to the changing opponent strategy. Hong, Su, Shann, Chang, and Lee (2018) propose to augment DQN (Mnih et al., 2015) with a module to learn 'policy features' based on observations of the actions of other agents. Jaques, Lazaridou, Hughes, Gulcehre, Ortega, Strouse, Leibo, and De Freitas (2019) propose to use a mutual information-based version of influence (similar to Allen & Zilberstein, 2007, discussed above) as an auxiliary reward, and Wang, Wang, Wu, and Zhang (2020) extended this to direct exploration in multiagent reinforcement learning. To some extent, all forms of agent modeling (e.g. Hernandez-Leal et al., 2017; Hernandez-Leal, Kartal, & Taylor, 2019; Tacchetti, Song, Mediano, Zambaldi, Kramár, Rabinowitz, Graepel, Botvinick, & Battaglia, 2019) or tracking (Sunberg, Ho, & Kochenderfer, 2017) can be seen as a some form of influence prediction, since one can think of the action of another agent as an influence source. However, few of these approaches further formalize the structure of this interaction, which means that they have not exploited the insight that one only may need to remember a subset of variables, even though this can lead to significant improvements (Suau de Castro et al., 2019a).

8.3 Other Forms of Abstraction

Influence-based abstraction is a form of state abstraction, which has a long tradition in AI planning and learning (e.g., Sacerdoti, 1974; Knoblock, 1993; McCallum, 1993; Dearden & Boutilier, 1997; Dean & Givan, 1997; Hoey, St-Aubin, Hu, & Boutilier, 1999; Givan, Leach, & Dean, 2000; Boutilier, Dearden, & Goldszmidt, 2000; Ravindran & Barto, 2003; Jong & Stone, 2005; Konidaris & Barto, 2009; Kaelbling & Lozano-Perez, 2012; Hostetler, Fern, & Dietterich, 2014; Anand, Noothigattu, Mausam, & Singla, 2016; Bai, Srivastava, & Russell, 2016; Abel, Arumugam, Asadi, Jinnai, Littman, & Wong, 2019) . Other types of abstraction (Mahadevan, 2010) are temporal abstractions, such as options and macro-actions (Sutton, Precup, & Singh, 1999; Theocharous & Kaelbling, 2004; Amato, Konidaris, Kaelbling, & How, 2019; Machado, Bellemare, & Bowling, 2017), and functional abstraction, which tries to identify appropriate basis functions (Keller, Mannor, & Precup, 2006; Parr, Painter-Wakefield, Li, & Littman, 2007; Mahadevan & Maggioni, 2007; Petrik, 2007), including the huge body of recent work on deep RL (Schmidhuber, 1991; Mnih et al., 2015; François-Lavet, Henderson, Islam, Bellemare, & Pineau, 2018). We will focus on related work on state abstraction.

Different manners of performing state abstraction in MDPs exist, such as state aggregation methods which cluster similar states together, or starting with one abstract state and subsequently splitting (Givan et al., 2003), or removing state factors with no impact on the policy or rewards (Jong & Stone, 2005; Dearden & Boutilier, 1997). At a technical level these approaches are based on the idea that the original MDP and the abstraction are bisimilar. MDP homomorphisms (Ravindran & Barto, 2002, 2003) generalize the idea of bisimilarity to also consider similarity of different actions. These ideas can also be used as metrics (Ferns, Panangaden, & Precup, 2004; Ferns & Precup, 2014), and many of these ideas lie at the core of recent model-based (deep) RL approaches (Corneil, Gerstner, & Brea, 2018; Gelada, Kumar, Buckman, Nachum, & Bellemare, 2019; Biza & Platt, 2019; Van der Pol, Kipf, Oliehoek, & Welling, 2020).

Other methods implement abstraction as part of the solution method (Hoey et al., 1999; Boutilier et al., 2000; St-Aubin, Hoey, & Boutilier, 2001). Different notions of which states to group together exist. Li et al. (2006) present a unifying framework that discriminates a number of types of exact state abstraction, and some of these were recently extended to approximate state abstractions (Abel et al., 2016). The introduced notions of model irrelevance/similarity are particularly relevant: they group together states that behave (approximately) identical in terms of rewards and transitions, which is also what IBA achieves in its influence-augmented local model.

However, there is one big difference between all these methods and the influence-based abstraction: in order to achieve a good approximation, all the previous notions can only group states together that have very similar (usually measured in L1 norm of) transition probabilities, which severely limits their applicability. Existing methods can generally not abstract away an entire state variable that is an influence source and still provide guarantees of near optimality. In contrast, IBA does enable abstracting away such influence sources, and thus groups together states that can have very different transition probabilities. IBA corrects for this by incorporating the influence in the IALM, by means of the dependence on the d-separating set D_i.

Another body of work casts abstracted, non-Markovian, models as models with imprecise probabilities (Givan et al., 2000; Iyengar, 2005; Sanner, Uther, & Delgado, 2010; Delgado, Sanner, & de Barros, 2011b; Delgado, de Barros, Cozman, & Sanner, 2011a; Petrik & Subramanian, 2014; Delgado, de Barros, Dias, & Sanner, 2016). These typically place intervals on the transition probabilities and compute 'robust' policies that give the optimal worst case (with respect to the realized transition probabilities) payoff. Essentially these models are equivalent to a two-player zero-sum game where the agent faces an adversarial environment that chooses the transition probabilities to sabotage the agent (Iyengar, 2005). A disadvantage of such approaches is that they are only useful if the uncertainty intervals are sufficiently small and, as above, this is very hard to guarantee when abstracting away entire state variables. As such, the contribution of IBA is complimentary: it shows that it is possible to create abstract models which have no uncertainty interval at all.

IBA also bears some similarity to the framework of mixed-observability MDPs (Ong, Png, Hsu, & Lee, 2009, 2010), which splits the state $s = \langle o,l \rangle$ into observable state factors o and hidden ones l. IBA, however, splits $s = \langle x_i,y_i \rangle$ into modeled x_i and non-modeled factors y_i. As such, the frameworks are complimentary: the local state space of an agent

after performing IBA can have mixed observability[21] and the hidden part l of a mixed-observability MDPs can be abstracted by using IBA.

Finally, abstractions have also been investigated as the basis for robotic decision making (Konidaris, Kaelbling, & Lozano-Pérez, 2018) and multi-robot decision making (Le & Plaku, 2018; Amato et al., 2019). These methods typically combine temporal and state abstraction. Specifically, Konidaris et al. (2018) focus on learning abstract state representation that support open loop planning using a given set of 'skills' (also called 'options', Sutton et al., 1999, or 'macro actions', Amato et al., 2019) and demonstrate this on a robot. While the formalization allows for probabilistic effects (they can reason about probability that the plan is executable), they assume that the skills are such that the effect of a skill σ does not depend on the previous state, such that $\Pr(s'|\sigma)$ is well defined. In practice, the approach typically requires small sets of states s' with positive support, or the probability of executability drops. As such, the framework is less suited for highly stochastic environments, such as those affected by other agents or other type of exogenous events (Boutilier et al., 1999). Thus, again, our work here is complementary, since it shows what parts of history may need to be retained to decrease this stochasticity. Le and Plaku (2018) focus on multi-robot motion planning. The difficulty here is to reason both about detailed motions, as well as the presence of multiple robots. To deal with this they propose to reason about the interaction (making use of multiagent path planning) in an abstract representation, this high-level plan is then used as a heuristic for the low-level motion planning. Amato et al. (2019) formalize hierarchical Dec-POMDPs, called Mac-Dec-POMDP (for 'macro-action') where multiple agents act using options. The focus of this work lies on how to plan with options in the Dec-POMDP setting, but the abstractions at higher levels are assumed to be given.

9. Conclusion, Discussion and Future Work

This paper makes a theoretical contribution to the field of decision making in factored multiagent settings by giving a rigorous definition of *influence-based abstraction (IBA)* in such settings. It defines a notion of 'influence' that enables an agent in a POMDP to perform a lossless abstraction of the decision making problem it faces. That is, we prove that, for a given abstraction in terms of a *local-form model*, an *influence point* is a sufficient statistic for the part of the problem that is abstracted away. The local-form model and influence point together induce what we call an *influence-augmented local model (IALM)*: a local model that is sufficient to compute an exact best response.

The proof of sufficiency also serves a practical purpose: it isolates the core technical property (in Section 6.2) that needs to hold for sufficiency. In this way it conveys insight into the nature of *how* abstraction of latent state factors affects value, provides a derivation that can be used to obtain simplifications of the definition of influence in simpler cases, and provides a recipe of how to prove similar results in more general cases.

At a higher level, IBA is important for the following reasons:

1. The theory presented in this paper presents a new perspective on abstraction in structured settings: it shows that such abstractions can be seen as special cases of POMDPs,

21. While the non-modeled factors y_i are hidden, (some of) the modeled state factors x_i can be fully observed: in our formalism such observability of a factor x^k would be modeled by introducing an observation factor that has x^k as its only parent and has the identity function as its conditional probability table.

where one only needs to remember about a subset of variables. Effectively, this can create a problem class in between MDPs and POMDPs: In this class, in order to predict the local dynamics, we will need to use memory, but this memory only needs to store information about the history of a subset of state variables.

2. It can enable more efficient best-response computation in fPOSGs, as well as providing a very natural form of approximation via approximate inference.
3. It provides a better understanding of previously identified sub-classes of fPOSGs (Becker et al., 2003, 2004; Nair et al., 2005; Petrik & Zilberstein, 2009; Oliehoek, 2010; Kumar et al., 2011) and how they relate to each other. The insightful connections that we have drawn promote extensions of specialized methods beyond their respective sub-classes as well as comparisons with one another in more general contexts. For instance, our work has identified a compact representation of influences in ND-POMDPs (where none was known) and identified a more compact representation for EDI-Dec-MDPs.
4. It demonstrates how the value function for essentially *any* factored Dec-POMDP can be decomposed into the sum of a number of *local* value functions. As such, IBA demonstrates that all such problems satisfy a weak form of locality of interaction (also 'value factorization')—a property that is exploited in several Dec-POMDP solution methods (Nair et al., 2005; Oliehoek, 2010; Kumar et al., 2011) and multiagent RL papers (Guestrin et al., 2002b; Kok & Vlassis, 2006; Kuyer et al., 2008; Van der Pol & Oliehoek, 2016; Sunehag et al., 2018; Rashid et al., 2018; Castellini et al., 2019; Böhmer et al., 2019; Son et al., 2019; Wang et al., 2019).
5. Influences can provide a more compact, yet sufficient statistic for the behavior of other agents in a MAS. We expect this to be important in multiagent reinforcement learning, since it is often easier to learn a compact statistic from the same amount of data.

We emphasize that this definition of influence is not a magic bullet: while the influence-augmented local model is sufficient to compute a best-response locally, the computation of the required influence point itself is an intractable inference problem in general. However, in certain cases where this problem *is* feasible it can enable faster best-response computations and search for multiagent plans via *influence search* (Witwicki & Durfee, 2010b; Witwicki et al., 2012). As such, an important direction of future work would investigate how the definition of influence presented in this paper can support influence search in more general settings.

Moreover, even in cases where influences are intractable to compute, the concept forms the basis for principled approximations. For instance, by being *optimistic* with respect to the influence sources, one is able to compute upper bounds on the optimal value of Dec-POMDPs with hundreds of agents, thus leading to firm guarantees on the quality of heuristic solutions (Oliehoek et al., 2015a). Furthermore, there is evidence, in the context of deep reinforcement learning, that such approximate versions of influence may in some problems improve learning, both in terms of speed as well as performance (Suau de Castro et al., 2019b). As such, a fruitful direction of research is to better understand such approximate characterization of influence (Congeduti et al., 2020). This article has provided the foundations for such an exploration.

An important direction of of future research would explore the applications of (approximate) forms of influence. For instance, it is possible that these can make a huge impact on

human-robot interactions (Shah, Wiken, Williams, & Breazeal, 2011; Nikolaidis, Ramakrishnan, Gu, & Shah, 2015). We note that even though the discussion in this paper was based on the more general case of multiagent systems, there is nothing that stops us from applying IBA in complex systems with just a single agent. As a case in point, Suau de Castro et al. (2019b) show improvements of learning on a single traffic intersection and on Atari games.

Finally, in this paper the discussion is limited to settings where structure is known. *If* we have structure, this can be exploited to define influence, possibly leading to more efficient best responses, or other benefits. Certainly, in many cases knowledge of the structure is not available. Future work could try to build off the advances in structure learning algorithms (Murphy, 2002; Koller & Friedman, 2009; Doshi-Velez, Wingate, Tenenbaum, & Roy, 2011; Murphy, 2012) and their integration with decision making problems (Degris, Sigaud, & Wuillemin, 2006; Strehl, Diuk, & Littman, 2007; Walsh, Goschin, & Littman, 2010; Doshi-Velez, 2009; Littman, 2012; Katt, Oliehoek, & Amato, 2019); as long as it is possible to learn a model our methods would apply. In particular, even though such an estimated model might be inaccurate, its reduction to an IALM would add no further estimation error. This observation may open up a new research direction in sequential decision making that forsakes approximate solution methods (e.g., Monte Carlo tree search, Browne, Powley, Whitehouse, Lucas, Cowling, Rohlfshagen, Tavener, Perez, Samothrakis, & Colton, 2012, RL techniques like DQN, Mnih et al., 2015, or other forms of approximate dynamic programming, Bertsekas, 2005, 2007; Powell, 2012) in favor of learning useful approximate models that give structured representations of interactions. Arguably, such approaches that make exact use of assumed (but approximate) models lie at the basis of many, if not most, engineering disciplines and thus served human intelligence well in the past.

Acknowledgments

We would like to thank Elena Congeduti, Rolf Starre and Miguel Suau, and Mikko Lauri for their helpful comments. Major parts of this work where performed while F. A. Oliehoek was affiliated with MIT, University of Amsterdam, and University of Liverpool, and while S. Witwicki was affiliated with EPFL. This paper is the result of research that received funding from various funding agencies, including AFOSR (MURI), NWO (VENI), and the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 758824—INFLUENCE).

Appendix A. Proofs and Derivations

Here we give proofs and derivations of a number of results. These are referred from the main text, and will be stated here without further explanation.

A.1 GFBRMs

A.1.1 Expected Reward

$$\begin{aligned}
R_i(b_i^g,a_i^t) &= \mathbf{E}_{\bar{s}_i^t\sim b_i^g,\bar{s}_i^{t+1}\sim\bar{T}(\bar{s}_i^t,a_i^t,\cdot)}\left[\bar{R}_i(\bar{s}_i^t,a_i^t,\bar{s}_i^{t+1})\right] \\
&= \sum_{\bar{s}_i^t} b_i^g(\bar{s}_i^t)\sum_{\bar{s}_i^{t+1}}\bar{T}(\bar{s}_i^{t+1}|\bar{s}_i^t,a_i^t)\bar{R}_i(\bar{s}_i^t,a_i^t,\bar{s}_i^{t+1}) \\
&= \sum_{\langle s^t,\vec{h}_{-i}^t\rangle} b_i^g(\langle s^t,\vec{h}_{-i}^t\rangle)\sum_{\langle s^{t+1},\vec{h}_{-i}^{t+1}\rangle}\Pr(\langle s^{t+1},\vec{h}_{-i}^{t+1}\rangle|\langle s^t,\vec{h}_{-i}^t\rangle,a_i)\bar{R}_i(\langle s^t,\vec{h}_{-i}^t\rangle,a_i,\langle s^{t+1},\vec{h}_{-i}^{t+1}\rangle) \\
&= \sum_{\langle s^t,\vec{h}_{-i}^t\rangle} b_i^g(\langle s^t,\vec{h}_{-i}^t\rangle)\sum_{s^{t+1}}\sum_{a_{-i}}\sum_{o_{-i}^{t+1}}\Pr(s^{t+1},a_{-i},o_{-i}^{t+1}|\langle s^t,\vec{h}_{-i}^t\rangle,a_i)R_i(s^t,a_i,a_{-i},s^{t+1}) \\
&= \sum_{\langle s^t,\vec{h}_{-i}^t\rangle} b_i^g(\langle s^t,\vec{h}_{-i}^t\rangle)\sum_{s^{t+1}}\sum_{a_{-i}}\Pr(s^{t+1},a_{-i}|\langle s^t,\vec{h}_{-i}^t\rangle,a_i)R_i(s^t,a_i,a_{-i},s^{t+1}) \\
&= \sum_{s^t}\sum_{s^{t+1}}\sum_{a_{-i}}\Pr(s^{t+1}|s^t,a)R_i(s^t,a,s^{t+1})\sum_{\vec{h}_{-i}^t}\Pr(a_{-i}|\vec{h}_{-i}^t)b_i^g(s^t,\vec{h}_{-i}^t)
\end{aligned}$$

A.1.2 Expected Observation Probability

$\Pr(o_i^{t+1}|b_i^g,a_i^t)$

$$\begin{aligned}
&= \mathbf{E}_{\bar{s}_i^t\sim b_i^g,\bar{s}_i^{t+1}\sim\bar{T}(\bar{s}_i^t,a_i^t,\cdot)}\left[\bar{O}(o_i^{t+1}|a_i^t,\bar{s}_i^{t+1})\right] \\
&= \sum_{\bar{s}_i^t} b_i^g(\bar{s}_i^t)\sum_{\bar{s}_i^{t+1}}\bar{T}(\bar{s}_i^{t+1}|\bar{s}_i^t,a_i^t)\bar{O}(o_i^{t+1}|a_i^t,\bar{s}_i^{t+1}) \\
&= \sum_{\langle s^t,\vec{h}_{-i}^t\rangle} b_i^g(\langle s^t,\vec{h}_{-i}^t\rangle)\sum_{\langle s^{t+1},\vec{h}_{-i}^{t+1}\rangle}\Pr(\langle s^{t+1},\vec{h}_{-i}^{t+1}\rangle|\langle s^t,\vec{h}_{-i}^t\rangle,a_i)\Pr(o_i|a_i,\langle s^{t+1},\vec{h}_{-i}^{t+1}\rangle) \\
&= \sum_{\langle s^t,\vec{h}_{-i}^t\rangle} b_i^g(\langle s^t,\vec{h}_{-i}^t\rangle)\sum_{s^{t+1}}\sum_{a_{-i}}\sum_{o_{-i}^{t+1}}\Pr(s^{t+1},a_{-i},o_{-i}^{t+1}|\langle s^t,\vec{h}_{-i}^t\rangle,a_i)\Pr(o_i|a_i,a_{-i},s^{t+1},o_{-i}^{t+1}) \\
&= \sum_{\langle s^t,\vec{h}_{-i}^t\rangle} b_i^g(\langle s^t,\vec{h}_{-i}^t\rangle)\sum_{s^{t+1}}\sum_{a_{-i}}\sum_{o_{-i}^{t+1}}\Pr(s^{t+1}|s^t,a_{-i},a_i)\Pr(a_{-i}|\vec{h}_{-i}^t,\pi_{-i})\Pr(o_{-i}^{t+1}|a_i,a_{-i},s^{t+1}) \\
&\quad \Pr(o_i|a_i,a_{-i},s^{t+1},o_{-i}^{t+1}) \qquad (A.1)
\end{aligned}$$

$$= \sum_{\langle s^t,\vec{h}^t_{-i}\rangle} b^g_i(\langle s^t,\vec{h}^t_{-i}\rangle) \sum_{s^{t+1}} \sum_{a_{-i}} \sum_{o^{t+1}_{-i}} \Pr(s^{t+1}|s^t,a_{-i},a_i) \Pr(a_{-i}|\vec{h}^t_{-i},\pi_{-i}) \Pr(o^{t+1}_{-i}|a_i,a_{-i},s^{t+1})$$
$$\frac{\Pr(o_i,o^{t+1}_{-i}|a_i,a_{-i},s^{t+1})}{\Pr(o^{t+1}_{-i}|a_i,a_{-i},s^{t+1})}$$
$$= \sum_{\langle s^t,\vec{h}^t_{-i}\rangle} b^g_i(\langle s^t,\vec{h}^t_{-i}\rangle) \sum_{s^{t+1}} \sum_{a_{-i}} \sum_{o^{t+1}_{-i}} \Pr(s^{t+1}|s^t,a) \Pr(a_{-i}|\vec{h}^t_{-i},\pi_{-i}) \Pr(o_i,o^{t+1}_{-i}|a_i,a_{-i},s^{t+1})$$
$$= \sum_{s^t} \sum_{s^{t+1}} \sum_{a_{-i}} \sum_{o^{t+1}_{-i}} \Pr(s^{t+1}|s^t,a) \Pr(o^{t+1}|a,s^{t+1}) \sum_{\vec{h}^t_{-i}} \Pr(a_{-i}|\vec{h}^t_{-i},\pi_{-i}) b^g_i(s^t,\vec{h}^t_{-i}) \tag{A.2}$$

A.2 LFMs

A.2.1 Expected Reward

Starting with (3.5), we have that $R_i(b^g_i,a_i)$

$$= \sum_{s^t} \sum_{s^{t+1}} \sum_{a_{-i}} \Pr(s^{t+1}|s^t,a) R_i(s^t,a,s^{t+1}) \sum_{\vec{h}^t_{-i}} \Pr(a_{-i}|\vec{h}^t_{-i},\pi_{-i}) b^g_i(s^t,\vec{h}^t_{-i})$$
$$= \sum_{s^t} \sum_{s^{t+1}} \sum_{a_{-i}} \Pr(s^{t+1}|s^t,a_i,a_{-i}) R_i(s^t,a,s^{t+1}) \sum_{\vec{h}^t_{-i}} \Pr(a_{-i}|\vec{h}^t_{-i},\pi_{-i}) b^g_i(s^t,\vec{h}^t_{-i})$$
$$= \{\text{restrict to actual dependencies of } R_i\}$$
$$\sum_{s^t} \sum_{s^{t+1}} \sum_{a_{-i}} \Pr(s^{t+1}|s^t,a_i,a_{-i}) R_i(x^t_i,a_i,x^{t+1}_i) \sum_{\vec{h}^t_{-i}} \Pr(a_{-i}|\vec{h}^t_{-i},\pi_{-i}) b^g_i(s^t,\vec{h}^t_{-i})$$
$$= \sum_{s^t} \sum_{a_{-i}} \sum_{x^{t+1}_i,y^{t+1}_i} \Pr(x^{t+1}_i,y^{t+1}_i|s^t,a_i,a_{-i}) R_i(x^t_i,a_i,x^{t+1}_i) \sum_{\vec{h}^t_{-i}} \Pr(a_{-i}|\vec{h}^t_{-i},\pi_{-i}) b^g_i(s^t,\vec{h}^t_{-i})$$

$$= \{\text{via (3.9)}\}$$
$$\sum_{x^t_i,y^t_i} \sum_{a_{-i}} \sum_{x^{t+1}_i} \Pr(x^{t+1}_i|s^t,a_i,a_{-i}) R_i(x^t_i,a_i,x^{t+1}_i) \sum_{\vec{h}^t_{-i}} \Pr(a_{-i}|\vec{h}^t_{-i},\pi_{-i}) b^g_i(s^t,\vec{h}^t_{-i})$$
$$= \sum_{x^t_i} \sum_{x^{t+1}_i} R_i(x^t_i,a_i,x^{t+1}_i) \sum_{y^t_i} \sum_{a_{-i}} \Pr(x^{t+1}_i|s^t,a_i,a_{-i}) \sum_{\vec{h}^t_{-i}} \Pr(a_{-i}|\vec{h}^t_{-i},\pi_{-i}) b^g_i(s^t,\vec{h}^t_{-i})$$
$$= \sum_{x^t_i} \sum_{x^{t+1}_i} R_i(x^t_i,a_i,x^{t+1}_i) \Pr(x^t_i,x^{t+1}_i|b^g_i,a^t_i,\pi_{-i}), \tag{A.3}$$

where we implicitly defined (remember $s^t = \langle x^t_i,y^t_i\rangle$)

$$\Pr(x^t_i,x^{t+1}_i|b^g_i,a^t_i,\pi_{-i}) \triangleq \sum_{y^t_i} \sum_{a_{-i}} \Pr(x^{t+1}_i|s^t,a_i,a_{-i}) \sum_{\vec{h}^t_{-i}} \Pr(a_{-i}|\vec{h}^t_{-i},\pi_{-i}) b^g_i(s^t,\vec{h}^t_{-i}) \tag{A.4}$$

A.2.2 Expected Observation Probability

In this case, the expected observation probability $\Pr(o_i^{t+1}|b_i^g,a_i)$ equals

$$
\begin{aligned}
&= \sum_{s^t}\sum_{s^{t+1}}\sum_{a_{-i}}\sum_{o_{-i}^{t+1}} \Pr(s^{t+1}|s^t,a)\Pr(o^t|a,s^{t+1}) \sum_{\vec{h}_{-i}^t} \Pr(a_{-i}|\vec{h}_{-i}^t,\pi_{-i}) b_i^g(s^t,\vec{h}_{-i}^t) \\
&= \sum_{s^{t+1}}\sum_{s^t}\sum_{a_{-i}}\sum_{o_{-i}^{t+1}} \Pr(o_i^{t+1},o_{-i}^{t+1}|a_i,a_{-i},s^{t+1})\Pr(s^{t+1}|s^t,a_i,a_{-i}) \\
&\quad \sum_{\vec{h}_{-i}^t} \Pr(a_{-i}|\vec{h}_{-i}^t,\pi_{-i}) b_i^g(s^t,\vec{h}_{-i}^t) \\
&= \{\text{marginalize}\} \\
&\quad \sum_{s^{t+1}}\sum_{s^t}\sum_{a_{-i}} \Pr(o_i^{t+1}|a_i,a_{-i},s^{t+1})\Pr(s^{t+1}|s^t,a_i,a_{-i}) \sum_{\vec{h}_{-i}^t} \Pr(a_{-i}|\vec{h}_{-i}^t,\pi_{-i}) b_i^g(s^t,\vec{h}_{-i}^t) \\
&= \{\text{restrict to actual dependencies}\} \\
&\quad \sum_{s^{t+1}}\sum_{s^t}\sum_{a_{-i}} \Pr(o_i^{t+1}|a_i,x_i^{t+1})\Pr(s^{t+1}|s^t,a_i,a_{-i}) \sum_{\vec{h}_{-i}^t} \Pr(a_{-i}|\vec{h}_{-i}^t,\pi_{-i}) b_i^g(s^t,\vec{h}_{-i}^t) \\
&= \sum_{x_i^{t+1},y_i^{t+1}}\sum_{s^t}\sum_{a_{-i}} \Pr(o_i^{t+1}|a_i,x_i^{t+1})\Pr(x_i^{t+1},y_i^{t+1}|s^t,a_i,a_{-i}) \sum_{\vec{h}_{-i}^t} \Pr(a_{-i}|\vec{h}_{-i}^t,\pi_{-i}) b_i^g(s^t,\vec{h}_{-i}^t) \\
&= \sum_{x_i^{t+1}} \Pr(o_i^{t+1}|a_i,x_i^{t+1}) \sum_{s^t}\sum_{y_i^{t+1}}\sum_{a_{-i}} \Pr(x_i^{t+1},y_i^{t+1}|s^t,a_i,a_{-i}) \sum_{\vec{h}_{-i}^t} \Pr(a_{-i}|\vec{h}_{-i}^t,\pi_{-i}) b_i^g(s^t,\vec{h}_{-i}^t) \\
&= \sum_{x_i^{t+1}} \Pr(o_i^{t+1}|a_i,x_i^{t+1}) \Pr(x_i^{t+1}|b_i^g,a_i,\pi_{-i}) \qquad \text{(A.5)}
\end{aligned}
$$

where we implicitly defined

$$
\Pr(x_i^{t+1}|b_i^g,a_i) \triangleq \sum_{s^t}\sum_{a_{-i}} \Pr(x_i^{t+1}|s^t,a_i,a_{-i}) \sum_{\vec{h}_{-i}^t} \Pr(a_{-i}|\vec{h}_{-i}^t\pi_{-i}) b_i^g(s^t,\vec{h}_{-i}^t). \qquad \text{(A.6)}
$$

A.3 IALMs

A.3.1 Expected Observation Probability

$$
\begin{aligned}
\Pr(o_i^{t+1}|b_i^l,a_i^t) &= \mathbf{E}_{\bar{s}_i^t\sim b_i^l,\bar{s}_i^{t+1}\sim\bar{T}(\bar{s}_i^t,a_i^t,\cdot)}\left[\bar{O}(o_i^{t+1}|a_i^t,\bar{s}_i^{t+1})\right] \\
&= \sum_{\bar{s}_i^t} b_i^l(\bar{s}_i^t) \sum_{\bar{s}_i^{t+1}} \bar{T}(\bar{s}_i^{t+1}|\bar{s}_i^t,a_i^t)\bar{O}(o_i^{t+1}|a_i^t,\bar{s}_i^{t+1})
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{x_i^t, D_i^{t+1}} b_i^l(x_i^t, D_i^{t+1}) \sum_{x_i^{t+1}, D_i^{t+2}} \Pr(x_i^{t+1}, D_i^{t+2} | x_i^t, D_i^{t+1}, a_i^t, I_{\to i}^{t+1}) \Pr(o_i^{t+1} | a_i^t, x_i^{t+1}) \\
&= \sum_{x_i^t, D_i^{t+1}} b_i^l(x_i^t, D_i^{t+1}) \sum_{x_i^{t+1}} \Pr(x_i^{t+1} | x_i^t, D_i^{t+1}, a_i^t, I_{\to i}^{t+1}) \Pr(o_i^{t+1} | a_i^t, x_i^{t+1}) \\
&= \sum_{x_i^{t+1}} \Pr(o_i^{t+1} | a_i^t, x_i^{t+1}) \left[\sum_{x_i^t, D_i^{t+1}} \Pr(x_i^{t+1} | x_i^t, D_i^{t+1}, a_i^t, I_{\to i}^{t+1}) b_i^l(x_i^t, D_i^{t+1}) \right] \\
&= \sum_{x_i^{t+1}} \Pr(o_i^{t+1} | a_i^t, x_i^{t+1}) \Pr(x_i^{t+1} | b_i^l, a_i^t, I_{\to i}^{t+1}), \qquad \text{(A.7)}
\end{aligned}
$$

where we implicitly defined

$$
\Pr(x_i^{t+1} | b_i^l, a_i, I_{\to i}^{t+1}) \triangleq \sum_{x_i^t, D_i^{t+1}} \Pr(x_i^{t+1} | x_i^t, D_i^{t+1}, a_i^t, I_{\to i}^{t+1}) b_i^l(x_i^t, D_i^{t+1}). \qquad \text{(A.8)}
$$

(consistent with equation 4.12).

A.3.2 Expected Reward

$$
\begin{aligned}
R_i(b_i^l, a_i^t) &= \mathbf{E}_{\bar{s}_i^t \sim b_i^l, \bar{s}_i^{t+1} \sim \bar{T}(\bar{s}_i^t, a_i^t, \cdot)} \left[\bar{R}_i(\bar{s}_i^t, a_i^t, \bar{s}_i^{t+1}) \right] \\
&= \sum_{\bar{s}_i^t} b_i^l(\bar{s}_i^t) \sum_{\bar{s}_i^{t+1}} \bar{T}(\bar{s}_i^{t+1} | \bar{s}_i^t, a_i^t) \bar{R}_i(\bar{s}_i^t, a_i^t, \bar{s}_i^{t+1}) \\
&= \sum_{x_i^t, D_i^{t+1}} b_i^l(x_i^t, D_i^{t+1}) \sum_{x_i^{t+1}, D_i^{t+2}} \Pr(x_i^{t+1}, D_i^{t+2} | x_i^t, D_i^{t+1}, a_i^t, I_{\to i}^{t+1}) R_i(x_i^t, a_i^t, x_i^{t+1}) \\
&= \sum_{x_i^t, D_i^{t+1}} b_i^l(x_i^t, D_i^{t+1}) \sum_{x_i^{t+1}} \Pr(x_i^{t+1} | x_i^t, D_i^{t+1}, a_i^t, I_{\to i}^{t+1}) R_i(x_i^t, a_i^t, x_i^{t+1}) \\
&= \sum_{x_i^t} \sum_{x_i^{t+1}} R_i(x_i^t, a_i^t, x_i^{t+1}) \left[\sum_{D_i^{t+1}} \Pr(x_i^{t+1} | x_i^t, D_i^{t+1}, a_i^t, I_{\to i}^{t+1}) b_i^l(x_i^t, D_i^{t+1}) \right] \\
&= \sum_{x_i^t} \sum_{x_i^{t+1}} R_i(x_i^t, a_i^t, x_i^{t+1}) \Pr(x_i^t, x_i^{t+1} | b_i^l, a_i^t, I_{\to i}^{t+1}) \qquad \text{(A.9)}
\end{aligned}
$$

where we implicitly defined

$$
\Pr(x_i^t, x_i^{t+1} | b_i^l, a_i^t, I_{\to i}^{t+1}) \triangleq \sum_{D_i^{t+1}} \Pr(x_i^{t+1} | x_i^t, D_i^{t+1}, a_i^t, I_{\to i}^{t+1}) b_i^l(x_i^t, D_i^{t+1}) \qquad \text{(A.10)}
$$

(consistent with equation 4.16).

Appendix B. List of Acronyms

Acronym	description
2DBN	2-stage dynamic Bayesian network
AOH	action-observation history
CPT	conditional probability table
DBN	dynamic Bayesian network
Dec-MDP	decentralized Markov decision process
Dec-POMDP	decentralized partially observable Markov decision process
EDI-Dec-MDP	Dec-MDP with event-driven interactions
fDec-POMDP	factored Dec-POMDP
fPOSG	factored POSG
GFBRMs	global-form best-response model
IALM	Influence-augmented local model
IBA	influence-based abstraction
ISDs	intra-stage dependencies
LFM	local-form model
MDP	Markov decision process
ND-POMDP	network-distributed POMDP
NLAF	non-locally affected factor
NMF	non-modeled factor
OLAF	only-locally affected factor
POMDP	partially observable Markov decision process
POSG	partially observable stochastic game
RL	reinforcement learning
TD-POMDP	transition-decoupled POMDP
TI-Dec-MDP	transition-independent Dec-MDP

Appendix C. List of Notation

symbol	description
General	
$\mathbf{E}[\cdot]$	expectation
$\Delta(\cdot)$	set of probability distributions over $\cdot$
$\mathbf{1}_{\{\cdot,\cdot\}}$	denotes the Kronecker delta function
$(\cdot)_i$	a variable of interest $(\cdot)$ associated with agent i
$(\cdot)_{-i}$	a tuple of variables associated with all agents except i
$(\cdot)_i^t$	a variable of interest $(\cdot)$ associated with agent i at time step t
$(\cdot)^{k:t}$	partial history of values of $(\cdot)$ (e.g., $l_{tgt}^{k:t}$ is the history of target locations)
$\vec{(\cdot)}^t$	history of values of $(\cdot)$. I.e., $\vec{(\cdot)}^t = (\cdot)^{0:t}$
Models	
$\mathcal{M}^{POSG}$	A partially observable stochastic game (POSG)
$\mathcal{M}^{LFM}$	A local-form model: includes local state definitions for each agent
$\mathcal{M}_i^{GFBR}$	A global-form best-response model (GFBRM) for agent i
$\mathcal{M}_i^{IALM}$	An influence-augmented local model can be computed from an LFM when fixing other policies: $\mathcal{M}_i^{IALM}(\mathcal{M}^{LFM},\pi_{-i})$
Model components	
$\mathcal{D}$	the set of agents or (d)ecision makers
$\mathcal{S}$	set of (global) states
s	a global (i.e., Markov) state
$\mathcal{A}$	set of (joint) actions $\mathcal{A} = \mathcal{A}_1 \times \ldots \mathcal{A}_n$
a	a (joint) action $a = \langle a_1, \ldots, a_n \rangle$
T	transition function specifies $\Pr(s^{t+1}\|s^t,a^t)$
$\mathcal{R}$	set of reward functions
R_i	reward function of agent i
$\mathcal{O}$	set of (joint) observations $\mathcal{O} = \mathcal{O}_1 \times \cdots \times \mathcal{O}_n$
o	a (joint) observation $o = \langle o_1, \ldots, o_n \rangle$
O	observation function specifies $\Pr(o\|a,s')$
γ	the discount factor
H	horizon of the problem
b^0	initial state distribution: $b^0 \in \Delta(\mathcal{S})$
$\bar{\mathcal{S}}_i$	set of *augmented* states. E.g., in a GFBRM
$\bar{T}_i$,$\bar{R}_i$,etc.	transitions, rewards, etc. over augmented states
histories and beliefs	
$\vec{h}_i^t$	the action-observation history (AOH) of agent i at stage t
$\vec{\mathcal{H}}_i^t$	the set of AOHs of agent i at stage t
b	belief of a single POMDP agent $b(s) \triangleq \Pr(s\|b^0,\vec{h}^t)$

symbol	description
$BU()$	The belief update $b' = BU(b,a,o)$
b_i^g	global-form belief of agent i
b_i^l	local-form belief of agent i
π_i	policy of agent i
Value functions	
V^t	The optimal value function at stage t with $H-t$ stages-to-go
Q^t	The optimal action-value function, or 'Q-function'
Factored States	
$\mathcal{F}$	the set of state factors $\mathcal{F} = \{F^1,\ldots,F^{\lvert\mathcal{F}\rvert}\}$ in a factored model, we have that $\mathcal{S} = \mathcal{F}^1 \times \cdots \times \mathcal{F}^{\lvert\mathcal{F}\rvert}$
F^k	the k-th state factor
$\mathcal{F}^k$	the set of values $f^k \in \mathcal{F}^k$ that F^k can take
f^k	a value of state factor k
$\text{ORel}_i(F)$	observation relevant factor of agent i
$\text{RRel}_i(F)$	reward relevant factor of agent i
Local states of agent i	
$\mathcal{S}_i$	state space of agent i (general term: also outside LFMs)
s_i	state for agent i (general term: also outside LFMs)
$S(i)$	the local state function of an LFM for agent i: partitions $\mathcal{F}$ into modeled state factors x^k and non-modeled ones y^k
x^k	k-th modeled factor
$\mathring{x}^k$	k-th only-locally-affected factor (OLAF): a modeled factor that is not an influence destination
$\tilde{x}^k$	k-th a non-locally-affected factor (NLAF): a modeled factor that is an influence destination
$\mathcal{X}_i$	local state space (of *modeled* factors) in an LFM
x_i	local state of agent i in an LFM
y^k	a non-modeled factor
y_i	instantiation of all non-modeled factors. I.e., $s = \langle x_i,y_i\rangle$
$OLAF(i)$	the set of OLAFs
$NLAF(i)$	the set of NLAFs
Influence notation	
$u^t_{\rightarrow i}$	instantiation of all direct influence sources for stage t: $u^t_{\rightarrow i} = \langle y_u^{t-1},a_u^{t-1},y_u^t\rangle$
y_u^{t-1}	the (non-modeled) state factors that are direct influence sources
y_u^t	the (non-modeled) state factors that are direct intra-stage influence sources
a_u^{t-1}	the actions (of some subset of agents) that are direct influence sources
$\vec{h}_u^{t-1}$	the AOHs of those other agents whose action is an influence source (i.e., $\vec{h}_u^{t-1}$ involves the same agents as a_u^{t-1})

symbol	description
v	indirect sources: y_v^{t-1}, a_u^{t-1}, y_v^t can effect the direct sources $u_{\rightarrow i}^t$
w	union of direct and indirect sources: $w = u \cup v$; e.g., π_w is the joint policy of those agents whose action is either a direct or an indirect influence source
D_i^t	a d-separating set for agent i's influence at stage t
d	the d-set update function: $D_i^{t+2} = d(x_i^t, a_i^t, x_i^{t+1}, D_i^{t+1})$
σ	the d-set compression function $\sigma(D_i^{t+1})$ that computes a sufficient statistic for D_i^{t+1}
$I_{\rightarrow i}(\pi_{-i})$	$I_{\rightarrow i}(\pi_{-i}) = \left(I_{\rightarrow i}^1(\pi_{-i}), \ldots, I_{\rightarrow i}^H(\pi_{-i})\right)$ is an incoming influence point $I_{\rightarrow i}(\pi_{-i})$
$I_{\rightarrow i}^t(\pi_{-i})$	The incoming influence at stage t: a conditional probability distribution over values of the influence sources
$I(u_{\rightarrow i}^t \mid D_i^t)$	shorthand for $I_{\rightarrow i}^t(u_{\rightarrow i}^t \mid D_i^t, b^0, \pi_{-i}) = I_{\rightarrow i}^t(\pi_{-i})(u_{\rightarrow i}^t \mid D_i^t, b^0)$
$p_{I_{\rightarrow i}^{t+1}}$	influence-induced CPT that specifies $p_{I_{\rightarrow i}^{t+1}}(\tilde{x}^{t+1} \mid x_i^t, D_i^{t+1}, a_i)$

References

Abel, D., Arumugam, D., Asadi, K., Jinnai, Y., Littman, M. L., & Wong, L. L. (2019). State abstraction as compression in apprenticeship learning. In *Proc. of the Thirty-Third AAAI Conference on Artificial Intelligence*, pp. 3134–3142.

Abel, D., Hershkowitz, D. E., & Littman, M. L. (2016). Near optimal behavior via approximate state abstraction. In *Proc. of the Thirty-Third International Conference on Machine learning*, pp. 2915–2923.

Acid, S., & De Campos, L. M. (1996). An algorithm for finding minimum d-separating sets in belief networks. In *Proc. of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pp. 3–10.

Albrecht, S., & Ramamoorthy, S. (2013). A game-theoretic model and best-response learning method for ad hoc coordination in multiagent systems. In *Proc. of the Twelfth International Conference on Autonomous Agents and Multiagent Systems*, pp. 1155–1156.

Allen, M., & Zilberstein, S. (2007). Agent influence as a predictor of difficulty for decentralized problem-solving. In *Proc. of the National Conference on Artificial Intelligence*, pp. 688–693.

Allen, M., & Zilberstein, S. (2009). Agent influence and intelligent approximation in multiagent problems. In *Proc. of the International Conference on Intelligent Agent Technology*, pp. 311–314.

Amato, C., Konidaris, G., Kaelbling, L. P., & How, J. P. (2019). Modeling and planning with macro-actions in decentralized POMDPs. *Journal of Artificial Intelligence Research*, *64*, 817–859.

Anand, A., Noothigattu, R., Mausam, & Singla, P. (2016). OGA-UCT: on-the-go abstractions in UCT. In *Proc. of the Twenty-Sixth International Conference on Automated Planning and Scheduling*, pp. 29–37.

Bai, A., Srivastava, S., & Russell, S. J. (2016). Markovian state and action abstractions for MDPs via hierarchical MCTS. In *Proc. of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 3029–3039.

Bazinin, S., & Shani, G. (2018). Iterative planning for deterministic QDec-POMDPs. In *Proc. of the 4th Global Conference on Artificial Intelligence*, Vol. 55, pp. 15–28.

Becker, R., Zilberstein, S., & Lesser, V. (2004). Decentralized Markov decision processes with event-driven interactions. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems*, pp. 302–309.

Becker, R., Zilberstein, S., Lesser, V., & Goldman, C. V. (2003). Transition-independent decentralized Markov decision processes. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems*, pp. 41–48.

Bertsekas, D. P. (2005). *Dynamic Programming and Optimal Control* (3rd edition)., Vol. I. Athena Scientific.

Bertsekas, D. P. (2007). *Dynamic Programming and Optimal Control* (3rd edition)., Vol. II. Athena Scientific.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Biza, O., & Platt, R. (2019). Online abstraction with MDP homomorphisms for deep learning. In *Proc. of the Eighteenth International Conference on Autonomous Agents and Multiagent Systems*, pp. 1125–1133.

Böhmer, W., Kurin, V., & Whiteson, S. (2019). Deep coordination graphs. *arXiv e-prints*, arXiv:1910.00091.

Boutilier, C., Dean, T., & Hanks, S. (1999). Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, *11*, 1–94.

Boutilier, C., Dearden, R., & Goldszmidt, M. (2000). Stochastic dynamic programming with factored representations. *Artificial Intelligence*, *121*(1-2), 49–107.

Boyen, X., & Koller, D. (1998). Tractable inference for complex stochastic processes. In *Proc. of Uncertainty in Artificial Intelligence*, pp. 33–42.

Brafman, R. I., Shani, G., & Zilberstein, S. (2013). Qualitative planning under partial observability in multi-agent domains. In *Proc. of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, pp. 130–137.

Browne, C., Powley, E. J., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., & Colton, S. (2012). A survey of Monte Carlo tree search methods. *IEEE Trans. Comput. Intellig. and AI in Games*, *4*(1), 1–43.

Carley, K. M., & Gasser, L. (1999). Computational organization theory. In *Multiagent systems: A modern approach to distributed artificial intelligence*, pp. 299–330. MIT Press.

Castellini, J., Oliehoek, F. A., Savani, R., & Whiteson, S. (2019). The representational capacity of action-value networks for multi-agent reinforcement learning. In *Proc. of the Eighteenth International Conference on Autonomous Agents and Multiagent Systems*, pp. 1862–1864.

Chitnis, R., & Lozano-Pérez, T. (2020). Learning compact models for planning with exogenous processes. In *Proceedings of the Conference on Robot Learning*, pp. 813–822.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1724–1734.

Claes, D., Oliehoek, F. A., Baier, H., & Tuyls, K. (2017). Decentralised online planning for multi-robot warehouse commisioning. In *Proc. of the Sixteenth International Conference on Autonomous Agents and Multiagent Systems*, pp. 492–500.

Claes, D., Robbel, P., Oliehoek, F. A., Hennes, D., Tuyls, K., & Van der Hoek, W. (2015). Effective approximations for multi-robot coordination in spatially distributed tasks. In *Proc. of the Fourteenth International Conference on Autonomous Agents and Multiagent Systems*, pp. 881–890.

Congeduti, E., Mey, A., & Oliehoek, F. A. (2020). Loss bounds for approximate influence-based abstraction. *arXiv e-prints*, arXiv:2011.01788.

Corneil, D., Gerstner, W., & Brea, J. (2018). Efficient model-based deep reinforcement learning with variational state tabulation. In *Proc. of the 35th International Conference on Machine Learning*, pp. 1049–1058.

Corona, R., Alaniz, S., & Akata, Z. (2019). Modeling Conceptual Understanding in Image Reference Games. *arXiv e-prints*, arXiv:1910.04872.

De Hauwere, Y.-M., Vrancx, P., & Nowé, A. (2010). Learning multi-agent state space representations. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems*, pp. 715–722.

Dean, T., & Givan, R. (1997). Model minimization in Markov decision processes.. In *Proc. of the National Conference on Artificial Intelligence*, pp. 106–111.

Dean, T., Givan, R., & Leach, S. M. (1997). Model reduction techniques for computing approximately optimal solutions for Markov decision processes.. In *Proc. of Uncertainty in Artificial Intelligence*, pp. 124–131.

Dearden, R., & Boutilier, C. (1997). Abstraction and approximate decision-theoretic planning. *Artificial Intelligence*, *89*(1-2), 219–283.

Degris, T., Sigaud, O., & Wuillemin, P.-H. (2006). Learning the structure of factored Markov decision processes in reinforcement learning problems. In *Proc. of the Twenty-Third International Conference on Machine learning*, pp. 257–264.

Delgado, K. V., de Barros, L. N., Dias, D. B., & Sanner, S. (2016). Real-time dynamic programming for Markov decision processes with imprecise probabilities. *Artificial Intelligence*, *230*, 192–223.

Delgado, K. V., de Barros, L. N., Cozman, F. G., & Sanner, S. (2011a). Using mathematical programming to solve factored Markov decision processes with imprecise probabilities. *International Journal of Approximate Reasoning*, *52*(7), 1000–1017.

Delgado, K. V., Sanner, S., & de Barros, L. N. (2011b). Efficient solutions to factored MDPs with imprecise transition probabilities. *Artificial Intelligence*, *175*(9-10), 1498–1527.

Doshi-Velez, F. (2009). The infinite partially observable Markov decision process. In *Advances in Neural Information Processing Systems 22*, pp. 477–485.

Doshi-Velez, F., Wingate, D., Tenenbaum, J. B., & Roy, N. (2011). Infinite dynamic Bayesian networks. In *Proc. of the Twenty-Eighth International Conference on Machine learning*, pp. 913–920.

Ferber, J., Gutknecht, O., & Michel, F. (2004). From agents to organizations: An organizational view of multi-agent systems. In *Proc. of the 4th International Workshop on Agent-Oriented Software Engineering*, pp. 214–230.

Ferns, N., Panangaden, P., & Precup, D. (2004). Metrics for finite Markov decision processes. In *Proc. of the Twentieth Conference on Uncertainty in Artificial Intelligence*, pp. 162–169.

Ferns, N., & Precup, D. (2014). Bisimulation metrics are optimal value functions. In *Proc. of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pp. 210–219.

Foerster, J. N., Nardelli, N., Farquhar, G., Afouras, T., Torr, P. H. S., Kohli, P., & Whiteson, S. (2017). Stabilising experience replay for deep multi-agent reinforcement learning. In *Proc. of the Thirty-Fourth International Conference on Machine learning*, pp. 1146–1155.

François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., & Pineau, J. (2018). An introduction to deep reinforcement learning. *Foundations and Trends in Machine Learning*, *11*(3-4), 219–354.

Gelada, C., Kumar, S., Buckman, J., Nachum, O., & Bellemare, M. G. (2019). DeepMDP: Learning continuous latent space models for representation learning. In *Proc. of the Thirty-Sixth International Conference on Machine learning*, Vol. 97, pp. 2170–2179.

Gerkey, B. P., & Mataric, M. J. (2003). Multi-robot task allocation: Analyzing the complexity and optimality of key architectures. In *Proc. of the International Conference on Robotics and Automation*, pp. 3862–3868.

Givan, R., Dean, T., & Greig, M. (2003). Equivalence notions and model minimization in Markov decision processes. *Artificial Intelligence*, *14*(1–2), 163–223.

Givan, R., Leach, S. M., & Dean, T. L. (2000). Bounded-parameter Markov decision processes. *Artificial Intelligence*, *122*(1-2), 71–109.

Gmytrasiewicz, P. J., & Doshi, P. (2005). A framework for sequential planning in multiagent settings. *Journal of Artificial Intelligence Research*, *24*, 49–79.

Gmytrasiewicz, P. J., & Durfee, E. H. (1995). A rigorous, operational formalization of recursive modeling. In *Proc. of the First International Conference on Multiagent Systems*, pp. 125–132.

Gmytrasiewicz, P. J., & Durfee, E. H. (2000). Rational coordination in multi-agent environments. *Journal of Autonomous Agents and Multi-Agent Systems*, *3*(4), 319–350.

Graves, A., Mohamed, A.-r., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6645–6649. IEEE.

Guestrin, C., & Gordon, G. (2002). Distributed planning in hierarchical factored MDPs. In *Proc. of Uncertainty in Artificial Intelligence*, pp. 197–206.

Guestrin, C., Koller, D., & Parr, R. (2002a). Multiagent planning with factored MDPs. In *Advances in Neural Information Processing Systems 14*, pp. 1523–1530.

Guestrin, C., Lagoudakis, M., & Parr, R. (2002b). Coordinated reinforcement learning. In *Proc. of the International Conference on Machine Learning*, pp. 227–234.

Hansen, E. A., Bernstein, D. S., & Zilberstein, S. (2004). Dynamic programming for partially observable stochastic games. In *Proc. of the National Conference on Artificial Intelligence*, pp. 709–715.

He, J., Suau, M., & Oliehoek, F. A. (2020). Influence-augmented online planning for complex environments. In *Advances in Neural Information Processing Systems 33*.

Hernandez-Leal, P., Kaisers, M., Baarslag, T., & Munoz de Cote, E. (2017). A Survey of Learning in Multiagent Environments: Dealing with Non-Stationarity. *arXiv e-prints*, arXiv:1707.09183.

Hernandez-Leal, P., Kartal, B., & Taylor, M. E. (2019). Agent modeling as auxiliary task for deep reinforcement learning. In *Proc. of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, pp. 31–37.

Hoey, J., St-Aubin, R., Hu, A. J., & Boutilier, C. (1999). SPUDD: Stochastic planning using decision diagrams. In *Proc. of Uncertainty in Artificial Intelligence*, pp. 279–288.

Hong, Z.-W., Su, S.-Y., Shann, T.-Y., Chang, Y.-H., & Lee, C.-Y. (2018). A deep policy inference Q-Network for multi-agent systems. In *Proc. of the Seventeenth International Conference on Autonomous Agents and MultiAgent Systems*, pp. 1388–1396.

Hostetler, J., Fern, A., & Dietterich, T. (2014). State aggregation in monte carlo tree search. In *Proc. of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 2446–2452.

Howard, R. A., & Matheson, J. E. (1984). Influence diagrams. In *The Principles and Applications of Decision Analysis, Vol. II.*, pp. 719–763. Strategic Decisions Group.

Iyengar, G. N. (2005). Robust dynamic programming. *Mathematics of Operations Research*, *30*(2), 257–280.

Jaques, N., Lazaridou, A., Hughes, E., Gulcehre, C., Ortega, P., Strouse, D., Leibo, J. Z., & De Freitas, N. (2019). Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *Proc. of the Thirty-Sixth International Conference on Machine learning*, pp. 3040–3049.

Jong, N. K., & Stone, P. (2005). State abstraction discovery from irrelevant state variables. In *Proc. of the Nineteenth International Joint Conference on Artificial Intelligence*, pp. 752–757.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, *37*(2), 183–233.

Jovanovic, B., & Rosenthal, R. W. (1988). Anonymous sequential games. *Journal of Mathematical Economics*, *17*(1), 77 – 87.

Jurtz, V. I., Johansen, A. R., Nielsen, M., Almagro Armenteros, J. J., Nielsen, H., Sønderby, C. K., Winther, O., & Sønderby, S. K. (2017). An introduction to deep learning on biological sequence data: examples and solutions. *Bioinformatics*, *33*(22), 3685–3690.

Kaelbling, L. P., Littman, M. L., & Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, *101*(1-2), 99–134.

Kaelbling, L. P., & Lozano-Perez, T. (2012). Integrated robot task and motion planning in the now. Tech. rep. TR-2012-018, MIT CSAIL.

Katt, S., Oliehoek, F. A., & Amato, C. (2019). Bayesian reinforcement learning in factored POMDPs. In *Proc. of the Eighteenth International Conference on Autonomous Agents and Multiagent Systems*, pp. 7–15.

Keller, P. W., Mannor, S., & Precup, D. (2006). Automatic basis function construction for approximate dynamic programming and reinforcement learning. In *Proc. of the Twenty-Third International Conference on Machine learning*, pp. 449–456.

Kim, Y., Nair, R., Varakantham, P., Tambe, M., & Yokoo, M. (2006). Exploiting locality of interaction in networked distributed POMDPs. In *Proc. of the AAAI Spring Symposium on Distributed Plan and Schedule Management*.

Kizilkale, A. C., & Caines, P. E. (2012). Mean field stochastic adaptive control. *IEEE Transactions on Automatic Control*, *58*(4), 905–920.

Knoblock, C. A. (1993). *Generating Abstraction Hierarchies: An Automated Approach to Reducing Search in Planning*. Kluwer Academic Publishers.

Kok, J. R., & Vlassis, N. (2006). Collaborative multiagent reinforcement learning by payoff propagation. *Journal of Machine Learning Research*, *7*, 1789–1828.

Koller, D., & Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

Konidaris, G., & Barto, A. G. (2009). Efficient skill learning using abstraction selection. In *Proc. of the Twenty-First International Joint Conference on Artificial Intelligence*, pp. 1107–1112.

Konidaris, G., Kaelbling, L. P., & Lozano-Pérez, T. (2018). From skills to symbols: Learning symbolic representations for abstract high-level planning. *Journal of Artificial Intelligence Research*, *61*, 215–289.

Kumar, A., Zilberstein, S., & Toussaint, M. (2011). Scalable multiagent planning using probabilistic inference. In *Proc. of the International Joint Conference on Artificial Intelligence*, pp. 2140–2146.

Kuyer, L., Whiteson, S., Bakker, B., & Vlassis, N. (2008). Multiagent reinforcement learning for urban traffic control using coordination graphs. In *Machine Learning and Knowledge Discovery in Databases*, pp. 656–671.

Lauri, M., Pajarinen, J., & Peters, J. (2020). Multi-agent active information gathering in discrete and continuous-state decentralized pomdps by policy graph improvement. *Journal of Autonomous Agents and Multi-Agent Systems*, *34*(42).

Le, D., & Plaku, E. (2018). Cooperative, dynamics-based, and abstraction-guided multi-robot motion planning. *Journal of Artificial Intelligence Research*, *63*, 361–390.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Li, L., Walsh, T. J., & Littman, M. L. (2006). Towards a unified theory of state abstraction for MDPs. In *International Symposium on Artificial Intelligence and Mathematics*.

Littman, M. L. (1994). Memoryless policies: Theoretical limitations and practical results. In *Proceedings of the Third International Conference on Simulation of Adaptive Behavior: From Animals to Animats 3*, pp. 238–245.

Littman, M. L. (2012). Inducing partially observable Markov decision processes. In *Proc. of the Eleventh International Conference on Grammatical Inference*, pp. 145–148.

Machado, M. C., Bellemare, M. G., & Bowling, M. (2017). A Laplacian framework for option discovery in reinforcement learning. In *Proc. of the Thirty-Fourth International Conference on Machine learning*, pp. 2295–2304.

Mahadevan, S. (2010). Representation discovery in sequential decision making. In *Proc. of the National Conference on Artificial Intelligence*.

Mahadevan, S., & Maggioni, M. (2007). Proto-value functions: A Laplacian framework for learning representation and control in Markov decision processes. *Journal of Machine Learning Research*, *8*, 2169–2231.

McCallum, A. (1993). Overcoming incomplete perception with util distinction memory. In *Proc. of the Tenth International Conference on Machine Learning*, pp. 190–196.

McCallum, A. K. (1995). *Reinforcement Learning with Selective Perception and Hidden State*. Ph.D. thesis, University of Rochester.

Melo, F. S., & Veloso, M. (2009). Learning of coordination: exploiting sparse interactions in multiagent systems. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems*, pp. 773–780.

Melo, F. S., & Veloso, M. (2010). Approximate planning for decentralized MDPs with sparse interactions. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems*, pp. 1389–1390.

Melo, F. S., & Veloso, M. (2011). Decentralized MDPs with sparse interactions. *Artificial Intelligence*, *175*(11), 1757–1789.

Meuleau, N., Peshkin, L., Kim, K.-E., & Kaelbling, L. P. (1999). Learning finite-state controllers for partially observable environments. In *Proc. of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pp. 427–436.

Mitchell, T. M. (1980). The need for biases in learning generalizations. Tech. rep., Department of Computer Science, Laboratory for Computer Science Research.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik,

A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533.

Mostafa, H., & Lesser, V. (2009). Offline planning for communication by exploiting structured interactions in decentralized MDPs. In *Proc. of International Conferefence on Web Intelligence and Intelligent Agent Technology*, pp. 193–200.

Murphy, K. P. (2002). *Dynamic Bayesian Networks: Representation, Inference and Learning*. Ph.D. thesis, UC Berkeley, Computer Science Division.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.

Nair, R., Tambe, M., Yokoo, M., Pynadath, D. V., & Marsella, S. (2003). Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings. In *Proc. of the International Joint Conference on Artificial Intelligence*, pp. 705–711.

Nair, R., Varakantham, P., Tambe, M., & Yokoo, M. (2005). Networked distributed POMDPs: A synthesis of distributed constraint optimization and POMDPs. In *Proc. of the National Conference on Artificial Intelligence*, pp. 133–139.

Nguyen, D. T., Kumar, A., & Lau, H. C. (2017). Collective multiagent sequential decision making under uncertainty. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pp. 3036–3043.

Nikolaidis, S., Ramakrishnan, R., Gu, K., & Shah, J. (2015). Efficient model learning from joint-action demonstrations for human-robot collaborative tasks. In *Proc. of Human-Robot Interaction*.

Oliehoek, F. A. (2010). *Value-Based Planning for Teams of Agents in Stochastic Partially Observable Environments*. Ph.D. thesis, Informatics Institute, University of Amsterdam.

Oliehoek, F. A. (2012). Decentralized POMDPs. In *Reinforcement Learning: State of the Art*, Vol. 12, pp. 471–503. Springer.

Oliehoek, F. A., & Amato, C. (2014). Best response Bayesian reinforcement learning for multiagent systems with state uncertainty. In *Proc. of the Ninth AAMAS Workshop on Multi-Agent Sequential Decision Making in Uncertain Domains (MSDM)*.

Oliehoek, F. A., & Amato, C. (2016). *A Concise Introduction to Decentralized POMDPs*. Springer.

Oliehoek, F. A., Spaan, M. T. J., & Vlassis, N. (2008a). Optimal and approximate Q-value functions for decentralized POMDPs. *Journal of Artificial Intelligence Research*, *32*, 289–353.

Oliehoek, F. A., Spaan, M. T. J., Whiteson, S., & Vlassis, N. (2008b). Exploiting locality of interaction in factored Dec-POMDPs. In *Proc. of the Seventh Joint International Conference on Autonomous Agents and Multiagent Systems*, pp. 517–524.

Oliehoek, F. A., Spaan, M. T. J., & Witwicki, S. (2015a). Factored upper bounds for multiagent planning problems under uncertainty with non-factored value functions. In

Proc. of the Twenty-Fourth International Joint Conference on Artificial Intelligence, pp. 1645–1651.

Oliehoek, F. A., Spaan, M. T. J., & Witwicki, S. (2015b). Influence-optimistic local values for multiagent planning — extended version. *ArXiv e-prints*, *arXiv:1502.05443*.

Oliehoek, F. A., Whiteson, S., & Spaan, M. T. J. (2013). Approximate solutions for factored Dec-POMDPs with many agents. In *Proc. of the Twelfth International Conference on Autonomous Agents and Multiagent Systems*, pp. 563–570.

Oliehoek, F. A., Witwicki, S., & Kaelbling, L. P. (2011). Heuristic search of multiagent influence space. In *Proc. of the 9th European Workshop on Multi-agent Systems (EUMAS 2011)*.

Oliehoek, F. A., Witwicki, S., & Kaelbling, L. P. (2012). Influence-based abstraction for multiagent systems. In *Proc. of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pp. 1422–1428.

Ong, S. C. W., Png, S. W., Hsu, D., & Lee, W. S. (2009). Pomdps for robotic tasks with mixed observability. In *Robotics: Science and Systems V*.

Ong, S. C. W., Png, S. W., Hsu, D., & Lee, W. S. (2010). Planning under uncertainty for robotic tasks with mixed observability. *International Journal of Robotics Research*, *29*(8), 1053–1068.

Pajarinen, J. K., & Peltonen, J. (2011). Periodic finite state controllers for efficient POMDP and DEC-POMDP planning. In *Advances in Neural Information Processing Systems 24*, pp. 2636–2644.

Parr, R., Painter-Wakefield, C., Li, L., & Littman, M. (2007). Analyzing feature generation for value-function approximation. In *Proc. of the Twenty-Fourth International Conference on Machine learning*, pp. 737–744.

Pearl, J. (1988). *Probabilistic Reasoning In Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Petrik, M. (2007). An analysis of Laplacian methods for value function approximation in MDPs. In *Proc. of the Twentieth International Joint Conference on Artificial Intelligence*, pp. 2574–2579.

Petrik, M., & Subramanian, D. (2014). RAAM: the benefits of robustness in approximating aggregated MDPs in reinforcement learning. In *Advances in Neural Information Processing Systems 27*, pp. 1979–1987.

Petrik, M., & Zilberstein, S. (2009). A bilinear programming approach for multiagent planning. *Journal of Artificial Intelligence Research*, *35*(1), 235–274.

Poupart, P. (2005). *Exploiting Structure to Efficiently Solve Large Scale Partially Observable Markov Decision Processes*. Ph.D. thesis, Department of Computer Science, University of Toronto.

Powell, W. B. (2012). Perspectives of approximate dynamic programming. *Annals of Operations Research*, 1–38.

Puterman, M. L. (1994). *Markov Decision Processes—Discrete Stochastic Dynamic Programming*. Wiley.

Pynadath, D. V., & Marsella, S. (2007). Minimal mental models. In *Proc. of the Twenty-Second AAAI Conference on Artificial Intelligence*, pp. 1038–1044.

Rashid, T., Samvelyan, M., Schroeder de Witt, C., Farquhar, G., Foerster, J., & Whiteson, S. (2018). QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. *ArXiv e-prints*.

Rathnasabapathy, B., Doshi, P., & Gmytrasiewicz, P. (2006). Exact solutions of interactive POMDPs using behavioral equivalence. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems*, pp. 1025–1032.

Ravindran, B., & Barto, A. G. (2002). Model minimization in hierarchical reinforcement learning. In *Proc. of the 5th International Symposium on Abstraction, Reformulation and Approximation*, pp. 196–211.

Ravindran, B., & Barto, A. G. (2003). SMDP homomorphisms: An algebraic approach to abstraction in semi-Markov decision processes. In *Proc. of the Eighteenth International Joint Conference on Artificial Intelligence*, pp. 1011–1018.

Robbel, P., Oliehoek, F. A., & Kochenderfer, M. J. (2016). Exploiting anonymity in approximate linear programming: Scaling to large multiagent MDPs. In *Proc. of the Thirtieth AAAI Conference on Artificial Intelligence*, pp. 2537–2543.

Russell, S., & Norvig, P. (2009). *Artificial Intelligence: A Modern Approach* (3rd edition). Pearson Education.

Sacerdoti, E. D. (1974). Planning in a hierarchy of abstraction spaces. *Artificial Intelligence*, *5*(2), 115–135.

Sanner, S., Uther, W., & Delgado, K. V. (2010). Approximate dynamic programming with affine ADDs. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems*, pp. 1349–1356.

Schmidhuber, J. (1991). Reinforcement learning in Markovian and non-Markovian environments. In *Advances in Neural Information Processing Systems 3 (NIPS 3)*, pp. 500–506.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117.

Seuken, S., & Zilberstein, S. (2008). Formal models and algorithms for decentralized decision making under uncertainty. *Journal of Autonomous Agents and Multi-Agent Systems*, *17*(2), 190–250.

Shah, J., Wiken, J., Williams, B., & Breazeal, C. (2011). Improved human-robot team performance using chaski, a human-inspired plan execution system. In *Proc. of the 6th international conference on Human-robot interaction*, pp. 29–36. ACM.

Sleight, J., & Durfee, E. H. (2012). A decision-theoretic characterization of organizational influences. In *Proc. of the Eleventh International Conference on Autonomous Agents and Multiagent Systems*, pp. 323–330.

Sleight, J., & Durfee, E. H. (2015). Effective influence abstractions for organizational design. In *Proc. of the Fourteenth International Conference on Autonomous Agents and Multiagent Systems*, pp. 1267–1274.

Son, K., Kim, D., Kang, W. J., Hostallero, D. E., & Yi, Y. (2019). QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *Proc. of the Thirty-Sixth International Conference on Machine learning*, pp. 5887–5896.

Spaan, M. T. J. (2012). Partially observable Markov decision processes. In *Reinforcement Learning: State of the Art*, pp. 387–414. Springer.

Spaan, M. T. J., & Melo, F. S. (2008). Interaction-driven Markov games for decentralized multiagent planning under uncertainty. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems*, pp. 525–532.

St-Aubin, R., Hoey, J., & Boutilier, C. (2001). APRICODD: Approximate policy construction using decision diagrams. In *Advances in Neural Information Processing Systems 13*, pp. 1089–1095.

Stone, P., Kaminka, G. A., Kraus, S., & Rosenschein, J. S. (2010). Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *Proc. of the Twenty-Fourth AAAI Conference on Artificial Intelligence*.

Strehl, A. L., Diuk, C., & Littman, M. L. (2007). Efficient structure learning in factored-state MDPs. In *Proc. of the Twenty-Second AAAI Conference on Artificial Intelligence*, pp. 645–650.

Suau de Castro, M., Congeduti, E., Starre, R. A., Czechowski, A., & Oliehoek, F. A. (2019a). Influence-aware Memory for Deep Reinforcement Learning. *arXiv e-prints*, arXiv:1911.07643.

Suau de Castro, M., Congeduti, E., Starre, R. A., Czechowski, A., & Oliehoek, F. A. (2019b). Influence-based abstraction in deep reinforcement learning. In *Proc. of the AAMAS Workshop on Adaptive Learning Agents (ALA)*.

Subramanian, J., & Mahajan, A. (2019). Reinforcement learning in stationary mean-field games. In *Proc. of the Eighteenth International Conference on Autonomous Agents and Multiagent Systems*, pp. 251–259.

Sunberg, Z. N., Ho, C. J., & Kochenderfer, M. J. (2017). The value of inferring the internal state of traffic participants for autonomous freeway driving. In *2017 American Control Conference*, pp. 3004–3010.

Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V. F., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J. Z., Tuyls, K., & Graepel, T. (2018). Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proc. of the Seventeenth International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2085–2087.

Sutton, R. S., Precup, D., & Singh, S. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, *112*(1), 181–211.

Tacchetti, A., Song, H. F., Mediano, P. A. M., Zambaldi, V., Kramár, J., Rabinowitz, N. C., Graepel, T., Botvinick, M., & Battaglia, P. W. (2019). Relational forward models for multi-agent learning. In *International Conference on Learning Representations*.

Tatman, J. A. (1990). Dynamic programming and influence diagrams. *IEEE Transactions on Systems, Man and Cybernetics*, *20*(2), 365–379.

Theocharous, G., & Kaelbling, L. P. (2004). Approximate planning in POMDPs with macro-actions. In *Advances in Neural Information Processing Systems 16*.

Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press.

Tian, J., Paz, A., & Pearl, J. (1998). Finding minimal d-separators. Tech. rep. R-254, University of California, Los Angeles.

Toussaint, M. (2009). Probabilistic inference as a model of planned behavior. *Künstliche Intelligenz*, *3*(9), 23–29.

Van der Pol, E., Kipf, T., Oliehoek, F. A., & Welling, M. (2020). Plannable approximations to MDP homomorphisms: Equivariance under actions. In *Proc. of the Nineteenth International Conference on Autonomous Agents and Multiagent Systems*.

Van der Pol, E., & Oliehoek, F. A. (2016). Coordinated deep reinforcement learners for traffic light control. In *NIPS'16 Workshop on Learning, Inference and Control of Multi-Agent Systems*.

van der Zander, B., & Liśkiewicz, M. (2020). Finding minimal d-separators in linear time and applications. In *Proc. of the 35th Conference on Uncertainty in Artificial Intelligence*, Vol. 115, pp. 637–647.

Varakantham, P., Adulyasak, Y., & Jaillet, P. (2014). Decentralized stochastic planning with anonymity in interactions. In *Proc. of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 2505–2512.

Varakantham, P., Marecki, J., Yabu, Y., Tambe, M., & Yokoo, M. (2007). Letting loose a SPIDER on a network of POMDPs: Generating quality guaranteed policies. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems*.

Varakantham, P., young Kwak, J., Taylor, M., Marecki, J., Scerri, P., & Tambe, M. (2009). Exploiting coordination locales in distributed POMDPs via social model shaping. In *Proc. of the International Conference on Automated Planning and Scheduling*, pp. 313–320.

Vázquez-Salceda, J., Dignum, V., & Dignum, F. (2005). Organizing multiagent systems. *Autonomous Agents and Multi-Agent Systems*, *11*(3), 307–360.

Velagapudi, P., Varakantham, P., Scerri, P., & Sycara, K. (2011). Distributed model shaping for scaling to decentralized POMDPs with hundreds of agents. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems*.

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156–3164.

Wainwright, M. J., Jordan, M. I., et al. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, *1*(1–2), 1–305.

Walsh, T. J., Goschin, S., & Littman, M. L. (2010). Integrating sample-based planning and model-based reinforcement learning. In *Proc. of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, pp. 612–617. AAAI Press.

Wang, T., Wang, J., Wu, Y., & Zhang, C. (2020). Influence-based multi-agent exploration. In *International Conference on Learning Representations*.

Wang, T., Wang, J., Zheng, C., & Zhang, C. (2019). Learning nearly decomposable value functions via communication minimization. *arXiv e-prints*, arXiv:1910.05366.

Weninger, F., Erdogan, H., Watanabe, S., Vincent, E., Le Roux, J., Hershey, J. R., & Schuller, B. (2015). Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR. In *International Conference on Latent Variable Analysis and Signal Separation*, pp. 91–99.

Witwicki, S., & Durfee, E. (2010a). From policies to influences: A framework for nonlocal abstraction in transition-dependent Dec-POMDP agents. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems*, pp. 1397–1398.

Witwicki, S., & Durfee, E. (2010b). Influence-based policy abstraction for weakly-coupled Dec-POMDPs. In *Proc. of the International Conference on Automated Planning and Scheduling*, pp. 185–192.

Witwicki, S., & Durfee, E. (2011). Towards a unifying characterization for quantifying weak coupling in Dec-POMDPs.. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems*, pp. 29–36.

Witwicki, S., Oliehoek, F. A., & Kaelbling, L. P. (2012). Heuristic search of multiagent influence space. In *Proc. of the Eleventh International Conference on Autonomous Agents and Multiagent Systems*, pp. 973–981.

Witwicki, S. J. (2011). *Abstracting Influences for Efficient Multiagent Coordination Under Uncertainty*. Ph.D. thesis, University of Michigan.

Wolpert, D. H. (1996). The lack of a priori distinctions between learning algorithms. *Neural Computation*, *8*(7), 1341–1390.

Wolpert, D. H., & Macready, W. G. (1995). No free lunch theorems for search. Tech. rep. SFI-TR-95-02-010, Santa Fe Institute.

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. *IEEE Computational intelligence magazine*, *13*(3), 55–75.