Self-Explaining Deviations for Coordination

Anonymous Author(s)
Affiliation
Address
email

Abstract

Fully cooperative, partially observable multi-agent problems are ubiquitous in the real world. In this paper, we focus on a specific subclass of coordination problems in which humans are able to discover *self-explaining deviations* (SEDs). SEDs are actions that deviate from the common understanding of what reasonable behavior would be in normal circumstances. They are taken with the intention of causing another agent or other agents to realize, using theory of mind, that the circumstance must be abnormal. We first motivate SED with a real world example and formalize its definition. Next, we introduce a novel algorithm, improvement maximizing self-explaining deviations (IMPROVISED), to perform SEDs. Lastly, we evaluate IMPROVISED both in an illustrative toy setting and the popular benchmark setting Hanabi, where it is the first method to produce so called *finesse* plays, which are regarded as one of the more iconic examples of human theory of mind.

1 Introduction

There has been a lot of progress in the deep multi-agent collaboration domain ranging from better learning [33, 10] and planning [19, 28] algorithms designed to achieve strong performance in self-play to finding policies that collaborate well with other agents include humans [11, 30]. However, these algorithms largely assume that the partners at test time will follow a fixed policy. Even in one of the most powerful search algorithms [19], the agent performing search assumes that other agents follow blueprint when planning forward, resulting in primarily unilateral deviations that improve the performance without relying on other players also deviating in response.

Dispatcher: Oregon 911.
Caller: I would like to order a pizza at...
Dispatcher: You called 911 to order a pizza?
Caller: Uh, Yeah, apartment...
Dispatcher: This is the wrong number to call for a pizza.
Caller: No no no... you're not understanding me.
Dispatcher: I'm getting you now. Is the other guy still there?
Caller: Yep. I need a large pizza.
Dispatcher: All right. How about medical. You need medical?
Caller: No. With pepperoni.
Dispatcher: Turn your sirens off before you get there. Caller ordered a pizza. And agreed with everything I said that there's domestic violence going on.

Figure 1: A real-life self-explaining deviation.

Humans generally assume other human follow certain social norms (blueprint policies) when acting in the society, and interpret other humans' behavior based on such norms to infer hidden information about the partially observable world. For example, when a driver sees a car ahead suddenly stopping on a road, they may infer that there is some accident ahead or the car is broken. However, human also have the ability to improvise when the conventional actions are restricted or there exist other actions that can bring superior outcomes in situations. Taking the event in Figure 1 as an example, the caller deviates from the conventional practice of stating the situation truthfully because it has negative consequences. In this situation, the caller would perform this deviation because the deviation is only beneficial if

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

the dispatcher figures out caller's real intention and also deviates instead of simply ending the call. This type of deviation is interesting because, firstly, it can be detected by the other person because it appears to be a mistake under common practice and, secondly, the expected deviation for the other person in response can be independently decided by both individuals based on the common understanding of the world and the deviation chosen by the first person. We refer to this type of phenomena as self-explaining deviations (SEDs).

SEDs are actions that, under normal circumstances, would not make sense, given the agents' common understanding. They are executed with the intention that teammates will use *theory of mind* to deduce that the situation is unusual in a particular and important way, and will adapt their behavior to account for this additional information. While there is ample evidence that humans perform SEDs, such as the *finesse play* in the card game Hanabi, we provide some evidence that neither existing planning nor learning algorithms can do so.

In this paper, we first formalize the problem setting and definition of SEDs. Then we introduce a novel planning algorithm, IMPROVement maxImizing Self-Explaining Deviations (IMPROVISED), for performing SEDs. We show that, under some assumptions, IMPROVISED performs the optimal SED in terms of expected return maximization. Next, we provide a motivating experiment, illustrating that in a small toy problem designed to require SEDs to perform optimally, IMPROVISED is able to compute an optimal joint policy, whereas other multi-agent algorithms are not. Lastly, we present experiments on Hanabi, where we show that IMPROVISED is able to produce finesse plays, which is one of the most interesting techniques that human experts perform frequently.

2 Background

FOSG and Public POMDP. For our notation, we use an adaption of factored observation stochastic games (**FOSG**) [17]. $\mathcal{W}$ is the set of **world states** and w^0 is a designated initial state. $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$ is the space of **joint actions**. $\mathcal{T}$ is the **transition function** mapping $\mathcal{W} \times \mathcal{A} \to \Delta(\mathcal{W})$. $\mathcal{R}\colon \mathcal{W} \times \mathcal{A} \to \mathbb{R}$ is the **reward function**. $\mathcal{O} = (\mathcal{O}_{\text{priv}(1)}, \ldots, \mathcal{O}_{\text{priv}(N)}, \mathcal{O}_{\text{pub}})$ is the **observation function** where $\mathcal{O}_{\text{priv}(i)}\colon \mathcal{W} \times \mathcal{A} \times \mathcal{W} \to \mathbb{O}_{\text{priv}(i)}$ specifies the **private observation** that player i receives. $\mathcal{O}_{\text{pub}}\colon \mathcal{W} \times \mathcal{A} \times \mathcal{W} \to \mathbb{O}_{\text{pub}}$ specifies the **public observation** that all players receive. $O_i{=}\mathcal{O}_i(w, a, w'){=}(\mathcal{O}_{\text{priv}(i)}(w, a, w'), \mathcal{O}_{\text{pub}}(w, a, w'))$ is player i's **observation** and a **history** is a finite sequence $h = (w^0, a^0, \ldots, w^t)$. The **set of histories** is denoted by $\mathcal{H}$. The **information state** for player i at $h = (w^0, a^0, \ldots, w^t)$ is $s_i(h) \coloneqq (O_i^0, a_i^0, \ldots, O_i^t)$. The **information state space** for player i is $\mathcal{S}_i \coloneqq \{s_i(h) \mid h \in \mathcal{H}\}$. The **legal actions** for player i at s_i is denoted $\mathcal{A}_i(s_i)$. A **joint policy** is a tuple $\pi = (\pi_1, \ldots, \pi_N)$, where **policy** π_i maps $\mathcal{S}_i \to \Delta(\mathcal{A}_i)$. The **public state** at h is the sequence $s_{\text{pub}}(h) \coloneqq s_{\text{pub}}(s_i(h)) \coloneqq (O_{\text{pub}}^0, \ldots, O_{\text{pub}}^t)$. The **information state set** for player i at $s \in \mathcal{S}_{\text{pub}}$ is $\mathcal{S}_i(s) \coloneqq \{s_i \in \mathcal{S}_i \mid s_{\text{pub}}(s_i) = s\}$, where $\mathcal{S}_{\text{pub}}$ is the space of public states. Finally, the **reach probability** of h under π is $P^\pi(h)$.

Rather than working with in a multi-agent setting directly, we invoke the public POMDP transformation that maps cooperative multi-agent settings to equivalent single-agent POMDPs. Given a common-payoff FOSG $\langle \mathcal{N}, \mathcal{W}, w^0, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{O} \rangle$, we can construct an equivalent **public POMDP** [21] $\langle \tilde{\mathcal{W}}, \tilde{w}^0, \tilde{\mathcal{A}}, \tilde{\mathcal{T}}, \tilde{\mathcal{R}}, \tilde{\mathcal{O}} \rangle$ as follows: The world states of the public POMDP $\tilde{\mathcal{W}}$ is the set $\{(s_1(h), \ldots, s_N(h)) : h \in \mathcal{H}\}$. The initial world state of the public POMDP $\tilde{w}^0$ is the tuple $(s_1(h^0), \ldots, s_N(h^0))$. The actions of the public POMDP are called **joint decision rules**. It is denoted by Γ and has N components. The ith component of it Γ_i is the **decision rule** for player i.

A decision rule Γ_i maps s_i to an element of $\mathcal{A}_i(s_i)$ for each $s_i \in \mathcal{S}_i(s_{\text{pub}}(h))$; it instructs a player in the common-payoff FOSG how to act as a function of its private information. Given $\tilde{w} \equiv (s_1, \ldots, s_n)$ and Γ, the transition distribution $\tilde{\mathcal{T}}(\tilde{w}, \Gamma)$ is induced by $\mathcal{T}((s_1, \ldots, s_n), a)$, where $a \equiv \Gamma((s_1, \ldots, s_n)) \coloneqq (\Gamma_1(s_1), \ldots, \Gamma_N(s_N))$. Given $\tilde{w} \equiv (s_1, \ldots, s_n)$ and $\tilde{w}' \equiv (s'_1, \ldots, s'_n)$, the reward and observation are given by $\tilde{\mathcal{R}}(\tilde{w}, \Gamma, \tilde{w}') \equiv \mathcal{R}((s_1, \ldots, s_n), \Gamma((s_1, \ldots, s_n)), (s'_1, \ldots, s'_n))$ and $\tilde{\mathcal{O}}(\tilde{w}, \Gamma, \tilde{w}') \equiv \mathcal{O}_{\text{pub}}((s_1, \ldots, s_n), \Gamma((s_1, \ldots, s_n)), (s'_1, \ldots, s'_n))$, respectively.

Every policy in the public POMDP corresponds to a joint policy in the underlying common-payoff FOSG, which receives exactly the same expected return. Therefore, it is sufficient to work with the public POMDP. See [27, 28] for further discussion.

When the public POMDP is considered as a belief MDP, rather than as a POMDP, its belief states are of the form $b^t : (s_1^t, \ldots, s_n^t) \mapsto P^{\Gamma^0,\ldots,\Gamma^{t-1}}(s_1^t, \ldots, s_N^t \mid s_{\text{pub}}^t)$. In words, this is the joint distribution over private information states, conditioned on the historical policy $(\Gamma^0, \ldots, \Gamma^{t-1})$ and the public state s_{pub}^t. This public belief MDP is abbreviated as the PuB-MDP.

3 Self-Explaining Deviations

While the name *self-explaining deviation* (SED) is novel to this work, the idea behind SEDs is not. Perhaps the most famous example of SEDs comes from the cooperative card game Hanabi [1] in the form of a play called a "finesse". Readers unfamiliar with Hanabi and finesse may first jump Section 5.2 and Section 5.3 for the detailed descriptions. In a finesse play, the acting player, i.e., the first player who initiates the deviation as part of the finesse, intentionally communicates misleading information to a receiving player, which would hurt the team's score if the receiving player acted upon it using the prior convention. The second player in the finesse, who acts after the first player but before the receiving player, realizes that the first player must have deviated from the prior convention after observing the seemingly disastrous move. However, knowing that the first player is rational, the second player realizes that they can also deviate to reach a better outcome than the one that the original convention would have led to. After the second player plays its part of the joint deviation, the original information from the first player is no longer misleading and the subsequent players can continue to follow the prior conventions.

SEDs may take place in any cooperative situation with at least two players where common knowledge blueprint policies that players follow under normal assumptions exist. Intuitively, SEDs capture a form of joint deviations where one agent takes an action that at first appears to be a mistake or otherwise highly off-policy from another player's perspective, i.e. zero or low probability under the blueprint. On the presumption that the first agent chose that action intelligently and deliberately and there is a unique off-policy action for the observing agent that could potentially result in an even-better-than-normal outcome for both, the observing agent may reason that the first agent "intends" them to take it—to take a leap of faith that the first agent has not erred, but rather knows both agents can get that better outcome, even if the observing agent does not have the information themselves to prove that this outcome will result. This work uses the term SEDs to describe this phenomenon of communication via apparent mistakes in a general context. We formalize SED in Section 3.1 and propose IMPROVISED, a novel planning algorithm that performs the optimal SEDs under some assumptions, in Section 4.

3.1 Examining Self-Explaining Deviations

Having defined the problem setting, we are now ready to investigate SEDs. To facilitate our investigation, let's consider the *trampoline-tiger* game in Figure 2 as an example. In the game, Alice is standing on a balcony, while Bob stands on the ground next to a lever. Pulling the lever will either deploy a trampoline below the balcony, or release a tiger. From above, Alice can observe which is the case, and then Alice decides whether to jump off the balcony or not. Bob *observes Alice's choice* but does *not* know whether the lever will deploy a trampoline or tiger, and decides to pull the lever or not. Alice wants to get down from the balcony, but will die from the fall unless Bob pulls the lever *and* it releases a trampoline. If Bob pulls the lever and releases a tiger, it will eat Bob.

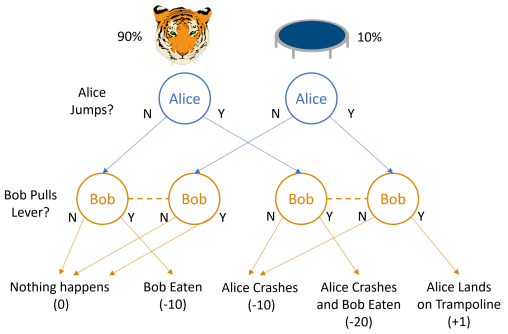

Figure 2: Alice knows whether Bob's lever will deploy a tiger or a trampoline, and then jumps or not. Bob observes only Alice's action and then pulls the lever or not. Dotted lines between two Bob nodes means that Bob cannot distinguish them.

Consider the joint policy $(* \mapsto \text{N}, * \mapsto \text{N})$. While not optimal, this joint policy is not unreasonable, as Bob won't be eaten and Alice won't fall to her death. Indeed, independent of the probability of a trampoline, this joint policy is a Nash equilibrium. In addition, as we shall see in Section 5.1, state-of-the-art MARL algorithms converge to this solution most of the time. However, there is a clear

opportunity for a SED under this blueprint. If there is a tiger, Alice should obviously never jump. So if Bob observes Alice choosing to jump, and trusts that Alice is intelligent and would only have chosen to jump if it could lead to a better result, Bob should realize that there must be a trampoline. Therefore Alice should trust that Bob will pull the lever if she jumps, as a result.

Under any planning algorithm that considers only unilateral deviations, Bob and Alice can't escape this local optimum. Alice will never jump because she is under the belief that Bob will choose not to pull the lever. Therefore, finding SEDs in general requires considering multi-lateral deviations, i.e. simultaneous deviations by more than one player.

Since finding multi-lateral deviations in general games with partially observable state is a complex problem, we restrict our focus to settings satisfying the following assumptions:

Assumption1. *We assume sequential and publicly observable actions.*

SEDs may still be possible with partially-observable actions, but for simplicity we focus on when Bob can directly observe Alice deviate from the blueprint, rather than having to deduce it from partial or incomplete observations.

Assumption2. *We consider only cases where the Bob's private information is not necessary for coordinating on the SED.*

For simplicity, we always refer to the first player who initiates the joint deviation as the *Alice* and the second player who figures out Alice's intention and plays their part of the joint deviation as *Bob* in the following discussions beyond the scope of this toy game. In other words, we only consider SEDs where Bob needs only act as a function of the common-knowledge state and the Alice's action, rather as a function of private knowledge of their own.

Under these assumptions, we define the SED as follows:

Definition1. *Given a common knowledge blueprint (BP) policy π, common public belief of Alice and Bob b, and information state of Alice s_1, SED is any pair of joint deviations (a'_1, a'_2) for Alice and Bob that satisfy all the following conditions:*

- $q_\pi(b, s_1, a'_1, a'_2) \geq q_\pi(b, s_1)$*, i.e., it gives higher expected future return than BP;*
- $P(a'_1 | \pi, b) \leq \epsilon$*, i.e.,* a'_1 *is highly unlikely under* π *under the common belief;*
- $a'_2 \sim f(b, a'_1)$*, i.e.,* a'_2 *is a function of the public belief and Alice's deviation action.*

Here, $q_\pi(b, s_1, [a_1, a_2])$ is Alice's estimate of the expected future return after executing the optional a_1, a_2 and following π afterwards. ϵ is a hyper-parameter. The function f is determined by the algorithm designer under the constraint that f only takes b and a'_1 as input. It is used both by Alice to predict Bob's response and by Bob to decide the response independently.

4 IMPROVISED

In this section we derive IMPROVISED, an algorithm to perform the SEDs. The core of the algorithm is to find a function f under the constraint of SED and an optimization procedure that Alice uses to decide whether to deviate given her information state s_1 and possible responses $a_2 \sim f$ from Bob.

4.1 Defining the Optimization Problem

With the observations in the prior sections, we are ready to write selecting a SED as a PuB-MDP optimization problem:

$$\begin{aligned}
\max_{\Gamma_1,\Gamma_2} q_\pi(b, \Gamma_1, \Gamma_2) &= \max_{\Gamma_1,\Gamma_2} \mathbb{E}_{s_1 \sim b}\, q_\pi(b, s_1, \Gamma_1(s_1), \Gamma_2) \\
&= \max_{\Gamma_1,\Gamma_2} \mathbb{E}_{s_1 \sim b}\, q_\pi(b, s_1, \Gamma_1(s_1), \Gamma_2 \circ \Gamma_1(s_1)) \\
&= \max_{\Gamma_1,\Gamma_2} \mathbb{E}_{s_1 \sim b} \max\left[q_\pi(b, s_1, \Gamma_1(s_1), \Gamma_2 \circ \Gamma_1(s_1)), q_\pi(b, s_1)\right].
\end{aligned} \tag{1}$$

The series of equalities above begins with a generic multi-lateral search in the PuB-MDP. The LHS is ranging over Alice and Bob's decision rules such that, if Alice uses Γ_1 and Bob uses Γ_2 in belief

state b, and all the players play according to π thereafter, the expected return is maximized. The first equality holds without applying any additional assumptions, simply based on the fact that Alice is able to observe s_1. The second equality holds by applying Assumptions 1 and 2, which imply both that Bob can observe Alice's action and that it is the only thing he needs to condition his decision upon (note that the change of the set over which Γ_2 ranges is left implicit). The final holds due to the design decisions that a) our algorithm only plays an SED for actions not supported in the BP and that b) players opt into the BP when there is no better deviation. The inner maximization expresses the fact that Alice can observe s_1 and may opt into the blueprint if the blueprint value exceeds that of the deviation. Because Bob knows which actions are supported by the blueprint, he can opt in exactly when Alice opts in, and play according to the deviation otherwise.

4.2 An Easier Special Case

Unfortunately, expression (1) remains difficult to optimize, as the number of Γ_1 and Γ_2 is combinatorial. To ameliorate, we can apply the simplifying constraint of only allowing Alice and Bob to deviate for a single action pair regardless of Alice's information state s_1. For simplicity, we also set $\epsilon = 0$. Let $\mathcal{D}(b)$ be the set of plausible deviation actions for Alice, i.e., action that cannot be played under blueprint: $\mathcal{D}(b) := \{a_1 \in \mathcal{A}_1 | \forall s_1 \in b : \pi(a_1|b, s_1) = 0\}$. Let $\mathcal{R}$ be the set of all plausible actions that Bob can play in response in every possible state, i.e., $\mathcal{R} := \bigcap_{s_2 \in \mathcal{S}_2} \mathcal{A}_2(s_2)$. If $\mathcal{R} = \emptyset$, Alice will skip searching for SED this turn and follow BP. Now we can augment the expected value of a deviation with the allowed range of actions:

$$\hat{q}_\pi(b, s_1, a_1, a_2) = \begin{cases} q_\pi(b, s_1, a_1, a_2), & \text{if } a_1 \in \mathcal{D}(b), a_2 \in \mathcal{R}, \\ -\infty, & \text{otherwise.} \end{cases} \tag{2}$$

Then we can rewrite equation (1) using our newly defined q function as follows:

$$(a_1^*, a_2^*) = \text{argmax}_{a_1, a_2} \mathbb{E}_{s_1 \sim b} \max[\hat{q}_\pi(b, s_1, a_1, a_2), q_\pi(b, s_1)]. \tag{3}$$

These values could be found in $O(|\mathcal{S}_1||\mathcal{A}_1||\mathcal{A}_2|)$ time, given the two inner q functions. Moreover, as all of these depend only on the public belief state, both players can compute them independently, assuming that there are no ties.

Once the deviation pair is computed, Alice decides whether to proceed with the deviation or not given her private information: Alice plays a_1^* if $\hat{q}_\pi(b, s_1, a_1^*, a_2^*) > q_\pi(b, s_1)$, and the blueprint action otherwise. It is possible that (a_1^*, a_2^*) is not a valid deviation pair if no plausible deviations exist, e.g., a_1^* could be outside of $\mathcal{D}(b)$. However, in this case $\hat{q}_\pi(b, s_1, a_1^*, a_2^*)$ is $-\infty$, and so Alice will resort to playing blueprint. Bob can detect the deviation as $\mathcal{D}(b)$ is a public knowledge and $a_1^* \in \mathcal{D}(b)$. Therefore, depending on his observation, he can either play a_2^* or respond as usual.

We can represent the value of the policy that allows a single deviation at b and plays blueprint afterwards as: $q_\pi^*(b) = \max_{a_1, a_2} \mathbb{E}_{s_1 \sim b} \max(\hat{q}_\pi[b, s_1, a_1, a_2), q_\pi(b, s_1)]$. It is easy to see that $q_\pi^* \geq q_\pi$, i.e., this algorithm has a weak policy improvement guarantee.

If we apply this procedure to the tiger-trampoline game with a blueprint policy that always no-ops:

$$\begin{aligned} q_\pi^*(b) &= \max_{a_1, a_2} \mathbb{E}_{s_1 \sim b} \max[\hat{q}_\pi(b, s_1, a_1, a_2), q_\pi(b, s_1)] \\ &= \max_{a_1, a_2} [P(\text{tiger}) \max(\hat{q}_\pi(b, \text{tiger}, a_1, a_2), 0) + P(\text{trampoline}) \max(\hat{q}_\pi(b, \text{trampoline}, a_1, a_2), 0)] \\ &= P(\text{trampoline}). \end{aligned}$$

As expected, we obtain a policy improvement with value equal to the probability of a trampoline. We discover the SED where Alice jumps when a trampoline is present and Bob pulls the lever. We provide a proof-of-principle implementation of IMPROVISED in the trampoline-tiger game that can be run online at `https://bit.ly/3KtMLT6`.

4.3 Coordination by Extending Conventions

In a coordination context, we cannot require Bob to perform arbitrary symmetry breaking, i.e., choose an action among several with the same expected value. For example, consider a case in which a set of multiple action pairs tied for having the maximal value $\{(a_1 = x, a_2 = y), (a_1 = x, a_2 = z), (a_1 = w, a_2 = z)\}$. Say that Alice picks $a_1 = x$. This is problematic for Bob because he has no

information about whether Alice is selecting x because she is in an information state in which (x, y) is good or whether she is in an information state in which (x, z) is good (the states for which they are good may be disjoint).

One way to resolve this issue is to force Alice to respect Bob's inabilities to break ties by making the response function stochastic, e.g., a softmax over the expected values. That is, given that Alice deviates with a_1, Alice assumes that Bob plays

$$a_2^* \sim \text{softmax}_{a_2} \left[\mathbb{E}_{s_1 \sim b} \max[\hat{q}_\pi(b, s_1, a_1, a_2), q_\pi(b, s_1)]/t\right] \tag{4}$$

with a temperature hyper-parameter to control the sharpness of the distribution. Then Alice chooses to deviate using

$$\text{argmax}_{a_1} \mathbb{E}_{s_1 \sim b} \max[\mathbb{E}_{a_2 \sim a_2^*(b, a_1)} \hat{q}_\pi(b, s_1, a_1, a_2), q_\pi(b, s_1)]. \tag{5}$$

Bob can either play according to the softmax or select the single maximum, as Alice has already picked the deviation assuming that Bob cannot break symmetries. This is optimal if Alice can only play one single deviation action across *all her information states*. In the following we'll address how to deal with multiple deviations, which also makes the formulation easier to understand.

4.4 Taking Alice's Information State Into Account

Our discussion above has largely ignored the fact that Alice can perform different deviations given her information state s_1 (i.e., private observation). Nonetheless, it helps us find the response function $f(b, a_1) = \text{softmax}_{a_2} \left[\mathbb{E}_{s_1 \sim b} \max[\hat{q}_\pi(b, s_1, a_1, a_2), q_\pi(b, s_1)]/t\right]$. Given that Alice knows Bob will response according to $a_2 \sim f(b, a_1)$, Alice is now free to optimize her action by taking her information state back into consideration. By applying Jensen's inequality, we observe that

$$\begin{aligned} &\max_{a_1} \mathbb{E}_{s_1 \sim b} \max \left[\mathbb{E}_{a_2 \sim f(b, a_1)} \hat{q}_\pi(b, s_1, a_1, a_2), q_\pi(b, s_1)\right] \\ &\leq \mathbb{E}_{s_1 \sim b} \max_{a_1} \max \left[\mathbb{E}_{a_2 \sim f(b, a_1)} \hat{q}_\pi(b, s_1, a_1, a_2), q_\pi(b, s_1)\right]. \end{aligned}$$

Therefore, at information state s_1, Alice opts out of the blueprint if the *optimal deviation action*, $a_1^* = \max_{a_1} \mathbb{E}_{a_2 \sim f(b, a_1)} \hat{q}_\pi(b, s_1, a_1, a_2)$, exceeds the expected return of the BP, $q_\pi(b, s_1)$. This is the final formulation of IMPROVISED^E. Please refer to the Appendix A for the detailed pseudocode.

We also define IMPROVISED^P, where Bob plays according to the probability of improvement

$$a_2^* \sim \text{softmax}_{a_2} \left[\mathbb{E}_{s_1 \sim b} I[\hat{q}_\pi(b, s_1, a_1, a_2) > q_\pi(b, s_1)]/t\right] \tag{6}$$

where I is the indicator function and Alice selects any action maximizing the expected improvement. While IMPROVISED^P does not maximize the expected return as IMPROVISED^E does, it may better reflect human SEDs since it finds deviation pairs that maximize the *probability of improvement*.

5 Experiments

We test the IMPROVISED in two different settings. The first setting is the trampoline-tiger game explained before. Secondly, we apply IMPROVISED to three-player Hanabi, where we start from a BP trained on human data.

5.1 Trampoline Tiger

As illustrated in Section 4.2, in the trampoline-tiger game IMPROVISED^E and IMPROVISED^P both recover the optimal SED, when starting from the no-op BP in 100% of the runs. Alice will decide to jump whenever there is a trampoline and Bob then pulls the lever to open the door, leading to the optimal expected return of 0.1.

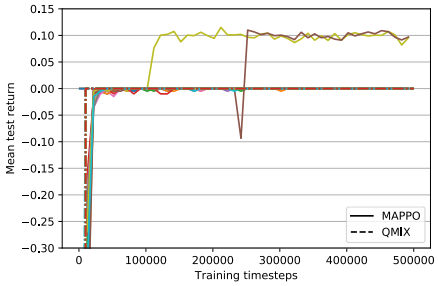

Figure 3: 20 independent runs of MAPPO (solid lines) and 24 of QMIX (dashed lines) on the tiger-trampoline toy problem. 2 MAPPO runs find the optimal strategy while no QMIX run does.

For comparison, we also ran 20 unique seeds of MAPPO and 24 different hyper-parameter combinations of QMIX [23] on the trampoline-tiger problem. The results are shown in Figure 3. All runs

Model	Finesse-able	Finesse-complete
SAD [10]	1375	597 (43.42%)
Other-Play [11]	1537	512 (33.31%)
OBL (Level 5) [13]	1100	356 (32.36%)
Behavior Clone [13]	1376	1183 (85.97%)

(a) Number of finesse-able situations over 1000 games and % of those where *player 3* will correctly respond to complete the finesse if *player 1* and *2* were forced to play the finesse move. The Behavior Clone model has a much higher chance to complete a finesse.

Method	Finesse
Blueprint	0
SPARTA [19]	15
IMPROVISED^E	51
IMPROVISED^P	62

(b) Number of finesses executed by different methods at 200 finesse-complete situations using Behavior Clone agent from Table 1a as blueprint.

Table 1: IMPROVISED for Finesse in Hanabi

converged extremely rapidly to a policy that avoids the highly negative payoffs, but only 2/20 of the MAPPO runs and 0/24 of the QMIX runs discovered the optimal strategy within 500k timesteps, leading to an average reward across runs of 0.01 for MAPPO and 0.0 for QMIX. The code and hyper-parameters are available in the supplementary material.

In our experiments, a variety of other standard multi-agent learning techniques could not solve this toy problem at all, including value-decomposition network [31] and simplified action decoder [10].

5.2 Hanabi

Hanabi has been proposed as a benchmark challenge for multi-agent coordination, and has been the subject of significant research [1, 12]. Briefly, in Hanabi, there is a a 50 card deck consisting of 5 different suits (colors) and 5 rank of card within each suit, for a total of 25 unique combinations, with some duplicates. 2 to 5 players cooperatively take turns playing cards, giving hints to other players, or discarding. Players may see their partners' hands, but *not* their own. The team's goal is to play exactly one card of each rank in each suit in increasing order, scoring 1 point per successful play. Upon three plays of cards duplicated or out-of-order, the players lose and instead score 0. Giving a hint spends one of 8 shared hint tokens, allowing one to name a color or rank in a partner's hand and indicate all cards of that color or rank in that hand. Discarding a card replenishes one hint token. See also [1] for a more thorough description of the rules and basic strategy.

5.3 What is a Finesse?

As mentioned earlier in Section 3, a clear example of a SED is the finesse move in Hanabi. Although described in far more detail with an example in [1], we will recap it briefly here. For convenience, we always refer to the current active player of a given turn as *player 1* and the players who move subsequently as *player 2*, *player 3* etc.

The standard, vanilla form of the finesse in Hanabi occurs when *player 1* gives a hint to *player 3* (i.e. the player who moves two steps later) that under the conventions common between the players implies that player *player 3* should play a particular card. Unbeknownst to *player 3*, that card, if played, would be out-of-order and would fail disastrously. Instead, *player 2* has just drawn the card that is both playable and if played makes *player 3*'s card playable next. *Player 2* is expected to realize that the only way that *player 1*'s hint could be good rather than a disaster is if *player 2* themself has newly drawn this exact card, and to take a leap of faith and play that newly drawn card blindly, having no explicit information about it. The end result is that *player 1* signals in only one action for both other players to each successfully play a card, a large gain.

While other finesse-like patterns are possible, in our Hanabi experiments for simplicity we focus only those finesse moves that follow the pattern described above.

5.4 IMPROVISED in Hanabi

We design experiments to evaluate IMPROVISED in Hanabi from both qualitative and quantitative perspectives. In the qualitative evaluation, we study whether IMPROVISED can indeed perform finesses in manually selected situations where finesse could be completed (finesse-complete situations). We are particularly interested in the finesse style SEDs as they are most natural and intuitive

way to demonstrate IMPROVISED's ability to perform SEDs. Note that IMPROVISED discovers many types of SEDs in Hanabi beyond the finesse style moves because it searches for any beneficial joint deviations that when *player 1* initiates a deviation, *player 2* can independently figure out the correct deviation in response. However, other types of SEDs may be hard for human to interpret and less intuitive to analyse and present. In the quantitative evaluation, we show that IMPROVISED as a planning algorithm improves the expected return when applied to test finesse-able situations and entire games.

To implement IMPROVISED in Hanabi, we first need a belief function from which we can sample game states given either public or private knowledge of the game to perform Monte Carlo rollouts. Luckily, the belief over possible hands in Hanabi can be computed analytically [19]. The belief is first initialized to cover all possible hands and then incrementally updated by filtering out hands that contradict with public knowledge revealed through hints and that would have caused the players to pick different moves at each time step. Since we are mainly concerned with the finesse style SEDs in the experiments, we restrict the search action space $\mathcal{A}_1$ to contain only the hint moves that target *player 3* and $\mathcal{A}_2$ to contain only the play moves. *Player 1* and *player 2* compute their own copies of the response function f independently. The detailed hyper-parameters and computational cost are in the Appendix.

To check IMPROVISED's ability to perform finesse, we first generate situations where seasoned human players may carry out finesse moves. We use a blueprint policy π to generate selfplay games over a range of decks (game seeds) and look for situations where *player 1* observes that *player 2*'s newest draw card is playable and has not been hinted at, and meanwhile *player 3* holds an un-hinted card that can be played after *player 2* plays their newest card (finesse-able situations). Then we manually override *player 1*'s and *player 2*'s moves in finesse-able situations to carry out the finesse and check if *player 3* will play the designated card under the blueprint to make the finesse complete and worthwhile (finesse-complete situations). We experiment with 4 different blueprints and show their statistics of finesse-complete situations over 1000 game seeds in Table 1a. All the agents are trained following the settings in corresponding prior works. Although finesse-able situations are pretty common in the games played by all four agents, the first three RL agents trained without human data rarely complete a manually enforced finesse move, making them undesirable for the subsequent experiment. It also worth noting that none of the RL blueprints perform finesses by themselves while the behavior clone agent performs only 1 finesse out of the 1183 finesse-complete situations.

We apply both IMPROVISED^E and IMPROVISED^P with the behavior clone blueprint on 200 randomly chosen finesse-complete situations to check how many finesse it performs. The results are shown in Table 1b. Due to the novelty of this problem setting, there has been no prior method to directly compare against. For reference, we run SPARTA [19], a strong search algorithm for Dec-POMDP designed to find unilateral deviations that maximizes the expect return, on the same situations with the same restrictions on the set of actions that the agents may deviate to. From the table, we see that both IMPROVISED algorithms perform significantly more finesses, indicating their effectiveness in finding SEDs on the fly. In the situations where IMPROVISED does not perform finesse, it can either be that IMPROVISED finds no beneficial deviations or it finds better, non-finesse SEDs. It is also interesting to see that IMPROVISED^P finds more finesses than IMPROVISED^E, which is in line with our hypothesis that humans might use the *probability of improvement* when reasoning about SEDs instead of relying on small difference between different (joint) q-values.

Then, we run IMPROVISED on the full game of Hanabi. At any time step, the active player will first check whether a deviation has been initiated by other players by checking whether previous players have picked actions with low probability under the blueprint. If a deviation has been initiated, they will play their corresponding role as either *player 2* or *player 3*. Otherwise the active player will decide whether to deviate using the IMPROVISED method. Over a fixed set of 100 game seeds, IMPROVISED^E performs **29** finesses while SPARTA only performs 3. Although it is still much less frequent than what expert human players will do, IMPROVISED is nonetheless a meaningful step along this novel and challenging direction.

Although our work is focused on understanding and replicating this human capability rather than optimizing for self-play score, we also report quantitative results showing a noticeable increase in score using IMPROVISED compare to the plain blueprint, giving evidence that the deviations found do have a meaningful impact on the game. If we run IMPROVISED only at the 200 finesse-complete situations from Table 1b and use blueprint for the rest of the timesteps, then IMPROVISED^E and

IMPROVISED^P achieve average scores of 18.58 and 18.36 respectively while then blueprint gets 17.52 on those same games. When we run IMPROVISED^E on the full game, we improve the average score from 17.8 to 23.54.

6 Related Work

6.1 Public Belief Methods

Our method builds on prior work that models Dec-POMDs as PuB-MDPs [21, 22, 7, 9, 28, 29]. More recently, this same approach has been extended to two-player zero-sum games [20, 18, 4, 14, 5] and two-team zero-sum games [15, 34]. PuB-MDPs enable theoretically sound planning algorithms without having to reason about the agents' entire policies because the public belief state serves as a sufficient statistic for planning. However, prior work on PuB-MDPs do not explicitly search for SEDs.

6.2 Search in Dec-POMDPs

Search and planning algorithms have been a critical component to AI breakthroughs in complex multi-agent environments, including backgammon [32], chess [6], go [24–26], and poker [20, 2, 3]. Closely related to our work are methods for policy improvement via planning in Dec-POMDPs, such as SPARTA [19]. SPARTA conducts a one-step lookahead and chooses the action that maximizes expected value assuming all players play according to a common-knowledge BP thereafter. However, SPARTA searches only for unilateral deviations from the BP rather than multilateral deviations, because searching over all of the latter would be intractable. As discussed in Section 3.1, discovering SEDs may require considering multi-lateral deviations from the BP. IMPROVISED is able to discover these multi-lateral deviations by searching over a constrained set of multi-lateral deviations. Other prior work has also investigated planning algorithms for Dec-POMDPs [16]. However, most of these techniques have typically searched over public belief states [7, 8, 28]. This greatly increases the complexity of the search procedure.

7 Conclusions

The human ability to coordinate with others using minimal explicit agreement and extending conventions "on the fly" is one of the most intriguing reasoning abilities and far ahead of any of our learning methods. In this paper we formalize the definition of such behavior as *self-explaining deviations*. We show why existing methods fail to learn these types of deviations and presented an optimal algorithm named IMPROVISED that can both discover them and correspond correctly when they are being carried out by other agents at test time. Crucially, IMPROVISED allows agents to improve performance at test time by improving upon a common knowledge blueprint policy, even when the blueprint is a Nash equilibrium.

Limitation and Future Work In this paper we first address SED with planning without taking into account the extra complexity of learning and approximating due to its difficulty and novelty. The resulted algorithm is computationally expensive as it requires Monte Carlo rollouts to estimate many crucial quantities at test time. However, it is possible to derive a more efficient, learning based version of IMPROVISED since the joint Q-function $q(b, s_1, a_1, a_2)$ can be learned. There are challenges to make these estimates accurate on the deviating states but our paper nonetheless provides a valuable stepping stone along this exciting future direction.

Societal Impact One of the most fundamental goal of multi-agent learning is to produce agents that collaborate well with human in real world. Therefore, it is important for the artificial agents to not only understand human conventions, but also decode human's intention when deviations occur so that agents can response correctly. Such agents should also be capable to initiate such self-explaining deviations when beneficial to maximize their social values. Therefore, this direction of work to understand and replicate human's ability to perform SEDs using theory of mind is important and positive to the society. We currently do not foresee any negative societal impact from this work.

References

[1] N. Bard, J. N. Foerster, S. Chandar, N. Burch, M. Lanctot, H. F. Song, E. Parisotto, V. Dumoulin, S. Moitra, E. Hughes, I. Dunning, S. Mourad, H. Larochelle, M. G. Bellemare, and M. Bowling. The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, 280, 2020. URL https://doi.org/10.1016/j.artint.2019.103216.

[2] N. Brown and T. Sandholm. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science*, page eaao1733, 2017.

[3] N. Brown and T. Sandholm. Superhuman AI for multiplayer poker. *Science*, page eaay2400, 2019.

[4] N. Brown, A. Bakhtin, A. Lerer, and Q. Gong. Combining deep reinforcement learning and search for imperfect-information games. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17057–17069. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/c61f571dbd2fb949d3fe5ae1608dd48b-Paper.pdf.

[5] O. Buffet, J. Dibangoye, A. Delage, A. Saffidine, and V. Thomas. On bellman's optimality principle for zs-posgs. *arXiv preprint arXiv:2006.16395*, 2020.

[6] M. Campbell, A. J. Hoane Jr, and F.-h. Hsu. Deep blue. *Artificial intelligence*, 134(1-2):57–83, 2002.

[7] J. S. Dibangoye, C. Amato, O. Buffet, and F. Charpillet. Optimally solving dec-pomdps as continuous-state mdps. *Journal of Artificial Intelligence Research*, 55:443–497, 2016.

[8] M. Fehr, O. Buffet, V. Thomas, and J. Dibangoye. ρ-pomdps have lipschitz-continuous ε-optimal value functions. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6933–6943, 2018.

[9] J. N. Foerster, H. F. Song, E. Hughes, N. Burch, I. Dunning, S. Whiteson, M. Botvinick, and M. Bowling. Bayesian Action Decoder for Deep Multi-Agent Reinforcement Learning. In *Proceedings of the 36th International Conference on Machine Learning*, pages 1942–1951. PMLR, 2019.

[10] H. Hu and J. N. Foerster. Simplified action decoder for deep multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=B1xm3RVtwB.

[11] H. Hu, A. Lerer, A. Peysakhovich, and J. Foerster. "Other-play" for zero-shot coordination. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4399–4410. PMLR, 13–18 Jul 2020.

[12] H. Hu, A. Peysakhovich, A. Lerer, and J. Foerster. "other-play"for zero-shot coordination. In *Proceedings of Machine Learning and Systems 2020*, pages 9396–9407. 2020.

[13] H. Hu, A. Lerer, B. Cui, L. Pineda, D. Wu, N. Brown, and J. N. Foerster. Off-belief learning. *(To Appear) ICML*, 2021. URL https://arxiv.org/abs/2103.04000.

[14] D. Kartik and A. Nayyar. Upper and lower values in zero-sum stochastic games with asymmetric information. *Dynamic Games and Applications*, pages 1–26, 2020.

[15] D. Kartik, A. Nayyar, and U. Mitra. Common information belief based dynamic programs for stochastic zero-sum games with competing teams. *arXiv preprint arXiv:2102.05838*, 2021.

[16] S. Katt, F. A. Oliehoek, and C. Amato. Learning in pomdps with monte carlo tree search. In *International Conference on Machine Learning*, pages 1819–1827. PMLR, 2017.

[17] V. Kovarík, M. Schmid, N. Burch, M. Bowling, and V. Lisý. Rethinking formal models of partially observable multiagent decision making. *CoRR*, abs/1906.11110, 2019. URL http://arxiv.org/abs/1906.11110.

[18] V. Kovařík, M. Schmid, N. Burch, M. Bowling, and V. Lisỳ. Rethinking formal models of partially observable multiagent decision making. *arXiv preprint arXiv:1906.11110*, 2019.

[19] A. Lerer, H. Hu, J. N. Foerster, and N. Brown. Improving policies via search in cooperative partially observable games. In *AAAI*, 2020.

[20] M. Moravčík, M. Schmid, N. Burch, V. Lisỳ, D. Morrill, N. Bard, T. Davis, K. Waugh, M. Johanson, and M. Bowling. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337):508–513, 2017.

[21] A. Nayyar, A. Mahajan, and D. Teneketzis. The Common-Information Approach to Decentralized Stochastic Control. Springer Verlag, 2014. doi: 10.1007/978-3-319-02150-8_4.

[22] F. A. Oliehoek. Sufficient plan-time statistics for decentralized pomdps. In *Twenty-Third International Joint Conference on Artificial Intelligence*. Citeseer, 2013.

[23] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 4295–4304. PMLR, 2018.

[24] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484, 2016.

[25] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550 (7676):354, 2017.

[26] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

[27] S. Sokota. Solving Common-Payoff Games with Approximate Policy Iteration, 2020. URL https://era.library.ualberta.ca/items/17edd0aa-ec13-4cd4-8e30-a77b5d8c5ccc.

[28] S. Sokota, E. Lockhart, F. Timbers, E. Davoodi, R. D'Orazio, N. Burch, M. Schmid, M. Bowling, and M. Lanctot. Solving common-payoff games with approximate policy iteration. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 9695–9703. AAAI Press, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17166.

[29] S. Sokota, H. Hu, D. J. Wu, J. Z. Kolter, J. N. Foerster, and N. Brown. A fine-tuning approach to belief state modeling. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=ckZY7DGa7FQ.

[30] D. Strouse, K. R. McKee, M. M. Botvinick, E. Hughes, and R. Everett. Collaborating with humans without human data. *CoRR*, abs/2110.08176, 2021. URL https://arxiv.org/abs/2110.08176.

[31] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, and T. Graepel. Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, pages 2085–2087, Richland, SC, July 2018. International Foundation for Autonomous Agents and Multiagent Systems.

[32] G. Tesauro. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3): 58–68, 1995.

[33] C. Yu, A. Velu, E. Vinitsky, Y. Wang, A. Bayen, and Y. Wu. The surprising effectiveness of mappo in cooperative, multi-agent games, 2021.

[34] B. H. Zhang, G. Farina, and T. Sandholm. Team belief DAG form: A concise representation for team-correlated game-theoretic decision making. *CoRR*, abs/2202.00789, 2022. URL https://arxiv.org/abs/2202.00789.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]