Generalized Beliefs for Cooperative AI

Darius Muglich [1] **Luisa Zintgraf** [1] **Christian Schroeder de Witt** [1] **Shimon Whiteson** [1] **Jakob Foerster** [1]

Abstract

Self-play is a common paradigm for constructing solutions in Markov games that can yield optimal policies in collaborative settings. However, these policies often adopt highly-specialized conventions that make playing with a novel partner difficult. To address this, recent approaches rely on encoding symmetry and convention-awareness into policy training, but these require strong environmental assumptions and can complicate policy training. We therefore propose moving the learning of conventions to the belief space. Specifically, we propose a belief learning model that can maintain beliefs over rollouts of policies not seen at training time, and can thus decode and adapt to novel conventions at test time. We show how to leverage this model for both search and training of a best response over various pools of policies to greatly improve ad-hoc teamplay. We also show how our setup promotes explainability and interpretability of nuanced agent conventions.

1. Introduction

One of the most popular ways of training a policy in multi-agent reinforcement learning (MARL) is via *self-play* (Samuel, 1959), where during training a joint policy controls the strategies of every player. This can lead to convergence to highly specialized policies that have enjoyed great success in zero-sum settings (Brown et al., 2020; Silver et al., 2018). Self-play can also yield highly proficient policies in cooperative settings (Hu & Foerster, 2019; Schroeder de Witt et al., 2019; Yu et al., 2021); however, if self-play is left unconstrained, agents that train independently with self-play and then meet at test time to play a cooperative game (i.e., engage in *cross-play*) may do much worse than when they were self-playing.

Agents in cooperative tasks often must learn to implicitly communicate, particularly when the available communication channels are too narrow to fully resolve mutual uncertainty (Heider & Simmel, 1944; Tian et al., 2020). In the case of the popular and frequently-studied card game Hanabi (Bard et al., 2020) (see Appendix A for details on Hanabi), consider the example *convention* that "hinting green" also means that the rightmost held card of the player receiving the hint is playable. One could then construct a second policy where instead "hinting yellow" bears this implicit cue. While these two policies are essentially equivalent, if two agents meet and one plays the former policy and the other plays the latter, they will perform poorly because they have each overfit to their respective conventions.

To improve the cross-play performance of independent self-play trained agents (i.e., to effectively engage in *zero-shot coordination* (Hu et al., 2020)), recent approaches have explored explicitly encoding environmental symmetries (where symmetry refers to state/action sequences that are equivalent up to relabelling of the state/action space) into policy training (Hu et al., 2020). However, this complicates the training process and requires domain knowledge.

Others have explored modular approaches that separately learn the complexities intrinsic to the task (i.e., rule-dependent behaviour) and the conventions specific to a partner (i.e., convention-dependent behaviour) (Shih et al., 2021). However, their method comes at the expense of scalability, with experiments limited to bandit problems and 1-colour Hanabi.

Yet another approach that has been explored is in controlling the cognitive-reasoning depth (i.e. controlling the extent to which agents build more complex conventions on top of the baseline expectation that earlier conventions are being followed) so to avoid formation of arbitrary conventions that can complicate cross-play (Hu et al., 2021b). While an interesting direction, it mitigates policies from adopting a large class of behaviours, and as well motivates the question of whether the true absence of such convention formation is strictly necessary for coordination?

We seek a paradigm to learn policies without the disadvantages listed above; that is, we seek to relax strong environmental assumptions, retain scalability, and to leave policy learning unconstrained. As such, in this paper, we propose shifting the burden of learning conventions and intent onto

[1]University of Oxford, England, United Kingdom. Correspondence to: Darius Muglich <dariusm1997@yahoo.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

the agent's maintained model of uncertainty over the environment, i.e., its *belief*. Specifically, we make the following main contributions:

1. We propose a methodology to learn an emulated belief (bypassing costly exact Bayesian updating) that can be used to decode agent intent and behaviour, and test this methodology on the *generalized* belief modelling task, i.e., the zero-shot task of maintaining belief over trajectories featuring a policy not seen at training time (this is a sequence-modelling task, where the trajectory is encoded and the belief state is decoded).

2. We leverage this emulated belief for improving cross-play performance via techniques such as Monte Carlo search and training a best response over a pool of policies. In this way we propose effective auxiliary mechanisms that can be flexibly added to any policy regime to enhance cross-play, whilst keeping training simple and scalable.

3. We use our belief model to promote interpretability of agent conventions by giving evidence to how turns of a game interrelate, and what environmental features agents use for implicit communication. Interpretability is critical for building trustworthy and safe systems (Wells & Bednarz, 2021).

2. Background

This section formalises the problem setting and defines the notion of beliefs.

2.1. Dec-POMDPs

We formalise the cooperative multi-agent setting as a decentralised partially-observable Markov decision process (Oliehoek, 2012, Dec-POMDP) which is a 9-tuple $(\mathcal{S}, \mathcal{N}, \{\mathcal{A}^i\}_{i=1}^n, \{\mathcal{O}^i\}_{i=1}^n, \mathcal{T}, \mathcal{R}, \{\mathcal{U}^i\}_{i=1}^n, T, \gamma)$, for finite sets $\mathcal{S}, \mathcal{N}, \{\mathcal{A}^i\}_{i=1}^n, \{\mathcal{O}^i\}_{i=1}^n$, denoting the set of states, agents, actions and observations, respectively, where a superscript i denotes the set pertaining to agent $i \in \mathcal{N} = \{1, \dots, n\}$ (i.e., $\mathcal{A}^i$ and $\mathcal{O}^i$ are the action and observation sets for agent i, and $a^i \in \mathcal{A}^i$ and $o^i \in \mathcal{O}^i$ are a specific action and observation agent i may undertake). We also write $\mathcal{A} = \times_i \mathcal{A}^i$ and $\mathcal{O} = \times_i \mathcal{O}^i$, the sets of joint actions and observations, respectively. $s_t \in \mathcal{S}$ is the state at time t and $s_t = \{s_t^k\}_k$, where s_t^k is state feature k of s_t. $a_t \in \mathcal{A}$ is the joint action of all agents taken at time t, which changes the state according to the transition distribution $s_{t+1} \sim \mathcal{T}(s_{t+1} \mid s_t, a_t)$. The subsequent joint observation of the agents is $o_{t+1} \in \mathcal{O}$, distributed according to $o_{t+1} \sim \mathcal{U}(o_{t+1} \mid s_{t+1}, a_t)$, where $\mathcal{U} = \times_i \mathcal{U}^i$; observation features of $o_{t+1}^i \in \mathcal{O}^i$ are notated analogously to state features; that is, $o_{t+1}^i = \{o_{t+1}^{i,k}\}_k$. The reward $r_{t+1} \in \mathbb{R}$ is distributed according to $r_{t+1} \sim \mathcal{R}(r_{t+1} \mid s_{t+1}, a_t)$. T is the horizon and $\gamma \in [0, 1]$ is the discount factor.

Notating $\tau_t^i = (a_0^i, o_1^i, \dots, a_{t-1}^i, o_t^i)$ for the action-observation history of agent i, agent i acts according to a policy $a_t^i \sim \pi^i(a_t^i \mid \tau_t^i)$. The agents seek to maximize the return, i.e., the expected discounted sum of rewards:

$$J_T := \mathbb{E}_{p(\tau_T)}[\sum_{t' \leq T} \gamma^{t'-1} r_{t'}], \tag{1}$$

where $\tau_t = (s_0, a_0, o_1, r_1, \dots, a_{t-1}, o_t, r_t, s_t)$ is the trajectory until time t.

2.2. Beliefs

As the Dec-POMDP is not fully observable, an agent i can operate under an estimate of the trajectory. The private belief of agent i is the posterior $b_t^i := p(\tau_t \mid \tau_t^i)$, i.e., the model of uncertainty agent i maintains about the trajectory and true state. The belief is a sufficient statistic for the environment state and characterizes the theory of mind (Baker et al., 2017) often requisite for successful performance in Dec-POMDPs.

The private belief in a Dec-POMDP may be iteratively maintained if the policy of agent j is prior knowledge. If so, then each action taken by agent j introduces a belief update across all other agents: supposing agent i has current belief b_{t-1}^i and next observes (a_t^j, o_t^i) (where we have broken out the action of player j from the observation), we can use Bayes' rule to obtain

$$\begin{aligned} b_t^i &= p(\tau_t \mid \tau_t^i) \\ &= p(\tau_t \mid \tau_{t-1}^i, a_t^j, o_t^i) \\ &= \frac{b_{t-1}^i \pi^j(a_t^j \mid \tau_{t-1}) p(o_t^i \mid \tau_{t-1}, a_t^j)}{\sum_{\tau'_{t-1}} b_{t-1}^i \pi^j(a_t^j \mid \tau'_{t-1}) p(o_t^i \mid \tau'_{t-1}, a_t^j)}. \end{aligned} \tag{2}$$

However, this manner of belief updating not only assumes π^j is prior knowledge, but also requires evaluation over all possible trajectories and is thus computationally intractable in large settings. To circumvent these issues, an aggregator function such as a recurrent neural network is commonly relied on to take the action-observation history so for its intermediate hidden states to form a statistic z_t^i of the history sufficient for predicting future observations (Hausknecht & Stone, 2015; Zhang et al., 2015; Zhu et al., 2017). Often z_t^i is trained by conditioning the policy on it and using gradient descent. However, the RL signal is often too weak to learn a rich representation for z_t^i that provides sufficient statistics for the filtering posterior over states, a phenomenon that has been empirically demonstrated, e.g., by Moreno et al. (2018) and Zintgraf et al. (2020). We therefore look neither to maintain explicit Bayesian updates nor to implicitly form sufficient statistics, but to *learn* a belief emulation.

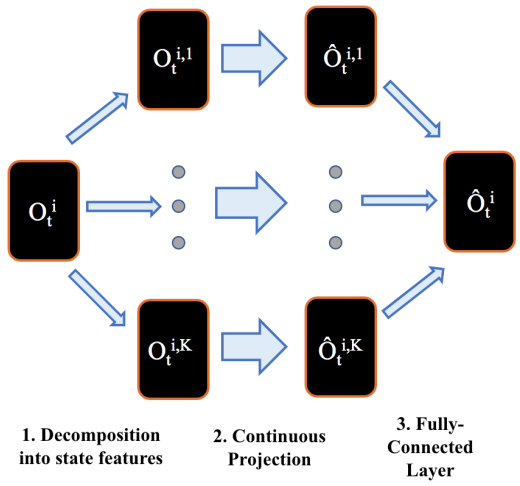

Figure 1. Proposed input embedding scheme for encoder module (Ψ_E), where we decompose a state into its features, project the features onto an information space, then aggregate them to obtain the state embedding.

3. Model

For generalized belief learning, our approach is to learn a belief model using supervised learning over rollouts of several different self-play trained policies, and our main question is whether this is sufficient to generalize reasoning over novel policies at test time. See Appendix B for further specification of learning beliefs in the Dec-POMDP setting and our autoregressive approach to belief modelling.

We largely use the same self-attention based architecture as Vaswani et al. (2017), but we modify the embedding mechanisms, which we describe presently.

Without loss of generality, we fix an agent i. Let $H = |\{c_1, \ldots, c_H\}|$ be the number of unobservable state features[1]. To input the action-observation history and the conditioned unobserved features, we require learnable embedding functions $\Psi_E : (\mathcal{O}^i)^T \mapsto \mathbb{R}^{T \times d}, \Psi_D : \mathcal{S}^H \mapsto \mathbb{R}^{H \times d}$ such that

$$\Psi_E(\tau_t^i) = \mathbf{x}_E, \ \Psi_D(c_1, \ldots, c_H) = \mathbf{x}_D,$$

where $\mathbf{x}_E, \mathbf{x}_D$ are then inputted to the encoder and decoder modules of our architecture, respectively (Vaswani et al., 2017). Here T and d are the maximum timestep[2] and the dimensionality of the embedding, respectively, where both are set as hyperparameters.

For Ψ_E, we apply a trainable embedding layer over the discrete observable state features $o_t^{i,1}, \ldots, o_t^{i,K}$, where $K = |o_t^i|$ for all t, so as to project them onto a continuous vector space of dimension d_{feature} to obtain $\hat{o}_t^{i,1}, \ldots, \hat{o}_t^{i,K}$. These embedding representations of the observable state features then become parameters of our model and are learned through the contexts the state features appear in during gameplay.

Following this continuous projection of the state features, a fully connected layer is applied over them so as to obtain an embedding $\hat{o}_t^i$ of the observable state o_t^i. The representations $\hat{o}_t^i$ for each t comprise $\mathbf{x}_E$ and are passed to the learning system.

For optimisation purposes, we add dropout to Ψ_E for regularisation (Srivastava et al., 2014), layer normalisation to assist with learning (Ba et al., 2016), and a positional embedding to provide a learnable signal encoding the timestep order (Vaswani et al., 2017). Figure 1 provides a high-level illustration of Ψ_E. For Ψ_D, we maintain an embedding layer of size H to project $c_1, \ldots, c_H$ onto a continuous space.

Our choice of Ψ_E does not directly project o_t^i onto a continuous space to obtain $\hat{o}_t^i$ because the continuous projective layer must maintain an enumeration of all members in the vocabulary. Letting d_k denote the number of values that observable state feature $o_t^{i,k}$ may take on, we have that the vocabulary for all o_t^i is of size $\prod_{k=1}^K d_k$.

Alternatively, using our choice of Ψ_E, the vocabulary we need maintain (over the state features) is just of size $\sum_{k=1}^K d_k$. Besides memory considerations, training on large vocabularies can be difficult and can result in poor performance on rarer tokens (Labeau, 2018).

Furthermore, the reason we combine the continuous state feature representations $\hat{o}_t^{i,1}, \ldots, \hat{o}_t^{i,K}$ in a fully-connected layer to obtain $\hat{o}_t^i$ is two-fold:

1. Without this fully-connected layer, $\mathbf{x}_E$ would be of size $KT \times d$ rather than $T \times d$, which is problematic given the quadratic memory and computation requirement of the dot-product self-attention mechanism: the fully-connected layer thus reduces the memory requirement from $O(K^2T^2)$ to $O(T^2)$ and the compute requirement from $O(dK^2T^2)$ to $O(dT^2)$, which on the Hanabi control task we have found to be the difference between prohibitively costly and efficiently executable;

2. The fully-connected layer gives context that all the state features belong to the same state, providing the model with meaningful structure about the data; this context can in turn aid with the problem of poor performance on rarer states, as it gives the model a means to compare states with similar state feature values.

[1] If H varies per timestep, set $H = \max\{H_1, \ldots, H_T\}$.

[2] If $t < T$, in practice we pad τ_t^i and $\mathbf{x}_E$ to make dimensions match.

4. Generalized Belief Learning

The training of the belief model is set up so as to be similar to policy learning, rather than the typical supervised regime of fixing a training set, a validation set and a test set: given a collection of pre-trained policies, we run a number of parallel simulators to simultaneously record the trajectories of games played by the pre-trained policies to an experience replay buffer.

Concurrently, we sample from the experience replay buffer and use the sampled rollouts for training the belief model with a cross entropy loss (see Appendix B). Training in such a way helps avoid over-fitting without manually tuning hyper-parameters and regularization. See Appendix D.1 for the hyperparameters used.

Section 4.1 describes the experimental setup used, and Section 4.3 shows the findings of the experiments.

4.1. Experimental Setup

We use the AI benchmark task and representative Dec-POMDP Hanabi (Bard et al., 2020) for our experiments. Hanabi is a unique and challenging card game that requires agents to formulate informative implicit conventions in order to be successful (see Appendix A).

We used thirteen pre-trained simplified action decoder (SAD) policies that were used in the work of Hu et al. (2020), and which we downloaded from their GitHub repository.[3] In addition, we trained 12 Other-Play (OP) policies (Hu et al., 2020). OP is a training regime for producing a class of policies that can cohesively maintain environmental symmetry so to effectively engage in zero-shot coordination. SAD and OP policies adopt markedly different styles of play and use very different conventions (as we shall explore more in depth), and so their complementary analysis will evince how our proposed methodologies can work for varied policy types. The OP policies were trained with SAD and Value Decomposition Networks (Sunehag et al., 2017). The policies in each pool differ by the seeds used for training, which were randomly chosen for each model.

The policies achieve average scores of 23.97 ± 0.04 in two-player Hanabi self-play (a perfect score in Hanabi is 25), but only average scores of 6.258 ± 1.51 in cross-play. This highlights just how specialized the respective conventions adopted in these policies are, where despite all the policies being trained with the same method and playing to analogous levels of proficiency, they lose much of their performance capability in cross-play. The OP policies used achieve average self-play scores of 23.1 ± 0.05, but cross-play scores of 16.88 ± 0.96.

[3] https://github.com/facebookresearch/hanabi_SAD

We henceforth refer to a belief model trained with trajectories featuring only one policy as a "Single" model, and a model trained with trajectories featuring multiple polices as a "Multi" model (a Multi model constitutes a "generalized" belief). "Multi"-x will refer to a Multi model trained with trajectories of x different policies.

Over both our SAD and OP pools, we trained several belief models: for the SAD pool, we randomly selected various policies to train 8 different Single beliefs, 6 different Multi-6 models, and 6 different Multi-12 models; for the OP pool, we similarly trained 8 different Single beliefs, 6 different Multi-6 models, and 6 different Multi-12 models. We then tested how these various models fared with maintaining belief over the policies they did not see at training time.

All the models were otherwise trained with the procedure described at the beginning of Section 4. Everything but the trajectories trained with were fixed so as to mitigate the effect of confounding variables. The trained models were then tested against trajectories featuring the held out policy. See Appendix D.1 for more details.

4.2. Grounded Belief

We compare against the grounded belief ψ_0 for agent i:

$$\psi_0(\tau_t \mid \tau_t^i) = \prod_{h=1}^{H} \mathbb{1}_{x=c_h} f(x = c_h \mid \tau_t^i), \tag{3}$$

where f is the probability mass function for the categorical distribution over the plausible values the unobservable state features may take on, where plausibility is determined by conditioning on *grounded* information. Grounded information is information that may be derived from the action-observation history only as can be deduced from the rules of the Dec-POMDP, and not from assumptions of what a policy may intend from certain behaviours. As such, the grounded belief assumes knowledge of environmental dynamics, but it does not take into account implicit cooperative cues that may be immanent of agent play. It is thus an important standard for testing against so as to measure how much implicit information our methodology takes into account.

4.3. Results

Figure 2 shows the results of our generalized belief learning methodology (see Appendix C for an intuitive interpretation of cross entropy scores over unobservable features in a Dec-POMDP). The Multi models can predict over rollouts of the unseen test policy with nearly the accuracy of the model trained on the test policy itself.

One may suspect there to be a correlation between how "aligned" the training policies are with the test policy, and the belief model's performance over the test policy. As such, to discuss the correlation between cross-play and cross entropy,

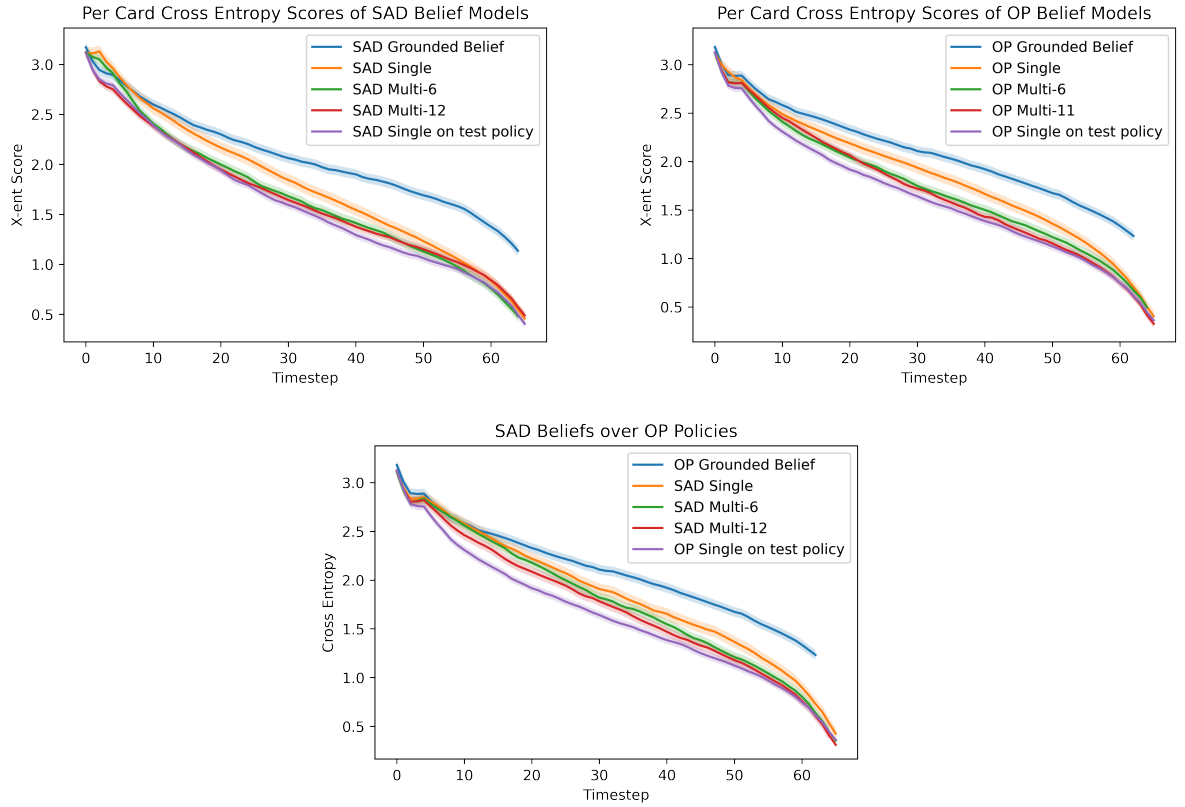

Figure 2. Per card cross entropy (X-ent) scores of the averages of the Single models and the different Multi models, all tasked with maintaining belief over trajectories featuring policies not seen at training time. The grounded belief and the belief model trained on the test policy itself are provided here for reference. The shading corresponds to the standard error of the mean at each timestep. The curves were computed over 20k randomly generated games.

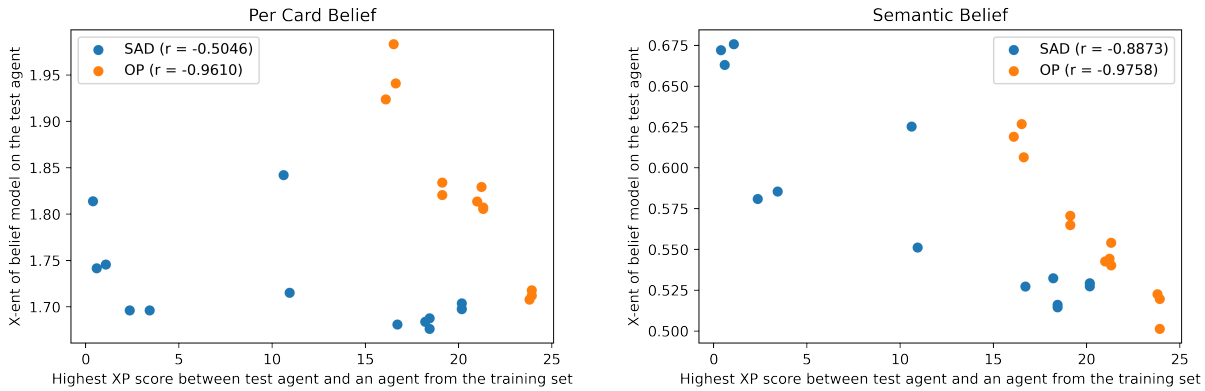

Figure 3. Left: Tested on two Multi-6 per card belief models. Right: Tested on two Multi-6 semantic belief models. The vertical axis represents the average cross entropy of the Multi-6 over a test policy, and the horizontal axis represents the highest cross-play (XP) score between the test agent and an agent from the training set. r denotes the Pearson's correlation coefficient (Benesty et al., 2009).

we consider the *semantic* belief: rather than the per card belief that considers the precise card identity, the semantic belief considers the general actionability of a card (whether it is playable, discardable, or otherwise). Figure 3 illustrates the differences in correlation between cross-play and cross entropy between the two types of beliefs. Interestingly, we see that for SAD policies, the trend is more concrete for the semantic belief than the per card belief, suggesting that

the conventions SAD policies form resolve how a card may be used relative to maximizing Dec-POMDP reward, rather than the exact card identity. In contrast, we can see that OP policies do in fact resolve per card identity. In this way we have demonstrated how our generalized belief models may be used to elucidate the blackbox of agent play (more applications in explainability may be found in Section 6). We speculate that for both types of beliefs, that training on a policy with a sufficiently high cross-play score with the test policy, or training with sufficiently many policies at training time, tends to give better generalization; since one in practice cannot control the former, we advocate the latter.

By training over a pool of policies, the generalized belief model may observe the different ways agents maintain symmetries in the Dec-POMDP, thus automating the learning of symmetries so to bypass strong environmental assumptions.

5. Improving Cross-Play

This section demonstrates how the generalized belief model may be leveraged to realise a policy suited for coordination.

5.1. Method

We propose two different schemes for leveraging the generalized belief: 1) a Monte Carlo search scheme; 2) training a best response over a pool of policies with the hidden state of the generalized belief model.

Search has been a critical component for the success of AI in several multi-agent settings, such as chess (Campbell et al., 2002), poker (Moravčík et al., 2017), and Go (Silver et al., 2018). Lerer et al. (2020) propose a method of note for the Dec-POMDP setting; their work maintains an exact belief using updates as in Equation (2), samples from this belief to obtain hypothetical values for the unobservable state features, then conducts Monte Carlo search rollouts for each potential action predicated on these sampled state features using the agreed-upon policy to determine the best action. Motivated by this setup, we propose a similar framework for zero-shot cooperative tasks between policies $\pi^1, \ldots, \pi^n$: we replace the exact belief with our learned generalized belief (thus bypassing costly Bayesian updating), and we conduct Monte Carlo rollouts by partnering π^1 with randomly chosen policies from the pool of policies the generalized belief was trained over. Intuitively, the generalized belief samples help infer the intent of policies π^2 through π^n, and the Monte Carlo rollouts calculate the most robust action for π^1 to take (by playing over rollouts of multiple different pool policies) predicated on this inference.

Along with the search scheme described above, we also consider training a best response. In general, one might not be able to expect high-level zero-shot performance between a best response trained over a pool of policies and a novel test policy. This is because, as mentioned in Section 2.2, it is difficult for the RL signal alone to form a rich belief representation (Moreno et al., 2018; Gangwani et al., 2020), especially in this setting over a pool of diverse policies. Furthermore, there is no guarantee that the consequent simple heuristics learned would generalize effectively to novel partners. Motivated by this, we propose training the best response with the hidden state of the generalized belief model. In this way, we essentially form a deeper, model-based architecture, and allow our best response to incorporate the demonstrated adaptability of the generalized belief to reason over novel conventions. In addition, training with the belief model's hidden state may even guide the formation of higher order beliefs (e.g., beliefs over beliefs). We train the best response using SAD and Independent Q-Learning (Tan, 1993).

Refer to Appendix D.2 for more details on the search and best response setups.

5.2. Results

SAD	W/O	SBS	GBS
BR W/O GEN. BELIEF	10.29± 1.05	11.32± 1.18	12.01±1.03
BR W/ GEN. BELIEF	12.36±0.96	12.03±1.11	12.47±1.02
OP	W/O	SBS	GBS
BR W/O GEN. BELIEF	17.49± 0.89	17.81± 0.92	18.31±0.85
BR W/ GEN. BELIEF	18.30±0.84	17.99±0.89	18.41±0.82
SAD FOR OP	W/O	SBS	GBS
BR W/O GEN. BELIEF	17.49± 0.89	18.47± 0.85	18.54±0.81
BR W/ GEN. BELIEF	18.11±0.82	18.68±0.87	18.99±0.84

Table 1. Scores achieved by the various methods when playing against several test policies. The mean and standard error of the mean are reported here. W/o indicates the method without search applied on top. SBS (single belief search) is search applied with a Single belief model, and the quantity here denotes the average SBS across several Single beliefs. GBS (generalized belief search) is search applied with a generalized (Multi) belief model. Search considers 200 rollouts per legal move on the agent's turn.

Table 1 contains the results of our experiments on the generalized belief's ability to improve cross-play. We again use the thirteen SAD policies from Hu et al. (2020), which achieve average cross-play scores of 6.258 ± 1.51, and twelve OP policies, which achieve average cross-play scores of 16.88 ± 0.96 (recall that 25 is a perfect score in Hanabi). For each pool of policies, we randomly choose six of the polices to train a generalized belief model. We also train best response functions over these six policies (with and without the generalized belief hidden state as input), and use these six policies to conduct search rollouts as described in Section 5.1. We then train six Single SAD models and six Single OP models using the six chosen policies from each pool, so to use these Single beliefs for search and compare performance with the generalized belief. We reserve the remaining policies ($13 - 6 = 7$ SAD policies, and $12 - 6 = 6$

OP policies) for testing to evaluate generalization ability. Of particular note, we test the ability of the SAD beliefs to improve cross-play of OP policies, thus testing generalization ability to policies outside of the population trained with.

To determine statistical significance between the methods in Table 1, we conduct Monte Carlo permutation tests (Eden & Yates, 1933; Dwass, 1957), for which we bound each derived p-value in a 99% binomial confidence interval.

We write the "BR w/ gen. belief" to refer to the best response trained with the generalized belief's hidden state as, and similarly write "BR w/o gen. belief" to refer to the best response trained without the generalized belief's hidden state. We write "x with GBS applied" and "x with SBS applied" to refer to a policy x with search applied, with the generalized belief and the single belief, respectively.

We first consider the intra-group cases; that is, using the SAD beliefs for SAD cross-play and OP beliefs for OP cross-play. The p-value for the one-tailed test for whether the SAD BR w/ gen. belief is different from the SAD BR w/o gen. belief is between $(0, 0.0000515]$, making the difference significant at the level $\alpha = 0.01$. For the OP case, the p-value is between $[0.0578, 0.0633]$, making the difference significant at the level $\alpha = 0.1$. Thus while both best responses trained with and without the generalized belief make use of the same training policies, the generalized belief offers a highly statistically significant edge for coordination for both pools of policies.

It can be seen that both SBS and GBS improve the performance of the BR w/o gen. belief for both SAD and OP. However, the p-value for the one-tailed test for whether the SAD BR w/o gen. belief with GBS applied is different from the SAD BR w/o gen. belief with SBS applied is between $[0.0808, 0.0872]$, making the difference significant at the level $\alpha = 0.1$. In the OP case, the p-value is between $[0.0349, 0.0393]$, making the difference significant at the level $\alpha = 0.05$. As well, in the cases with the BR w/ gen. belief, SBS on average worsens performance. These observations in combination demonstrate the necessity of the generalized belief for search, and isolate its contribution from the search itself.

The one-tailed test between the SAD BR w/o gen. belief with GBS applied and the SAD BR w/o gen. belief without search applied finds the p-value to be between $[0.00131, 0.00223]$, making the difference significant at the level $\alpha = 0.01$. In contrast, the one-tailed test between the BR w/ gen. belief with GBS applied and the BR w/ gen. belief without search applied finds the p-value to be between $[0.451, 0.462]$, and similarly for the OP case the p-value is between $[0.393, 0.404]$. This shows that further applying GBS onto the BR w/ gen. belief does not confer a statistically significant advantage, in spite of search averaging over thousands of variations with rollouts leveraging a diverse pool of agents, and as such suggests that the BR w/ gen. belief has learned sufficiently high-order play to bypass the need for costly search rollouts.

Now we consider the inter-group case;, that is, using the SAD beliefs for OP cross-play. The p-value for the one-tailed test for whether the BR w/ gen. belief is different from the BR w/o gen. belief is between $[0.0431, 0.0467]$, making the difference significant at the level $\alpha = 0.05$. The p-value for the one-tailed test for whether the BR w/ gen. belief with GBS applied is different from the BR w/ gen. belief with SBS applied is between $[0.0926, 0.0978]$, making the difference significant at the level $\alpha = 0.1$. This shows that improvement can be realised across policies from different populations (e.g. SAD beliefs for OP policies), evincing the generality of our method.

Sections 4 and 5 demonstrate in combination over an array of policies and over a variety of methods that the generalized belief indeed learns generalizable conventions, such that in Section 5 the generalized belief's adaptability yields improved cross-play. Further, the generalized belief allows improvement in cross-play without having to constrain against the conventions the policies are able to form. Next, we take a closer look at what the belief model learns.

6. Belief Model Introspection

Attention Heads One of the advantages of attention-based architectures is the transparency and interpretability of the attention weights. Appendix E shows some of the attention patterns learned in our autoregressive model, that show how "pertinent" the model views a given timestep x when considering a timestep y.

State Feature Embeddings We can also visualise the representations learned for the observable state features by applying singular value decomposition (Eckart & Young, 1936) to the $K \times d_{\text{feature}}$ matrix of feature embeddings (see Section 3) and considering the first two singular values to obtain the principal components as in Figure 4.

Because we input the state features into Ψ_E in a fixed concatenated order, the linear layer that aggregates the projected representations of the features learns this order and weights each feature accordingly. Hence for our context, the position each state feature embedding occupies in the vector space relative to different types of state features does not matter. However, the distribution of principal components relative to a single state feature type, and in particular, which state feature types have the most variance, indicates the expressiveness of a state feature type (if all the learned representations of a state feature type are similar, then the state feature is not very informative).

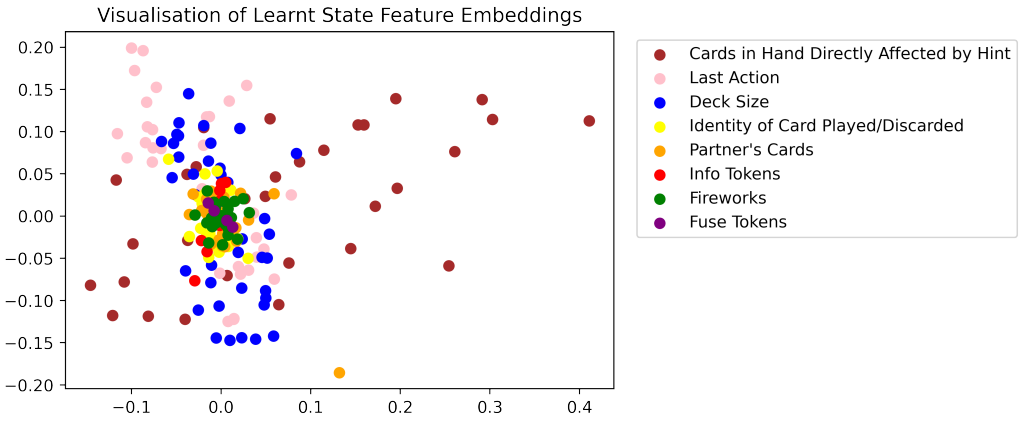

Figure 4. 2-D projection of the embedding representations learned through the context of Hanabi gameplay of SAD policies (see Section 3 for how these representations are learned). The legend is ordered by features with most variance to least. More variance indicates more importance.

Figure 4 shows that the observable state feature types with the most variation are which cards in hand were directly hinted,[4] the last action executed in the game, and the remaining size of the deck, indicating that the model found these features to be most informative for forming belief. While it is clear that much of the grounded information is manifested in these state features, we conclude that the policy also must mostly use these features to convey the implicit information because the variances of the other features are much lower. This may take the form of a secondary meaning attached to a hint based on which cards were directly hinted, or a pre-agreed convention on how to act relative to the number of cards remaining in the deck.

Thus while it may in general be difficult to interpret the nuanced tactics employed by an expert policy, these visual analytics can provide evidence regarding the style of play.

7. Related Work

Learning to Coordinate A central challenge in AI is devising agents that can coordinate with a novel partner (Canaan et al., 2020; Carroll et al., 2019; Hu et al., 2020; Kleiman-Weiner et al., 2016; Papoudakis & Albrecht, 2020; Smith et al., 2020; Stone et al., 2010; Zintgraf et al., 2021). As mentioned in Section 1, naively applying self-play approaches to one another can yield poor performance. While dynamic programming approaches may be considered as an approach to Dec-POMDPs (Hansen et al., 2004), such planning methods assume that at test time an agent's teammates will execute their part of the same centrally planned joint policy as them. In Hu et al. (2020), a policy is taught to pay attention to symmetry in the Dec-POMDP at training time so as to form generalizable conventions that can enhance cross-play performance. In this way, the agent aims to learn an unambiguous but potentially sub-optimal policy. In Shih et al. (2021), a modular approach is adopted to separately learn rule-dependent and convention-dependent behaviours to facilitate coordination, where their experiments are focused on a small version of Hanabi. See Section 1 for how our method compares with Hu et al. (2020) and Shih et al. (2021).

Lupu et al. (2021) consider training a diverse pool of policies suited for zero-shot coordination, whereas we focus on aggregating the conventions of existing policies to generalize over new ones. Hu et al. (2021b) propose using a belief model trained over a fixed policy, and train a new policy that predicates decisions on samples from the trained belief. The ability of the new policy to then effectively coordinate with a test class of partner policies is highly dependent on the relation between the fixed policy the belief is trained on, and the test policies. In contrast, we train our autoregressive belief on multiple policies to aggregate a diversity of convention-dependent behaviours.

An important difference with our work and other approaches is as follows: existing literature often proposes methods for constraining policy training in such a way to allow trained policies to effectively engage in cross-play amongst one another, and these works focus experimentation in this realm. In contrast, we propose a method for improving cross-play ability against classes of policies not necessarily pre-configured for cross-play (i.e. ad-hoc teamplay), and demonstrate that this can indeed be done.

[4]We say "directly" hinted to mean the cards that receive the explicit grounded information: e.g. if I hint that your second and fourth cards are of rank 1, then your second and fourth cards were directly hinted; meanwhile your first, third and fifth cards were indirectly hinted of not being of rank 1.

Self-Attention and Transformers in Reinforcement Learning Parisotto et al. (2020) proposed the addition of extra layers into the transformer architecture to stabilize training in the high-variance RL domain, where these additional layers could possibly be used to enhance the methodology proposed in this paper, which we leave for future work. Chen et al. (2021) proposed using the transformer architecture for tasks in offline RL. Hu et al. (2021c) explore the usage of semantic embeddings over observable features for the transformer architecture, but we improve both on input scalability and providing context to the data by aggregating the features with a fully-connected layer (see Section 3).

8. Conclusion

In this paper, we proposed a method for learning a generalized belief across multiple policies, where the model learns to reason over the specialized conventions of each policy. We tested this model on diverse collections of policies that, such that especially for the case of the SAD policies, were unable to collaborate well in cross-play due to their high degree of conventional specialization. Our proposed model was nonetheless able to factorise the respective conventions of these policies and understand the implicit information they conveyed. We also showed that training over a set of conventions can suffice for generalizing to new ones, so that one can decode the intent of a given trajectory based on the known intents of other policies and the learned symmetry in the Dec-POMDP; we showed all this can be done without strong environmental assumptions, and without constraining policy training. We further leveraged this belief model to improve coordination, and proposed two frameworks under which the generalized belief can be used. In particular, we found that inter-population generalization was possible with our frameworks, such that beliefs trained over one population could be used to improve coordination over another (i.e. SAD beliefs for OP policies).

In addition, we proposed architectural adaptations to suit the transformer (and other self-attention based models) to the MARL domain, and empirically verified the ability for a model to effectively learn under the architectural adaptations. Finally, we proposed visual analytic schemes to help with interpretability of the belief and policy, which is critical for increasing transparency and trust of RL systems (Wells & Bednarz, 2021).

In future work, we will explore even larger Dec-POMDPs, perhaps larger variants of Hanabi, or realistic settings: for realistic settings, one has a choice in one of two beliefs: 1) enumerating the unobservable features of the environment and learning the per-feature belief; 2) considering only the agent's action space and learning the semantic belief. One may train a best response with either belief, as in the paper, or with the per-feature belief one may additionally use a search scheme, as in the paper.

Acknowledgements

The experiments were made possible by a generous equipment grant from NVIDIA. Luisa Zintgraf is supported by the 2017 Microsoft Research PhD Scholarship Program, and the 2020 Microsoft Research EMEA PhD Award. Christian Schroeder de Witt is generously funded by the Cooperative AI Foundation.

References

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Baker, C. L., Jara-Ettinger, J., Saxe, R., and Tenenbaum, J. B. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour*, 1(4):1–10, 2017.

Bard, N., Foerster, J. N., Chandar, S., Burch, N., Lanctot, M., Song, H. F., Parisotto, E., Dumoulin, V., Moitra, S., Hughes, E., et al. The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, 280:103216, 2020.

Benesty, J., Chen, J., Huang, Y., and Cohen, I. Pearson correlation coefficient. In *Noise reduction in speech processing*, pp. 1–4. Springer, 2009.

Brown, N., Bakhtin, A., Lerer, A., and Gong, Q. Combining deep reinforcement learning and search for imperfect-information games. *arXiv preprint arXiv:2007.13544*, 2020.

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

Canaan, R., Gao, X., Chung, Y., Togelius, J., Nealen, A., and Menzel, S. Evaluating the rainbow dqn agent in hanabi with unseen partners. *arXiv preprint arXiv:2004.13291*, 2020.

Carroll, M., Shah, R., Ho, M. K., Griffiths, T., Seshia, S., Abbeel, P., and Dragan, A. On the utility of learning about humans for human-ai coordination. *Advances in Neural Information Processing Systems*, 32:5174–5185, 2019.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. *arXiv preprint arXiv:2106.01345*, 2021.

Dwass, M. Modified randomization tests for nonparametric hypotheses. *The Annals of Mathematical Statistics*, pp. 181–187, 1957.

Eckart, C. and Young, G. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Eden, T. and Yates, F. On the validity of fisher's z test when applied to an actual example of non-normal data.(with five text-figures.). *The Journal of Agricultural Science*, 23(1):6–17, 1933.

Gangwani, T., Lehman, J., Liu, Q., and Peng, J. Learning belief representations for imitation learning in pomdps. In *Uncertainty in Artificial Intelligence*, pp. 1061–1071. PMLR, 2020.

Hansen, E. A., Bernstein, D. S., and Zilberstein, S. Dynamic programming for partially observable stochastic games. In *AAAI*, volume 4, pp. 709–715, 2004.

Hausknecht, M. and Stone, P. Deep recurrent q-learning for partially observable mdps. In *2015 aaai fall symposium series*, 2015.

Heider, F. and Simmel, M. An experimental study of apparent behavior. *The American journal of psychology*, 57(2): 243–259, 1944.

Hu, H. and Foerster, J. N. Simplified action decoder for deep multi-agent reinforcement learning. *arXiv preprint arXiv:1912.02288*, 2019.

Hu, H., Lerer, A., Peysakhovich, A., and Foerster, J. "other-play" for zero-shot coordination. In *International Conference on Machine Learning*, pp. 4399–4410. PMLR, 2020.

Hu, H., Lerer, A., Brown, N., and Foerster, J. Learned belief search: Efficiently improving policies in partially observable settings. *arXiv preprint arXiv:2106.09086*, 2021a.

Hu, H., Lerer, A., Cui, B., Pineda, L., Wu, D., Brown, N., and Foerster, J. Off-belief learning. *arXiv preprint arXiv:2103.04000*, 2021b.

Hu, S., Zhu, F., Chang, X., and Liang, X. Updet: Universal multi-agent reinforcement learning via policy decoupling with transformers. *arXiv preprint arXiv:2101.08001*, 2021c.

Jensen, J. L. W. V. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta mathematica*, 30(1):175–193, 1906.

Kleiman-Weiner, M., Ho, M. K., Austerweil, J. L., Littman, M. L., and Tenenbaum, J. B. Coordinate to cooperate or compete: abstract goals and joint intentions in social interaction. In *CogSci*, 2016.

Labeau, M. *Neural language models: Dealing with large vocabularies*. PhD thesis, Université Paris-Saclay (ComUE), 2018.

Lerer, A., Hu, H., Foerster, J., and Brown, N. Improving policies via search in cooperative partially observable games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7187–7194, 2020.

Lupu, A., Cui, B., Hu, H., and Foerster, J. Trajectory diversity for zero-shot coordination. In *International Conference on Machine Learning*, pp. 7204–7213. PMLR, 2021.

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

Moreno, P., Humplik, J., Papamakarios, G., Pires, B. A., Buesing, L., Heess, N., and Weber, T. Neural belief states for partially observed domains. In *NeurIPS 2018 workshop on Reinforcement Learning under Partial Observability*, 2018.

Oliehoek, F. A. Decentralized pomdps. In *Reinforcement Learning*, pp. 471–503. Springer, 2012.

Papoudakis, G. and Albrecht, S. V. Variational autoencoders for opponent modeling in multi-agent systems. *arXiv preprint arXiv:2001.10829*, 2020.

Parisotto, E., Song, F., Rae, J., Pascanu, R., Gulcehre, C., Jayakumar, S., Jaderberg, M., Kaufman, R. L., Clark, A., Noury, S., et al. Stabilizing transformers for reinforcement learning. In *International Conference on Machine Learning*, pp. 7487–7498. PMLR, 2020.

Samuel, A. L. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229, 1959.

Schroeder de Witt, C., Foerster, J., Farquhar, G., Torr, P., Boehmer, W., and Whiteson, S. Multi-agent common knowledge reinforcement learning. *Advances in Neural Information Processing Systems*, 32:9927–9939, 2019.

Shih, A., Sawhney, A., Kondic, J., Ermon, S., and Sadigh, D. On the critical role of conventions in adaptive human-ai collaboration. *arXiv preprint arXiv:2104.02871*, 2021.

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

Smith, M. O., Anthony, T., Wang, Y., and Wellman, M. P. Learning to play against any mixture of opponents. *arXiv preprint arXiv:2009.14180*, 2020.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

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

Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J. Z., Tuyls, K., et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

Tan, M. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, pp. 330–337, 1993.

Tian, Z., Zou, S., Davies, I., Warr, T., Wu, L., Ammar, H. B., and Wang, J. Learning to communicate implicitly by actions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7261–7268, 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Wells, L. and Bednarz, T. Explainable ai and reinforcement learning—a systematic review of current approaches and trends. *Frontiers in artificial intelligence*, 4:48, 2021.

Yu, C., Velu, A., Vinitsky, E., Wang, Y., Bayen, A., and Wu, Y. The surprising effectiveness of mappo in cooperative, multi-agent games. *arXiv preprint arXiv:2103.01955*, 2021.

Zhang, M., Levine, S., McCarthy, Z., Finn, C., and Abbeel, P. Policy learning with continuous memory states for partially observed robotic control. *CoRR, abs/1507.01273*, 2015.

Zhu, P., Li, X., Poupart, P., and Miao, G. On improving deep reinforcement learning for pomdps. *arXiv preprint arXiv:1704.07978*, 2017.

Zintgraf, L., Shiarlis, K., Igl, M., Schulze, S., Gal, Y., Hofmann, K., and Whiteson, S. Varibad: A very good method for bayes-adaptive deep rl via meta-learning. In *International Conference on Learning Representation (ICLR)*, 2020.

Zintgraf, L., Devlin, S., Ciosek, K., Whiteson, S., and Hofmann, K. Deep interactive bayesian reinforcement learning via meta-learning. *AAMAS (Extended Abstract)*, 2021.

A. Hanabi

Hanabi is a cooperative card game that can be played with 2 to 5 people. Hanabi is a popular game, having been crowned the 2013 "Spiel des Jahres" award, a German industry award given to the best board game of the year. Hanabi has been proposed as an AI benchmark task to test models of cooperative play that act under partial information (Bard et al., 2020). To date, Hanabi has one of the largest state spaces of all Dec-POMDP benchmarks.

The deck of cards in Hanabi is comprised of five colours (white, yellow, green, blue and red), and five ranks (1 through 5), where for each colour there are three 1's, two each of 2's, 3's and 4's, and one 5, for a total deck size of fifty cards. Each player is dealt five cards (or four cards if there are 4 or 5 players). At the start, the players collectively have eight information tokens and three fuse tokens, the uses of which shall be explained presently.

In Hanabi, players can see all other players' hands but their own. The goal of the game is to play cards so as to collectively form five consecutively ordered stacks, one for each colour, beginning with a card of rank 1 and ending with a card of rank 5. These stacks are referred to as fireworks, as playing the cards in order is meant to draw analogy to setting up a firework display.[5]

We call the player whose turn it is the active agent. The active agent must conduct one of three actions:

- **Hint** - The active agent chooses another player to grant a hint to. A hint involves the active agent choosing a colour or rank, and revealing to their chosen partner all cards in the partner's hand that satisfy the chosen colour or rank. Performing a hint exhausts an information token. If the players have no information tokens, a hint may not be conducted and the active agent must either conduct a discard or a play.
- **Discard** - The active agent chooses one of the cards in their hand to discard. The identity of the discarded card is revealed to the active agent and becomes public information. Discarding a card replenishes an information token should the players have less than eight.
- **Play** - The active agent attempts to play one of the cards in their hand. The identity of the played card is revealed to the active agent and becomes public information. The active agent has played successfully if their played card is the next in the firework of its colour to be played, and the played card is then added to the sequence. If a firework is completed, the players receive a new information token should they have less than eight. If the player is unsuccessful, the card is discarded, without replenishment of an information token, and the players lose a fuse token.

The game ends when all three fuse tokens are spent, when the players successfully complete all five fireworks, or when the last card in the deck is drawn and all players take one last turn. If the game finishes by depletion of all fuse tokens, the players receive a score of 0. Otherwise, the score of the finished game is the sum of the highest card ranks in each firework, for a highest possible score of 25.

[5] Hanabi (花火) means 'fireworks' in Japanese.

B. Autoregressive Beliefs

Firstly, we assume that we can factor the unobservable state features from s_t, as is the usual case for Dec-POMDPs. Proceeding, let $H = |\{c_1, \ldots, c_H\}|$ be the number of unobservable state features, and assume C is the number of different values each c_h may take on. We thus take the belief to be over these unobservable features, and consider the identification of belief learning as an autoregressive task (Hu et al., 2021a); namely, to find

$$\psi \quad \text{s.t.} \quad c_h \sim \psi(c_h \mid \mathbf{c}_{<h}, \tau_t^i), \tag{4}$$

where ψ is our probabilistic mapping between the action-observation history τ_t^i and the unobservable state features $c_1, \ldots, c_H$, where $\mathbf{c}_h := (c_1, \ldots, c_{h-1})$ and $\mathbf{c}_1 := \emptyset$. In which case, for ψ parameterized as ψ_θ, we produce the belief approximation

$$b_t^i \approx \prod_{h=1}^{H} \psi_\theta(c_h \mid \mathbf{c}_{<h}, \tau_t^i), \tag{5}$$

which we learn in a supervised fashion through minimization of the cross entropy loss,

$$CE(\psi_\theta, \tau_t) = -\frac{1}{H} \sum_{h=1}^{H} \sum_{x=1}^{C} \mathbb{1}_{x=c_h}(x \mid s_t) \log \psi_\theta(x \mid \mathbf{c}_{<h}, \tau_t^i). \tag{6}$$

C. Cross Entropy Interpretation

Definition C.1. For an agent i, ψ is said to have narrowed down $c_1, \ldots, c_H$ over τ_t^i to at most n if $n \geq e^{\mathbb{E}_{p(\tau_t)}[CE(\psi, \tau_t)]}$.

This definition is motivated as follows: first suppose the indicator function corresponding to each unobservable state feature h is chosen uniformly at random over C. Then,

$$\begin{aligned}
\mathbb{E}_{p(\mathbb{1})}[CE(\psi, \tau_t)] &= -\frac{1}{CH} \sum_{h=1}^{H} \sum_{x=1}^{C} \log \psi(x \mid \mathbf{c}_{<h}, \tau_t^i) \\
&\geq -\log\Big(\frac{1}{CH} \sum_{h=1}^{H} \sum_{x=1}^{C} \psi(x \mid \mathbf{c}_{<h}, \tau_t^i)\Big) \\
&= -\log\Big(\frac{1}{CH} \sum_{h=1}^{H} 1\Big) \\
&= -\log \frac{1}{C},
\end{aligned}$$

where the second line follows from Jensen's inequality and convexity of the negative logarithm (Jensen, 1906), and the third line follows from probabilities summing to 1.

But note that in a Dec-POMDP where the distribution over unobservable state features is uniform (e.g. a shuffled deck of cards in Hanabi), we have

$$\mathbb{E}_{p(\mathbb{1})}[CE(\psi, \tau_t)] = \mathbb{E}_{p(\tau_t)}[CE(\psi, \tau_t)]. \tag{7}$$

Hence $n \geq e^{\mathbb{E}_{p(\tau_t)}[CE(\psi, \tau_t)]} \geq e^{-\log \frac{1}{C}} = C$; that is, the number of different values the unobservable features may take on is at most n.

One can thus use the Monte Carlo estimate

$$e^{\mathbb{E}_{p(\tau_t)}[CE(\psi, \tau_t)]} \approx e^{\frac{1}{m} \sum_{\tau_{t,l}} CE(\psi, \tau_{t,l})}, \tag{8}$$

where $\sum_{\tau_{t,l}}$ denotes a sum over m randomly chosen trajectories.

To give examples, for the Hanabi control task, when the belief model achieves an average cross entropy score less than or equal to $-\log \frac{1}{5} \approx 1.609$, it will have narrowed down the number of possible cards that the average card in hand can be to at most 5. When the model knows the exact identity of a card, it will achieve a cross entropy score of $-\log 1 = 0$.

D. Experimental Setup

We base our training setup on the codebases of Hu & Foerster (2019); Hu et al. (2021b; 2020), which we link here: https://github.com/facebookresearch/hanabi_SAD, https://github.com/facebookresearch/off-belief-learning. With these links, one will also find the pre-trained SAD policies used in this work.

The machine used for experimentation consisted of 2 NVIDIA GeForce RTX 2080 Ti GPUs and 40 CPU cores.

D.1. Belief Learning

The code for learning a belief model with the transformer architecture may be found here: https://github.com/gfppoy/hanabi-belief-transformer.

Table 2. Hyperparameter settings of transformer for belief emulation.

Hyperparameters	Value
Number of layers	6
Number of attention heads	8
State embedding dimension (d in Section 3)	256
Feature embedding dimension (d_{feature} in Section 3)	128
Maximum sequence length (T in Section 3)	80
Feedforward network dimension	2048
Nonlinearity	ReLU
Batchsize	256
Dropout	0.1
Learning rate	2.5×10^{-4}
Warm-up period	10^5
Learning rate decay	Inverse square root

D.2. Improving Cross-Play

The code for learning a best response without the belief model hidden state may be found here: https://github.com/gfppoy/hanabi-br-wobelief.

The code for learning a best response with the belief model hidden state may be found here: https://github.com/gfppoy/hanabi-br-withbelief.

The code for generalized belief search may be found here: https://github.com/gfppoy/hanabi-gbs.

E. Attention Head Visualisations

The following are visualisations of the attention heads in our generalized belief models trained over the SAD policies, where here it is emulating the belief distribution at the 45^{th} timestep of a randomly generated game of Hanabi; a point (x, y) on the visualisation represents how "pertinent" the model views timestep x when considering timestep y, where x varies along the horizontal axis and y along the vertical axis. The visualisation shows the model found that considering adjacent turns provides more context for deducing implicit cooperative cues.

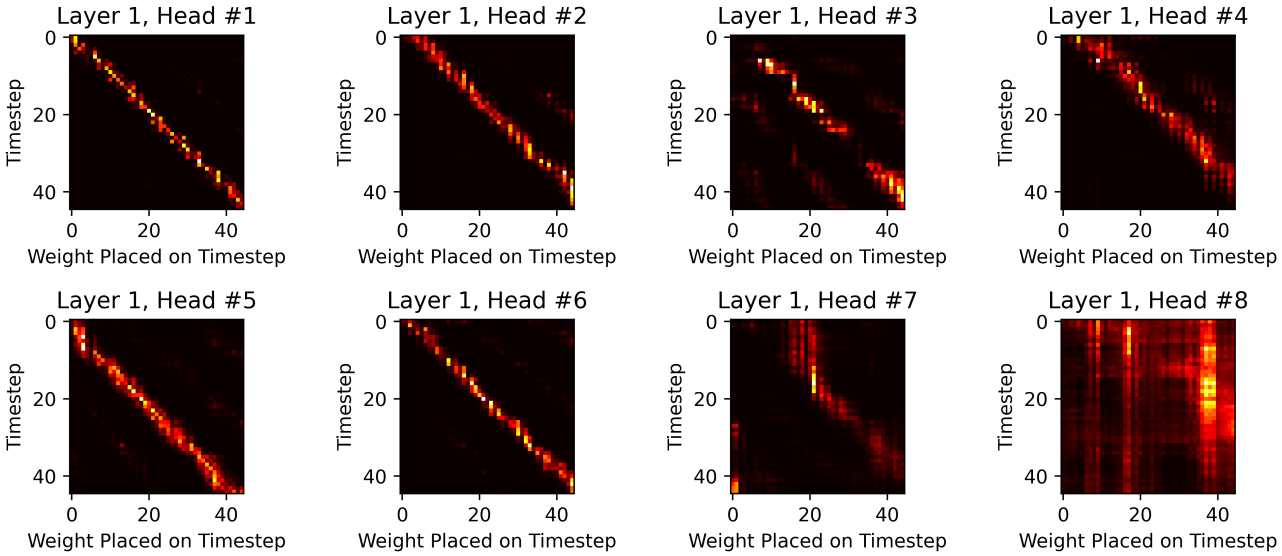

Figure 5. The 8 attention head weights in the first layer.

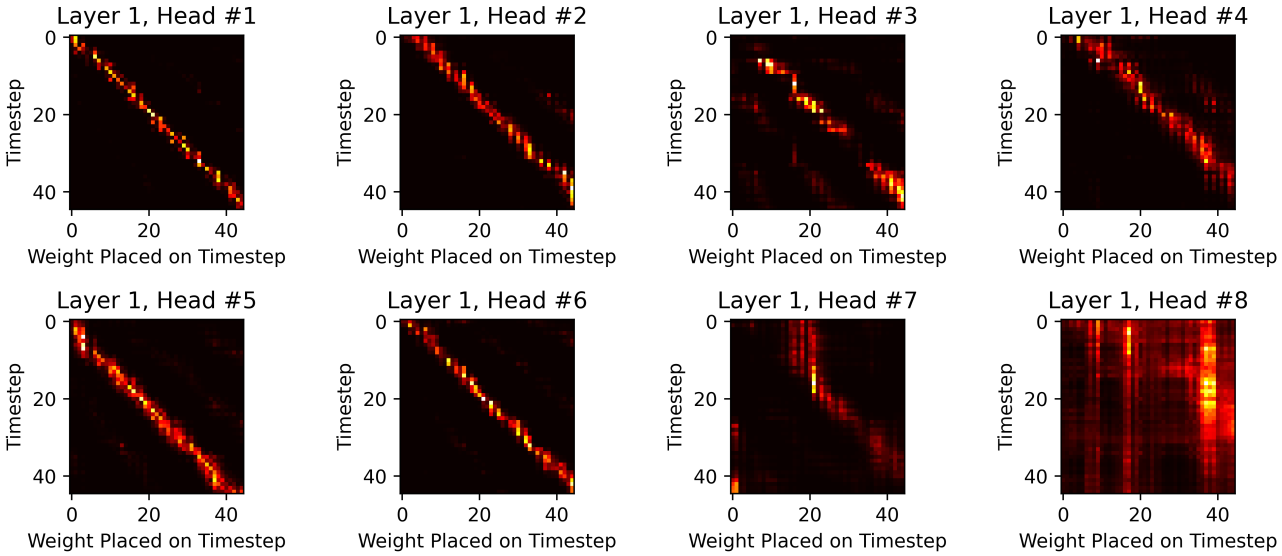

Figure 6. The 8 attention head weights in the first layer.

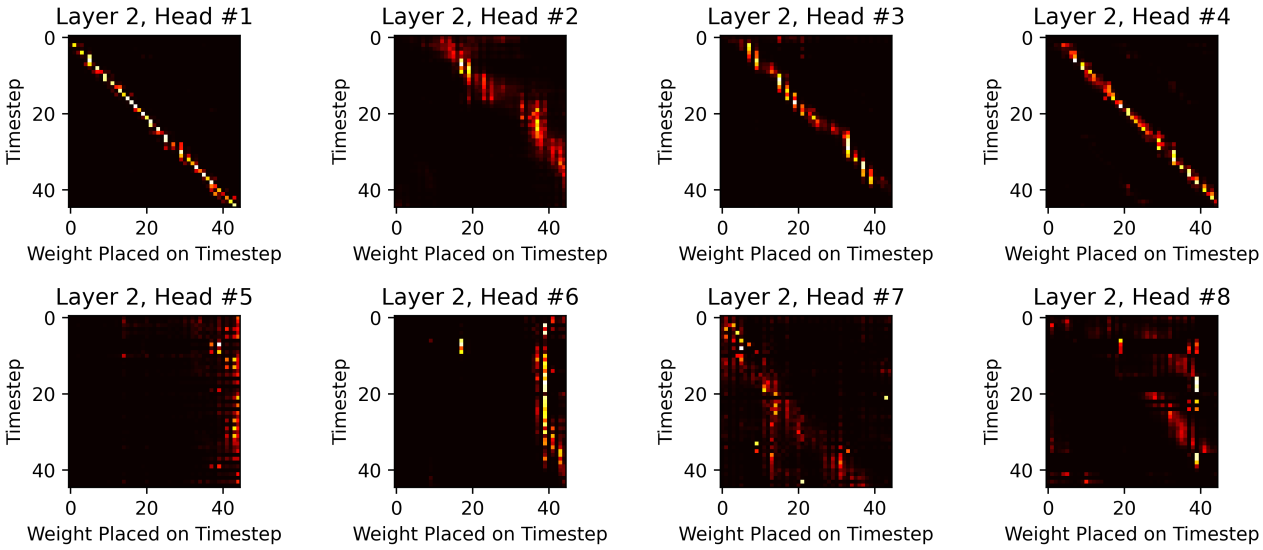

Figure 7. The 8 attention head weights in the second layer.

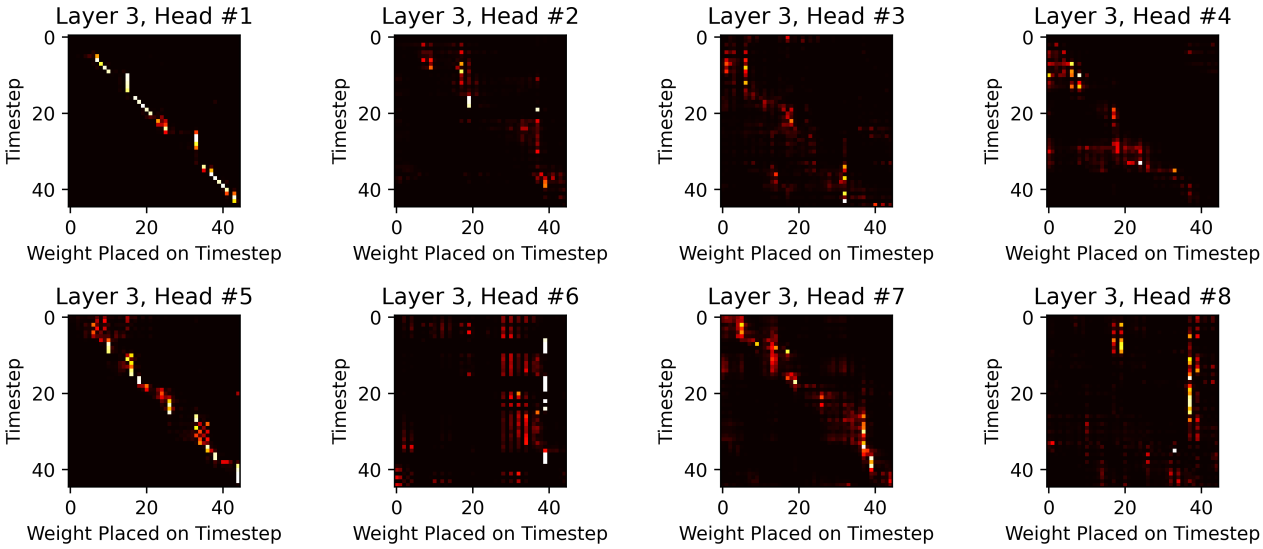

Figure 8. The 8 attention head weights in the third layer.

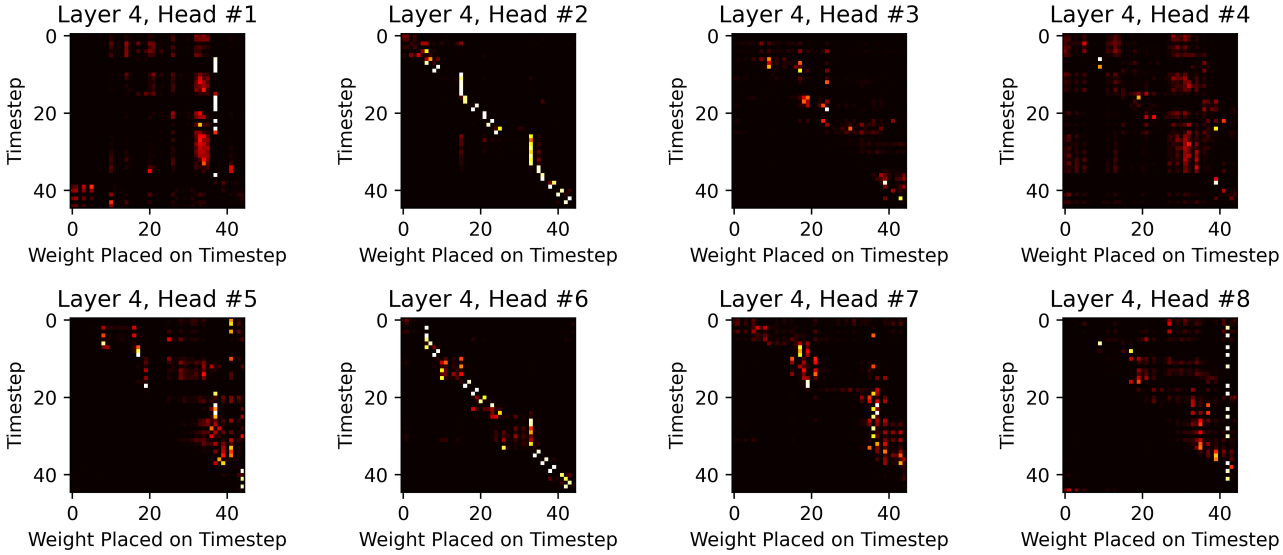

Figure 9. The 8 attention head weights in the fourth layer.

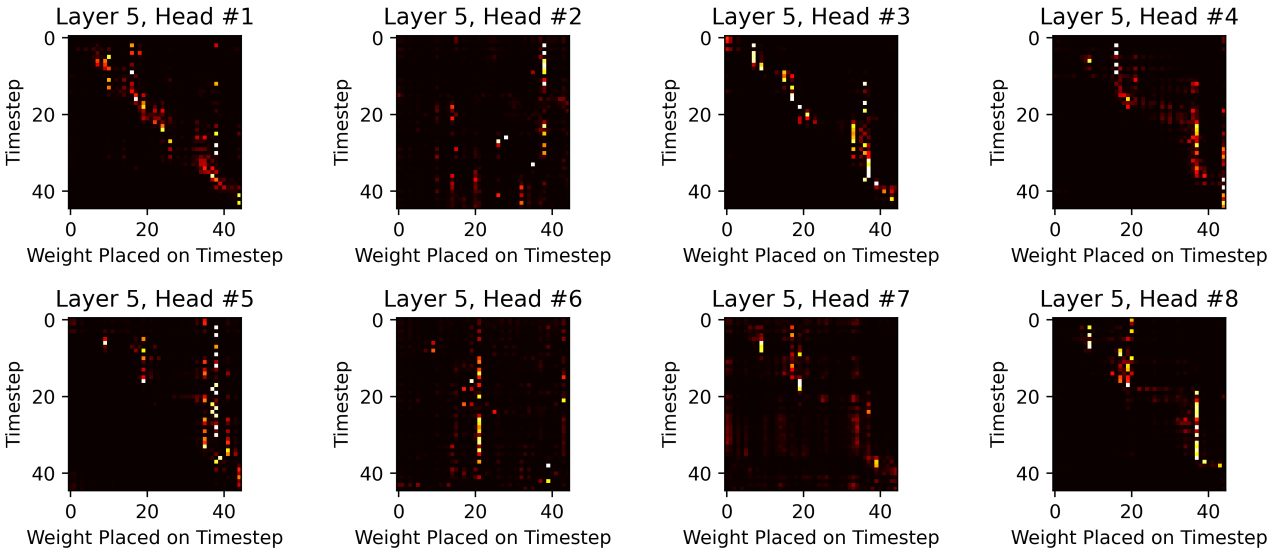

Figure 10. The 8 attention head weights in the fifth layer.

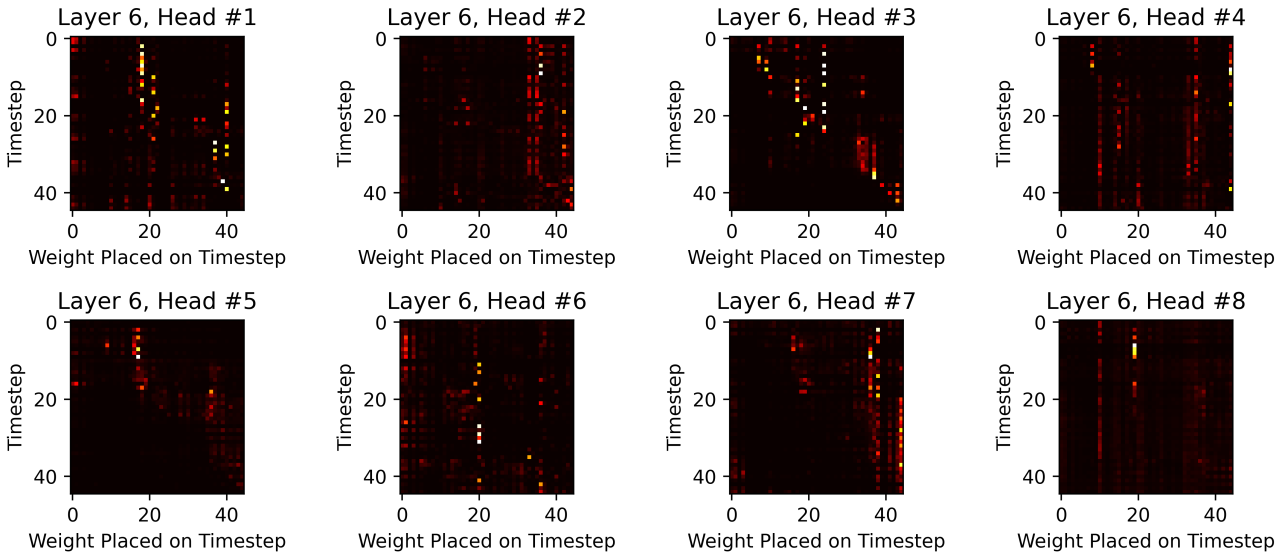

Figure 11. The 8 attention head weights in the sixth layer.

F. Distributions of Play

The figures illustrate the Hanabi cross-play score frequencies of various policies matched with SAD and OP policies, respectively. The best responses and beliefs here are trained over a pool of SAD policies and OP policies, respectively.

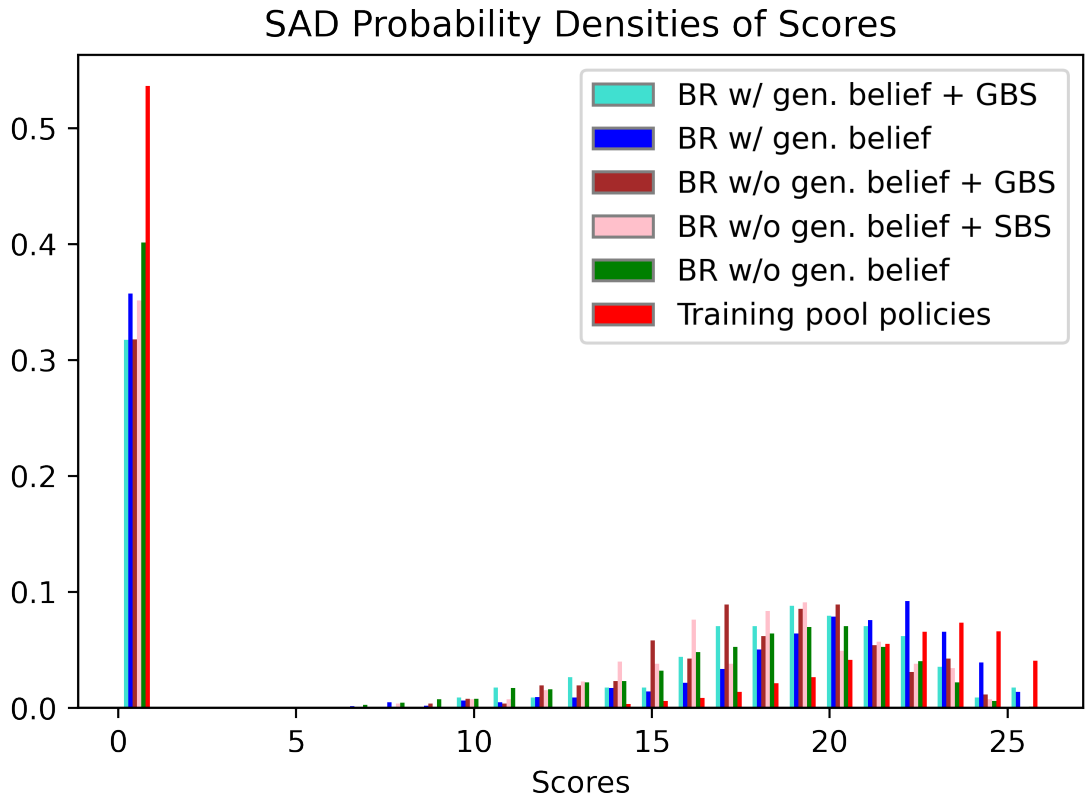

Figure 12. Probability densities of scores attained by various SAD models.

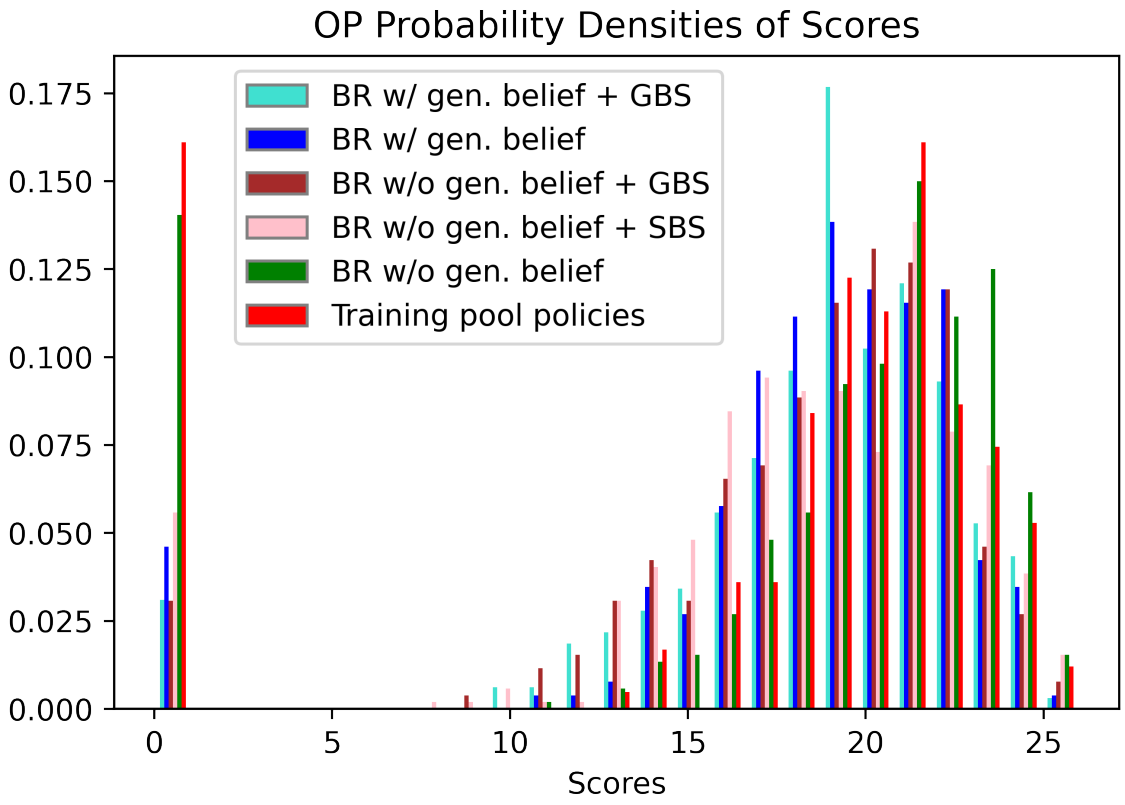

Figure 13. Probability densities of scores attained by various OP models.

G. LSTM vs Transformer Beliefs

Here we compare the performance of an LSTM encoder-decoder architecture with that of the transformer architecture used in this work for maintaining belief of policies not seen at training time. Transformers have advantages in parallelizability and interpretability, but here we demonstrate additional advantages in the ability to maintain belief. For both architectures, we use our proposed embedding mechanism (see Section 3).

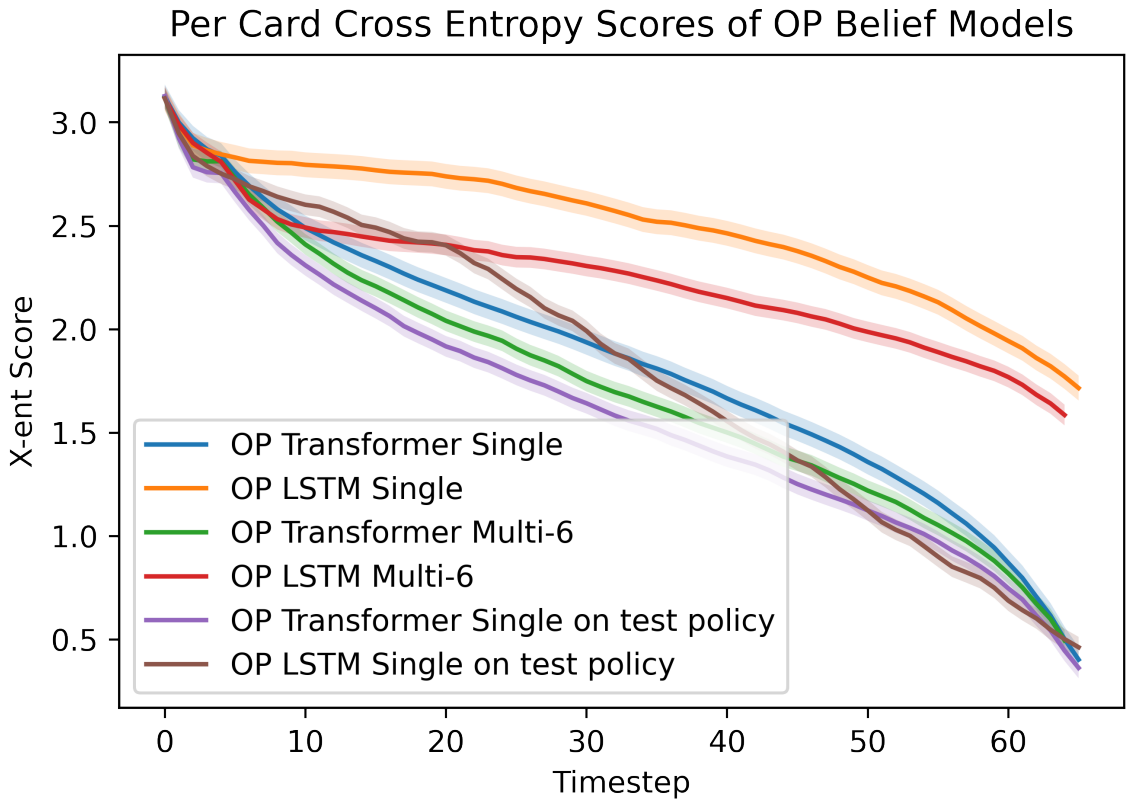

Figure 14. Per card cross entropy (X-ent) scores of the averages of the Single models and Multi-6 models, all tasked with maintaining belief over trajectories featuring a policy not seen at training time. The belief model trained on the test policy itself is provided here for reference. The shading corresponds to the standard error of the mean at each timestep. The curves were computed over 20k randomly generated games.