# Off-Team Learning

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

*Zero-shot coordination (ZSC)* is a proxy framework and pre-requisite for human-AI coordination and evaluates agents under cross-play (XP), i.e. paired with partners independently trained with the same algorithm. *Off-belief learning (OBL)* is a recent method for ZSC which prevents agents from learning arbitrary conventions. It achieves state-of-the-art results in ZSC, ad-hoc teamplay, and proxy human-AI coordination in the complex card game Hanabi. However, OBL suffers from two issues: First of all, it relies heavily on a belief model which is trained on a fixed, given policy $\pi_0$, but evaluated on the *trained* policy $\pi_1$, potentially leading to large covariate shifts. Secondly, OBL agents are trained only on the *narrow* distribution induced by their training partner, which means they can easily be taken off distribution at test time. We present two methods addressing these issues. The first, off-team belief learning (OT-BL), is a general algorithm that allows training belief models off-team—i.e., to learn the belief for a given policy, $\pi$ on the distribution induced by a different team, $\pi_b$. The second, off-team reinforcement learning (OT-RL), allows training a policy off-team—i.e., learning values and policies corresponding to one policy, $\pi$, on rollouts from a different team, $\pi_b$. We empirically show that OBL+OT-BL outperforms OBL in challenging variants of Hanabi. Furthermore, we show that OBL+OT-RL achieves near-optimal ZSC and can be applied to mitigate large covariate shifts in ad-hoc teamplay and proxy human-AI coordination.

## 1 Introduction

A core objective of multi-agent reinforcement learning (MARL) is to develop methods for coordinating with previously unseen partners such as other artificial agents or humans. MARL has been studied through the lens of *self-play (SP)*, where agents are only evaluated with their training partners [21]. In partially observable, fully cooperative settings SP policies develop efficient but opaque arbitrary conventions that make collaboration with novel partners, or even independent training runs of the same algorithm, extremely difficult [6, 4]. To study this issue, the zero-shot coordination (ZSC) framework was introduced as a new proxy setting towards human-AI coordination [6]. Here agents are evaluated in *cross-play* (XP), i.e. with other, independently trained agents from the same method. Under ZSC agents cannot rely on arbitrary conventions and must instead adopt strategies that can be independently discovered and interpreted by other agents.

Off-belief learning (OBL) [7] is a recent MARL training algorithm designed to address the problem of arbitrary conventions in ZSC. OBL starts by training a policy $\pi_1$ that reinterprets past actions as having been taken by a fully random policy $\pi_0$. This forces $\pi_1$ to only take actions based on *grounded* information rather than inferring the private information of other agents from observing their past actions. Similarly, this grounded policy does not actively communicate private information since it assumes this information will be ignored in future time steps. OBL then iterates upon this grounded policy by training a hierarchy where each level builds upon the conventions induced by optimal play

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

at the level below. For each $\pi_\ell$, OBL trains a belief model $\hat{\mathcal{B}}_{\pi_\ell}$ through supervised learning, which is then used to interpret past actions when training $\pi_{\ell+1}$. OBL not only achieved state of the art results in ZSC and human-AI coordination on the popular benchmark Hanabi [1], but formally also converges to a unique policy given any fixed $\pi_0$.

In contrast to the empirical success, there are two major problems with the current formulation of the algorithm from [7]. First of all, the belief model $\hat{\mathcal{B}}_{\pi_\ell}$ is *trained* via supervised learning on the distribution of trajectories induced by $\pi_\ell$. However, it is *evaluated* on the trajectories induced by the OBL policy at the next level, $\pi_{\ell+1}$. When the distributions of $\pi_\ell$ and $\pi_{\ell+1}$ differ substantially, this will bring $\hat{\mathcal{B}}_{\pi_0}$ off-distribution leading to *covariate shift* and cause it to be unreliable. For instance, a fully random $\pi_0$ may lead to the early termination of episodes, while an optimal $\pi_1$ will play longer episodes. Crucially, the goal is then to learn a belief model which predicts the state of the world *had* the agent arrived at the current action-observation history by playing with $\pi_b$, e.g. a random partner, even if getting there with that partner is very unlikely in the first place.

Secondly, while OBL prevents agents from learning arbitrary conventions, OBL policies are still fundamentally brittle, since they are only trained on the distribution of trajectories induced by their training partner. As a consequence they can easily be taken *off-distribution* and experience covariate shift when paired with a novel test time partner, e.g. a human or another AI agent.

To address the covariate shift experienced by $\hat{\mathcal{B}}_{\pi_\ell}$ we propose off-team belief learning (OT-BL). Rather than training $\hat{\mathcal{B}}_{\pi_\ell}$ only on the distribution induced by $\pi_\ell$, we introduce a belief-bootstrapping operator that allows us to train our belief model *off-team* on the distribution induced by $\pi_{\ell+1}$. We evaluate OT-BL empirically and show that it resolves the covariate shift experienced by belief models when training OBL and significantly outperforms vanilla OBL in the challenging 2-life variant of the card game Hanabi.

Next, to mitigate the test time covariate shift experienced by $\pi_\ell$ when paired with an unseen agent, we propose off-team reinforcement learning (OT-RL). OT-RL builds on the insight that OBL can be applied *off-team*, allowing OBL to be trained on broader distributions to reduce the test time covariate shift. We evaluate OT-RL empirically showing it achieves near optimal, state-of-the-art XP scores, higher than standard OBL. Additionally, we apply OT-RL a small number of target trajectories from ad-hoc teamplay and proxy human-AI coordination. With this process we are able to improve upon vanilla OBL and reduce large test time covariate shift.

# 2 Related Work

## 2.1 Prior Work on Zero-Shot Coordination (ZSC)

Our framework of interest, zero-shot coordination (ZSC) formalizes the task of coordinating with partners unseen during train time. It was first introduced by [6] and further analyzed in [22], and led to a series of works meant to address the challenges of the setting. In their original paper, [6] introduced Other-Play, a method for training policies that are equivariant with respect to the symmetries of the game, thus avoiding conventions that rely on arbitrary symmetry breaking. Other works have trained hierarchies of policies for ZSC [5] or yet leveraged diverse populations to train a robust common best response [12, 19]. One of the most significant breakthroughs in ZSC is off-belief learning (OBL) [7], which we detail in section 3.3 and upon which our work builds directly.

## 2.2 Belief learning

One crucial requirement for scaling OBL to large settings is learning beliefs about the private information of other players. Belief states can be explicitly tracked. Modern approaches to this problem involve deep sequence models, such as recurrent networks [20] or transformers [25]. However, the standard training setup for these models leverages Monte Carlo sampling. In contrast, OT-BL trains beliefs models via approximate dynamic programming. In this respect, OT-BL is most similar to the work of Sokota et al. [16], who propose a framework for belief fine-tuning. In this framework, a belief model for the current time-step is used to bootstrap a belief model for the next time-step using approximate dynamic programming. Sokota et al. show that belief fine-tuning can improve the quality of belief models and adapt them to new policies at inference time. As mentioned above, belief fine-tuning and OT-BL are similar in that they use bootstrapping to learn beliefs, however whereas Sokota et al. [16] examined belief fine-tuning in an inference-time context, we present OT-BL as a mechanism for performing off-team train-time learning.

# 3 Background

## 3.1 Dec-POMDPs

This work considers decentralized partially observable Markov decision processes (Dec-POMDP) [3]. Dec-POMDPs have state space $s \in S$, observation functions $o^i = \Omega^i(s)$ for each agent $i$ through which they obtain a partial observation of the underlying state, joint action space $A = A^1 \times A^2 \times \cdots \times A^N$, reward function $R\colon S \times A \to \mathbb{R}$ and transition function $\mathcal{T}\colon S \times A \to \Delta S$ that defines the distribution of next state given current state and joint action. We further denote a trajectory as $\tau_t = (s_1, a_1, r_1, \cdots, s_t)$ and each players action observation history (AOH) $\tau_t^i = (o_1^i, a_1, r_1, \cdots, o_t^i)$. When $t$ equals to terminal $T$, the subscript is dropped and the two notations are simplified as $\tau$ and $\tau^i$ respectively. Each agent acts according to their policy $\pi^i(a|\tau_t^i)$ that returns a distribution over actions given action observation history and the joint policy is simply defined as $\pi = (\pi^1, \cdots, \pi^N)$. An important quantity in Dec-POMDPs is the belief function $\mathcal{B}_\pi(\tau_t|\tau_t^i) = P(\tau_t|\tau_t^i, \pi)$ that returns an agent's belief over the distribution of the state given their AOH and the group's joint policy.

## 3.2 Problem Settings

**Ad-Hoc Teamplay** The ad-hoc teamplay setting's goal is to cooperate effectively with unknown teammates, where each teammate can contribute meaningfully to the task [17]. This setting has been used in numerous environments including Overcooked [4], RoboCup 2D [2], and Hanabi [5, 7]. Some works leverage small amounts of data to guide agents towards equilibria or "social conventions" [11, 23]. The ad-hoc teamplay setting includes cooperating with *any* well performing policy. However, in Hanabi it is well known that strong SP policies aren't necessarily well suited for coordination [6]. As a result, for all experiments involving ad-hoc teamplay, we choose policies that are amenable to meaningful coordination.

**Zero-Shot Coordination (ZSC)** Training in SP has been shown to give rise to policies that employ "arbitrary" conventions. One example is converging on one out of combinatorially many, mutually incompatible possible protocols for conveying information through their actions. Informally, this happens because both past observations and future actions are produced by the policy being optimized. This forms a feedback loop in which coincidences that occur when agents explore are reinforced until they form arbitrary conventions. Clearly, this kind of failure points to a fundamental incompatibility of current MARL methods with the coordination setting.

Inspired by this, zero-shot coordination (ZSC) [6] was proposed as a proxy setting towards human-AI coordination. Under ZSC the goal is to maximise the cross-play (XP) score, the average score between independently trained agents under the same high level algorithm. Clearly, this rules out any methods that rely on arbitrary conventions for maximizing rewards, since those lead to incompatible results across independent training runs. ZSC closely mirrors the *independent reasoning process* that humans are able to undertaken when coordination with others without prior agreement.

**Human-AI Coordination** The long standing goal of AI has been to coordinate with humans. Multiple studies have evaluated AI agents in coordinating with humans including in Hanabi [6, 15], Overcooked [4, 18], and Dota 2 [13]. A commonly used proxy setting for human-AI coordination is to evaluate with an agent trained on human data [4, 5, 7]. We use this proxy setting in this work.

## 3.3 Off-Belief Learning

**Intuition.** Off-belief learning (OBL) [7] is an algorithm designed to break the standard SP training feedback loop and avoid arbitrariness. At every time step OBL *reinterprets* the AOH as if it generated by playing with fixed, given policy $\pi_0$, preventing the development of conventions. When $\pi_0$ is a fully random policy past actions themselves are treated as conveying no information at all, since the probability of an action is independent of the state of the environment. Thus, the only information retained from an action is what it explicitly reveals about the environment. In turn, any accidental attempt to encode private information through the actions will be ignored in subsequent time steps and will not be reinforced. Instead, agents have to adopt *grounded play*, acting only on the basis of the grounded information and prioritizing actions that reveal grounded information in turn.

**Implementation.** In practice, OBL trains a sequence of policies starting with $\pi_0$ a uniform random policy and iteratively using the trained policy $\pi_1$ as the new $\pi_0$ until the improvement $\Delta(J) = J(\pi_1) - J(\pi_0)$ diminishes. Keeping track of the exact belief is computationally expensive, thus OBL is implemented by first training a neural network belief model on $\pi_0$ using supervised learning. This model is used to sample *fictitious states* $\tau_t'$ by evaluating it on $\pi_1$'s data, i.e. $\tau_t' \sim \hat{\mathcal{B}}_{\pi_0}(\tau_t|\tau_t^i)$, where

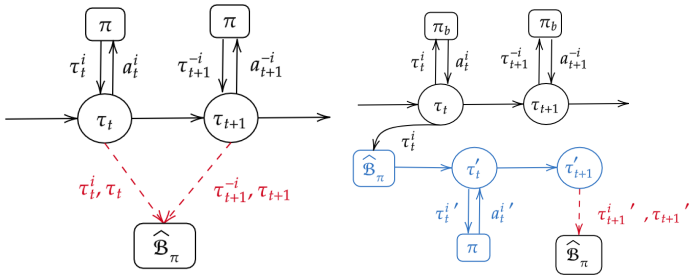


Figure 1: Visualization of belief training schemas by *vanilla OBL* (**left**) and *OT-BL* (**right**). Here superscript $i$ indicates player and $-i$ indicates partner. The red arrow indicates the values used for belief training. The blue colors represent a *fictitious transition*. *Vanilla OBL* solely trains $\hat{\mathcal{B}}_\pi$ on $P_\pi$. *OT-BL* is presented with only adding one timestep, but all timesteps are added in a similar manner. Notice that with OT-BL, we can train any belief model off-team. In this case $P_{\pi_b}(\tau_t^i)$ is generated by an arbitrary joint policy $\pi_b$. Importantly, $\hat{\mathcal{B}}_\pi$ is only trained on fictitious transitions generated by $\pi_\ell$.

$\tau_t \sim P_{\pi_1}(\tau_t^i)$. At every timestep $t$, the policy $\pi_1$ is queried to produce an action on the *fictitious state*. This action is used to perform a *fictitious transition*, which is used to calculate the target value for training OBL (See figure 2 for a visual representation of the fictitious transition).

# 4 The Off-Team Learning Problem

In single agent RL, *off-policy RL* is the task of learning information about a target policy $\pi$, from rollouts generated by a distinct behavior policy $\pi_b$. Similarly, in MARL we define *off-team* (OT) learning to be the task of learning policies or values for target joint policy $\pi$, from rollouts generated by playing with a distinct behavior team $\pi_b$. Despite the similarities in definitions, note that the methods for the former do not transfer to the latter. Indeed, unlike in single-agent settings the AOH is only Markov given fixed teammates. We therefore need to carefully approach off-team learning to prevent $\pi_b$'s conventions from bleeding into $\pi$, the joint policy being learned. Formally, we define the OT-Belief Learning (OT-BL) problem as follows

$$\min_{\hat{\mathcal{B}}} \sum_t \sum_i \mathbb{E}_{\tau_t^i \sim P(\pi_b)} \mathcal{L}(\mathcal{B}_\pi(\cdot \mid \tau_t^i), \hat{\mathcal{B}}_\pi(\cdot \mid \tau_t^i)). \tag{1}$$

In words, the goal is to learn an accurate belief model, $\mathcal{B}_\pi$, for the policy $\pi$ when evaluated on the distribution of AOH induced by the policy $\pi_b$.

Similarly, formally, we define the OT-RL problem as follows

$$\max_{\pi} \sum_t \sum_i \mathbb{E}_{\tau_t^i \sim P(\pi_b)} \mathbb{E}_{\tau_t \sim \mathcal{B}_\pi(\tau_t^i)} \mathbb{E}_{a \sim \pi(\tau)} Q_{\perp(\pi)}(\tau, a) \tag{2}$$

where $\perp(\pi)$ denotes $\pi$ but caveats that it cannot be optimized. Notably, unlike in off-policy reinforcement learning, where it is often desirable to re-weight the data according to the target policy reach probabilities, in off-team learning, the fact that the objective is weighted according to the behavior policy reach probabilities is a core part of the problem setting.

# 5 Off-Team Learning

Off-team learning is a difficult problem because it requires weighting the loss values by the reach probabilities induced by $\pi_b$. However, we can't simply naively train our learned belief model, $\hat{\mathcal{B}}_\pi$, on data generated by $\pi_b$, since this would result in a belief model for $\pi_b$ rather than $\pi$. Similarly, we cannot simply naively train our policy on data generated by $\pi_b$ because such a policy would leverage the correlations between $\pi_b$'s actions and its private information. To address this problem we introduce *off-team belief learning (OT-BL)* and *off-team reinforcement learning (OT-RL)*, which use bootstrapping to perform forward induction on the beliefs and backward induction the values (respectively), and trajectory resampling to ensure that each loss is weighted by the correct value.

Say we have learned a belief model $\hat{\mathcal{B}}_\pi$ up to time $t-1$, and a value function $Q_\pi$ for time $t$ and thereafter. We describe off-team learning's procedure for generating data for time $t$ in three steps.

1. Off-team learning samples trajectories according to the behavior policy $\pi_b$, thereby guaranteeing that the samples $\tau_{t-1}^i$ are weighted by the behavior distribution.

**Algorithm 1** off-team belief learning: fictitious transition

**Require:** belief model $\hat{\mathcal{B}}_\pi$ being trained, policy $\pi$, AOH $\tau_t^i$ from the viewpoint of agent $i$ induced by $\tau_t \sim P_{\pi_b}(\tau_t)$

1: Sample fictitious state $\tau'_t \sim \hat{\mathcal{B}}(\cdot|\tau_t^i)$
2: Sample fictitious action $a'_t \sim \pi(\cdot|\tau_t'^i)$
3: Perform a fictitious transition $\mathcal{T}(\tau'_t, a'_t) = \tau'_{t+1}$
4: Use $(\tau_{t+1}'^{-i}, \tau'_{t+1})$ to train $\hat{\mathcal{B}}_\pi$ via supervised learning.

**Algorithm 2** off-team reinforcement learning training transition

**Require:** belief model $\hat{\mathcal{B}}_\pi$, training policy $\pi$, partner policy $\pi_b$, state $\tau_t \sim P_{(\pi,\pi_b)}(\tau_t)$

1: Observe $\tau_t^i$ from the viewpoint of agent $i$
2: Sample fictitious state $\tau'_t \sim \hat{\mathcal{B}}_\pi(\cdot|\tau_t^i)$
3: Sample real action $a_t \sim \pi(\cdot|\tau_t^i)$, perform fict. transition $\mathcal{T}(\tau'_t, a_t) = \tau'_{t+1}$ and obtain reward $r'_t$
4: Sample fictitious action $a'_{t+1} \sim \pi(\cdot|\tau_{t+1}'^{-i})$, perform fict. transition $\mathcal{T}(\tau'_{t+1}, a'_{t+1}) = \tau'_{t+2}$ and obtain reward $r'_{t+1}$
5: Use $r'_t + \gamma r'_{t+1} + \gamma^2 \max_a Q_\pi(a|\tau'_{t+2})$ as a target for $\pi$ on observation $\tau_t^i$

2. Off-team learning corrects for the fact that $\tau_{t-1}$ was played to $t-1$ by $\pi_b$, rather than $\pi$. It does so by resampling $\tau'_{t-1}$ using the learned belief model, $\hat{\mathcal{B}}_\pi(\tau_{t-1} \mid \tau_{t-1}^i)$, taking advantage of the belief model being accurate up to time $t-1$.
3. We propagate this trajectory forward in time by sampling an action $a \sim \pi(\tau'_{t-1})$ from the target policy, $\pi$, and sampling a trajectory for the next time step $\tau'_t \sim \mathcal{T}(\tau'_{t-1}, a)$ using this sampled action. This $\tau'_t$ is distributed approximately according to $\mathcal{B}_\pi(\tau_t^{i\prime})$ but only generated by OT-BL with the reach probability induced by $\pi_b$. Thus, by training $\hat{\mathcal{B}}_\pi(\tau_t^{i\prime})$ to maximize the likelihood of $\tau'_t$, we are minimizing the loss defined in expression 1. Analogously, by performing a TD-update between $Q(\tau_{t-1}^{i\prime}, a)$ and $R(\tau'_{t-1}, a) + \max_{a'} Q(\tau_t^{i\prime}, a')$, we are maximizing the objective defined in expression 2.

The procedure described above is described in pseudocode in algorithms 1 and 2 and visualizations are provided in Figure 1 and Figure 2.

# 6 Experimental Setup

## 6.1 Hanabi

Hanabi [1] is a challenging benchmark for Dec-POMDP research. It is a cooperative game for 2-5 players played with 8 hint tokens, 3 life tokens, and a deck of 50 cards, each having a rank between 1 and 5 and one of five colors. In the 2-player variant, on which this paper focuses, each player holds 5 cards, backwards, such that they can see their partners' hand but not their own. On their turn, a player can either *play*, *discard* or *hint*. The goal is to play cards in rank order for each color, with each correctly played card providing one point for a maximum score of 25. The team loses a life token each time a player attempts to play an unplayable card. Hinting consumes a hint token and can be used to point all cards of a chosen color or rank in a partner's hand[1]. The team regains a hint token when a player discards a card or completes a stack by successfully playing the 5 of the corresponding color. A player draws a new card after playing or discarding. The game terminates when all 5 stacks are completed, the deck is empty, or all three lives are lost. In the latter case the team loses all the points they have earned.

## 6.2 Training Details

We implement our algorithms based on the open-sourced code for OBL [2] and extend it with ideas from synchronous training [10, 5], training all models simultaneously, thus enabling effective implementations of OT-BL and OT-RL. When synchronously training OBL models, every $n = 50$ steps we save a copy of the model then query for and update all dependencies.

[1] e.g. "Your first and third card are rank 3."

[2] `https://github.com/facebookresearch/off-belief-learning`

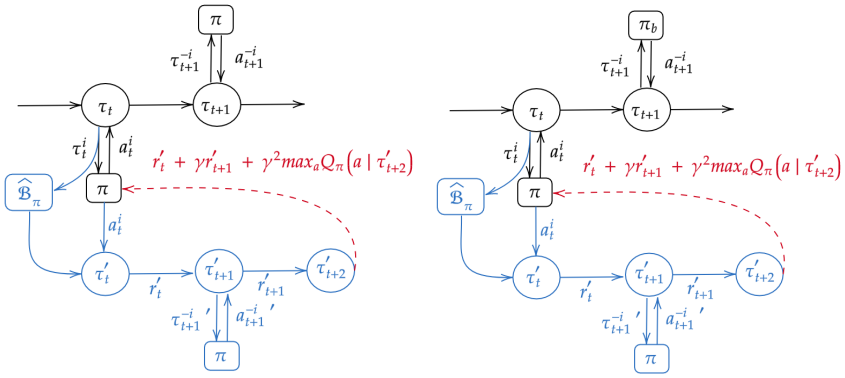


Figure 2: Visualization of policy training schemas for *vanilla OBL* (**left**), and *OT-RL* on $P_{\pi,\pi_b}(\tau_t^i)$ (**right**) for $m = 2$ step update. Here superscript $i$ indicates player and $-i$ partner. The red arrow is the computed target value. The blue colors represent a *fictitious transition*. *Vanilla OBL* computes its target by sampling $\tau'_t \sim \hat{B}_\pi(\cdot|\tau_t^i)$, which assumes $\pi$ has been played to this point, and performs two fictitious steps with $\pi$. We show how OT-RL can be used to train a policy $\pi$ with $\hat{\mathcal{B}}_\pi$ on any distribution $P_{\pi,\pi_b}(\tau_t^i)$, where $\pi_b$ is any policy. *OT-RL* always computes its target value by sampling from $\hat{\mathcal{B}}_\pi$ and using $\pi$ to produce the two fictitious steps.

We follow practices in the original OBL paper [7] to train each policy and belief model. The backbone is R2D2 [8], a distributed recurrent Q-learning method with parallel environment workers and centralized replay buffer and trainer. We briefly describe the loss functions for the Q-network and belief models here. For more details, please refer to Appendix A.1 or the OBL paper [7].

The Q-network is trained to minimize the TD-error between its Q-value and targets from fictitious transition:

$$\mathcal{L}(\boldsymbol{\theta}|\tau) = \frac{1}{2}\sum_{t=1}^{T}\left[G'_t - Q_{\boldsymbol{\theta}}(a_t|\tau_t)\right]^2$$
$$G'_t = r'_t + \gamma r'_{t+1} + \gamma^2 \max_a Q_{\hat{\boldsymbol{\theta}}}(a|\tau'_{t+2})$$

where $\hat{\boldsymbol{\theta}}$ is a slightly outdated target network. We pre-compute the fictitious target $G'_t$ and store them into the replay buffer to avoid branching LSTM rollouts into ficitious transitions at training time.

The belief models are trained to predict the player's hand given $\tau^i$. At a given timestep $t$, the belief model will take in $\tau_t^i$ as input and predict the hand of player $i$ auto-regressively from oldest card to newest. We train all OBL models as recurrent models via supervised learning to optimize:

$$\mathcal{L}(\boldsymbol{h}|\tau_t^i) = -\sum_{k=1}^{n}\log p(h_k|\tau_t^i, h_{1:k-1}),$$

where $h_k$ is the $k$th card in hand and $n$ is the hand size.

### 6.3 Evaluation

We repeat each training method five times with different seeds. For SP and XP, every team is evaluated on 5000 games and then averaged over five seeds. The XP score is the average score of all teams formed by all independently trained agents. The goal of ZSC is to evaluate the consistency of an algorithm across different seeds. To better reflect improvement in ZSC, we introduce the *self-play cross-play gap (SXG)* metric, which is computed as SXG $= 100 \cdot \frac{(SP-XP)}{SP}$. This normalizes increases in XP score relative to the theoretical maximum self-play score. Additionally, SXG acts as a proxy for the covariate shift as it compares the evaluated training distribution (SP) with the test distribution (XP). The higher the SXG the larger the covariate shift.

### 6.4 Experimental Design for OT-BL

We design experiments for OT-BL to emphasize its advantage over the normal belief learning procedure when evaluated off-team. In normal OBL, the belief model is trained on trajectories from $\pi_\ell$ but utilized on trajectories from $\pi_{\ell+1}$ to produce ficitious transitions, making it vulnerable to

covariate shift. With OT-BL, half of environments used to train $\hat{\mathcal{B}}_{\pi_\ell}$ follow typical belief model training. The other half uses OT-BL to train $\hat{\mathcal{B}}_{\pi_\ell}$ off-team on $P_{\pi_{\ell+1}}(\tau_t^i)$.

Ideally, we want to evaluate our algorithm on environments with a *larger covariate shift*. Therefore, we make a few modifications to the Hanabi environment to exacerbate the covariate shift. *First* we reduce the total number of life tokens from 3 to 2, making the game harder so that the average score and average trajectory length of different OBL levels differ more significantly. It is particularly challenging for the belief model only trained on shorter trajectories to generalize to longer unseen trajectories. *Second*, we consider *sampling valid hands*. Prior works including OBL utilize a checker provided in the Hanabi challenge [1] to check if a hand satisfies the known public knowledge. If a hand is invalid, the environment can fail to produce key input features. We further modify the 2-life Hanabi environment to remove the checker and allow invalid samples to corrupt the environment as is, which only affects the fictitious transition. Specifically, if a sampled card contradicts the public knowledge of that card, the observation encoder will just encode the card knowledge for that card to be all zero instead of rejecting the sampled hand. We call this *invalid card penalty*.

Finally, we measure the belief improvements through the percentage of valid samples for fictitious transitions. For this we focus on the level 1 policy training process in the OBL hierarchy, where there is the largest covariate shift. To train the level 1 policy, we need to unroll a belief model $\hat{\mathcal{B}}_{\pi_0}$ on trajectories generated by the learning policy $\pi_1$. At convergence, $\pi_1$ can achieve 20 points. However, the belief model is trained on $\pi_0$, a uniform random policy with average score close to zero. We compute the percentage of timesteps where $\hat{\mathcal{B}}_{\pi_0}$ is able to produce a valid sample when training $\pi_1$ as a proxy for the amount the belief model has deteriorated due to covariate shift.

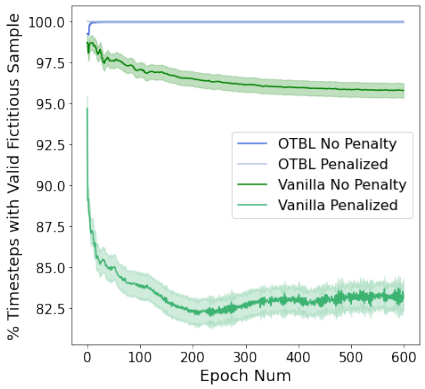


Figure 3: Comparison between belief models trained with vanilla OBL (greens) and with OT-BL (blues) in 2-life Hanabi with and without penalty for invalid samples. Experiments with and without penalty sample $p = 1, 10$ hands respectively. We plot the % of timesteps with a valid fictitious sample (standard error is shaded) when training OBL level 1. Normally trained belief models struggle to gather a valid sample; it steadily declines in unpenalized cases and drops to only $\sim 83\%$ of timesteps in the penalty cases. OT-BL is able to consistently gather a valid sample during training ($> 99.5\%$ of the time), regardless of penalty. OT-BL mitigates the covariate shift experienced during OBL training.

### 6.5 Experimental Design for OT-RL

**OT-RL for ZSC** We design experiments to evaluate OT-RL's performance in ZSC. For normal OBL, the policy, $\pi_\ell$, is trained on trajectories from $P_{\pi_\ell}(\tau_t^i)$ but evaluated on $P_{\pi_\ell,\pi_b}(\tau_t^i)$. This means $\pi_\ell$ experiences test time covariate shift. To evaluate OT-RL in ZSC, when training a given policy $\pi_\ell$, we train on real trajectories from from $P_{\pi_{\ell-1},\pi_\ell}(\tau_t^i), P_{\pi_\ell}(\tau_t^i), P_{\pi_\ell,\pi_{\ell+1}}(\tau_t^i)$ and use OT-RL to train off-team. By training on broader distributions, we increase the support of our policies. Since $\pi_{\ell-1}$ and $\pi_{\ell+1}$ are OBL policies, they are also reasonable and do well in Hanabi. We also test a variant where we use OT-RL and robustify the belief by also training the belief model on $P_{\pi_{\ell-1},\pi_\ell}(\tau_t^i), P_{\pi_\ell}(\tau_t^i), P_{\pi_\ell,\pi_{\ell+1}}(\tau_t^i)$. We refer to this variant as OT-RL + robust belief.

**OT-RL for ad-hoc teamplay and human-AI coordination** This set of experiments aims to demonstrate the ability for OT-RL to improve test time performance with only a small amount of data from test time agents. These experiments address the ad-hoc teamplay and human-AI coordination settings where there are *large covariate shifts*. We utilize *rank bot* for the ad-hoc teamplay setting and *clone bot*, a bot trained via supervised learning on human data, as a proxy for human-AI coordination. See appendix A.3 and A.4 for more details on training *rank* and *clone bot* respectively. Each setting we assume access to only a small number, $o = 3200$, of full trajectories generated by the target agents. We retrain a new policy, $\pi$, with a belief model trained on level 3 of sequentially trained OBL with the typical OBL training procedure. In this case half of the games in our distributed RL environment are the real target trajectories and we train $\pi$ in an off-policy manner with OT-RL on these trajectories. By training on these target trajectories, we can partially mitigate the covariate shift experienced when $\pi$ is evaluated. We compare our results with a vanilla sequentially trained OBL level 4.

| Level | without penalty | | with penalty | |
|---|---|---|---|---|
| | OBL (XP & SXG) | OT-BL (XP & SXG) | OBL (XP & SXG) | OT-BL (XP & SXG) |
| 1 | $20.17 \pm 0.15$ (0.84) | $21.05 \pm 0.02$ (0.05) | $16.71 \pm 0.06$ (1.56) | $21.15 \pm 0.02$ (0.09) |
| 2 | $22.92 \pm 0.21$ (2.18) | $23.41 \pm 0.02$ (0.56) | $21.24 \pm 0.11$ (4.28) | $23.48 \pm 0.02$ (0.38) |
| 3 | $23.05 \pm 0.30$ (3.27) | $23.76 \pm 0.02$ (0.46) | $22.53 \pm 0.23$ (4.43) | $23.83 \pm 0.01$ (0.33) |

Table 1: XP and SXG (in parentheses) for OBL and OT-BL in 2-life Hanabi with and without *invalid card penalty*. OT-BL always achieves higher XP scores. OT-BL also has a very small SXG $< 1\%$ which is $4 - 18\times$ lower than corresponding vanilla OBL. The $\pm$ represents the team's standard error.

| Level | OBL (XP & SXG) | OT-RL (XP & SXG) | OT-RL + Robust Belief (XP & SXG) |
|---|---|---|---|
| 1 | $21.50 \pm 0.05$ (0.42) | $21.31 \pm 0.01$ (0.19) | $21.61 \pm 0.03$ (0.32) |
| 2 | $23.75 \pm 0.02$ (0.46) | $23.72 \pm 0.01$ (0.17) | $23.78 \pm 0.01$ (0.21) |
| 3 | $24.10 \pm 0.02$ (0.41) | $24.10 \pm 0.01$ (0.17) | $24.22 \pm 0.01$ (0.17) |
| 4 | $24.03 \pm 0.10$ (0.91) | $24.22 \pm 0.01$ (0.21) | $24.29 \pm 0.01$ (0.16) |
| 5 | $23.99 \pm 0.11$ (1.28) | $24.24 \pm 0.01$ (0.12) | $24.30 \pm 0.01$ (0.16) |

Table 2: XP and SXG (in parentheses) for vanilla OBL, OT-RL, and OT-RL with belief models trained on broader distributions in 3-life Hanabi. Vanilla OBL overfits leading to a drop in XP scores. OT-RL prevents overfitting and continues to improve its XP score. OT-RL almost always has a higher XP scores than vanilla OBL. OT-RL has a very small SXG $< 1\%$ and this gap is up to 10 times lower than that of vanilla OBL. By training our belief models on broader distributions, we further increase our XP score while maintaing a low SXG. The $\pm$ represents the team's standard error.

# 7 Results and Discussion

## 7.1 OT-BL in 2-life Hanabi

The the XP and SXG results of OT-BL on 2-life Hanabi is shown in Table 1. Without the *invalid card penalty*, both OBL with vanilla belief learning and OT-BL are able to achieve reasonably high XP scores in 2-life Hanabi. However, OT-BL consistently achieves higher XP scores in 2-life Hanabi across all levels. At level 3, OT-BL is able to achieve a final XP score of $23.76 \pm 0.02$. Lastly, the SXG for OT-BL is always lower than that of vanilla OBL (8-16x lower).

In Table 1, we also show the performance of OT-BL on 2-life Hanabi *with invalid card penalty* on the right. Similarly, OT-BL consistently achieves higher XP scores across all levels. OT-BL is able to achieve a small SXG (always $< 1\%$) and this gap is up to $18\times$ lower than vanilla OBL. OT-BL achieves similar results between the penalized and unpenalized environments since the belief model is more on distribution, thus it almost always samples valid samples ($> 99.5\%$ of the time). However, OBL with vanilla belief learning suffers significantly from the penalty, especially for level 1 where the covariate shift from a uniform random policy to a learned OBL policy is the largest. This indicates that the test time covariate shift is lower for OT-BL when compared to Vanilla OBL. The reduced covariate shift improves training and leads to better test time results.

We plot the percentage of timesteps $\hat{\mathcal{B}}_{\pi_0}$ is able to produce a valid sample when training $\pi_1$ in Figure 3, to illustrate the deteriorating belief model in normal OBL and show the benefit of OT-BL. *Without penalty*, vanilla OBL steadily decreases in the percentage of timesteps producing valid samples. *With penalty*, vanilla OBL is unable to produce a high percentage of valid samples and can never recover. This high percentage of invalid samples contributes to the steep score drop for OBL level 1 in the invalid state experiments. On the other hand, OT-BL mitigates the belief model's training covariate shift and consistently samples valid states (over $99.5\%$ of the time). This contributes to the high scores for OT-BL and explains why OT-BL maintains high scores in penalized Hanabi.

## 7.2 OT-RL

**ZSC Results** We present the XP and SXG results of OT-RL in table 2. All variants, normal OBL, OT-RL , and OT-RL + robust belief all achieve high XP scores. However, at level 4 vanilla OBL starts to overfit, whereas OT-RL and OT-RL + robust belief continue to improve. This results in OT-RL achieving a XP score of $24.24 \pm 0.01$ and OT-RL + robust belief achieves a XP score of $24.30 \pm 0.01$. As Vanilla OBL overfits, the SXG increases, indicating an increase in the test time covariate shift. On the other hand, both OT-RL and OT-RL + robust belief always have a SXG$< 1\%$ and at level 5 the SXG is almost $0$. By level 5, the SXG of OT-RL and OT-RL + robust belief are at least 8 times

| Training Type | XP & SXG | w/ rank bot | w/ clone bot | w/ Vanilla OBL |
|---|---|---|---|---|
| Vanilla OBL | $23.66 \pm 0.06$ (1.42) | $14.47 \pm 0.67$ | $16.19 \pm 0.21$ | - |
| OT-RL w/ Rank Bot | $23.59 \pm 0.08$ (1.38) | $15.66 \pm 0.56$ | $16.36 \pm 0.23$ | $23.67 \pm 0.04$ |
| OT-RL w/ Clone Bot | $23.54 \pm 0.06$ (1.13) | $14.33 \pm 0.63$ | $16.64 \pm 0.20$ | $23.64 \pm 0.04$ |

Table 3: Evaluating OT-RL in ad-hoc teamplay (w/ rank bot), proxy human-AI coordination, and w/ vanilla OBL. By training on a small number of target trajectories, we can improve ad-hoc teamplay scores with rank bot and proxy human-AI coordination scores. OT-RL also doesn't significantly hurt XP scores, the SXG, or the unseen target policies in ad-hoc teamplay or human-AI coordination. Additionally, by training with OT-RL, we still achieve very similar scores with vanilla OBL, indicating we are only broadening the policies support. The $\pm$ here represents the team's standard error.

lower than that of vanilla OBL. Overall, by training on broader distributions, we reduce the test time covariate shift, achieve near optimal ZSC performance, and improve OBL in off-team evaluation.

**Ad-Hoc Teamplay and Proxy Human-AI Coordination Results** We present the results appling OT-RL Ad-hoc teamplay and Human-AI coordination in table 3. With OT-RL and only 3200 target policy trajectories, we can increase *rank bot* to $14.47 \pm 0.67$ to $15.66 \pm 0.56$ and increase *clone bot* scores from $16.19 \pm 0.21$ to $16.64 \pm 0.20$ without heavily impacting XP scores or the SXG. Additionally, vanilla OBL in XP achieves near identical scores to OT-RL trained on additional trajectories is paired with vanilla OBL. Therefore, the additional trajectories merely broaden the training policies support and don't negatively impact policy training. Overall, by training on target distributions off-policy and off-team with OT-RL we are able to reduce the test time covariate shift and increase ad-hoc teamplay and proxy human-AI coordination scores.

# 8 Limitations

Although OT-BL enables $\mathcal{B}_\pi$ to train off-team on any distribution $P_{\pi_b}(\tau_t^i)$, it assumes $\pi$ is also on distribution when we produce fictitious moves with OT-BL. As a result, if $\pi$ is too off distribution OT-BL might not fully mitigate the covariate shift. Additionally, although our work provides a positive step to mitigate the large covariate shift that can occur in ad-hoc teamplay and human-AI coordination with little target policy data, there is still exists covariate shift.

# 9 Conclusion

We present *off-team belief learning*, a method that allows us to train any belief *off-team*. Belief models are very ubiquitous in POMDPs and Dec-POMDPs. However, they can experience covariate shift when deployed on unseen distributions. We demonstrate that standard belief model training can suffer from covariate shift, but OT-BL reduces it. Overall, by training OBL with OT-BL, we are able to achieve strong results in ZSC in 2-life Hanabi and minimize the gap between SP and XP.

We also present *off-team reinforcement learning*, a method that allows us to train policies *off-team*. The goal of off-team training is to train with various teams with the goal of improving performance with unseen teams. With OT-RL, we can train policies on broader distributions, increase their support, and improve test time performance. By training OBL with OT-RL we were able to achieve near optimal-ZSC performance and improve ad-hoc teamplay and proxy human-AI coordination.

For follow-up work, a promising direction involves combining multiple ways of reducing test time covariate shift. For instance, Strouse et al. [19] trained on a population of various checkpoints, Cui et al. [5] synchronously trained cognitive hierarchies, and Lupu et al. [12] learned a common best response to diverse agents; all works which could complement our methods by further broadening the training distribution. Additional research is also needed to produce agents that can accommodate human behaviour such as "finessing" or "bluffing" in Hanabi, which relies on theory of mind and can also induce large covariate shifts.

# 10 Broader Impact

We have demonstrated both OT-RL and OT-BL can reduce covariate shift. OT-BL reduces covariate shift in the difficult 2-life Hanabi case in ZSC compared to vanilla OBL. OT-RL reduces covariate shift in ad-hoc teamplay, ZSC, and proxy human-AI coordination. Overall, OT-RL and OT-BL are general methods for belief and policy robustification. As a result, we have provided positive steps towards numerous difficult coordination settings. Although, no technology is immune to misuse, we are working in purely cooperative settings.

# 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*, 280:103216, 2020. ISSN 0004-3702.

[2] S. Barrett, A. Rosenfeld, S. Kraus, and P. Stone. Making friends on the fly: Cooperating with new teammates. *Artificial Intelligence*, 242:132–171, 2017. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2016.10.005. URL `https://www.sciencedirect.com/science/article/pii/S0004370216301266`.

[3] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of markov decision processes. *Math. Oper. Res.*, 27(4):819–840, Nov. 2002. ISSN 0364-765X. doi: 10.1287/moor.27.4.819.297. URL `https://doi.org/10.1287/moor.27.4.819.297`.

[4] M. Carroll, R. Shah, M. K. Ho, T. Griffiths, S. Seshia, P. Abbeel, and A. Dragan. On the utility of learning about humans for human-ai coordination. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/f5b1b89d98b7286673128a5fb112cb9a-Paper.pdf`.

[5] B. Cui, H. Hu, L. Pineda, and J. Foerster. K-level reasoning for (human-ai) zero-shot coordination in hanabi. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 4190–4203. Curran Associates, Inc., 2021.

[6] 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.

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

[8] S. Kapturowski, G. Ostrovski, J. Quan, R. Munos, and W. Dabney. Recurrent experience replay in distributed reinforcement learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[9] D. Kingma and J. Ba. Adam: A method for stochastic optimization, 2014.

[10] M. Lanctot, V. Zambaldi, A. Gruslys, A. Lazaridou, K. Tuyls, J. Perolat, D. Silver, and T. Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 4190–4203. Curran Associates, Inc., 2017.

[11] A. Lerer and A. Peysakhovich. Learning existing social conventions via observationally augmented self-play. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '19, page 107–114, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450363242.

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

[13] OpenAI, :, C. Berner, G. Brockman, B. Chan, V. Cheung, P. Dębiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, R. Józefowicz, S. Gray, C. Olsson, J. Pachocki, M. Petrov, H. P. d. O. Pinto, J. Raiman, T. Salimans, J. Schlatter, J. Schneider, S. Sidor, I. Sutskever, J. Tang, F. Wolski, and S. Zhang. Dota 2 with large scale deep reinforcement learning, 2019. URL `https://arxiv.org/abs/1912.06680`.

[14] T. Schaul, J. Quan, I. Antonoglou, and D. Silver. Prioritized experience replay. In Y. Bengio and Y. LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.05952.

[15] H. C. Siu, J. Peña, E. Chen, Y. Zhou, V. Lopez, K. Palko, K. Chang, and R. Allen. Evaluation of human-ai teams for learned and rule-based agents in hanabi. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 16183–16195. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf.

[16] 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.

[17] P. Stone, G. A. Kaminka, S. Kraus, and J. S. Rosenschein. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, page 1504–1509. AAAI Press, 2010.

[18] D. Strouse, K. McKee, M. Botvinick, E. Hughes, and R. Everett. Collaborating with humans without human data. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 14502–14515. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/797134c3e42371bb4979a462eb2f042a-Paper.pdf.

[19] D. Strouse, K. McKee, M. Botvinick, E. Hughes, and R. Everett. Collaborating with humans without human data. *Advances in Neural Information Processing Systems*, 34, 2021.

[20] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014. URL http://arxiv.org/abs/1409.3215.

[21] G. Tesauro. Td-gammon, a self-teaching backgammon program, achieves master-level play. *Neural computation*, 6(2):215–219, 1994.

[22] J. Treutlein, M. Dennis, C. Oesterheld, and J. Foerster. A new formalism, method and open issues for zero-shot coordination. *arXiv preprint arXiv:2106.06613*, 2021.

[23] M. Tucker, Y. Zhou, and J. Shah. Adversarially guided self-play for adopting social conventions. *ArXiv*, abs/2001.05994, 2020.

[24] H. van Hasselt, A. Guez, and D. Silver. Deep reinforcement learning with double q-learning. In D. Schuurmans and M. P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2094–2100. AAAI Press, 2016. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12389.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.

[26] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas. Dueling network architectures for deep reinforcement learning. In M. Balcan and K. Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1995–2003. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/wangf16.html.

# 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?
- Did you include the license to the code and datasets? [No]
- 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? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
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)? [Yes]
   (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? [Yes]
   (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]