Differentiable Belief-based Opponent Shaping

Aarav G. Sane **Karthik Sivachandran** **Rohan R. Paleja**
Department of Computer Science
Purdue University
{sane0, ksivacha, rpaleja}@purdue.edu

Abstract

Human coordination often relies on the ability to influence the beliefs of others through strategic action. In multi-agent reinforcement learning, opponent shaping attempts to replicate this influence, though existing methods typically operate within an opponent's parameter, policy, or value space. Meanwhile, belief-manipulation techniques in hidden-role games often rely on hard-coded objectives, such as deception or belief saturation. We propose Differentiable Belief-based Opponent Shaping (D-BOS), a first-order method that treats each observer's belief as the shaped opponent state and differentiates through k-step softmax-Bayes belief dynamics. Rather than explicitly rewarding deceptive or cooperative behavior, our method treats the belief state as the target for shaping. This allows the optimal strategy to emerge naturally from the environment's reward structure. This belief-space formulation provides an opponent-shaping signal by differentiating through opponent belief updates, and naturally extends to multiple observers by aggregating gradients over their individual inferred belief trajectories. Empirically, D-BOS shows promise in belief-space shaping and performs moderately better than certain baselines in specific environment settings.

1 Introduction

Humans routinely reason about what others believe, and use that reasoning to coordinate, persuade, conceal, or reveal information. This ability is central to Theory of Mind [1, 2] and has become an important motivation for computational models of human-agent collaboration and human-robot teaming [3, 4]. In collaborative settings, from shared autonomy to ad hoc teaming with heterogeneous strategies [5, 6], the useful partner is not merely the one that chooses good actions in isolation, but the one that understands how its actions change what others think [7, 8]. We study this problem in hidden-role partially observable Markov games, such as Avalon [9], Rescue-the-General [10], and the Multi-Agent Coin Game [11], which are controlled multi-agent testbeds where roles, intentions, and goals are latent, and where an agent's behavior changes the posterior beliefs of allies and adversaries.

Current multi-agent reinforcement learning methods struggle to capture this interaction. Existing opponent-shaping algorithms [12, 13, 14, 15, 16] anticipate an opponent's learning update to optimize returns, but operate in high-dimensional spaces, require strong assumptions, and are often formulated for two-player settings. Conversely, belief-manipulation methods like BBM [10] model hidden beliefs but rely on myopic, prescribed objectives like fixed deception. These limitations motivate a natural synthesis: an agent should shape opponent beliefs over multiple steps driven entirely by task reward. Subordinating belief manipulation to task reward ensures it acts as an instrumental mechanism for optimal play rather than a rigid behavioral prior.

We propose **Differentiable Belief-based Opponent Shaping (D-BOS)**, a first-order method for opponent shaping in belief space. D-BOS treats each observer's (i.e., any other agent in the environment) posterior over hidden roles as the target state to be shaped. It unrolls a differentiable softmax-Bayes update for k steps and backpropagates through the resulting belief trajectory, thereby

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

explicitly optimizing the long-term task value of the induced future belief states. The direction of belief shaping is therefore determined through the task rather than prescribed: in an adversarial setting, D-BOS can obscure the agent's role, while in a cooperative setting, it can make the correct role easier to infer. We evaluate D-BOS in hidden-role games including a mixed-motive setting, where an agent may need to conceal its identity from adversaries while revealing it to allies.

Our contributions are:

- **Differentiable belief dynamics for opponent shaping**: D-BOS backpropagates through k-step softmax-Bayes belief updates, using the explicit softmax Jacobian to produce temporally extended belief-shaping gradients.
- **A LOLA-style view in belief space**: We show that, under a Bayesian observer model, shaping beliefs is equivalent to shaping the opponent's induced policy, yielding the same meta-gradient structure as LOLA but with a low-dimensional belief state.
- **Error bounds for second-order Theory of Mind**: We bound how approximation error in second-order observations propagates through the k-step belief chain and into the shaping gradient, exposing the tradeoff between longer lookahead and belief-model accuracy.
- **Hidden-role experiments**: We evaluate D-BOS against multiple baselines in Rescue-the-General, Avalon, and multi-agent CoinGame to study whether belief-space shaping improves return, belief trajectories, and stability.

2 Related Work

We situate our method at the intersection of three primary lines of research: algorithms for active opponent shaping, computational models of Theory of Mind for belief manipulation, and the embedding of differentiable belief updates within neural architectures.

Opponent Shaping Opponent shaping studies how one learner can improve its own long-term outcome by accounting for how its actions change other learners. LOLA [12] introduced this idea by differentiating through an opponent's anticipated gradient step, which yields a meta-gradient involving mixed second derivatives and requires a differentiable model of the opponent's learning update. Subsequent work refined this basic formulation: SOS [17] modifies the shaping term to improve stability in differentiable games, COLA [15] addresses consistency when both agents use opponent-aware updates, and POLA [14] uses a proximal formulation to reduce sensitivity to policy parameterization. M-FOS [18] instead treats opponent shaping as a meta-learning problem, avoiding a specific differentiable opponent optimizer. Other work reduces or changes the object being differentiated through. LOQA [13] assumes opponents act through softmax Q-values rather than through an explicit policy-gradient learner. Advantage Alignment [16] shows that LOLA- and LOQA-style shaping terms can be expressed through advantage products, yielding a first-order algorithm. D-BOS is closest in spirit to this line of work: it also seeks a first-order shaping signal. Rather than shaping the advantage function, D-BOS shapes an explicit posterior belief state with a known softmax-Bayes update. This preserves the opponent-shaping viewpoint while moving the differentiable update map into the hidden-role belief dynamics.

Belief Manipulation and Theory of Mind Belief manipulation has been studied most directly in hidden-role and deception settings. Bayesian Belief Manipulation (BBM) [10] is our closest predecessor. In this framework, the agent estimates what an observer believes about its role and computes a Bayes factor to quantify the information revealed by a specific action, as shown in Equation (1).

$$\rho = \frac{\pi_{z^*}(a|h_{i,j,i})}{\sum_{z\in\mathcal{Z}} \pi_z(a|h_{i,j,i}) P_{i,j}(A_i^z)} \tag{1}$$

Here, ρ is the Bayes factor, π_{z^*} is the likelihood under the shaping agent's true role z^*, and $P_{i,j}$ represents the observer's current belief. $h_{i,j,i}$ represents the second-order Theory of Mind estimate of what agent A_i (the shaping agent) thinks agent A_j (the observer) thinks A_i observes. This value is then transformed into an intrinsic reward signal, as expressed by $r_{int} = -\log(\rho)$, incentivizing deception. While BBM provides a practical way to train deceptive agents, its shaping signal is fundamentally *myopic*, as it is computed independently at each timestep and relies on a prescribed direction (deception) rather than the environment's task reward.

More broadly, Theory-of-Mind models have been used to infer agents' latent goals, beliefs, and future behavior. ToMnet [19] learns to model other agents from behavior, and Oguntola [20] formalizes

recursive Theory of Mind across several multi-agent testbeds. Similarly, Alon et al. (2023) [21] study how recursive reasoning can lead to deception in communication games. These works operate primarily as passive observers or heuristic frameworks. They do not yield a differentiable shaping gradient that can be used for end-to-end reinforcement learning, which precludes them from serving as active opponent-shaping baselines. Finally, while Model-Based Opponent Modeling (MBOM) [22] uses environment models to adapt to reasoning learners, it is distinct from D-BOS as it focuses on adaptation based on an inferred policy of the opponent. Furthermore, MBOM requires the training of an explicit environment transition model to simulate recursive imagination, while D-BOS is environment-model-free, requiring only the differentiable map of the observer's belief update.

Differentiable Belief Updates Differentiable filtering methods show that Bayesian updates can be embedded inside neural computation. QMDP-net [23] encodes Bayesian filtering and QMDP planning as differentiable neural network layers for partially observable control. IPOMDP-net [24] extends this idea to multi-agent settings with interactive belief updates. These methods use differentiable beliefs to help an agent plan under its own uncertainty. D-BOS uses the same computational affordance for a different purpose: differentiating through the belief update of an *observer* to shape what that observer will believe in the future and how the agent's actions affect these belief dynamics. This distinction is important in hidden-role games, where the belief state being optimized is not only the agent's internal uncertainty, but another agent's posterior over the shaper's latent role.

3 Preliminaries

We study n-player *partially observable hidden-role games*, a special case of stochastic games [25] and partially observable stochastic games [26]. We formalize each game as a tuple $\langle \mathcal{N}, \mathcal{Z}, \mathcal{S}, \{\mathcal{O}^i\}, \mathcal{A}, T, \{r^i\}, \Omega, \gamma \rangle$, where $\mathcal{N} = \{1, \ldots, n\}$ is the set of agents; $\mathcal{Z} = \{0, \ldots, |\mathcal{Z}|-1\}$ is a finite set of *hidden role hypotheses* (e.g. agent identities or communication slots); $\mathcal{S}$ is the shared world state; $\mathcal{O}^i$ is agent i's private observation space; $\mathcal{A}$ is the joint action space; $T : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition kernel; $r^i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is agent i's reward; $\Omega : \mathcal{S} \to \prod_i \Delta(\mathcal{O}^i)$ is the observation function; and $\gamma \in (0, 1)$ is the discount factor. At the start of each episode, a latent role $z \in \mathcal{Z}$ is sampled and held fixed; agents hold knowledge of their own assigned role. All agents share a role-conditioned policy network $\pi_\theta(a \mid o, z)$, which outputs action probabilities given a local observation and a role hypothesis. To clarify the interaction dynamics within our framework, we distinguish between two primary entities:

- Agent: We refer to the entity that shapes and optimizes the belief-based meta-gradient as the agent.
- Observers: All other participants in the environment whose internal beliefs are the target of the shaping process are referred to as observers.

In any given environment, an observer maintains a belief distribution over the set of possible role assignments $z \in \mathcal{Z}$. While the specific composition and size of $\mathcal{Z}$ vary according to the rules of the environment, we treat $\mathcal{Z}$ as a generic discrete set for our mathematical formulation.

3.1 Learning with Opponent-Learning Awareness (LOLA)

LOLA [12] studies games in a two-player setting, where an agent (agent 1) accounts for how its own policy parameters θ_1 influence the anticipated update of the observer (agent 2), as shown in Eq. (2):

$$\Delta\theta_1 = \nabla_{\theta_1} V_1(\theta_1, \theta_2 + \Delta\theta_2) \tag{2}$$

In Equation (2), V_1 is the value function of the agent and $\Delta\theta_2$ is the observer's learning step. LOLA optimizes through this parameter update, which typically requires second-order derivatives and knowledge of the observer's learning dynamics.

3.2 BBM and Second-Order Belief Updates

BBM [10] adopts a second-order Theory of Mind (ToM) framework where the agent estimates an observer's posterior over the agent's hidden role. Because the agent cannot directly access an observer j's private view of the agent i, it must estimate what the observer thinks the agent can see. This second-order observation estimate is denoted as $\hat{o}_{i,j,i}$. The agent's approximation of the observer's belief update is defined in Equation (3).

$$\hat{b}^{(j),z}_{t+1} = \frac{\pi_\theta(a^i_t \mid \hat{o}_{i,j,i}, z)\hat{b}^{(j),z}_t}{\sum_{z' \in Z} \pi_\theta(a^i_t \mid \hat{o}_{i,j,i}, z')\hat{b}^{(j),z'}_t} \tag{3}$$

In Equation (3), $\hat{b}_{t+1}^{(j),z}$ is the agent's estimate of the probability that observer j assigns to agent i's role being z, and $\pi_\theta(a_t \mid \hat{o}_{i,j,i}, z)$ is the role-conditioned action likelihood for role z.

3.3 Observation Proxies

The implementation of $\hat{o}_{i,j,i}$ in Equation (3) requires a proxy for the observer's perspective. We utilize two distinct modes depending on the social complexity of the environment:

- Canonical Mode: In environments where observers share a fundamentally similar perspective, the agent uses its own observation as a robust heuristic stand-in. This reduces approximation error from additional learned modules and serves as our primary baseline.
- Estimated Mode: In mixed-motive settings, an agent must simultaneously deceive adversaries and coordinate with allies, requiring distinct belief trajectories for each observer. To achieve true multi-observer shaping, we deploy a learned backward-prediction module to explicitly infer $\hat{o}_{i,j,i}$ (what observer j believes agent i can see).

Because the Estimated Mode is uniquely suited for navigating conflicting observer perspectives, we focus its evaluation on our mixed-motive testbed, Avalon, while relying on the Canonical proxy alongside public state information for the other environments.

4 Method

4.1 Differentiable Belief Shaping of Opponents

The Bayesian belief update (Eq. 3) involves a ratio: a likelihood multiplied by a prior, divided by a normalizing constant. This is functionally equivalent to the softmax operation applied in log-space. To make the dependence on θ explicit and differentiable, we first define a log-likelihood vector $\boldsymbol{\ell}_t \in \mathbb{R}^{|\mathcal{Z}|}$, where each element $\ell_t^z = \log \pi_\theta(a_t \mid o_t, z)$. We then rewrite the belief update as expressed in Equation (4).

$$\mathbf{b}_{t+1} = \text{softmax}(\boldsymbol{\ell}_t + \log \mathbf{b}_t) \tag{4}$$

In Equation (4), the softmax operates over the latent role dimension $|\mathcal{Z}|$, ensuring the belief vector remains a valid probability distribution. The Jacobian of this softmax operation provides a clean gradient flow through each update step, as defined as $\frac{\partial b_{t+1}^{z'}}{\partial \ell_t^z} = b_{t+1}^{z'}\big(\mathbb{1}[z' = z] - b_{t+1}^{z}\big)$. Here $\mathbb{1}$ is a function evaluating to 1 when the target role z' matches the hypothesis role z, and 0 otherwise.

Given this differentiable update rule, we train a ***BeliefCritic***, which acts as a learned value function of the agent's local state and the joint estimated belief state of all m observers in the environment. We define the multi-observer belief matrix as $\hat{\mathbf{B}}_t \in (\Delta^{|\mathcal{Z}|})^m$, where each row corresponds to a specific observer's tracked posterior over the agent's role. This function is defined in Equation (5).

$$V_\phi : \mathcal{O}^i \times (\Delta^{|\mathcal{Z}|})^m \to \mathbb{R}, \qquad V_\phi(o_t^{(i)}, \hat{\mathbf{B}}_t) \approx \mathbb{E}\left[\sum_{s \geq t} \gamma^{s-t} r_s^{\text{ext}} \,\middle|\, o_t^{(i)}, \hat{\mathbf{B}}_t, \pi_\theta\right] \tag{5}$$

In Equation (5), V_ϕ takes the full observer belief matrix $\hat{\mathbf{B}}_t$ as a direct input, learning the task-value of the environment's overall belief configuration for the agent under the current reward structure. By formalizing the domain over $(\Delta^{|\mathcal{Z}|})^m$, the critic naturally scales to environments with multiple distinct observers (such as the Estimated Mode), while supporting scenarios where observers share identical proxy trajectories (such as the Canonical Mode) without altering the underlying architecture. V_ϕ is trained alongside π_θ via standard temporal difference (TD) regression against bootstrapped returns. We formalize the k-step D-BOS loss as $\mathcal{L}_{\text{BOS}}^k = -V_\phi\left(o_{t+k}^{(i)}, \hat{\mathbf{B}}_{t+k}\right)$ Here, the multi-observer belief matrix $\hat{\mathbf{B}}_{t+k}$ is obtained by chaining k softmax-Bayes updates for all modeled observers. Minimizing $\mathcal{L}_{\text{BOS}}^k$ drives the policy parameters θ toward actions whose induced k-step joint belief trajectories land in high-value regions. The critic parameters ϕ are held fixed when computing the D-BOS gradient; the gradient flows exclusively through $\hat{\mathbf{B}}_{t+k}$'s dependence on θ.

Training Algorithm Algorithmically, D-BOS is implemented as PPO with an additional belief-shaping correction computed from the on-policy rollout. Algorithm 1 gives the implementation-independent training loop. In practice, we instantiate the correction either by directly differentiating

the k-step belief critic objective or by using an equivalent likelihood-coefficient surrogate that detaches the endpoint coefficients for stability; Appendix A.5 gives the exact implementation details.

Multiple observers and observation modes. When several agents observe the shaper, D-BOS can theoretically maintain a separate belief trajectory $\hat{\mathbf{b}}^{(j)}_{t:t+k}$ for each observer A_j, aggregating the shaping loss across observers. However, the capacity to uniquely shape multiple observers depends on the chosen observation proxy (Section 3.3). Under the *canonical mode*, the agent uses its own observation $o_t^{(i)}$ for all observers, effectively collapsing the update into a single, generalized hypothetical observer. True multi-observer shaping, where distinct belief trajectories are modeled, requires the *estimated mode*, which provides observer-specific proxies $\hat{o}_{i,j,i}$.

4.2 Gradient Correction (LOLA-Style)

Rather than adding $\mathcal{L}^k_{\text{BOS}}$ directly to the PPO surrogate loss, D-BOS follows the LOLA paradigm and injects the shaping gradient as a fixed correction. The total shaping gradient is computed *once* from the on-policy rollout trajectory π_{θ_0} prior to any PPO minibatch updates, as shown in Equation (6):

$$g_{\text{BOS}} = \nabla_\theta \frac{\lambda}{T-k} \sum_t \mathcal{L}^k_{\text{BOS}}(t) \Big|_{\theta=\theta_0} \tag{6}$$

Here λ controls the scale of the shaping vector and θ_0 denotes the rollout parameters. During optimization, PPO performs M total inner gradient steps (where $M = \text{epochs} \times \text{minibatches}$). At each of these steps, the parameter update is applied as shown in Equation (7):

$$\theta \leftarrow \theta - \alpha\Big(\nabla_\theta \mathcal{L}_{\text{PPO}} + \frac{g_{\text{BOS}}}{M}\Big) \tag{7}$$

Our implementation also supports a likelihood-coefficient form that computes the same endpoint value sensitivity with respect to role-conditioned action likelihoods, detaches these coefficients, and optimizes a stable surrogate policy loss during PPO. Both forms use the same softmax-Bayes chain and the same endpoint critic $V_\phi(o_{t+k}, B_{t+k})$; they differ only in how the resulting correction is applied numerically. Further details of our algorithm are provided in Appendix A.5.

4.3 Belief as a Sufficient Statistic for Opponent Policy

In hidden-role games, we can represent a Bayesian-rational opponent conditioning their actions on their posterior belief about the shaping agent's role:

$$\pi^*_{\text{opp}}(a \mid o_t) = \sum_{z \in Z} b^z_t \cdot \pi^*_{\text{opp}}(a \mid o_t, z) \tag{8}$$

Proposition 1 (Belief is a sufficient statistic)**.** *Let the opponent be Bayesian-rational with policy taking the role-mixture form (Eq. 8), and assume the role-conditioned responses $\pi^*_{opp}(\cdot \mid o_t, z)$ are independent of the shaper's parameters θ_1. Then the marginal action distribution $\pi^*_{opp}(\cdot \mid o_t)$ depends on θ_1 only through b_t:*

$$\frac{\partial \pi^*_{opp}(a \mid o_t)}{\partial \theta_1} = \sum_{z \in Z} \frac{\partial b^z_t}{\partial \theta_1} \cdot \pi^*_{opp}(a \mid o_t, z) \tag{9}$$

The gradient of the opponent's policy w.r.t. the shaper's parameters θ_1 is fully determined by $\partial \mathbf{b}_t / \partial \theta_1$, the D-BOS gradient.

A proof is provided in Appendix A.7.

4.4 A Meta-Gradient View (and Relation to LOLA)

Opponent shaping methods can be seen as differentiating through an *opponent update map*. To make this precise, we introduce an opponent internal state η_t that parameterizes their behavior (e.g., network weights, beliefs, or a sufficient statistic).

Theorem 1 (Opponent shaping via differentiable opponent state)**.** *Let the opponent act according to a policy $\pi_{opp}(\cdot \mid o_t, \eta_t)$ and update an internal state $\eta_t \in \mathbb{R}^d$ via a differentiable map*

$$\eta_{t+1} = U(\eta_t, \theta_1, \xi_t), \tag{10}$$

where θ_1 are the shaper's parameters and ξ_t collects variables observed by the opponent (including the shaper's action). For any objective $J_1(\theta_1, \eta_t)$ differentiable in both arguments,

$$\nabla_{\theta_1} J_1(\theta_1, \eta_t) = \underbrace{\left.\frac{\partial J_1}{\partial \theta_1}\right|_{\eta_t}}_{\textit{direct term}} + \underbrace{(\nabla_{\theta_1} \eta_t)^\top \nabla_{\eta_t} J_1(\theta_1, \eta_t)}_{\textit{shaping term}}. \tag{11}$$

Moreover, $\nabla_{\theta_1} \eta_t$ is obtained by differentiating through the recursion in Eq. 10.

A proof is provided in Appendix A.8.

Corollary 1 (D-BOS as belief-state shaping)**.** *Under a Bayesian-rational observer model (Eq. 8), we may take $\eta_t = \mathbf{b}_t$ and U to be the Bayes/softmax update (Eq. 4). Here, the shaping term in Eq. 11 depends on $\nabla_{\theta_1} \mathbf{b}_t$, which D-BOS computes by backpropagating through the k-step belief chain.*

Corollary 2 (LOLA as parameter-state shaping)**.** *LOLA is recovered as another instantiation of Theorem 1 by taking $\eta_t = \theta_2$ (opponent parameters) and the update map*

$$U(\theta_2, \theta_1, \cdot) = \theta_2 + \alpha \nabla_{\theta_2} J_2(\theta_1, \theta_2). \tag{12}$$

Differentiating through this U introduces the mixed Hessian $\partial^2 J_2 / \partial \theta_1 \partial \theta_2$ that characterizes LOLA.

This framing shows that **D-BOS and LOLA share the same meta-gradient structure**, but D-BOS chooses an opponent state (belief) with a known, low-dimensional, closed-form update map.

4.5 Advantages of Belief-Space Shaping

Dimensionality. $\mathbf{b}_t \in \Delta^{|Z|}$ has dimension $|Z| - 1$, which is typically far smaller than the parameter space of a deep opponent policy. This compact state makes it tractable to maintain and shape separate belief trajectories for multiple observers at the same time.

First-order belief gradients. The softmax-Bayes formulation in Eq. 4 makes the observer belief update differentiable with respect to the shaper's policy parameters through the likelihood of the shaper's observed actions. D-BOS therefore obtains a first-order shaping signal through the belief update itself, rather than differentiating through an opponent optimizer.

Different assumptions. D-BOS models the state of the observer's posterior rather than the opponent's optimizer state. The opponent need not literally execute Bayes' rule; the belief is a differentiable surrogate maintained by the shaper to model and influence what the observer is likely to infer.

5 Theoretical Analysis: Gradient Error from ToM Approximation

The D-BOS gradient is computed through the shaper's model of what each observer believes. This model may use a second-order estimate $\hat{o}_{i,j,i}$ that differs from the observation proxy $o_{j,i}$ used by the observer's ideal Bayesian update. We bound how this approximation error propagates through the k-step belief chain and into the shaping gradient.

5.1 Per-Step Belief Error

Definition 1 (Log-likelihood error)**.** *For observer A_j at step t, define the per-step log-likelihood error as:*

$$\varepsilon_t^{(j)} = \max_{z \in Z} |\log \pi_\theta(a_t \mid o_{j,i}, z) - \log \pi_\theta(a_t \mid \hat{o}_{i,j,i}, z)| \tag{13}$$

This measures the worst-case discrepancy over roles between the log-likelihoods computed with the observer's true estimate versus the shaper's second-order estimate.

Lemma 1 (Softmax Lipschitz property)**.** *For any two input vectors $x, y \in \mathbb{R}^{|Z|}$:*

$$\|\mathrm{softmax}(x) - \mathrm{softmax}(y)\|_1 \leq 2 \|x - y\|_\infty \tag{14}$$

A proof is provided in Appendix A.9.

Proposition 2 (k-step belief error)**.** *Let $\mathbf{b}_{t+k}^{(j)}$ be the observer's true belief after k steps and $\hat{\mathbf{b}}_{t+k}^{(j)}$ be the shaper's approximation using $\hat{o}_{i,j,i}$ at each step. Assume the initial beliefs agree: $\hat{\mathbf{b}}_t^{(j)} = \mathbf{b}_t^{(j)}$. Then:*

$$\left\|\hat{\mathbf{b}}_{t+k}^{(j)} - \mathbf{b}_{t+k}^{(j)}\right\|_1 \leq 2\sum_{s=0}^{k-1} \varepsilon_{t+s}^{(j)} \tag{15}$$

If the per-step error is bounded uniformly by $\varepsilon^{(j)} = \max_t \varepsilon_t^{(j)}$:

$$\left\|\hat{\mathbf{b}}_{t+k}^{(j)} - \mathbf{b}_{t+k}^{(j)}\right\|_1 \leq 2k\varepsilon^{(j)} \tag{16}$$

A proof is provided in Appendix A.10.

5.2 Gradient Error Bound

Theorem 2 (Gradient error from ToM approximation)**.** *Assume: (1) the per-step log-likelihood error is bounded by $\varepsilon^{(j)}$, (2) the policy gradient is Lipschitz in observations with constant L_π, (3) beliefs are bounded away from zero: $b_{t+s}^{(j),z} \geq b_{\min} > 0$ and $\hat{b}_{t+s}^{(j),z} \geq b_{\min} > 0$ for all z, s, (4) the future-state value function $V_\phi(o_{t+k}^{(i)}, \cdot)$ is L_V-Lipschitz in its belief argument, and (5) the gradient $\nabla_b V_\phi(o_{t+k}^{(i)}, \cdot)$ is L_g-Lipschitz in its belief argument (in $\|\cdot\|_1$). Then the gradient error over a k-step window satisfies:*

$$\left\|\nabla_\theta L_{\text{appr.}}^{(j)} - \nabla_\theta L_{\text{true}}^{(j)}\right\|_2 \leq \frac{C(\mathcal{Z}) \cdot k \cdot \max(L_V, L_g)}{b_{\min}^2} \cdot \left(k\varepsilon^{(j)} G_\pi + L_\pi \cdot \sup_t \|o_{j,i} - \hat{o}_{i,j,i}\|_\infty\right), \tag{17}$$

where $G_\pi = \max_{t,z} \|\nabla_\theta \log \pi_\theta(a_t \mid \hat{o}_{i,j,i}, z)\|_2$ is the maximum per-role policy-gradient norm and $C(\mathcal{Z}) = (22|\mathcal{Z}| - 18)$

A complete proof is provided in Appendix A.11.

5.3 Implications

The bound highlights three practical tradeoffs:

Window size k vs. ToM accuracy. The gradient error grows with the length of the belief rollout. Longer lookahead windows can expose useful multi-step belief trajectories, but they also amplify errors in the shaper's second-order observation model. This predicts the empirical need to tune k rather than always choosing the longest available horizon.

Number of observers. Observers for which the shaper has poor ToM estimates (large $\varepsilon^{(j)}$) contribute disproportionately to the gradient error. This suggests observer weighting as a natural extension, e.g., allocating more shaping weight to observers whose beliefs can be modeled reliably.

Belief concentration. The $1/b_{\min}^2$ factor means the gradient error is amplified when the observer's belief is nearly certain about some role. This is intuitive: when the observer is very confident, a small error in the estimated belief corresponds to a large error in the log-belief gradient. In practice, this suggests that the D-BOS gradient is most reliable when the observer is still uncertain, before the belief state has saturated.

Comparison to BBM. BBM also uses a second-order observer estimate, but only to construct a single-step intrinsic reward. D-BOS trades this myopic reward for a differentiable k-step belief trajectory. The benefit is temporal credit assignment over future beliefs; the cost is that approximation error can accumulate through the unrolled belief updates. The bound makes this tradeoff explicit.

5.4 Empirical Validation of the Gradient Bound

We test the gradient-error bound in a small closed-form hidden-role game in which exact and approximate belief-update gradients can be computed. We vary ε and the number of belief-update steps k, and measure how much the gradient changes and whether it still points in a similar direction. Because the observation-mismatch term of Theorem 2 is zero by construction, this tests the $k\varepsilon$ term

of the bound. The error grows close to linearly in ε (fitted exponent 0.92) and sub-quadratically in k (fitted exponent 0.67), consistent with the theorem as an upper bound, and the approximate gradient remains positively aligned with the exact gradient in all 24 tested configurations (mean cosine similarity 0.863).

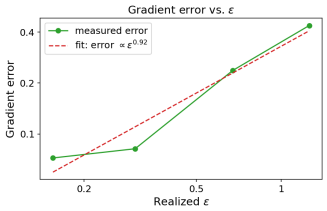


Figure 1: Gradient error vs. approximation error ε ($k = 4$). Measured scaling is $\varepsilon^{0.92}$.

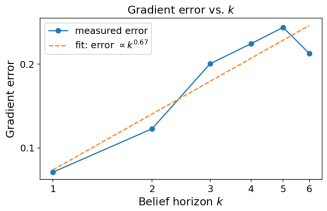


Figure 2: Gradient error vs. belief horizon k ($\varepsilon = 1$). Measured scaling is $k^{0.67}$.

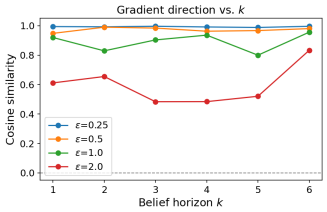


Figure 3: Cosine similarity between the approximate D-BOS gradient and the exact gradient remains positive.

6 Experiments and Results

We evaluate D-BOS across three environments of varying social complexity: Rescue-the-General, Avalon, and Multi-Agent Coin Game. Our experiments aim to answer three questions. First, can differentiating through belief dynamics produce a nonzero shaping signal that is meaningfully coupled to future return? Second, does increasing the belief-planning horizon k improve performance in controlled mixed-motive settings? Third, how stable is the shaping signal in richer hidden-role games with several observers and nonstationary co-trained policies?

Baselines

1. **PPO / no shaping**: role-conditioned PPO trained on extrinsic reward only.
2. **BBM** [10]: PPO with the per-timestep Bayes-factor intrinsic reward.
3. **Advantage Alignment (AA)** [16]: Advantage alignment where cooperation vs competition is defined by historical advantage interactions
4. **LOLA-DiCE** [27]: PPO with a one-step LOLA opponent lookahead estimated with the DiCE objective
5. **Belief-ValuePPO (ablation)**: PPO whose shaper uses the D-BOS BeliefCritic as its value baseline, without the D-BOS shaping gradient (canonical and estimated observation modes).
6. **D-BOS (ours)**: k-step differentiable belief-based opponent shaping with different k

We run Advantage Alignment and LOLA-DiCE in Avalon and the Multi-Agent Coin Game, but exclude them from the visually complex Rescue-the-General environment. Our policies use a shared encoder with role-conditioned policy heads, so both baselines are adapted to this architecture.

Metrics For each environment, we report the shaping agent's score or win rate when available over 6 seeds, as well as significance tests. The specific role hypotheses $\mathcal{Z}$ vary by setting:

- **Avalon**: A 5-player social deduction game where spies (team of 2) use social actions to deceive resistance members (team of 3) and sabotage missions.
- **Rescue-the-General (RTG)**: A 6-agent visual grid world where a red team aims to eliminate a general before a blue team can successfully rescue them.
- **Multi-agent CoinGame**: A spatial game where a shaper with a randomized role (altruistic vs. selfish) competes against three observers to collect coins.

We provide specific environment details in Appendix A.2

6.1 Results

Tables 1, 2, and 3 summarize the performance of D-BOS against PPO, BBM, Advantage Alignment, and LOLA-DiCE. BBM falls below vanilla PPO in both Avalon (win rate 0.317 vs. 0.385) and RTG (0.077 vs. 0.127). This is consistent with a core motivation of D-BOS: a hardcoded, myopic deception objective can be counterproductive, since an agent that greedily minimizes a Bayes factor at every

step behaves in ways observers can detect. D-BOS instead lets the task reward determine the direction and magnitude of belief shaping over a multi-step horizon which provides modest gains over PPO and BBM but not Advantage Alignment.

Method	Win Rate
PPO (No shaping)	0.127 ± 0.024
BBM	0.077 ± 0.023
D-BOS, $k = 1$	$\mathbf{0.149 \pm 0.055}$
D-BOS, $k = 3$	$\mathbf{0.149 \pm 0.038}$
D-BOS, $k = 5$	0.087 ± 0.023

Table 1: Rescue-the-General win rate of the red (shaper) agent on training episodes. Mean $\pm$ standard error over 6 seeds of each seed's last five values.

Method	Coin Return
PPO (No shaping)	-5.727 ± 0.089
BBM	-6.088 ± 0.382
Advantage Alignment	$\mathbf{7.878 \pm 0.161}$
LOLA-DiCE	-5.457 ± 0.141
D-BOS, $k = 1$	-6.199 ± 0.180
D-BOS, $k = 3$	-6.103 ± 0.129
D-BOS, $k = 5$	-5.749 ± 0.066

Table 2: Multi-Agent Coin Game return of the red (shaper) agent on evaluation episodes. Mean $\pm$ standard error over 6 seeds of each seed's last five evaluations.

Method	Win Rate
PPO (No shaping)	0.385 ± 0.039
BBM	0.317 ± 0.050
Advantage Alignment	$\mathbf{0.711 \pm 0.018}$
LOLA-DiCE	0.395 ± 0.028
Belief-ValuePPO	0.319 ± 0.019
Belief-ValuePPO (Est.)	0.345 ± 0.019
D-BOS, $k = 1$	0.370 ± 0.039
D-BOS, $k = 3$	0.408 ± 0.067
D-BOS, $k = 5$	0.515 ± 0.068
D-BOS, $k = 1$ (Est.)	0.427 ± 0.051
D-BOS, $k = 3$ (Est.)	0.414 ± 0.024

Table 3: Avalon win rate of the spy (shaper) agent on evaluation episodes. Mean $\pm$ standard error over 6 seeds of each seed's last five evaluations.

Avalon: Social Deduction with Mixed-Motive Shaping. Avalon is our primary testbed; its information asymmetry requires the agent to shape multiple competing observer beliefs using purely social actions. As Table 3 shows, canonical D-BOS ($k \geq 3$) outperforms PPO and BBM in mean win rate (up to 0.515 vs. 0.385 at $k = 5$), suggesting a benefit from multi-step belief planning in social deduction, although the significance tests don't reveal a high significance. Because Avalon is a mixed-motive setting, relying on a shared Canonical proxy may limit the agent's ability to model distinct perspectives. Using the **Estimated Mode** to model these conflicting viewpoints gives a D-BOS variant with results significantly above PPO on held-out evaluation ($k = 1$, $p = 0.026$). Belief-ValuePPO, which uses the same BeliefCritic without the shaping gradient, does not improve over PPO (0.319 canonical, 0.345 estimated), so the BeliefCritic alone does not explain this improvement. Observer-specific proxies are intended to let the shaping gradient treat adversaries and allies differently. We provide belief trajectory graphs in Appendix A.15.

Rescue-the-General (RTG): High-Dimensional Scaling. RTG introduces severe visual complexity, which we expect to increase the Theory of Mind approximation error (ε). Theorem 2 then suggests that shorter belief horizons should give a more reliable shaping gradient. Table 1 is consistent with this: D-BOS with $k = 1$ and $k = 3$ shows a small improvement over PPO (0.149 vs. 0.127), while $k = 5$ falls below PPO (0.087), consistent with errors compounding over longer horizons. Visual deception natively suffers from high variance [10], and in some seeds D-BOS sustains performance through the mid-training collapse that PPO and BBM undergo, though not consistently across seeds. Full training curves are shown in Appendix A.15. This motivates more careful tuning of k, but D-BOS shows potential even in highly complex games.

CoinGame: Boundary Conditions. CoinGame tests a strict algorithmic limitation: latent roles resample ephemerally and signaling is purely spatial. Here, D-BOS does not improve over PPO: it falls below PPO for $k = 1$ and $k = 3$ and matches it for $k = 5$, and on held-out evaluation $k = 1$ matches PPO. This suggests the method's scope: multi-step belief shaping is most useful with persistent identities, and provides little advantage when roles constantly reset and belief channels carry sparse information or change temporally.

Comparison with opponent-shaping baselines. Advantage Alignment outperforms D-BOS in both Avalon and the Coin Game, while LOLA-DiCE performs on par with PPO. We attribute part of this gap to the shaping target: an advantage is a reward-derived, low-variance training signal, whereas D-BOS shapes an inferred belief state whose useful direction is task-dependent and must itself be learned. Advantage Alignment also modifies the advantages of every agent, whereas D-BOS and LOLA-DiCE change only the shaper's update. (Appendix A.12).

Statistical significance. We evaluate final checkpoints on 50 held-out episodes per seed and compare each method against PPO with a Mann-Whitney U test over the pooled episodes (Appendix A.13). In Avalon, D-BOS $k = 1$ (Est.) is significantly above PPO (0.463 vs. 0.373, $p = 0.026$), as is Advantage Alignment (0.713, $p < 0.001$). Advantage Alignment is also significantly above PPO in the Coin Game ($p < 0.001$). In Rescue-the-General, BBM ($p = 0.004$) and D-BOS $k = 5$ ($p = 0.006$) are significantly below PPO. The remaining differences are not significant.

7 Discussion and Limitations

D-BOS is best viewed as an explicit mechanism for belief-space opponent shaping, not as a drop-in reward bonus that automatically dominates PPO. The experiments show that the mechanism is active: the BeliefCritic can learn future belief-state value, the softmax-Bayes chain produces differentiable gradients, and these gradients are injected into the policy update. They also show the main optimization challenge. In complex hidden-role games, the shaping signal can be small or noisy relative to the policy gradient. Increasing λ or adaptively amplifying the gradient makes the effect visible, but can hurt performance if the critic or belief trajectory introduces error.

Broader impacts and interpretability. Beyond the specific hidden-role games studied here, D-BOS addresses a fundamental challenge in multi-agent systems: how an agent should systematically influence the mental models others form about it. This dynamic arises whenever agents with private information must coordinate or compete. Practical examples include autonomous vehicles signaling intent to human drivers, assistive robots communicating capabilities to human teammates, and negotiation algorithms managing impressions in strategic interactions. By grounding opponent shaping in an explicit, low-dimensional belief simplex rather than opaque parameter perturbations, D-BOS provides an auditable, realistic mechanism. Further, practitioners can directly inspect what the agent is attempting to make observers believe by analyzing the critic's value assignments over the belief space. This transparency is a strict requirement for deploying influence-aware agents in domains where accountability is necessary.

Evaluation against co-trained opponents. Because D-BOS shapes the beliefs of opponents that are simultaneously being trained, the resulting policies are adapted to each other. The win rates reflect co-trained evaluation, and transferability to unseen opponents remains an open question. Evaluating D-BOS against held-out or independently trained opponents would provide a stronger test of whether the learned shaping strategies generalize beyond the co-training distribution.

Future directions. Extending D-BOS to continuous belief spaces (A.1), analyzing performance with n-order Theory of Mind, and developing adaptive observer-weighting schemes (as suggested by the per-observer error bound in Theorem 2) are natural next steps. We also notice that recursive modeling reduces ε but introduces additional learned modules whose own approximation error must be controlled, motivating a joint search over the horizon k and reasoning depth n. Further, the belief horizon k is currently set per environment; selecting or adapting it online, for example from estimates of the observer-model error that drives the bound in Theorem 2, is a promising direction.

8 Conclusion

We introduced Belief-based Opponent Shaping, an opponent-shaping method that treats opponent belief states as the differentiable opponent state. D-BOS backpropagates through k steps of softmax-Bayes belief dynamics, allowing the reward function rather than a hand-coded deception objective to determine whether beliefs should move toward cooperation, concealment, or adversarial confusion and could tell us which beliefs contribute to such strategies. Our theory connects the method to meta-gradient opponent shaping and bounds the cost of approximate second-order Theory of Mind. Empirically, D-BOS produces real shaping gradients and improves on PPO and BBM in mean performance in several settings, although it does not reach state-of-the-art performance: Advantage Alignment remains stronger in our experiments, and richer hidden-role games reveal the sensitivity of belief-space shaping to critic quality and gradient scale. D-BOS nonetheless offers an explicit, interpretable opponent state, whose shaping direction follows the task reward rather than a hand-coded objective. We see this as a first step toward opponent shaping methods that operate on the beliefs agents form about one another, rather than directly on policy parameters.

Acknowledgments and Disclosure of Funding

This research project was supported in part by the Defense Advanced Research Projects Agency (DARPA) Young Faculty Award. The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of DARPA or the U.S. Government.

References

[1] David Premack and Guy Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, 1978.

[2] Heinz Wimmer and Josef Perner. Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children's understanding of deception. *Cognition*, (1), 1983.

[3] Emre Erdogan, Frank Dignum, Rineke Verbrugge, and Pinar Yolum. Abstracting minds: Computational theory of mind for human-agent collaboration. In *HHAI2022: Augmenting Human Intellect*, 2022.

[4] Manisha Natarajan, Esmaeil Seraj, Batuhan Altundas, Rohan Paleja, Sean Ye, Letian Chen, Reed Jensen, Kimberlee Chang, and Matthew Gombolay. Human-robot teaming: Grand challenges. *Current Robotics Reports*, 4:1–20, 08 2023.

[5] Letian Chen, Rohan Paleja, Muyleng Ghuy, and Matthew Gombolay. Joint goal and strategy inference across heterogeneous demonstrators via reward network distillation. In *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2020.

[6] Rohan Paleja, Muyleng Ghuy, Nadun Ranawaka Arachchige, Reed Jensen, and Matthew Gombolay. The utility of explainable ai in ad hoc human-machine teaming. 2022.

[7] Patrick Shafto, Noah D Goodman, and Thomas L Griffiths. A rational account of pedagogical reasoning: Teaching by, and learning from, examples. *Cognitive Psychology*, 2014.

[8] Mark K. Ho, Michael L. Littman, James MacGlashan, Fiery Cushman, and Joseph L. Austerweil. Showing versus doing: Teaching by demonstration. In *Annual Conference on Neural Information Processing Systems 2016*, 2016.

[9] Jack Serrino, Max Kleiman-Weiner, David C. Parkes, and Josh Tenenbaum. Finding friend and foe in multi-agent games. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019.

[10] Matthew Aitchison, Lyndon Benke, and Penny Sweetser. Learning to deceive in multi-agent hidden role games. In *First International Workshop on Deceptive AI*, Communications in Computer and Information Science, 2021.

[11] Roberta Raileanu, Emily Denton, Arthur Szlam, and Rob Fergus. Modeling others using oneself in multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, Proceedings of Machine Learning Research, 2018.

[12] Jakob N Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 2018.

[13] Milad Aghajohari, Juan Agustin Duque, Tim Cooijmans, and Aaron Courville. LOQA: learning with opponent q-learning awareness. In *International Conference on Learning Representations, ICLR 2024*, 2024.

[14] Stephen Zhao, Chris Lu, Roger B. Grosse, and Jakob N. Foerster. Proximal learning with opponent-learning awareness. In *Neural Information Processing Systems 2022*, 2022.

[15] Timon Willi, Alistair Letcher, Johannes Treutlein, and Jakob N. Foerster. COLA: consistent learning with opponent-learning awareness. In *International Conference on Machine Learning, ICML 2022*, 2022.

[16] Juan Agustin Duque, Milad Aghajohari, Tim Cooijmans, Razvan Ciuca, Tianyu Zhang, Gauthier Gidel, and Aaron Courville. Advantage alignment algorithms. In *International Conference on Learning Representations (ICLR)*, 2025. Oral.

[17] Alistair Letcher, Jakob N. Foerster, David Balduzzi, Tim Rocktäschel, and Shimon Whiteson. Stable opponent shaping in differentiable games. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[18] Christopher Lu, Timon Willi, Christian A. Schröder de Witt, and Jakob N. Foerster. Model-free opponent shaping. In *International Conference on Machine Learning, ICML 2022*, 2022.

[19] Neil C. Rabinowitz, Frank Perbet, H. Francis Song, Chiyuan Zhang, S. M. Ali Eslami, and Matthew Botvinick. Machine theory of mind. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, Proceedings of Machine Learning Research, 2018.

[20] Ini Oguntola. *Theory of Mind in Multi-Agent Systems*. PhD thesis, Carnegie Mellon University, 2025.

[21] Nitay Alon, Lion Schulz, Jeffrey S. Rosenschein, and Peter Dayan. A (dis-)information theory of revealed and unrevealed preferences: Emerging deception and skepticism via theory of mind. *Open Mind*, 2023.

[22] Xiaopeng Yu, Jiechuan Jiang, Wanpeng Zhang, Haobin Jiang, and Zongqing Lu. Model-based opponent modeling. In *Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*, 2022.

[23] Péter Karkus, David Hsu, and Wee Sun Lee. Qmdp-net: Deep learning for planning under partial observability. In *Annual Conference on Neural Information Processing Systems 2017*, 2017.

[24] Yanlin Han and Piotr J. Gmytrasiewicz. Ipomdp-net: A deep neural network for partially observable multi-agent planning using interactive pomdps. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, 2019.

[25] Lloyd S. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences*, 1953.

[26] Eric A. Hansen, Daniel S. Bernstein, and Shlomo Zilberstein. Dynamic programming for partially observable stochastic games. In *AAAI Workshop - Technical Report*, 2004.

[27] Jakob Foerster, Gregory Farquhar, Maruan Al-Shedivat, Tim Rocktäschel, Eric P. Xing, and Shimon Whiteson. Dice: The infinitely differentiable monte-carlo estimator, 2018.

A Technical appendices and supplementary material

A.1 Continuous Role Spaces

A natural question is whether our method only applies to discrete role spaces. While we focus on a discrete role set Z, the opponent-state view in Theorem 1 suggests a generalization to *continuous* latent roles $z \in \mathcal{Z} \subset \mathbb{R}^m$. In that setting, the observer maintains a belief density $b_t(z)$ and performs a Bayes update:

$$b_{t+1}(z) \propto \pi_\theta(a_t \mid o_t, z)\, b_t(z), \tag{18}$$

This remains a differentiable opponent update map. Practically, computing the normalizer requires integration over $\mathcal{Z}$, so one would represent b_t with an approximate family (e.g., particles, Gaussian/mixture variational beliefs, or a normalizing-flow posterior) and differentiate through the resulting approximate filter. This would preserve the key benefit of belief-space shaping, i.e., differentiating through a known state update map, while trading exactness for approximate inference in continuous spaces. This direction is complementary to first-order LOLA approximations such as Advantage Alignment [16], which remove Hessians in *parameter*-space shaping; here, the goal is to shift the shaped opponent state from parameters θ_2 to an explicit latent belief representation.

A.2 Environment Details

For fair comparison across methods, we use the same architectural backbone within each environment. For Rescue-the-General, we use the default CNN policy path, whereas for Avalon and CoinGame, we use an MLP policy path. The rollout length and total training budget vary by environment.

Rescue-the-General (RTG). RTG is a partially observable grid-world hidden-role game with six agents: one red agent, one green agent, and four blue agents. The main evaluation scenario uses a 32×32 map with local RGB observations. With the default view radius used in the paper runs, each agent receives a $3 \times 45 \times 45$ uint8 egocentric crop containing terrain, visible agents, projectiles, bodies, vote information, and task objects. Roles are not globally revealed; visibility depends on the scenario's role-visibility setting and local observation.

The discrete action set has ten actions:

$$\{\text{NOOP}, \text{UP}, \text{DOWN}, \text{LEFT}, \text{RIGHT}, \text{ACT}, \text{SHOOT-UP}, \text{SHOOT-DOWN}, \text{SHOOT-LEFT}, \text{SHOOT-RIGHT}\}. \tag{19}$$

Movement changes the agent's grid location subject to walls and map constraints. Shooting actions damage the first target in the selected direction subject to range and cooldown. The ACT action is context dependent: it can interact with the general or map objects, and it can initiate a vote when the agent is near a dead body or vote button. During a vote, actions are interpreted as vote choices rather than movement/shooting commands.

The red team's objective is to find and kill or prevent rescue of the general; the blue team attempts to rescue the general; the green agent is an additional role-conditioned participant in the RTG benchmark dynamics. Terminal outcomes are logged with RTG-compatible strings: `general_killed` corresponds to a red win and `general_rescued/blue_win` corresponds to a blue win. We train RTG with the CNN policy path, role-conditioned policy/value heads, and a deception/backward-prediction module when estimated-observation ablations are enabled. The main RTG runs use 64 parallel environments, rollout length 16, and one million environment interactions per logged epoch.

Avalon5. Avalon5 is a five-player Resistance/Avalon-style social deduction environment. It has two spies and three resistance players. The game alternates through three phases: PROPOSE, VOTE, and QUEST. In the proposal phase, the current leader selects a mission team. In the voting phase, all five players simultaneously approve or reject the proposal. A strict majority approves the team; five consecutive rejected proposals give the spies a win. In the quest phase, mission-team members choose success/fail actions; spies may sabotage by choosing fail, while resistance players are forced to contribute success. The first side to win three missions wins the game.

The observation is a compact public-history vector with shape $(128, 1, 1)$. It contains played mission history, proposed team masks, vote records, mission outcomes, sabotage counts, the current round, phase, leader identity, whether the observing player is leader or on the proposed team, consecutive rejection count, current spy/resistance mission scores, current mission size, and private bits. Private bits

include whether the observing player is a spy and whether the observing player is currently on mission. In the non-blind variant, spies additionally observe a partner-spy bitmask; in `avalon5_blind`, this partner-spy bit is removed. Thus all agents know their own role, but resistance players do not observe spy identities and blind spies do not observe the other spy.

The action space has size ten. In the proposal phase, the leader's action indexes one of the valid mission-team combinations of the required size; non-leader actions are ignored. In the voting phase, actions are thresholded into approve/reject votes. In the quest phase, selected spies can play success or fail, while selected resistance actions are interpreted as success. Rewards are team based: each mission gives $+1$ to the winning side and -1 to the losing side, and the terminal game bonus is ± 10. For D-BOS, role 0 is the primary shaper spy, role 1 is the second spy, and roles 2–4 are resistance players. Avalon5 uses the MLP policy path with role-conditioned policy/value heads; adapter runs add a shaper-specific adapter and role-0 policy correction so the D-BOS gradient is restricted to the shaper role path.

Multi-agent CoinGame. The multi-agent CoinGame is a controlled mixed-motive hidden-role grid environment. One fixed red shaper acts with a latent role $z \in \{0, 1\}$, and there are three blue observers. Each episode is a toroidal grid game with grid size 5 and horizon 30. At reset, all agents and a single coin are placed uniformly at random, and the red role is sampled uniformly.

Each agent observes a 14-dimensional vector:

$$[t/T,\ x_a,\ y_a,\ x_r,\ y_r,\ x_{b_1},\ y_{b_1},\ x_{b_2},\ y_{b_2},\ x_{b_3},\ y_{b_3},\ x_c,\ y_c,\ c], \tag{20}$$

where (x_a, y_a) is the observing agent's position, (x_r, y_r) is red's position, (x_{b_i}, y_{b_i}) are the positions of all three blue observers, (x_c, y_c) is the coin position, and c is the coin color. Coordinates are normalized before conversion to uint8. The action space has four movement actions: up, down, left, and right, with wrap-around at grid boundaries. When the coin is collected, a new coin spawns at a uniformly random cell with its color sampled uniformly at random, independently of the agents' roles; if several agents reach the coin at once, the lowest-indexed agent collects it. Blue observers act with their own role-conditioned policy head, separate from both of red's latent roles.

Collecting any coin gives the collector $+1$. If red collects a blue coin, the blue observers receive a total penalty of -2 shared equally across blue agents. If red's latent role is altruistic, red also receives an additional -1 for collecting a blue coin. If a blue observer collects a red coin, red receives -2. The red return is the primary plotted score and blue score is the sum over blue observers. D-BOS beliefs are over red's latent role; because blue observers share the same reward structure but maintain separate belief rows, CoinGame is a lightweight testbed for whether a matrix belief critic can aggregate multiple observer beliefs.

A.3 Implementation Stabilization for the Belief Bound

The gradient bound in Theorem 2 assumes all belief coordinates stay above $b_{\min} > 0$. This is not only a technical assumption: hidden-role likelihoods can become extremely sharp, pushing posterior beliefs to the simplex boundary. Near the boundary, $\log b^z$ has derivative $1/b^z$, so converting belief error to log-belief error introduces the $1/b_{\min}$ factors in Lemma 2. If the implementation allowed $b^z \to 0$, the theoretical constant would become vacuous, and the numerical gradient through $\log b^z$ could become unstable.

In code, we handle this in three ways. First, all Bayes updates are performed in log space, followed by a softmax, so the update remains differentiable and normalized:

$$b_{t+1} = \mathrm{softmax}(\ell_t + \log b_t). \tag{21}$$

Second, in the MLP environments, we use an optional belief floor. After each update, the posterior is mixed with a uniform distribution in (22):

$$\tilde{b}_{t+1} = (1 - \alpha_{\text{floor}}) b_{t+1} + \alpha_{\text{floor}} \frac{\mathbf{1}}{|\mathcal{Z}|}, \tag{22}$$

where $\alpha_{\text{floor}} =$ belief floor. This implies $\tilde{b}^z_{t+1} \geq \alpha/|\mathcal{Z}|$, so in the theorem one may take $b_{\min} = \alpha/|\mathcal{Z}|$ for runs using this floor. We also use likelihood temperature defined in (23) with $\tau > 1$ in some ablations, which prevents a single action likelihood from saturating the posterior too early.

$$\ell_t \leftarrow \ell_t / \tau, \tag{23}$$

Third, we also added a coefficient implementation that gates or clips unstable updates. Windows whose terminal beliefs are already too certain are assigned zero coefficient, since these are exactly the windows where the softmax-Bayes Jacobian is least informative. The coefficients can also be RMS-normalized and clipped. This does not change the formal belief-error bound, but it prevents the applied policy correction from being dominated by rare high-magnitude windows. In RTG, the CNN path additionally clips the final shaping gradient to the same maximum norm scale used by PPO and optionally caps adaptive amplification relative to the PPO gradient norm. Thus, the $b_{\min}$ dependence is handled both analytically, by assuming beliefs remain inside the simplex interior, and operationally, by keeping beliefs away from zero and clipping the applied shaping correction.

A.4 Canonical and Estimated Observation Models

The Bayes likelihood in the paper is written as $\pi_\theta(a_i \mid h_{i,j,i}, z)$, where $h_{i,j,i}$ is the shaper's estimate of what observer j believes the shaper observes. In the canonical approximation, we set $h_{i,j,i}$ equal to the shaper's own local observation. This means all observers share the same role-conditioned action likelihood at a given timestep, but maintain separate priors $b_t^{(j)}$. Thus, the canonical setting shapes a matrix of observer beliefs, while the evidence model is shared unless observers have distinct priors or visibility weights.

In estimated-observation ablations, $h_{i,j,i}$ is predicted rather than copied. In RTG, this is the existing backwards observation-prediction head in the deception model: during training, it is supervised with observer-local observations, and D-BOS uses the predicted observer observation in the differentiable Bayes likelihood. In the MLP environments, the estimated-observation ablation uses a small LSTM predictor trained from rollout targets. Its inputs are the shaper observation trajectory together with observer role and observer slot embeddings; its target is the observer's local observation. The D-BOS likelihood is given in Equation (24)

$$\ell_t^{j,z} = \log \pi_\theta(a_t^i \mid \hat{o}_{i,j,i,t}, z), \tag{24}$$

Here, $\hat{o}_{i,j,i,t}$ is the predicted observer-conditioned observation. We do not use the observer's true private observation directly for final claims; using it directly would be a privileged-information diagnostic rather than an inferred Theory-of-Mind model.

A.5 Likelihood-Coefficient D-BOS Correction

The direct autograd form of D-BOS minimizes

$$\mathcal{L}_{\mathrm{BOS}}^k(\theta) = -V_\phi(o_{t+k}, B_{t+k}), \tag{25}$$

where B_{t+k} is the full observer-by-role belief matrix obtained by chaining k softmax-Bayes updates. Backpropagating this loss through the belief chain is conceptually simple, but in practice, it can create noisy gradients through repeated softmax Jacobians. Our coefficient implementation computes the endpoint sensitivity in likelihood space and then applies it as a stable surrogate policy loss.

For a window starting at t, let $\ell_s^{j,z} = \log \pi_\theta(a_s^i \mid \hat{o}_{i,j,i,s}, z)$ be the log-likelihood of the shaper's observed action under role hypothesis z for observer j. The belief chain is

$$B_{s+1} = U(B_s, \ell_s), \qquad s = t, \ldots, t+k-1, \tag{26}$$

where U applies the softmax-Bayes update row-wise over observer beliefs. We first compute

$$c_s^{j,z} = \frac{\partial[-V_\phi(o_{t+k}, B_{t+k})]}{\partial \ell_s^{j,z}}. \tag{27}$$

These coefficients already contain the product of the endpoint value gradient, the intervening belief Jacobians, and the softmax-Bayes Jacobian. We then detach c_s and optimize the surrogate

$$\tilde{\mathcal{L}}_{\mathrm{BOS}} = \frac{1}{k} \sum_{s=t}^{t+k-1} \frac{1}{|\mathcal{J}||\mathcal{Z}|} \sum_{j,z} \bar{c}_s^{j,z}\, \ell_s^{j,z}, \tag{28}$$

where $\bar{c}_s$ denotes the optionally gated, normalized, and clipped coefficient tensor. Detaching the coefficients makes the correction first-order with respect to the policy parameters in the PPO update while preserving the direction computed by the differentiable belief planner at the rollout parameters.

Algorithm 1 Differentiable Belief-based Opponent Shaping (D-BOS) with PPO

Require: Role-conditioned policy $\pi_\theta(a \mid o, z)$; belief critic V_ϕ; rollout length T; shaping horizon k; shaping weight λ; observers $\mathcal{J}$

Initialize per-observer beliefs $\{\hat{\mathbf{b}}_0^j\}_{j\in\mathcal{J}}$
for $t = 0, \ldots, T-1$ **do**
  Sample shaper action $a_t \sim \pi_\theta(\cdot \mid o_t^{(i)}, z_i)$ and step the environment
  for each observer $j \in \mathcal{J}$ **do**
    Choose observer proxy $\hat{o}_{i,j,i,t}$: estimated DeceptionModel proxy when available, otherwise canonical proxy
    Compute role likelihoods $\ell_t^{j,z} \leftarrow \log \pi_\theta(a_t \mid \hat{o}_{i,j,i,t}, z)$ for all $z \in \mathcal{Z}$
    Update belief $\hat{\mathbf{b}}_{t+1}^j \leftarrow \mathrm{softmax}(\ell_t^j + \log \hat{\mathbf{b}}_t^j)$
  end for
  Store $(o_t, a_t, r_t, z_i, \log \pi_\theta(a_t \mid o_t^{(i)}, z_i), \{\hat{\mathbf{b}}_t^j\}_{j\in\mathcal{J}})$ in the rollout buffer
end for
Compute bootstrapped extrinsic returns $\{R_t\}_{t=0}^{T-1}$
Train V_ϕ by MSE regression on future endpoints:
$V_\phi(o_{t+k}^{(i)}, \hat{\mathbf{B}}_{t+k}) \approx R_{t+k}, \quad t = 0, \ldots, T-k-1$
Freeze ϕ and compute a rollout-level belief correction g_{BOS} from $\{-V_\phi(o_{t+k}^{(i)}, \hat{\mathbf{B}}_{t+k})\}_{t=0}^{T-k-1}$
Clip, gate, or adaptively scale g_{BOS} using rollout diagnostics
for each PPO minibatch **do**
  Compute PPO gradient $g_{\mathrm{PPO}} \leftarrow \nabla_\theta \mathcal{L}_{\mathrm{PPO}}$
  Apply the PPO update with the D-BOS correction to the shaper policy path:
  $\theta \leftarrow \theta - \alpha(g_{\mathrm{PPO}} + \mathrm{Apply}(g_{\mathrm{BOS}}))$
end for

For hidden-role shaping, adapter runs apply the D-BOS correction only to the shaper's role-conditioned policy path rather than to opponent-role policy rows. In coefficient mode, this is implemented by zeroing all coefficient entries except the shaper-role entry $z = 0$ before normalization and clipping. This prevents the shaping term from directly modifying observer-role policies while keeping the base PPO training pipeline shared across methods.

The coefficient implementation logs the mean absolute coefficient, RMS coefficient, entropy-gate fraction, clipping fraction, approximate KL, pre/post clipping D-BOS gradient norms, and final injected D-BOS gradient norm. These diagnostics are used to verify that the shaping term is present but not overwhelming the PPO gradient.

A.6 Algorithm outlined

Construction of o_{t+k} and B_{t+k}. The implementation loops over valid rollout windows $t = 0, \ldots, T-k-1$. For each environment and each observer j, it initializes the observer belief from the rollout buffer,

$$B_t = \{\mathbf{b}_t^j\}_{j\in\mathcal{J}}, \tag{29}$$

where `buf.per_obs_beliefs[t]` stores one role posterior per observer. It then applies exactly k row-wise softmax-Bayes updates using the shaper action likelihoods at rollout times $t, \ldots, t+k-1$:

$$\mathbf{b}_{s+1}^j = \mathrm{Reg}\big[\mathrm{softmax}\big(\ell_s^j + \log \mathbf{b}_s^j\big)\big], \qquad s = t, \ldots, t+k-1, \tag{30}$$

where $\ell_s^j = (\ell_s^{j,z})_{z\in\mathcal{Z}}$ and Reg denotes the belief floor/regularization used to avoid simplex-boundary saturation. After the final update, the code flattens the matrix $B_{t+k} \in \mathbb{R}^{|\mathcal{J}|\times|\mathcal{Z}|}$ into a vector and evaluates the belief critic on the shaper's future observation:

$$o_{t+k} = \texttt{buf.obs[t+k, shaper_index]}, \qquad V_\phi(o_{t+k}, \mathrm{vec}(B_{t+k})). \tag{31}$$

The belief critic is trained on the same indexing convention: for role-0/shaper samples, it regresses $V_\phi(o_{t+k}, B_{t+k})$ to the rollout return stored at time $t+k$. Thus the shaping query and critic target both use the same future pair (o_{t+k}, B_{t+k}).

A.7 Proof of Proposition 1: Belief is a sufficient statistic

Proof. Since $\pi^*_{\text{opp}}(a \mid o_t, z)$ is independent of θ_1 by assumption, the only dependence on θ_1 in Eq. 8 is through b_t^z. Therefore holding b_t fixed, the marginal distribution $\pi^*_{\text{opp}}(a \mid o_t)$ is fully determined and does not depend on θ_1, meaning that b_t is a sufficient statistic.
Differentiating Eq. 8 with respect to θ_1 using the product rule:

$$\frac{\partial \pi^*_{opp}(a|o_t)}{\partial \theta_1} = \sum_z \frac{\partial b_t^z}{\partial \theta_1} \cdot \pi^*_{opp}(a|o_t, z) + \sum_z b_t^z \cdot \underbrace{\frac{\partial \pi^*_{opp}(a|o_t, z)}{\partial \theta_1}}_{=0 \text{ (by assumption)}} = \sum_z \frac{\partial b_t^z}{\partial \theta_1} \cdot \pi^*_{\text{opp}}(a \mid o_t, z)$$

The gradient flows entirely through b_t. □

A.8 Proof of Theorem 1: Opponent shaping via differentiable opponent state

Proof. Since J_1 is differentiable in both arguments and η_t depends on θ_1 through the differentiable recursion in Eq. 10, the chain rule gives

$$\nabla_{\theta_1} J_1(\theta_1, \eta_t) = \left.\frac{\partial J_1}{\partial \theta_1}\right|_{\eta_t} + (\nabla_{\theta_1} \eta_t)^\top \nabla_{\eta_t} J_1(\theta_1, \eta_t), \tag{32}$$

which is Eq. 11.

To compute $\nabla_{\theta_1} \eta_t$, differentiate the state recursion (Eq. 10):

$$\nabla_{\theta_1} \eta_{t+1} = \frac{\partial U}{\partial \eta_t} \nabla_{\theta_1} \eta_t + \frac{\partial U}{\partial \theta_1}, \tag{33}$$

where the partial derivatives are evaluated at $(\eta_t, \theta_1, \xi_t)$. We Initialize with $\nabla_{\theta_1} \eta_0 = 0$ since the initial state η_0 is independent of θ_1. Unrolling this recursion yields an explicit expression for $\nabla_{\theta_1} \eta_t$ in terms of Jacobians of U:

$$\nabla_{\theta_1} \eta_{t+1} = \sum_{i=0}^{t} \left(\prod_{j=i+1}^{t} \frac{\partial U}{\partial \eta_j} \right) \left.\frac{\partial U}{\partial \theta_1}\right|_{t=i} \tag{34}$$

□

A.9 Proof of Lemma 1: Softmax Lipschitz property

Proof. Let $f(x) = \text{softmax}(x)$. For any x write $b = f(x)$. The Jacobian of f at x has entries

$$J_{z',z}(x) = b^{z'} \left(\mathbf{1}[z' = z] - b^z\right), \tag{35}$$

or equivalently $J(x) = \text{diag}(b) - bb^\top$.

Bounding $\|J(x)\|_{\infty \to 1}$. For any vector $v \in \mathbb{R}^{|\mathcal{Z}|}$,

$$J(x)\, v = \text{diag}(b)\, v - b\, (b^\top v) = b \odot v - (b^\top v)\, b, \tag{36}$$

where $\odot$ denotes component-wise multiplication. Taking the 1-norm and applying the triangle inequality,

$$\|J(x)v\|_1 \le \|b \odot v\|_1 + |b^\top v| \cdot \|b\|_1. \tag{37}$$

Each term: $\|b \odot v\|_1 = \sum_z b^z |v^z| \le \|b\|_1 \cdot \|v\|_\infty = \|v\|_\infty$ (since $\|b\|_1 = 1$ for a probability vector) and $|b^\top v| \le \|b\|_1 \|v\|_\infty = \|v\|_\infty$ by Hölder. Combining,

$$\|J(x)v\|_1 \le \|v\|_\infty + \|v\|_\infty = 2\|v\|_\infty. \tag{38}$$

Hence $\|J(x)\|_{\infty \to 1} \le 2$ at every x.

Define $z(t) = y + t(x - y)$. By the fundamental theorem of calculus,

$$\begin{aligned}
\|f(x) - f(y)\|_1 &= \left\| \int_0^1 J(z(t))(x-y)\, dt \right\|_1 \\
&\leq \int_0^1 \|J(z(t))(x-y)\|_1\, dt \quad \text{(triangle inequality)} && (39)\\
&\leq \int_0^1 2\|x-y\|_\infty\, dt \quad \text{(operator norm bound)} && (40)\\
&= 2\|x-y\|_\infty. && (41)
\end{aligned}$$

□

A.10 Proof of Proposition 2: k-step belief error

Proof. Recall the per-step log-likelihood vector

$$\ell_r := \log \pi_\theta(a_r \mid o_r, \cdot) \in \mathbb{R}^{|\mathcal{Z}|}, \tag{42}$$

whose z-th entry $\ell_r^z = \log \pi_\theta(a_r \mid o_r, z)$ is the log-probability that role z would have taken action a_r given observation o_r. The approximate version $\hat{\ell}_r := \log \pi_\theta(a_r \mid \hat{o}_r, \cdot)$ uses RED's second-order observation estimate $\hat{o}_r$ in place of the observer's true estimate o_r.

Define the cumulative log-likelihood vectors

$$L_{t+k} := \log b_t + \sum_{i=0}^{k-1} \ell_{t+i}, \qquad \hat{L}_{t+k} := \log b_t + \sum_{i=0}^{k-1} \hat{\ell}_{t+i}, \tag{43}$$

where the starting log-belief $\log b_t$ is shared (since $b_t = \hat{b}_t$ by assumption). Applying k softmax-Bayes updates is equivalent to a single softmax of the cumulative logits:

$$b_{t+k} = \mathrm{softmax}(L_{t+k}), \qquad \hat{b}_{t+k} = \mathrm{softmax}(\hat{L}_{t+k}). \tag{44}$$

Applying Lemma 1,

$$\|\hat{b}_{t+k} - b_{t+k}\|_1 = \|\mathrm{softmax}(\hat{L}_{t+k}) - \mathrm{softmax}(L_{t+k})\|_1 \leq 2\|\hat{L}_{t+k} - L_{t+k}\|_\infty. \tag{45}$$

The cumulative difference is

$$\hat{L}_{t+k} - L_{t+k} = \sum_{i=0}^{k-1} \big(\hat{\ell}_{t+i} - \ell_{t+i}\big), \tag{46}$$

so by the triangle inequality and the per-step error bound $\|\hat{\ell}_{t+i} - \ell_{t+i}\|_\infty \leq \varepsilon_{t+i}$ by definition 1,

$$\|\hat{L}_{t+k} - L_{t+k}\|_\infty \leq \sum_{i=0}^{k-1} \|\hat{\ell}_{t+i} - \ell_{t+i}\|_\infty \leq \sum_{i=0}^{k-1} \varepsilon_{t+i}. \tag{47}$$

Combining,

$$\|\hat{b}_{t+k} - b_{t+k}\|_1 \leq 2 \sum_{i=0}^{k-1} \varepsilon_{t+i} \leq 2k\varepsilon \tag{48}$$

under the uniform bound $\varepsilon = \max_t \varepsilon_t$. □

A.11 Proof of Theorem 2: Gradient error from ToM approximation

We prove Theorem 2 in stages. First we establish two auxiliary results: a log-conversion lemma and a trajectory-drift corollary. We then bound the magnitude of each factor in the chain-rule expansion of $\nabla_\theta L^{(j)}$.

Throughout, we drop the observer index (j) for compactness, fixing one observer j. The bound applies per-observer, aggregation across observers is handled separately. We use the norm conventions: $\|\cdot\|_1$ on belief vectors, $\|\cdot\|_\infty$ on log-beliefs and log-likelihood vectors, $\|\cdot\|_2$ on parameter-space vectors.

Lemma 2 (Log conversion lemma)**.** *Let $b, \hat{b} \in \Delta^{|\mathcal{Z}|}$ satisfy $b^z \geq b_{\min}$ and $\hat{b}^z \geq b_{\min}$ for all $z \in \mathcal{Z}$, where $b_{\min} > 0$. Then*

$$\|\log \hat{b} - \log b\|_\infty \leq \frac{1}{b_{\min}} \|\hat{b} - b\|_1. \tag{49}$$

Proof. Let $z^* \in \arg\max_z |\log \hat{b}^z - \log b^z|$, so that

$$\|\log \hat{b} - \log b\|_\infty = |\log \hat{b}^{z^*} - \log b^{z^*}|. \tag{50}$$

Both b^{z^*} and $\hat{b}^{z^*}$ lie in $[b_{\min}, 1]$. By the Mean Value Theorem applied to $f(x) = \log x$ on this interval, there exists $c \in [b_{\min}, 1]$ such that

$$\log \hat{b}^{z^*} - \log b^{z^*} = \frac{1}{c}\big(\hat{b}^{z^*} - b^{z^*}\big). \tag{51}$$

Taking absolute values and using $c \geq b_{\min}$:

$$|\log \hat{b}^{z^*} - \log b^{z^*}| = \frac{1}{c}|\hat{b}^{z^*} - b^{z^*}| \leq \frac{1}{b_{\min}}|\hat{b}^{z^*} - b^{z^*}|. \tag{52}$$

Finally, the absolute difference at any single coordinate is bounded by the total ℓ_1 difference:

$$|\hat{b}^{z^*} - b^{z^*}| \leq \sum_z |\hat{b}^z - b^z| = \|\hat{b} - b\|_1. \tag{53}$$

Combining these,

$$\|\log \hat{b} - \log b\|_\infty \leq \frac{1}{b_{\min}} \|\hat{b} - b\|_1. \tag{54}$$

□

Proposition 2 establishes that for the endpoint of a k-steps,

$$\|\hat{b}_{t+k} - b_{t+k}\|_1 \leq 2 \sum_{s=0}^{k-1} \varepsilon_{t+s} \leq 2k\varepsilon, \tag{55}$$

where $\varepsilon = \max_t \varepsilon_t$. The same induction applies at every intermediate step, giving the following trajectory-wide drift bound.

Corollary 3 (Log-belief trajectory drift)**.** *Under the assumptions of Proposition 2, for every $m \in \{0, 1, \ldots, k\}$,*

$$\|\hat{b}_{t+m} - b_{t+m}\|_1 \leq 2 \sum_{s=0}^{m-1} \varepsilon_{t+s} \leq 2m\varepsilon. \tag{56}$$

Proof. Apply Proposition 2 with horizon m in place of k. The induction in the proof of Proposition 2 makes no use of k being distinguished from any other horizon length □

Combining Corollary 3 with Lemma 2 gives the corresponding bound in ∞-norm of log-beliefs, which we use in subsequent steps.

Corollary 4. *Under the assumptions of Proposition 2 and Theorem 2, for every $m \in 0, 1, ..., k$*

$$\|\log \hat{b}_{t+m} - \log b_{t+m}\|_\infty \leq \frac{2m\varepsilon}{b_{\min}}. \tag{57}$$

Proof. Apply Lemma 2 to b_{t+m} and $\hat{b}_{t+m}$:

$$\|\log \hat{b}_{t+m} - \log b_{t+m}\|_\infty \leq \frac{1}{b_{\min}} \|\hat{b}_{t+m} - b_{t+m}\|_1 \leq \frac{2m\varepsilon}{b_{\min}}. \tag{58}$$

□

A.11.1 Chain-rule expansion of $\nabla_\theta L$

We expand $\nabla_\theta L = -\nabla_\theta V_\phi(o_{t+k}, b_{t+k})$ using the chain rule. The parameter θ enters the loss through k separate channels, one per step in the window: at step $s \in \{t, \dots, t+k-1\}$, θ controls the log-likelihood vector

$$\ell_s := \log \pi_\theta(a_s \mid o_s, \cdot) \in \mathbb{R}^{|\mathcal{Z}|}, \tag{59}$$

which then propagates through softmax-Bayes updates $b_{r+1} = \mathrm{softmax}(\ell_r + \log b_r)$ until reaching b_{t+k}, which feeds into V_ϕ.

By the multivariate chain rule, summing over all k entry points,

$$\nabla_\theta L = \sum_{s=t}^{t+k-1} \frac{\partial L}{\partial b_{t+k}} \cdot \left(\prod_{r=s+1}^{t+k-1} \frac{\partial b_{r+1}}{\partial b_r} \right) \cdot \frac{\partial b_{s+1}}{\partial \ell_s} \cdot \nabla_\theta \ell_s. \tag{60}$$

For compactness, denote the four kinds of factors as

$$g := \frac{\partial L}{\partial b_{t+k}} \qquad \text{(endpoint gradient)}, \tag{61}$$

$$J_r := \frac{\partial b_{r+1}}{\partial b_r} \qquad \text{(belief-to-belief Jacobian)}, \tag{62}$$

$$S_s := \frac{\partial b_{s+1}}{\partial \ell_s} \qquad \text{(softmax Jacobian)}, \tag{63}$$

$$G_s := \nabla_\theta \ell_s \qquad \text{(policy-gradient matrix)}. \tag{64}$$

and let

$$\Pi_s := \prod_{r=s+1}^{t+k} J_r \tag{65}$$

denote the belief-Jacobian chain from step $s+1$ to step $t+k$. Equation 60 can compactly be written as

$$\nabla_\theta L = \sum_{s=t}^{t+k-1} g \cdot \Pi_s \cdot S_s \cdot G_s. \tag{66}$$

The approximate gradient $\nabla_\theta L_{\text{approx}}$ has the same form as Eq. 66, but every factor is evaluated along the approximate trajectory $(\hat{b}_s)_{s=t}^{t+k}$ generated using $\hat{o}_s$ in place of o_s. We denote the approximate factors with hats: $\hat{g}, \hat{J}_r, \hat{S}_s, \hat{G}_s, \hat{\Pi}_s$. Both trajectories share the same starting belief, $\hat{b}_t = b_t$.

A.11.2 Magnitude bounds on the chain-rule factors

We bound each of the chain-rule factors in the appropriate operator norm.

Proposition 3 (Magnitude bounds)**.** *Under the assumptions of Theorem 2, the factors of Eq. 66 satisfy*

$$\|g\|_\infty \le L_V, \tag{67}$$

$$\|S_s\|_{\infty \to 1} \le 2, \tag{68}$$

$$\|G_s\|_{2 \to \infty} \le G_\pi. \tag{69}$$

The same bounds hold for $\hat{g}, \hat{S}_s, \hat{G}_s$ along the approximate trajectory.

Proof. **Bound on g.** Assumption (4) states that $V_\phi(o, \cdot)$ is L_V-Lipschitz in $\|\cdot\|_1$:

$$|V_\phi(o, b) - V_\phi(o, b')| \le L_V \|b - b'\|_1 \quad \forall b, b' \in \Delta^{|\mathcal{Z}|}. \tag{70}$$

By the duality between $\|\cdot\|_1$ and $\|\cdot\|_\infty$, the gradient $g = \partial L / \partial b_{t+k} = -\partial V_\phi / \partial b_{t+k}$ satisfies

$$\|g\|_\infty = \sup_{\|\delta b\|_1 \le 1} |g \cdot \delta b| \le L_V. \tag{71}$$

Bound on S_s. Recall $b_{s+1} = \mathrm{softmax}(\ell_s + \log b_s)$, so $S_s = \partial b_{s+1}/\partial \ell_s$ is the Jacobian of $\mathrm{softmax}$ evaluated at $\ell_s + \log b_s$. Lemma 1 states

$$\|\mathrm{softmax}(x) - \mathrm{softmax}(y)\|_1 \le 2\|x - y\|_\infty \quad \forall x, y \in \mathbb{R}^{|\mathcal{Z}|}, \tag{72}$$

i.e., $\mathrm{softmax}$ is 2-Lipschitz from $\|\cdot\|_\infty$ to $\|\cdot\|_1$. For any direction $v \in \mathbb{R}^{|\mathcal{Z}|}$ and $\eta > 0$, applying this to $x = u + \eta v$ and $y = u$ gives

$$\|\mathrm{softmax}(u + \eta v) - \mathrm{softmax}(u)\|_1 \le 2\eta\|v\|_\infty. \tag{73}$$

Dividing by η and taking $\eta \to 0$ yields $\|J_{\mathrm{softmax}}(u)\, v\|_1 \le 2\|v\|_\infty$ for all v, so the softmax Jacobian satisfies $\|J_{\mathrm{softmax}}(u)\|_{\infty\to 1} \le 2$ at every input u. In particular, evaluating at $u = \ell_s + \log b_s$,

$$\|S_s\|_{\infty\to 1} \le 2. \tag{74}$$

Bound on G_s. The matrix $G_s \in \mathbb{R}^{|\mathcal{Z}|\times d_\theta}$ has rows $G_s^{z,:} = \nabla_\theta \log \pi_\theta(a_s \mid o_s, z) \in \mathbb{R}^{d_\theta}$ indexed by role z. For any $x \in \mathbb{R}^{d_\theta}$, the z-th coordinate of $G_s x$ is the dot product $G_s^{z,:} \cdot x$, so

$$\|G_s x\|_\infty = \max_z |G_s^{z,:} \cdot x|. \tag{75}$$

Taking the supremum over $\|x\|_2 \le 1$ and exchanging $\sup$ and $\max$,

$$\|G_s\|_{2\to\infty} = \sup_{\|x\|_2\le 1} \max_z |G_s^{z,:} \cdot x| = \max_z \sup_{\|x\|_2\le 1} |G_s^{z,:} \cdot x|. \tag{76}$$

The inner supremum is the dual norm of $G_s^{z,:}$ under $\|\cdot\|_2$; since $\|\cdot\|_2$ is self-dual, Cauchy–Schwarz gives $\sup_{\|x\|_2\le 1} |G_s^{z,:} \cdot x| = \|G_s^{z,:}\|_2$, with equality attained at $x = G_s^{z,:}/\|G_s^{z,:}\|_2$. Hence

$$\|G_s\|_{2\to\infty} = \max_z \|G_s^{z,:}\|_2 = \max_z \|\nabla_\theta \log \pi_\theta(a_s \mid o_s, z)\|_2 \le G_\pi, \tag{77}$$

where the last inequality follows from the definition of G_π.

The bounds for the approximate factors $\hat{g}, \hat{S}_s, \hat{G}_s$ follow from the same arguments applied along the approximate trajectory. □

A.11.3 Decomposition of the gradient error

Define the per-summand contributions to $\nabla_\theta L$ and $\nabla_\theta L_{\text{approx}}$ as

$$T_s := g \cdot \Pi_s \cdot S_s \cdot G_s, \qquad \hat{T}_s := \hat{g} \cdot \hat{\Pi}_s \cdot \hat{S}_s \cdot \hat{G}_s, \tag{78}$$

so that $\nabla_\theta L_{\text{approx}} - \nabla_\theta L_{\text{true}} = \sum_{s=t}^{t+k-1} (\hat{T}_s - T_s)$.

Within each summand, the hat and no-hat versions differ in all four factors. To isolate one source of error at a time, we telescope by swapping one factor at a time, from left to right. Adding and subtracting intermediate hybrids,

$$\begin{aligned} \hat{T}_s - T_s = {} & \underbrace{(\hat{g} - g)\, \hat{\Pi}_s \hat{S}_s \hat{G}_s}_{\text{(A)}} + \underbrace{g\, (\hat{\Pi}_s - \Pi_s) \hat{S}_s \hat{G}_s}_{\text{(B)}} \\ & + \underbrace{g\, \Pi_s (\hat{S}_s - S_s) \hat{G}_s}_{\text{(C)}} + \underbrace{g\, \Pi_s S_s (\hat{G}_s - G_s)}_{\text{(D)}}. \end{aligned} \tag{79}$$

Each piece isolates the perturbation of a single factor. Summing over s and applying the triangle inequality in $\|\cdot\|_2$,

$$\left\|\nabla_\theta L_{\text{approx}} - \nabla_\theta L_{\text{true}}\right\|_2 \le \sum_{s=t}^{t+k-1} \Big(\|(\text{A})_s\|_2 + \|(\text{B})_s\|_2 + \|(\text{C})_s\|_2 + \|(\text{D})_s\|_2 \Big). \tag{80}$$

The four pieces are bounded individually in the following subsections.

A.11.4 Bound on piece (D): policy-gradient error

Piece (D) isolates the error from evaluating the policy gradient at $\hat{o}_s$ instead of o_s:

$$(D)_s = g \cdot \Pi_s \cdot S_s \cdot (\hat{G}_s - G_s). \tag{81}$$

By submultiplicativity of operator norms,

$$\|(D)_s\|_2 \le \|g\|_\infty \cdot \|\Pi_s\|_{1\to 1} \cdot \|S_s\|_{\infty\to 1} \cdot \|\hat{G}_s - G_s\|_{2\to\infty}. \tag{82}$$

The first three factors are bounded via Proposition 3 and Section A.11.7. We bound the difference factor A.11.7 $\|\hat{G}_s - G_s\|_{2\to\infty}$ here.

The matrix $\hat{G}_s - G_s \in \mathbb{R}^{|\mathcal{Z}|\times d_\theta}$ has rows

$$(\hat{G}_s - G_s)^{z,:} = \nabla_\theta \log \pi_\theta(a_s|\hat{o}_s, z) - \nabla_\theta \log \pi_\theta(a_s|o_s, z) \in \mathbb{R}^{d_\theta}. \tag{83}$$

By Assumption (2), each row satisfies

$$\|(\hat{G}_s - G_s)^{z,:}\|_2 \le L_\pi \|\hat{o}_s - o_s\|_\infty. \tag{84}$$

Following the same argument as in the proof of Proposition 3 (the $2 \to \infty$ operator norm equals the maximum row 2-norm),

$$\|\hat{G}_s - G_s\|_{2\to\infty} = \max_z \|(\hat{G}_s - G_s)^{z,:}\|_2 \le L_\pi \|\hat{o}_s - o_s\|_\infty. \tag{85}$$

Combining with the magnitude bounds from Proposition 3 ($\|g\|_\infty \le L_V$, $\|S_s\|_{\infty\to 1} \le 2$), Eq. 82 becomes

$$\|(D)_s\|_2 \le 2L_V \cdot \|\Pi_s\|_{1\to 1} \cdot L_\pi \|\hat{o}_s - o_s\|_\infty, \tag{86}$$

where the bound on $\|\Pi_s\|_{1\to 1}$ is established in Section A.11.7.

A.11.5 Bound on piece (A): endpoint error

By submultiplicativity of operator norms,

$$\|(A)_s\|_2 \le \|\hat{g} - g\|_\infty \cdot \|\hat{\Pi}_s\|_{1\to 1} \cdot \|\hat{S}_s\|_{\infty\to 1} \cdot \|\hat{G}_s\|_{2\to\infty}. \tag{87}$$

Bounding $\|\hat{g} - g\|_\infty$ uses Assumption 5 that $\nabla_b V_\phi$ itself be Lipschitz in b, with some constant L_g. Under this assumption,

$$\|\hat{g} - g\|_\infty \le L_g \|\hat{b}_{t+k} - b_{t+k}\|_1 \le 2k\varepsilon \cdot L_g \tag{88}$$

by Corollary 3.

Substituting into Eq. 87 and using Proposition 3,

$$\|(A)_s\|_2 \le 2k\varepsilon L_g \cdot \|\hat{\Pi}_s\|_{1\to 1} \cdot 2 \cdot G_\pi = 4k\varepsilon L_g G_\pi \cdot \|\hat{\Pi}_s\|_{1\to 1}, \tag{89}$$

where the bound on $\|\hat{\Pi}_s\|_{1\to 1}$ is established in Section A.11.7.

A.11.6 Bound on piece (C): softmax error

By submultiplicativity,

$$\|(C)_s\|_2 \le \|g\|_\infty \cdot \|\Pi_s\|_{1\to 1} \cdot \|\hat{S}_s - S_s\|_{\infty\to 1} \cdot \|\hat{G}_s\|_{2\to\infty}. \tag{90}$$

The softmax Jacobian admits the closed form $S_s = \mathrm{diag}(b_{s+1}) - b_{s+1} b_{s+1}^\top$. Letting $\delta := \hat{b}_{s+1} - b_{s+1}$ and using $\hat{x}\hat{x}^\top - xx^\top = \hat{x}\delta^\top + \delta x^\top$,

$$\hat{S}_s - S_s = \mathrm{diag}(\delta) - \hat{b}_{s+1}\delta^\top - \delta b_{s+1}^\top. \tag{91}$$

We bound each of the three terms in $\infty \to 1$ operator norm. For the diagonal term, $\|\mathrm{diag}(\delta)x\|_1 = \sum_z |\delta_z x_z| \le \|\delta\|_1 \|x\|_\infty$. For each rank-one term $uv^\top$, the output $u(v^\top x)$ has 1-norm $\|u\|_1 \cdot |v^\top x| \le \|u\|_1 \|v\|_1 \|x\|_\infty$ by Hölder's inequality, and since $\|b_{s+1}\|_1 = \|\hat{b}_{s+1}\|_1 = 1$ (probability vectors), each rank-one term contributes at most $\|\delta\|_1$. By the triangle inequality,

$$\|\hat{S}_s - S_s\|_{\infty\to 1} \le 3\|\hat{b}_{s+1} - b_{s+1}\|_1 \le 6k\varepsilon \tag{92}$$

where the second inequality is Corollary 3.

Combining with Proposition 3,

$$\|(C)_s\|_2 \le 6k\varepsilon L_V G_\pi \cdot \|\Pi_s\|_{1\to 1}. \tag{93}$$

A.11.7 Bound on piece (B): Jacobian-chain error

The chain Π_s is a product of $t+k-s-1$ belief-to-belief Jacobians. We establish that the chain Π_s *collapses algebraically* to a single Jacobian whose form depends only on the starting belief b_{s+1} and the ending belief b_{t+k}, with all intermediate beliefs disappearing. This collapse keeps the dependence on $1/b_{\min}$ polynomial.

Lemma 3 (Jacobian-chain collapse)**.** *For any r and $m \geq 0$,*

$$J_{r+m} \cdot J_{r+m-1} \cdots J_{r+1} \cdot J_r = D_{r+m+1} \left(I - \mathbf{1}\, b_{r+m+1}^\top\right) D_r^{-1}. \tag{94}$$

In particular, applying this with $r = s+1$ and $m = t+k-s-2$,

$$\Pi_s = D_{t+k} \left(I - \mathbf{1}\, b_{t+k}^\top\right) D_{s+1}^{-1}, \qquad \hat{\Pi}_s = \hat{D}_{t+k} \left(I - \mathbf{1}\, \hat{b}_{t+k}^\top\right) \hat{D}_{s+1}^{-1}, \tag{95}$$

where $D_r := \mathrm{diag}(b_r)$ and $\hat{D}_r := \mathrm{diag}(\hat{b}_r)$.

Proof. By induction on m.

Base case ($m = 0$). The claim reads

$$J_r = D_{r+1}(I - \mathbf{1}\, b_{r+1}^\top)\, D_r^{-1}. \tag{96}$$

We derive this from the chain rule. The forward computation is $b_{r+1} = \mathrm{softmax}(\ell_r + \log b_r)$, where $\ell_r := \log \pi_\theta(a_r \mid o_r, \cdot) \in \mathbb{R}^{|\mathcal{Z}|}$ is the log-likelihood vector at step r. Note that ℓ_r depends on θ, a_r, and o_r but not on b_r. Writing $u := \ell_r + \log b_r$ for the input to softmax, the dependency chain is

$$b_r \;\longrightarrow\; \log b_r \;\longrightarrow\; u = \ell_r + \log b_r \;\longrightarrow\; b_{r+1} = \mathrm{softmax}(u), \tag{97}$$

giving by the chain rule

$$J_r = \frac{\partial b_{r+1}}{\partial b_r} = \frac{\partial b_{r+1}}{\partial u} \cdot \frac{\partial u}{\partial (\log b_r)} \cdot \frac{\partial (\log b_r)}{\partial b_r}. \tag{98}$$

We compute each factor.

The map $b_r \mapsto \log b_r$ acts component-wise with $d \log x / dx = 1/x$, so its Jacobian is the diagonal matrix

$$\frac{\partial (\log b_r)}{\partial b_r} = \mathrm{diag}(1/b_r) = D_r^{-1}. \tag{99}$$

Since ℓ_r does not depend on b_r,

$$\frac{\partial u}{\partial (\log b_r)} = \frac{\partial \ell_r}{\partial (\log b_r)} + \frac{\partial (\log b_r)}{\partial (\log b_r)} = 0 + I = I. \tag{100}$$

The standard softmax Jacobian formula gives, for $b_{r+1} = \mathrm{softmax}(u)$,

$$\frac{\partial b_{r+1}}{\partial u} = \mathrm{diag}(b_{r+1}) - b_{r+1} b_{r+1}^\top. \tag{101}$$

Using $b_{r+1} = \mathrm{diag}(b_{r+1})\mathbf{1}$, we factor $b_{r+1} b_{r+1}^\top = \mathrm{diag}(b_{r+1})\, \mathbf{1}\, b_{r+1}^\top$, hence

$$\frac{\partial b_{r+1}}{\partial u} = \mathrm{diag}(b_{r+1}) \left[I - \mathbf{1}\, b_{r+1}^\top\right] = D_{r+1}(I - \mathbf{1}\, b_{r+1}^\top). \tag{102}$$

Combining the three factors,

$$J_r = D_{r+1}(I - \mathbf{1}\, b_{r+1}^\top) \cdot I \cdot D_r^{-1} = D_{r+1} \left(I - \mathbf{1}\, b_{r+1}^\top\right) D_r^{-1}. \tag{103}$$

This establishes the base case.

Inductive step. Assume the claim holds for m. Multiplying on the left by $J_{r+m+1} = D_{r+m+2}(I - \mathbf{1} b_{r+m+2}^\top) D_{r+m+1}^{-1}$:

$$\begin{aligned} J_{r+m+1} \cdot \left(J_{r+m} \cdots J_r\right) &= D_{r+m+2}(I - \mathbf{1} b_{r+m+2}^\top)\, D_{r+m+1}^{-1} \cdot D_{r+m+1}(I - \mathbf{1} b_{r+m+1}^\top)\, D_r^{-1} \\ &= D_{r+m+2}(I - \mathbf{1} b_{r+m+2}^\top)(I - \mathbf{1} b_{r+m+1}^\top)\, D_r^{-1}, \end{aligned} \tag{104}$$

where $D_{r+m+1}^{-1} D_{r+m+1} = I$ cancels. We claim that

$$(I - \mathbf{1}b_{r+m+2}^\top)(I - \mathbf{1}b_{r+m+1}^\top) = I - \mathbf{1}b_{r+m+2}^\top. \tag{105}$$

For any x, the right-hand factor gives $(I - \mathbf{1}b_{r+m+1}^\top)x = x - \mathbf{1}(b_{r+m+1}^\top x)$. Applying the left-hand factor and using $(I - \mathbf{1}b_{r+m+2}^\top)\mathbf{1} = \mathbf{1} - \mathbf{1}\,(b_{r+m+2}^\top \mathbf{1}) = 0$ (since $b_{r+m+2}^\top \mathbf{1} = 1$), the second term vanishes, leaving $(I - \mathbf{1}b_{r+m+2}^\top)x$. Equation 105 follows. Substituting,

$$J_{r+m+1} \cdots J_r = D_{r+m+2}(I - \mathbf{1}b_{r+m+2}^\top)\, D_r^{-1}, \tag{106}$$

which is the inductive claim for $m + 1$. □

The closed form in Eq.95 reduces Π_s to three matrix factors. We now bound the magnitude $\|\Pi_s\|_{1\to 1}$, which is used in the bounds on pieces (A), (C), (D), and the difference $\|\hat{\Pi}_s - \Pi_s\|_{1\to 1}$ used in piece (B).

Lemma 4 (Magnitude bound on Π_s)**.** *Under the assumptions of Theorem 2,*

$$\|\Pi_s\|_{1\to 1} \le \frac{|\mathcal{Z}| - 1}{b_{\min}}. \tag{107}$$

The same bound holds for $\|\hat{\Pi}_s\|_{1\to 1}$.

Proof. Recall the identity $\|M\|_{1\to 1} = \max_j \sum_i |M_{ij}|$ (maximum column 1-norm). By Eq. 95 and submultiplicativity of operator norms,

$$\|\Pi_s\|_{1\to 1} \le \|D_{t+k}\|_{1\to 1} \cdot \|I - \mathbf{1}b_{t+k}^\top\|_{1\to 1} \cdot \|D_{s+1}^{-1}\|_{1\to 1}. \tag{108}$$

We bound each factor.

Bound on $\|D_{t+k}\|_{1\to 1}$. Each column of D_{t+k} has a single nonzero entry $b_{t+k}^j \in [0, 1]$, so

$$\|D_{t+k}\|_{1\to 1} = \max_j b_{t+k}^j \le 1. \tag{109}$$

Bound on $\|D_{s+1}^{-1}\|_{1\to 1}$. Each column of D_{s+1}^{-1} has a single nonzero entry $1/b_{s+1}^j$.

$$\|D_{s+1}^{-1}\|_{1\to 1} = \max_j \frac{1}{b_{s+1}^j} = \frac{1}{\min_j b_{s+1}^j} \le \frac{1}{b_{\min}}. \tag{110}$$

Bound on $\|I - \mathbf{1}b_{t+k}^\top\|_{1\to 1}$. Column j in $\|I - \mathbf{1}b_{t+k}^\top\|_{1\to 1}$ has entry $(1 - b_{t+k}^j)$ at position j and $-b_{t+k}^j$ at each of the other $|\mathcal{Z}| - 1$ positions. Its 1-norm is

$$(1 - b_{t+k}^j) + (|\mathcal{Z}| - 1)\, b_{t+k}^j = 1 + (|\mathcal{Z}| - 2)\, b_{t+k}^j \le |\mathcal{Z}| - 1. \tag{111}$$

Combining,

$$\|\Pi_s\|_{1\to 1} \le 1 \cdot (|\mathcal{Z}| - 1) \cdot \frac{1}{b_{\min}} = \frac{|\mathcal{Z}| - 1}{b_{\min}}. \tag{112}$$

The same bound applies to $\|\hat{\Pi}_s\|_{1\to 1}$ via the analogous argument along the approximate trajectory. □

We now bound the difference $\|\hat{\Pi}_s - \Pi_s\|_{1\to 1}$, which feeds directly into piece (B).

Lemma 5 (Perturbation bound on Π_s)**.** *Under the assumptions of Theorem 2,*

$$\|\hat{\Pi}_s - \Pi_s\|_{1\to 1} \le \frac{(6|\mathcal{Z}| - 4)k\,\varepsilon}{b_{\min}^2}, \tag{113}$$

Proof. By the closed forms in Eq.95 and add-subtract telescoping along the three factors,

$$\begin{aligned}\hat{\Pi}_s - \Pi_s = &\underbrace{(\hat{D}_{t+k} - D_{t+k})(I - \mathbf{1}\hat{b}_{t+k}^\top)\hat{D}_{s+1}^{-1}}_{\text{(i)}} \\ &+ \underbrace{D_{t+k}\,\mathbf{1}(b_{t+k} - \hat{b}_{t+k})^\top\,\hat{D}_{s+1}^{-1}}_{\text{(ii)}} \\ &+ \underbrace{D_{t+k}(I - \mathbf{1}b_{t+k}^\top)(\hat{D}_{s+1}^{-1} - D_{s+1}^{-1})}_{\text{(iii)}},\end{aligned} \tag{114}$$

where in (ii) the I's cancel and the rank-one difference $(I-\mathbf{1}\hat{b}_{t+k}^\top)-(I-\mathbf{1}b_{t+k}^\top) = \mathbf{1}(b_{t+k}-\hat{b}_{t+k})^\top$ remains. We bound each piece in $\|\cdot\|_{1\to1}$ via submultiplicativity, using the magnitude bounds from the proof of Lemma 4 and Corollary 3.

Piece (i). The diagonal difference satisfies $\|\hat{D}_{t+k} - D_{t+k}\|_{1\to1} = \max_z |\hat{b}_{t+k}^z - b_{t+k}^z| \le \|\hat{b}_{t+k} - b_{t+k}\|_1 \le 2k\varepsilon$. Combined with $\|I - \mathbf{1}\hat{b}_{t+k}^\top\|_{1\to1} \le |\mathcal{Z}| - 1$ and $\|\hat{D}_{s+1}^{-1}\|_{1\to1} \le 1/b_{\min}$,

$$\|\text{(i)}\|_{1\to1} \le 2k\varepsilon \cdot (|\mathcal{Z}| - 1) \cdot \frac{1}{b_{\min}} = \frac{2(|\mathcal{Z}| - 1)\,k\varepsilon}{b_{\min}}. \tag{115}$$

Piece (ii). The rank-one matrix $\mathbf{1}(b_{t+k} - \hat{b}_{t+k})^\top$ has column j equal to $(b_{t+k}^j - \hat{b}_{t+k}^j)\mathbf{1}$, whose 1-norm is $|\mathcal{Z}| \cdot |b_{t+k}^j - \hat{b}_{t+k}^j|$. Hence $\|\mathbf{1}(b_{t+k} - \hat{b}_{t+k})^\top\|_{1\to1} \le |\mathcal{Z}|\,\|b_{t+k} - \hat{b}_{t+k}\|_\infty \le 2|\mathcal{Z}|k\varepsilon$. Combined with $\|D_{t+k}\|_{1\to1} \le 1$ and $\|\hat{D}_{s+1}^{-1}\|_{1\to1} \le 1/b_{\min}$,

$$\|\text{(ii)}\|_{1\to1} \le 1 \cdot 2|\mathcal{Z}|k\varepsilon \cdot \frac{1}{b_{\min}} = \frac{2|\mathcal{Z}|\,k\varepsilon}{b_{\min}}. \tag{116}$$

Piece (iii). The diagonal difference $\hat{D}_{s+1}^{-1} - D_{s+1}^{-1}$ has entries

$$\frac{1}{\hat{b}_{s+1}^z} - \frac{1}{b_{s+1}^z} = \frac{b_{s+1}^z - \hat{b}_{s+1}^z}{b_{s+1}^z \hat{b}_{s+1}^z}. \tag{117}$$

Using $b_{s+1}^z, \hat{b}_{s+1}^z \ge b_{\min}$ from Assumption (3),

$$\|\hat{D}_{s+1}^{-1} - D_{s+1}^{-1}\|_{1\to1} \le \max_z \frac{|b_{s+1}^z - \hat{b}_{s+1}^z|}{b_{s+1}^z \hat{b}_{s+1}^z} \le \frac{2k\varepsilon}{b_{\min}^2}. \tag{118}$$

Combined with $\|D_{t+k}\|_{1\to1} \le 1$ and $\|I - \mathbf{1}b_{t+k}^\top\|_{1\to1} \le |\mathcal{Z}| - 1$,

$$\|\text{(iii)}\|_{1\to1} \le 1 \cdot (|\mathcal{Z}| - 1) \cdot \frac{2k\varepsilon}{b_{\min}^2} = \frac{2(|\mathcal{Z}| - 1)\,k\varepsilon}{b_{\min}^2}. \tag{119}$$

Combining. By the triangle inequality applied to Eq. 114, and using $1/b_{\min} \le 1/b_{\min}^2$ since $b_{\min} \le 1$,

$$\begin{aligned}\|\hat{\Pi}_s - \Pi_s\|_{1\to1} &\le \frac{2(|\mathcal{Z}| - 1)\,k\varepsilon}{b_{\min}} + \frac{2|\mathcal{Z}|\,k\varepsilon}{b_{\min}} + \frac{2(|\mathcal{Z}| - 1)\,k\varepsilon}{b_{\min}^2} \\ &\le \frac{(6|\mathcal{Z}| - 4)\,k\varepsilon}{b_{\min}^2}.\end{aligned} \tag{120}$$

□

We can now bound piece (B). By submultiplicativity,

$$\|(B)_s\|_2 \le \|g\|_\infty \cdot \|\hat{\Pi}_s - \Pi_s\|_{1\to1} \cdot \|\hat{S}_s\|_{\infty\to1} \cdot \|\hat{G}_s\|_{2\to\infty}. \tag{121}$$

Combining Lemma 5 with the magnitude bounds from Proposition 3,

$$\|(B)_s\|_2 \le L_V \cdot \frac{(6|\mathcal{Z}| - 4)\,k\varepsilon}{b_{\min}^2} \cdot 2 \cdot G_\pi = \frac{2(6|\mathcal{Z}| - 4)\,k\varepsilon\,L_V\,G_\pi}{b_{\min}^2}. \tag{122}$$

A.11.8 Assembly: combining the four pieces

We combine the bounds on pieces (A), (B), (C), (D), together with the bound $\|\Pi_s\|_{1\to 1} \le (|\mathcal{Z}| - 1)/b_{\min}$ from Lemma 4, to produce the final bound stated in Theorem 2.

Per-summand bounds. Substituting Lemma 4 into Eqs. 86, 89, and 93:

$$\|(D)_s\|_2 \le \frac{2(|\mathcal{Z}|-1)L_V L_\pi \|\hat{o}_s - o_s\|_\infty}{b_{\min}}, \tag{123}$$

$$\|(A)_s\|_2 \le \frac{4(|\mathcal{Z}|-1)k\varepsilon L_g G_\pi}{b_{\min}}, \tag{124}$$

$$\|(C)_s\|_2 \le \frac{6(|\mathcal{Z}|-1)k\varepsilon L_V G_\pi}{b_{\min}}, \tag{125}$$

$$\|(B)_s\|_2 \le \frac{2(6|\mathcal{Z}|-4)k\varepsilon L_V G_\pi}{b_{\min}^2} \quad \text{(Lemma 5)}. \tag{126}$$

Sum across the window. Summing each piece across $s = t, \dots, t+k-1$ (with k summands), and using $\|\hat{o}_s - o_s\|_\infty \le \sup_t \|\hat{o}_t - o_t\|_\infty$:

$$\sum_s \|(D)_s\|_2 \le \frac{2(|\mathcal{Z}|-1)kL_V L_\pi \sup_t \|\hat{o}_t - o_t\|_\infty}{b_{\min}}, \tag{127}$$

$$\sum_s \|(A)_s\|_2 \le \frac{4(|\mathcal{Z}|-1)k^2\varepsilon L_g G_\pi}{b_{\min}}, \tag{128}$$

$$\sum_s \|(C)_s\|_2 \le \frac{6(|\mathcal{Z}|-1)k^2\varepsilon L_V G_\pi}{b_{\min}}, \tag{129}$$

$$\sum_s \|(B)_s\|_2 \le \frac{2(6|\mathcal{Z}|-4)k^2\varepsilon L_V G_\pi}{b_{\min}^2}. \tag{130}$$

Final assembly. By Eq. 80, $\|\nabla_\theta L_{\text{approx}} - \nabla_\theta L_{\text{true}}\|_2$ is bounded by the sum of the four. Using $1/b_{\min} \le 1/b_{\min}^2$ (since $b_{\min} \le 1$) to put all terms on a common denominator, and $\max(L_V, L_g)$ to absorb both Lipschitz constants, the G_π contributions from (A), (B), (C) combine as

$$\begin{aligned}&\frac{\left[4(|\mathcal{Z}|-1) + 6(|\mathcal{Z}|-1) + 2(6|\mathcal{Z}|-4)\right] k^2\varepsilon \, \max(L_V, L_g)\, G_\pi}{b_{\min}^2} \\ &= \frac{(22|\mathcal{Z}|-18)\, k^2\varepsilon \, \max(L_V, L_g)\, G_\pi}{b_{\min}^2},\end{aligned} \tag{131}$$

while the L_π contribution from (D) is bounded by $\frac{2(|\mathcal{Z}|-1)kL_V L_\pi \sup_t \|\hat{o}_t - o_t\|_\infty}{b_{\min}^2}$.

Combining,

$$\left\|\nabla_\theta L_{\text{approx}} - \nabla_\theta L_{\text{true}}\right\|_2 \le \frac{(22|\mathcal{Z}|-18)\cdot k \cdot \max(L_V, L_g)}{b_{\min}^2}\left[k\varepsilon G_\pi + L_\pi \sup_t \|\hat{o}_t - o_t\|_\infty\right]. \tag{132}$$

This completes the proof of Theorem 2.

A.12 Baselines Implementation

Advantage Alignment We use the integrated advantage-alignment term of the reference implementation [16]: each agent's advantage is augmented by the product of its own past advantages with the other agents' current advantages. As in the reference self-play setting, the term is applied to every agent, not only the shaper. In Avalon, "other agents" is restricted to teammates, since summing over the opposing team in this strictly adversarial team game destabilized training. All other hyperparameters match PPO.

LOLA-DiCE The opponents take one virtual gradient step on the DiCE objective [27] of the reference implementation (per-step discounted rewards with a value baseline, over the joint log-likelihood of all agents), restricted to their rows of the role-conditioned policy head. The correction updates only the shaper's policy-head rows, and its norm is capped at 15% of the PPO gradient norm on those rows; without the cap it exceeds the PPO gradient by several orders of magnitude and dominates the shaper's update which destabilized learning.

D-BOS In Avalon and the Coin Game, the D-BOS correction updates only the shaper's rows of the role-conditioned policy head (and a shaper-only adapter), leaving the shared encoder unchanged. In Rescue-the-General, it is restricted to the shaper's rows within the policy head but also passes through the shared convolutional encoder; its norm is rescaled to 15% of the PPO gradient norm at each update.

A.13 Evaluation Protocol and Significance Tests

All methods are trained with 6 seeds. Each entry in the tables is the mean $\pm$ standard error over seeds of each seed's last five values (to smoothen out noise). For the significance tests, each final checkpoint is evaluated on 50 held-out episodes, and each method is compared against PPO with a two-sided Mann-Whitney U test over the resulting 300 pooled episodes. Held-out evaluation covers PPO, BBM, and D-BOS $k \in \{1, 3, 5\}$ in Rescue-the-General; PPO, BBM, D-BOS $k = 1$, Advantage Alignment, and LOLA-DiCE in the Coin Game; and PPO, D-BOS $k = 1$ (Est.), Advantage Alignment, and LOLA-DiCE in Avalon.

A.14 Setup of the Gradient-Bound Validation

The game in Section 5.4 has $|\mathcal{Z}| = 4$ latent types and 3 actions. Role-conditioned policies are softmax policies with logits drawn from $\mathcal{N}(0, 1)$. The shaper's observer model adds a fixed random perturbation to these logits, scaled to a target magnitude; the realized error ε is the largest absolute log-likelihood difference for the true role's row. The loss is $-\log \hat{b}^{z^*}_{t+k}$, and beliefs use the same softmax-Bayes update with a belief floor of 0.02 as our implementation, which keeps $b_{\min} > 0$ as Theorem 2 assumes. For each $k \in \{1, \ldots, 6\}$ and target $\varepsilon \in \{0, 0.25, 0.5, 1, 2\}$, we draw 8 random instances and compute the exact and approximate gradients by enumerating all 3^k action sequences, each weighted by its exact probability under the true policy. The ε-exponent is fitted at $k = 4$ and the k-exponent at target $\varepsilon = 1$, by least squares in log–log space. With $\varepsilon = 0$ the two gradients agree to within 10^{-7}. The observation-mismatch term of the bound is zero by construction.

A.15 Additional Training and Diagnostic Figures

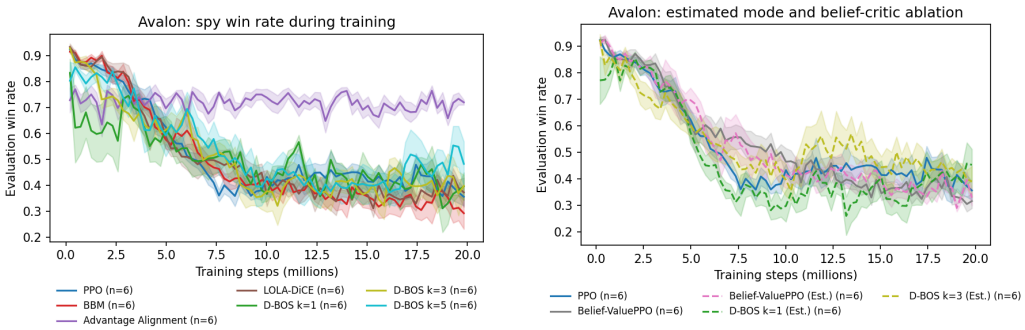
The main Avalon comparison in Figure 4 graphs use a 6-seed evaluation of PPO, BBM, LOLA-DiCE, Advantage Alignment and D-BOS and its ablations. Estimated-observation Avalon runs are also included in the graphs. We noticed a performance boost with using the estimated-observation mode possibly because observer-specific proxies give separate evidence for each resistance and spy player . However, Advantage Alignment, still outperforms us due to a richer signal and having access to opponent advantages.

Figure 5 showcases the training graphs for the Rescue-the-General environment for multiple different methods. We can see that D-BOS sustains performance for a longer period of time compared to baselines. All methods eventually drop in performance due to the environment dynamics, where blue ends up winning. We also notice a drop in performance proportional to the value of k as suggested by Theorem 2.

The BeliefCritic $V_\phi(o, \hat{B})$ explains 0.62-0.65 of the variance of realized returns in Avalon, 0.75–0.88 in Rescue-the-General, 0.45-0.46 in the belief-relay game, and 0.29-0.31 in the Coin Game (averaged over training and 6 seeds), so the endpoint value is informative about future return, most strongly in Avalon and Rescue-the-General.

A.16 Compute and Hyperparameters

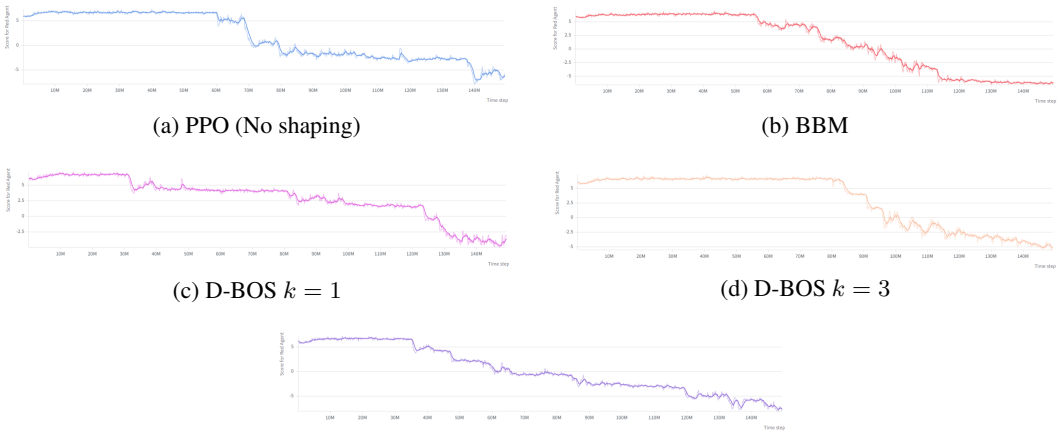
All methods within an environment use the same rollout budget, optimizer family, policy architecture, and seed set. RTG uses the CNN policy path with 64 parallel environments, rollout length 16, PPO



(a) Comparison with baselines

(b) Estimated-observation D-BOS and the Belief-ValuePPO ablation.

Figure 4: Avalon spy win rate on evaluation episodes during training (mean $\pm$ standard error over 6 seeds), the same metric as Table 3.



(a) PPO (No shaping)

(b) BBM

(c) D-BOS $k = 1$

(d) D-BOS $k = 3$

(e) D-BOS $k = 5$

Figure 5: Rescue-the-General Training graphs over time. We observe that D-BOS with $k = 1$ sustains performance significantly longer than baseline methods before the characteristic performance drop at 120M steps.

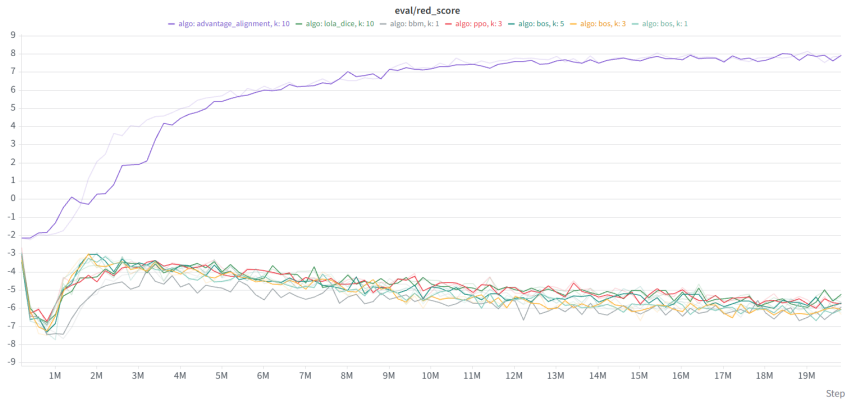


Figure 6: Multi-Agent Coin Game red-agent evaluation return over training (6 seeds). D-BOS keeps observers' beliefs near chance (Fig. 9) but does not improve return over PPO when roles resample every episode.

learning rate 2.5×10^{-4}, discount $\gamma = 0.99$, GAE $\lambda = 0.95$, and checkpoints/evaluation points every 10M environment steps. The main RTG runs use 150 million environment steps per seed. Avalon5 and CoinGame use the MLP policy path with 16 parallel environments, rollout length 32, two PPO epochs, two minibatches, learning rate 5×10^{-4}, discount $\gamma = 0.99$, GAE $\lambda = 0.95$, and a 20M-step training budget. Avalon uses a hidden dimension 128 and an entropy coefficient of 0.02; CoinGame uses hidden dimension of 64 and an entropy coefficient of 0.01.

We ran the experiments on A100-40GB, A100-80GB, and A30 GPUs. RTG was the most expensive setting, taking roughly two days per method/seed at the final budget. Avalon and CoinGame runs were substantially lighter, typically completing overnight for each method/seed, depending on the queue and GPU type. The dominant compute difference is across environments rather than across PPO, BBM, and D-BOS within the same environment.

Shaping-strength settings. The scalar shaping coefficient is not directly comparable across BBM and D-BOS. BBM scales an auxiliary belief-manipulation reward, while D-BOS scales a belief-gradient correction. We therefore select the shaping strength separately for each shaping mechanism while keeping the environment, architecture, optimizer, rollout budget, and seeds fixed. In Avalon, BBM uses $\lambda = 0.5$, matching the setting reported in the BBM reference paper [10] and the best stable value from our quick validation runs. The final Avalon D-BOS runs use $\lambda = 1$, which gave the most reliable behavior in our tuning runs. In CoinGame, both BBM and D-BOS use $\lambda = 0.5$; in RTG, D-BOS and BBM both use $\lambda = 0.5$. These coefficients should be interpreted as method-specific tuning parameters rather than matched parameters

Belief-relay game. The belief-relay game is a minimal three-agent signaling game in which observers' beliefs are exactly computable, so belief trajectories can be inspected directly. A shaper has a hidden type $z \in \{0, 1, 2, 3\}$, resampled at the start of every block of $k = 3$ rounds (5 blocks per episode); types $\{0, 1\}$ and $\{2, 3\}$ form two guilds. Each round the shaper emits one of 9 joint signals, which is decoded into two 3-symbol channels: the ally observes only the first and the adversary only the second, so the two observers receive different evidence about z. Each observer updates a Bayesian posterior over z from its own channel, using its own learned model of the shaper's signaling policy, fit by maximum likelihood on past rollouts; this posterior is part of the observer's observation. On the last round of each block, the ally chooses a venture to invest in and the adversary a venture to raid. A correct investment (matching the shaper's guild) gives $+1$ to both the ally and the shaper, and a correct raid gives $+1$ to the adversary and -1 to the shaper; incorrect choices give 0. Ally and adversary beliefs start identical (uniform) in every block.

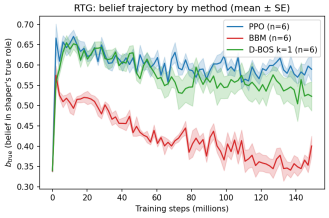


Figure 7: RTG: observers' belief in the shaper's true role. BBM's per-step deception bonus lowers it far more than PPO or D-BOS ($k = 1$).

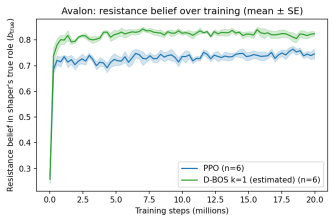


Figure 8: Avalon: resistance observers' belief in the shaper's true role, PPO vs. D-BOS ($k = 1$, Est.). It is higher under D-BOS.

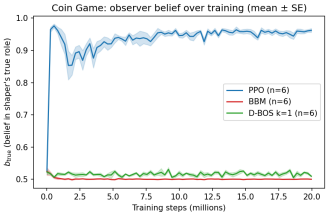


Figure 9: Coin Game: under PPO observers become nearly certain of the shaper's role, while under D-BOS and BBM their belief stays near chance.

Figure 10: Belief trajectories over training (mean ± SE over 6 seeds).

A.17 Broader Impact and Societal Considerations

D-BOS provides agents with the ability to influence what others believe. While we study this capability in controlled game environments, the underlying mechanism of optimizing actions to shape observer posteriors has a dual-use potential. On the positive side, belief-aware agents can improve coordination in human-robot teaming, make autonomous systems more legible to human partners,

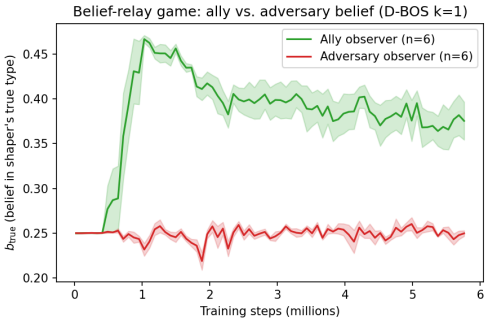


Figure 11: Belief-relay game: D-BOS ($k = 1$) shapes the ally's and the adversary's beliefs differently. Both start at the uniform prior (0.25); the shaper learns to send guild information on the ally's channel while keeping the adversary's uninformative, so the ally's belief in the true type rises toward the 0.5 ceiling reached when only the guild is revealed, while the adversary's stays near chance. This separation emerges purely from task reward (mean $\pm$ SE over 6 seeds). However, PPO also shows some similar separation.

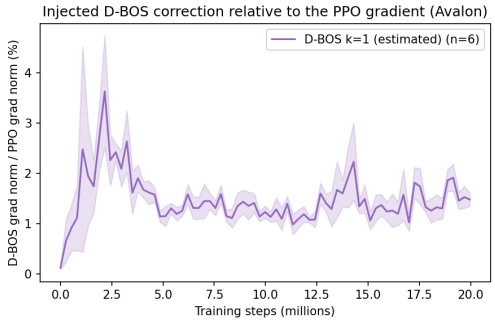


Figure 12: Norm of the injected D-BOS correction relative to the PPO gradient during training (Avalon, D-BOS $k = 1$ Est., mean $\pm$ SE over 6 seeds). After early training it stays around 1–2%.

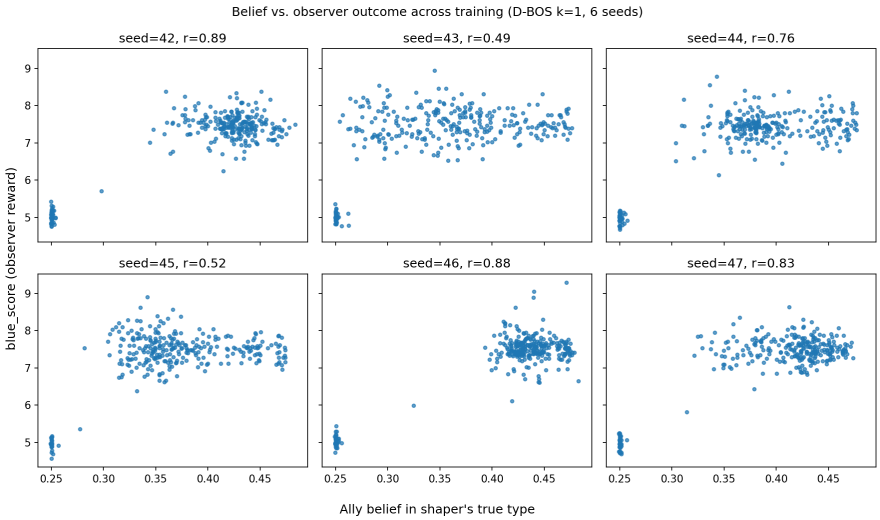


Figure 13: Belief-relay game: the ally's belief in the shaper's true type against the observers' score over training, one panel per seed (D-BOS, $k = 1$). The two are positively correlated in every seed ($r = 0.49$-0.89), largely reflecting the early-training transition in which the ally starts investing correctly.

and enable more effective communication under information asymmetry. On the negative side, the same mechanism could in principle be applied to train agents that systematically deceive human users or manipulate beliefs in adversarial settings. We note that D-BOS operates within a fixed game structure with known role sets and does not generalize to open-ended deception in natural language or unconstrained environments. Nevertheless, as belief-shaping methods mature, careful attention to deployment contexts and the development of counter-deception methods will be important. While we do test in an adversarial setting, our goal is to eventually use our method for better cooperation amongst humans and robots and do not condone the misuse of belief-based opponent shaping.

A.18 LLM Usage

We used large language models as a writing and editing aid during manuscript preparation, including assistance with prose drafting, LaTeX formatting, and proof presentation and formatting. All technical content, mathematical derivations, experimental design, implementation, and scientific claims are the original work of the authors. We referred to some help from LLMs in implementing and refining our codebase, but LLMs were not an important part of the research, nor did they contribute any technical details integral to our methods.

NeurIPS Paper Checklist

1. **Claims**
   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?
   Answer: [Yes]
   Justification: We thoroughly analyze D-BOS in multi-agent hidden role games and provide theoretical analysis for Theory of Mind Approximations. We also analyze our method against belief manipulation baselines as well as opponent shaping baselines in hidden role games.
   Guidelines:
   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.
2. **Limitations**
   Question: Does the paper discuss the limitations of the work performed by the authors?
   Answer: [Yes]
   Justification: We analyze limitations and future work in Section 7
   Guidelines:
   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.
3. **Theory assumptions and proofs**
   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?
   Answer: [Yes]
   Justification: We provide theoretical results and assumptions in the main paper with proofs and other technicalities in the appendix
   Guidelines:
   - The answer [N/A] means that the paper does not include theoretical results.

- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**
   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?
   Answer: [Yes]
   Justification: We release our code at `https://github.com/skyguy-netizen/d-bos`
   Guidelines:
   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**
   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?
   Answer: [Yes]
   Justification: We release our code at `https://github.com/skyguy-netizen/d-bos`
   Guidelines:
   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not

including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**
   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?
   Answer: [Yes]
   Justification: All training details are specified in the Appendix
   Guidelines:
   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.
7. **Experiment statistical significance**
   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?
   Answer: [Yes]
   Justification: All standard error across seeds reported wherever applicable and define statistical significance testing.
   Guidelines:
   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.
8. **Experiments compute resources**
   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?
   Answer: [Yes]
   Justification: We specify compute details, and time required in our runs along with hyperparameter details in Appendix A.16

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We adhere to the NeurIPS Code of Ethics as detailed in the link

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We motivate our problem with its impacts, and include its potential and negative societal impacts in Appendix A.17

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: N/A

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**
    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?
    Answer: [Yes]
    Justification: We credit owners of any resource we use promptly
    Guidelines:
    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.
13. **New assets**
    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?
    Answer: [N/A]
    Justification: We do not have any new assets to disclose. However, we do implement our own version of well-known games that are linked in the code repository. Since they are recreations of well-known games, we do not claim them to be new assets that we introduce.
    Guidelines:
    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.
14. **Crowdsourcing and research with human subjects**
    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?
    Answer: [N/A]
    Justification: N/A
    Guidelines:
    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: We outline our LLM Usage in A.18

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.