The Cost of Blind Confidence: Opponent Modeling under Imperfect Information

author names withheld

Under Review for NExT-Game 2026

Abstract

Opponent Modeling (OM) is a powerful framework in Multi-Agent Reinforcement Learning (MARL) to anticipate and adapt to the strategies of other agents. However, its success is highly dependent on the assumption of high-quality observations. In many real-world applications, agents must operate under imperfect information that can lead to inaccurate model representations. In this paper, we investigate the drawbacks of agents conditioning their policies on flawed opponent models that cause significant performance degradation compared to model-agnostic baselines. To address this, we introduce Strategy Weighting for Adaptive Policies (SWAP), a novel adaptive framework that treats strategy selection as an online learning problem. Employing the EXP4 algorithm, our agent treats a predictive OM-based policy and a robust conservative policy as competing experts, dynamically switching between them based on their observed performance. Our experimental results demonstrate the advantages of adopting a conservative approach when information is flawed and using predictive modeling when information is reliable, outperforming state-of-the-art methods in these critical scenarios.

Keywords: Multi-Agent Reinforcement Learning, Opponent Modeling, Multi-Armed Bandit, Imperfect Information, Noise

1. Introduction

Traditional Reinforcement Learning (RL) approaches can be extended to the multi-agent problem by treating concurrent agents as part of the non-stationarity of the environment [16]. Instead, Opponent Modeling (OM) methods seek to explicitly represent the policies or goals of other agents [1]. Leveraging expressive Deep Learning architectures, modern OM approaches achieved impressive levels of performance in both cooperative and competitive environments [6] [17].

However, the effectiveness of these models is highly dependent on the quality of the information they process [15]. In real-world applications, agents often operate under imperfect information characterized by noise, partial observability, or latency. We argue that existing OM methods frequently assume access to perfect information, leading to blind confidence in the robustness of their predictions. In fact, conditioning a policy on a flawed latent representation can result in behavior that is significantly worse than if the opponent were ignored entirely.

This work investigates the critical threshold where OM transitions from a strategic asset to a performance liability. Our preliminary analysis reveals that, in the presence of high epistemic uncertainty, models can significantly degrade policy performance. To mitigate this, we propose an uncertainty-aware switching mechanism, which frames strategy selection as an online learning problem. As the information becomes corrupted, this mechanism allows the agent to shift the proba-

bility mass from the ill-conditioned OM-based policy toward the more conservative expert. Through empirical evaluation, we demonstrate that this approach provides robust worst-case performance, allowing the agent to distrust its own model when necessary.

2. Related Work

Opponent modeling has evolved significantly from its foundations in classical Game Theory, where tools like Fictitious Play [3] and Minimax [14] were used to compute equilibrium solutions. As the field progressed, research diversified into specialized methodologies depending on the desired outcome. Many approaches based on recursive reasoning [17], role reconstruction [8], latent representation [12], and goal inference [18] emerged. In general, modern OM methods are based on high-dimensional feature extraction, using expressive neural architectures to encode agent trajectories. These advancements allow for highly adaptive behavior, yet many of them remain reliant on the assumption of accessing high-accuracy input information. This reliance introduces a critical vulnerability as the performance of these architectures is tied to the quality of the data they process. Although the problem is clear, very little research has been conducted in this area. Lazaric et al. [10] demonstrated that using an inaccurate opponent model can be more detrimental to performance than using no model at all. Overall, many modern frameworks prioritize architectural complexity over information integrity, leaving a significant gap in our understanding of when a model transitions from a strategic asset to a vulnerability.

3. Preliminaries

We consider our setting as a Partially Observable Stochastic Game (POSG) [5], defined as the tuple $(\mathcal{I}, \mathcal{S}, \{\mathcal{A}_i\}, \mathcal{P}, \{\mathcal{R}_i\}, \{\Omega_i\}, \mathcal{O}, \gamma)$, where $\mathcal{I} = \{1, 2 \ldots, n\}$ is the set of n agents, $\mathcal{S}$ is the state space, $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_n$ is the joint action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition function, $\mathcal{R}_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function for agent i, Ω_i is the observation space of agent i, $\mathcal{O} : \mathcal{S} \times \mathcal{I} \to \Delta(\Omega_i)$ is the observation function, and $\gamma \in \mathbb{R}$ is the discount factor. The goal of each agent is to maximize its discounted return:

$$G_i = \sum_{t=0}^{\infty} \gamma^t \mathcal{R}_i \left(s^t, a_i^t, a_{-i}^t\right),$$

where a_i^t is the action of agent i, a_{-i}^t is the joint action of all the other agents, both conditioned on the respective observations.

Opponent Modeling (OM) is a technique used to mitigate the non-stationarity of the Multi-Agent Reinforcement Learning (MARL) setting by learning a representation of the other agents' policies [1]. Let $h_i^t = \left(o_i^0, a_i^0, \ldots, o_i^t\right)$ be the local history of agent i.

The agent can map this history to a predicted policy $\hat{\pi}_{-i}^t$ or, more frequently, to a latent representation z_{-i}^t of the opponents' behavior. Then, this latent representation is used to condition an augmented policy $\pi_i(a_i^t | o_i^t, z_{-i}^t)$. While theoretically sound, the accuracy of z_{-i}^t is highly sensitive to the quality of o_t^i. Under imperfect information, the distribution $\mathcal{O}(s_t, i)$ may yield observations that result in a misaligned latent representation. This means that the perturbation in the observation space is propagated into the latent space, producing an inconsistent model of the opponent.

To address model unreliability, we frame strategy selection as a problem of learning from expert advice. We consider a set of K experts, where each expert k provides a recommendation in the form

of a probability distribution over actions, $\xi_k \in \Delta(\mathcal{A}_i)$. In our framework, these experts represent the OM-conditioned policy π_{OM} and a model-agnostic policy π_{AG}. Our implementations of the two policies will be detailed in the next section.

We employ the EXP4 algorithm [2], which at each step t updates a uniformly initialized weight vector $w_t \in \mathbb{R}^K$ based on the received reward r_i^t and the experts' predictions:

$$w_k^{t+1} = w_k^t e^{\left(\frac{\eta \hat{r}_i^t}{K}\right)},$$

where η is the learning rate and $\hat{r}_i^t$ is a reward estimator. This mechanism allows the agent to treat the trustworthiness of the opponent model as an online optimization task, guaranteeing low regret against the best-performing strategy in hindsight.

4. Methodology

Taking inspiration from Fu et al. [4], we introduce Strategy Weighting for Adaptive Policies (SWAP). Our architecture consists of three core components: a Conservative Policy (π_{AG}) providing a model-agnostic baseline, a predictive Greedy Policy (π_{OM}) that explicitly leverages Opponent Modeling (OM), and a Multi-Armed Bandit (MAB) that dynamically selects between them. The core of SWAP lies in the use of EXP4 [2] to choose between these two policies throughout the episode, updating the associated weights at each timestep based on the feedback received. This intra-episode switch improves over their per-episode approach, allowing more fine-grained control. Furthermore, while their method focuses on robustness against unknown opponents, lacking a known representation, our goal is to handle generic imperfect information, such as noisy observations. This allows the agent to adopt a more conservative behavior when the environment is less informative and to switch to a policy tailored to the opponent only when the information quality is sufficient for exploitation.

4.1. Conservative Policy

The Conservative Policy π_{AG} is trained using PPO [13]. We employ a two-stage training. In the first phase, the agent is pretrained using a self-play approach, in which the agent and the opponent take turns in their updates. In the second phase, the agent is further trained against a set of pretrained opponent policies Π_{TRAIN} (combining both deep learning models and human heuristics). The goal of this procedure is to develop a sufficiently robust policy against a wide variety of opponents.

4.2. Greedy Policy

The Greedy Policy π_{OM} is trained using PPO against the set of pretrained policies Π_{TRAIN}. Unlike π_{AG}, it explicitly includes an opponent modeling mechanism. A Variational Autoencoder (VAE) [9] is used to produce a latent representation of the observed agent that will condition the policy.

The VAE is trained separately from π_{OM} using a dataset generated through the interaction between π_{AG} and the pretrained opponent policies Π_{TRAIN}. Specifically, it receives as input the observation of the controlled agent o_i^t and its previous action a_i^{t-1}, and reconstructs the next action of the observed agent a_{-i}^t together with some additional information, depending on the environment.

The VAE encoder $\mathcal{E}_{\text{VAE}}$ includes an LSTM [7] to allow the model to process temporal sequences. In particular, the VAE operates on a temporal window of 5 time-steps. This design choice enables the model to capture short-term behavioral patterns of the opponent while avoiding overfitting to

longer and potentially noisier sequences. After training, the VAE decoder $\mathcal{D}_{\text{VAE}}$ is discarded, and only the encoder is used to produce the latent embedding z^t. Importantly, at inference time, the controlled agent does not have access to the opponent's observations and actions. Therefore, the use of the latent variable allows the model to perform opponent modeling under partial observability.

4.3. Strategy Selection via EXP4

Due to its structure, the Conservative Policy π_{AG} is more suitable against previously unseen opponents and when available information is deceptive, while the Greedy π_{OM} tends to achieve higher performance against already encountered opponents and in environments with accurate information. To navigate this trade-off, we employ the EXP4 framework to treat these policies as experts. At each time step t, the algorithm computes a weighted mixture of their predictions, $\xi_{\text{AG}}(o_i^t) = \pi_{\text{AG}}(\cdot|o_i^t)$ and $\xi_{\text{OM}}(o_i^t) = \pi_{\text{OM}}(\cdot|o_i^t)$, to produce the final state-conditioned action distribution:

$$\xi_{\text{SWAP}}(o_i^t) = \frac{w_{\text{AG}}^t \cdot \xi_{\text{AG}}(o_i^t) + w_{\text{OM}}^t \cdot \xi_{\text{OM}}(o_i^t)}{w_{\text{AG}}^t + w_{\text{OM}}^t},$$

where w_{AG}^t and w_{OM}^t are the associated weights. The weights are updated exponentially based on the importance-weighted reward estimates of each policy. This allows the agent to dynamically shift towards the policy that best matches the current environmental conditions.

5. Experiments

In this section, we evaluate the robustness of standard OM frameworks against imperfect information environments and compare them to our proposed solution. We first analyze the performance degradation of two baseline methods, LIAM [12] and GSCU [4], when subjected to noisy observations in both cooperative and competitive environments. Then, we demonstrate how SWAP maintains more stable performance under these same conditions.

5.1. Experimental Setup

As previously mentioned, our method is compared with:

- *LIAM*, a classic OM architecture that uses a latent representation to condition the policy. This latent vector is obtained from the current trajectory using a recurrent autoencoder.
- *GSCU*, a more recent method that uses EXP3 [2] to choose between an OM-based policy and a conservative one. This switch depends on whether the opponent's policy was encountered during the training phase.

We use two distinct environments, taken from the Multi Particle Environments 2 (MPE2) library [11], to cover both cooperative and competitive scenarios:

- *Double Speaker Listener* (DSL), named Simple Reference in MPE2, is a cooperative task where the controlled agent must communicate with a partner to reach a landmark. To interpret the advice, the agent must infer the partner's language based on the messages received.
- *Predator Prey* (PP), named Simple Tag in MPE2, is a competitive environment where the controlled agent acts as the prey. The objective is to escape predators while staying within the boundaries of the environment.

Table 1: DSL Environment Results

		Seen		Unseen	
Mod.	**Noise**	**Avg.**	**Δ**	**Avg.**	**Δ**
LIAM	None	-10.2	–	-24.7	–
	Random	-14.9	-46%	-23.0	+7%
	Gaussian	-20.1	-96%	-21.9	+12%
GSCU	None	-15.3	–	-25.8	–
	Random	-16.8	-10%	-23.8	+8%
	Gaussian	-17.3	-13%	-24.0	+7%
SWAP	None	-17.9	–	-23.5	–
	Random	-18.9	**-6%**	-22.6	**+4%**
	Gaussian	-19.4	**-8%**	-22.1	**+6%**

Table 2: PP Environment Results

		Seen		Unseen	
Mod.	**Noise**	**Avg.**	**Δ**	**Avg.**	**Δ**
LIAM	None	-9.0	–	-14.6	–
	Gaussian	-34.5	-283%	-38.3	-162%
	Opponent Pos.	-50.6	-462%	-51.3	-251%
GSCU	None	-26.1	–	-32.0	–
	Gaussian	-27.2	-4%	-32.5	-2%
	Opponent Pos.	-31.0	-19%	-34.2	-7%
SWAP	None	-14.7	–	-23.1	–
	Gaussian	-17.9	**-22%**	-26.8	**-16%**
	Opponent Pos.	-27.6	**-88%**	-34.1	**-48%**

5.2. Cooperative Environment: Double Speaker Listener

In the Double Speaker Listener (DSL) environment, an agent must reach a specific landmark by interpreting guidance from a partner. Because different partners utilize distinct languages, the agent is required to accurately model its opponent to decode advice correctly. To evaluate the resilience of these models, we introduce two forms of noise to the observations and messages. First, we implement *Random Messages*, where the correct message is replaced by a random one with a 50% probability to determine if the model can effectively ignore meaningless input. Second, we implement *Global Gaussian Noise* by adding values sampled from $\mathcal{N}(0, 0.5)$ to the entire observation vector, allowing us to evaluate general sensitivity. Results are summarized in Table 1, including performance against opponents encountered during training (*Seen*) and novel opponents (*Unseen*).

5.2.1. RESULTS ANALYSIS

The experimental results reveal how different opponent modeling architectures manage observation uncertainty. While LIAM achieves the highest performance in noise-free environments with known opponents, it exhibits a clear fragility when subjected to perturbations. Specifically, LIAM's episodic return drops by 46% under communication noise and by 96% under observation noise. This suggests that LIAM's recurrent autoencoder attempts to interpret noise as a meaningful signal, leading to highly unpredictable and incorrect navigation behavior.

GSCU demonstrates better inherent stability than LIAM, due to its architecture, limiting the performance drop to 13%. However, our method proves to be the least sensitive architecture. Under the same noise intensity, it keeps a steadier return with a maximum 8% degradation.

Interestingly, all models show a slight increase in return under noisy conditions when facing unseen partners compared to their respective noise-free baselines. As the noise-free performance is already quite low, the additional disturbances may not have much of an impact. Within this category, the performances are comparable; however, SWAP achieves the highest mean return.

5.3. Competitive Environment: Predator Prey

In the Predator Prey (PP) environment, the controlled agent acts as the prey, whose objective is to escape from a team of three predators. This competitive setting tests the model's ability to main-

tain safe distances from menaces while remaining within the environment's boundaries. To assess robustness, we introduce noise to the prey's observation vector under two specific conditions. First, we implement *Global Gaussian Noise* by adding values sampled from $\mathcal{N}(0, 0.1)$ to the entire observation vector to measure global sensitivity. Second, we implement *Opponent Position Noise* by applying the same distribution exclusively to the relative positions of the predators, as these represent critical features for an effective evasion strategy. Results are reported in Table 2.

5.3.1. RESULTS ANALYSIS

This study highlights the impact of spatial uncertainty in settings where precise movement is crucial. LIAM is highly sensitive to perturbations; its performance decreases significantly when facing both seen and unseen opponents, with a reward decrease of up to 462% due to an increase in collisions.

GSCU maintains the highest stability among the three models, with a maximum drop in return of 19%. However, because its noise-free returns are already suboptimal, the introduction of further perturbations does not substantially alter its qualitative behavior, making the degradation negligible.

Finally, our method provides an effective balance between noise-free efficiency and robust adaptation. In the absence of noise, SWAP achieves higher returns than GSCU. When noise is introduced, it shows a physiological reduction in return, more pronounced in the relative position case, while maintaining a performance that significantly exceeds that of LIAM under identical conditions.

5.4. Discussion of Global Robustness

The results across both the DSL and PP environments demonstrate that standard OM frameworks like LIAM are optimized for noise-free settings but lack an explicit mechanism to handle information uncertainty. In contrast, GSCU maintains stability but at the cost of significantly lower absolute performance, as its conservative nature prevents it from fully exploiting strategic opportunities.

SWAP, using its switching mechanism, retains the high-performance capabilities of opponent modeling when observations are accurate while effectively reverting to a robust baseline when noise is detected. This dual approach allows it to outperform GSCU in absolute terms while avoiding the performance collapses that characterize LIAM in imperfect information environments.

6. Conclusion

In this paper, we analyzed the impact of a model's blind confidence, demonstrating that adopting OM can be counterproductive when grounded in imperfect information. Our results revealed that conditioning a policy on flawed representations may result in performance that is even lower than that of model-agnostic baselines. Proposing SWAP, we showed that agents can mitigate this risk by adaptively retreating to a conservative policy whenever the model's performance proves unreliable.

Despite the robustness of this bandit-based approach, it currently faces a structural limitation, as it relies on immediate reward signals to evaluate expert performance. This requires high-density feedback during inference, limiting the framework's efficacy in environments with sparse rewards.

Our future work will focus on developing a more sophisticated gating mechanism that moves beyond reward-based switching. We aim to train a neural architecture that processes the agent's trajectory to predict which policy is best suited for the current information state. By leveraging these high-dimensional temporal features, this extension is expected to fluidly navigate the trade-off between strategic modeling and robust play, prioritizing the real-time reliability of information.

References

[1] Stefano V. Albrecht and Peter Stone. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 258:66–95, 2018. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2018.01.002. URL https://www.sciencedirect.com/science/article/pii/S0004370218300249.

[2] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002. doi: 10.1137/S0097539701398375. URL https://doi.org/10.1137/S0097539701398375.

[3] George W. Brown. Iterative solution of games by fictitious play. In T. C. Koopmans, editor, *Activity Analysis of Production and Allocation*. Wiley, New York, 1951.

[4] Haobo Fu, Ye Tian, Hongxiang Yu, Weiming Liu, Shuang Wu, Jiechao Xiong, Ying Wen, Kai Li, Junliang Xing, Qiang Fu, and Wei Yang. Greedy when sure and conservative when uncertain about the opponents. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 6829–6848. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/fu22b.html.

[5] Eric A Hansen, Daniel S Bernstein, and Shlomo Zilberstein. Dynamic programming for partially observable stochastic games. In *AAAI*, volume 4, pages 709–715, 2004.

[6] He He, Jordan Boyd-Graber, Kevin Kwok, and Hal Daumé, III. Opponent modeling in deep reinforcement learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1804–1813, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/he16.html.

[7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9 (8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

[8] Zican Hu, Zongzhang Zhang, Huaxiong Li, Chunlin Chen, Hongyu Ding, and Zhi Wang. Attention-guided contrastive role representations for multi-agent reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=LWmuPfEYhH.

[9] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. URL https://arxiv.org/abs/1312.6114.

[10] Alessandro Lazaric, Mario Quaresimale, and Marcello Restelli. On the usefulness of opponent modeling: the kuhn poker case study. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3*, pages 1345–1348, 2008.

[11] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Neural Information Processing Systems (NIPS)*, 2017.

[12] Georgios Papoudakis, Filippos Christianos, and Stefano Albrecht. Agent modelling under partial observability for deep reinforcement learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 19210–19222. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/a03caec56cd82478bf197475b48c05f9-Paper.pdf.

[13] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

[14] John Von Neumann and Oskar Morgenstern. Theory of games and economic behavior: 60th anniversary commemorative edition. In *Theory of games and economic behavior*. Princeton university press, 2007.

[15] Likun Yang, Pei Xu, Shiyue Cao, Yongjian Ren, Xiaotang Chen, and Kaiqi Huang. Uncertainty-aware opponent modeling for deep reinforcement learning. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, pages 2217–2225, 2025.

[16] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and YI WU. The surprising effectiveness of ppo in cooperative multi-agent games. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 24611–24624. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9c1535a02f0ce079433344e14d910597-Paper-Datasets_and_Benchmarks.pdf.

[17] XiaoPeng Yu, Jiechuan Jiang, Wanpeng Zhang, Haobin Jiang, and Zongqing Lu. Model-based opponent modeling. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 28208–28221. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b528459c99e929718a7d7e1697253d7f-Paper-Conference.pdf.

[18] Xiaopeng Yu, Jiechuan Jiang, and Zongqing Lu. Opponent modeling based on subgoal inference. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 60531–60555. Curran Associates, Inc., 2024. doi: 10.52202/079017-1936. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/6fb9ea5197c0b8ece8a64220fb82cdfe-Paper-Conference.pdf.