ABSTRACT

Training multiple agents to coordinate is an essential problem with applications in robotics, game theory, economics, and social sciences. However, most existing Multi-Agent Reinforcement Learning (MARL) methods are online and thus impractical for real-world applications in which collecting new interactions is costly or dangerous. While these algorithms should leverage offline data when available, doing so gives rise to what we call the offline coordination problem. Specifically, we identify and formalize the strategy agreement (SA) and the strategy fine-tuning (SFT) coordination challenges, two issues at which current offline MARL algorithms fail. Concretely, we reveal that the prevalent model-free methods are severely deficient and cannot handle coordination-intensive offline multi-agent tasks in either toy or MuJoCo domains. To address this setback, we emphasize the importance of inter-agent interactions and propose the very first model-based offline MARL method. Our resulting algorithm, Model-based Offline Multi-Agent Proximal Policy Optimization (MOMA-PPO) generates synthetic interaction data and enables agents to converge on a strategy while fine-tuning their policies accordingly. This simple model-based solution solves the coordination-intensive offline tasks, significantly outperforming the prevalent model-free methods even under severe partial observability and with learned world models.

KEYWORDS

Multi-Agent Learning; Coordination; Offline Reinforcement Learning; Model-Based Reinforcement Learning; Deep Learning; World Models

ACM Reference Format:


This work is licensed under a Creative Commons Attribution International 4.0 License.

1 INTRODUCTION

Multi-agent problems are ubiquitous in real-world scenarios, including traffic control, distributed energy management, multi-robot coordination, auctions, marketplaces, and social networks [6, 8, 10, 13, 15, 16, 21, 25, 42, 44, 53, 58]. This makes the development of efficient multi-agent algorithms a crucial research area in artificial intelligence and machine learning with substantial implications in various fields including robotics, game theory, economics, and social sciences [35, 37, 51, 60, 68]. However, existing methods are mostly online and require interacting with the environment throughout learning which often makes them costly or even dangerous for real-world applications [34]. In contrast, offline Reinforcement Learning (offline RL) obviates the need for interactions with the environment as it allows learning from existing datasets that do not have to be collected by experts. It is therefore well suited to tasks where: a) we cannot afford to materialize the situation in practice, b) it is unfeasible to build a simulator, and c) there exist datasets of realizations collected by experts. It is therefore well suited to tasks where: a) we cannot afford to materialize the situation in practice, b) it is unfeasible to build a simulator, and c) there exist datasets of realizations collected by experts. Consequently, we hypothesize that offline multi-agent approaches will be key for tackling real-world multi-agent problems. Let us imagine for instance trying to understand how autonomous actors (i.e., governments, international organizations, industries, etc.) must maneuver to reduce the severity of a worldwide pandemic while preventing economic collapse. It goes without saying that: a) starting pandemics is not a viable way to gain real-world practice; b) building a simulator is a colossal task that would suggest emulating our society and its economy; and c) the impact of past decisions (such as implementing lockdown policies, travel restrictions, and vaccination campaigns) on the unfolding of the pandemic and the economy is well-documented. Hopefully, these records can be used to derive new strategies in the future.

The offline coordination challenge. In general, actors such as individuals, organizations, robots, software processes, or cars are self-governed and ultimately act autonomously. However, during the offline learning phase, it is reasonable to assume that learners can share pieces of information, which makes the Centralized Training Decentralized Execution (CTDE) formulation a natural approach for the offline training regime. Unfortunately, as shown in this work, simply combining CTDE MARL and offline RL methods on a dataset of multi-agent interactions does not necessarily yield policies that perform well together. This is what we call the offline
We refer to this coordination challenge as Strategy Agreement (SA) which may only vary on how they break symmetries present in the environment (i.e., which global strategy is being chosen and how individual behaviors blend together). Therefore, it is difficult for offline learners to coordinate. Figure 1 illustrates this issue that has been overlooked by the offline MARL community; showing that current methods, which are all model-free, fail at offline coordination even in simple environments.

In this work, we propose that multi-agent coordination can be decomposed into two distinct challenges. First, since most multi-agent tasks allow multiple optimal team strategies [5] – some of which may only vary on how they break symmetries present in the task [24] – agents must collectively select one strategy over another such that they individually converge toward coherent behaviors. We refer to this coordination challenge as Strategy Agreement (SA).

Additionally, for a chosen team strategy, agents have to precisely calibrate and adjust their behaviors to one another in order to reach the corresponding optimal group behavior. We call this Strategy Fine-Tuning (SFT). During online learning, agents continuously interact together in the environment, therefore, changes due to one agent’s local optimization directly impact other agents’ experiences and teammates are able to adapt: coordination occurs through interactive trial and error. Conversely, when learning from a fixed dataset of interactions, agents cannot interact with other agents to generate new data that explicitly measures the outcomes of current policies in the environment (i.e., which global strategy is being chosen and how individual behaviors blend together). Therefore, it is difficult for offline learners to coordinate. Figure 1 illustrates this in policy space: (a) During online learning, agents continuously interact to collect up-to-date data that inform the optimization of both individual policies and team strategy. This allows agents to converge to a global optimum by co-adapting and improving on each other’s updates. (b) In the offline setting however, agents must independently decide towards which of the two optima they aim to converge (i.e., strategy agreement between \( \pi^* \)). Assuming they pick \( \pi^a \), they must then derive their corresponding optimal policy (i.e., strategy fine-tuning toward \( \pi^* \), \( i = 1, 2 \)). Crucially, offline agents must rely only on interactions present in the dataset (corresponding to policies \( \pi^D \), \( i = 1, 2 \)), thus likely without interactions corresponding to their past or current policies.

Figure 1: Comparing online learning and offline learning in policy space to illustrate the offline coordination problem. (a) During online learning, agents continuously interact using their current policies and collect new data that informs the next update of the co-evolution from \((\pi^0_1, \pi^0_2)\) to \((\pi^a_1, \pi^a_2)\). (b) During offline learning, agents cannot collect new data and thus, they can only estimate updates from the dataset of interactions (here collected with \( \pi^D_1 \) and \( \pi^D_2 \)). To reach an optimal strategy agents must (1) agree on which optimum to target between \( *^a \) or \( *^b \), – i.e. solve strategy agreement –, and (2) respectively derive the policy corresponding to that strategy \((\pi^*_1, \pi^*_2)\), – i.e. solve strategy fine-tuning.

Current methods [27, 43, 64] deal with offline MARL by simply extending single-agent offline RL to the multi-agent setting. To do so, they either consider that agents are independent learners, or they leverage the CTDE paradigm. Provided that the dataset has enough coverage, it is in theory possible for CTDE agents to learn the different optimal value functions and policies. Yet, as we will see in Section 6, agents may still fail to agree on which team strategy to pick. We highlight such offline coordination failure in an offline version of the well-established iterated Coordination Game [5].

The crux of offline MARL is therefore that learners cannot interact together in the environment to collect new data that measures how individual policies blend together and what are the outcomes of the current team strategy. This relates to the absence of interactions during learning and is not a centralization issue. Thus, we motivate the need for generating synthetic data that measure the outcomes of the team’s current strategy to allow for coordinated agents.

We propose MOMA-PPO, a simple model-based approach that generates synthetic interactions. We show that it allows for offline coordination even in complex Multi-Agent MuJoCo (MAMuJoCo) [45] tasks with partial observability and learned world model. Across tasks and domains, MOMA-PPO significantly outperforms the offline MARL baselines. Our contributions are:

- identifying and defining the offline coordination problem, an issue that has been overlooked by the offline MARL community;
- proposing benchmarks in the form of new datasets and partially observable multi-agent tasks that test for offline coordination;
- showing that current methods, which are all model-free, fail at offline coordination even in simple environments;
- suggesting a link between the offline coordination problem and the lack of inter-agent interactions during learning that is inherent to model-free offline approaches;
- proposing to address the coordination problem with the first-ever model-based offline MARL method;
- experimentally validating the benefits of coordinating by interacting through a world model.
2 BACKGROUND

We consider the formalism of Dec-POMDP [41] with states $s_t \in S$. There are $i = 1, ..., N_\text{A}$ agents that partially observe this state from their stochastic observation function $s'_t \sim \mathcal{O}(s_t)\pi_t(a_t)$ and choose action $a'_t \in \mathcal{A}_i$ at every time step. Each agent has an action-observation history given by $h_t^i = \{s_0, a_0, ..., s_t, a_t\}$ and picks its action $a'_t$ using its stochastic policy $\pi_t(a_t|h_t^i)$. The environment’s transition depends on the joint action $a_t$ such that $s_{t+1} \sim \mathcal{P}(s'|s_t, a_t)$. The game is fully cooperative so all agents receive the same reward and have a single absorbing state $s_\text{abs}$.

3.1 Learning a centralized world model

In this work, we propose MOMA-PPO, a Dyna-like [55] model-based approach to multi-agent CTDE offline learning that relies on PPO [50]. The method can be decomposed into two steps: 1) learning a world model from the dataset, and 2) using the world model to train the agents’ policies.

3.2 Model-based Offline Multi-Agent PPO (MOMA-PPO)

Once $M$ has been trained on the offline dataset $\mathcal{D}$, it can be used to train online reinforcement learning algorithms in a Dyna-like manner. Here, we use MAPPO, a CTDE multi-agent version of PPO [65].

The synthetic data used to train the PPO policies is collected by sampling states from the offline dataset $\mathcal{D}$ and using the current policies $\pi_t$ alongside the world model $M$ to generate PPO’s training rollouts of size $k$. Terminating a rollout when the world model uncertainty exceeds $\lambda_e$ allows for adaptive rollout length and avoids training the current global state $s$ and the joint action $a$. The parameters are learned from the offline dataset $\mathcal{D}$ using Gaussian negative log-likelihood for $N(\mu_r, \sigma^2_r)$ and $N(\mu_p, \sigma^2_p)$, and binary cross-entropy for Bern($\rho$).

We also estimate the epistemic uncertainty of the reward using the variance of the predicted rewards across the ensemble:

$$\epsilon_r = \frac{\sum_{m=1}^{N} (\hat{r}_m - \bar{r})^2}{N - 1}, \quad \bar{r} = \frac{\sum_{m=1}^{N} \hat{r}_m}{N}.$$
policies on unfeasible data. Rollouts are always terminated with a
timeout mask $\zeta_t$ such that:

$$
\zeta_t = 1 - I(t = k - 1 \cup \epsilon_g \geq \ell_e),
$$

where $I$ is the indicator function. Note that the timeout mask $\zeta_t$ is
different from the model-predicted mask $\hat{f}_t$ which is solely related
to the environment and indicates terminal absorbing states that
occur for instance when the agent dies or reaches the goal. On the
PPO side, we adapt the generalized advantage estimation [49] to
account for the timeouts $\zeta_t$ and ensure that there is no accumulation
across rollouts while computing returns. We use the value of the
last state as an estimation of the return-to-go (see Appendix A).

The generation of length $k = 3$ rollouts is illustrated in Figure 2
where it can be seen that a rollout is interrupted early because the
world model was unreliable when generating the associated state
(shown in red).

In addition to the adaptive rollout length, the model epistemic
uncertainty is used to penalize the agents so they avoid exploiting
the world model in poorly reconstructed regions of the state space.

The final uncertainty-penalized reward is given by:

$$
r_f = r_t - \lambda_s \epsilon_r - \lambda_g \epsilon_g,
$$

where $\epsilon_r$ and $\epsilon_g$ are hyperparameters that weigh the severity of the penalty.

**Algorithm 1**

Require: offline dataset $D$, world model $M$, rollout horizon $k$, rollout
batch size $b$, uncertainty penalty coefficients $\lambda_s$ and $\lambda_g$, uncertainty
threshold $\ell_e$, MAPPO agents. Initialize MAPPO policies $\pi_i$ and value function $V$.

for epoch 1, 2, ... do

Generate synthetic data

for $b$ (in parallel) do

Sample history $h_t = (h^i_j)_{j=1}^\omega$ from $D$.

for $j = t, t+1, \ldots, t+k-1$ do

Sample $a^i_j \sim \pi_i(a^i_t | h^i_t) \forall i$.

Sample $s_{j+1}, f_j, f_{j+1}, \epsilon_r, \epsilon_g$.

Compute $f_j = f_t - \lambda_s \epsilon_r - \lambda_g \epsilon_g$.

Compute $\zeta_j = 1 - I(j = t + k - 1 \cup \epsilon_g \geq \ell_e)$.

Add sample $(h^i_j, a^i_j, f^i_j, f^i_{j+1}, s^i_{j+1})$ to $\mathcal{R}$.

Get $h^i_{j+1}$ from $h^i_j$ and $s^i_{j+1}$. Set $s^i_{j+1} = s^i_{j+1}$.

Train agents

Use synthetic rollouts in $\mathcal{R}$ to train policies $\pi_i$ and value
function $V$ with MAPPO.

return multi-agent policies $\pi_i$.

Practical considerations and limitations. As stated in Sec-
tion 2, we assume CTDE and that the global state $s_t$ fully observes
the environment, therefore we do not equip the world model with
memory. The agents, on the other hand, only have access to par-
tial observations and must rely on action-observation histories $h^i_t$.
In practice, we restrict action-observation histories to 10 steps in
the past (either from the dataset or from the generated rollouts) and
process them with one layer of self-attention [59] followed by
one layer of soft-attention [1]. The resulting embeddings are
concatenated to the agent’s current state, and for simplicity, we
abuse notation by denoting this “memory enhanced” state with $h^i_t$.

In MAPPO our centralized value function uses the QMIX value-
decomposition [47] with $w^i(s_t)$ and $b(s_t)$ provided by a learnable
neural network: $V(s_t) = \sum_i w^i(s_t) V^i(h^i_t) + b(s_t)$.

Finally, it is important to note that the task of the MOMA-PPO
agents is quite different from the task of the agents that generated
the dataset. First, the MOMA-PPO agents’ initial state distribution
is now the dataset’s state distribution. Then the reward of the task
has been altered to account for the model uncertainty. Last but
not least, agents are only allowed to stray $k$ steps away from the
dataset’s coverage. While this restriction mitigates world model
abuse, it can also prevent the agents from discovering goals that
are further away from the offline data. Our resulting model-based
offline multi-agent method is illustrated in Algorithm 1 and more
details are provided in Appendix A.

4 RELATED WORK

Multi-agent coordination. Coordination has been a challenge
of interest since the early works on cooperative MARL [5, 7, 11, 12,
36] and has consistently been a central focus of the multi-agent
literature [26, 33, 38, 69]. While different works consider different
aspects of coordination – such as behavior predictability and syn-
chronous sub-policy selection [48], structured team exploration
[40] or the emergence of communication and cooperative guiding
[2, 32, 63] – our definition of coordination is closest to the seminal
work of [5]. Indeed, we consider coordination in terms of agents
agreeing to individually follow the same team strategy (that is a
policy over joint actions) and finetuning their behavior to one an-
other in tasks where multiple distinct optimal team strategies exist.

A similar notion of coordination has been used in the zero-shot co-
ordination problem investigated by [24] where agents are trained so
that they are able to perform with agents they have never seen be-
fore. Yet, while their focus is on deriving standardized coordinated
strategies that can generalize to unseen teammates, coordination is
still learned through online interactions.

Coordination with teammates without direct interactions is often
referred to as ad-hoc coordination [4, 54]. Recent works assume
access to offline demonstrations of the teammates’ behaviors. These
can be used to guide the agent’s self-play training toward adopting
the appropriate equilibrium (or “social conventions”) of its future
teammates [33, 57]. Similarly, [9] uses offline data to learn a model of
the teammate’s behavior and use it to train the agent to coordinate
with that ally. In ad-hoc coordination, teammates’ behaviors are
fixed and can be estimated a priori from the dataset: the learner
merely has to identify the team strategy and adopt it. Conversely,
in our offline coordination setting, all the agents are learning and
therefore have unknown changing behaviors: they must identify
the different potential team strategies and agree on which one to
follow (strategy agreement). Simultaneously, they must finetune
their behaviors to one another in order to reach this team policy
(strategy fine-tuning). All this without being able to interact with
other agents or the environment.

Offline MARL. Recent works have been investigating offline
solutions to the MARL problem. All of these methods build on
model-free single-agent approaches and constrain the policy to stay
in the dataset’s distribution by using either SARSA-like schemes
which use off-policy algorithms, MOMA-PPO is based on MAPPO, with timeouts rather than with MOReL’s terminal states meaning (such as ICQ [64] and IQL [30]) or policy regularization (such as CQL [31] and TD3+BC [19]).

Some methods investigate specific modifications to improve performance in the multi-agent setting. For instance, in the decentralized setting, MABCQ [27] enforces an optimistic importance sampling modification that assumes that independent agents will strive toward similar high-rewarding states, yet since this does not discriminate between which high-rewarding state to favor, the strategy agreement issue remains. For discrete actions spaces problems, [56] propose a Transformer-based approach that learns a centralized teacher and distills its policy into independent student policies. Finally, OMAR [43] proposes to alleviate miscoordination failure in offline MARL by adding a zeroth order optimization method on top of multi-agent CQL, achieving state-of-the-art performance on a variety of tasks. We share these works’ goal of learning coordinated and efficient multi-agent teams in the offline setting. Yet, we believe that interacting learners and agents are essential to coordination and therefore take a different approach by focusing on model-based methods rather than model-free ones.

**Offline model-based RL.** Model-based approaches have been investigated in the single-agent offline RL setting. Notoriously, MOPO [67] proposed to learn an ensemble-based world model and use it to generate rollouts from the offline dataset to train a SAC agent [23]. They also proposed an uncertainty-based reward penalty to prevent the learner from exploiting the model. MOReL [28] takes a similar approach but prevents model abuse by learning a pessimistic MDP in which states that are outside of the dataset coverage become absorbing terminal states. COMBO [66] proposed a similar but more conservative version of MOPO by using CQL instead of SAC and learning on both generated and dataset’s states. Finally, ROMI [61] also uses model-free offline RL to derive a policy from a model-based augmented offline dataset, yet they enforce additional conservatism by learning a reverse policy and dynamics model to generate rollouts that lead to target states contained in the dataset. This mitigates against generating rollouts outside of the dataset’s coverage.

We believe that offline RL algorithms are ill-suited to learn on non-stationary data such as the one generated by updating policies be it in a world model or in a real environment. Therefore, our method MOMA-PPO uses an online RL algorithm instead of an offline one. Additionally, to enforce conservatism and avoid world model exploitation we use both uncertainty penalty and early rollout termination. In that sense, MOMA-PPO unifies what would be multi-agent extensions of MOPO and MOReL. Yet, instead of penalizing aleatoric uncertainty as in MOPO, we focus on epistemic uncertainty and estimate it by monitoring the coherence between the different models in the ensemble. Section 6.3 reports the benefits of using the epistemic uncertainty (due to the finite amount of training data) over the aleatoric uncertainty (from the environment’s stochasticity). Also, MOMA-PPO’s early rollout termination is done with timeouts rather than with MOReL’s terminal states meaning that it does not penalize agents while still avoiding using unfeasible rollouts to train them. Finally, in contrast with MOReL and MOPO which use off-policy algorithms, MOMA-PPO is based on MAPPO, an on-policy method that has reliably achieved state-of-the-art performance in MARL tasks [65]. It is therefore well suited to the multi-agent setting in which slight changes in a teammate’s policy can drastically impact the overall group behavior and quickly make previous interactions obsolete.

**Model-based multi-agent RL.** In the online setting, model-based approaches aim to improve sample efficiency by reducing the number of interactions with the environment. Therefore, these methods use off-policy schemes and focus on how and when to collect additional environment data for refining the world model and the policies [62, 70, 71]. In offline MARL, sample efficiency is not a consideration since the environment data has already been collected offline and additional samples are not an option. Yet, as we show here, model-based approaches can benefit multi-agent coordination in the offline setting by allowing multiple learners to interact through the world model.

### 5 BASELINES, ENVIRONMENTS, TASKS, AND DATASETS

![Environments illustrations.](image)

**Figure 3** Environments illustrations. (a) Pay off matrix of the Iterated Coordination Game. (b) Two-agent Reacher, red and blue agents respectively control the torque on $\theta_1$ and $\theta_2$. (c) Four-agent Ant, each agent controls a different limb (shown with different colors). In PO tasks, agents only observe the limb they control while the torso observations – in white – are available only to the yellow agent.

#### 5.1 Baselines

We compare with a large and varied array of baselines. First, we consider a centralized training and centralized execution approach by observing the global state and selecting the joint action. We use IQL [30], a state-of-the-art single-agent model-free offline RL algorithm for this setting. Considering centralized execution gives an upper bound on what can be achieved in terms of strategy agreement. Indeed, a single learner controls all the agents and can thus choose for the whole team the strategy to adopt.

Then, we extend IQL [30] to the multi-agent setting by using the QMIX value decomposition on the $Q$ and $V$ value networks. This gives MAIQL, a very competitive model-free CTDE offline MARL algorithm that should allow for fine-tuning with additional online data after training. We refer to the finetuned version as MAIQL-ft and follow IQL’s finetuning procedure [30]: MAIQL-ft is first trained to convergence on the offline data and then finetuned for the same number of training steps by progressively introducing additional interaction data. At the end of finetuning, the replay buffer contains as many offline interactions as online ones. For completeness, we also consider MATD3+BC the CTDE multi-agent version of [19] with QMIX.
Recent literature in model-free MARL [14, 39] and notably in the offline setting [43], advocates for decentralized value functions and independent learners. Therefore, we consider several independent learner approaches, starting with Independent Behavioral Cloning (IBC). Despite its simplicity, BC [46] produces surprisingly efficient baselines for Imitation Learning and offline RL [3, 52]. Finally, we consider the independent learners extensions to [31] and [19], respectively ICQL and ITD3+BC, as well as the state-of-the-art model-free offline MARL method, OMAR [43].

5.2 Offline Iterated Coordination Game

To illustrate the strategy agreement coordination challenge, we propose an offline version of the Iterated Coordination Game [5] presented in Figure 3 (a). Agents must pick the same direction in order to succeed and we investigate three offline datasets of interactions (see Table 1). In the most favorable dataset, data is collected by coordinated agents that select the same option of going right most of the time. In the less favorable setting, agent 1 goes left most of the time while agent 2 is more likely to go right. In the neutral setting, agents act uniformly. However, each dataset does contain both coordinated and uncoordinated behaviors in which agents simultaneously choose the same – respectively, different – directions.

It is therefore straightforward for a centralized critic to learn that \(Q(\rightarrow, \rightarrow) = Q(\leftarrow, \leftarrow) = 1\) while \(Q(\rightarrow, \leftarrow) = Q(\leftarrow, \rightarrow) = 0\), regardless of the specific dataset. Yet, decentralized actors remain unaware of whether they should go left or right since both strategies are equivalent and actors have no way of consistently picking one over the other.

**Table 1: Policies used to collect the datasets in the Iterated Coordination Game and resulting average scores of the datasets**

<table>
<thead>
<tr>
<th>Policies</th>
<th>(P(a^1 = \rightarrow))</th>
<th>(P(a^2 = \rightarrow))</th>
<th>Avg. Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>favorable</td>
<td>0.75</td>
<td>0.75</td>
<td>0.623</td>
</tr>
<tr>
<td>neutral</td>
<td>0.5</td>
<td>0.5</td>
<td>0.502</td>
</tr>
<tr>
<td>unfavorable</td>
<td>0.25</td>
<td>0.75</td>
<td>0.375</td>
</tr>
</tbody>
</table>

5.3 Offline MAMuJoCo

Building upon D4RL [18] and MAMuJoCo [45], we propose offline multi-agent continuous control tasks with various datasets and full or partial observability.

**Two-agent Reacher with a mixture-of-expert dataset.** To investigate strategy agreement in a more complex continuous control setting, we propose a two-agent version of the Reacher environment as shown in Figure 3 (b). The offline dataset is collected as follows: in the first stage, we train online MAPPO on the fully observable two-agent Reacher task (every agent observes all the joint angles and velocities as well as the target position – in black – and the target to fingertip – in green – vector). Depending on the seed of the run, teams converge to counter-clockwise (\(\theta_2 \geq 0\) as in Figure 3 (b)) or clockwise (\(\theta_2 \leq 0\) arm bends. Thus we can build a mixture-of-expert dataset by combining equal proportions of demonstrations from clockwise and counter-clockwise teams. Finally, we explore the impact of Full Observability (FO) versus Partial Observability (PO) by considering three types of observation functions: all-observant (FO: every agent fully observes the environment), independent (PO: each agent only sees the target and the velocity and angle of the joint it controls), and leader-only (PO: both agents observes the two joints but only the red agent observes the target’s position). Note that with PO no agent observes the target to fingertip vector. Thus, in leader-only, only the leader can estimate whether or not the fingertip matches the target. In independent, no agent has enough information to estimate this, yet both observe the target position. These tasks are very challenging and require agents to agree on following a specific convention (either clockwise or counter-clockwise arm bend) to reach a given target location and get rewards.

**Four-agent Ant.** Similarly, we use a MAMuJoCo-like decomposition [45] of the D4RL [18] offline ant task to make it multi-agent: each individual limb (composed of two joints) is controlled by a different agent. For the offline datasets, we use the single-agent D4RL datasets and consider two types of observation functions. For fully observable tasks, every agent observes the whole robot: torso observations (i.e., vertical position, orientation, angular and translational velocities) and the observations of all the limbs (i.e., angle and angular velocity of each joint). For the partially observable tasks, agents only observe the limb they control, and the torso observations are only made available to the yellow limb agent. PO tasks are very challenging in this case because only the yellow agent knows if the ant is moving in the correct direction and it must therefore learn to “steer” the whole robot.

Table 2 details all the datasets’ score distributions as well as the reference agents’ returns – expert and random – that are used to normalize performance. Note that ant datasets are from D4RL [18] (single agent datasets that we split into multi-agent observations and actions) while we generated the two-agent reacher mixture-of-expert dataset using MAPPO.

**Table 2: Normalized measures of datasets’ scores distributions and normalization performances.**

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Scores</th>
<th>min (%)</th>
<th>mean (%)</th>
<th>median (%)</th>
<th>max (%)</th>
<th>expert return</th>
<th>random return</th>
</tr>
</thead>
<tbody>
<tr>
<td>teacher</td>
<td>expert-mix</td>
<td>38.8</td>
<td>100.0</td>
<td>98.9</td>
<td>152.3</td>
<td>-4.237</td>
<td>-11.145</td>
</tr>
<tr>
<td>ant</td>
<td>random</td>
<td>-3</td>
<td>6.4</td>
<td>7.2</td>
<td>10.3</td>
<td>3879.7</td>
<td>-325.6</td>
</tr>
<tr>
<td></td>
<td>medium</td>
<td>-4.8</td>
<td>8.0</td>
<td>95.1</td>
<td>107.2</td>
<td>142.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>full-replay</td>
<td>-22.4</td>
<td>72.0</td>
<td>77.8</td>
<td>134.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>expert</td>
<td>-32.8</td>
<td>117.4</td>
<td>129.4</td>
<td>142.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6 RESULTS

**Table 3: Teams’ performances on the Iterated Coordination Game.** MOMA-PPO is the only decentralized execution method to solve it for all the datasets.

<table>
<thead>
<tr>
<th></th>
<th>IQL</th>
<th>MAIQL</th>
<th>IBC</th>
<th>MOMA-PPO</th>
</tr>
</thead>
<tbody>
<tr>
<td>fav.</td>
<td>1. ± 0.</td>
<td>1. ± 0.</td>
<td>1. ± 0.</td>
<td>1. ± 0.</td>
</tr>
<tr>
<td>neutral</td>
<td>1. ± 0.</td>
<td>0.9 ± 0.1</td>
<td>0.55 ± 0.11</td>
<td>1. ± 0.</td>
</tr>
<tr>
<td>unfav.</td>
<td>1. ± 0.</td>
<td>0. ± 0.</td>
<td>0. ± 0.</td>
<td>1. ± 0.</td>
</tr>
</tbody>
</table>
Table 4: Teams’ performances on two-agent Reacher with mixture-of-experts dataset for different observation functions. Scores are normalized with expert and random performances. (a) Independent learners fail on datasets that contain a mixture of incompatible experts while MOMA-PPO (and to some extent MAIQL) are able to coordinate agents. (b) Current model-free methods are unable to adapt agents’ behaviors while MOMA-PPO significantly outperforms the baseline across all settings.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Algorithms</th>
<th>centralized</th>
<th>model-free</th>
<th>independent learners</th>
<th>model-based (ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>IQL</td>
<td>MAIQL</td>
<td>MATD3+BC</td>
<td></td>
</tr>
<tr>
<td>FO</td>
<td>all-observant</td>
<td>1.07 ± 0.01</td>
<td>0.96 ± 0.05</td>
<td>1.04 ± 0.01</td>
<td></td>
</tr>
<tr>
<td>PO</td>
<td>independent learner-only</td>
<td>0.92 ± 0.04</td>
<td>0.59 ± 0.03</td>
<td>0.76 ± 0.04</td>
<td>0.95 ± 0.06</td>
</tr>
<tr>
<td></td>
<td>leader-only</td>
<td>0.80 ± 0.05</td>
<td>0.73 ± 0.02</td>
<td>0.84 ± 0.02</td>
<td>0.39 ± 0.02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Algorithms</th>
<th>centralized</th>
<th>model-free</th>
<th>independent learners</th>
<th>model-based (ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>IQL</td>
<td>MAIQL</td>
<td>MATD3+BC</td>
<td></td>
</tr>
<tr>
<td>FO</td>
<td>ant-random</td>
<td>0.12 ± 0.00</td>
<td>0.28 ± 0.01</td>
<td>0.28 ± 0.03</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ant-medium</td>
<td>0.97 ± 0.02</td>
<td>0.85 ± 0.02</td>
<td>0.81 ± 0.02</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ant-full-replay</td>
<td>1.22 ± 0.02</td>
<td>0.77 ± 0.21</td>
<td>0.95 ± 0.13</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ant-expert</td>
<td>1.26 ± 0.01</td>
<td>1.24 ± 0.00</td>
<td>1.06 ± 0.07</td>
<td></td>
</tr>
<tr>
<td>PO</td>
<td>ant-random</td>
<td>0.31 ± 0.00</td>
<td>0.34 ± 0.04</td>
<td>0.31 ± 0.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ant-medium</td>
<td>0.14 ± 0.02</td>
<td>0.11 ± 0.01</td>
<td>0.17 ± 0.01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ant-full-replay</td>
<td>0.18 ± 0.02</td>
<td>-0.07 ± 0.10</td>
<td>0.21 ± 0.02</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ant-expert</td>
<td>-0.16 ± 0.01</td>
<td>-0.23 ± 0.02</td>
<td>0.05 ± 0.04</td>
<td></td>
</tr>
</tbody>
</table>

The experimental procedure such as hyperparameters, training routine, and raw learning curves are detailed in Appendix B. All algorithms are trained to convergence and we used 10 seeds for the Iterated Coordination Game and 3 seeds for MAMuJoCo tasks. Tables are normalized and report the mean evaluation performance and the standard error of the mean across seeds. Evaluation is done for 100 episodes using the greedy policies (no sampling).

6.1 Strategy Agreement

Table 3 reports the results for the offline Iterated Coordination Game and validates most of our intuitions about strategy agreement: the centralized execution (IQL) and model-based (MOMA-PPO) approaches are able to coordinate agents regardless of the datasets. On the other hand, independent BC agents imitate the dataset behavior and therefore coordinate only if the dataset majorly demonstrates coordination. Surprisingly, the CTDE model-free approach MAIQL is able to break symmetry and coordinate agents in the neutral dataset. We hypothesize that small numerical errors in the centralized value approximation have the team favor one equivalent strategy over the other. Unfortunately, the conservatism of model-free methods forces agents to stay close to the demonstrated behaviors and prevails over this brittle symmetry-breaking mechanism in the unfavorable – i.e., uncoordinated – dataset.

Table 4 (a) confirms that these insights on strategy agreement hold in the more complex two-agent Reacher environment: model-free methods struggle with strategy agreement, especially under partial observability. Again, CTDE methods – particularly MAIQL – tend to fare better than independent learners. Interestingly, IBC fares best among independent learners. Finally, our model-based CTDE approach MOMA-PPO solves strategy agreement and performs on par with centralized execution (IQL), a setting that sidesteps the strategy agreement issue altogether. MOMA-PPO matches or significantly outperforms all other baselines.

6.2 Strategy Fine-Tuning

From Table 4 (b) one can investigate how the different offline methods cope with strategy finetuning. First, IQL performs on par with the other model-free methods which suggests that centralized execution (single-agent) vs. decentralized execution (multi-agent) is less a consideration for strategy fine-tuning than it is for strategy agreement. Yet, this also highlights that model-free methods (even when centralized) are unable to perform strategy fine-tuning. Indeed, they are surpassed by our model-based method, MOMA-PPO. This latter generates additional synthetic experiences that allow for strategy fine-tuning in addition to strategy agreement. For model-free methods, independent learners tend to outperform CTDE ones (which echoes [43] observations). Additionally, comparing IQL with MAIQL performance does highlight that offline multi-agent coordination is more challenging in varied datasets (i.e. medium or full-replay that display both coordinated and uncoordinated behaviors) than in uniformly coordinated datasets (i.e. expert).
In partially observable (PO) tasks, model-free methods are unable to adapt the behaviors demonstrated in the datasets and they result in teams that run in circles because the yellow agent (the only one to observe the torso’s headings and velocities) fails to correct the other limbs’ motions. Conversely, with MOMA-PPO, the yellow agent steers the ant toward the correct direction, and the teams reach very satisfactory performances provided that the datasets have enough coverage to learn a world model that can simulate diverse and robust behaviors (cf. the lower performance for the expert dataset).2

Finally, the poor performance of MAIQL-rt suggests that IQL’s finetuning abilities might not carry over to the multi-agent setting (even though we used the ground-truth simulator to generate the rollouts). While there might be multiple causes, we hypothesize that it is mainly due to MAIQL’s instability since it required intensive hyperparameters finetuning and filtering out collapsed runs for ant tasks. We believe that MAIQL’s instability is exacerbated by the induced non-stationarity of the training data when augmenting it with online interactions. Unlike online methods, offline algorithms are designed to learn on fixed datasets and are thus ill-equipped to deal with data continually collected by changing policies. Similarly, experiments that used MAIQL instead of MAPPO for MOMA (i.e., MOMA-IQL) quickly led to unstable learning and exploding losses.

In conclusion, our results show that current model-free offline MARL methods fail at offline coordination. Crucially, this deficiency remains even in very simple domains such as the Iterated Coordination Game. Also, it sometimes leads to underperforming basic imitation learning (i.e., IBC) or the datasets’ average score (see Table 2). Conversely, our model-based approach is able to coordinate agents even under severe partial observability and with learned world models by restoring inter-agent interactions throughout learning.

6.3 Ablations
We validate our design choice of using epistemic uncertainty over aleatoric. To do so we replace MOMA-PPO’s penalty with the aleatoric uncertainty penalty of MOPO [67]. For a fair comparison, we followed [67]’s parameter search procedure (i.e., coefficient values of 1 and 5). Additionally, we investigated aleatoric penalty both with and without the use of adaptive rollout length (which is based on epistemic uncertainty). The resulting learning curves on Ant-full-replay with full and partial observability are displayed in Figure 6 in Appendix B.2. It appears that using epistemic uncertainty always significantly outperforms using aleatoric uncertainty except in the case of full observability and adaptive rollouts where the comparison is not statistically significant. This validates the soundness of our choice regarding the use of epistemic uncertainty over aleatoric uncertainty in deterministic environments.

Appendix B.2 reports additional ablations, which main insights we discuss here. First, using the ground-truth simulator yielded higher scores suggesting that the performance of MOMA-PPO can be further improved by learning a more accurate world model. Then, not clipping the generated next-states to the dataset’s bounding box prevented generating length 10 rollouts as the states’ magnitude exploded after a few world-model steps. Also, the use of uncertainty penalty (\(\lambda_d\)) and adaptive rollouts’ length (\(l_e\)) are necessary to achieve satisfactory results. Finally, varying the rollout’s maximum length from 5 to 50 did not significantly impact performance.

7 DISCUSSION AND CONCLUSION
This work explores coordination in offline MARL and highlights the failures of current model-free methods. For instance, they struggle in the presence of multiple equivalent but incompatible optimal team strategies (strategy agreement), or when partial observability requires the team to adapt the behaviors demonstrated in the dataset (strategy fine-tuning). In such scenarios, prevalent model-free methods might even fail to match the dataset’s performance and fall short compared to behavioral cloning. To address these problems, we propose MOMA-PPO which is, to our knowledge, the first model-based offline MARL approach. Our method is able to coordinate teams of offline learners and significantly outperforms model-free alternatives. Interestingly, it also outperforms the fully centralized model-free method IQL [30] even though this latter completely bypasses the strategy agreement problem. This suggests that model-free methods, even when fully centralized, are unable to deal with strategy fine-tuning. We also observe in our experiments that the single-agent approach of fine-tuning offline methods (i.e., IQL) with online interactions [30] fails in the multi-agent setting (i.e., MAIQL). Our model-based approach succeeds at both multi-agent strategy agreement and strategy fine-tuning problems which goes to show that the benefits of offline model-based approaches over offline model-free ones [67] hold in the multi-agent setting. This might indicate that world models may generalize better than values networks in offline learning’s limited data regime. Finally, for tasks that require adapting the team’s behavior, dataset coverage might be more desirable than demonstrated performance. Indeed, methods fare better on random datasets than on expert ones.

MOMA-PPO’s successes put forward the benefits of model-based methods that leverage online policy optimization. Nevertheless, our approach of coordinating agents through a world model is general and some might be interested in extending it to more sample-efficient policy learning algorithms (i.e., MOMA-SAC). Also, the dataset bounding box clipping and adaptive rollouts developed for MOMA-PPO might benefit the single-agent setting.

Finally, this work aims to pinpoint an overlooked issue in the offline MARL community and proposes a new avenue of model-based solutions that shifts away from conventional model-free approaches. Therefore, we look forward to future work that will further analyze model-based offline MARL approaches and scale them to more domains and datasets.

ACKNOWLEDGMENTS
The authors would like to thank Luis Pineda, Hengyuan Hu, and Eugene Vinitsky for their help, insightful discussions, and advice. This research was enabled in part by support provided by Calcul Québec and the Digital Research Alliance of Canada. This work was partially conducted while Paul was interning at Meta AI - FAIR.
REFERENCES


