Self-Learning Governance of Black-Box Multi-Agent Systems

Michael Oesterle[1][0000−0001−6939−1028], Christian Bartelt[1][0000−0003−0426−6714], Stefan Lüdtke[1][0000−0002−1488−4236], and Heiner Stuckenschmidt[2][0000−0002−0209−3859]

[1] Institute for Enterprise Systems (InES), University of Mannheim, Germany
{oesterle, bartelt, luedtke}@es.uni-mannheim.de
[2] University of Mannheim, Germany heiner@informatik.uni-mannheim.de

Abstract. Agents in Multi-Agent Systems (MAS) are not always built and controlled by the system designer, e.g., on electronic trading platforms. In this case, there is often a system objective which can differ from the agents' own goals (e.g., price stability). While much effort has been put into modeling and optimizing agent behavior, we are concerned in this paper with the platform perspective. Our model extends Stochastic Games (SG) with dynamic restriction of action spaces to a new self-learning governance approach for black-box MAS. This governance learns an optimal *restriction policy* via Reinforcement Learning.
As an alternative to the two straight-forward approaches—fully centralized control and fully independent learners—, this novel method combines a sufficient degree of autonomy for the agents with selective restriction of their action spaces. We demonstrate that the governance, though not explicitly instructed to leave any freedom of decision to the agents, learns that combining the agents' and its own capabilities is better than controlling *all* actions. As shown experimentally, the self-learning approach outperforms (w.r.t. the system objective) both "full control" where actions are always dictated without any agent autonomy, and "ungoverned MAS" where the agents simply pursue their individual goals.

Keywords: Multi-Agent System · Governance · Self-Learning System · Reinforcement Learning · Electronic Institution.

1 Introduction

1.1 Motivation

Multi-Agent Systems (MAS) are widely used as a general model for the interaction of autonomous agents, and have been applied to a vast range of real-world settings, for example Algorithmic Trading [1], Traffic Management [33], and Multi-Player Video Games [25] (see [42] for a recent survey of MAS applications).

Example 1. Consider a stock market where high-frequency trading algorithms typically generate the vast majority of orders. Obviously, agents in this setting act autonomously and in a self-interested manner in order to maximize their profit. As is known, this behavior leads to problems like high volatility and extreme stock price behavior [26]. It is therefore crucial for regulators to provide both stability (i.e., ensure that extreme price movement flash crashes will not happen) and opportunity (i.e., ensure that investors can still use intricate, proprietary strategies to make profit).

In this example—as in many other applications areas—the agents cannot (or should not) be fully controlled, but must have a sufficient degree of freedom regarding their actions. At the same time, some level of control needs to be imposed on the agents such that a system objective can be achieved.

The scope of this paper is therefore a subclass of MAS with three more assumptions, inspired by the concept of *Electronic Institutions* (EI) [4] as described in Sec. 2:

(a) The agents are truly autonomous entities whose goals and strategies cannot be known ("black boxes"), but only observed through their actions,
(b) in addition to the agents' individual goals, there is a *system objective* which does not necessarily coincide with any of the former goals, and
(c) agent actions can be restricted by a *governance* which has the power to enforce such restrictions.

We propose a novel approach to governing an MAS which combines the restriction concept of EI with dynamic rule-setting, provided by a Reinforcement Learning (RL) component (the *governance*). This governance observes the public information of the MAS, i.e., actions and transitions, and learns optimal restrictions, which depend on the system state and the respective agent's observation.

A common method for governing agents in an EI is the use of *norms* with a focus on rewards and sanctions as the means of influencing agent behavior, while the action space itself is not affected. This makes two essential assumptions about the agents: First, "the effectiveness of these norms depends heavily on the importance of the affected social reality for the individual" [6], and second, the normative awareness needs to be comparable for all participating agents (*interpersonal utility comparison*). For unknown agents, we argue that these assumptions cannot be expected to hold true, which is why we base our governance on (mandatory) restrictions of the agents' action sets. The dynamic nature of the rule-setting process (*rule synthesis*) is due to the fact that agents themselves can act strategically and are therefore able to exploit any static rule set.

Of course, the governance's "power to restrict" requires some sort of physical control over the MAS. This requirement is satisfied in a wide range of applications, for example by any digital platform where agents are software components, and actions are chosen by exchanging messages. Therefore, we assume the adherence to restriction to be given in this work.

1.2 Illustration of the Governance Approach

The simultaneous execution and learning of a *Governed Multi-Agent System* (GMAS) is shown in Fig. 1 (see the formal model in Sec. 3 for the definition and explanation of the variables, and Algorithm 1 for the actual run-time loop). The governance is used, i.e., its restriction policy is queried, at every execution step of the MAS to determine the set of allowed and forbidden actions, whereas the learning happens in between those execution steps.

In each learning step, the governance optimizes its restriction policy in order to maximize the system objective, given the observation of the last step. At the same time, the agents can update their own action policies, but this is not part of the GMAS (black-box agents).

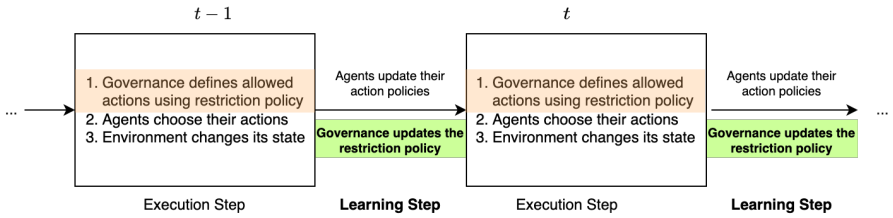


Fig. 1. Sequence of execution and learning steps in a Governed Multi-Agent System

1.3 Contribution

We show in this paper how a self-learning governance with the ability to restrict action spaces can add value to an MAS. This is demonstrated by comparing its performance to two natural alternatives (see also [37]):

- *Ungoverned MAS* (UMAS), in which the agents alone decide on their actions, such that coordination or cooperation (if any) evolves on its own, and
- *Fully Controlled MAS* (FMAS), where the governance prescribes all agent actions, leaving no room for autonomous decisions.

The main contributions of this work are: We give a formal definition of a Governed Multi-Agent System (Sec. 3), we conceptualize an RL governance for this model, analyzing the assumptions made in the model and describing the governance's learning behavior (Sec. 4), and we present experiments (Sec. 5) to demonstrate that this method can significantly outperform both alternatives: UMAS and FMAS.

2 Related Work

Most MAS literature focuses on the agents' perspective, attempting to improve their learning behavior [32, 35]. The underlying model, the *Stochastic Game*

(SG), is both an extension of a *Markov Decision Process* (MDP) to multiple agents, and an extension of a *Normal-Form Game* to multiple states. Hence, methods from both Stochastic Processes and Game Theory have been adapted to this setting. Both in Game Theory and in Machine Learning, it is very common to assume discrete time steps and therefore a synchronized interaction between agents. We will make use of this assumption for the interaction between environment, agents and governance (see Sec. 3).

For a single-agent (stationary) MDP, the most common approach—Reinforcement Learning—includes a variety of algorithms which have been proven to converge to an optimal strategy [38]. What makes it hard to transfer these algorithms to multi-agents settings is the fact that the rewards and transitions in an SG depend on the joint action of all agents, making the system non-stationary from the perspective of each agent. Coming from the game-theoretic side, the extension of solution approaches for normal-form games (mostly based on the notion of equilibrium strategies) to SG is no less challenging.

Nevertheless, there have been many successful approaches to the Multi-Agent Learning problem by introducing new concepts for equilibria (e.g. correlated equilibria [19] and cyclic equilibria [43]) or by making additional assumptions: Among others, agents can learn optimal strategies if all agents receive the same rewards (Team Markov Games [39]), if the game is a Zero-Sum Game [23], if all opponents are stationary [13], or if the "rate of non-stationarity" is bounded by a *variation budget* [12]. The general problem of finding an optimal strategy in a model-free, general-sum SG, however, is still an open challenge [42].

As a consequence, researchers have introduced additional support for the learning agents. This support can be either restricted to the interaction between the agents, or it can involve another entity besides the agents.

For the first type, agents are usually allowed to exchange additional information in order to find optimal strategies [21, 11] (see also the recent MARL surveys of Zhang et al. [42] and Gronauer and Diepold [20]).

The second type relies on non-agent components to solve the learning problem: In its most general notion, the concept of *Environment-Mediated Multi-Agent Systems* (EMMAS) states: "When designing a system that is based only on local interactions in the environment and the emergent properties resulting from these interactions, it is a difficult research problem on the one hand to obtain the required global behavior of the system and on the other hand to avoid undesired global properties", and therefore suggests to "off-load some of the agent complexity into the processes of the dynamic agent environment" [40].

Electronic Institutions (EI) [30, 16] provide an *institution* as the entity which regulates agent interactions, among many other features. The framework contains an "implementation of the control functionality of the institution infrastructure [which] takes care of the institutional enforcement", which can refer to both *norms*—which can be violated—and enforced *rules*. While these two terms are not always used consistently, we use here the convention that rules are "norms that can be effectively controlled and thus enforced, such that violation is impossible" [27].

The EI framework itself does not only describe rule-setting capabilities, but also Agents, Roles, a Performative Structure, and Normative Rules, among others [16]. The same holds for alternative models for social coordination, e.g., ANTE [24], or INGENIAS [18] ([3] includes details of all these frameworks). However, we use only one feature of EI: The ability to restrict the space of available actions for the participating agents. This has been described as an important part of an EI by Esteva et al.: "An electronic institution defines a set of rules that structure agent interactions, establishing what agents are permitted and forbidden to do" [15]. Aldewereld et al. emphasize that "organisational objectives are not necessarily shared by any of the individual participants, but can only be achieved through their combined action" [3], and that "one cannot make any assumptions about the inner workings of participants. [...] Rather, external aspects of the participants (actions, interactions, etc.) have to be leveraged to create the required coordination structures" [3].

Norms are a very common approach for achieving system goals in MAS. The distinction between norms and rules ("[Norms] are a concept of social reality [...] Therefore, it is possible to violate them" [6]) has been made many times in the literature; they have been called "social conventions" and "explicit prescriptions" [14], "legalistic view of norms" and "interactionist view of norms" [9], "norms" and "regimented norms" [6], or "norms" and "hard constraints" [17, 27].

Normative Multi-Agent Systems (NorMAS) [14, 8] embrace the idea that agent communities can self-regulate their interactions without a controlling force. Therefore, the field focuses on (violable) norms, their creation or emergence, observation, revision, adherence or violation, and sanctioning mechanisms. However, this requires the two assumptions mentioned in Sec. 1.1: Norm-awareness and inter-agent utility comparability. In our opinion, these requirements do not hold for black-box agents with individual goals ("How to deal with a lack of normative awareness and if it is being considered, how to check the lack of normative awareness if an agent's knowledge base is not accessible?" [6]). In consequence, our focus lies on the other type of institutional enforcement: Rules for allowed and forbidden actions.

The original implementation of EI (and its development environment EIDE [31]) envisaged a clear distinction between rule/norm creation at design-time and agent interaction at run-time (i.e., all rules/norms are given independently of the agents and do not change during execution). A logical next step was the Autonomic Electronic Institutions (AEI) approach [10, 5]: Acknowledging the fact that static norms are not always sufficient for dealing with self-adapting agents, it moved norm creation from the design time to the run-time and allowed for dynamic changes. EI was therefore extended to include an evolutionary norm adaptation mechanism (e.g., a genetic algorithm). As we will see later, this is somewhat similar to our governance (defining and updating institutional rules at run-time).

Like Multi-Agent Learning in general, normative capabilities in MAS can either be part of the agents [34], or part of an additional entity [2] (or both). While early work defined static norms at design-time [37, 7], the field has since

evolved towards run-time norm creation, synthesis and adaptation [28], applying methods like Automated Theorem Proving [29] or Deep Learning [2] to NorMAS.

This development towards dynamic norm creation and adaptation has, to our knowledge, not yet been examined for rules (i.e., hard constraints). In this paper, we fill the gap by demonstrating that dynamic rules do have the potential to enhance the capabilities of an MAS. Moreover, the RL approach employed here for the governance component is shown to be well-suited for on-line learning of a restriction policy in an environment where the agents and their behavior can only be observed from outside.

3 Model

3.1 Notation

Vectorized Variables Let $\mathcal{S}$ be a set, and I be an index set. A single variable $s \in \mathcal{S}$ is written in regular face, whereas a vector $\boldsymbol{s} = (s_i)_{i \in I} \in \mathcal{S}^I$ is written in bold face. The index set is usually omitted when the context is clear. Variables that change over time always have the current time step as a superscript, as in $s^{(t)}$ or $\pi_i^{(t)}$.

Categorical Distribution Given a finite set $\mathcal{S}$, $\Delta(\mathcal{S})$ denotes the set of all discrete probability distributions over $\mathcal{S}$, i.e., the set of all functions $p : \mathcal{S} \to [0, 1]$ with $\sum_{s \in \mathcal{S}} p(s) = 1$.

Image and Support Let $f : A \to B$ be a function. Then $\mathrm{im}(f) := \{f(x) : x \in A\}$ is the *image* of f. If $B = \mathbb{R}$, $\mathrm{supp}(f) := \{x \in A : f(x) \neq 0\}$ is the *support* of f.

3.2 Multi-Agent System

Consider a *Partially Observable Stochastic Game* (POSG) over discrete time steps $t \in \mathbb{N}_0$, i.e., a 7-tuple $(I, \mathcal{S}, \mathcal{O}, \boldsymbol{\sigma}, \mathcal{A}, \boldsymbol{r}, \delta)$ with agent set $I = \{1, ..., n\}$, state set $\mathcal{S}$, observation set $\mathcal{O}$, observation functions $\sigma_i : \mathcal{S} \to \mathcal{O} \, \forall i \in I$, fundamental action set $\mathcal{A}$ with $k := |\mathcal{A}| \in \mathbb{N}$, agent reward functions $r_i : \mathcal{S} \times \mathcal{A}^I \to \mathbb{R} \, \forall i \in I$ and a probabilistic transition function $\delta : \mathcal{S} \times \mathcal{A}^I \to \Delta(\mathcal{S})$.

Each agent has an (unknown) stochastic action policy $\pi_i : \mathcal{O} \times 2^{\mathcal{A}} \to \Delta(\mathcal{A})$ which defines its behavior. These policies take as input not only the agent's current observation, but also a set $A \subseteq \mathcal{A}$ of allowed actions. Referring to the assumption of non-violable rules (see Sec. 1.1), we take as a given that forbidden actions are never chosen, hence $\mathrm{supp}\, \pi_i(s, A) \subseteq A \, \forall i \in I, s \in \mathcal{S}$.

An action policy is called *static* if it is constant in t; otherwise it is called *dynamic*. Note that a static policy π can still be non-deterministic, since the concrete action is sampled from the categorical distribution $\pi(o, A) \in \Delta(A)$.

3.3 Governance

The governance component returns a set $A \subseteq \mathcal{A}$ of allowed actions when given an input pair consisting of the overall environmental state and an agent's ob-

servation. This function is called the *governance policy* $\pi_G : \mathcal{S} \times \mathcal{O} \to 2^{\mathcal{A}}$. Note that the set of allowed actions can never be empty, i.e., $\varnothing \notin \mathrm{im}(\pi_G)$.

In contrast to a standard MAS, where the environment provides all the input for the agents' action policies, there is now an intermediary step in which the governance computes the set of allowed actions for each agent, which is then passed to the agent's policy in addition to its observation.

The system objective is given as a reward function $r_G : \mathcal{S} \times \mathcal{A}^I \to [0, 1]$, allowing the governance to directly measure the success of its restrictions after each environment step. The normalized range of r_G is chosen for ease of comparability.

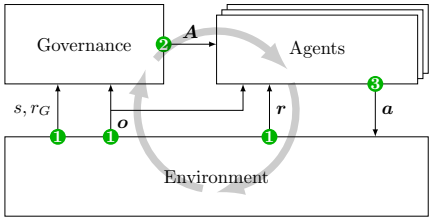
Definition 1. *A* Governed Multi-Agent System *(GMAS) is the 9-tuple*

$$(I, \mathcal{S}, \mathcal{O}, \boldsymbol{\sigma}, \mathcal{A}, \boldsymbol{r}, \delta, \pi_G, r_G) \ .$$

The governance is a *centralized controller* insofar as it observes the entire MAS and defines restrictions in a centralized way. However, the fundamental difference to the usual notion of "centralized control" is that the governance leaves a substantial amount of autonomy to the agents. This is not enforced by its design, but emerges naturally: The synergy between the governance's and the agents' capabilities gives a performance advantage over full control, causing the governance to allow multiple actions at most times (see Sec. 5).

3.4 Sequence of Actions in a GMAS

Fig. 2 shows the exchange of data in one execution step (see Fig. 1) of a GMAS: The environment provides the agents with their respective rewards and observations, while passing to the governance the environment state, the governance reward and agent observations ❶. The governance then calculates the sets of allowed actions for each agent, and passes them to the respective agent ❷. Finally, the agents choose their actions and communicate them back to the environment ❸ which executes the transition. For simplicity and clarity of presentation, all n queries to the governance have been wrapped up into one arrow.



$s \in \mathcal{S}$: Environmental state
$r_G \in \mathbb{R}$: Governance reward
$\boldsymbol{o} \in \mathcal{O}^I$: Agent observations
$\boldsymbol{A} \subseteq \mathcal{A}^I$: Allowed actions
$\boldsymbol{r} \in \mathbb{R}^I$: Agent rewards
$\boldsymbol{a} \in \mathcal{A}^I$: Agent actions

Fig. 2. Execution Step of a GMAS

In pseudocode (see Algorithm 1), the run-time loop is very similar to the standard execution of an RL environment (e.g., in OpenAI Gym), with an additional governance step.

Algorithm 1: Run-time loop of a governed MAS

Data: GMAS $G = \left(I, \mathcal{S}, \mathcal{O}, \boldsymbol{\sigma}, \mathcal{A}, \boldsymbol{r}, \delta, \pi_G^{(0)}, r_G\right)$
Choose initial environmental state $s^{(0)} \in \mathcal{S}$;

for $t \in \{0, ..., T\}$ **do**
  `// Execution step`
  for $i \in I$ **do**
    $o_i^{(t)} \leftarrow \sigma_i(s^{(t)})$ `// Compute agent observation from state`
    $A_i^{(t)} \leftarrow \pi_G(s^{(t)}, o_i^{(t)})$ `// Get allowed actions from governance`
    $a_i^{(t)} \leftarrow \pi_i(o_i^{(t)}, A_i^{(t)})$ `// Get chosen action from agent`
  end
  $\boldsymbol{r}^{(t)} \leftarrow \boldsymbol{r}(s^{(t)}, \boldsymbol{a}^{(t)})$ `// Get rewards`
  $s^{(t+1)} \leftarrow \delta(s^{(t)}, \boldsymbol{a}^{(t)})$ `// Execute transition`

  `// Learning step`
  $\pi_G^{(t+1)} \leftarrow \text{train}(\pi_G^{(t)})$ `// Train governance`
  for $i \in I$ **do**
    $\pi_i^{(t+1)} \leftarrow \text{train}(\pi_i^{(t)})$ `// Train agent`
  end
end

3.5 Degree of Restriction

There is a natural trade-off between achieving the system objective and preserving agent freedom: The more actions the governance forbids, the higher its level of control over the agents—in the extreme case, only a single action is allowed for any given observation, resulting in a fully deterministic trajectory. On the other end of the spectrum, the governance always allows all actions, reducing the GMAS to an ordinary MAS.

It is therefore reasonable to measure the *degree of restriction*, i.e., the percentage of forbidden actions, and to assess this metric in relation to the governance's performance:

Definition 2. *For an individual agent $i \in I$ and time step $t \in \mathbb{N}_0$, the* degree of restriction *is defined as*

$$\rho_i^{(t)} := 1 - \frac{\left|\pi_G\left(s^{(t)}, o_i\left(s^{(t)}\right)\right)\right|}{|\mathcal{A}|} \in [0, 1]\,.$$

The overall degree of restriction $\rho^{(t)} := \frac{1}{n}\sum_{i \in I} \rho_i^{(t)}$ is simply the mean over all agents. The higher the degree of restriction, the lower the autonomy of the agents.

It should be noted that real-world agents oftentimes cannot choose every action at every step. Instead, only a subset of actions is feasible, depending on the environmental state (*parametric action spaces*). In this case, the degree of

restriction should be defined as the ratio between forbidden actions and feasible actions.

4 Model Analysis

4.1 Fairness

Agents who make the same observation $o \in \mathcal{O}$ at a time step t are always allowed to perform the same actions $\pi_G\left(s^{(t)}, o\right)$. This is in line with a common-sense definition of fairness: The governance treats all agents the same way. To achieve this, learning (i.e., a change of the governance policy) cannot take place within a time step, but only after all agents have been given their action sets.

4.2 Learning

The GMAS model does not specify any particular learning algorithm, but only requires a governance policy π_G to be available for querying at all times. The restriction policy can be any function $\mathcal{S} \times \mathcal{O} \to 2^{\mathcal{A}}$, but, of course, the governance's goal is to find a restriction policy which maximizes the reward r_G, given the agents' behavior. Since the governance interacts with the ungoverned MAS in a cycle of information, reward and action, RL seems to be the natural way to optimize this policy.

From this perspective, the governance itself is a Reinforcement Learning agent which acts on the entire MAS as its environment: The governance interacts with the MAS environment and the agents, but only sees how its own actions (i.e., defining sets of allowed actions) influence its reward and the environmental state. Therefore, it can be treated as a reinforcement learner with action policy π_G and reward r_G. Its environment has the transition function $\delta' : \mathcal{S} \times \left(2^{\mathcal{A}}\right)^I \to \Delta(\mathcal{S})$ with $\delta'(s, \boldsymbol{A}) := \delta\left(s, \boldsymbol{\pi}(\boldsymbol{\sigma}(s), \boldsymbol{A})\right)$, which is a composition of observation functions $\boldsymbol{\sigma}$, agent policies $\boldsymbol{\pi}$ and MAS transition function δ.

δ' is not explicitly known to the governance, such that a model-free algorithm must be used. Moreover, since the governance policy is the action policy of the governance, standard model-free RL algorithms like A3C, DQN or PPO can be directly applied. The governance is structurally equivalent to a multi-label classifier: Its policy outputs a subset of the (finite) fundamental action set. Thus, specialized network architectures for this type of classifier could also be applied in order to build a more effective governance policy.

Since agents can (and probably will) change their behavior according to the current restriction policy, a GMAS is inherently dynamic and therefore an *online* learning problem: Both sides (agents and governance) react to the other side's actions and strategies by continuously adapting their own action policies. The initial restriction policy can be a random function, or it can be set to simply allow all actions, i.e., $\pi_G^{(0)}(s, o) := \mathcal{A}\ \forall s \in \mathcal{S}, o \in \mathcal{O}$. At run-time, the governance needs to learn continuously in order to keep up with changing agent behavior. Therefore, there is no distinction between traning and evaluation as in classical

RL, but the governance learning process continues throughout the lifecycle of the GMAS.

4.3 Stationarity

It is known [12] that, for a stationary MDP, near-optimal regret bounds can be achieved via RL. The situation is more complicated in the non-stationary case, depending on whether non-stationarity occurs in discrete steps (piece-wise stationarity) or continuously (among other criteria).

The transition function δ is assumed to be stochastic, but stationary. Therefore, the defining factor for the stationarity of a GMAS, seen from the governance's view, is the set of agent policies $\boldsymbol{\pi}$: δ' is stationary if and only if all agent policies are static.

While using static pre-trained models is very common for NLP, Computer Vision and Speech Recognition [41], this is unusual for agent models, since on-line learning lies at the heart of useful behavior in an unknown world. Nevertheless, safety-critical agent-based systems like fully autonomous cars will likely require some sort of certification ensuring that they behave (exactly or approximately) in a certain way, which means that their policy should not, even when learning how to deal with unforeseen situations, be allowed to deviate too far from the approved policy.

Hence, we cannot generally assume that a GMAS is stationary, but in some domains there can be (quasi-)stationary agents, which means that the governance is likely to perform better than in a setting where the agents adapt their strategies arbitrarily fast.

5 Experimental Evidence

The goal of the experiments is to investigate the effect of the governance. For this purpose, we define a game in which the agents need to agree on an action, and then compare three types of systems: Ungoverned MAS (UMAS) which does not have a governance component at all, Fully Controlled MAS (FMAS), and Governed MAS (GMAS).

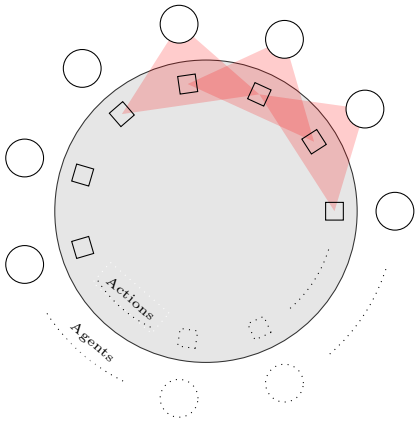


Fig. 3. The dining diplomats' problem

5.1 The Dining Diplomats' Problem

Consider an MAS with agent set $I = \{1, ..., n\}$ and action set $\mathcal{A} = \{1, ..., k\}$

for all agents. The agents are positioned in a circle such that each agent can only see their immediate neighbors (see Fig. 3). At each step, the agents play a card corresponding to one of their available actions. The environmental state represents the currently played cards, i.e., $\mathcal{S} = \mathcal{A}^n$ and $\mathcal{O} = \mathcal{A}^3$.

The agents' goal is to learn to coordinate their actions in order to play the same cards. In the style of the famous *dining philosophers' problem*, we call this problem the *dining diplomats' problem*, requiring the participating agents to come to an agreement under imperfect information.

5.2 Reward Functions

Consider two reward functions—a *state-based* reward and an *observation-based* reward:

$$r_s : \mathcal{S} \to \mathbb{R},\ r_s(s) = \begin{cases} 1 & \text{if } s_1 = \cdots = s_n \\ 0 & \text{else} \end{cases}$$

$$r_o : \mathcal{O} \to \mathbb{R},\ r_o(o) = \begin{cases} 1 & \text{if } o_1 = o_2 = o_3 \\ 0 & \text{else} \end{cases}$$

The state-based reward function only differentiates between "no coordination" and "full coordination", while the observation-based reward also shows local coordination between three agents (i.e., the observation space of one agent). The three system types use these reward functions as follows:

	Agents	Governance
UMAS	r_o	-
FMAS	r_s	r_s
GMAS	r_o	r_s

In the FMAS type, agents and governance have the same information about achieving their goals, so the governance cannot use the agents as an additional source of intelligence. In GMAS, however, the agents have access to more detailed information through r_o. Hence, the two pivotal dimensions are (a) access to low-level/high-level information and (b) dense and sparse rewards.

5.3 Configurations

We compare the three types for four different problem sizes: Tiny ($n = 5, k = 3$), small ($n = 10, k = 5$), medium ($n = 15, k = 7$) and large ($n = 20, k = 10$). This

allows us to see clearly at which complexity the non-GMAS types fail to achieve coordination, and therefore highlights the value added by the synergy.

The size $|\mathcal{S}| = k^n$ of the state space grow polynomially in the number of actions, but exponentially in the number of agents: In the tiny configuration, there are $3^5 = 243$ states, while this number is $5^{10} \approx 10^7$ for the small configuration, $7^{15} \approx 4 \cdot 10^{12}$ for the medium configuration, and 10^{20} for the large configuration.

5.4 Frameworks and Algorithms

For our experiments, we used the *RLlib* library [22] for Multi-Agent learning, which is based on the *Ray* distributed computing framework. Both agents and governance use a standard configuration of the Proximal Policy Optimization (PPO) algorithm [36].

The interaction between agents, governance, and environment requires a sequential MAS execution: The governance needs to act (i.e., produce a set of allowed actions) before an agent can choose from this set. All agent actions, in turn, cause the environment to proceed to the next state. Therefore, the governance is queried n times for each environmental step, while the agents each only act once during the same period.

All experiments were run in ten independent samples for $5 \cdot 10^6$ steps each (empirically determined to ensure sufficient convergence of the action policies).

5.5 Reproducibility

The source code to perform the experiments and generate the graphs is publicly available as a Jupyter notebook, allowing for simple reproduction of the results. The exact results shown in Fig. 4 are stored as Tensorboard log files in the same public repository.

5.6 Results

The results of the experiments can be found in Fig. 4. The governance reward r_G, as the main performance indicator, is shown on the left side, while the graphs on the right depict the corresponding degree of restriction ρ (see Definition 2).

Since the reward at every step is either 0 or 1, the governance reward $r_G^{(t)}$ is the average reward over time, i.e., the percentage of steps where full coordination of all agent actions has been achieved.

In each graph, the mean of the ten samples (thick line) and the individual samples (thin lines) are plotted. The numbers vary strongly between samples, i.e., the mean should be seen as a general trend, but not as the "average run".

Since the governance policy is initialized randomly, all governed types start with $\rho^{(0)} \approx \frac{1}{2}$. The progression of ρ depends on whether the governance is able to learn a "fully controlling" way to create a high reward. If it succeeds, ρ goes up to $\frac{k-1}{k}$ and stays there. Otherwise, the governance must utilize the agents' freedom, and therefore allows more than one action. Notably, the degrees of restriction turn out to be roughly equal in the FMAS and GMAS types.

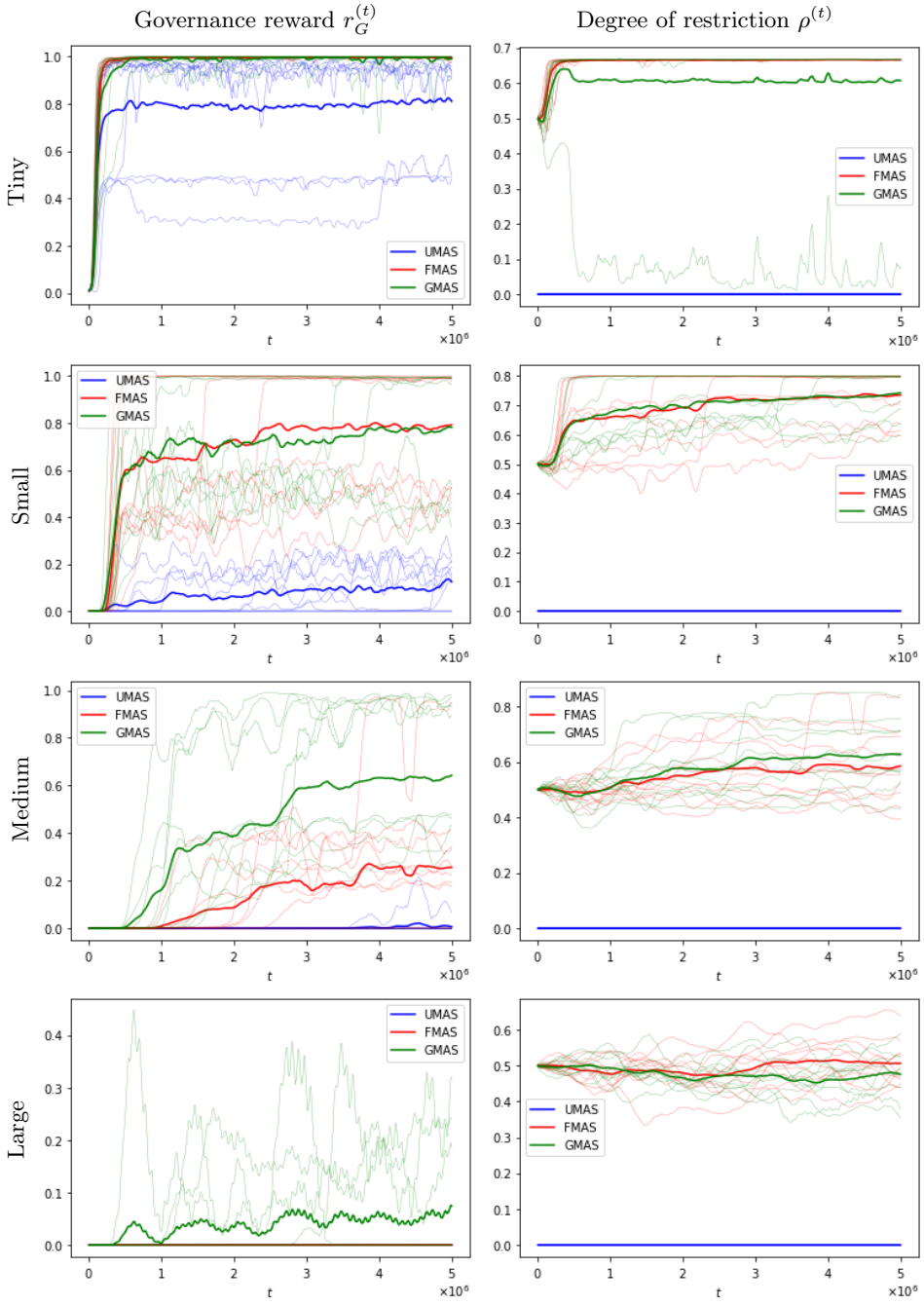


Fig. 4. Experimental results

Thick lines show the mean of $r_G^{(t)}$ and $\rho^{(t)}$ over ten independent samples, while thin lines are the results of the individual samples.

Tiny Configuration Both FMAS and GMAS achieve an almost perfect reward. While the FMAS solves the task by simply allowing a single action for each observation ($\rho^{(t)} \to \frac{k-1}{k} = \frac{2}{3}$), the GMAS uses a slightly lower degree of restriction. The problem is relatively easy, so that the agents in the UMAS can also find a solution, albeit not a perfect one.

Small Configuration This is challenging for the UMAS, but FMAS and GMAS both achieve similar, good results. Sometimes the GMAS uses the maximum degree of restriction, but mostly, agents are given two or three actions.

Medium Configuration The difference becomes larger: The UMAS cannot find a system state that results in a nonzero reward at all, and the FMAS performs approximately half as well as the GMAS. We can see from ρ that even the FMAS governance does not use a maximally restrictive policy, since it cannot find the optimal actions for each observation.

Large Configuration Finally, both UMAS and FMAS are not able to get any rewards. In contrast, the GMAS still achieves a reward of more than 15-20% in four out of ten samples, using a degree of restriction around 50%.

The results show that the GMAS type succeeds in achieving full coordination of the agent actions in a substantial number of time steps. As expected, the average reward decreases with increasing complexity of the setting, but it can handle systems where neither UMAS nor FMAS are able to get any rewards.

5.7 Discussion

Qualitatively, we make the following observations of the three types:

	Tiny	Small	Medium	Large
UMAS	✓	✓		
FMAS	✓	✓	✓	
GMAS	✓	✓	✓	✓

The hypothesis that the synergy of agents and governance significantly outperforms the conventional approaches of ungoverned agents and centralized control, indeed holds true. Notably, the agents simply apply their own (self-learning) strategies, have no normative awareness, and their rewards are not influenced by the governance.

In this section, we give an interpretation of the observed results:

System Objective and Degree of Restriction The governance in the GMAS type has the power of fully controlling the MAS—it could simply allow only one action for any state and observation. Therefore, the crucial observation in the experiments is that the degree of restriction does *not* converge to $\frac{k-1}{k}$.

Instead, the right side of Fig. 4 clearly shows that the governance leaves a substantial amount of freedom to the agents, and that this freedom causes the

governance reward to be much higher than using full control (i.e., the FMAS type).

The balance between governance control and agent freedom is constantly changing, depending on how well the system objective (as measured by the governance reward function) is achieved. It is a crucial feature of our approach that the optimal balance is determined via RL and not defined in advance.

Micro-level and Macro-level Knowledge There are different types of knowledge in the GMAS: The governance can see the entire environmental state and knows which states are most desirable, but does not know effective actions to get there, since its reward function only indicates whether the system objective has been fully achieved. The agents, on the other hand, lack a view of the big picture, but have a better grasp of how to act on a lower level, since their reward function tells them when they are locally coordinated.

In the UMAS, the overall state is not available to the agents at all, not even through the governance. This prevents the agents from finding a globally coordinated solution, even though they can coordinate locally. In the FMAS, the governance sees the big picture, but cannot figure out the necessary actions for the agents to move in the right direction.

The combination of these two levels allows the GMAS to reach global coordination—without ever being instructed how to combine agent and governance knowledge. This setting was chosen since it represents a common pattern in MAS: Individual agents are situated at a specific location in the environment and only able to perceive their surroundings, i.e., a small part of the environment. On the other hand, this small part is where their actions have the biggest impact. The system designer or operator, in contrast, sees the environment as a whole, but does not have the micro-level knowledge about optimal or even useful agent actions. Therefore, the goal is clear, but the way to get there is unknown.

Incentives for Autonomy and Restriction The governance can freely choose the restrictions without being penalized for high degrees of restriction. Consequently, there is no real incentive for the governance to allow multiple actions: The chosen degree of restriction directly reflects the highest expected reward. In the small scenarios, we observe that allowing only one action per observation is a feasible strategy which leads to high rewards. As the scenarios get more complex, however, the governance policy is not maximally restrictive anymore: The governance learns that the autonomous decisions of the agents are more helpful than centralized control. Still, by selectively forbidding actions, the governance can support the agents' action policies.

Penalties for Restrictions A reasonable goal for the governance is to use the least amount of restrictions to achieve its objective, and therefore strive to reduce the degree of restriction whenever this does not counteract the system objective. To this end, we experimented with giving the governance a penalty in

proportion to the current degree of restriction by redefining its reward function as $r'_G := r_G - \alpha \cdot \rho$ with a constant weighting parameter α. This resulted in a much lower reward (even when ignoring the penalty), making the governance drop nearly all restrictions early in the training, before it then defined new, more effective restrictions. However, the penalty often prevented the governance from sufficiently exploring the possible restrictions, so there were many samples where there was never any reward, even in small scenarios.

6 Conclusion and Future Work

In this paper, we have motivated the need for governed MAS, a synergy-based approach for black-box MAS with an additional system objective. We have demonstrated that full control as well as ungoverned learning agents fail to achieve their goals even in simple scenarios; a challenge solved considerably better by GMAS.

The model and experiments give rise to several questions for future work:

- In the experiments presented here, the objectives of agents and governance were strongly correlated. How can the approach be applied to an arbitrary combination of goals, and how do conflicts in the objective functions influence learning?
- What does an extension of the restriction policy to continuous action spaces look like?
- How do action space restrictions compare (empirically and theoretically) to other forms of governance, e.g., norms or inter-agent communication?
- Is the approach viable for asynchronous MAS (e.g., cyber-physical systems)?

Acknowledgements This work is supported by the German Federal Ministry for Economic Affairs and Energy (BMWi).

References

1. Abdunabi, T., Basir, O.: Holonic Intelligent Multi-Agent Algorithmic Trading System (HIMAATS). International Journal of Computers and their Applications (2014)
2. Aires, J.P., Meneguzzi, F.: Norm conflict identification using deep learning. In: AAMAS workshops (2017)
3. Aldewereld, H., Boissier, O., Dignum, V., Noriega, P., Padget, J. (eds.): Social coordination frameworks for social technical systems. Springer (2016)
4. Arcos, J.L., Esteva, M., Noriega, P., Rodríguez-Aguilar, J.A., Sierra, C.: Environment engineering for multiagent systems. In: Engineering Applications of Artificial Intelligence (2004)
5. Arcos, J.L., Rodríguez-Aguilar, J.A., Rosell, B.: Engineering autonomic electronic institutions. In: Engineering environment-mediated multi-agent systems: International workshop, EEMMAS 2007. Springer-Verlag, Berlin, Heidelberg (2008)
6. Balke, T., da Costa Pereira, C., Dignum, F., Lorini, E., Rotolo, A., Vasconcelos, W., Villata, S.: Norms in MAS: Definitions and Related Concepts. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik (2013), pages: 31

7. Barbuceanu, M.: Coordinating agents by role based social constraints and conversation plans. In: AAAI/IAAI (1997)
8. Boella, G., van der Torre, L., Verhagen, H.: Introduction to normative multiagent systems. Computational & Mathematical Organization Theory **12**(2) (2006)
9. Boella, G., van der Torre, L., Verhagen, H.: Introduction to the special issue on normative multiagent systems. Autonomous Agents and Multi-Agent Systems **17**(1) (2008)
10. Bou, E., López-Sánchez, M., Rodríguez-Aguilar, J.A.: Towards Self-configuration in Autonomic Electronic Institutions. In: Noriega, P., Vázquez-Salceda, J., Boella, G., Boissier, O., Dignum, V., Fornara, N., Matson, E. (eds.) Coordination, Organizations, Institutions, and Norms in Agent Systems II. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
11. Cacciamani, F., Celli, A., Ciccone, M., Gatti, N.: Multi-agent coordination in adversarial environments through signal mediated strategies. In: Proceedings of the 20th international conference on autonomous agents and MultiAgent systems. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2021)
12. Cheung, W.C., Simchi-Levi, D., Zhu, R.: Reinforcement Learning for Non-Stationary Markov Decision Processes: The Blessing of (More) Optimism. In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event. (2020)
13. Conitzer, V., Sandholm, T.: AWESOME: A General Multiagent Learning Algorithm that Converges in Self-Play and Learns a Best Response Against Stationary Opponents. Machine Learning **67** (2003)
14. Conte, R., Falcone, R., Sartor, G.: Introduction: Agents and Norms: How to fill the gap? Artificial Intelligence and Law **7**(1) (1999)
15. Esteva, M., Rodríguez-Aguilar, J., Arcos, J.L., Sierra, C., Noriega, P., Rosell, B., de la Cruz, D.: Electronic institutions development environment. In: AAMAS Demo Proceedings. vol. 3. International Foundation for Autonomous Agents and Multiagent Systems (2008)
16. Esteva, M., Rodríguez-Aguilar, J., Sierra, C., Calves, P., Arcos, J.L.: On the Formal Specification of Electronic Institutions. In: Agent Mediated Electronic Commerce, vol. 1991. Springer (2001)
17. Frantz, C., Pigozzi, G.: Modelling norm dynamics in multi-agent systems. Journal of Applied Logic **5** (2018)
18. Gomez-Sanz, J.J., Fuentes Fernandez, R.: Ingenias. In: Social Coordination Frameworks for Social Technical Systems. Springer (2016)
19. Greenwald, A., Hall, K.: Correlated-q learning. In: Proceedings of the twentieth international conference on international conference on machine learning. ICML'03, AAAI Press (2003)
20. Gronauer, S., Diepold, K.: Multi-agent deep reinforcement learning: a survey. Artificial Intelligence Review (2021)
21. Hwang, K., Jiang, W., Chen, Y.: Model Learning and Knowledge Sharing for a Multiagent System With Dyna-Q Learning. IEEE Transactions on Cybernetics **45**(5) (2015)
22. Liang, E., Liaw, R., Nishihara, R., Moritz, P., Fox, R., Goldberg, K., Gonzalez, J.E., Jordan, M.I., Stoica, I.: RLlib: Abstractions for distributed reinforcement learning. In: ICML (2018)
23. Littman, M.L.: Markov games as a framework for multi-agent reinforcement learning. In: Proceedings of the eleventh international conference on international con-

ference on machine learning. ICML'94, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1994)
24. Lopes Cardoso, H., Urbano, J., Rocha, A., Castro, A.J.M., Oliveira, E.: ANTE: A Framework Integrating Negotiation, Norms and Trust. In: Social Coordination Frameworks for Social Technical Systems, vol. 30. Springer (2016)
25. Marín-Lora, C., Chover, M., Sotoca, J.M., García, L.A.: A game engine to make games as multi-agent systems. Advances in Engineering Software **140** (2020)
26. McGroarty, F., Booth, A., Gerding, E., Chinthalapati, V.L.R.: High frequency trading strategies, market fragility and price spikes: an agent based model perspective. Annals of Operations Research **282**(1) (2019)
27. Mellema, R., Jensen, M., Dignum, F.: Social Rules for Agent Systems. In: Aler Tubella, A., Cranefield, S., Frantz, C., Meneguzzi, F., Vasconcelos, W. (eds.) Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems XIII. Springer International Publishing, Cham (2021)
28. Morales, J.: On-line norm synthesis for open Multi-Agent systems. Ph.D. thesis, Universitat de Barcelona (2016)
29. Neufeld, E., Bartocci, E., Ciabattoni, A., Governatori, G.: A Normative Supervisor for Reinforcement Learning Agents. In: Platzer, A., Sutcliffe, G. (eds.) Automated Deduction – CADE 28. Springer International Publishing, Cham (2021)
30. Noriega, P.: Agent-mediated auctions: The fishmarket metaphor. Ph.D. thesis, Universitat Autonoma de Barcelona (1997)
31. Noriega, P., Jonge, D.: Electronic Institutions: The EI/EIDE Framework. In: Social Coordination Frameworks for Socio-Technical Systems, vol. 30. Springer (2016)
32. Nowé, A., Vrancx, P., De Hauwere, Y.M.: Game Theory and Multi-agent Reinforcement Learning. In: Wiering, M., van Otterlo, M. (eds.) Reinforcement Learning: State-of-the-Art. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
33. Padakandla, S., K. J., P., Bhatnagar, S.: Reinforcement learning algorithm for non-stationary environments. Applied Intelligence **50**(11) (2020)
34. Riad, M., Golpayegani, F.: Run-time Norms Synthesis in Multi-Objective Multi-Agent Systems (2021)
35. Rizk, Y., Awad, M., Tunstel, E.: Decision Making in Multi-Agent Systems: A Survey. IEEE Transactions on Cognitive and Developmental Systems **PP**, 1–1 (2018)
36. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal Policy Optimization Algorithms (2017)
37. Shoham, Y., Tennenholtz, M.: On social laws for artificial agent societies: off-line design. Artificial Intelligence **73**(1) (1995)
38. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. A Bradford Book, Cambridge, MA, USA (2018)
39. Wang, X., Sandholm, T.: Reinforcement learning to play an optimal nash equilibrium in team markov games. In: NIPS (2002)
40. Weyns, D., Brückner, S., Demazeau, Y.: Engineering Environment-Mediated Multi-Agent Systems. Springer (2007)
41. Zaib, M., Sheng, Q.Z., Zhang, W.E.: A short survey of pre-trained language models for conversational AI-A NewAge in NLP (2021)
42. Zhang, K., Yang, Z., Basar, T.: Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms. In: Handbook of Reinforcement Learning and Control. Springer (2019)
43. Zinkevich, M., Greenwald, A., Littman, M.L.: Cyclic equilibria in markov games. In: Proceedings of the 18th international conference on neural information processing systems. NIPS'05, MIT Press, Cambridge, MA, USA (2005)