GENERATIVE INTEGRATION NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper presents an unbiased exploration framework for the belief state $p(s)$ in non-cooperative, multi-agent, partially-observable environments through differentiable recurrent functions. As well as single-agent exploration via intrinsic reward and generative RNNs, several researchers have proposed differentiable multi-agent communication models such as CommNet and IC3Net for scalable exploration through multiple agents. However, none of the existing frameworks so far capture the unbiased belief state in non-cooperative settings as with the nature due to biased examples reported from adersarial agents. *Generative integration networks* (GINs) is the first unbiased exploration framework insipired by honest reporting mechanisms in economics. The key idea is *synchrony*, an inter-agent reward to discriminate the honest reporting and the adversarial reporting **without real examples**, which is the different point from the GANs. Experimental results obtained using two non-cooperative multi-agent environments up to 20 agents denote that GINs show state-of-the-art performance in the exploration frameworks.

1 INTRODUCTION

Learning rich generative models for belief state $p(s)$ is a common objective of reinforcement learning in partially-observed environments (e.g., robot agents in the real world). Generative recurrent neural networks such as dynamic Boltzmann machines (Osogami and Otsuka, 2015), world models (Ha and Schmidhuber, 2018) and generative query networks (Eslami et al., 2018) have been introduced to capture the hidden Markov process by updating low-dimensional beliefs using high-dimensional observations. Intrinsic rewards such as cusiosity (Houthooft et al., 2016) encourage the agent *exploration* to find the optimal path by maximizing information gain. In multi-agent settings, *communication* such as CommNet (Sukhbaatar, Fergus, and others, 2016) and IC3Net (Singh, Jain, and Sukhbaatar, 2018) can also be combined for exploration in which agents cooperatively reduce uncertainty by integrating information such as observation and the beliefs.

However, in non-cooperative (i.e., competitive and mixed) multi-agent settings, the existing frameworks can barely achieve task-invariant belief states due to *adversarial reporting*. For instance, in an imperfect-information two-player zero-sum game as a pathologic case, an agent fools the opponent to take another action to "minimize" her return by reporting her fake information in the Bayesian Nash equilibriums, and vice versa. Namely, multi-layer perceptrons are vulnerable to adversarial examples (Goodfellow, Shlens, and Szegedy, 2014), which have perturbations combined to the real information. Hence, naive optimizations in multi-agent deep reinforcement learning fails to extract true knowledge, i.e., the belief state, from the environment. Furthermore, constructing discriminators as with generative adversarial networks (Radford, Metz, and Chintala, 2015) is impossible since we cannot directly draw samples from $p(s)$.

This paper proposes unbiased communication framework in non-cooperative settings inspired by honest reporting mechanisms in game theory. The first contribution is *synchrony*, a competitive intrinsic reward formulated by errors between individual reports and the integrated belief sent to each agent in every time-step. Intuitively, it encourages agents to predict other agents' beliefs. As we show in the paper, synchrony is a quasi-discriminator. That is, if the model is a fully-learned state, synchrony can discriminate the true distribution $p(s)$. As we show in the paper, the optimal policy to maximize synchrony is to report the information obtained from the environment without any perturbation. Although synchrony is also known as a proper scoring rule (Miller, Resnick, and Zeckhauser, 2005) in honest reporting mechanisms, our contribution is to apply it to the problem of adversarial reporting. As synchrony is zero-sum, it does not affect the total returns of the agents.

As the second contribution, we construct a muti-agent generative framework, the *generative integration network* (GIN) using synchrony. The exploration scenario in GIN is achieved by communication among n non-cooperative agents made of the controller and two additional modules, a differentiable **generator** G_i to send adversarial reporting and a shared **validator** V to receive differentiable reports and distribute synchrony to the other agents. At convergence, synchrony approaches zero and all generators draw samples from $p(s)$. This paper shows an implementation of GIN utilizing IC3Net (Singh, Jain, and Sukhbaatar, 2018), which is the state-of-the-art model for communication frameworks in non-cooperative environments. To confirm GINs learning belief states, we demonstrate numerical experiments with two non-cooperative, partially-observed environments up to 20 agents. We demonstrate that GINs records the state-of-the-art performance by outperforming existing frameworks for control under uncertainty such as recurrent neural networks and communication in non-cooperative multi-agent settings.

2 Related Work

Belief state $p(s)$ is a typical target of control under uncertainty. Generative recurrent neural networks can be used to abstract high-dimensional observation of low-dimensional belief. DyBM (Osogami and Otsuka, 2015) is a bio-inspired generative RNN that captures beliefs with seven neurons using the Boltzmann machine inpired by STDP mechanisms. World models (Ha and Schmidhuber, 2018) and GQN (Eslami et al., 2018) use variational autoencoders to capture a belief. Consciousness prior (Bengio, 2017) is a discrete hierarchical recurrent model to capture the symbols from the environment. There are several variations in terms of curiosity for exploration, such as prediction error for observation (Pathak et al., 2017) and information gain (Houthooft et al., 2016). Prediction error of observation has a noisy TV problem (Azar et al., 2019), which yields positive rewards for non-useful information. Information gain is vulnerable to adversarial reporting since false reports have more information gain than true information.

Typical utility of multi-agent reinforcement learning (MARL) is a variation reduction. MADDPG (Celikyilmaz et al., 2018) employs actor-critic to control multiple agents in a policy gradient manner and aggregates TD-error to the centralized critic. MADDPG-GCPN (Ryu, Shin, and Park, 2018) is an agent-communication model to estimate state with centralized training. RIAL (Foerster et al., 2016) uses discrete variables and Q-learning to control which to send other agents. Zhang uses graph structure to constraint who can receive the report (Zhang et al., 2018). CommNet (Sukhbaatar, Fergus, and others, 2016) uses differentiable continuous vectors and optimizes reports with backpropagation.

The closest work to this paper is IC3Net (Singh, Jain, and Sukhbaatar, 2018), which extends CommNet for non-cooperative environments. IC3Net controls when to communicate through binary action of each agent, and stops the agent from sending information to reduce their expected returns. As both CommNet and IC3Net do not guarantee sending of task-invariant beliefs, the agent learns adversarial attacks to fool other agents. The first introduced adversarial example is the fast gradient sign method (Goodfellow, Shlens, and Szegedy, 2014), which adds perturbation to the true samples. There are at least two directions to improve robustness against adversarial examples. One is to change either network topology or the optimizer (Carlini and Wagner, 2017; Cisse et al., 2017; Fawzi, Fawzi, and Frossard, 2018). The other is to create discriminators to classify adversarial examples and true samples (Goodfellow et al., 2014). As the discriminator should assume that the optimizer can draw samples from the true distribution, it cannot be applied to latent beliefs.

Counterfactual reward (Agogino and Tumer, 2006) is an intrinsic reward used to deal with credit assignment problems in multi-agent settings. COMA (Foerster et al., 2018) extends counterfactual rewards with an actor-critic optimizer. NaaA (Ohsawa et al., 2018) proposes counterfactual returns that take summation over times. The constraint of counterfactual reward is only applied to discrete actions, which prevent us from applying it to continuous communication.

In the field of security and distributed computing, the algorithm to treat adversarial reporting is called Byzantine Fault Tolerance (BFT). Paxos (Lamport and others, 2001) and practical BFT (Castro, Liskov, and others, 1999) employ consensus mechanisms with multi-step communication to make the agent converge on the true information. Prediction market (Barbu and Lay, 2012) is used for aggregating reports for uncertain information such as future observation. Our mechanism brings the truthful mechanism to multi-agent control problems.

3 Problem Definition

Before introducing adversarial reporting and synchrony, we formulate multi-agent communication in partially-observed environments from the perspective of exploration as the problem setting.

3.1 Partially Observable MDP

The definition starts with a single-agent model in a partially observable MDP (POMDP). An *environment* is a seven-tuple $\langle S, A, r, \gamma, T, \Omega, O\rangle$ where S is a discrete state space, A is a descrete action space, $r : S \times A \to \mathbb{R}$ is a reward function, $T : S \times A \times S \to [0, 1]$ is a state-transition probability, Ω is a high-dimensional continuous observation space (e.g., images) and $O : S \times \Omega \to [0, 1]$ is an observation probability. An *agent* is a two-tuple $\langle \pi, \xi_t\rangle$ where $\pi(a_t|\xi_{t-1})$ is a probabilistic policy and $\xi_t := \langle o_1, a_1, r_1, \ldots, o_t, a_t, r_t\rangle$ is a history. At each time step t, an agent receives an observation $o_t \sim O(o|s_t)$, sends action $a_t \sim \pi(a|\xi_t)$, gets rewarded $r_t := r(s_t, a_t)$, and integrate to a history $\xi_t = \langle \xi_{t-1}, o_t, a_t, r_t\rangle$. An environment transition state $s_{t+1} \sim T(s|s_t, a_t)$.

In contrast to MDP, an agent in POMDP cannot observe the true state s_t. Instead, the agent update belief of the state $p(s_t|\xi_t)$ under the situation ξ_t. Hence, an agent maximizes its expected return

$$J^\pi_\theta(\xi_t) := \int_S V^\pi(s) dp_\theta(s|\xi_t) \tag{1}$$

where $V^\pi(s)$ is a value function and $p_\theta(s)$ is a belief state model with a prior θ. As there is the optimal policy π^* to maximize $V^{\pi^*}(s)$ in MDP if both S and A is descrete, our target is obtaining belief $p(s)$. inferring $p(s|\xi_t)$. As the observation o_t is high dimensional variable, the exisisting probabilistic recurrent neural networks such as dynamic Boltzman machines are intractable.

3.2 From Exploration to Communication

Our goal is to obtain the generative model for the belief state $p_\theta(s|\xi_t)$ through rich deteraministic functions (e.g., neural networks). What is different from the existing deteraministic generative models (Radford, Metz, and Chintala, 2015) is that the error with the real sample s is not observable, and there are multiple solutions for $p(s)$ because s has $|\dim S|!$ freedoms for permutation . Hence, we employ an entropy minimization framework $\min_\xi \mathbb{H}[s|\xi] = \mathbb{H}[s]$ for exploration (Houthooft et al., 2016). A finite-horizon entropy minimization framework maximizes total information gain $\sum_{t=1}^{\tau}[\mathbb{H}[s|\xi_t] - \mathbb{H}[s|\xi_{t-1}]]$ after τ-steps. As the path-finding problem is intractable, reinforcement learning is used for control. Typically, it employs *curiosity* (Houthooft et al., 2016) as the intrinsic reward,

$$r'(s_t, a_t, o_{t+1}) = r(s_t, a_t) + \eta I(o_{t+1}; \xi_t), \tag{2}$$

where $I(o_{t+1}; \xi_t) := \mathbb{H}[s|\xi_{t+1}] - \mathbb{H}[s|\xi_t]$ is the curiosity in each time step, and $\eta > 0$ is a hyperparameter to control the exploration-exploitation tradeoffs. We can naturally enhance the curiosity-driven exploration of the n-agent environments,

$$\mathbf{r}'(s_t, \mathbf{a}_t, \mathbf{o}_{t+1}) = \mathbf{r}(s_t, \mathbf{a}_t) + \eta I(\mathbf{o}_{t+1}; \boldsymbol{\xi}_t), \tag{3}$$

where bold symbols are n-dimensional vectors for joint variables. We denote $i \in \{1, \ldots, n\}$ as the index of an agent. For instance, ξ^i_t is an i-th agent's history.

Communication is a sequential process of reporting $z^i_t \sim p(z|\xi^i_t)$ and integration $p(s|\mathbf{z}_t)$. For instance, CommNet (Sukhbaatar, Fergus, and others, 2016) uses an LSTM cell for reporting and uses a mean field approximation as an integrator $p(s|\mathbf{z}_t) = (1/n)\sum_{i=1}^n p(s|z^i_t)$. As all the agent observe the environment through the reporting $\mathbf{z}_t$, we can define curiosity in terms of comunication. The relationship between exploration and communication can be written as follows,

$$I(\mathbf{z}_{t+1}; \boldsymbol{\xi}_t) = \mathbb{H}[s|\boldsymbol{\xi}_t] - \mathbb{H}[s|\mathbf{z}_{t+1}, \boldsymbol{\xi}_t]. \tag{4}$$

3.3 Adversarial Reporting

If the environment is non-cooperative, i.e., $\mathbf{r}$ is non-monotonic, the Bayesian Nash equilibrium in multi-agent communication cannot maximize the total return. Considering an imperfect-information, zero-sum, two-player game, the greedy optimization problem can be written as the following mini-max problem:

$$\min_{a_t^1, z_t^1} \max_{a_t^2, z_t^2} : J(\mathbf{a}_t, \mathbf{z}_t; \boldsymbol{\xi}_{t-1}) := \int_S Q(s, \mathbf{a}_t) dp(s|\mathbf{z}_t, \boldsymbol{\xi}_{t-1}). \tag{5}$$

where $Q(s_t, \mathbf{a}_t)$ is a joint state-action value function. Note that agent 1 can control agent 2's action by sending information z_t^1. In this case, agent 1 can draw false information from reporting policy $\rho(z_t^1|\xi) = \int \rho(z_t^1|z) dp(z|\xi_t^1)$ instead of honest reporting $p(s|\xi_t^1)$. An instance of adversarial reporting is an adversarial example with a fast gradient sign method (Goodfellow, Shlens, and Szegedy, 2014) that adds perturbation $\nu = \operatorname{sign}(\nabla_z J)$ to the true information: $\rho(\hat{z}|z) = z + \epsilon\nu$ where $\epsilon > 0$ is the size of the perturbation. Reporting bias $\mathbb{H}[s|\epsilon\nu]$ prevented us from obtaining true information $\mathbb{H}[s]$.

In this case, curiosity could not be used since it also encourages the reporting bias $\mathbb{H}[s|\rho] - \mathbb{H}[s]$.

$$I(\mathbf{z}_{t+1}; \boldsymbol{\xi}_t, \rho) - I(\mathbf{z}_{t+1}; \boldsymbol{\xi}_t) = \mathbb{H}[s|\mathbf{z}_{t+1}, \boldsymbol{\xi}_t, \rho] - \mathbb{H}[s|\mathbf{z}_{t+1}, \boldsymbol{\xi}_t] \tag{6}$$

4 Proposed Method

4.1 Synchrony

Our approach is to distribute an intrinsic reward to the agents to encourage honest reporting. In addition to the reward from the environemnt, the validator destributes each agent a desined reward, *synchrony*, inspired by honest reporting mechanisms in economics. Synchrony uses the characteristic condition $\mathbb{H}\left[z|\xi_t^i\right] - \mathbb{H}\left[s|\boldsymbol{\xi}_t\right] = 0$ at the optimally $\mathbb{H}\left[z|\xi_t^i\right] = \mathbb{H}[s]$ for all i's. Synchrony is defined as the following equation:

$$u_i(z_t^i|\mathbf{z}_t) = -D_{\mathrm{KL}}\left(p(z_t^i|\xi_t^i) \,||\, p(s|\mathbf{z}_t)\right) + \frac{1}{n} D_{\mathrm{KL}}\left(p(\mathbf{z}_t|\boldsymbol{\xi}_t) \,||\, p(s|\mathbf{z}_t)\right), \tag{7}$$

where the first term is the variance of i's reporting from the integrated report $p(s|\mathbf{z}_t)$, and the second term is the bias between the joint distribution and the integrated report. Synchrony is a zero-sum reward because

$$\sum_{i=1}^{n} D_{\mathrm{KL}}\left(p(z_t^i|\xi_t^i) \,||\, p(s|\mathbf{z}_t)\right) = \sum_{i=1}^{n} \int \log \frac{p(z_t^i|\xi_t^i)}{p(s|\mathbf{z}_t)} dp(s|\mathbf{z}_t)$$

$$= \int \log \frac{p(\mathbf{z}_t|\boldsymbol{\xi}_t)}{p(s|\mathbf{z}_t)} dp(s|\mathbf{z}_t) = D_{\mathrm{KL}}\left(p(\mathbf{z}_t|\boldsymbol{\xi}_t) \,||\, p(s|\mathbf{z}_t)\right). \tag{8}$$

Intuitively, it encourages agents to report more unbiased beliefs by predicting other agents' beliefs. This is also known as a proper scoring rule (Miller, Resnick, and Zeckhauser, 2005) in the honest reporting mechanism. Fig. 1 illustrates synchrony in a binary state $S = \{0, 1\}$.

4.2 Generative Integration Networks

The intrinsic reward naturally leads us to construct a game-theoric generative framework for controlling problems. *generative integration network* (GIN). We utilize IC3Net (Singh, Jain, and Sukhbaatar, 2018), which is the state-of-the-art model for communication frameworks in non-cooperative environments. Fig. 2 shows a network structure. The network repeats k sampling iterations in a step until it obtains the final belief $p(s|\xi_t)$ on the basis of MCMC. In every j-th phase, each **generator** G_i forwarded the following maps:

$$\begin{aligned} m_{t,j}^i &= \sigma(W^i(p_{t,j-1}^s + \epsilon\nu) + H^i(h_{t,j}^i)), \\ g_{t,j}^i &= f^i(m_{t,j}^i), \;\; h_{t,j}^i = H^i(h_{t,j-1}^i), \\ z_{t,j}^i &= m_{t,j}^i g_{t,j}^i + \frac{1}{2}(1 - g_{t,j}^i), \end{aligned} \tag{9}$$

where $\nu \sim B(1/2)^m$ is a binary noise and f_i is a binary action to decide whether to send a true report $p(s|\xi_t, \nu)$ or a fully false report $p(s|\nu)$. The action is trained by REINFORCE (Williams, 1992).

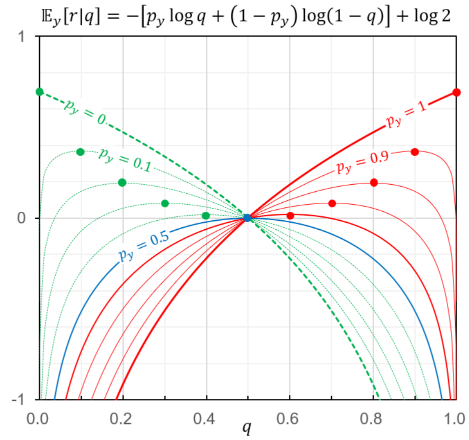


Figure 1: synchrony in a binary state $s \in \{0, 1\}$. We can confirm that the optimal policy for maximizing synchrony is to report the information obtained from the environment without any perturbation. The characteristics can be extended to m-bit tapes $S = \{0, 1\}^m$ with the naive Bayes model $p(s) = \prod_i p(s_i)$, a practical model representing word distribution (McCallum, Nigam, and others, 1998). In this case, each bit can be represented as events, words or symbols. Namely, the optimal policy for the uninformed agent (blue) is to report $z = 1/2$, which maximizes entropy $\mathbb{H}[s] = \log 2$.

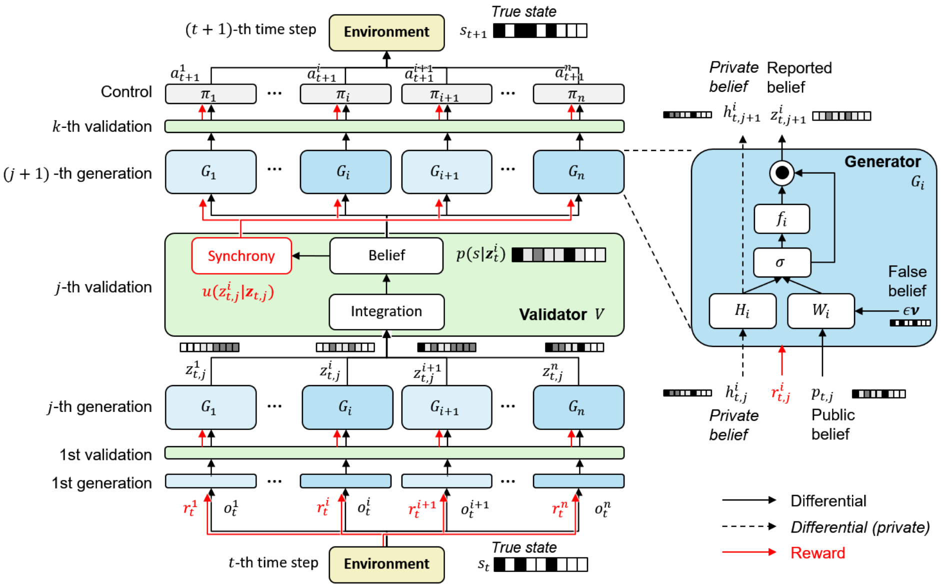
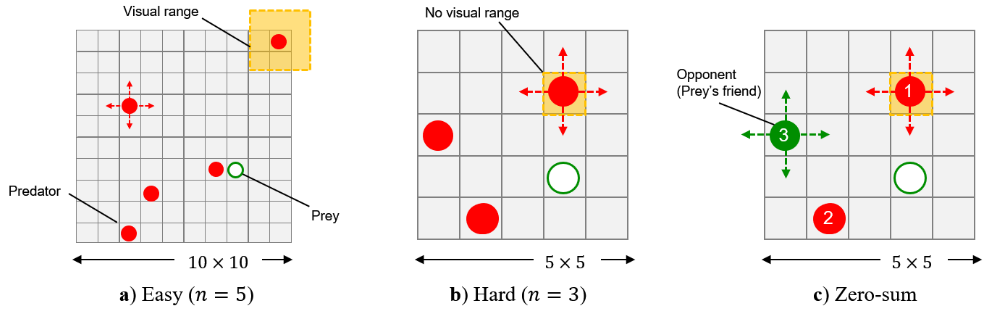


Figure 2: The generative integration networks. The exploration scenario in GIN is achieved by communication among n non-cooperative agents made of the controller and two additional modules, a differentiable **generator** G_i to send adversarial reporting, and a shared **validator** V to receive the differentiable reports and distribute synchrony to the other agents. At convergence, synchrony became zero and all the generators drew samples from $p(s)$. The exploration mechanism works without any other intrinsic reward such as curiosity, which has negative relation to synchrony.

The shared **validator** V receives reports $z_{t,j}^i$, estimated integrated distribution, and distributed syn-



a) Easy (n = 5) **b**) Hard (n = 3) **c**) Zero-sum

Figure 3: An illustration of the predator prey (PP) tasks in three difficulties. Each agent continuously receives a reward signal -0.05 in each timestep until the arrival on the prey. After the predators reaches their goals, they receive a competitive reward $1/m$ that m is the number of predators who reached the prey. Every episode ends in fixed steps. Although communication is used to tell the position of prey, predators could fool other predators to corner the prey.

chrony.

$$\begin{aligned} p^s_{t,j} &= \frac{1}{n}\sum_{i=1}^{n} z^i_{t,j}, \quad b^s_{t,j} = \mathbf{1}(p^s_{i,j} \geq 1/2), \\ r^i_{t,j} &= -[b^s_{t,j}\log p^s_{t,j} + (1-b^s_{t,j})\log(1-p^s_{t,j})], \\ u^i_{t,j} &= r^i_{t,j} - \frac{1}{n}\sum_{i=1}^{n} r^i_{t,j}, \end{aligned} \tag{10}$$

After k iterations, the final belief is a consensus to $p^s_t = p^s_{t,k}$, and each agent drew actions from the controllers.

$$a^i_t \sim \pi(\cdot|p^s_t) = \mathrm{softmax}\, W^i(p^s_t). \tag{11}$$

5 Experiments

To confirm that GIN learns belief states, we demonstrate numerical experiments with two non-cooperative, partially-observed environments, each with three difficulties up to 20 agents. We demonstrate that GINs outperform existing methods such as VIME, CommNet, and IC3Net by learning states.

5.1 Experimental Settings

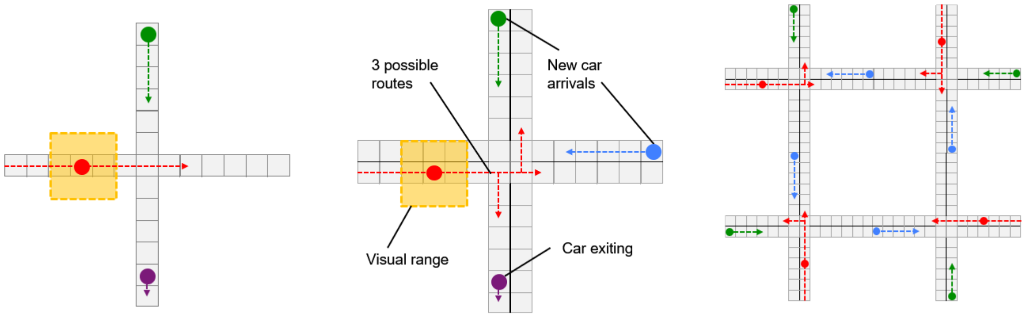
In the experiment, we use two environments, Predator-Prey (PP) and Traffic-Junction (TP). We train the model in 2,000 epochs with 500 steps. The details of every tasks are show in Figure 3 and 4.

5.1.1 Predator-Prey (PP)

Predator-prey (PP) is a multi-agent limited-sight task in which predators explores for prey at the randomly initialized point in the grid world. PP is a widely used benchmark of MARL (Barrett, Stone, and Kraus, 2011; Sukhbaatar, Fergus, and others, 2016; Singh, Jain, and Sukhbaatar, 2018) The state space is a gridworld (Sutton, Barto, and others, 1998), and the prey is at the goal. The sight of the predators is limited so that they could only observe around a few blocks. Hence, the predators should have communicated to other predators so they would know the position of the prey and visited areas.

5.1.2 Traffic-Junction (TJ)

Traffic-junction (TJ) is a simplified traffic junction in which n cars with limited-sight exchange their positions to avoid collision. We also vary the difficulty of TJ with three modes. In the easy



a) Easy ($n = 5$) **b**) Medium ($n = 10$) **c**) Hard ($n = 20$)

Figure 4: An illustration of the traffic junction (TJ) tasks in three difficulties. Each car had two actions, ”accelerate” to proceed 1 step and ”brake” to stop. At the initial state, each car is given the starting point and destination point, and is instructed to run the path as fast as possible by avoiding collisions. To incentivize running faster, they receives a negative reward -0.05 in each time step. After reaching their destination, the agent receives 0. If two cars would collide, both receives a negative reward -1.To avoid collision, it is important to check if the other cars are reaching each other by multi-agent communication. This is similar to sending a winker and brake pump in the real world. The difficulty is that the setting is not monotonically cooperative. Hence, the agents sent stop-signals to other agents to go as fast as possible.

mode, they solve the task for interchanging two orthogonal one-way roads. In the second difficulty (medium), there is two-way traffic, and each car could go straight as well as turning left or right. In the most difficult mode (hard), there are two parallel two-way traffic streams with four junctions.

5.2 Baselines

We compare GINs to the existing exploration method such as recurrent neural networks, intrinsic models, and multi-agent communication through learning states.

- LSTM: the individually controlled agents that had a recurrent neural network to obtain the state. We confirme that multi-agent communication enhanced exploration ability by exchanging information.
- CommNet (Sukhbaatar, Fergus, and others, 2016): A multi-agent communication method assumed cooperative settings. The baseline is used for the cooperative since the model fails to obtain true belief due to adversarial communication.
- IC3Net (Singh, Jain, and Sukhbaatar, 2018): State-of-the-art communication method in non-cooperative settings. It had a gate to control “when to communicate” to deal with non-cooperative rewards.

We use GIN as well as *Curious GIN*, which use negative hyperparamter $\eta = -10.0$.

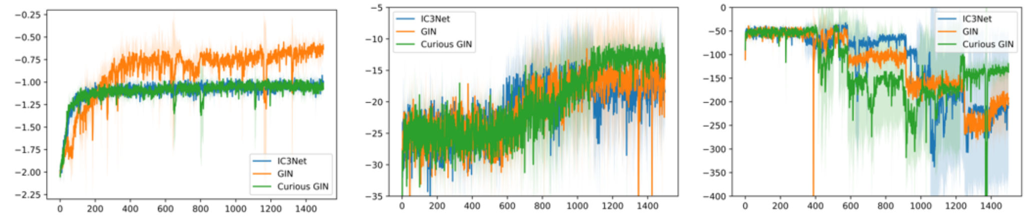
5.3 Experimental Result

Table 1 shows the experimental results in five tasks, PP-easy, -hard, TJ-easy, -medium, and -hard. We can confirm that GIN and the its variation record the stat-of-the-art performance in all the five tasks. The reason why Curious GIN records the best result in cooperative task is there are few adversarial attacks because the taks is not competitive but mixed task. The improvement becomes higher in the harder tasks. We also show the learning curves for the harder tasks in Figure 5.

The learning curves of synchrony and fractions of true reporting are show in Figure 6. Notice that synchrony is a zero-sum intrinsic reward, the mean value is always zero, and the deviation varies. We can confirm that synchrony validaete the adversarial attacks to send negative reward. 6 (c) shows the fractions of true reporting. We can see that synchrony make agents to sending the true infomration.

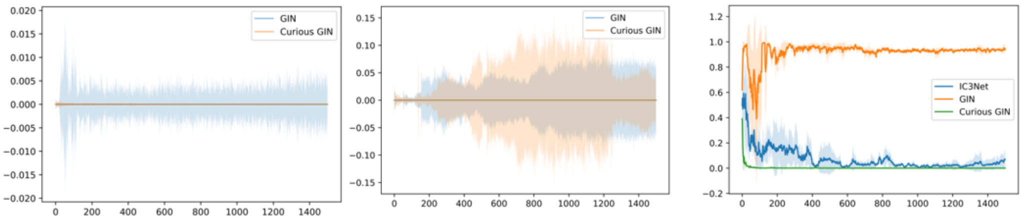
Table 1: Comparison of return in each baselines. The experiment repeats 3 times, and mean value and standard deviations are written. The **bold** texts indicates the best score, and the *italic* text indicates the second best.

	PP (competitive)		TJ (mixed)		
	easy	hard	easy	medium	hard
LSTM	-4.97 ± 1.33	-1.92 ± 0.35	-22.32 ± 1.04	-47.91 ± 41.2	-819.97 ± 438.7
CommNet	-4.30 ± 1.14	-1.54 ± 0.33	-6.86 ± 6.43	-26.63 ± 4.56	-463.91 ± 460.8
IC3Net	*-2.44 ± 0.18*	*-1.03 ± 0.06*	*-4.35 ± 0.72*	-17.54 ± 6.44	-216.31 ± 131.7
GIN	**-2.34** ± **0.21**	**-0.69** ± **0.14**	-5.39 ± 4.14	*-13.30 ± 4.98*	*-195.29 ± 58.46*
Curious GIN	-2.61 ± 0.24	-1.04 ± 0.06	**-3.93** ± **1.46**	**-12.83** ± **2.50**	**-132.60** ± **17.91**



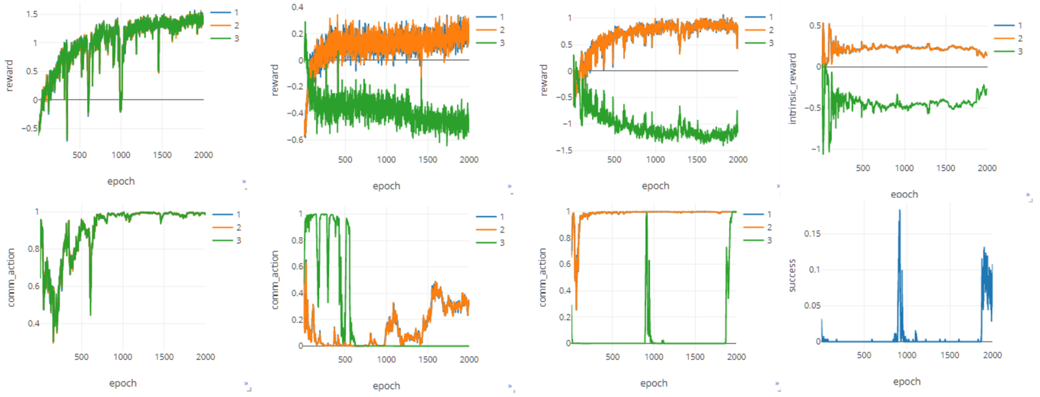
a) PP-hard **b)** TJ-medium **c)** TJ-hard

Figure 5: Comparison in three methods for learning curves in the three harder tasks (PP-hard, TJ-medium, TJ-hard) .



a) Synchrony (PP-hard) **b)** Synchrony (TJ-hard) **c)** True reporting (PP-hard)

Figure 6: Details of the intrinsic reward and truthful reporting. Note that synchrony is relatively lower than the external reward.



a) IC3 ($f = 0$) **b)** IC3Net ($f = 1$) **c)** GIN ($f = 1, \eta = 10.0$)

Figure 7: A learning curve in a zero-sum predator-prey task (n=3).

We show a comparison of IC3Net and GIN in fully-adversarial PP with f opponents in Fig. 7. In the game, the maximum return is 1.5 in $f = 0$. From the perspective of the frequency of communication, all the agents communicated at the convergence. In the setting of $f = 1$, although the expected reward is expected to be 1.0, the reward of IC3Net decreases to 0.2. To confirm the frequency, the opponent $f = 1$ sending fake reports and the other two agents learn that the channel is not informative, and decide not to communicate with each other. After that, the opponent had no incentive to send information; at this point, the two agents began to communicate. On the other hand, the maximum reward of GIN is 1.0.

To see frequency, the channel is used by the two honest agents. Seeing the synchrony in Fig. 7, the honest agents gained the positive reward, and the opponent, the negative reward. As fake information is detected by the validators, the opponent learned to not send information to the other agents. Interestingly, the opponent finally learned to send information about the prey. To see the behavior at this point, the success rate at which all the agents reached their prey is in the range of 5%-10%. This indicated that the opponent also reached the prey, and reported the informative belief $p(s)$ to increase synchrony. Thus, we confirmed that GIN learns belief state through interaction between the generators and the validator.

6 Future Directions

Before concluding, we discuss the limitations to showing the future direction. As far as we know, there are several extensions of GIN to complement the drawbacks.

- *Discrete-reporting GIN* Although GIN assumes a continuous vector, the method cannot be applied to the case wherein the system allows discrete reporting for some reason. For instance, Pommerman (Resnick et al., 2018) uses NeurIPS'19 competition, provided by Facebook AI, allowing only discrete reporting. In real-world traffic, communication between cars is achieved by bits such as the winker and brake pump. In these cases, the network structure should have changed into one that optimized by Q-learning such as RIAL (Foerster et al., 2016).
- *Continuous-state GIN* As the implementation in synchrony assumes a discrete state space, it should enhance continuous state space as in normal distributions. Furthermore, the reader can refer to other proper scoring rules (Miller, Resnick, and Zeckhauser, 2005) such as quadratic and spherical rules.
- *No-reward GIN* GIN has a pathologic solution to exchanging constants with each other without assuming an external reward. To deal with the problem, one can use error prediction for observation.
- *Conditional GIN* As synchrony assumes all messages are obtained independently, it enhances the report that shares weights. It means the validator is vulnerable to civil attack where the adversary attacks with majorities. To defend the civil attack, Bayesian consensus models (Morris, 1974; Winkler, 1981) can be introduced. There are several methods to estimate reliability of agents (Morris, 1974) and calculate the correlation between reports (Winkler, 1981).
- *Exploration over multiple Nash equilibriums* Our work assumed there is one Nash equilibrium that maximizes total return of the agents. Although in a single-agent finite discrete state MDP the agent has one optimal policy $\pi^* = \mathrm{argmax}_\pi(Q^\pi(s_0, \cdot))$, in multi-agent settings, the characteristics do not generally hold. The policy profile $\pi = (\pi_1, \cdots, \pi_n)$ converges on several Bayesian Nash equilibria depending on the initial value, and there is no guarantee that any equilibrium maximizes total returns.
- *Asynchronous GIN* GIN assumes that all agents repeat k-iterations in a step. This assumption constrains the protocols in the case where a time-step is very short, such as 60 fps in TV games. In such cases, asynchronous mechanisms are needed in which the belief is shared if several agents cannot respond within a time-out. Several asynchronous policy gradients such as A3C (Mnih et al., 2016) can be applied in such situations.

7 Conclusion

What has not been achieved in prior models of multi-agent reinforcement learning is task-invariance in non-cooperative settings. This paper points out adversarial reporting in non-cooperative communication. Learning the belief state $p(s)$ is a common objective for exploring partially observed spaces. However, the existing frameworks can barely achieve belief states in non-cooperative settings due to adversarial reporting. Our goal was to introduce a game-theoretic intrinsic reward and synchrony, inspired by honest reporting mechanisms in economics. The intrinsic reward naturally led us to construct a game-theoretic generative framework for reinforcement learning, the generative integration network (GIN). We demonstrated that GINs outperform existing frameworks for control under uncertainty such as recurrent neural networks, intrinsic reward, and communication in non-cooperative multi-agent settings.

REFERENCES

Agogino, A. K., and Tumer, K. 2006. Quicr-learning for multi-agent coordination. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, 1438.

Azar, M. G.; Piot, B.; Pires, B. A.; Grill, J.-B.; Altché, F.; and Munos, R. 2019. World discovery models. *arXiv preprint arXiv:1902.07685*.

Barbu, A., and Lay, N. 2012. An introduction to artificial prediction markets for classification. *Journal of Machine Learning Research* 13(Jul):2177–2204.

Barrett, S.; Stone, P.; and Kraus, S. 2011. Empirical evaluation of ad hoc teamwork in the pursuit domain. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, 567–574. International Foundation for Autonomous Agents and Multiagent Systems.

Bengio, Y. 2017. The consciousness prior. *arXiv preprint arXiv:1709.08568*.

Carlini, N., and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, 39–57. IEEE.

Castro, M.; Liskov, B.; et al. 1999. Practical byzantine fault tolerance. In *OSDI*, volume 99, 173–186.

Celikyilmaz, A.; Bosselut, A.; He, X.; and Choi, Y. 2018. Deep communicating agents for abstractive summarization. *arXiv preprint arXiv:1803.10357*.

Cisse, M.; Bojanowski, P.; Grave, E.; Dauphin, Y.; and Usunier, N. 2017. Parseval networks: Improving robustness to adversarial examples. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 854–863. JMLR. org.

Eslami, S. A.; Rezende, D. J.; Besse, F.; Viola, F.; Morcos, A. S.; Garnelo, M.; Ruderman, A.; Rusu, A. A.; Danihelka, I.; Gregor, K.; et al. 2018. Neural scene representation and rendering. *Science* 360(6394):1204–1210.

Fawzi, A.; Fawzi, O.; and Frossard, P. 2018. Analysis of classifiers robustness to adversarial perturbations. *Machine Learning* 107(3):481–508.

Foerster, J.; Assael, Y.; de Freitas, N.; and Whiteson, S. 2016. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, 2137–2145.

Foerster, J. N.; Farquhar, G.; Afouras, T.; Nardelli, N.; and Whiteson, S. 2018. Counterfactual multi-agent policy gradients. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Ha, D., and Schmidhuber, J. 2018. World models. *arXiv preprint arXiv:1803.10122*.

Houthooft, R.; Chen, X.; Duan, Y.; Schulman, J.; De Turck, F.; and Abbeel, P. 2016. Vime: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems*, 1109–1117.

Lamport, L., et al. 2001. Paxos made simple. *ACM Sigact News* 32(4):18–25.

McCallum, A.; Nigam, K.; et al. 1998. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, 41–48. Citeseer.

Miller, N.; Resnick, P.; and Zeckhauser, R. 2005. Eliciting informative feedback: The peer-prediction method. *Management Science* 51(9):1359–1373.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, 1928–1937.

Morris, P. A. 1974. Decision analysis expert use. *Management Science* 20(9):1233–1241.

Ohsawa, S.; Akuzawa, K.; Matsushima, T.; Bezerra, G.; Iwasawa, Y.; Kajino, H.; Takenaka, S.; and Matsuo, Y. 2018. Neuron as an agent.

Osogami, T., and Otsuka, M. 2015. Learning dynamic boltzmann machines with spike-timing dependent plasticity. *arXiv preprint arXiv:1509.08634*.

Pathak, D.; Agrawal, P.; Efros, A. A.; and Darrell, T. 2017. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 16–17.

Radford, A.; Metz, L.; and Chintala, S. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Resnick, C.; Eldridge, W.; Ha, D.; Britz, D.; Foerster, J.; Togelius, J.; Cho, K.; and Bruna, J. 2018. Pommerman: A multi-agent playground. *arXiv preprint arXiv:1809.07124*.

Ryu, H.; Shin, H.; and Park, J. 2018. Multi-agent actor-critic with generative cooperative policy network. *arXiv preprint arXiv:1810.09206*.

Singh, A.; Jain, T.; and Sukhbaatar, S. 2018. Learning when to communicate at scale in multiagent cooperative and competitive tasks. *arXiv preprint arXiv:1812.09755*.

Sukhbaatar, S.; Fergus, R.; et al. 2016. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*, 2244–2252.

Sutton, R. S.; Barto, A. G.; et al. 1998. *Introduction to reinforcement learning*, volume 2. MIT press Cambridge.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4):229–256.

Winkler, R. L. 1981. Combining probability distributions from dependent information sources. *Management Science* 27(4):479–488.

Zhang, K.; Yang, Z.; Liu, H.; Zhang, T.; and Başar, T. 2018. Fully decentralized multi-agent reinforcement learning with networked agents. *arXiv preprint arXiv:1802.08757*.