IQ-Flow: Mechanism Design for Inducing Cooperative Behavior to Self-Interested Agents in Sequential Social Dilemmas

Bengisu Güresti, Abdullah Vanlıoğlu, Nazım Kemal Üre

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023)

June 2, 2023

Outline

Motivation and Approach

- Problem: Ensure cooperation of individually trained agents in a shared multi-agent environment
- Individually trained agents are self-interested $\rightarrow$ social dilemmas
- We consider multi-agents learning independently with reinforcement learning in sequential social dilemma environments
- Introduce a mechanism to incentivize all agents according to the state and taken actions
- Our goal is to remove the social dilemma from the environment via the external incentivizing mechanism
- Accomplish the goal without knowledge of how agents learn

Problem Environments - Iterated Matrix Games

Table 1: Prisoner's Dilemma

PD	C_2	D_2
C_1	(3, 3)	(0, 4)
D_1	(4, 0)	(1, 1)

Table 2: Chicken Game

Chicken	C_2	D_2
C_1	(3, 3)	(1, 4)
D_1	(4, 1)	(0, 0)

Table 3: Stag Hunt

Stag Hunt	C_2	D_2
C_1	(4, 4)	(0, 3)
D_1	(3, 0)	(1, 1)

Problem Environments - N-Player Escape Room

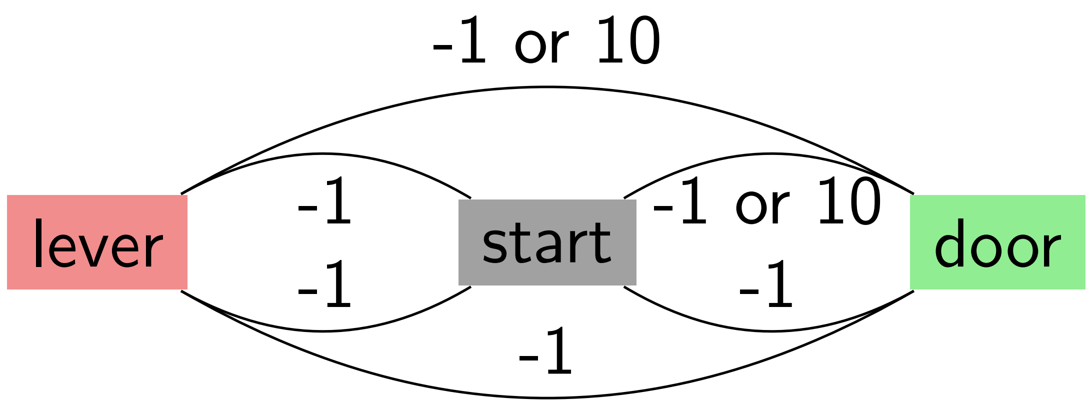


Figure 1: The N-player *Escape Room* game ER(N, M) [1].

States: Lever, Start, Door

If fewer than M agents pull the lever, all agents get -1 for changing states. Otherwise, the agent(s) that change state to door get $+10$ end the episode.

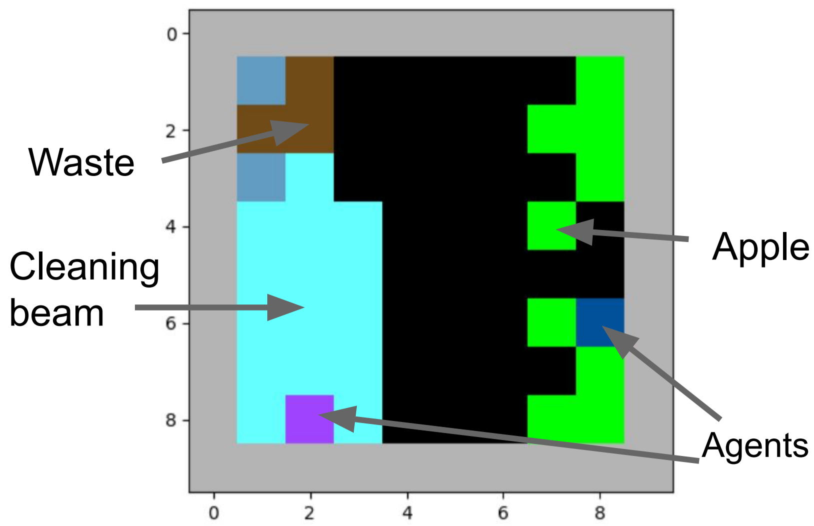


Figure 2: *2 Player Cleanup* (10×10 map) [1]: apple spawn rate decreases with increasing waste, which agents can clear with a cleaning beam. ID and we use (7×7) version of this map.

7×7 version of this map is used in this work.

Related Work - Mostly Focused

- Adaptive Mechanism Design (AMD) [2]
  - Based on estimating effect of incentives on the learning update of agents
  - Uses a first-order Taylor expansion for this process
  - Evaluated on iterated matrix games
  - Full access to agents' policy parameters by mechanism - opponent modelling proposed in case access is not possible
- Incentive Designer (ID) [3]
  - Based on estimating effect of incentives on the learning update of agents
  - Uses meta-gradients with online cross validation for this process
  - Evaluated on Escape Room, 2 Player Cleanup, and Gather-Trade-Build environments
  - Full access to agents' policy parameters by mechanism - opponent modelling as solution in case access is not possible
  - We use our re-implementation to use for comparison

Core Contributions

- Proposing to focus on removing the underlying dilemma from the system instead of focusing on how agents learn and update their policies
- Proposing to detect and infer the dilemma in the system and the cooperative policy using offline Reinforcement Learning with replay buffer
- Removing the requirement of accessing or making assumptions on agents' internal learning state and policy parameters for incentive design. But we still need the full data gathered by the agents.
- Removing the requirement of cost regularization for meta-gradient based incentive design in SSDs (nevertheless cost regularization still increases performance)
- Although agents are trained online, continually learning with changing incentives from mechanism, mechanism is trained offline with a replay buffer to make use of past data and not forget the previous policies that were detected to defective

Social Dilemma Conditions

Table 4: Matrix Game payoff table

	C	D
C	R, R	S, T
D	T, S	P, P

According to preliminary work in social dilemmas [4], [5], a Matrix Game such as Table 4 is a Social Dilemma if it satisfies the following conditions:

1. $R > P$
2. $R > S$
3. $2R > T + S$
4. $T > R$ or $P > S$

We aim to **reverse the 4th condition** to remove the dilemma.

IQ-Flow Pseudocode

Algorithm 1 Incentive Q-Flow

procedure TRAIN IQ-FLOW MECHANISM(ϕ^0, ϕ^1,...,ϕ^{N-1}, args) *Input: policy of all agents, hyperparameters*

Initialize η, θ_{coop}, θ_{env}, θ_{ind}, ψ_{coop}, ψ_{env}, ψ_{ind}

$num\_episode \leftarrow 0$

for number of episodes to train **do**

Run agents with policies ϕ^0, ϕ^1,...,ϕ^{N-1} for an episode with incentives given by η

$num\_episode \leftarrow num\_episode + 1$

Add the transitions from episode to replay buffer of IQ-Flow

Update agent policies ϕ^0, ϕ^1,...,ϕ^{N-1} according to their private learning rules

Update θ_{coop}, θ_{env}, θ_{ind}, ψ_{coop}, ψ_{env}, ψ_{inc} using equations in 19

sample train set $\mathcal{B}_T$ and validation set $\mathcal{B}_V$ for metaupdate

simulate mechanism critic update for K times using $\mathcal{B}_T$, θ_{ind}

Update η using $\mathcal{B}_V$ (with equations 3 or 5)

end for

end procedure

Inferring Dilemma

We extend the losses from Implicit Q-Learning for our Multi-Agent RL framework. Let the optimal actions of the cooperative policy and incentivized behavior policy be defined respectively as:

$$
\begin{aligned}
a^i_{coop} &=_{a^i} Q^i_{\boldsymbol{\pi}_{coop}}(s, a^{i^-}, .) \\
a^i_{b} &=_{a^i} Q^i_{\boldsymbol{\pi}_{b}, ind}(s, a^{i^-}, .)
\end{aligned} \tag{1}
$$

Let the optimal actions for the self-interested policy of agents under standard environment conditions with no extra incentives be defined as:

$$
a^i_{env} =_{a^i} Q^i_{\boldsymbol{\pi}_{env}, env}(s, a^{i^-}, .) \tag{2}
$$

Action that causes a dilemma: $a^i_b \neq a^i_{coop}$.
a_{coop}: regarded as target labels and use a modified version of cross-entropy loss, for probabilistic view of Q-Values: pass them from a softmax layer.

The necessity of the modification in the cross-entropy loss: we only want the flow as long as there is a dilemma in the system so that there is no unnecessary and excessive flow.

Meta-Loss

$$L^m_\eta(\hat{\theta}_{ind}) := -\frac{1}{I_B N}\sum_{k=0}^{I_B-1}\sum_{i=0}^{N-1}\sum_{\tilde{a}=0}^{|A|-1} 1\left(\tilde{a} = a^i_{coop,k}\right)$$
$$\times\left(1 - 1(a^i_{b,k} = a^i_{coop,k})\right)\log\left(\sigma\left(Q^i_{\pi_b,ind}\left(s_k, a^i, a^{i^-}_k, \hat{\theta}_{ind}\right)\right)\right)\Big|_{a^i=\tilde{a}} \quad (3)$$
$$\sigma(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

Loss is masked when mechanism infers no dilemma!

Full Meta Update

Our final incentive loss for η is given below as $L_\eta^{R_{inc}}(\hat{\theta}_{ind})$:

$$L_\eta^{R_{inc}}(\hat{\theta}_{env}, \hat{\theta}_{inc}) = L_\eta^{m} + c_1 L_\eta^{cost_1}(\hat{\theta}_{inc}) + c_2 L_\eta^{cost_2}(\hat{\theta}_{inc}) \tag{4}$$

$$\begin{aligned} &\hat{\eta} \leftarrow \eta + \alpha \nabla_\eta L_\eta^{R_{inc}}(\hat{\theta}_{env}, \hat{\theta}_{inc}) \\ &\nabla_\eta L_\eta^{R_{inc}}(\hat{\theta}_{env}, \hat{\theta}_{inc}) = \frac{\partial L_\eta^{R_{inc}}(\hat{\theta}_{env}, \hat{\theta}_{inc})}{\partial \hat{\theta}_{inc}} \frac{\partial \hat{\theta}_{inc}}{\partial \eta} = \\ &\frac{\partial L_\eta^{m} + c_1 L_\eta^{cost_1}(\hat{\theta}_{inc}) + c_2 L_\eta^{cost_2}(\hat{\theta}_{inc})}{\partial \hat{\theta}_{inc}} \frac{\partial \hat{\theta}_{inc}}{\partial \eta} \end{aligned} \tag{5}$$

Although our experiments show that these cost regularization terms are not required to get a successful performance, especially in simple problems, we find that including them leads to higher performance.

IPD $R - T$ and $S - P$ plot for Q-Values

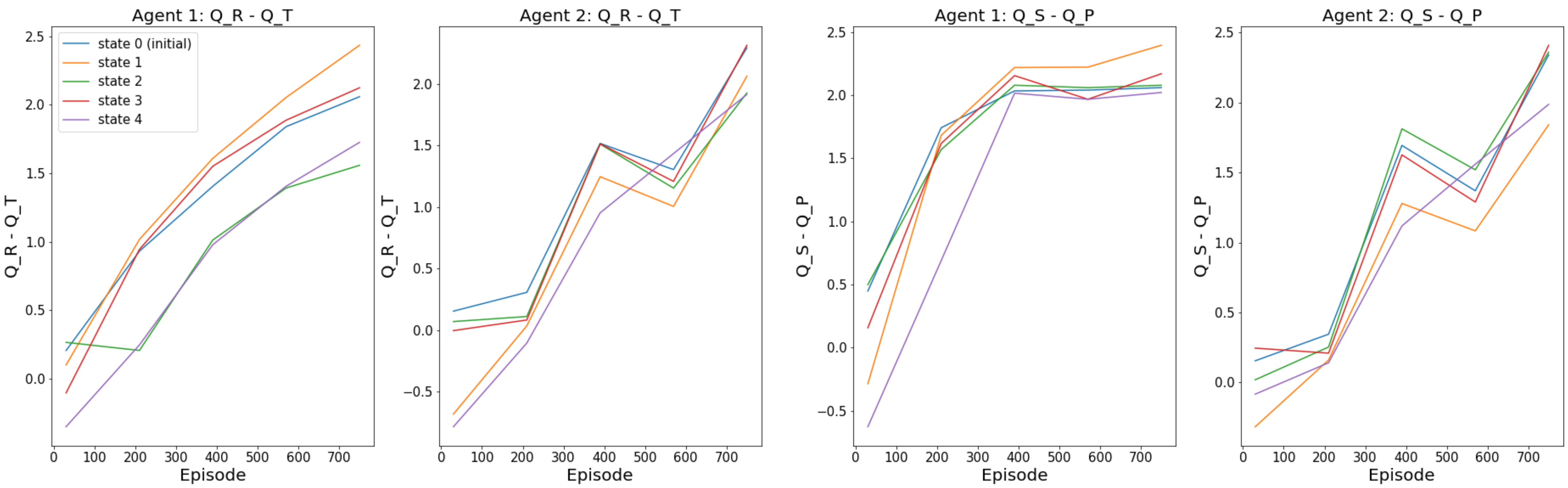


Figure 3: IPD $R - T$ and $S - P$ plot for Q-Values

Cleanup Results

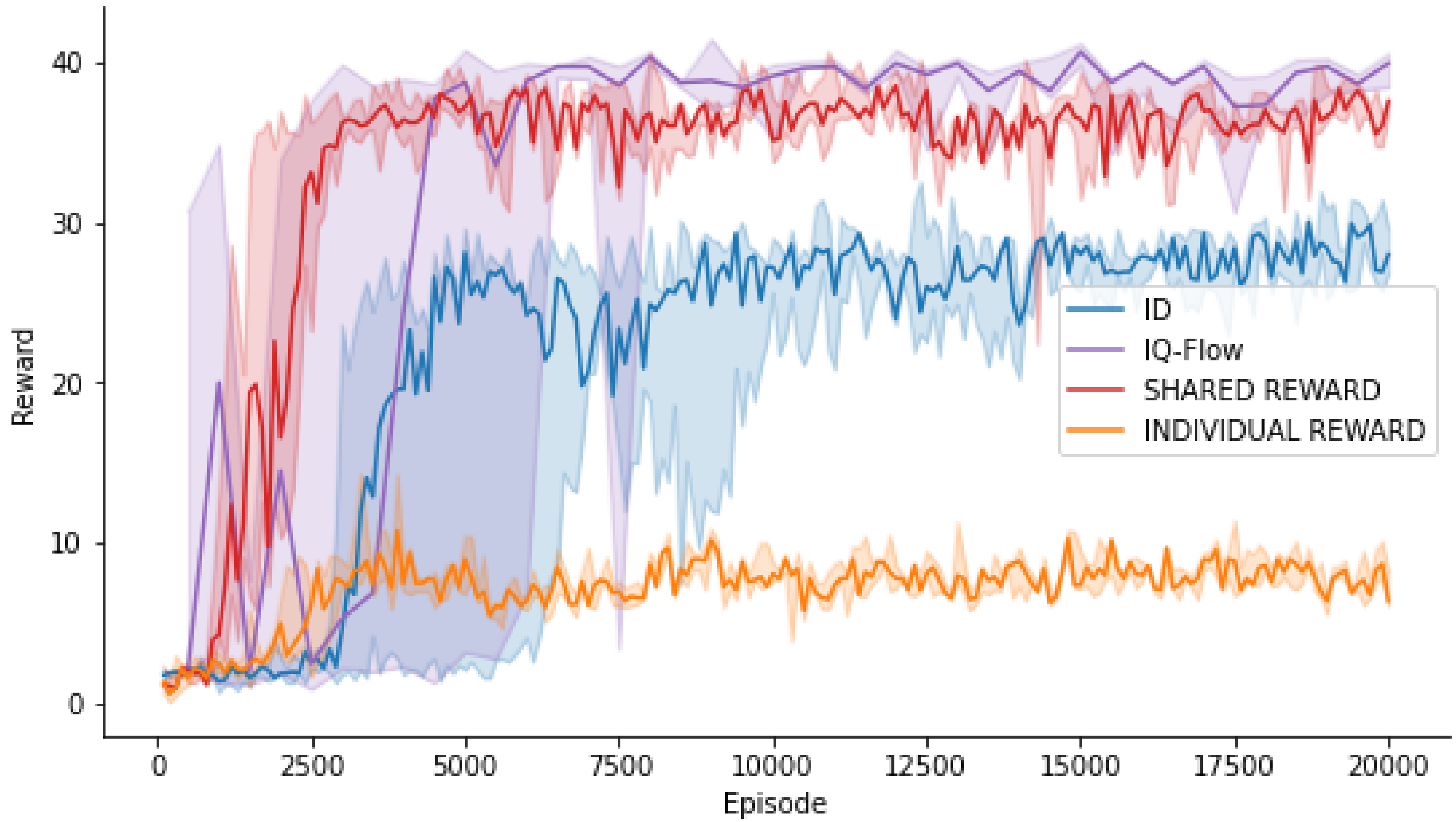


Figure 4: Cleanup Experiment Results: 7×7

Cleanup Results - Details

- IQ-Flow performs better than the baselines ID and independent actor-critic learner setup, while reaching the return upper bound identified by the shared setup's performance.
- While IQ-Flow performs better and reaches the upper bound, it can lose stability close to the end of training due to being disconnected from the agents that are trained online.
- In order to obtain a more stable training, we reset the actor-critic agents in the environment each 1000 episodes. Since after each reset operation the actor-critic agents start learning from scratch, we sample evaluation results each 500 episodes in order to have a fair comparison of the mechanism performance with the other algorithms.

2 Player Cleanup Results - Ablation

IQ-Flow: standard algorithm with cost regularization cost 1 and cost 2.
IQ-Flow C: cost coefficient 1 is 0
IQ-Flow C2: there is no cost regularization.

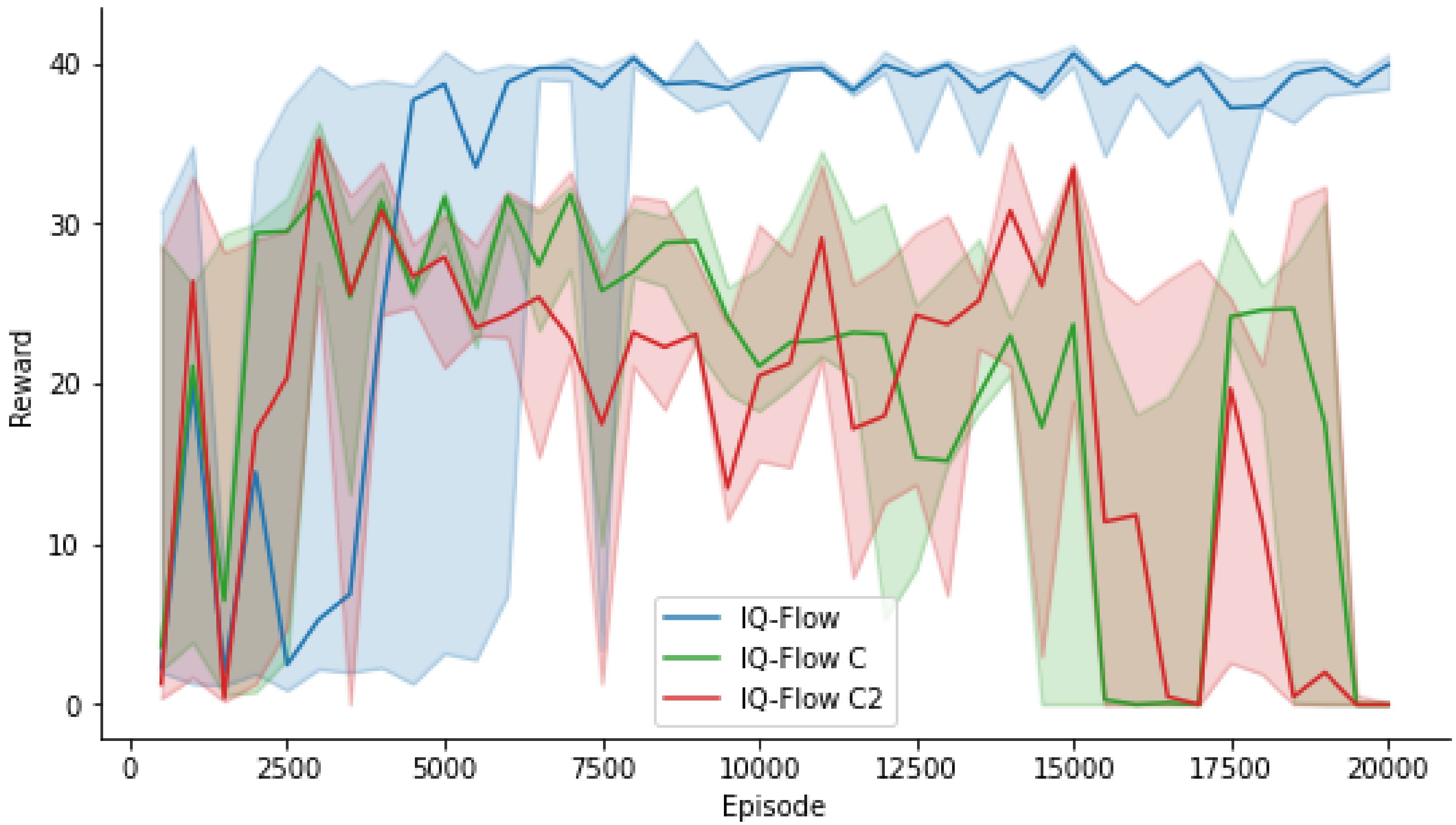


Figure 5: 2 Player Cleanup Experiment Ablation Results

Comparison Between Pretrained IQ-Flow Mechanism and Shared Reward

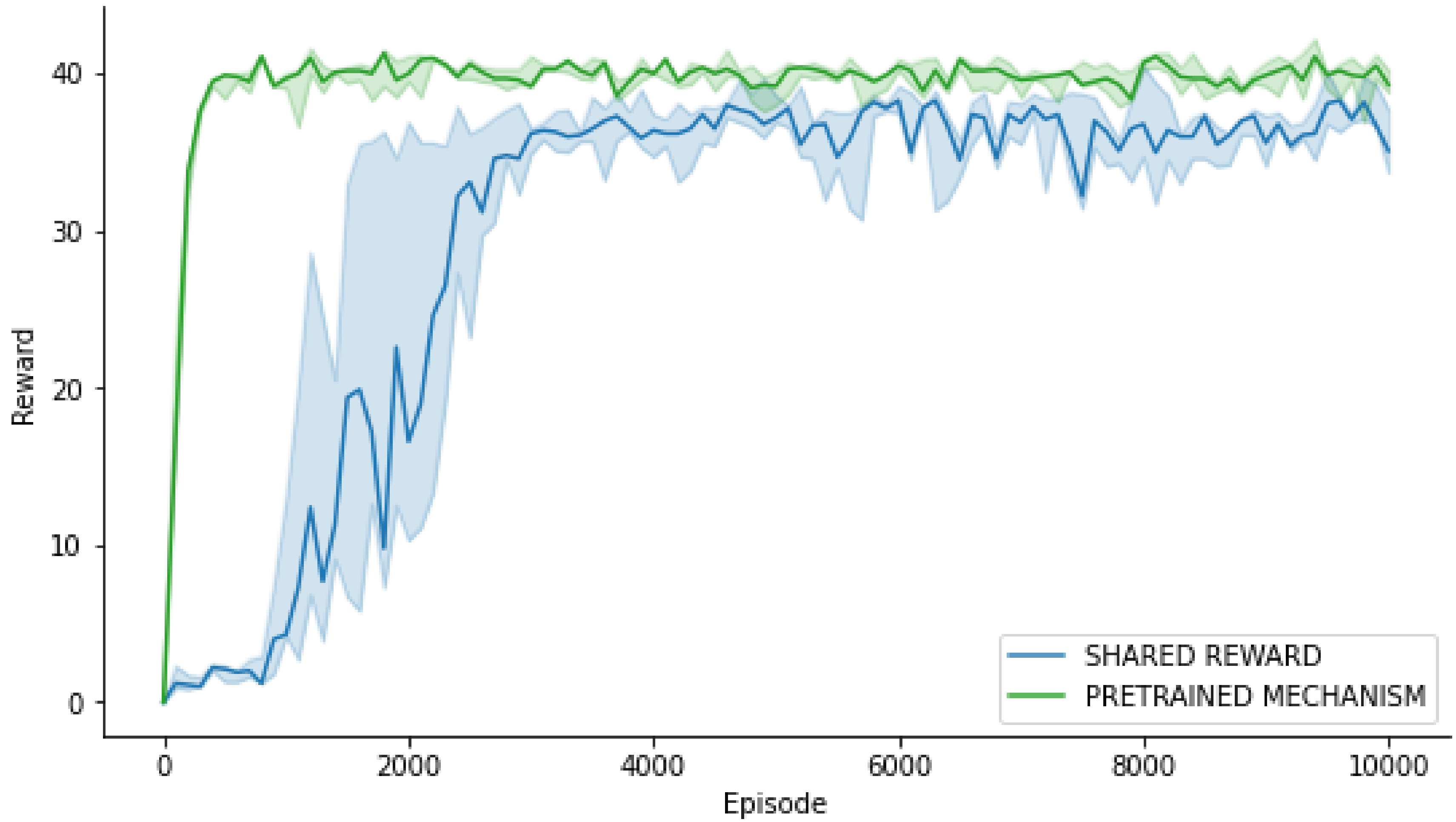
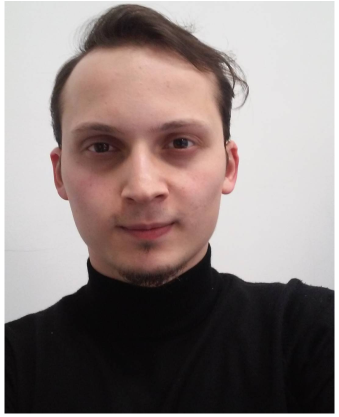


Figure 6: Comparison between pretrained IQ-Flow mechanism and shared reward setup

Contact

Thank you for listening!

Bengisu Guresti



Abdullah Vanlioglu

Nazim Kemal Ure

Contact information: guresti15@itu.edu.tr

References I

[1] J. Yang, A. Li, M. Farajtabar, P. Sunehag, E. Hughes, and H. Zha, “Learning to incentivize other learning agents”, *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 208–15 219, 2020.

[2] T. Baumann, T. Graepel, and J. Shawe-Taylor, “Adaptive mechanism design: Learning to promote cooperation”, *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, 2020.

[3] J. Yang, E. Wang, R. Trivedi, T. Zhao, and H. Zha, “Adaptive incentive design with multi-agent meta-gradient reinforcement learning”, *arXiv preprint arXiv:2112.10859*, 2021.

References II

[4] M. W. Macy and A. Flache, "Learning dynamics in social dilemmas", *Proceedings of the National Academy of Sciences*, vol. 99, no. suppl_3, pp. 7229–7236, 2002. DOI: 10.1073/pnas.092080099. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.092080099. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.092080099.

[5] J. Z. Leibo, V. Zambaldi, M. Lanctot, J. Marecki, and T. Graepel, "Multi-agent reinforcement learning in sequential social dilemmas", *arXiv preprint arXiv:1702.03037*, 2017.