# Capacity-Limited Decentralized Actor-Critic for Multi-Agent Games

Tyler Malloy, Chris R. Sims
*Cognitive Science*
*Rensselaer Polytechnic Institute*
Troy, NY, USA
{mallot,simc3}@rpi.edu

Tim Klinger, Miao Liu,
Matthew Riemer, Gerald Tesauro
*IBM Research AI*
Yorktown, NY, USA
{tklinger,mdriemer,gtesauro}@us.ibm.com
{miao.liu1}@ibm.com

***Abstract*—This paper explores information-theoretic constraints on methods for multi-agent reinforcement learning (MARL) in mixed cooperative and competitive games. Within this domain, decentralized training has been employed to increase learning sample efficiency. However, these approaches do not explicitly discourage complex policies, which can lead to over-fitting. To address this, we apply an information theoretic constraint onto agents' policies that discourages overly complex behaviour when it is not associated with a significant increase in reward. A second challenge in MARL is the non-stationarity of the environment introduced by other agents' changing policies. Previous methods in MARL have sought to reduce the impact of non-stationarity by inferring other agents' policies, but this can lead to over-fitting to previously observed behaviour. To avoid this, a similar information-theoretic constraint is applied onto the inference of other agents' policies, resulting in a more robust estimate. We evaluate the effects of these information-theoretic constraints on a test suite of multi-agent games, and report an overall improvement in performance, with greater improvements found in competitive domains compared to cooperative games.**

***Index Terms*—Multi-Agent Games, Reinforcement Learning, Information Theory**

## I. Introduction

Recent work in Multi-Agent Reinforcement Learning (MARL) has achieved impressive success in complex tasks such as the expert-level performance demonstrated in the video game Starcraft II [1] and emergent coordination in simulated soccer [2]. Other methods in MARL have focused on tasks with smaller environment representations that nevertheless require complex learning such as predicting other agent's behaviour and coordination among teammates [3], [4], [5]. The ability of MARL agents to effectively coordinate with teammates and compete with opponents is complicated by the non-stationarity introduced by other agents behaviour changing throughout learning. As other agent's behaviour changes, previous experience could be a poor indication of the optimal behaviour relative to other agent's current policies. This work investigates methods that apply an information constraint on agents' own policies and policy inference in the MARL setting to better handle its unique non-stationarity.

Previous work in reinforcement learning has demonstrated that an information-theoretic constraint on policy complexity can improve performance [6] and generalization [7] in continuous control tasks, as well as robustness to changing environment dynamics in more simple environments [8]. These methods train agents to learn so-called *capacity-limited* policies with lower informational complexity. The result is a learned behaviour that is less susceptible to over-fitting past experience, and thus more robust to changes within the learning environment. However, these methods have not been thoroughly explored in the MARL setting. The benefits demonstrated by capacity-limited RL are closely connected to the challenges introduced in MARL, indicating that the multi-agent setting may be a good candidate for applications in capacity-limited policies.

The work presented in this paper seeks to further the understanding of the potential for capacity-limited methods in MARL. To achieve this, we identify two closely related opportunities for applying capacity-limits. The first is in a similar manner as in previous approaches, by applying a penalty to agent's policies based on their informational complexity. The second method represents a novel application of capacity limits, by training MARL agents to learn less informationally complex approximations of other agent's policies. The specific ways of achieving these features as well as their theoretical motivation will be discussed further in the sections that describe their implementation.

## II. Related Work

### A. Capacity-Limited Reinforcement Learning

**Policy Mutual Information:** The Capacity-Limited approach is motivated by a desire to avoid overly complex policies that are not associated with a significant increase in reward [7]. Policy complexity is represented in information-theoretic terms by taking the policy function to be a Shannon information channel that communicates the action an agent should perform based on the state they are in [8]. Under this conceptualization, policy complexity can be represented using the mutual information of the policy which can be represented for discrete state and action spaces as:

$$\mathcal{I}(\pi(a|s)) = \sum_{s \in S} \sum_{a \in A} p_{(A,S)}(a,s) \log \left( \frac{p_{(A,S)}(a,s)}{p_A(a) p_S(s)} \right) \quad (1)$$

In practice, this policy mutual information term can be computationally expensive to calculate in the case of discrete

978-1-6654-3886-5/21/$31.00 ©2021 IEEE

state and action spaces, and intractable for general continuous spaces. To avoid this complex direct calculation, it is possible to break this policy mutual information term into its constituent entropies:

$$\begin{aligned}\mathcal{I}(\pi(a|s)) &= \mathcal{H}(\pi(a)) - \mathcal{H}(\pi(a|s)) \\ &= -\sum_{a\in A}\pi(a)\log(\pi(a)) + \sum_{\substack{a\in A\\ s\in S}}\pi(a|s)\log(\pi(a|s))\end{aligned} \tag{2}$$

In practice, the component entropies in Eq. 2 are often easier to approximate compared to the direct calculation (Eq. 1). The precise method for approximating the constituent entropies $\mathcal{H}(\pi(a))$ and $\mathcal{H}(\pi(a|s))$ are domain specific, and will be further detailed in the methods section.

**Capacity-Limited RL Objective:** The capacity-limited approach applied to reinforcement learning is a member of a broad class of methods that alter the objective function that the agent seeks to optimize. In the case of capacity-limited RL this is done by penalizing the reward observed based on the weighted mutual information of the agent's policy to encourage less informationally complex policies. This gives the capacity-limited reinforcement learning objective as follows:

$$J(\pi) = \sum_{t=0}^{T}\mathbb{E}_{(s_t,a_t)\sim p_\pi}[r(s_t,a_t) - \beta_\pi\mathcal{I}(\pi(a_t|s_t)] \tag{3}$$

where $r(s_t, a_t)$ indicates the reward signal associated with state $s$ and action $a$ at time step $t$, and $\beta_\pi$ is the policy-mutual information regularization coefficient that controls the trade-off between reward maximization and policy simplicity. As $\beta_\pi \rightarrow 0$ the objective retains the traditional reward maximization approach, and as $\beta_\pi \rightarrow \infty$ the agent will learn a more and more informationally simplistic policy. Optimizing this $\beta_\pi$ parameter equips an agent with the appropriate reward-complexity trade-off for the specific environment they are in. Importantly, any non-zero value of $\beta_\pi$ will lead the agent to learn a policy that is most informationally simplistic among all policies that achieve the same level of reward [7].

**Relation to MERL:** Another example of reward regularization is maximum entropy reinforcement learning (MERL), which uses policy entropy $\mathcal{H}(\pi(a|s))$ as its reward regularization term. An example of this is the Soft Actor-Critic (SAC) method which uses this regularization to encourage exploration in continuous control environments [9]. Because of the relation between policy mutual information and entropy (Eq. 2), these two methods are mathematically similar, however their motivations are different in two main aspects.

Firstly, the capacity-limited RL objective seeks to alter the long-term behaviour of the agent by training it to have a policy that is less informationally complex, whereas MERL seeks to improve exploration during training and eventually reach a policy that optimizes reward. Secondly, informationally constrained policies are motivated by the desire to improve generalization and robustness and may therefore better handle the non-stationarity of MARL. These features of capacity-limited RL will be further explored in the methods section which will describe how it is applied to the multi-agent setting.

### *B. Multi-Agent Deep Deterministic Policy Gradient*

**Model Structure:** The model presented in this paper follows the general structure for a decentralized actor/centralized critic (DACC) introduced in the Multi-Agent Deep Deterministic Policy Gradient (MADDPG) method [3]. This model can be considered as a member of a class of MARL methods that use a combination of centralized and decentralized training and execution. In the DACC method, a decentralized actor function performs actions using only local information available to the agent, with a centralized critic function has access to global information. The structure of this model is shown in Figure 1.

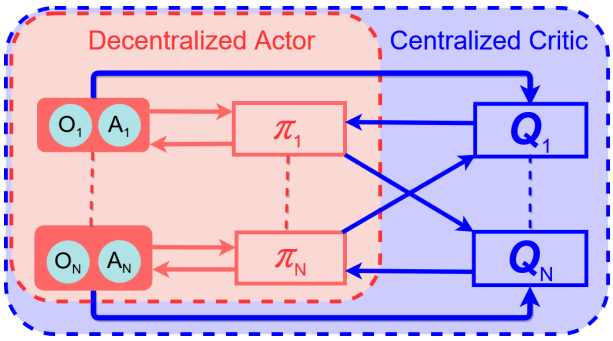


Fig. 1. Structure of a DACC model for multi-agent reinforcement learning as described in [3]. Agents learn policies based solely on their own observations but update these policies through an individual centralized critic. Arrows from agents' policies to other agents' Q-functions represent inference of other agent's policies. Importantly, these are approximations based on inference and not the true values of the policies themselves.

The method presented in this paper differs from MADDPG primarily in terms of how the centralized critic Q-function is calculated as well as the policy inference method. Because of this, we focus our detail of the MADDPG method on the centralized critic and policy inference. For a more complete description see [3].

**Centralized Critic:** The decentralized nature of learning in the DACC model ensures that at execution time actions can be performed without information from other agents, while improving learning through the centralized action-value function $Q_i^\mu$ which is updated as:

$$\begin{aligned}\mathcal{L}(\theta_i) &= \mathbb{E}_{x,a,r,x'}[(Q_i^\mu(x,a_1,...,a_N) - y)^2], \\ y &= r_i + \gamma Q_i^{\mu'}(x',a'_1,...,a'_N)|_{a'_j=\mu'_j(o_j),}\end{aligned} \tag{4}$$

In this equation, the N agents actions are $\{a_1, ...a_N\}$, target policies with delayed parameters $\theta'_i$ are $\{\mu_{\theta'_1},...\mu_{\theta'_N}\}$, agents policies are $\{\pi_i, ..., \pi_N\}$ which are parameterized by $\{\theta_i, ..., \theta_N\}$ respectively. Additionally, $x$ represents the state information that can be simply the agents observations $(o_1, ..., o_N)$ or alternatively additional information of the environment. While this Q-function requires information of other agent's policies that is not known generally, it can be approximated through policy inference.

**Policy Inference:** Because access to other agents' policies is not guaranteed generally, as other agents may be opponents who would not freely give this information, in some cases it

must be inferred through observations. This is done through policy inference by maintaining for each other agent an approximation $\hat{\mu}_i^j$ with parameters $\phi_i^j$. This approximation is updated through the learning objective:

$$\mathcal{L}(\phi_i^j) = -\mathbb{E}_{o_j,a_j}\big[\log \hat{\mu}_i^j - \lambda \mathcal{H}(\hat{\mu}_i^j)\big] \tag{5}$$

Here we see a regularization based on the policy entropy $\mathcal{H}(\hat{\mu}_i^j)$ and coefficient $\lambda$. This can be motivated by the desire to add a certain level of randomness to our model of other agents behaviour that prevents overly confident predictions about their actions. This is especially necessary in cases when our approximation is not an accurate reflection of other agent's behaviour. This policy inference model is updated using examples sampled from memory using a replay buffer.

## III. Methods

The application of a capacity-limited learning and inference method onto a decentralized structure results in a capacity-limited decentralized actor-critic (CLDAC). This section details how the capacity-limited objective is applied onto the decentralized domain.

### A. Policy Information Approximation

Exactly calculating the mutual information of an agent's policy generally requires summing over the full state and action space which is further complicated in the multi-agent setting through the non-stationarity introduced by other agent's policies as this impacts the steady-state distribution. To allow for a simple approximation of the agent's policy information, we adopt an approach based on first deconstructing the policy into its constituent entropies before approximating each entropy based on the current mini-batch memory sample at each training step.

To aid in approximating the policy mutual information based on mini-batch samples, we embed the replay buffer with the instantaneous policy entropy for action $a_i$ as $H(a_i)$ (referred to as $h_i$ moving forward) at each experience step. This results in the individual agent's memory at time $t$ being represented by $M_{0:t-1}$ where $(o_i, a_i, r_i, o_{i+1}, h_i) \in M_{0:t-1} \ \forall_{0<i<t-1}$. This inclusion allows us to approximate the policy mutual information for a mini-batch sample $D$ as follows:

$$\begin{aligned}\mathcal{I}(\pi(a|o)) &= \mathcal{H}(\pi(a)) - \mathcal{H}(\pi(a|o)) \\ &\approx -\sum_{a\in D} \pi(a)\log\pi(a) + \sum_{o\in D}\pi(a|o)\log\pi(a|o) \\ &\approx \sum_{h\in D} h - \sum_{o\in D}\mathcal{H}_{\pi(a|o)}(a|o)\end{aligned} \tag{6}$$

Where $\mathcal{H}_{\pi(a|o)}(a|o)$ is the entropy of the agent's current predicted action $a = \pi(a|o)$ conditioned on the observation $o$ sampled from memory. This approximation will approach the true policy mutual information in the limit of infinite experience under the assumption that all agents' behaviour converges as the actions $a$ and observations $o$ approach the steady-state distribution relative to each agents' policy, and the average of individual action entropies $h$ approaches the true marginal action probability $H(\pi(a))$.

In practice, for timescales used in comparing agents learning in complex environments this approximation does not reach the true policy mutual information. Additionally, in continuous state or action environments an exact calculation of policy mutual information is intractable, making an empirical comparison of approximated and true policy information impossible. It would be possible to estimate the policy mutual information based on the entire memory instead of only the current training mini-batch. However, this would be significantly more computationally expensive and not expected to produce a more accurate estimate. This is confirmed through experimentation comparing the two possible approximation approaches in the Experiments section.

Generally, the closeness of the policy information approximation to ground truth may not be critical for an update relative to the current batch from memory, as the most useful approximation for training is not necessarily the true policy information. This is because the true policy mutual information may over represent states that are irrelevant to the current training batch being used to update an agent's behaviour. As long as the policy information approximation effectively penalizes reward based on policy complexity, the desired impact on training is achieved.

### B. Capacity-Limited Policies

The capacity-limited learning objective is applied to the DACC model through altering the decentralized Q-function from the original shown in Eq. 4 as follows:

$$\begin{aligned}\mathcal{L}(\theta_i) &= \mathbb{E}_{x,a,r,x'}[(Q_i^\mu(x, a_q, ..., a_N) - y)^2] \\ y &= r_i - \beta_\pi \mathcal{I}(\pi_i(a|s)) + \gamma Q_i^{\mu'}(x', a'_1, ..., a'_N)|_{a'_j=\mu'_j(o_j))}\end{aligned} \tag{7}$$

where $\beta_\pi \mathcal{I}(\pi_i(a|s))$ is the weighted approximated policy mutual information for the agent with policy $\pi_i(a|s)$ as defined in Eq. 2. The result is a training regime that penalizes policies that are informationally complex in relation to the current batch that is being used to update agent's behaviour.

This change has the most significant impact on agent's performance, and is responsible for achieving the desired goal of training agents to have less informationally complex policies. In experimentation, we demonstrate the improvement in performance that is observed when using this capacity-limited objective alone, as well as a comparison of performance when incorporating the additional option to impose a capacity-limit on information in policy inference. When using the policy inference method, this Q-function no longer assumes access to other agents' policies and can be re-written as:

$$\begin{aligned}\mathcal{L}(\theta_i) &= \mathbb{E}_{x,a,r,x'}[(Q_i^\mu(x, a_q, ..., a_N) - y)^2] \\ y &= r_i - \beta_\pi \mathcal{I}(\pi_i(a|s)) + \gamma Q_i^{\mu'}(x', \hat{a}'_1, ..., \hat{a}'_N)|_{\hat{a}'_j=\hat{\mu}'_j(o_j))}\end{aligned} \tag{8}$$

where the predicted actions are based on the approximated policies of other agents such that $\forall i \neq j : \hat{a}_j \sim \hat{\mu}'_j(o_j)$ and

where $\hat{\mu}'_j$ is the approximated policy learned through the policy inference method, as shown in [3].

The motivation for applying an information capacity-limit to the Q-function in this way is to discourage the agent from learning a policy that maximizes performance relative to sampled experience at the expense of flexibility to newer observed behaviour of other agents. As agents' memory becomes more homogenous with experience, over-fitting to maximize reward relative to this can result in behaviour that is at risk of being exploited by other agents. Finding the appropriate mutual information regularization coefficient $\beta_\pi$ allows the agent to appropriately balance informational simplicity and reward maximization to avoid this issue.

### C. Capacity-Limited Policy Inference

The capacity-limited policy inference aspect of the CLDAC approach describes the information constraint that is applied to an agent's model of other agent's behaviour. When modelling other agent's behaviour in MARL, it is important to not make strong predictions on which action another agent will make. In the cooperative setting, this is important as the model of other agents behaviour may be inaccurate, and failing to account for other possibilities can have a negative impact on collaboration. The competitive setting is perhaps more relevant for the desire for less informationally complex models of other agent's behaviour. This is because opponents could theoretically take advantage of a policy inference model that is overfit by changing their behaviour.

For these reasons we investigate an approach for modelling other agent's behaviour that replaces the entropy regularization approach described in Eq. 5 with a policy information regularization as follows:

$$\mathcal{L}(\phi_i^j) = -\mathbb{E}_{o_j,a_j}\left[\log \hat{\mu}_i^j - \beta_\mu \mathcal{I}(\hat{\mu}_i^j)\right] \tag{9}$$

Where $\mathcal{I}(\hat{\mu}_i^j)$ is the estimated policy mutual information of other agent's inferred policies $\hat{\mu}_i^j$. This value is estimated in the same manner as the estimated policy entropy in the standard inference method, based on observations from opponents behaviour.

While the entropy regularization based approach could be motivated by a similar desire as previously mentioned to motivate this capacity-limited approach, they result in slightly different models of other agent's behaviour. By using the policy mutual information regularization, the policy inference is encouraged to predict behaviour that is more similar to another agent's marginal action, independent of the state they are in. This is similar to the concept of trying to learn a default behaviour that describes another agent, and assuming that they will perform similarly to that default behaviour. This approach can be beneficial in modelling other agent's behaviour in states that have not been previously observed, which occur frequently in environments with continuous state spaces as the environments used in this paper.

The capacity-limited policy inference method uses the mutual information coefficient $\beta_\mu$, which could be optimized to maximize performance. However, the policy inference used in MADDPG has a single policy inference entropy coefficient $\lambda = 1\text{e}{-3}$ across all models. For that reason, all results presented in this paper use the same policy inference mutual information coefficient $\beta_\mu = 1\text{e}{-3}$ to allow for a valid comparison of the CLDAC and MADDPG models.

### D. Capacity-Limited DAC

Taken together, the application of capacity-limits onto agent's policies and policy inferences results in the learning structure for the CLDAC model shown in Figure 2.

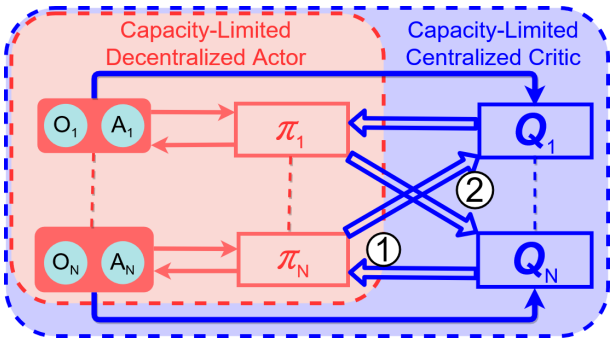


Fig. 2. Structure of a capacity-limited DAC model. Double arrows indicate the presence of an information-theoretic capacity constraint introduced through the training methods. Double arrows between and agent's decentralized q-function and their policy (labelled #1) indicate the information constraint on agent's policy complexity described in Eq. 8 Double arrows between agent's policies and other agent's Q-functions (labelled #2) indicate the information constraint on policy inference described in Eq. 9.

### E. Capacity-Limited DAC Algorithm

**Algorithm 1:** Capacity-Limited DAC

Initialize: Memory $\mathcal{D} = \emptyset$
**for** *each episode* **do**
  **for** *each environment step* **do**
    Execute actions $(a_1, ..., a_n)$
    $h \leftarrow (\mathcal{H}(\pi_1(a)), ..., \mathcal{H}(\pi_n(a)))$
    Observe rewards r and new states x'
    $\mathcal{D} \leftarrow \mathcal{D} \cup (x, a, r, x', h)$
  **for** *each agent i = 1 to N on gradient step* **do**
    Sample S samples $(x^j, a^j, r^j, x'^j, h^j)$ from $\mathcal{D}$
    $\mathcal{I}(\pi_i^j(a^j|x^j)) \leftarrow \beta_\pi\left(\sum h^j - \sum \mathcal{H}(\pi_i(\cdot|x^j))\right)$
    $\hat{r}_i^j \leftarrow r_i^j - \beta_\pi \mathcal{I}(\pi_i(a^j|x^j))$
    $y^j \leftarrow \hat{r}_i^j + \gamma Q_i^{\mu'}(x'^j, a_1'^j, ..., a_N'^j)|_{a'_k=\mu'_k(o_k^j))}$
    Update critic w.r.t the loss:
    $\mathcal{L}(\theta_i) = 1/S \sum_j \left(y^j - Q_i^\mu(x^j, a_1^j, ..., a_N^j)\right)^2$
    Update agent policy w.r.t the policy gradient:
    $\nabla_{\theta_i} J \approx$
     $1/S \sum_j \nabla_{\theta_i} \mu_i(o_i^j) \nabla_{a_i} Q_i^\mu(x^j, a_1^j, ..., a_N^j)|_{a_i=\mu_i(o_i^j)}$
    Update policy inference w.r.t the loss:
    $\mathcal{L}(\phi_i^j) = -\mathbb{E}_{o_j,a_j}\left[\log \hat{\mu}_i^j - \beta_\mu \mathcal{I}(\hat{\mu}_i^j)\right]$
  Update target network parameters for all agents i:
   $\theta'_i \leftarrow \tau\theta_i + (1-\tau)\theta'_i$

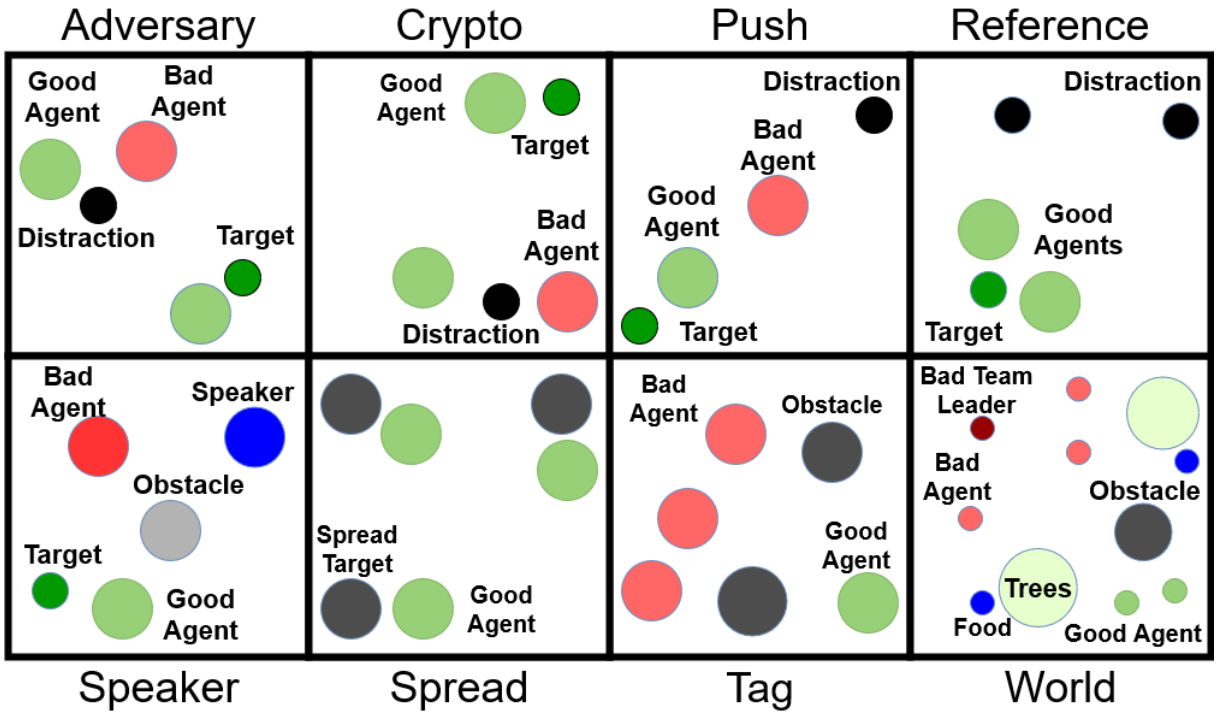


Fig. 3. Multiagent Particle Environments [3]. All environment names are taken from the codebase. **Adversary**: 2 Good agents and 1 adversary are rewarded by closeness to a target, good agents must not reveal which object is the target by spreading to both the target and distraction. **Crypto**: 1 Good agent communicates target landmark to another good agent over a public communication channel, 1 adversary attempts to decode communicated target. **Push**: 1 good agent moves towards target landmark while avoiding 1 adversary. **Reference**: 2 mobile good agents communicate to determine which landmark is the target. **Speaker**: 1 static good agent communicates to 1 mobile good agent which landmark is the target. **Spread**: 3 good agents spread to cover all landmarks. **Tag**: 1 good agent moves to distance itself from 3 adversaries using obstacles to slow their approach. **World**: 2 good agents move to gather food, hide within trees, and avoid 4 adversaries, 1 adversary leader can observe good agents hiding in trees and communicate their location.

The capacity-limited DAC algorithm is adapted from the related MADDPG method [3], with additions to approximate agent's own policy mutual information as well as the mutual information of other agents' policies. $\mathcal{I}(\pi_i(a^j|x^j))$ represents the policy mutual information relative to the portion of the full state representation $x^j$ that agent i observes, or $o^j$. These estimates are used to penalize complexity of both agent's policies as well as the inference of other agent's policies. Notation used matches the equations defining capacity-limited Q-function loss in Eq. 8 and capacity-limited inference Eq. 9. For a complete description of additional notation see [3].

## IV. Experiments

The learning environments used to compare performance of the CLDAC method are the multi-agent particle environments[1] (MPE) that were originally presented alongside the MADDPG method [3] and based on [10]. This suite of environments is a popular testbed for MARL methods, especially in decentralized training centralized execution approaches [11]. While the original work presented 4 environments, the environment suite contains 4 additional environments that are in some cases more complex, containing more agents and environment features. For completion we include all 8 environments[2] in our testing, detailed in Figure 3.

The structure of the CLDAC method applies capacity-limits in two separate ways, one in policy complexity and one in policy inference complexity, making it possible to compare the impact on performance of these alterations individually as well as together. In the following section presenting experimentation training results we compare the complete CLDAC model with two alternate versions that apply the information constraint to only agents policies, and only agent's policy inferences.

### A. Experiment Results

Figure 4 shows training results for the MPE environments comparing the CLDAC model against MADDPG. These results indicate that the CLDAC model affords an improvement over the baseline MADDPG method in the majority of environments we tested. This difference in performance is most significant towards the end of training, as the policy mutual information approximation becomes more accurate. The most significant difference in end of training performance is observed in the Crypto, Push, Tag and World environments

[1]https://github.com/shariqiqbal2810/multiagent-particle-envs

[2]Due to limitations in the original codebase, 2 environments are not included in tests that involved approximating other agent's policies, as the environments contained agents actions being represented by unsupported types of action distributions. These are the Reference and World environments.


Fig. 4. MPE environment results comparing CLDAC and MADDPG models averaged across 10 agents trained for 50K training episodes. For CLDAC policy mutual information coefficient $\beta_\pi$ parameters are optimized based on performance from 3 left out agents not included in these results. Policy inference $\beta_\mu$ parameters are set to $1e{-}3$ for all environments. Reference and World environments do not support policy inference and CL-$\pi$ results are shown in place of CLDAC results. Shaded region represents standard deviation of reward, all rewards are averaged over a window of 500 episodes. A one-way between agent ANOVA showed a significant effect of model type on end of training reward (1K episodes) $(F = 19.11, p = 1.4e{-}05)$ using environments as conditions.

which are 4 of the 5 competitive environments. The cooperative environments have a relatively similar performance. This indicates that the CLDAC method may be more impactful in competitive environments as the positive impact of informationally simplistic policies and policy inference is more critical in competitive environments.

To understand how the capacity-limited policy training and capacity-limited policy inference interact with each other to result in improved performance, we compare performance of a complete CLDAC model with an ablation of models using only capacity-limited policies (CL-$\pi$) and only capacity-limited policy inference (CL-$\mu$). For these cases we compare average performance in the final 1K training episodes as it represents performance after the CLDAC method has learned a useful policy mutual information approximation. The results shown in Figure 5 show the performance of the CLDAC model against the baseline MADDPG model, with reward normalized to the setting with only MADDPG agents.

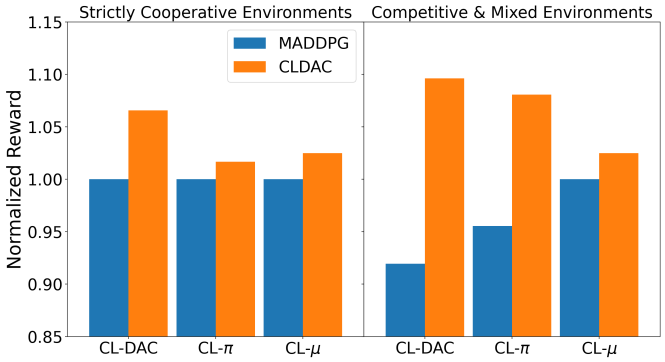


Fig. 5. Final 1K training episode reward averaged across all 8 environments of 3 different types of CLDAC models. For the strictly cooperative environments, MADDPG reward is normalized to 1. For competitive and mixed environments a CLDAC vs. MADDPG reward was normalized against a baseline of MADDPG vs. MADDPG reward. As World and Reference environments do not support policy approximation these are only included in the CL-$\pi$ results.

These results indicate that the CLDAC method demonstrates improved performance over MADDPG in the MPE environments through the integration of the capacity-limited learning objective and policy inference method. The most notable difference in performance occurs when CLDAC agents are competing against MADDPG agents in competitive and mixed environments. This is expected as any advantage afforded by the capacity-limited approach will be magnified when competing against agents that do not use this training method. Additionally, the original justification for utilizing capacity-limited policies in MARL was accounting for non-stationarity of other agents' changing policies, which can have a larger reward impact in competition. Within the competitive and mixed environments, the largest contribution to improved performance is from the capacity-limited policies, demonstrated by the relatively large difference in performance for the MADDPG vs. CL-$\pi$ results. This conforms to intuitive expectations as our agent's behaviour is more impactful on their observed reward compared to how accurately they model other agent's behaviour. A one-way between agent ANOVA

showed a significant effect of model type (CLDAC vs. CL-$\pi$ vs. CL-$\mu$) on end of training reward (1K episodes) ($F = 44.17, p = 5.60e-11$) using environments as conditions.

While the largest difference in performance can be attributed to the capacity-limited policy training method, as mentioned previously the capacity-limited policy inference did not optimize the mutual information coefficient $\beta_\mu$. Additional testing on optimizing this parameter for each environment may demonstrate further improvement in performance from the CLDAC and CL-$\mu$ models. However this would require additionally optimizing the entropy coefficient $\lambda$ used in the base policy approximation method for validity, for that reason this is outside of the scope of the present work.

To better understand the performance of the CLDAC method in individual environments, we compare performance against the MADDPG model in each environment in Figure 6. Results in environments that are strictly cooperative show the same general trend of relatively low difference in performance between the CLDAC and MADDPG methods. Results in competitive and mixed environments show a larger difference in performance, and importantly the more complex World and Speaker environments have the largest difference in performance among their respective groups. This indicates that the improvement afforded by the capacity-limited approach is not only observed in less complex environments.

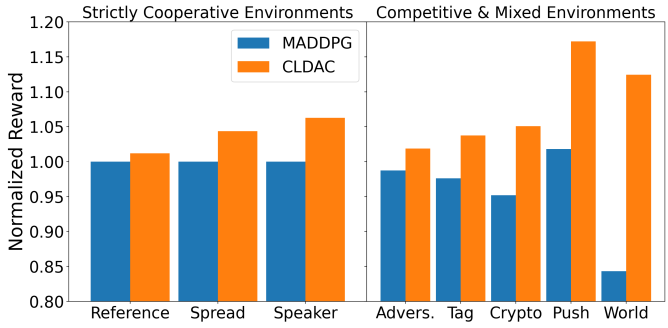


Fig. 6. Final 1K training episode reward from all 8 environments for 10 CLDAC vs. MADDPG models. Left group is strictly cooperative environments, right is competitive and mixed environments. All environments have $\beta_\mu = 1e{-}3$ and $\beta_\pi$ optimized using 3 left out models, values listed in appendix. Reference and World environments do not accommodate policy inference and CL-$\pi$ results are shown in place of CLDAC results.

### B. Policy Information Approximation

As mentioned previously, the continuous state space of the MPE environments makes a direct calculation of agent's policy mutual information intractable. To analyze the accuracy of the policy information approximation method, we instead compare the approximation used in training with an approximation calculated using the entire agent memory.

Additionally, we are interested in determining the impact on policy mutual information and end of training reward that occurs by varying the policy information coefficient $\beta_\pi$. To compare this effect we vary the $\beta_\pi$ coefficient in the Spread task and report the end of training mutual information approximation as well as reward (mean $\pm$ standard deviation) averaged across three agents.

| MI Coef $\beta_\pi$ | Batch Approx. | Full Approx. | Reward |
|---|---|---|---|
| 1e-1 | 0.1710 | 0.1723 | -456.9 $\pm$ 27.2 |
| 1e-2 | 0.2360 | 0.2348 | -453.5 $\pm$ 28.7 |
| 1e-3 | 0.2454 | 0.2457 | **-448.3** $\pm$ 28.2 |
| 1e-4 | 0.2521 | 0.2526 | -454.9 $\pm$ 26.2 |
| 0 | 0.2664 | 0.2666 | -466.7 $\pm$ 21.4 |

These results demonstrate the expected impact that varying the policy mutual information coefficient has on the approximated policy mutual information, as increasing the value of the coefficient decreases the approximated mutual information. While this trend can be seen, the effect is not statistically significant due to the limited number of agents trained (n=3), and additional agents would be required to confirm this trend. The impact of varying this coefficient on reward matches previously presented results, very low values of $\beta_\pi$ have little impact on performance, and too high values have a negative impact, while a moderate value (1e-3) improves performance. Additionally we can see that approximating the mutual information based on only the current batch alone is very close to an approximation that uses the full memory. This demonstrates that approximating the mutual information using the entire agent memory would be unlikely to result in an impactful difference in training.

## V. Discussion

In this paper we presented the Capacity-Limited Decentralized Actor Critic (CLDAC) method which trains agents to have less informationally complex behaviour to improve their performance in multi-agent settings. This was done through the use of a capacity-limited learning objective and similarly constrained policy inference method. Results from comparisons of performance on a suite of 8 complex cooperative and competitive environments showed improved performance after optimizing the trade-off of reward and policy complexity when training agents. These results demonstrate that the capacity-limited approach is a good candidate for applications in a range of MARL environments.

The main source of comparison for the CLDAC method was the MADDPG approach which leverages a mix of centralized training and decentralized execution to combat non-stationarity and increase sample efficiency [3]. More recent methods in decentralized MARL have demonstrated improved performance over MADDPG in some learning tasks, such as the Multi-Actor-Attention-Critic [5] which avoids concatenating all agents observations into the decentralized critic through the use of attention heads. This is motivated by the desire to avoid overly large critic observation spaces for environment with very many agents. While this approach shows improved performance on these tasks, it differs in motivation from the CLDAC method presented in this work. For this reason it is possible that the capacity-limited learning objective can be extended to the MAAC structure as well, though this is outside the scope of the present work.

Other related work has recently shown that centralized training decentralized execution MARL methods may suffer from the common overestimation bias in Q-learning, and can benefit from reward regularization when learning in the MPE environments [11]. The results presented in this work are complementary to this finding, as fundamentally the capacity-limited learning objective regularizes the reward in a similar manner. However, the key difference between these approaches is the way that the capacity-limited method uses the policy information as a regularization term, resulting in a bias for more informationally simplistic behaviour as long as it is not associated with a significant decrease in expected reward. In this way, the capacity-limited RL approach when applied to MARL serves a dual purpose of avoiding overly informationally complex policies that can be exploited by opponents, while additionally regularizing the value estimate.

Another approach that is closely related to the motivation of Capacity-Limited RL is the Information Bottleneck Actor Critic (IBAC) [12]. This method seeks to improve generalization and reduce over-fitting to experience of RL agents by minimizing $\mathcal{I}(X, Z)$, the mutual information of the input and a stochastic latent variable Z. Although closely related, this method differs in application from capacity-limited RL which approximates policy complexity based on states and actions alone, instead of using a latent representation of a state. Additionally their motivations differ slightly, as IBAC is attempting to account for issues introduced by methods from supervised learning such as batch normalization and dropout which can have a negative impact on performance in RL agents [12], whereas the motivation of CLDAC focuses more on issues of generalization and non-stationarity in MARL.

## REFERENCES


[1] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev *et al.*, "Grandmaster level in starcraft ii using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[2] S. Liu, G. Lever, J. Merel, S. Tunyasuvunakool, N. Heess, and T. Graepel, "Emergent coordination through competition," in *International Conference on Learning Representations*, 2018.

[3] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 6382–6393.

[4] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[5] S. Iqbal and F. Sha, "Actor-attention-critic for multi-agent reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2961–2970.

[6] F. Leibfried and J. Grau-Moya, "Mutual-information regularization in markov decision processes and actor-critic learning," in *Conference on Robot Learning*. PMLR, 2020, pp. 360–373.

[7] T. Malloy, C. R. Sims, T. Klinger, M. Liu, M. Riemer, and G. Tesauro, "Deep rl with information constrained policies: Generalization in continuous control," *arXiv preprint arXiv:2010.04646*, 2020.

[8] R. Lerch and C. R. Sims, "Rate-distortion theory and computationally rational reinforcement learning," *Proceedings of Reinforcement Learning and Decision Making (RLDM) 2019*, pp. 7–10, 2019.

[9] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1861–1870.

[10] I. Mordatch and P. Abbeel, "Emergence of grounded compositional language in multi-agent populations," *arXiv preprint arXiv:1703.04908*, 2017.

[11] L. Pan, T. Rashid, B. Peng, L. Huang, and S. Whiteson, "Softmax with regularization: Better value estimation in multi-agent reinforcement learning," *arXiv preprint arXiv:2103.11883*, 2021.

[12] M. Igl, K. Ciosek, Y. Li, S. Tschiatschek, C. Zhang, S. Devlin, and K. Hofmann, "Generalization in reinforcement learning with selective noise injection and information bottleneck," *Advances in Neural Information Processing Systems*, vol. 32, pp. 13 978–13 990, 2019.


## VI. APPENDIX

Table I: Environment policy mutual information coefficients $\beta_\pi$ used in CLDAC and CL-$\pi$ training methods. $\beta_\pi$ parameters were fit using best performing average end of training performance from 3 left out models. $\beta_\pi$ coefficients tested were (1e-3, 5e-3, 1e-2, 5e-2, 1e-1).

| Environment | Poliy Coef $\beta_\pi$ |
|---|---|
| Adversary | 1e-2 |
| Crypto | 1e-1 |
| Push | 1e-1 |
| Reference | 1e-2 |
| Speaker | 1e-1 |
| Spread | 1e-2 |
| Tag | 1e-1 |
| World | 5e-3 |

Table II: Training parameters used across all experiment results presented. Parameters are kept as originally presented in the codebase alongside [3], apart from the number of episodes which was increased from 25K to 50K to better differentiate performance, and allow for the capacity-limited method to approximate policy mutual information.

| Parameter | Value |
|---|---|
| Number of Episodes | 50000 |
| Episode Step Length | 25 |
| Learning Rate | 1e-2 |
| Discount ($\gamma$) | 0.95 |
| Batch Size | 1024 |
| MLP Units | 64 |
| MLP Depth | 2 |
| Memory Max | 1M |

Table III: Environment number of good agents, bad agents, observation space and action space. In some environments the adversary agent has a different action or state space due to the differences of the tasks between the good and bad agents. In these cases observation and state spaces are represented as (bad agent space, good agent space). In all cases observation spaces are represent by continuous values over the range $(-\infty, \infty)$, and action spaces are discrete.

| Environment | Obs. Space | Action Space | # Good | # Bad |
|---|---|---|---|---|
| Adversary | (8,10) | (5) | 2 | 1 |
| Crypto | (4,8) | (4) | 2 | 1 |
| Push | (8,19) | (5) | 1 | 1 |
| Reference | (21) | (50) | 2 | 0 |
| Speaker (speaker) | (3) | (3) | 2 | 0 |
| Speaker (listener) | (11) | (5) | 2 | 0 |
| Spread | (18) | (5) | 3 | 0 |
| Tag | (14,16) | (5) | 1 | 3 |
| World | (28,34) | (5,20) | 2 | 4 |