Is Independent Learning All You Need in the StarCraft Multi-Agent Challenge?

Christian Schroeder de Witt[*†] **Tarun Gupta**[*‡] **Denys Makoviichuk**[§]

Viktor Makoviychuk[‡‡] **Philip H.S. Torr**[†] **Mingfei Sun**[‡] **Shimon Whiteson**[‡]

Abstract

Most recently developed approaches to cooperative multi-agent reinforcement learning in the *centralized training with decentralized execution* setting involve estimating a centralized, joint value function. In this paper, we demonstrate that, despite its various theoretical shortcomings, Independent PPO (IPPO), a form of independent learning in which each agent simply estimates its local value function, can perform just as well as or better than state-of-the-art joint learning approaches on popular multi-agent benchmark suite SMAC with little hyperparameter tuning. We also compare IPPO to several variants; the results suggest that IPPO's strong performance may be due to its robustness to some forms of environment non-stationarity.

1 Introduction

Many practical control problems feature a team of multiple agents that must coordinate to achieve a common goal [5, 12]. Cooperative multi-agent reinforcement learning (MARL) has shown considerable promise in solving tasks that can be described as a Dec-POMDP [17], i.e., where agents optimise a single scalar team reward signal in a partially observable environment while choosing actions based only on their own local action-observation histories [18, 2, 29].

Independent learning (IL) decomposes an n-agent MARL problem into n decentralised single-agent problems where all other agents are treated as part of the environment, and learning policies that condition only on an agent's local observation history. While easy to distribute and decentralisable by construction, IL suffers from a variety of theoretical limitations that may result in learning instabilities or suboptimal performance observed in practice [27, 9, IQL, IAC]. Firstly, the presence of other learning and exploring agents renders the resulting environment non-stationary from the given agent's perspective, forfeiting convergence guarantees [27]. Secondly, independent learners are not always able to distinguish environment stochasticity from another agent's exploration, making them unable to learn optimal policies in some environments [6].

In fact, decentralised policies need not be learnt in a decentralized fashion. For safety and efficiency reasons [26], MARL training frequently takes place centrally in a laboratory or in simulation, allowing agents access to each other's observations during training, as well as otherwise unobservable extra state information. Centralized training allows training of a single joint policy for all agents that conditions on the joint observations and extra state information. While centralized joint learning reduces or removes issues surrounding partial observability and environment non-stationarity, it must cope with joint action spaces that grow exponentially with respect to the number of agents, as well as a variety of learning pathologies that can result in suboptimal policies [32]. Importantly,

---

[*]*: equal contribution. Corresponding author: Christian Schroeder de Witt <cs@robots.ox.ac.uk>.
[†] Department of Engineering Science, [‡] Department of Computer Science; University of Oxford, UK;
[§] Snap Inc., USA; [‡‡] NVIDIA, USA

vanilla joint policies are not inherently decentralisable and naive policy distillation approaches are often ineffective [4]. Joint learning does not immediately address the multi-agent credit assignment problem either.

Recent research has focused on algorithms that can exploit the benefits of combining centralised training with decentralised execution [17, CTDE], while mitigating some of the pathologies of vanilla centralized joint learning. A particularly successful line of research uses value factorisation of the joint Q-value function in order to reduce the size of the joint action space [VDN, QMIX, FacMADDPG][25, 19, 7].

However, value factorization is prone to a learning pathology called *relative overgeneralization* [32] where policies erroeneously converge to a suboptimal joint action. Relative overgeneralisation commonly arises when multiple agents must coordinate their actions but receive negative rewards if only a subset of them do so. In this case, random occurrences of such successful coordination events are a needle in a haystack and monotonic value factorisations cannot represent the non-monotonic team reward function [3]. A variety of recent approaches employ joint optimisation of, or transfer learning between, factored and unrestricted joint value functions in order to overcome the representational limitations of factored value functions, with varying success [4, 24].

In this paper, we present empirical evidence for Independent PPO (IPPO), a multi-agent variant of *proximal policy optimization* [23], that shows IPPO matches or outperforms state-of-the-art MARL CTDE algorithms such as QMIX [19] or MAVEN [15] on multiple hard maps on SMAC [20].

Given the purported pathologies of existing IL approaches such as IQL and IAC, we hypothesise that algorithmic choices made by PPO such as policy clipping help mitigate some forms of environment non-stationarity, rendering fundamental theoretical limitations less important to practical performance. We empirically show that the effect of policy clipping in IPPO cannot be emulated by lowering the effective learning rate alone.

2 Related Work

Actor-critic algorithms have been shown to benefit from learning centralized joint critics alongside decentralised policies [9, Central-V]. As the critics are not needed during execution, this approach is inherently decentralizable. COMA [9] extends this approach with a counterfactual multi-agent critic baseline based on temporal difference errors [28] in order to facilitate multi-agent credit assignment. Joint Q-learning can also be made decentralizable. Value Decomposition Networks [25, VDN] decompose joint state-action value functions into sums of decentralised utility functions that can be used during greedy execution. QMIX [19] extends this additive decomposition to arbitrary centralized monotonic mixing networks. Both centralized joint critics and factored joint value functions can reap some benefits of centralized joint learning, while bypassing the joint action space explosion, imposing an effective prior on multi-agent credit assignment, and mitigating practical learning pathologies associated with centralized joint learning on popular benchmark environment StarCraft II [20, SMAC]. Value factorisation has also been successfuly transferred to continuous action spaces and combined with actor critic approaches [7, COMIX, FacMADDPG].

Independent learning (IL) dates back to the early days of multi-agent reinforcement learning [27, IQL]. While initially tabular, IL algorithms using neural networks as function approximators were subsequently developed [IAC, deep IQL][19, 9]. The question of whether independent agents are better at learning cooperative behaviour than multiple *complete observing* agents [33] was first comprehensively investigated by Tan [27], who concludes that while sharing policies or experience was generally advantageous, sharing extra sensory information could also negatively interfere with learning, e.g., by expanding the state space.

Trust region optimisation for reinforcement learning was popularised by TRPO [21] which implements iterative guaranteed monotonic improvements. PPO [23] preserves many of TRPO's empirical benefits while trading in theoretical guarantees for computational speed. PPO's policy update regularisation using clipped probability ratios has been subject to much scrutiny in single-agent settings: *Truly PPO* [31] suggests modifications in order to ensure guaranteed monotonic improvements with little computational overhead. Recent work confirms that PPO's performance may crucially depend on the choice of policy surrogate objective [11]. In addition, code-level design choices have been shown to significantly impact PPO performance in practice [8].

While popular for single agent tasks, PPO has only recently been applied to decentralised cooperative multi-agent tasks. Concurrent work proposes MAPPO [1], an actor-critic multi-agent algorithm based on PPO. Like IPPO, MAPPO employs weight sharing between each agent's critic. In contrast to IPPO, MAPPO uses a centralized value function that conditions on the full state (or concatenation of agent observations, if the full state is unavailable). As such, MAPPO is not an *independent* learning algorithm. We show that IPPO substantially outperforms MAPPO on a variety of hard SMAC maps and even beats other state-of-the-art algorithms, such as QMIX, on some of these, with only modest hyperparameter tuning.

3 Background

Dec-POMDPs. We consider a *fully cooperative multi-agent task* A *decentralised partially observable Markov decision process* [Dec-POMDP 17] describes multi-agent tasks where a team of cooperative agents chooses sequential actions under partial observability and environment stochasticity. Dec-POMDPs can be formally defined by a tuple $\langle \mathcal{N}, \mathcal{S}, \mathcal{U}, P, r, \mathcal{Z}, O, \rho, \gamma \rangle$. Here $s \in \mathcal{S}$ describes the state of the environment, discrete or continuous, and $\mathcal{N} := \{1, \ldots, N\}$ denotes the set of N agents. $s_0 \sim \rho$, the initial state, is drawn from distribution ρ. At each time step t, all agents $a \in \mathcal{N}$ simultaneously choose actions $u_t^a \in \mathcal{U}$ which may be discrete or continuous. This yields the joint action $\boldsymbol{u}_t := \{u_t^a\}_{a=1}^N \in \mathcal{U}^N$. The next state $s_{t+1} \sim P(s_t, \boldsymbol{u}_t)$ is drawn from transition kernel P after executing the joint action $\boldsymbol{u}_t$ in state s_t. Subsequently, the agents receive a scalar team reward $r_t = r(s_t, \boldsymbol{u}_t)$.

Instead of being able to observe the full state s_t, in a Dec-POMDP each agent $a \in \mathcal{N}$ can only draws an individual local observation $z_t^a \in \mathcal{Z}, \boldsymbol{z}_t := \{z_t^a\}_{a=1}^N$, from the observation kernel $O(s_t, a)$. The history of an agent's observations and actions is denoted by $\tau_t^a \in \mathcal{T}_t := (\mathcal{Z} \times \mathcal{U})^t \times \mathcal{Z}$. The set of all agents' histories is given by $\boldsymbol{\tau}_t := \{\tau_t^a\}_{a=1}^N$. Each agent a chooses its actions with a decentralised policy $u_t^a \sim \pi^a(\cdot | \tau_t^a)$ that is based only on its individual history.

The team of cooperative agents attempts to learn a *joint policy* $\pi(\boldsymbol{u} | \boldsymbol{\tau}_t) := \prod_{a=1}^N \pi^a(u^a | \tau_t^a)$ that maximises their expected discounted return, $J(\boldsymbol{\pi}) \doteq \mathbb{E}[\sum_{t=0}^\infty \gamma^t r_t]$, where $\gamma \in [0, 1)$ is a discount factor. $\pi(\boldsymbol{u} | \boldsymbol{\tau}_t)$ induces a joint action-value function Q^π that estimates the expected discounted return of joint action $\boldsymbol{u}_t$ in state s_t with histories $\boldsymbol{\tau}_t$. Using the joint action-value function, agents then follow a joint policy π using $Q^\pi(s_t, \boldsymbol{\tau}_t, \boldsymbol{u}_t) := \mathbb{E}\left[\sum_{i=0}^\infty \gamma^i r_{t+i}\right]$.

Centralised learning with decentralised execution. Reinforcement learning policies can often be learnt in simulation or in a laboratory. In this case, on top of their local observation histories, agents may have access to the full environment state and share each other's policies and experiences during training. The framework of *centralised training with decentralised execution (CTDE)* [17, 14] formalises the use of centralised information to facilitate the training of decentralisable policies.

Trust Region Policy Optimization TRPO [21] is a class of policy-gradient methods that restricts the update of policies to within the trust region of the behavior policy by enforcing a KL divergence constraint on policy update at each iteration. Formally, TRPO optimizes the following:

$$\max_\theta \quad \mathbb{E}_{s_t, u_t} \left[\frac{\pi_\theta(u_t | s_t)}{\pi_{\theta_{old}}(u_t | s_t)} A(s_t, u_t) \right] \tag{1}$$

$$\text{subject to} \quad \mathbb{E}_{s_t, u_t} \left[\text{KL} \left(\pi_{\theta_{old}}, \pi_\theta \right) \right] \leq \delta, \tag{2}$$

where θ_{old} are the policy parameters before the update and $A(s_t, u_t)$ is an approximation of advantage function. The formulation is computationally expensive as it requires the computation of multiple Hessian-vector products for nonlinear conjugate gradients when approximating the KL constraint. To resolve this, Proximal Policy Optimization [23] approximates the trust region constraints by policy ratio clippings, i.e. the policy loss becomes:

$$\mathcal{L}(\theta) = \mathbb{E}_{s_t, u_t} \left[\min \left(\frac{\pi_\theta(u_t | s_t)}{\pi_{\theta_{old}}(u_t | s_t)} A(s_t, u_t), \text{clip} \left(\frac{\pi_\theta(u_t | s_t)}{\pi_{\theta_{old}}(u_t | s_t)}, 1 - \epsilon, 1 + \epsilon \right) A(s_t, u_t) \right) \right]. \tag{3}$$

4 Independent PPO

In this paper, we use PPO to learn decentralized policies π^a for agents with individual policy clipping where each agent's independent policy updates are clipped based on the objective defined in Equation 3. We consider a variant of the advantage function based on independent learning, where each agent a learns a local observation based critic $V_\phi(z_t^a)$ parameterised by ϕ using *Generalized Advantage Estimation* (GAE) [22] with discount factor $\gamma = 0.99$ and $\lambda = 0.95$. The network parameters ϕ, θ are shared across critics, and actors, respectively. We also add an entropy regularization term to the final policy loss [16]. For each agent a, we have its advantage estimation as follows:

$$A_t^a = \sum_{l=0}^{h} (\gamma\lambda)^l \delta_{t+l}^a, \tag{4}$$

where $\delta_t^a = r_t(z_t^a, u_t^a) + \gamma V_\phi(z_{t+1}^a) - V_\phi(z_t^a)$ is the TD error at time step t and we vary h as shown in Table 1 (marked as steps num). We use the team reward $r_t(s_t, a_t)$ to approximate $r_t(z_t^a, u_t^a)$. The final policy loss for each agent a becomes:

$$\mathcal{L}^a(\theta) = \mathbb{E}_{z_t^a, u_t^a} \left[\min \left(\frac{\pi_\theta(u_t^a | z_t^a)}{\pi_{\theta_{old}}(u_t^a | z_t^a)} A_t^a, \ \text{clip} \left(\frac{\pi_\theta(u_t^a | z_t^a)}{\pi_{\theta_{old}}(u_t^a | z_t^a)}, 1 - \epsilon, 1 + \epsilon \right) A_t^a \right) \right]. \tag{5}$$

Value Clipping: In addition to clipping the policy updates, our method also uses value clipping to restrict the update of critic function for each agent a to be smaller than ϵ as proposed by [22, GAE] using:

$$\mathcal{L}^a(\phi) = \mathbb{E}_{z_t^a} \left[\min \left\{ \left(V_\phi(z_t^a) - \hat{V}_t^a \right)^2, \left(V_{\phi_{old}}(z_t^a) + \text{clip}(V_\phi(z_t^a) - V_{\phi_{old}}(z_t^a), -\epsilon, +\epsilon) - \hat{V}_t^a \right)^2 \right\} \right], \tag{6}$$

where ϕ_{old} are old parameters before the update and $\hat{V}_t^a = A_t^a + V_\phi(z_t^a)$. The update equation restricts the update of the value function to within the trust region, and therefore helps us to avoid overfitting to the most recent batch of data. For each agent, the overall learning loss becomes:

$$\mathcal{L}(\theta, \phi) = \sum_{a=1}^{n} \mathcal{L}^a(\theta) + \lambda_{\text{critic}} \mathcal{L}^a(\phi) + \lambda_{\text{entropy}} \mathcal{H}(\pi^a), \tag{7}$$

where $\mathcal{H}(\pi^a)$ denotes the entropy of policy π^a, λ_{critic} and λ_{entropy} vary as shown in Table 1. In Section 5.3, we compare IPPO to a baseline in which both policy and value clipping are ablated, yielding a variant of IAC [9].

Learning Architecture: We use a variance scaling initializer with truncated normal distribution [10] with scale = 2.0 to initialize the parameters of our policy and value function NNs that has been shown to work well with ReLU activations. The input to our NN comprises of stacked observations for the past few times steps (marked as frames in Table 1), which is passed through three conv1d convolution layers with varying number of filters (marked as net arch in Table) and fixed kernel_size = 3, strides = (2, 1, 1), padding = (same, valid, valid) for each conv layer, followed by two MLP layers with (256, 128) units and ReLU activations (see Table 1 for more details). We use gradient clipping to restrict the norm of the gradient to be less than 0.5 and normalize the advantage by subtracting the mean and dividing by the standard deviation once before training - we find that advantage normalisation by minibatch yields worse performance. We use a discount factor of $\gamma = 0.99$, a learning rate of 10^{-4}, and clipping parameter $\epsilon = 0.2$.

5 Empirical Results

In this section, we evaluate Independent PPO (IPPO) on 16 maps from the StarCraft Multi-Agent Challenge [20, SMAC] maps, performing only mild hyperparameter tuning for individual maps (see Table 1). SMAC consists of a diverse set of StarCraft II [30, SCII] unit micromanagement tasks of varying difficulty, where a collaborative team of SCII units needs to defeat an enemy team of units controlled by the built-in AI. SCII units consist of a balanced set of both melee and long-range attack units (plus healing units, *medivacs*) and winning strategies often entail precisely coordinated unit movements strategies, including *kiting* [19], in order to gain positional advantages. For all our experiments, we use game version 4.6 and select the hardest AI difficulty level.

(a) corridor

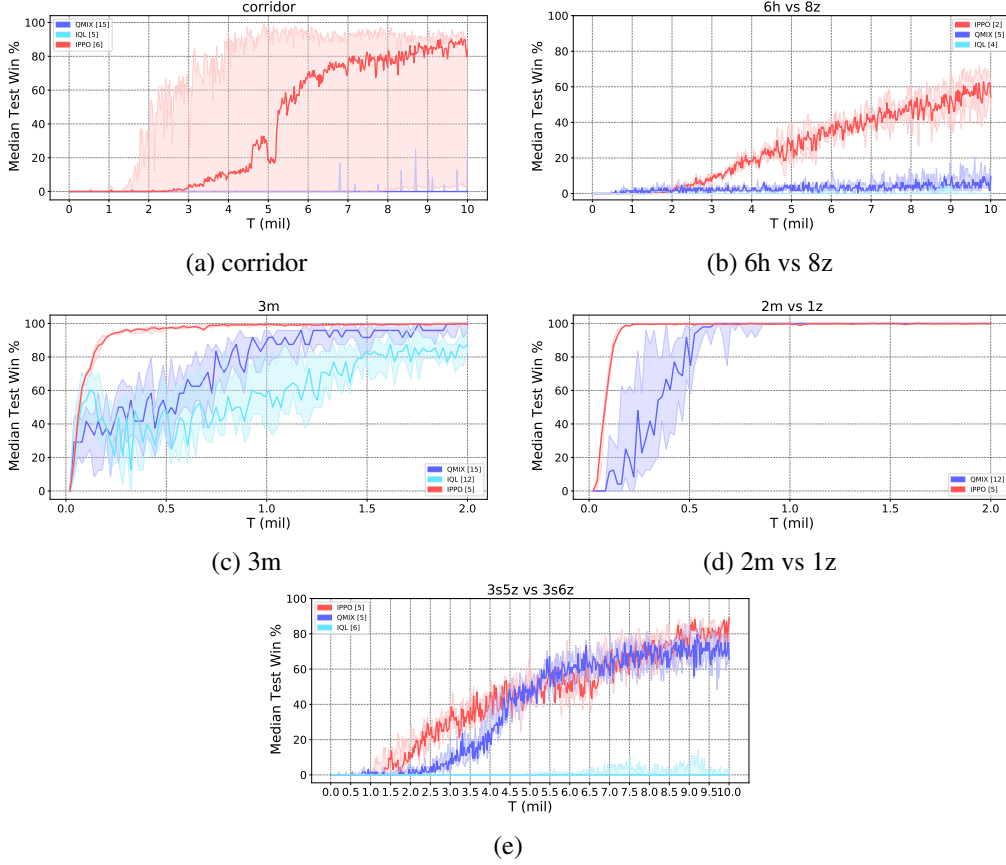(b) 6h vs 8z

(c) 3m

(d) 2m vs 1z

(e)

Figure 1: Results on select SMAC maps, comparing IPPO to QMIX and IQL. Uncertainty regions depict $[0.25, 0.75]$ confidence intervals of the median test win rate.

Map	corridor	$6h\_vs\_8z$	$3s5z\_vs\_3s6z$	$3m$	$2m\_vs\_1z$
IPPO	80	60	90	100	100
MAPPO	0	8.95	80	99.875	100

Figure 2: Median win test rates on select SMAC maps at 10M steps, comparing IPPO to MAPPO results reported by [1].

5.1 IPPO Performance

We benchmark IPPO and a number of ablations on a number of SMAC maps (see Figure 5). Despite its simplicity, we find that IPPO significantly outperforms two strong MARLbaselines that exploit centralized state during learning, namely MAPPO (see Table 5 and QMIX on three difficult maps, namely *3s5z vs 3s6z*, *corridor* and *6h vs 8z*. On *corridor*, IPPO even exceeds the particularly strong performance of MAVEN [15]. In addition, IPPO outperforms QMIX on easy SMAC maps *2m vs 1z* and *3m* and is competitive on a few more (see Appendix 9). IPPO is also able to learn maps renowned for their difficulty, such as *3s5z vs 3s6z* and *6h vs 8z*. We also find that IPPO performance is superior to that of IQL, the other independent learning algorithm tested [19], and it is also generally more stable, across a large variety of maps. Further results may be found in Appendix 9).

5.2 Role of Centralised Value Functions

IPPO exploits centralised training solely through sharing network parameters between agent critics. However, SMAC additionally provides full state information that lifts partial observability of the environment during learning. While at first glance, conditioning the critic on full state information should

5

(a) 3m

(b) 3s5z vs 3s6z
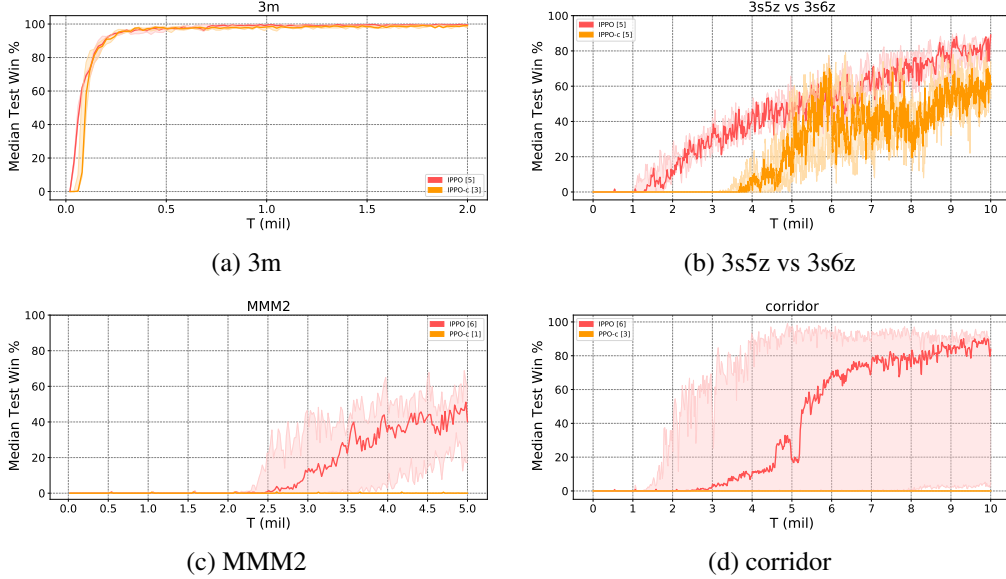
(c) MMM2

(d) corridor

Figure 3: Results on select SMAC maps comparing IPPO and MAPPO.

help simply due to it containing more information, already early empirical research in independent vs centralised learning concludes that, perhaps counter-intuitively, this is not necessarily the case as the larger size of the full state and/or it including information irrelevant to the task may adversely impact learning [27]. Nevertheless, previous work on SMAC have reported higher performance when the centralised state was being exploited during learning[19, 1] A direct comparison (see Table 5.2) shows that replacing IPPO's local critics with parameter-sharing critics conditioning on full state information performs substantially worse on a selection of hard SMAC maps, calibrating our implementation with centralised value function performance stated in concurrent work [1, MAPPO].

5.3 Ablation Studies

To understand better why IPPO performs outperforms other independent learning algorithms, including Independent Actor-Critic (IAC) [9], we investigate the performance of a number of ablations (see Figure 5.3). We find that IPPO with neither policy nor value clipping, which corresponds to a variant of IAC [9], performs poorly across all six SMAC maps studied. Furthermore, we find that policy clipping is essential to performance.

In conjunction with policy clipping, we find that value clipping improves performance on some maps (e.g., *corridor* and *MMM2*). The selective usefulness of value function clipping is in line with empirical observations in single-agent settings, which suggest that value function clipping may only be advantageous if the critic estimator suffers from high variance [13].

An alternative explanation for our findings is that IPPO's policy clipping objective reduces the effective policy learning rate, thus stabilising learning. We test this hypothesis by showing reducing the learning rate for IAC. This does not yield the anticipated performance gains relative to IPPO (see Figure 5.3).

6 Discussion

Our empirical results give rise to a number of interesting insights about the utility of IPPO in relation to other, state-of-the-art MARL algorithms, in particular those relying on value factorization methods, as well as the role of SMAC in future MARL research.

First, IPPO outperforms a number of algorithms using centralised state, including QMIX, MAPPO, and MAVEN, as well as both independent learning algorithms IAC and IQL, on a number of both

(a) corridor

(b) bane vs bane

(c) 6h vs 8z

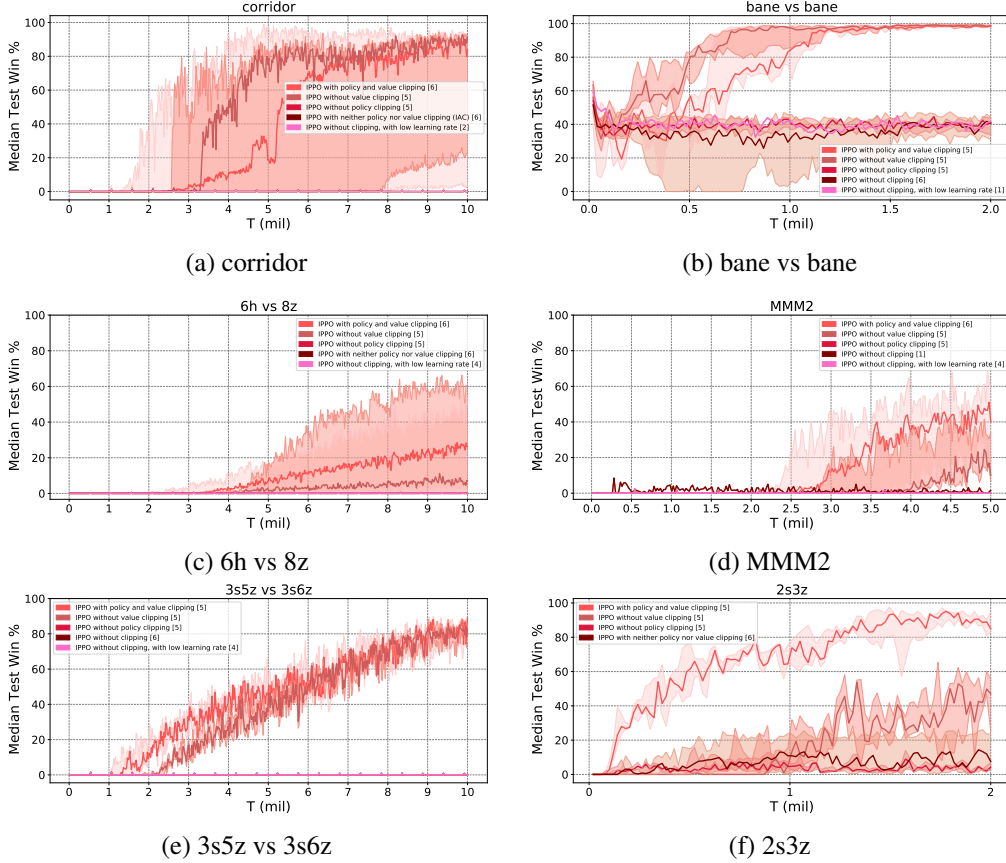(d) MMM2

(e) 3s5z vs 3s6z

(f) 2s3z

Figure 4: Ablation study for IPPO with different combinations of policy and value clipping

hard and easy SMAC maps. This is surprising, given the community's recent focus on developing MARL algorithms that can exploit state during centralised training.

Second, PPO's optimisation objective, in particular policy clipping, is crucial to performance in cooperative deep multi-agent reinforcement learning on popular benchmark environment SMAC. According to our empirical ablation studies, this effect cannot be explained through a decrease in the effective learning rate alone. Given IPPO's improved learning stability over IAC and IQL, it seems that its (approximate) surrogate objective might mitigate certain forms of environment non-stationarity that other independent learning algorithms are prone to, e.g., by suppressing updates catastrophic to performance.

Third, we show that, at least on some hard SMAC maps, the currently best performance is achieved by an algorithm that does not require the learning of neither joint nor centralised value functions, and does not exploit central state information in any other way. This implies that the fundamental obstacles to independent learning methods posed by certain pathological matrix games [6] are not present in such SMAC tasks. A possible explanation is that the sequential nature of SMAC tasks allows the decomposition of tricky simultaneous coordination tasks into a temporal sequence of easier coordination tasks. The fact that the common learning pathology of relative overgeneralisation does not seem to be present in these SMAC tasks (see Fig. 9 in the Appendix for an empirical evidence that IPPO – like many value factorisation algorithms – is in principle prone to relative overgeneralisation).

Finally, it remains unclear what exactly the value of central state information is in SMAC. Empirical ablation studies in value factorisation algorithms, such as QMIX, clearly show that using central state information on top of local observations can accelerate training. However, IPPO's strong performance on particularly hard SMAC maps and, in particular, its outperformance of MAPPO on these, raise the question as to what exactly makes central state information useful in QMIX.

map name	critic coef	entropy coef	frames	lr	mini epochs	mini batch	norm input	steps num	type	net arch
10m vs 11m	2	0.005	4	1e-4	4	2560	True	128	cnn	[256, 512, 1024]
1c3s5z	2	0.005	4	1e-4	1	4096	False	128	cnn	[64, 128, 256]
27 vs 30m	2	0.005	4	1e-4	1	2560	False	64	cnn	[64, 128, 256]
2c vs 64zg	2	0.005	4	1e-4	4	512	False	64	cnn	[64, 128, 256]
2m vs 1z	1	0.005	1	1e-4	4	1024	False	128	mlp	[256, 128]
2s3z	1	0.001	1	5e-4	4	1536	True	128	cnn	[64, 128, 256]
2s vs 1sc	2	0.005	4	1e-4	1	2048	False	128	cnn	[64, 128, 256]
3m	1	0.001	1	5e-4	4	1536	True	128	mlp	[256, 128]
3s5z	2	0.005	4	1e-4	1	4096	False	128	cnn	[64, 128, 256]
3s5z vs 3s6z	2	0.005	4	1e-4	1	4096	False	128	cnn	[64, 128, 256]
3s vs 5z	2	0.005	4	1e-4	1	1536	False	128	cnn	[64, 128, 256]
5m vs 6m	2	0.005	4	1e-4	1	2560	False	128	cnn	[64, 128, 256]
6h vs 8z	2	0.005	4	1e-4	1	3072	False	128	cnn	[64, 128, 256]
MMM	2	0.005	4	1e-4	1	2560	False	64	cnn	[64, 128, 256]
MMM2	2	0.005	4	1e-4	1	2560	False	64	cnn	[64, 128, 256]
bane vs bane	2	0.005	4	1e-4	1	3072	False	128	cnn	[64, 128, 256]
corridor	2	0.005	4	1e-4	1	3072	False	128	cnn	[64, 128, 256]

Table 1: IPPO hyperparameters across different SMAC maps. Fixed hyperparameters: $\gamma = 0.99$, $\lambda = 0.95$, $e_c lip = 0.2$, $grad\ norm = 0.5$ and $\epsilon = 0.2$. We do not employ any learning rate schedule, and always normalize the advantage function. We fix the number of actors to 8. *Net arch* indicates successive channels across layers for *conv1d* architecture, and number of activations for *mlp*. All convolutions have stride 3.

7   Conclusion

In this paper, we studied the empirical performance of Independent PPO (IPPO), a multi-agent variant of the popular PPO algorithm. Despite the purported limitations of independent learning for cooperative MARL tasks, we found that IPPO performs competitively on a range of state-of-the-art benchmark tasks, outperforming state-of-the-art value factorization methods on some maps. A number of ablation studies indicated that PPO's policy clipping objective is crucial to performance, and that the value of central state information in SMAC is unclear. Furthermore, our results raise the question of whether relative overgeneralization really matters in practice: Despite of the diversity of its scenarios, it does not seem to be an obstacle to independent learning in SMAC.

As a consequence, we suggest that the MARL community revisit the question of whether relative overgeneralisation really matters in practice. Secondly, these results suggest it would be fruitful to further improve independent learning approaches such as IPPO, rather than solely focusing research on joint value function factorisation.

8   Acknowledgments and Disclosure of Funding

References

[1] Anonymous. Benchmarking Multi-Agent Deep Reinforcement Learning Algorithms. September 2020.

[2] Lucian Busoniu, Robert Babuska, and Bart De Schutter. A Comprehensive Survey of Multiagent Reinforcement Learning. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38:156–172, April 2008.

[3] Wendelin Böhmer, Vitaly Kurin, and Shimon Whiteson. Deep Coordination Graphs. *arXiv:1910.00091 [cs]*, June 2020. arXiv: 1910.00091.

[4] Wendelin Böhmer, Tabish Rashid, and Shimon Whiteson. Exploration with Unreliable Intrinsic Reward in Multi-Agent Reinforcement Learning. *arXiv:1906.02138 [cs]*, June 2019. arXiv: 1906.02138.

[5] Yongcan Cao, Wenwu Yu, Wei Ren, and Guanrong Chen. An Overview of Recent Progress in the Study of Distributed Multi-agent Coordination. *arXiv:1207.3231 [math]*, September 2012. arXiv: 1207.3231.

[6] C. Claus and Craig Boutilier. The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems. In *AAAI/IAAI*, 1998.

[7] Christian Schroeder de Witt, Bei Peng, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Böhmer, and Shimon Whiteson. Deep Multi-Agent Reinforcement Learning for Decentralized Continuous Cooperative Control. *arXiv:2003.06709 [cs, stat]*, June 2020. arXiv: 2003.06709.

[8] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation Matters in Deep Policy Gradients: A Case Study on PPO and TRPO. *arXiv:2005.12729 [cs, stat]*, May 2020. arXiv: 2005.12729.

[9] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual Multi-Agent Policy Gradients. *arXiv:1705.08926 [cs]*, December 2017. arXiv: 1705.08926.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[11] Chloe Ching-Yun Hsu, Celestine Mendler-Dünner, and Moritz Hardt. Revisiting Design Choices in Proximal Policy Optimization. *arXiv:2009.10897 [cs, stat]*, September 2020. arXiv: 2009.10897.

[12] Maximilian Hüttenrauch, Adrian Šošić, and Gerhard Neumann. Guided Deep Reinforcement Learning for Swarm Systems. *arXiv:1709.06011 [cs, stat]*, September 2017. arXiv: 1709.06011.

[13] ikostrikov. Value Function Clipping for PPO · Issue #136 · ikostrikov/pytorch-a2c-ppo-acktr-gail.

[14] Landon Kraemer and Bikramjit Banerjee. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing*, 190:82–94, May 2016.

[15] Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. MAVEN: Multi-Agent Variational Exploration. *arXiv:1910.07483 [cs, stat]*, January 2020. arXiv: 1910.07483.

[16] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous Methods for Deep Reinforcement Learning. *arXiv:1602.01783 [cs]*, June 2016. arXiv: 1602.01783.

[17] Frans A. Oliehoek and Christopher Amato. *A Concise Introduction to Decentralized POMDPs*. SpringerBriefs in Intelligent Systems. Springer International Publishing, 2016.

[18] Liviu Panait and Sean Luke. Cooperative Multi-Agent Learning: The State of the Art. *Autonomous Agents and Multi-Agent Systems*, 11:387–434, November 2005.

[19] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. *arXiv:2003.08839 [cs, stat]*, August 2020. arXiv: 2003.08839.

[20] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob Foerster, and Shimon Whiteson. The StarCraft Multi-Agent Challenge. *arXiv:1902.04043 [cs, stat]*, December 2019. arXiv: 1902.04043.

[21] John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust Region Policy Optimization. *arXiv:1502.05477 [cs]*, April 2017. arXiv: 1502.05477.

[22] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-Dimensional Continuous Control Using Generalized Advantage Estimation. *arXiv:1506.02438 [cs]*, October 2018. arXiv: 1506.02438.

[23] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[24] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. QTRAN: Learning to Factorize with Transformation for Cooperative Multi-Agent Reinforcement Learning. *arXiv:1905.05408 [cs, stat]*, May 2019. arXiv: 1905.05408.

[25] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. Value-Decomposition Networks For Cooperative Multi-Agent Learning. *arXiv:1706.05296 [cs]*, June 2017. arXiv: 1706.05296.

[26] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning, second edition: An Introduction*. Bradford Books, Cambridge, Massachusetts, second edition edition edition, November 2018.

[27] Ming Tan. Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents. In *In Proceedings of the Tenth International Conference on Machine Learning*, pages 330–337. Morgan Kaufmann, 1993.

[28] Kagan Tumer and Adrian Agogino. Distributed agent-based air traffic flow management. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, AAMAS '07, pages 1–8, New York, NY, USA, May 2007. Association for Computing Machinery.

[29] Karl Tuyls and Gerhard Weiss. Multiagent Learning: Basics, Challenges, and Prospects. *AI Magazine*, 33(3):41–41, September 2012. Number: 3.

[30] Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, John Quan, Stephen Gaffney, Stig Petersen, Karen Simonyan, Tom Schaul, Hado van Hasselt, David Silver, Timothy Lillicrap, Kevin Calderone, Paul Keet, Anthony Brunasso, David Lawrence, Anders Ekermo, Jacob Repp, and Rodney Tsing. StarCraft II: A New Challenge for Reinforcement Learning. *arXiv:1708.04782 [cs]*, August 2017. arXiv: 1708.04782.

[31] Yuhui Wang, Hao He, Chao Wen, and Xiaoyang Tan. Truly Proximal Policy Optimization. *arXiv:1903.07940 [cs, stat]*, January 2020. arXiv: 1903.07940.

[32] Ermo Wei and Sean Luke. Lenient Learning in Independent-Learner Stochastic Cooperative Games. *Journal of Machine Learning Research*, 17(84):1–42, 2016.

[33] Steven Whitehead. A Complexity Analysis of Cooperative Mechanisms in Reinforcement Learning. pages 607–613, January 1991.

9 Appendix

(a) 2s vs 1sc

(b) bane vs bane

(c) MMM

(a) 2s3z

(b) 3s5z

(c) 1c3s5z

(g) 5m vs 6m

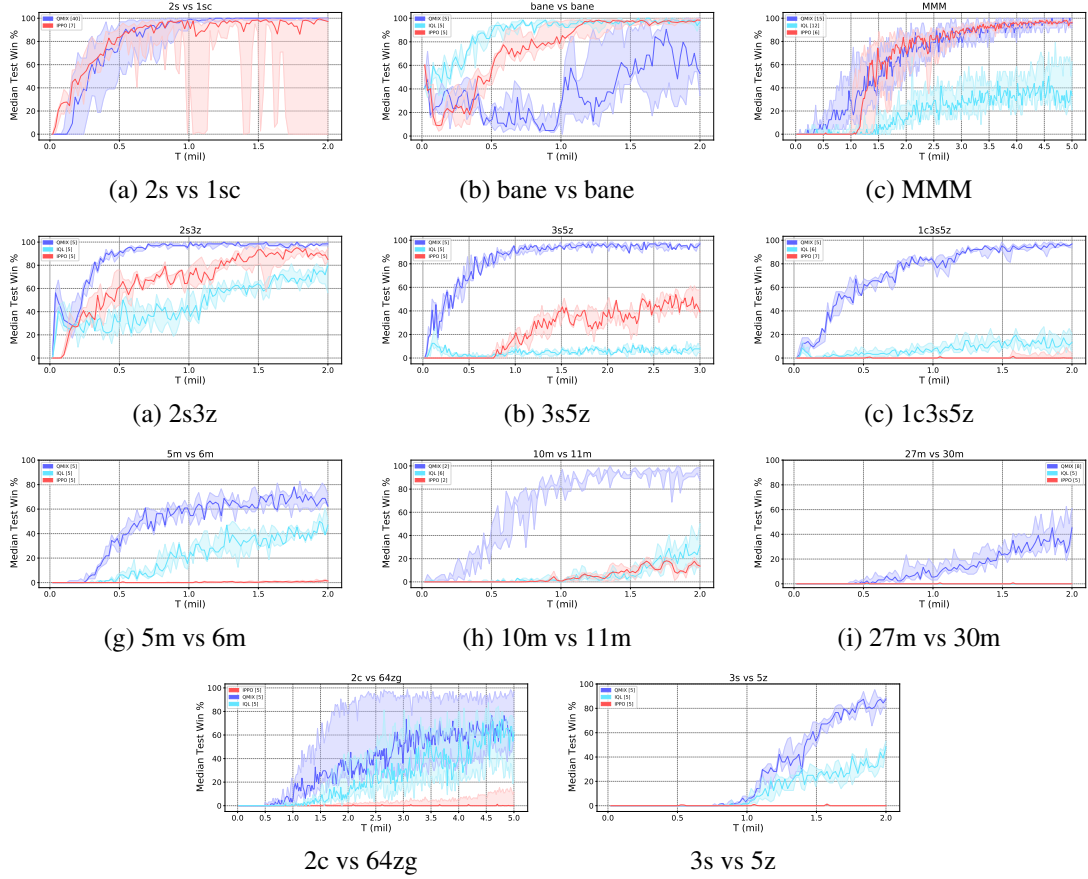(h) 10m vs 11m

(i) 27m vs 30m

2c vs 64zg

3s vs 5z

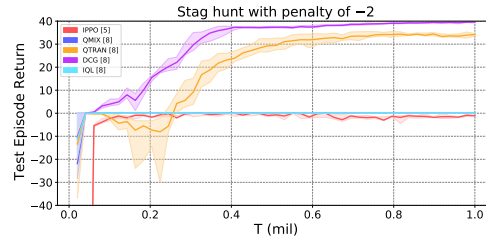Figure 5: Results on select SMAC maps comparing IPPO and IPPO-C with QMIX, VDN and IQL

Figure 6: Relative overgeneralization, as experienced by select MARL algorithms on Stag Hunt with penalty -2 (see [3]). Note that DCG is not fully decentralized during execution.