Counter-Example Guided Policy Refinement in Multi-agent Reinforcement Learning

Briti Gangopadhyay
IIT Kharagpur
India
briti_gangopadhyay@iitkgp.ac.in

Pallab Dasgupta
IIT Kharagpur
India
pallab@cse.iitkgp.ac.in

Soumyajit Dey
IIT Kharagpur
India
soumya@cse.iitkgp.ac.in

A ADDITIONAL DETAILS OF CACC ENVIRONMENT

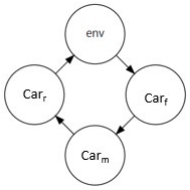

Figure 1: Agent Environment Cycle game structure for CACC environment

We formulate the CACC environment as an Agent Environment Cycle (AEC) game. The environment has three vehicles. Each vehicle follows the Kinematic Bicycle Model [2]. Car_f, Car_m and Car_r denote the front middle and rear vehicles respectively. When one vehicle applies an action the states of the other vehicles persist as per the previous action. The observation space for each Car_i is ($\Delta target_i$, Δx_{lonf_i},Δx_{lonr_i}, v_i, Δ_{yaw_i},v_j,Δx_{lonf_j},Δx_{lonr_j}) where $\Delta target_i$ is the distance of the vehicle from the target of the i^{th} vehicle, Δx_{lonf_i},Δx_{lonr_i} denote the longitudinal distance from the front and rear car respectively, v_i is the vehicle's velocity and Δ_{yaw_i} is the difference of the vehicle's position from the given way-points. Additionally, each vehicle receives the speed of the other vehicles denoted by v_j and their longitudinal distance from the front and rear car respectively. The reward function is formulated as follows:

$$\left\{\begin{array}{ll} r_c = -C_1 & \text{Collision} \\ r_l = -1 * (\Delta yaw_i) & \text{Deviation route} \\ r_v = -1 * |(v_{target} - v_i)| & \text{Diff Target speed} \\ r_{lon} = 1 * (\Delta x_{lonf_i} + \Delta x_{lonr_i}) & \text{Safe Distance} \end{array}\right\} \quad (1)$$

$path_{error}$ is calculated as the euclidean distance from the target way-point for each vehicle. Now we illustrate how safety specification are converted into an optimization formulation with derivation of a safety specification from the CACC environment. Consider the specification $\forall i, j \in car\ dist(i, j) > 0 \implies speed_{platoon} > 0$. We expand the specification as follows:

$$\begin{aligned} &\{dist(Car_r, Car_m) > 0 \wedge dist(Car_m, Car_f) > 0) \implies \\ &\quad (Speed_r > 0 \wedge Speed_m > 0 \wedge Speed_f > 0)\} \\ &\equiv \{\neg(dist(Car_r, Car_m) > 0 \wedge dist(Car_m, Car_f) > 0))\vee \\ &\quad (Speed_r > 0 \wedge Speed_m > 0 \wedge Speed_f > 0)\} \end{aligned} \quad (2)$$

We take the negation of the safety specification for deriving the objective function:

$$\begin{aligned} &\neg\{\neg(dist(Car_r, Car_m) > 0 \wedge dist(Car_m, Car_f) > 0))\vee \\ &\quad (Speed_r > 0 \wedge Speed_m > 0 \wedge Speed_f > 0)\} \\ &\equiv \{(dist(Car_r, Car_m) > 0 \wedge dist(Car_m, Car_f) > 0))\wedge \\ &\quad (Speed_r < 0 \vee Speed_m < 0 \vee Speed_f < 0)\} \end{aligned} \quad (3)$$

This is converted into the optimization problem following [1] as follows:

$$\begin{aligned} &min\{min(max(dist(Car_r, Car_m)), \\ &\quad max(dist(Car_m, Car_f))) \\ &, (max(min(Speed_r), min(Speed_m), min(Speed_f))\} \end{aligned} \quad (4)$$

The Graph in Figure 3b (main text) shows high variance during training because CACC is an unstable environment. Slight variations in acceleration and steering can lead to crashes and termination of episodes with negative rewards. MAPPO being a stochastic policy may choose an unsafe action even toward the end of training.

B ADDITIONAL EXPERIMENTAL DETAILS

We use 200 iterations of Bayesian Optimization to uncover counterexamples for different specifications over parameters of uncertainties for various entities in the environment. We reported the value of the samples selected and the corresponding objective function evaluation for different specifications in the multi-walker environment in Figure 4 a), b), and c) (main text). Though multiple state variables contribute to the violation of the specification, only samples of hull angle are plotted for comprehensibility. In order to demonstrate that the refinement methodology corrects the old counterexamples without introducing any new counterexamples to the policy, we retested the refined policy π_{safe} with 200 BO iterations. Figure 4 d), e) and f) (main text) illustrates that all objective function evaluations are positive for all specifications in the refined policy π_{safe}.

We illustrate similar results hold for the CACC environment in Figure 3. Secondary training is required to correct paths in CACC where the vehicles speed up and crash in order to reach the goal. Secondary training prevents these paths by encouraging the agents to maintain lower speed over crashing. The samples are shown only with respect to the position of the middle car in Fig 3. We had shown the comparison of secondary policy training time in the main article. However, these plots were not visible due to low training time. In Figure 2 we illustrate the secondary policy training time and the reward obtained for each environment. The plots demonstrate that seconadry training takes lower time due to a low number of trajectories. Our experiments took 45K steps for multi-walker, 18K steps for CACC, and 4M steps for multi-ant. In Figure

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

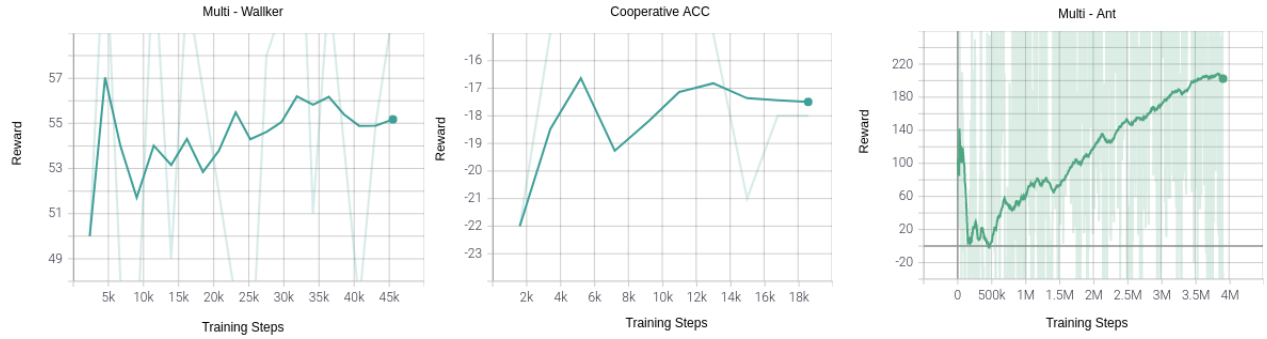

Figure 2: Reward vs training step pots for secondary training in multi-walker, Cooperative ACC and multi-ant environments

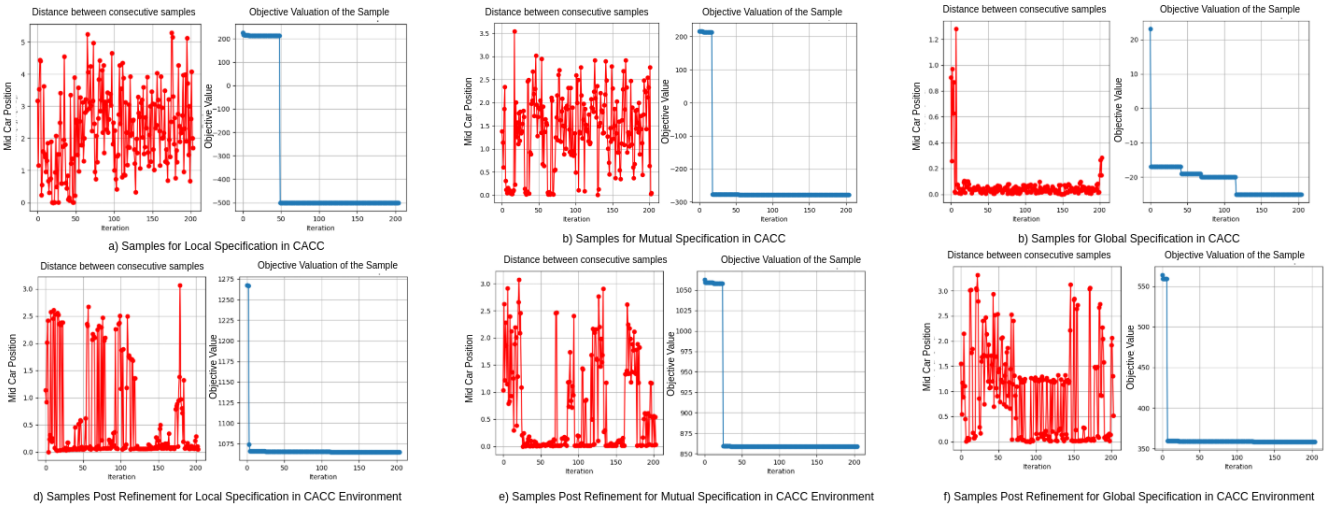

Figure 3: Plots of Bayesian Optimization samples collected and the objective function value for different a) Local b) Mutual c) Global specifications for Cooperative ACC environment. Plots d) e) f) illustrates samples collected post refinement which show no further specification violations.

Table 1: Feed Forward Neural Network architectures used to train original MARL policies, sub policies and safe policies on different multi-agent environments.

Network Type	Layers	Dimension	Activation Functions
Actor	Hidden Layer 1	64 Units	ReLU
	Hidden Layer 2	64 Units	ReLU
	Output Layer	Action Space Shape	Discrete: Softmax Continuous: Normal Distribution
Critic	Hidden Layer 1	64 Units	ReLU
	Hidden Layer 2	64 Units	ReLU
	Output Layer	1 Unit	

4 we illustrate the change in layer-wise weight distribution due to refinement showing the shift of weights in the policy as a result of refinement.

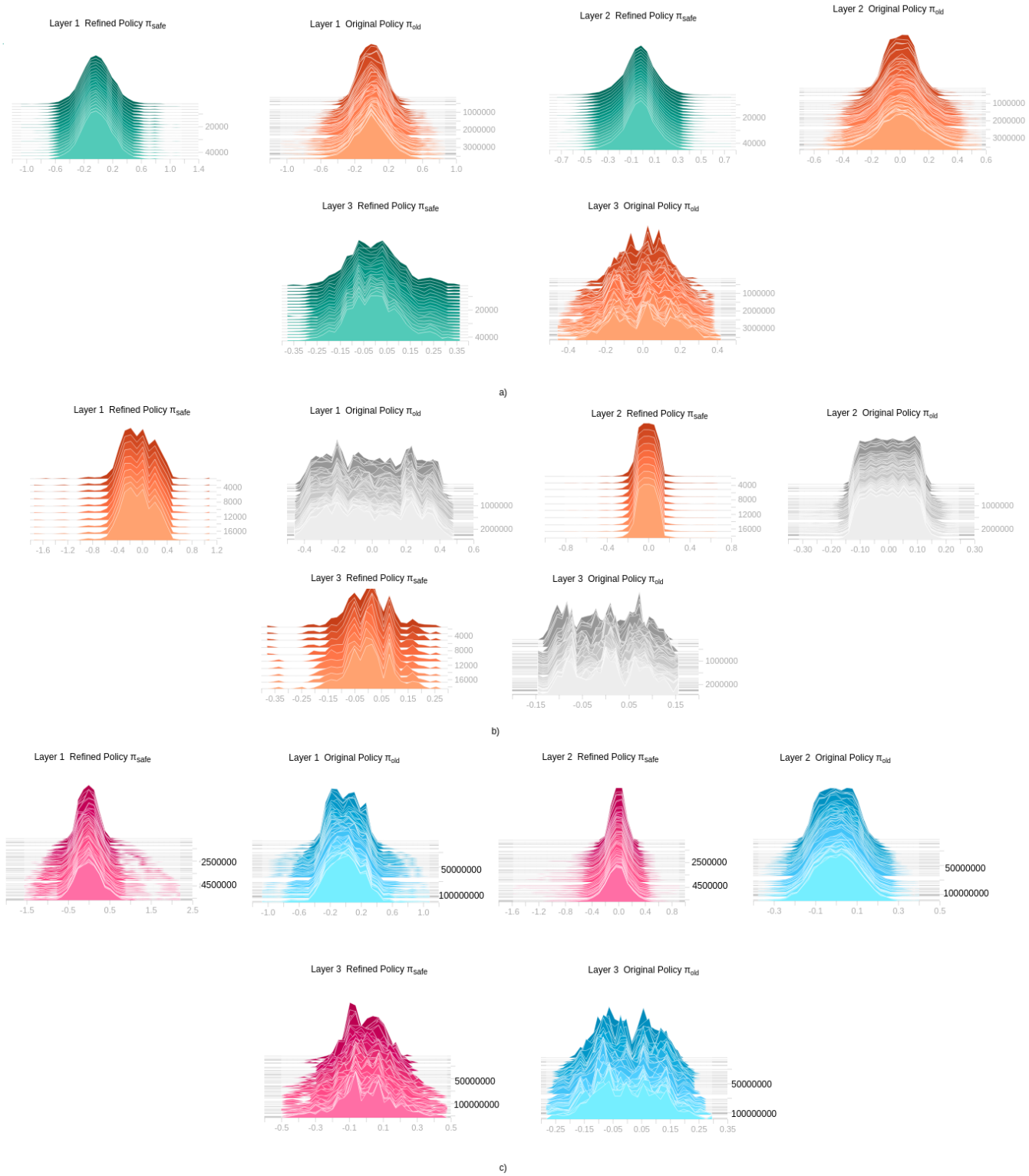

Figure 4: Change in weight distribution over multiple iterations of gradient update of the original policy and the updated policy for a) multi-walker, b) Cooperative ACC, c) muulti-ant environments. X axis denotes the weight distribution.

Hyperparameter	Value
Horizon (T)	1000
Adam step-size	$6.0e^{-4}$
Number of Updates per iteration	10
Minibatch size	64
clip	0.3
Discount (γ)	0.99
GAE λ	0.95

Table 2: Hyper-parameters for multi-walker environment

Hyperparameter	Value
Horizon (T)	570
Adam step-size	$2.5e^{-4}$
Number of Updates per iteration	10
Minibatch size	64
clip	0.2
Discount (γ)	0.99
GAE λ	0.95

Table 3: Hyper-parameters for Cooperative ACC environment

Hyperparameter	Value
Horizon (T)	1000
Adam step-size	$9e^{-5}$
Number of Updates per iteration	15
Number of training threads	4
Number of rollout threads	16
Minibatch size	40
clip	0.2
Discount (γ)	0.99
GAE λ	0.95

Table 4: Hyper-parameters for Multi-ant environment

C IMPLEMENTATION DETAILS

The shared Feed Forward Neural Network architectures mentioned in Table 1 are used to train original MARL policies, sub-policies, and safe policies on different multi-agent environments. The hyper-parameters used for each environment is reported in the Tables 2, 3 and 4.

ACKNOWLEDGMENTS

If you wish to include any acknowledgments in your paper (e.g., to people or funding agencies), please do so using the 'acks' environment. Note that the text of your acknowledgments will be omitted if you compile your document with the 'anonymous' option.

REFERENCES

[1] S. Ghosh, F. Berkenkamp, G. Ranade, S. Qadeer, and A. Kapoor. 2018. Verifying Controllers Against Adversarial Examples with Bayesian Optimization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 7306–7313. https://doi.org/10.1109/ICRA.2018.8460635

[2] Jason Kong, Mark Pfeiffer, Georg Schildbach, and Francesco Borrelli. 2015. Kinematic and dynamic vehicle models for autonomous driving control design. In *2015 IEEE Intelligent Vehicles Symposium (IV)*. 1094–1099. https://doi.org/10.1109/IVS.2015.7225830