Safe Policy Learning in Online Reinforcement Learning Using Augmented PPO and CMDPs for Robot Navigation

Zhongna Zhou, K. C. Ho*, Zhao Sun, Trevor Tran †‡

1 Abstract

This work focuses on online reinforcement learning (RL) in the presence of environmental constraints. Specifically, we consider applications involving robot agents exploring in an environment where obstacles and unsafe zones are present, and the agents must maximize cumulative rewards and at the same time meet the environmental constraints. To address this challenge, we formulate the problem using the constrained Markov Decision Process (CMDP) and incorporate the environmental constraint costs into the policy updates in the proposed Augmented Proximal Policy Optimization (APPO) algorithm. At each state and for each possible action, we apply a Variational Auto-Encoder (VAE) [1] to obtain a probabilistic estimate of the discounted cumulative future environmental constraint costs and integrate them as a regularization term to the reward function. This augmented reward function updates the action-value functions within the APPO algorithm, which is trained by an efficient optimization scheme. Experimental results demonstrate that our methodology enables robot agents to navigate within the safety-constrained regions effectively.

*Zhongna Zhou and K. C. Ho are with the Electrical Engineering and Computer Science Department, University of Missouri, Columbia, MO 65211, USA (e-mail: zz3kb@missouri.edu, hod@missouri.edu)

†Zhao Sun and Trevor Tran are with the Electrical and Computer Engineering Department, Hampton University, Hampton, VA 23669, USA (e-mail: zhao.sun@hamptonu.edu, trevor.tran@my.hamptonu.edu)

‡This work was supported in part by National Science Foundation under grant award number 2101227.

2 Introduction

We handle constraints for online RL by formulating a CMDP with components:

$$(\mathcal{S}, \mathcal{A}, P, r),$$

where $\mathcal{S}$ is the state space (e.g., agent's 2D position), $\mathcal{A}$ is the action space, $P(s'|s,a)$ defines the transition dynamics, and $r(s,a)$ is the reward function with $s, s' \in S$, $a \in A$. In this setting, the agent aims to maximize the expected cumulative rewards while satisfying certain constraints such as obstacle avoidance and safety passages for navigation. The optimization problem is formulated as

$$\text{Maximize } J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t \, r(s_t, a_t)\right], \tag{1}$$

$$\text{Subject to } C_i(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t \, c_i(s_t, x, y)\right] \leq d_i, \quad i = 1, \ldots, m, \tag{2}$$

where $J(\theta)$ is the accumulated rewards, π_θ is the policy, $C_i(\theta)$ is the overall constraints cost, θ are the policy parameters, $r(s_t, a_t)$ is the reward function at time t with state $s_t \in S$ and action $a_t \in A$, γ is the forgetting factor, $c_i(s_t, x, y)$ are the constraint cost functions, d_i are the constraint thresholds, and m is the number of environmental constraints.

3 Proposed Method

3.1 Learning a Probabilistic Safety Shield of Unsafe Zones with a VAE

We propose to learn a probabilistic safety shield for unsafe zones [2] with a VAE. Assume that the robot explores a planar workspace $\Omega \subset \mathbb{R}^2$. At discrete time steps $t \in \{1, \ldots, T\}$, it records the location

$$\boldsymbol{\ell}_t = (x_t, y_t) \in \Omega, \qquad u_t \in \{0, 1\}, \tag{3}$$

where $u_t = 1$ indicates that the position is *unsafe* or be avoided. Given the dataset $\mathcal{D} = \{((x_i, y_i), u_i)\}_{i=1}^{N}$ from N exploration steps, we apply a VAE to

estimate a *belief function*

$$b : \Omega \to [0,1], \qquad b(\boldsymbol{\ell}) = \Pr\big(u = 1 \mid \boldsymbol{\ell}, \mathcal{D}\big), \tag{4}$$

which returns the probability that a location is unsafe. Introducing a latent vector $\boldsymbol{z} \sim p(\boldsymbol{z}) = \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$, we model the encoder as:

$$q_\phi(\boldsymbol{z} \mid \boldsymbol{\ell}, u) = \mathcal{N}\big(\boldsymbol{z}; \boldsymbol{\mu}_\phi(\boldsymbol{\ell}, u), \mathrm{diag}(\boldsymbol{\sigma}^2_\phi(\boldsymbol{\ell}, u))\big). \tag{5}$$

Sampling uses the reparameterization $\boldsymbol{z} = \boldsymbol{\mu}_\phi + \boldsymbol{\sigma}_\phi \odot \boldsymbol{\varepsilon}$ with $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$. The decoder for Locations conditioned on $\boldsymbol{z}$ is,

$$p_\theta(\boldsymbol{\ell} \mid \boldsymbol{z}) = \mathcal{N}\big(\boldsymbol{\ell}; \boldsymbol{\mu}_\theta(\boldsymbol{z}), \mathrm{diag}(\boldsymbol{\sigma}^2_\theta(\boldsymbol{z}))\big), \tag{6}$$

and the decoder for Safety Labels is:

$$p_\theta(u = 1 \mid \boldsymbol{z}) = \sigma\big(g_\theta(\boldsymbol{z})\big), \qquad \sigma(s) = \tfrac{1}{1+e^{-s}}. \tag{7}$$

Furthermore, $p_\theta(\boldsymbol{\ell}, u \mid \boldsymbol{z}) = p_\theta(\boldsymbol{\ell} \mid \boldsymbol{z})\, p_\theta(u \mid \boldsymbol{z})$.
After Training, for a given new location $\boldsymbol{\ell}(x, y)$ without a label, we form the approximate posterior

$$q_\phi(\boldsymbol{z} \mid \boldsymbol{\ell}) = \mathcal{N}\big(\boldsymbol{z}; \boldsymbol{\mu}_\phi(\boldsymbol{\ell}), \mathrm{diag}(\boldsymbol{\sigma}^2_\phi(\boldsymbol{\ell}))\big), \tag{8}$$

and estimate the environment constraint cost $c_i(s_t, x, y)$ as the inverse of the safety probability (logistic function),

$$p = \sigma(s) = \tfrac{1}{1+e^{-s}} \tag{9}$$

$$1 + exp(-x) = 1/p \tag{10}$$

$$c_i(s_t, x, y) \approx \frac{1}{\sigma\big(g_\theta(\boldsymbol{z})\big)}. \tag{11}$$

The smaller is the safety probability, the larger is the cost.

3.2 Augmented Proximal Policy Optimization Algorithm

APPO is a refined Trust Region Policy Optimization that uses a value network and a policy network for modeling. Based on the APPO framework, we propose an APPO algorithm (Algorithm 1) that incorporates the environment

constraints to the policy update functions, with the environmental constraint costs from the VAE integrated as a regularization term to the policy updates. By incorporating the probabilistic environment constraint cost, the policy can be learned to avoid the unsafe zone in the environment. The details corresponding to the steps in Algorithm 1 are as follows.

Value Function:

$$V_\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \, r(s_t, a_t) \,\middle|\, s_0 = s \right]. \tag{12}$$

Modified Reward:

$$r'(s,a) = r(s,a) - \alpha \, \hat{c}_i(s,x,y), \tag{13}$$

with $\alpha > 0$ serving as a scaling factor for penalty intensity for incorporating the learned environment cost.

Action-Value Function Update:

$$Q_\pi(s,a) = r'(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s' \mid s,a) \sum_{a' \in \mathcal{A}} \pi(a' \mid s') \, Q_\pi(s',a'). \tag{14}$$

where $Q_\pi(s',a')$ is modeled by the value network and $\pi(a' \mid s')$ by the policy network, and $P(s' \mid s,a)$ is from the parametric action distribution.

Advantage Function:

$$A_\pi(s,a) = Q_\pi(s,a) - V_\pi(s). \tag{15}$$

Policy Loss:

$$\mathcal{L}_\pi = \mathbb{E}_{s,a} \Big[-\log \pi(a \mid s) \, A_\pi(s,a) \Big]. \tag{16}$$

Value Loss:

$$\mathcal{L}_v = \text{mean} \Big((\text{advantage policy values} - \text{baseline policy values})^2 \Big) \times 0.25. \tag{17}$$

where 0.25 is a scaling coefficient used in the experiment.

Entropy Loss:

$$\mathcal{L}_e = -\mathbb{E}_{s,a}\Big[\pi(a \mid s) \log \pi(a \mid s)\Big]. \tag{18}$$

Total Loss for Agent Training:

$$\mathcal{L} = \mathcal{L}_\pi + \mathcal{L}_v + \mathcal{L}_e. \tag{19}$$

Algorithm 1 Augmented Proximal Policy Optimization with Environment Constraints

Input: Policy parameters θ, learning rate η_0, discount factor γ, and environment constraint definitions.

for each iteration $k = 0, 1, 2, \ldots$ **do**

Policy Evaluation: Compute $V_\pi(s)$ as in Eq. (12).

Environment Cost: Update the probabilistic environment constraint cost using the VAE (see Eq. (11))

Modified Reward: Compute $r'(s, a)$ according to Eq. (13).

Action-Value and Advantage: Compute $Q_\pi(s, a)$ and $A_\pi(s, a)$ based on the policy network and value network (see Eqs. (14) and (15)).

Loss Components: Compute policy loss $\mathcal{L}_\pi$ by Eq. (16), value loss $\mathcal{L}_v$ by Eq. (17), and entropy loss $\mathcal{L}_e$ by Eq. (18).

Total Loss: Form the total loss $\mathcal{L} = \mathcal{L}_\pi + \mathcal{L}_v + \mathcal{L}_e$ (see Eq. (19)).

Policy Update: Update parameters:

$$\theta \leftarrow \theta - \eta_0 \nabla_\theta \mathcal{L}.$$

end for

4 Experimental Results

In the experiment, we train the VAE for computing the environment constraints cost for each position, obtain a cost probability value, then apply it in the APPO policy updating.

The simulation used the MuJoCo simulator [3] with a robot agent. The experiments were conducted on a system equipped with an NVIDIA T4 GPU (15 GB GPU RAM), 51 GB of system RAM, and a 236 GB disk. The software stack

includes JAX and OpenAI Safety Gym, and the implementation was performed in Python using Google Colab. The environment is a maze with black holes. We treat these black holes as unsafe zones and they are indicated in Figure 1 by the red dots.

Our method extends the PPO framework by combining a VAE to learn the probabilistic environment constraint cost and then uses it in the modified reward functions during policy network updating. Figure 2 shows the safety cost distribution, the closer to the unsafe zone, the higher is the cost. Figure 1 illustrates a trajectory after applying the learned policy with the probabilistic shield (constraints) cost. The agent can avoid the unsafe zone and reach the goal point. During our experiments, we observe that without the probability shield cost, we can't prevent the robot agent from falling into the unsafe zone (Figure 3). If the agent enters an unsafe zone, it will return to its starting point. The results show that the probabilistic environment constraint cost can direct the agent according to the safety constraints and produce an optimized trajectory to reach the destination.

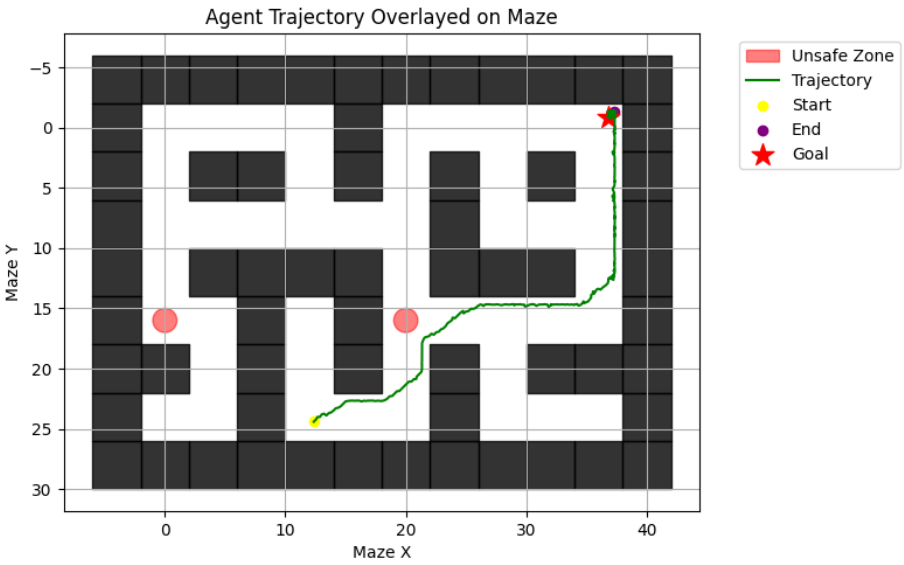

Figure 1: Online RL trajectory with the probabilistic environment safety constraint cost

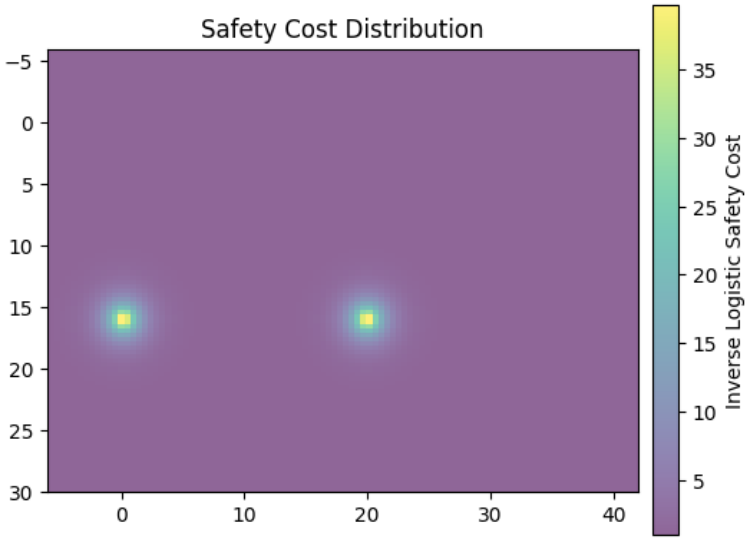

Figure 2: Probabilistic safety cost distribution

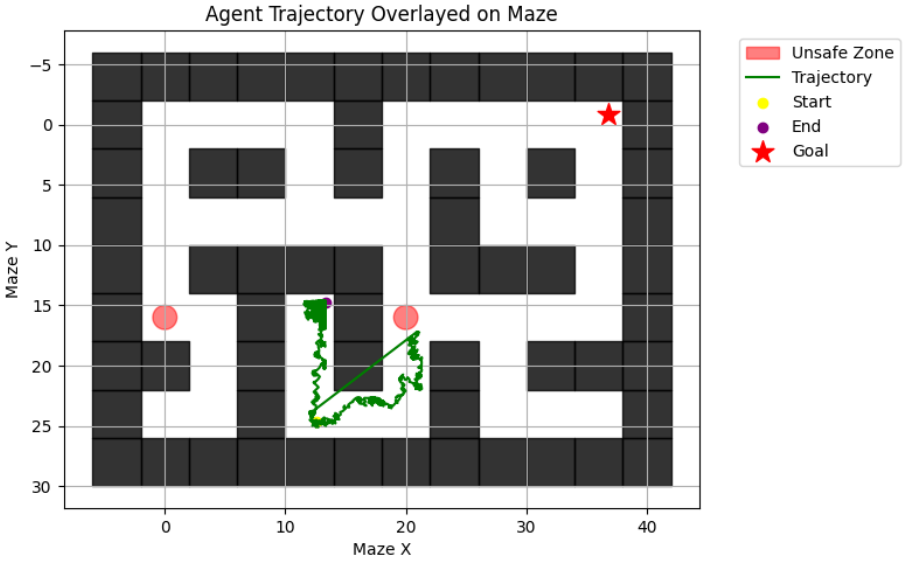

Figure 3: Illustration of online RL without the proposed probabilistic environment safety constraint cost

5 Conclusion

We proposed the use of the inverse of the safety probability as the probabilistic environment constraints cost and developed the APPO algorithm that leverages multi-step loss components for robot navigation in an environment where unsafe zones or obstacles are present. We developed a VAE to learn and then infer the safety probability, to form the probabilistic environment constraints cost. The cost is used to modify the reward value functions to update the policy network. Experimental results show that the probabilistic environment constraints cost improves the optimum trajectory of the agent. Without the probabilistic safety shield, the agent cannot prevent from falling into an unsafe/restricted area.

References

[1] B. Ivanovic, K. Leung, E. Schmerling, and M. Pavone, "Multimodal Deep Generative Models for Trajectory Prediction: A Conditional Variational Autoencoder Approach," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 295-302, Apr. 2021.

[2] H. Odriozola-Olalde, M. Zamalloa, and N. Arana-Arexolaleiba, "Shielded Reinforcement Learning: A review of reactive methods for safe learning," in *Proc. IEEE SICE International Symposium on System Integration*, Atlanta, USA, Jan. 2023.

[3] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *Proc. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Algarve [Portugal], Oct. 2012, pp. 5026–5033.

[4] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.