Nightmare Dreamer: Dreaming About Unsafe States And Planning Ahead

Oluwatosin Oseni, Shengjie Wang, Jun Zhu and Micah Corah

Abstract

Model-based Safe RL algorithm that proactively “dreams” about unsafe future states and plans preventive actions. We adopt a bi-actor architecture with predictive planning that switches between control and safety policies based on anticipated violations.

Approach

Main components of Nightmare Dreamer approach for Safe-RL:

World Model Learning: Learn environment dynamics, including safety violations for predictive planning.

Predictive Planning: Uses world model rollouts to anticipate violations.

Bi-Actor Architecture: Separate policies for reward maximization **(Control)** and safety constraints **(Safe)** (switching between policies based on potential safety violation)

- **Control policy:** Optimizes rewards
- **Safe policy:** Optimizes constraints while imitating control actions through **discriminator-based regularization**. The safe policy fools a discriminator to mimic control actions.

World Model Learning

Goal: Learn environment dynamics **including** safety violations

Key Components:

- **Recurrent Model:** Temporal dependencies
- **Cost/Reward Model:** Predicts safety violations an task performance respectively
- **Transition Model:** State dynamics

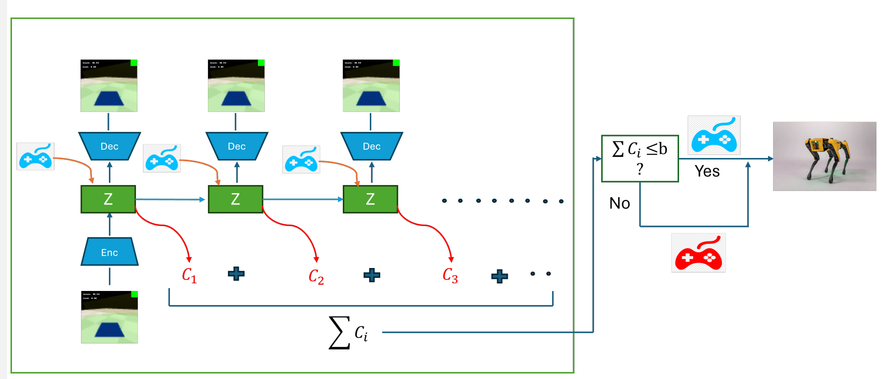
Why Important: Enables predicting future safety violations *before* they happen

Model Loss Function:

$$\mathcal{L}(\epsilon) \doteq \sum_{t=1}^{T} \underbrace{-\alpha_c \ln(p_\epsilon(c_t|h_t,z_t))}_{\text{cost log loss}} \underbrace{-\alpha_r \ln(p_\epsilon(r_t|h_t,z_t))}_{\text{reward log loss}} \underbrace{-\ln(p_\epsilon(o_t|h_t,z_t))}_{\text{reconstruction loss}} \underbrace{-\ln(p_\epsilon(y_t|h_t,z_t))}_{\text{discount log loss}}. + \underbrace{\mathrm{KL}\left[q_\epsilon(z_t|h_t,o_t)\,||\,sg(p_\epsilon(z_t|h_t))\right]}_{\text{representation loss}}$$

Algorithm for Safe Action Selection

The blue gamepad signifies the action from the Controller, while the red gamepad refers to an action from the Safe Actor.



Algorithm Planning Ahead of Risks for Safe Action Selection

Input: Current state s_t, safety budget b_s
Output: Action a_t to execute
Compute current cost $C_t(h_t, z_t, o_t)$ based on current observation
Initialize $C_{\text{sum}} \leftarrow C_t(h_t, z_t, so_t)$
for $i \leftarrow 1$ ***to*** H **do**
  Predict next latent state using learned dynamics model Estimate cost C_{t+i} for predicted state $C_{\text{sum}} \leftarrow C_{\text{sum}} + C_{t+i}$
end
if $C_{\text{sum}} > b_s$ **then**
  $a_t \sim \pi_\rho(a|s_t)$; `// Sample action from safe policy`
else
  $a_t \sim \pi_\phi(a|s_t)$; `// Sample action from Control policy`
end
return a_t

Safe and Control Policy Learning

- **Control Policy:** We train a Control Policy using rollouts from World Model
- **Safe Policy:** We train the Multi-Objective loss function that minimizes Cost while maximising Reward by Imitating the Control Policy Actions

$$\mathcal{L}(\rho) \doteq \sum_{t=1}^{H-1}\Big(\underbrace{\lambda_p C_t^{\lambda}}_{\text{target cost value}} \quad \underbrace{-D(a_t, s_t)}_{\text{control policy behaviour imitation}} \quad \underbrace{-\eta H[\pi_\phi(a_t|s_t)]}_{\text{entropy regularizer}} \Big).$$

Solving the Multi-Objective Optimization using the classic Primal-Dual Method

$$\pi_* = \arg\max_{\pi_\theta} J^R(\pi_\theta) \quad \text{s.t} \quad J^C(\pi_\theta) \le b$$

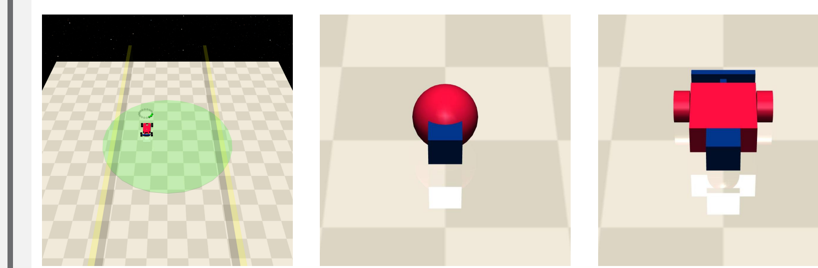
Multi-Objective Optimization Formulation

$$\min_{\pi_\phi}\max_{\lambda_p \ge 0} \; J_{\text{task}}(\pi_\phi) - \lambda_p(J_{\text{constraint}}(\pi_\rho) - b$$

Primal-dual via the Lagrangian Method

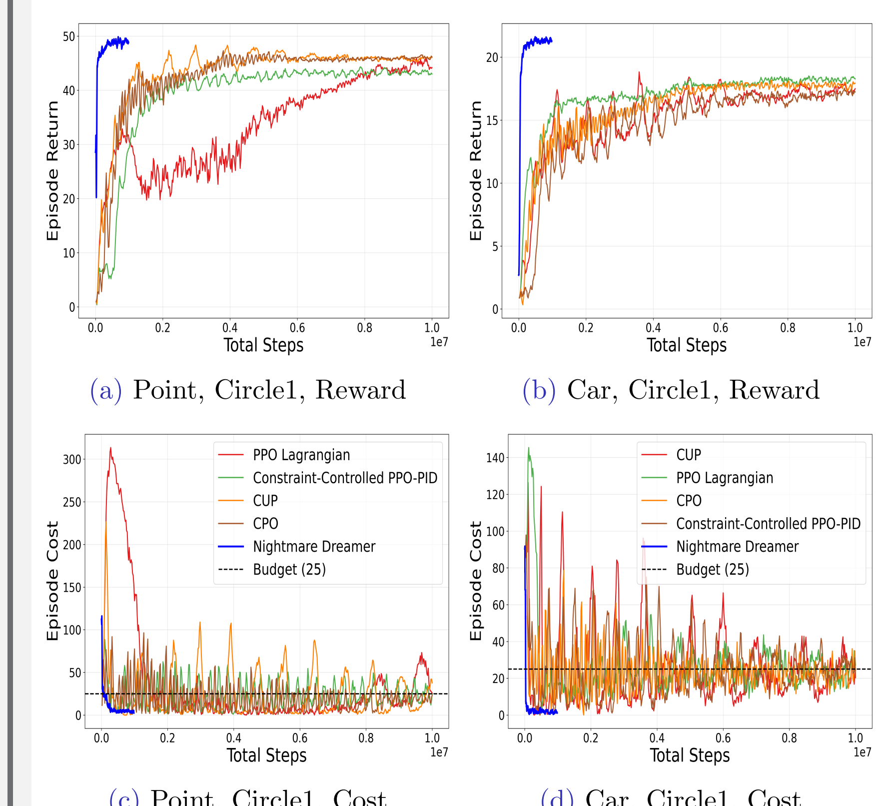
Experimental Results

Experiments on Circle 1 environment and Safety-Gymnasium agents



(a) Circle 1 (b) Point (c) Car

Circle 1 Performance Comparison with Benchmarks



(a) Point, Circle1, Reward (b) Car, Circle1, Reward (c) Point, Circle1, Cost (d) Car, Circle1, Cost

Takeaways:

- Competitive control performance compared with other baseline methods
- Near Zero Constraint violation
- 20x Comparable sample efficiency

Future Work

- Beating other Safe RL Benchmarks Environments and Agents.
- Comparison to other Model-Based Safe RL algorithms

{oluwatosin_oseni, micah.corah}@mines.edu