Nightmare Dreamer: Dreaming About Unsafe States And Planning Ahead

Oluwatosin Oseni[*1], Shengjie Wang[*2], Jun Zhu[2], and Micah Corah[1]

[1]Colorado School of Mines [2]Tsinghua University

[*]Equal contribution. Corresponding author: `oluwatosin_oseni@mines.edu`

Abstract—**Reinforcement Learning (RL) has shown remarkable success in real-world applications, particularly in robotics control. However, RL adoption remains limited due to insufficient safety guarantees. We introduce *Nightmare Dreamer*, a model-based Safe RL algorithm that addresses safety concerns by leveraging a learned world model to predict potential safety violations and plan actions accordingly. *Nightmare Dreamer* achieves nearly zero safety violations while maximizing rewards. *Nightmare Dreamer* outperforms model-free baselines on Safety Gymnasium tasks using only image observations, achieving nearly a 20x improvement in efficiency.**

I. INTRODUCTION

Reinforcement Learning (RL) has shown impressive success across various domains, from surpassing human performance in games like Go [24], to champion-level drone racing [19], and advanced robotic tasks like catching dynamic objects with dexterous hands [20]. Recent model-based RL methods, such as DreamerV3 [15], further extend these capabilities—enabling robots to learn locomotion in hours and solving complex tasks like Minecraft diamond collection. Notably, RL has also demonstrated real-world impact in areas like plasma control for nuclear fusion [8] and autonomous navigation of stratospheric balloons [4]. Despite these successes, deployment of RL in real-world applications remains limited due to fundamental safety concerns. The exploratory nature of RL algorithms can lead agents to adopt dangerous or harmful behaviours during training, posing unacceptable risks in safety-critical environments [9].

This challenge is notable when deploying RL agents in environments where they interact with or operate around humans, such as autonomous vehicles, robotic assistants, or industrial control systems. Safe Reinforcement Learning (SafeRL) addresses these concerns by formulating the learning problem as a Constrained Markov Decision Process (CMDP) [2], where agents must maximize rewards while satisfying explicit safety constraints. Current approaches primarily rely on two methodologies: Lagrangian-based methods known as the primal-dual method [5] that use dual optimization to balance rewards and constraints with algorithms like PPO-Lag and TRPO-Lag [21], and primal methods that attempt to apply the cost constraints with clever design of the objective functions as well as updating the policy without much use of dual variables [26, 6]. State-of-art Model-free approaches like CPO [1] and PPO-Lagrangian [21], while theoretically sound, suffer from sample inefficiency and struggle to maintain safety guarantees throughout training, particularly in high-dimensional visual

environments. Conversely, model-based methods often fail to fully exploit the predictive capabilities of learned world models for proactive safety planning, limiting their effectiveness in preventing future constraint violations. To address these limitations, we introduce *Nightmare Dreamer*, a model-based SafeRL algorithm that leverages learned world models to predict potential safety violations and plan actions accordingly. Our key innovation lies in the integration of dual specialized actors—a control actor optimized for reward maximization and a safe actor focused on constraint satisfaction—with an online planning algorithm that switches between policies based on predicted future costs. Unlike existing approaches that treat safety as a reactive constraint, *Nightmare Dreamer* proactively "dreams" about unsafe future states and takes preventive action. Our main contributions are threefold:

1) A bi-actor architecture that separates reward optimization from safety constraint satisfaction, enabling more effective multi-objective learning;
2) A predictive safety planning mechanism that uses world model rollouts to anticipate constraint violations;
3) Demonstration that discriminator-based regularization can achieve stable training and superior performance compared to traditional behaviour cloning approaches.

Experimental evaluation on Safety Gymnasium benchmarks demonstrates that *Nightmare Dreamer* achieves nearly zero safety violations while maintaining competitive reward performance. Moreover, *Nightmare Dreamer* demonstrates strong sample efficiency, surpassing baseline performance with as little as 1/20 of the interaction steps.

II. RELATED WORK

Safety will play a vital role in potential everyday adoption of RL. SafeRL seeks to address this challenge by maximizing an objective function (reward) while simultaneously maintaining safety constraints (cost) below a predefined safety budget.

Constrained Policy Optimization (CPO) [1], a primal method, was the first state-of-the-art policy gradient algorithm to solve the CMDP problem. CPO performs two policy updates: first, updating the policy in the direction of objective optimization (similar to Trust Region Policy Optimization (TRPO) [22]), followed by projecting the policy back into the constraint set. While CPO may outperform primal-dual methods on some tasks and converge to the safety bound, it is computationally intensive due to being a second-order method involving inversion of high-dimensional Hessians [27].

Primal-dual methods such as PPO-Lagrangian and TRPO-Lagrangian [21] are the standard approach for CMDP problems. These methods apply Lagrangian duality in SafeRL and have achieved considerabe success. However, primal-dual methods remain challenging to apply due to parameter sensitivity in the learning rate of the Lagrangian multiplier.

Model-based Vision-only Safe RL Model-based methods have historically outperformed model-free approaches due to their superior sample efficiency. LAMBDA [3] added safe planning capabilities to DreamerV1 [13], but suffers from the same suboptimal performance due to its base algorithm, DreamerV1, compared to its later improvement, DreamerV2. Safe SLAC [16] achieves comparable performance to LAMBDA, yet does not fully exploit the world model's safety augmentation potential, bypassing imaginary rollouts that could enhance safe policy learning. Safe Dreamer [17], an adaptation of DreamerV3, combines Lagrangian methods with online planning to achieve strong performance. However, their planning is computationally intensive at inference time.

III. PRELIMINARY

Safe RL tries to solve the Constrained RL problem known as Constrained Markov Decision Processes (CMDP)[2], an extension of MPD in classical RL. The CMDP can be represented in a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{C}, \mathbb{P}, \mu, \gamma)$. $\mathcal{S}$ refers to the state space; $\mathcal{A}$ refers to the action space; $\mathcal{R}(r|s, a)$ refers to the reward obtained by the agent in state s after taking action a; $\mathcal{C}(c|s, a)$ refer to cost which will be subject to a constraint; $\mathbb{P}(s'|s, a)$ is the transition probability of going to state s' from state s taking action a while receiving $\mathcal{R}(r|s, a)$ and $\mathcal{C}(c|s, a)$, $\mu : S \rightarrow [0, 1]$ is the starting state distribution, and finally $\gamma \in [0, 1)$ is the reward discount factor. We define a parametrized policy $\pi_\theta(a|s)$ to maximize the cumulative discounted reward defined in the objective function below:

$$J^R(\pi_\theta) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma \mathcal{R}(r|s, a)\right] \tag{1}$$

In SafeRL, we aim to learn a policy that maximises the above objective while satisfying the constraint:

$$J^C(\pi_\theta) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma \mathcal{C}(c|s, a)\right] \leq b. \tag{2}$$

In other words, we maintain cumulative discounted cost below a threshold known as the safety budget b. We formulate the SafeRL problem as finding the optimal policy that satisfies:

$$\pi_* = \arg\max_{\pi_\theta} J^R(\pi_\theta) \quad \text{s.t} \quad J^C(\pi_\theta) \leq b. \tag{3}$$

IV. SAFE WORLD MODEL LEARNING

Nightmare Dreamer learns a world model based on the work of Hafner et al. [12], adopting the Recurrent State-Space Model. The world model components are parametrized by ϵ and learns the world dynamics. Following the partial-observability framework of RL problems, our model takes in the current camera image and the observation $o_t \sim$

$p(o_t|h_t, z_t)$. Then, without access to the real state, our model computes an internal state h_t, z_t to learn the world dynamics as well as to predict future observations, rewards, and costs:

$$\text{Recurrent Model: } h_t = f_\epsilon(h_{t-1}, z_{t-1}, a_t)$$
$$\text{Encoder Model: } z_t \sim q_\epsilon(z_t \mid h_t, o_t)$$
$$\text{Decoder Model: } \hat{o}_t \sim p_\epsilon(o_t \mid h_t, z_t)$$
$$\text{Transition Model: } \hat{z}_t \sim p_\epsilon(\hat{z}_t \mid h_t)$$
$$\text{Reward Model: } \hat{r}_t \sim p_\epsilon(\hat{r}_t \mid h_t, z_t)$$
$$\text{Cost Model: } \hat{c}_t \sim p_\epsilon(\hat{r}_t \mid h_t, z_t)$$
$$\text{Discount Model: } \hat{\gamma}_t \sim p_\epsilon(\gamma_t \mid h_t, z_t).$$

We define the loss function, parametrized by ϵ, below:

$$\mathcal{L}(\epsilon) \doteq \sum_{t=1}^{T} \underbrace{-\alpha_c \ln(p_\epsilon(c_t|h_t, z_t))}_{\text{cost log loss}} \underbrace{-\alpha_r \ln(p_\epsilon(r_t|h_t, z_t))}_{\text{reward log loss}}$$
$$\underbrace{-\ln(p_\epsilon(o_t|h_t, z_t))}_{\text{reconstruction loss}} \underbrace{-\ln(p_\epsilon(y_t|h_t, z_t))}_{\text{discount log loss}}. \tag{4}$$
$$\underbrace{+\text{KL}\left[q_\epsilon(z_t|h_t, o_t) \mid\mid sg(p_\epsilon(z_t|h_t))\right]}_{\text{representation loss}}.$$

V. BI-ACTOR CRITIC LEARNING

We take a multi-agent Actor-Critic Model-based RL approach to tackle the SafeRL problem. We train a Control and Safe Actor and a Reward and Cost Critic in pairs and aim to optimize their respective policy and value functions separately. To optimize both Policy functions, we utilize the learned world model and perform trajectory rollouts under each policy. The rollouts involve sampling a real observation stored from a previous environment interaction and, with the learned dynamics, imagining future states by applying the sampled action from the actors for H horizon steps. However, during environmental interaction, we switch between the Control and Safe actor to ensure safety.

A. Control Actor and Safe Actor

In our framework, we define two actor types. The Control actor, parametrized by ϕ, executes action a_c sampled by $\pi_\phi(a|s)$ that maximizes future expected reward. On the other hand, the safe actor, parametrized by ρ, aims to find an action a_s sampled by policy $\pi_\rho(a|s)$ that satisfies the cost constraint. Both actors are implemented as Multi-layer Perceptron (MLP) networks and employ the Exponential Linear Unit (ELU) activation function [7]. Action predictions are modelled using a truncated normal distribution.

B. Planning Ahead of Risks

Obstacle avoidance is a fundamental concept in robotics and planning. We pursue a similar approach (pseudocode in Alg. 1) by leveraging the learned world model. Starting from the current observation and state embedding, we perform rollouts under the Control Policy $\pi_\phi(a|s)$ in the latent space. This allows us to predict the potential cost violation if we continue to act under the Control policy. If the cost violation exceeds a threshold (safety budget b), we simply perform action selection by sampling from the safe policy $\pi_\rho(a|s)$ and sample from
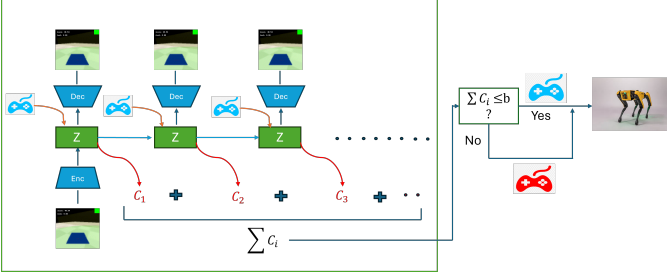
Fig. 1: Action selection during environment interaction: The blue gamepad signifies the action from the Controller, while the red gamepad refers to an action from from the Safe Actor.

the Control policy $\pi_\phi(a|s)$ otherwise. We provide the action selection equation below:

$$a \sim (1 - \mathbf{1}(C_{\text{sum}} \leq b_s)) \cdot \pi_\rho(a|s) + \mathbf{1}(C_{\text{sum}} \leq b_s) \cdot \pi_\phi(a|s)$$

$$C_{\text{sum}} = C_t(h_t, z_t, o_t) + \sum_{t+1}^{t+1+H} C_t(h_t, z_t) \quad (5)$$

where C at time step t is predicted from the agent's observation (Posterior), and the consequent Costs are predicted using rollout from the learned model (Prior) without access to the observation. Figure 1 illustrates this action selection process.

C. Control & Safe Actors Learning

In our approach, we train two policies, a Control policy and a Safe policy. To train the Control policy, which actors aim to find a set of parameters that maximize the objective (Reward). We achieve this by directly maximizing the target Value function (gradient ascent) wrt the Control Policy, which is given by:

$$V_t^\lambda = r_t + \gamma_t \begin{cases} (1-\lambda)v_\xi(\hat{z}_{t+1}) + \lambda V_{t+1}^\lambda & \text{if } t < H, \\ v_\xi(\hat{z}_H) & \text{if } t = H. \end{cases} \quad (6)$$

Where the value function v_ξ aims to maximise the sum of discounted rewards. Finally, we obtain the following loss function for the Control policy parametrized by ϕ [14]:

$$\mathcal{L}(\phi) \doteq \sum_{t=1}^{H-1} (\underbrace{-V_t^\lambda}_{\text{target value}} - \underbrace{\eta H[\pi_\rho(a_t|s_t)]}_{\text{entropy regularizer}}). \quad (7)$$

The entropy regulariser term ensures policy exploration. To perform policy improvement, we perform rollouts under the Control policy by sampling states from the buffer and performing H horizon steps from the sampled initial states. The rollout (Imagination) involves starting from some initial state and, with the learned model (Transition Model), computing all recurring states given an Action.

1) Safe Actor Learning and Lagrangian Formulation:
The Primal-dual method [5], also known as the Lagrangian method, is the most common approach to the Multi-objective CMDP problem [21]. The Lagrangian formulation balances our task objective with cost constraints and is given below:

$$\max_{\pi_\phi} \min_{\lambda_p \geq 0} \quad J_{\text{task}}(\pi_\phi) - \lambda_p(J_{\text{constraint}}(\pi_\rho) - b). \quad (8)$$

The Safe actor, parametrized by ρ, aims to find a policy $\pi_\rho(a|s)$ that solves the multi-objective optimization problem (8) and is given by:

$$\mathcal{L}(\rho) \doteq \sum_{t=1}^{H-1} (\underbrace{\lambda_p C_t^\lambda}_{\text{target cost value}} \underbrace{-D(a_t, s_t)}_{\text{behaviour cloning}} \underbrace{-\eta H[\pi_\phi(a_t|s_t)]}_{\text{entropy regularizer}}). \quad (9)$$

The first term aims to solve the cost constraint in (3) by directly minimizing the target value cost function wrt to the Safe Policy and is weighed by λ_p, provided in (10).

We achieve maximising the control objective in 3 by **Behaviour Cloning** the Control Policy $\pi_\phi(a|s)$ rather than direct reward maximization. This Multi-Objective loss function ensures the satisfaction of safety constraints while maintaining goal-directed behaviour.

To imitate the Control policy, we train a Discriminator network that learns to predict if an action is sampled from a Control or a Safe policy, given a current state. The discriminator outputs 0 for safe actions and 1 for control actions, and we maximize $-D(s, a)$ wrt the Safe Policy to encourage control-like behaviour.

Algorithm 1: 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 $C_{\text{sum}} \leftarrow C_t(h_t, z_t, o_t)$;
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;

---

2) Lagrangian Method and Update:
The learnable Lagrangian value λ_p in Equ 9 is updated based with:

$$\lambda_{p_{k+1}} = \text{Clip}(\lambda_{p_k} - \alpha(C_k - \text{budget}), \lambda_{p_{\min}}, \lambda_{p_{\max}}) \quad (10)$$

where the Clip function ensures the Lagrangian value does not explode or become infinitesimal, and the online mean cost C_k is defined as the moving average cost over the past $l =$

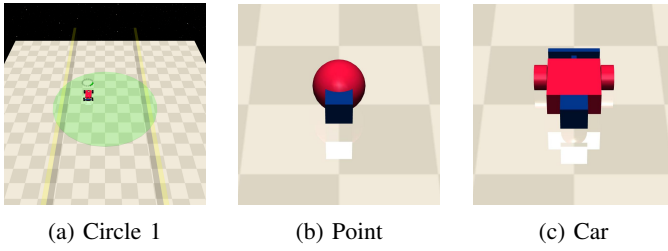(a) Circle 1   (b) Point   (c) Car

Fig. 2: Safety Circle Agents

50 time steps. This online mean provides an estimate of the current performance of the agent that allows us to update the Lagrangian multiplier λ_p appropriately.

3) Behaviour Imitation via Action Discrimination: Discriminators are often used with adversarial networks and GANs [11] to perform image generation. We use a Discriminator term to ensure our safe policy selects actions that maximise the reward by mimicking the control policy. Our experiments show that direct Discriminator optimization provides superior regularization and generalization compared to standard behaviour cloning methods like KL loss or log probability. The discriminator estimates the log likelihood of predicting if an action is sampled from the Control policy or a safe policy given a current state s_t:

$$\mathcal{L}(D(a_t, s_t)) \doteq \mathbb{E}_{a \sim \pi_\phi(a|s)} \log(D(a_t, s_t)). \quad (11)$$

D. Critics Learning

The reward and cost critics are trained to predict future discounted reward and cost, respectively, given the states from the imagination rollouts, similar to the actors. The Critics are MLP networks and use ELU activation functions that output a distribution of the critic estimation. We found that this helps address the sparse nature of the cost from the environment. Leveraging the learned model allows us to predict the sparse cost in the environment. We compute the future discounted sum of both Cost and reward using the generalised λ target values [23, 25]. using the same setup for DreamerV2 for both the Cost and Reward Value functions. The loss functions are thus formulated to maximize the log-likelihood of predicting the value function from the λ target values.

VI. EXPERIMENTAL SETUP AND RESULTS

Nightmare Dreamer was trained on Safety Gymnasium, a Safe Policy Optimization Benchmark (SafePO) for Safe RL [18]. This benchmark is an extension of the now-discontinued Safety Gym previously maintained by OpenAI [10].

A. Task Description

Safety Gymnasium provides several environments; we focus on the circle environment. There are 3 circle tasks with progressively difficult constraints (Fig. 2). All involve the agent moving around in a circle, but the agent incurs a cost of 1 for every step while outside a constraint boundary. There are also several agents with different control complexities (Fig. 2).
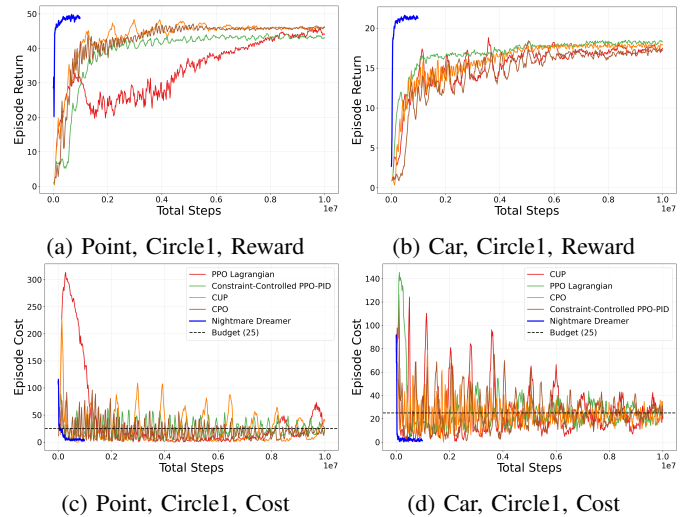

(a) Point, Circle1, Reward   (b) Car, Circle1, Reward

(c) Point, Circle1, Cost   (d) Car, Circle1, Cost

Fig. 3: Safety Circle 1 Reward and Cost Performance Comparison with Benchmark algorithms

B. Results

Figure 3 shows experimental results of our algorithm on SafePointCircle and SafeCarCircle compared to SOTA SafeRL algorithms from the SafePo benchmark. We observe promising results, with Nightmare converging to optimal reward and cost faster than benchmark algorithms. The dashed line indicates the safety budget that cost values must remain below. Nightmare stays below this budget while achieving near-zero cost. Due to Nightmare's computational requirements, we compare 1e6 environmental interactions against 1e7 for benchmark methods. We compare our approach to SOTA model-free algorithms that are more computationally efficient than our approach but significantly less sample efficient, similar to the Safe-Dreamer [17] evaluation.

VII. CONCLUSION

We introduced *Nightmare Dreamer*, a model-based safe RL algorithm that achieves zero constraint violations while maximizing rewards from visual inputs. Our method trains two specialized actors—control and safe—using a learned world model, with a planning mechanism that switches between policies based on predicted future costs. Our key innovations is discriminator-based policy regularization approach. Experiments on Safety Gymnasium's Circle task demonstrate faster convergence to safe policies compared to model-free baselines. While currently validated on Circle tasks, the framework provides a foundation for extending model-based safe RL to more complex environments. In the future, we hope to adapt *Nightmare Dreamer* to other tasks and to perform tests with real-world robots and constraints and with comparison to other model-based Safe RL methods.

REFERENCES

[1] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization, 2017. URL https://arxiv.org/abs/1705.10528.

[2] Eitan Altman. Constrained markov decision processes. 1999. URL https://api.semanticscholar.org/CorpusID:14906227.

[3] Yarden As, Ilnura Usmanova, Sebastian Curi, and Andreas Krause. Constrained policy optimization via bayesian world models, 2022. URL https://arxiv.org/abs/2201.09802.

[4] Marc G Bellemare, Salvatore Candido, Pablo Samuel Castro, Jun Gong, Marlos C Machado, Subhodeep Moitra, Sameera S Ponda, and Ziyu Wang. Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature*, 588(7836):77–82, 2020.

[5] D P Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334, 1997. doi: 10.1057/palgrave.jors.2600425. URL https://doi.org/10.1057/palgrave.jors.2600425.

[6] Shicong Cen, Chen Cheng, Yuxin Chen, Yuting Wei, and Yuejie Chi. Fast global convergence of natural policy gradient methods with entropy regularization, 2021. URL https://arxiv.org/abs/2007.06558.

[7] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus), 2016.

[8] Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego Casas, Craig Donner, Leslie Fritz, Cristian Galperti, Andrea Huber, James Keeling, Maria Tsimpoukelli, Jackie Kay, Antoine Merle, Jean-Marc Moret, and Martin Riedmiller. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602:414–419, 02 2022. doi: 10.1038/s41586-021-04301-9.

[9] Shuo Feng, Haowei Sun, Xintao Yan, Haojie Zhu, Zhengxia Zou, Shengyin Shen, and Henry X Liu. Dense reinforcement learning for safety validation of autonomous vehicles. *Nature*, 615(7953):620–627, 2023.

[10] Scott Fujimoto, Edoardo Conti, Mohammad Ghavamzadeh, and Joelle Pineau. Benchmarking batch deep reinforcement learning algorithms, 2019. URL https://arxiv.org/abs/1910.01708.

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[12] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels, 2019. URL https://arxiv.org/abs/1811.04551.

[13] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination, 2020. URL https://arxiv.org/abs/1912.01603.

[14] Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=0oabwyZbOu.

[15] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models, 2023.

[16] Yannick Hogewind, Thiago D. Simao, Tal Kachman, and Nils Jansen. Safe reinforcement learning from pixels using a stochastic latent representation, 2022. URL https://arxiv.org/abs/2210.01801.

[17] Weidong Huang, Jiaming Ji, Chunhe Xia, Borong Zhang, and Yaodong Yang. Safedreamer: Safe reinforcement learning with world models, 2024. URL https://arxiv.org/abs/2307.07176.

[18] Jiaming Ji, Borong Zhang, Jiayi Zhou, Xuehai Pan, Weidong Huang, Ruiyang Sun, Yiran Geng, Yifan Zhong, Juntao Dai, and Yaodong Yang. Safety-gymnasium: A unified safe reinforcement learning benchmark. *arXiv preprint arXiv:2310.12567*, 2023.

[19] Elia Kaufmann, Leonard Bauersfeld, Antonio Loquercio, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. Champion-level drone racing using deep reinforcement learning. *Nature*, 620(7976):982–987, 2023.

[20] Fengbo Lan, Shengjie Wang, Yunzhe Zhang, Haotian Xu, Oluwatosin Oseni, Ziye Zhang, Yang Gao, and Tao Zhang. Dexcatch: Learning to catch arbitrary objects with dexterous hands, 2024. URL https://arxiv.org/abs/2310.08809.

[21] Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708*, 7(1):2, 2019.

[22] John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization, 2017. URL https://arxiv.org/abs/1502.05477.

[23] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation, 2018.

[24] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[25] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[26] Tengyu Xu, Yingbin Liang, and Guanghui Lan. Crpo: A new approach for safe reinforcement learning with convergence guarantee, 2021. URL https://arxiv.org/abs/2011.05869.

[27] Linrui Zhang, Li Shen, Long Yang, Shixiang Chen, Bo Yuan, Xueqian Wang, and Dacheng Tao. Penalized proximal policy optimization for safe reinforcement learning, 2022. URL https://arxiv.org/abs/2205.11814.