Safety-Informed Mutations for Evolutionary Deep Reinforcement Learning

Enrico Marchesini, Christopher Amato
Northeastern University
Boston, Massachusetts, USA
{e.marchesini,c.amato}@northeastern.edu

ABSTRACT

Evolutionary Algorithms have been combined with Deep Reinforcement Learning (DRL) to address the limitations of the two approaches while leveraging their benefits. In this paper, we discuss objective-informed mutations to bias the evolutionary population toward exploring the desired objective. We focus on Safe DRL domains to show how these mutations exploit visited unsafe states to search for safer actions. Empirical evidence on a 12 degrees of freedom locomotion benchmark and a practical navigation task, confirm that we improve the safety of the policy while maintaining comparable return with the original DRL algorithm.

CCS CONCEPTS

• **Computing methodologies → Reinforcement learning**; *Evolutionary robotics.*

KEYWORDS

Reinforcement Learning, Evolutionary Algorithms, Deep, Mutations, Robotics

ACM Reference Format:
Enrico Marchesini, Christopher Amato. 2022. Safety-Informed Mutations for Evolutionary Deep Reinforcement Learning. In *Genetic and Evolutionary Computation Conference Companion (GECCO '22 Companion), July 9–13, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3520304.3533980

1 INTRODUCTION

In the physical world, evolution and learning cooperate to assimilate the benefits of both solutions [36], while addressing their limitations. A recent research field takes inspiration from this natural phenomenon, proposing the combination of Deep Reinforcement Learning (DRL) [39] and Evolutionary Algorithms (EAs) [11].

DRL is well-known for solving complex decision-making problems where agents interact with an environment in a trial and errors fashion to maximize a long-term objective called *return*. In particular, DRL achieved astonishing progress in a wide variety of domains, ranging from robotics [12, 31] to games [28, 35]. However, despite the successes, this trial and error learning paradigm suffers from several issues in practical applications, where exploration and safety are two key aspects. For example, DRL suffers from premature convergence to local optima, which is mainly caused by the lack of diverse exploration when operating in high-dimensional spaces [13]. Conversely, the redundancy of population-based EAs has the advantage of enabling diverse exploration, leading to a more stable convergence. For this reason, EAs have been employed as a gradient-free optimization alternative to DRL. Genetic Algorithms (GAs) [29], in particular, achieved competitive results compared to gradient-based DRL [38] without involving computational demanding gradient computations. However, the lack of gradient information causes poor generalization skills, and GAs are thus significantly less sample efficient than gradient-based DRL.

Hence, an emergent research field proposed the combination of gradient-free population-based approaches and gradient-based solutions [4, 6, 15, 16, 21, 24, 32]. While the specific combination strategy may vary among these recent algorithms, the general idea is to have an evolutionary population that interacts with independent copies of the environment, producing diverse trajectories. The combined frameworks that focus on the evolutionary component [4, 15, 16], thus select the best individuals using a fitness metric, and then generate a new population by applying crossover and mutation operators. In this context, a DRL agent that is trained in parallel is periodically injected into the population to transfer its *gradient-based knowledge*. In contrast, the approaches that focus on the gradient-based component [21, 24] use the population to favor exploration and diversity, and the evolutionary information is transferred back to the DRL agent using soft updates [34].

Due to our interest in practical applications, in this paper we focus on the latter frameworks, to highlight the importance of the safety aspect for the gradient-based agent. In more detail, we consider problems where unsafe behaviors are specified with an auxiliary cost signal that is separate from the task objective [33]. Hence, our goal is to bias policies toward safety without constrained optimization, which is typical of recent DRL algorithms that are used due to the intuitive way of constraints (on the cost) to encode safety criteria [1, 20, 37]. To this end, we extend naive gradient-based mutations [17] to bias current policies to explore safer behaviors. Our safe mutations exploit the visited unsafe states (according to the cost) to approximate the per-weight sensitivity of the actions over such undesired situations. Then, such sensitivity is used to compute safety-informed perturbations that locally bias the agent policy to explore different actions in the proximity of the unsafe states. In the context of combined approaches, we periodically generate an evolutionary population from the DRL policy using the safe mutations. Therefore, the individuals are evaluated independently over a set of trials to select the individual with a comparable return to

GECCO '22 Companion, July 9–13, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9268-6/22/07…$15.00
https://doi.org/10.1145/3520304.3533980

the DRL agent and a lower cost. The parameters of this individual then replace the ones of the DRL agent [21].

We show the performance of our safe mutations in two scenarios: (i) a complex locomotion task of the recent Safety Gym benchmarks [33], (ii) a practical navigation scenario based on a Turtlebot3 robot [25, 27].[1] We compare over the baseline PPO [10] and constrained DRL (Lagrangian PPO [37], Constrained Policy Optimization (CPO) [1], Interior Point Optimization (IPO) [20]) as the latter is the most closely related work that employs cost functions to characterize safety. Our empirical evaluation analyzes the return and cost trade-off, confirming a successful exchange of information between the evolutionary part and the gradient-based agent. In particular, we achieve comparable or superior performance across the considered tasks.

2 PRELIMINARIES

In this section, we briefly discuss the main ideas of previous constrained DRL approaches and naive gradient-based mutations.

2.1 Constrained Deep Reinforcement Learning

A CMDP [2] is a Markov Decision Process with an additional set of constraints C based on $C_i : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ ($i \in \{1, \dots, k\}$) cost functions and $\mathbf{h} \in \mathbb{R}^k$ thresholds for the constraints. The C_i*-return* is defined as $J_{C_i}(\pi) := \mathbb{E}_{\tau\sim\pi}[\sum_{t=0}^{\infty} \gamma^t C_i(s_t, a_t)]$, where $\gamma \in (0, 1)$ is the discount, $\tau = (s_0, a_0, \dots)$ is a trajectory, $\pi = \{\pi(a|s) : s \in \mathcal{S}, a \in \mathcal{A}\}$ is a policy in state $\mathcal{S}$ and action $\mathcal{A}$ spaces. Constraint-satisfying policies Π_C, and optimal policies π^* are thus defined as:

$$\Pi_C := \{\pi \in \Pi : J_{C_i}(\pi) \leq h_i,\ \forall i\}, \quad \pi^* = \arg\max_{\pi \in \Pi_C} J(\pi)$$

where $J(\pi) := \mathbb{E}_{\tau\sim\pi}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)]$ is the expected discounted return that we aim at maximizing in a standard MDP; Π are the stationary policies, and $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function. Without loss of generality, we consider the case of one cost function (as in recent constrained DRL literature [20, 33, 37]).

2.2 Evolutionary Algorithms

Evolutionary approaches typically evolve a population of $p \in \mathbb{N}$ individuals (genomes), represented by parameters (weights) θ_i ($i \in \{1, \dots, p\}$). The individuals are evaluated to produce a fitness score used by the selection operator to choose the best genome. However, simple Gaussian-based mutations $\mathcal{N}$ can lead to disruptive changes [17] that can be naively address uses zero-mean and low standard deviation [26]. Otherwise, if we define a genome as a Deep Neural Network (DNN) parametrized by θ that represents a function $f_\theta : \mathcal{D}_\mathbf{x} \rightarrow \mathcal{D}_\mathbf{y}$ (input $\mathbf{x} \in \mathcal{D}_\mathbf{x} \subseteq \mathbb{R}^n$ and output $\mathbf{y} \in \mathcal{D}_\mathbf{y} \subseteq \mathbb{R}^m$, with input/output size n, m), and a vector of states $\mathbf{s}$, we can express the average divergence of the outputs $\mathbf{y}$ as a result of a perturbation δ as:

$$d(f_\theta, \delta) = \frac{\|f_\theta(\mathbf{s}) - f_{\theta+\delta}(\mathbf{s})\|_2}{|\mathbf{s}|} \tag{1}$$

where $f_\theta(\mathbf{s})$ are the forward propagations of the states through the DNN. Otherwise, a more flexible way to avoid disruptive mutations employs a differentiable DNN to approximate d with gradient information [17]. In detail, it considers the following first-order Taylor expansion to model an output $y_j \in \mathbf{y}$ ($j \in \{0, \dots, |\mathbf{y}|\}$) as a function of perturbations δ over the states $\mathbf{s}$:

$$y_j(f_\theta, \delta) = f_\theta(\mathbf{s})_j + \delta \nabla_\theta f_\theta(\mathbf{s})_j \tag{2}$$

In the following section, we discuss how to specialize naive gradient-based mutations of Equation 2 to explore safer behaviors.

3 METHODS

Our mutation operator extends the SUPE-RL framework [21], and the general flow of the safe mutation method is presented in Algorithm 1. We first augment the agent training with a *cost-buffer* B_c which stores all the visited states deemed unsafe according to the cost. Hence, the training process proceeds as follows:

- Periodically, we sample a batch b from B_c to compute the per-weight safety-informed sensitivity λ of the agent outputs over its weights θ_a. This is used to generate a population of n individuals $\mathcal{P} = \{p_1, \dots, p_n\} \cup \{p_a\}$ with weights $\theta_\mathcal{P}$ (p_a is a copy of the agent), voted to explore for safer behaviors.
- $\mathcal{P}$ is evaluated in a set of epochs to collect the individuals average reward R_p and cost C_p that define the fitness score $\mathcal{P}$-fitness $= (R_p, C_p)\ \forall p \in \mathcal{P}$
- Such fitness is used to select the best individual, i.e., the one with greater or equal reward with respect to the copy of the DRL agent, and lower cost. By choosing an appropriate number of evaluation epochs for the population, we assume the best individual to be safer than p_a as it has higher (or equal) rewards and lower (or equal) cost.

3.1 Safety-Informed Mutations

Gradient information can be used to design mutations that avoid detrimental behaviors, normalizing the perturbation by a per-weight measure of sensitivity [17].

We leverage the cost function to avoid disruptive changes to the policy while biasing it to safety. In detail, we consider a baseline Gaussian noise $\mathcal{G} \sim \mathcal{N}(0, mut_v)$ for the perturbations and normalize it with our safety-informed sensitivity λ. The resultant mutations δ_{SM} are applied to the agent weights θ_a to generate $\mathcal{P}$. One way to compute λ considers the gradient of the actual divergence (Equation 1) [17]:

$$\begin{aligned} \nabla_{\theta_a} d(f_{\theta_a}, \mathcal{G}) &\approx \nabla_{\theta_a} d(f_{\theta_a}, 0) + H_{\theta_a}(d(f_{\theta_a}, 0))\mathcal{G} \\ \lambda_{f_{\theta_a}} &= abs(\nabla_{\theta_a} d(f_{\theta_a}, \mathcal{G})) \end{aligned} \tag{3}$$

however, the Hessian H_{θ_a} of divergence with respect to θ_a requires second-order approximations, and therefore it is computationally demanding. In contrast, we rely on the per-weight magnitude of the gradient of the outputs $\mathbf{y} = f_{\theta_a}(b)$, where b is a batch of unsafe states randomly sampled from B_c, to estimate the sensitivity λ to that weight with a first-order approximation:

$$\begin{aligned} \lambda_{f_{\theta_a}} &= \sum_{\mathbf{y}} \left(\frac{\sum_\mathbf{s} abs(\nabla_{\theta_a} f_{\theta_a}(\mathbf{s}))}{|\mathbf{s}|} \right) \frac{1}{|\mathbf{y}|} \\ \delta_{SM}(f_{\theta_a}) &= \frac{\mathcal{G}}{\lambda_{f_{\theta_a}}} \end{aligned} \tag{4}$$

[1] www.turtlebot.com

where each unsafe experience equally contributes to λ to reduce the overall changes to the policy. In practice, we note that using a threshold λ to limit the mutation rescaling (i.e., the hyper-parameter λ_{max}) leads to better performance. To summarize, our idea is to design safety-oriented gradient information using visited unsafe states to bias the policy to explore different actions in the proximity of such situations.

Algorithm 1 Safety-Informed Mutations

```
Input:
    • a DRL agent with weights $\theta_a$
    • a cost-buffer $B_c$ for the unsafe samples
    • scale $mut_v$ for the Gaussian $\mathcal{G}$ and threshold $\lambda_{max}$
$b \leftarrow$ Sample an unsafe batch from $B_c$
Compute $\mathcal{G} \leftarrow \mathcal{N}(0, mut_v)\ \forall$ weight $\in \theta_p,\ \forall p \in \mathcal{P}$
$\lambda \leftarrow$ Equation 4 using $b$, replacing values $\leq \lambda_{max}$ with $\lambda_{max}$
$\theta_p \leftarrow \theta_p + \frac{\mathcal{G}}{\lambda},\ \forall p \in \mathcal{P}$
```

4 EXPERIMENTS

The following data are collected on an RTX 2070, using the hyper-parameters of the original authors' implementations for the baselines, and ten independent runs with different seeds. In particular, we report the Pareto frontier of average cost (x-axis) versus average reward (y-axis) at convergence. We compare the baseline PPO, and three constrained DRL algorithms: L-PPO [37], IPO [20], CPO [1] as they are the most closely related work to the idea of addressing safety using a cost function.

4.1 Experiments on SafetyGym

We initially evaluate the safe mutation operator on a complex locomotion task of the SafetyGym suite, namely DoggoGoal1 [33]. In this problem, a simulated Doggo robot with 12 degrees of freedom has to learn locomotion behaviors to reach random targets in the environment while avoiding obstacles that trigger a positive cost signal upon collision.

Figure 1 shows the results of a SUPE-RL implementation of PPO with our safe mutation operator, namely SM-PPO. In this environment, we note that L-PPO maintains the imposed cost limit but fails at learning the locomotion of the complex 12-joint robot. In contrast, the baseline PPO does not consider the cost signal, achieving significant performance in terms of reward, but failing at reducing the cost. Finally, our approach achieves comparable rewards over CPO and IPO but significantly reduces the cost value.

4.2 Experiments on Robotic Navigation

Given our interest in practical applications and the importance of safety in these contexts, we discuss and report the results of a realistic robotic mapless navigation task. The setup is straightforward and widely adopted in the field of DRL for mapless navigation [18, 23].

Figure 2 on the left shows the simulation environment, where a TurtleBot3 has to learn how to navigate in an indoor environment with obstacles to reach random targets, using only local observations (e.g., laser scans). The reward has a dense component during the trajectory: $(d_{t-1} - d_t)$, where d_{t-1}, d_t indicates the Euclidean distance between the robot and the goal at two consecutive time steps, and the cost function is simply triggered upon collision with an obstacle Similar to the Doggo scenario, Figure 2 on the right reports the Pareto frontier. Crucially, it significantly improves (up and to the left) with SM-PPO further confirming the benefits of the safe mutation operator. In the same fashion, the baseline PPO does not integrate the cost in the optimization and presents both a higher reward and cost (i.e., unsafer behaviors).

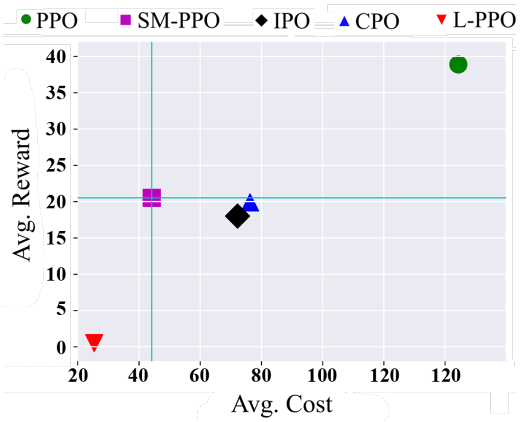

Figure 1: Pareto frontier at convergence of PPO, SM-PPO, IPO, CPO, L-PPO in DoggoGoal1.

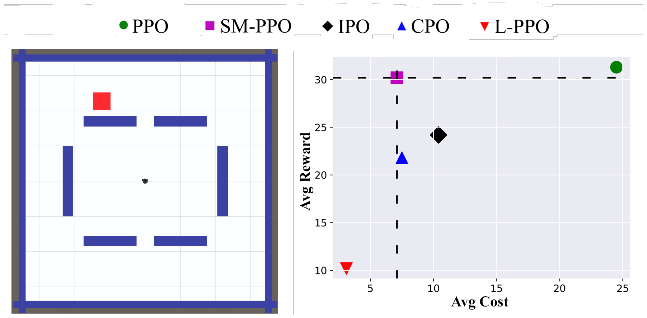

Figure 2: Comparison in our robotic mapless navigation task. (Left) Overview of the environment, where obstacles and walls are depicted in blue and the Turtlebot3 goal in red. (Right) Pareto frontier of average reward versus average cost at convergence of PPO, SM-PPO, IPO, CPO, L-PPO.

5 RELATED WORK

Safety critics [3, 40] rely on estimating the probability of incurring into unsafe states, given a state-action pair. However, such approaches could return misleading information for policy improvement, and each step has to compute different samples (e.g., the action, the probability of failure), which can hinder their application to the physical hardware that requires high-frequency control. Previous work also consider Formal Verification [8, 19] approaches to inject safety specifications [9, 22], but the focus of this work is to show that the only Safety-Informed Mutation operator is sufficient to bias the policies toward safer regions.

We compared our framework with constrained DRL as it is more related to our approach. In more detail, CPO [1] has near-constrained satisfaction guarantees, but the Taylor approximations lead to inverting a Fisher matrix, possibly resulting in infeasible updates and demanding recovery steps. Lyapunov-based algorithms [5], combine a projection step with action-layer interventions. However, the cardinality of Lyapunov constraints equals the number of states, resulting in a non-negligible implementation cost. Lagrangian methods [33, 37] reduce the complexity of prior approaches, by transforming the following constrained objective: $\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{s.t.} \quad g(\mathbf{x}) = 0$ into an unconstrained one using the Lagrange multiplier (or penalty) l_λ that form the Lagrangian:$\mathcal{L}(\mathbf{x}, \lambda) = f(\mathbf{x}) + l_\lambda g(\mathbf{x})$. In particular, these Lagrangian-based DRL algorithms represent a well-known constrained baseline due to their simplicity and good cost-limit satisfaction [33, 37]. Similarly, IPO [20] reduces the constrained problem into an unconstrained one by augmenting the objective with logarithmic barrier functions, which provide sub-optimal solutions.

However, constraints naturally limit exploration, causing getting stuck in local optima or failing to learn properly [7, 14, 30]. In contrast, we leverage EAs to design the safe mutation operator as prior combinations of DRL and EAs show a beneficial transfer of information between the two approaches [15, 16, 21]. These methods, however, use the evolutionary component only for improving the return and can not be trivially extended to address the safety component.

6 DISCUSSION

In this paper, we propose a safe mutation operator to enhance previous combinations of EAs and DRL. In detail, our operator proposes the design of an informed mutation strategy that preserves the policy behaviors while biasing exploration towards the desired objective (e.g., safety).

Our results in a Safety Gym benchmark and a practical robotic navigation task, confirm that we successfully address the trade-off between return and cost, achieving comparable returns to unconstrained algorithms and comparable cost values to constrained DRL.

The proposed objective-informed operator has several potential impacts on society as it addresses safety, a crucial aspect of practical DRL applications. In particular, we show that it is possible to augment exploration toward the desired objective and successfully transfer beneficial information into a DRL agent, which opens interesting opportunities for future research on multi-objective optimization.

REFERENCES

[1] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. 2017. Constrained Policy Optimization. In *International Conference on Machine Learning (ICML)*.

[2] Eitan Altman. 1999. Constrained Markov Decision Processes. In *CRC Press*.

[3] Homanga Bharadhwaj, Aviral Kumar, Nicholas Rhinehart, Sergey Levine, Florian Shkurti, and Animesh Garg. 2021. Conservative Safety Critics for Exploration. In *International Conference on Learning Representations (ICLR)*.

[4] Pietro Lio' Bodnar, Ben Day. 2020. Proximal Distilled Evolutionary Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*.

[5] Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. 2018. A Lyapunov-based Approach to Safe Reinforcement Learning. In *Conference on Neural Information Processing Systems (NeurIPS)*.

[6] Cédric Colas, Olivier Sigaud, and Pierre-Yves Oudeyer. 2018. GEP-PG: Decoupling Exploration and Exploitation in Deep Reinforcement Learning Algorithms. In *International Conference on Machine Learning (ICML)*.

[7] Edoardo Conti, Vashisht Madhavan, Felipe Petroski Such, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. 2018. Improving Exploration in Evolution Strategies for Deep Reinforcement Learning via a Population of Novelty-Seeking Agents. In *Conference on Neural Information Processing Systems (NeurIPS)*.

[8] Davide Corsi, Enrico Marchesini, and Alessandro Farinelli. 2021. Formal verification of neural networks for safety-critical tasks in deep reinforcement learning. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence (Proceedings of Machine Learning Research, Vol. 161)*, Cassio de Campos and Marloes H. Maathuis (Eds.). PMLR, 333–343.

[9] Davide Corsi, Enrico Marchesini, Alessandro Farinelli, and Paolo Fiorini. 2020. Formal Verification for Safe Deep Reinforcement Learning in Trajectory Generation. In *2020 Fourth IEEE International Conference on Robotic Computing (IRC)*. 352–359. https://doi.org/10.1109/IRC.2020.00062

[10] Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. 2017. OpenAI Baselines. https://github.com/openai/baselines.

[11] David Fogel. 2006. Evolutionary computation - toward a new philosophy of machine intelligence (3. ed.).

[12] S. Gu, E. Holly, T. Lillicrap, and S. Levine. 2017. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *IEEE International Conference on Robotics and Automation (ICRA)*.

[13] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. 2018. Deep Reinforcement Learning that Matters. In *AAAI Conference on Artificial Intelligence*.

[14] Zhang-Wei Hong, Tzu-Yun Shann, Shih-Yang Su, Yi-Hsiang Chang, and Chun-Yi Lee. 2018. Diversity-Driven Exploration Strategy for Deep Reinforcement Learning. In *Conference on Neural Information Processing Systems (NeurIPS)*.

[15] Shauharda Khadka, Somdeb Majumdar, Tarek Nassar, Zach Dwiel, Evren Tumer, Santiago Miret, Yinyin Liu, and Kagan Tumer. 2019. Collaborative Evolutionary Reinforcement Learning. In *International Conference on Machine Learning (ICML)*.

[16] Shauharda Khadka and Kagan Tumer. 2018. Evolutionary Reinforcement Learning. In *Conference on Neural Information Processing Systems (NeurIPS)*.

[17] Joel Lehman, Jay Chen, Jeff Clune, and Kenneth O. Stanley. 2018. Safe Mutations for Deep and Recurrent Neural Networks through Output Gradients. In *GECCO*.

[18] Ming Liu Lei Tai, Giuseppe Paolo. 2017. Virtual-to-real Deep Reinforcement Learning: Continuous Control of Mobile Robots for Mapless Navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

[19] Changliu Liu, Tomer Arnon, Christopher Lazarus, Christopher Strong, Clark Barrett, and Mykel J. Kochenderfer. 2021. Algorithms for Verifying Deep Neural Networks. *Foundations and Trends® in Optimization* 4, 3-4 (2021), 244–404. https://doi.org/10.1561/2400000035

[20] Yongshuai Liu, Jiaxin Ding, and Xin Liu. 2020. IPO: Interior-point Policy Optimization under Constraints. In *AAAI*.

[21] Enrico Marchesini, Davide Corsi, and Alessandro Farinelli. 2021. Genetic Soft Updates for Policy Evolution in Deep Reinforcement Learning. In *International Conference on Learning Representations (ICLR)*.

[22] E. Marchesini, D. Corsi, and A. Farinelli. 2022. Exploring Safer Behaviors for Deep Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*.

[23] E. Marchesini and A. Farinelli. 2020. Discrete Deep Reinforcement Learning for Mapless Navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*.

[24] E. Marchesini and A. Farinelli. 2020. Genetic Deep Reinforcement Learning for Mapless Navigation. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

[25] Enrico Marchesini and Alessandro Farinelli. 2021. Centralizing State-Values in Dueling Networks for Multi-Robot Reinforcement Learning Mapless Navigation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 4583–4588.

[26] Josè Antonio Martin H. and Javier de Lope. 2009. Learning Autonomous Helicopter Flight with Evolutionary Reinforcement Learning. In *Computer Aided Systems Theory*.

[27] Luca Marzari, Davide Corsi, Enrico Marchesini, and Alessandro Farinelli. 2021. Curriculum Learning for Safe Mapless Navigation. *arXiv* (2021).

[28] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. In *Workshop of Conference on Neural Information Processing Systems (NeurIPS)*.

[29] D. Montana and L. Davis. 1989. Training Feedforward Neural Networks Using Genetic Algorithms. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

[30] Olle Nilsson and Antoine Cully. 2021. Policy Gradient Assisted MAP-Elites. In *GECCO '21*.

[31] OpenAI, Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei Zhang. 2019. Solving Rubik's Cube with a Robot Hand. In *arXiv*.

[32] Aloïs Pourchot and Olivier Sigaud. 2019. CEM-RL: Combining evolutionary and gradient-based methods for policy search. In *International Conference on Learning Representations (ICLR)*.

[33] Alex Ray, Joshua Achiam, and Dario Amodei. 2019. Benchmarking Safe Exploration in Deep Reinforcement Learning. In *OpenAI*.

[34] Olivier Sigaud. 2022. Combining Evolution and Deep Reinforcement Learning for Policy Search: a Survey. In *arXiv*.

[35] David Silver, Aja Huang, Chris Maddison, and et al. 2018. Mastering the game of Go with deep neural networks and tree search.. In *Nature*.

[36] George Gaylord Simpson. 1953. The Baldwin Effect. In *Evolution*.

[37] Adam Stooke, Joshua Achiam, and Pieter Abbeel. 2020. Responsive Safety in Reinforcement Learning by PID Lagrangian Methods. In *ICML*.

[38] Felipe Petroski Such, Vashisht Madhavan, Edoardo Conti, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. 2017. Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning. In *CoRR*.

[39] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. The MIT Press.

[40] Brijen Thananjeyan, Ashwin Balakrishna, Ugo Rosolia, Felix Li, Rowan McAllister, Joseph E. Gonzalez, Sergey Levine, Francesco Borrelli, and Ken Goldberg. 2020. Safety Augmented Value Estimation from Demonstrations (SAVED): Safe Deep Model-Based RL for Sparse Cost Robotic Tasks. In *RA-L*.