# Guiding Evolutionary Strategies by Differentiable Robot Simulators

**Vladislav Kurenkov**[*]
Tinkoff
v.kurenkov@tinkoff.ru

**Bulat Maksudov**
Independent Researcher
b.maksudov@lua.tatar

## Abstract

In recent years, Evolutionary Strategies were actively explored in robotic tasks for policy search as they provide a simpler alternative to reinforcement learning algorithms. However, this class of algorithms is often claimed to be extremely sample-inefficient. On the other hand, there is a growing interest in Differentiable Robot Simulators (DRS) as they potentially can find successful policies with only a handful of trajectories. But the resulting gradient is not always useful for the first-order optimization. In this work, we demonstrate how DRS gradient can be used in conjunction with Evolutionary Strategies. Preliminary results suggest that this combination can reduce sample complexity of Evolutionary Strategies by 3x-5x times in both simulation and the real world.

## 1 Introduction

Evolutionary Strategies is a class of zeroth-order black-box optimization algorithms that were successfully applied in various simulated robotics tasks [13, 16]. They are known as easy to parallelize, and a small number of hyperparameters makes them easy to tune. However, these methods were observed to exhibit a higher sample complexity (in comparison to Reinforcement Learning algorithms) and a strong dependence on an initial network initialization [2], making their adoption for training robots directly in reality troublesome.

Recently, there was a rapid growth of work on Differentiable Robot Simulators (DRS) [6, 10, 8, 15, 17, 5]. The promise is to drastically reduce the number of samples needed for finding successful policies in comparison to reinforcement learning or black-box policy search methods. However, implementing DRS is non-trivial and requires specific approaches (especially for collision handling [8, 15]) to make the resulting gradient useful for first-order optimization. Moreover, first-order optimization methods that rely on DRS's gradient may get stuck in local optima and some specific treatment could be needed.

Lately, multiple attempts were made to merge Evolutionary Strategies (ES) and Reinforcement Learning (RL) algorithms [4, 11, 14]. Following this line of work, instead of combining ES and RL, we aim to combine the ends of the expected sample complexity spectrum, namely, DRS and Evolutionary Strategies. Specifically, in cases where the former is not useful for optimization with first-order methods. We propose to use recently introduced Guided Evolutionary Strategies (Guided-ES) [12] and treat DRS's gradient as a surrogate. Using this combination, we demonstrate that it is possible to reduce sample complexity of Evolutionary Strategies by 3x-5x times when training directly on a real robot (Section 3.1). Furthermore, we show in simulation that even misleading gradients from a DRS can be utilized to speed up the convergence of Evolutionary Strategies (Section 3.2).

[*]Primary author; Personal web-page: vkurenkov.me

**Algorithm 1** Evolutionary Strategies Guided by DRS

```
Input: Initial solution $\theta_0$; Optimizer $opt$; Cost function $f(\theta)$; DRS Gradient $\nabla f_{drs}$
Output: Final solution $\theta_T$
 1: for $t = 0$ to $T$ do
 2:     // DRS Part
 3:     Get DRS gradient $\nabla f_{drs}(\theta_t)$
 4:
 5:     // Guided-ES Part
 6:     Update low-dimensional guiding subspace $U$ with the DRS gradient
 7:     Define search covariance $\Sigma = \frac{\alpha}{n} I + \frac{1-\alpha}{k} UU^T$
 8:     for $i = 1$ to $P$ do
 9:         Sample perturbation $\epsilon_i \sim \mathcal{N}(0, \sigma^2 \Sigma)$
10:         Compute antithetic pair of losses $f(\theta_t + \epsilon_i)$ and $f(\theta_t - \epsilon_i)$
11:     end for
12:     Compute Guided ES gradient estimate $g = \frac{\beta}{2\sigma^2 P} \sum_{i=1}^{P} \epsilon_i [f(\theta_t + \epsilon_i) - f(\theta_t - \epsilon_i)]$
13:     Update parameters using given optimizer $\theta_{t+1} = opt.step(\theta_t, g)$
14: end for
15: return $\theta_T$
```

## 2 Evolutionary Strategies with Differentiable Robot Simulators

To unite DRS and Evolutionary Strategies, we propose to use an algorithm introduced in [12] – Guided Evolutionary Strategies. This method can make use of any surrogate gradients that are correlated with the true gradient to accelerate the convergence of evolutionary strategies. The surrogate gradients can be corrupted or biased in any way, the only requirement for them is to preserve a positive correlation with the true gradient. We presume that this is the case for the gradients computed with DRS and therefore propose to use it as a surrogate in Guided-ES.

The proposed approach is outlined in Algorithm 1. The key idea is to compute surrogate gradient $\nabla f_{drs}$ using a differentiable robot simulator. Then a history of $k$ previous surrogates can be formed into a matrix of size $n \times k$ (where $n$ is a size of search space) and an orthonormal basis $U$ extracted. This basis is next used to define a covariance matrix $\Sigma$ utilized for generating the perturbations. The hyperparameter $\alpha$ controls how biased should the perturbations be in the direction of the surrogate gradient.

We note that there is no need to rely on the proposed zeroth-order optimization if one has access to the exact gradient of the objective function. However, there are cases where the gradient obtained with DRS is not effective when used with first-order optimization methods to optimize the objective. In the following section, we will demonstrate two of them and show how one can benefit from the proposed approach to improve the sample efficiency of evolutionary strategies.

## 3 Experiments

### 3.1 Accelerated Learning on Real Robot

Evolutionary strategies and their variants were observed to be on par with reinforcement learning algorithms Salimans et al. [16], but require a larger amount of episodes for training in some cases. Still, algorithms of this class were successfully applied to train robots directly in the real world at the expense of higher experimental time [7]. Here, we expect that incorporating information from DRS into the training process should accelerate the convergence of evolutionary strategies.

We use the approach described in Section 2, where a surrogate gradient is taken as a difference between current parameters and parameters obtained after a fixed amount of optimization steps in simulation with DRS. The ascent direction obtained with DRS does not exactly match the ascent direction of the objective function $f_{real}(\theta)$ that depends on the real world, but we expect it to be at least positively correlated. A detailed algorithm can be found in the appendix[2].

[2]Source code: https://github.com/vkurenkov/guided-es-by-differentiable-simulators

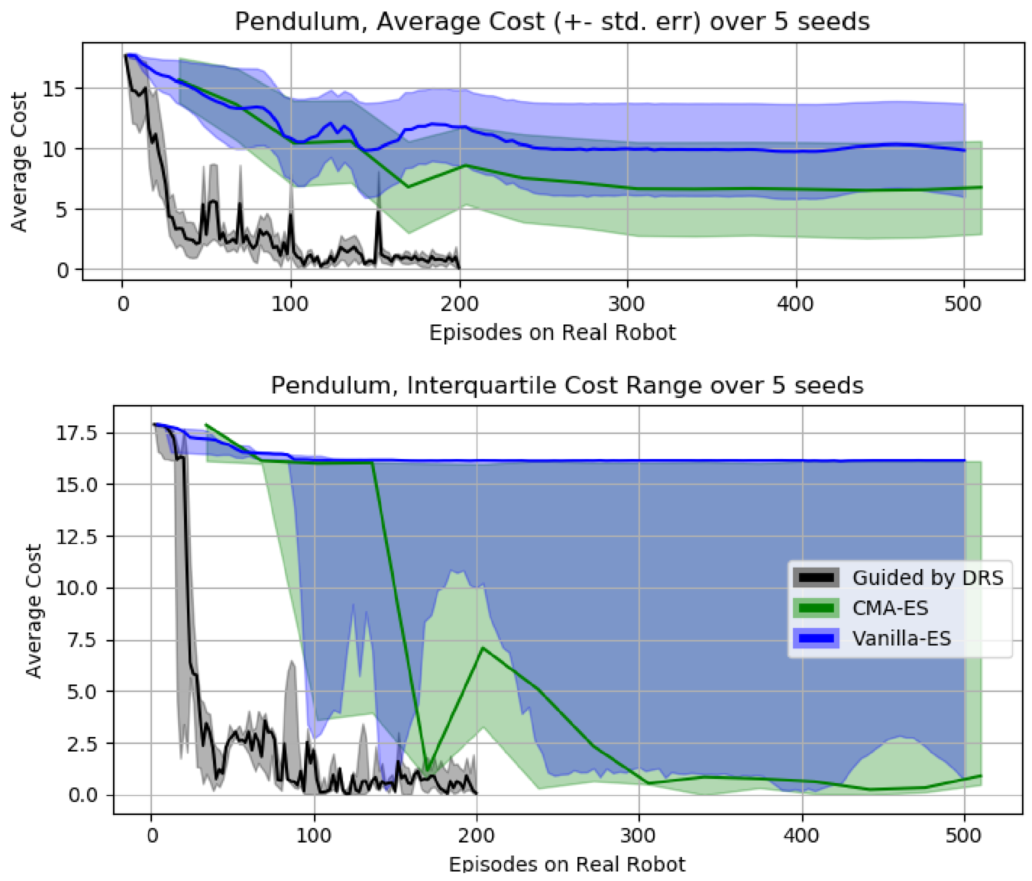

Figure 1: Guided-ES with DRS achieves considerably lower sample complexity for training on real robot in comparison to vanilla evolutionary strategies (Vanilla-ES) and Covariance Matrix Adaptation Evolution Strategy (CMA-ES). Moreover, the method is more robust across random seeds as indicated on the interquartile cost range graph.

Due to limited access to the experimental platform with physical robots, we only consider one problem – a swinging pendulum [3], where the goal is to start swinging constantly at 180 degrees. The cost function is defined as a sum of differences between current energy and target energy at each timestep. The state space is represented by the last four measurements of position and velocity.

We compare Guided-ES with DRS against Vanilla Evolutionary Strategies (Vanilla-ES) [16] and also against Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [9] to involve a method that adapts the covariance matrix during the training process. In Figure 1, we observe that the proposed method converges faster than both Vanilla-ES and CMA-ES, moreover, the convergence is robust to random seeds in opposition to other considered algorithms. We notice that Vanilla-ES and CMA-ES are highly dependent on initial network initialization (to the extent of non-convergence under our experimental budget), which is not the case for Guided-ES with DRS.

We find the results of this experiment encouraging as it gives a piece of evidence in favor of leveraging DRS (and a closed-loop transfer from simulation in general) for data-efficient evolutionary strategies. But more sophisticated robots and problems should be probed further.

## 3.2 When DRS Gradients Are Misleading

To demonstrate that the convergence of Guided-ES with DRS is possible even when the gradients are not useful for first-order optimization methods, we rely on a Mass-Spring simulator depicted in Figure 4. [10] observed that a naive implementation of this simulator results in a gradient that can not be used with a stochastic gradient descent – optimization process does not converge to a satisfactory solution. And a specific approach to collision handling (time-of-impact fix) is necessary. In this setup, we want to probe how well Guided-ES performs if provided with gradients computed using a naive version of the simulator, therefore we remove the fix proposed by [10].

We observe in Figure 2 that the proposed approach is able to find a satisfactory solution even when guided by the ineffective gradient. Moreover, the convergence on average is faster than with simple evolutionary strategies. However, we observe that variation between training iterations is quite high, but we believe it can be overcome with proper learning rate scheduling. On the other side, in-sample variation – within a training iteration but over multiple seeds, is low suggesting that a satisfactory result is found fast independent of a network initialization (which is not the case for Vanilla-ES). The results of this experiment suggest that if one has access to a differentiable robot simulator with inaccurate backward propagation, they should not abandon such a simulator. As there is a possibility it can still be used to reduce sample complexity of evolutionary strategies, which are easier to tune than reinforcement learning algorithms.

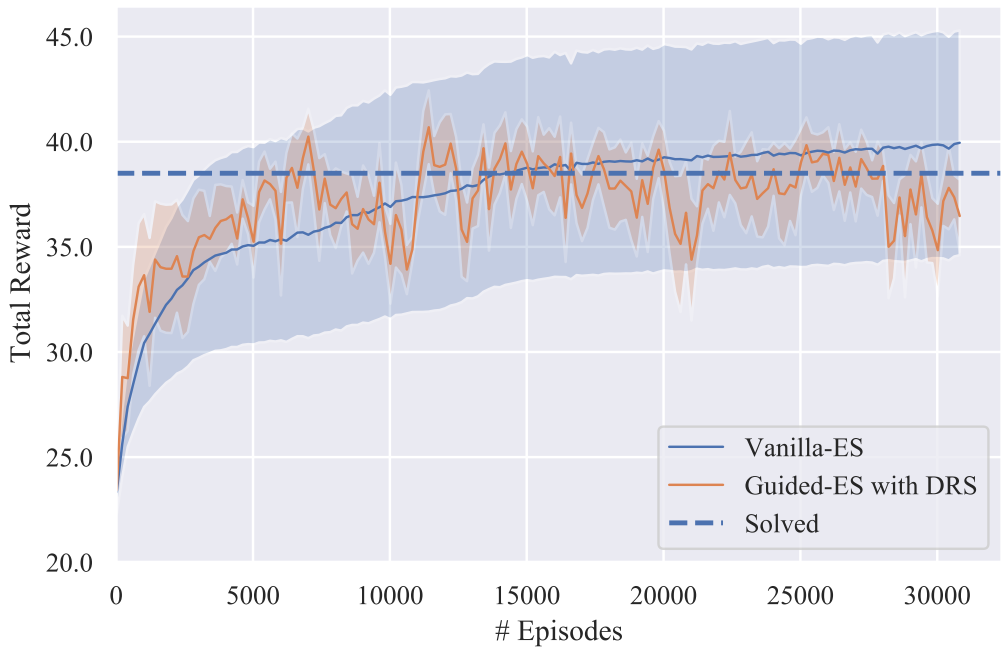

Figure 2: Guided-ES with DRS achieves lower sample complexity in comparison to vanilla evolutionary strategies (Vanilla-ES) with lower in-sample variation. *[Computed across 5 seeds for each algorithm, shaded area represents 95%-confidence interval.]*

## 4 Conclusion and Future Work

In this work, we proposed a natural way to combine Differentiable Robot Simulators and Evolutionary Strategies, and demonstrated two cases where such a combination could be beneficial to reduce sample complexity of the latter. We find these results encouraging, especially if one is interested to train robots directly in real life. However, future work should include more control robotic problems to probe the viability of the proposed method in more intricated setups that involve real and simulated robots.

## Acknowledgments

The authors would like to thank Oleg Balakhnov and Sergei Savin for helpful discussions and help with the robot hardware.

## References

[1] J. Bernstein, A. Vahdat, Y. Yue, and M.-Y. Liu. On the distance between two neural networks and the stability of learning. *arXiv:2002.03432 [cs, math, stat]*, Jan. 2021.

[2] P. Chrabaszcz, I. Loshchilov, and F. Hutter. Back to Basics: Benchmarking Canonical Evolution Strategies for Playing Atari. *arXiv:1802.08842 [cs]*, Feb. 2018.

[3] C. C. Chung and J. Hauser. Nonlinear control of a swinging pendulum. *Automatica*, 31(6): 851–862, June 1995. ISSN 0005-1098. doi: 10.1016/0005-1098(94)00148-C.

[4] E. Conti, V. Madhavan, F. Petroski Such, J. Lehman, K. Stanley, and J. Clune. Improving Exploration in Evolution Strategies for Deep Reinforcement Learning via a Population of Novelty-Seeking Agents. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[5] F. de Avila Belbute-Peres, K. Smith, K. Allen, J. Tenenbaum, and J. Z. Kolter. End-to-End Differentiable Physics for Learning and Control. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[6] J. Degrave, M. Hermans, J. Dambre, and F. wyffels. A Differentiable Physics Engine for Deep Learning in Robotics. *Frontiers in Neurorobotics*, 13:6, 2019. ISSN 1662-5218. doi: 10.3389/fnbot.2019.00006.

[7] D. Floreano and S. Nolfi. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. Intelligent Robotics and Autonomous Agents Series. A Bradford Book, Cambridge, MA, USA, Nov. 2000. ISBN 978-0-262-14070-6.

[8] M. Geilinger, D. Hahn, J. Zehnder, M. Bächer, B. Thomaszewski, and S. Coros. ADD: Analytically differentiable dynamics for multi-body systems with frictional contact. *ACM Transactions on Graphics*, 39(6):190:1–190:15, Nov. 2020. ISSN 0730-0301. doi: 10.1145/3414685.3417766.

[9] N. Hansen, S. D. Müller, and P. Koumoutsakos. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18, Mar. 2003. ISSN 1063-6560. doi: 10.1162/106365603321828970.

[10] Y. Hu, L. Anderson, T.-M. Li, Q. Sun, N. Carr, J. Ragan-Kelley, and F. Durand. DiffTaichi: Differentiable Programming for Physical Simulation. *arXiv:1910.00935 [physics, stat]*, Feb. 2020.

[11] S. Khadka and K. Tumer. Evolution-Guided Policy Gradient in Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[12] N. Maheswaranathan, L. Metz, G. Tucker, D. Choi, and J. Sohl-Dickstein. Guided evolutionary strategies: Augmenting random search with surrogate gradients. *arXiv:1806.10230 [cs, stat]*, June 2019.

[13] H. Mania, A. Guy, and B. Recht. Simple random search provides a competitive approach to reinforcement learning. *arXiv:1803.07055 [cs, math, stat]*, Mar. 2018.

[14] A. Pourchot and O. Sigaud. CEM-RL: Combining evolutionary and gradient-based methods for policy search. *arXiv:1810.01222 [cs, stat]*, Feb. 2019.

[15] Y.-L. Qiao, J. Liang, V. Koltun, and M. C. Lin. Scalable Differentiable Physics for Learning and Control. *arXiv:2007.02168 [cs, stat]*, July 2020.

[16] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever. Evolution Strategies as a Scalable Alternative to Reinforcement Learning. *arXiv:1703.03864 [cs, stat]*, Sept. 2017.

[17] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. Battaglia. Learning to Simulate Complex Physics with Graph Networks. In *Proceedings of the 37th International Conference on Machine Learning*, pages 8459–8468. PMLR, Nov. 2020.

## 5 Appendix

### 5.1 Hyperparameters

For experiments on the real robot, we did an extensive hyperparameter search for each algorithm in simulation and used the best set of them for training on the real robot. For experiments in Section 3.2 we also did an extensive hyperparameter search and reported the best curves for both of the algorithms.

We tried several optimization algorithms (SGD, Adam, RMSProp, AdaGrad, Fromage) and found the Fromage [1] to perform most stable when DRS gradients are involved, we believe this is due to its inherent property of normalizing the gradients, which we observed to be a necessary additional step for all other optimization algorithms to converge with DRS gradient.

### 5.2 Pseudocode for Accelerated Learning on Real Robot

**Algorithm 2** Evolutionary Strategies Guided by DRS for Training on Real Robot

```
Input: Initial solution $\theta_{(0)}$; Optimizers $opt_{real}, opt_{sim}$; Cost functions $f_{real}(\theta)$, $f_{drs}(\theta)$; DRS Gradient $\nabla f_{drs}$
Output: Final solution $x_{T_{real}}$
 1: for $t = 0$ to $T_{real}$ do
 2:     // DRS Part
 3:     $\theta_{sim} = \theta_t$
 4:     for $j = 1$ to $T_{sim}$ do
 5:         Make a step with DRS gradient $\theta_{sim} = opt_{sim}.step(\theta_{sim}, \nabla f_{drs}(\theta_{sim}))$
 6:     end for
 7:     Compute a simulation descent direction $g_{surrogate} = \theta_{sim} - \theta_t$
 8:
 9:     // Guided-ES Part
10:     Update low-dimensional guiding subspace $U$ with the $g_{surrogate}$
11:     Define search covariance $\Sigma = \frac{\alpha}{n} I + \frac{1-\alpha}{k} UU^T$
12:     for $i = 1$ to $P$ do
13:         Sample perturbation $\epsilon_i \sim \mathcal{N}(0, \sigma^2 \Sigma)$
14:         Compute antithetic pair of losses $f(\theta_t + \epsilon_i)$ and $f(\theta_t - \epsilon_i)$
15:     end for
16:     Compute Guided ES gradient estimate $g = \frac{\beta}{2\sigma^2 P} \sum_{i=1}^{P} \epsilon_i [f(\theta_t + \epsilon_i) - f(\theta_t - \epsilon_i)]$
17:     Update parameters using given optimizer $\theta_{t+1} = opt_{real}.step(\theta_t, g)$
18: end for
19: return $\theta_T$
```

### 5.3 Environments

#### 5.3.1 Pendulum

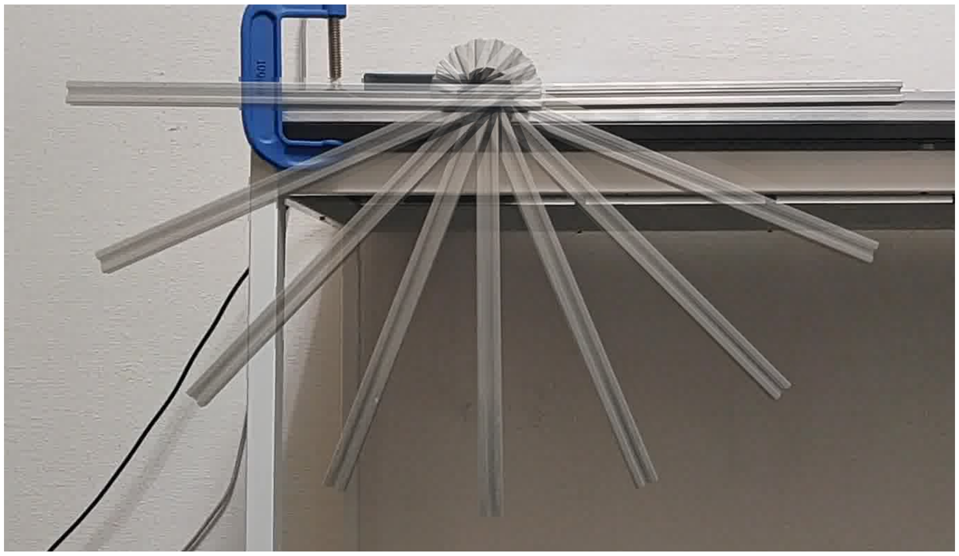

Figure 3: Pendulum robot. The goal is to start swinging at 180 degrees. The robot starts pointing downward at zero velocity and acceleration.

**Cost Function**

$$Cost = \sum_{t=0}^{Horizon} (Energy_t - TargetEnergy)^2$$
$$Energy_t = 0.5 * J * velocity^2 + mgl * sin(angle)$$
$$TargetEnergy = mgl * sin(2\pi)$$

#### 5.3.2 Mass-Spring

The environment is depicted in Figure 4, the objective function is defined as a difference between the center of mass positions for first and last timesteps projected at $x$.

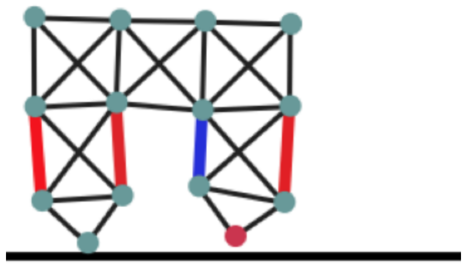

Figure 4: Mass-Spring robot environment. The robot should advance to the right as far as possible in a fixed amount of time. The actuation is done by varying the lengths of the springs colored in red or blue. One simulation corresponds to 8 seconds of real-time. For videos check Differentiable Mass-Spring Simulator section at `https://github.com/yuanming-hu/difftaichi`