ES Is More Than Just a Traditional Finite-Difference Approximator

Joel Lehman, Jay Chen, Jeff Clune, and Kenneth O. Stanley
Uber AI Labs
San Francisco, CA 94103
`{joel.lehman,jayc,jeffclune,kstanley}@uber.com`

Abstract

An evolution strategy (ES) variant recently attracted significant attention due to its surprisingly good performance at optimizing neural networks in challenging deep reinforcement learning domains. It searches directly in the parameter space of neural networks by generating perturbations to the current set of parameters, checking their performance, and moving in the direction of higher reward. The resemblance of this algorithm to a traditional finite-difference approximation of the reward gradient in parameter space naturally leads to the assumption that it is just that. However, this assumption is incorrect. The aim of this paper is to definitively demonstrate this point empirically. ES is a gradient approximator, but optimizes for a different gradient than just reward (especially when the magnitude of candidate perturbations is high). Instead, it optimizes for the average reward of the entire population, often also promoting parameters that are robust to perturbation. This difference can channel ES into significantly different areas of the search space than gradient descent in parameter space, and also consequently to networks with significantly different properties. This unique robustness-seeking property, and its consequences for optimization, are demonstrated in several domains. They include humanoid locomotion, where networks from policy gradient-based reinforcement learning are far less robust to parameter perturbation than ES-based policies that solve the same task. While the implications of such robustness and robustness-seeking remain open to further study, the main contribution of this work is to highlight that such differences indeed exist and deserve attention.

1 Introduction

Salimans et al. (2017) recently demonstrated that an evolutionary-inspired approach can compete on modern reinforcement learning (RL) benchmarks that require large-scale deep learning architectures. While the field of evolution strategies (ES; Schwefel 1993) has a rich history encompassing a broad variety of search algorithms (see Beyer and Schwefel 2002), Salimans et al. (2017) has drawn attention to the particular form of ES applied in that paper (which does not reflect the field as a whole), in effect a simplified version of natural ES (NES; Wierstra et al. 2014). Because this form of ES is the focus of this paper, herein it is referred to simply as *ES*. One way to view ES is as a policy gradient algorithm applied to the *parameter space* instead of to the *state space* as is more typical (Williams, 1992), and the distribution of parameters (rather than actions) is optimized to maximize the expectation of performance. Central to this interpretation is how ES estimates (and follows) the gradient of increasing performance with respect to the current distribution of parameters. In particular, in ES many independent parameter vectors are drawn from the current distribution, their performance is evaluated, and this information is then aggregated to estimate a gradient of distributional improvement.

The implementation of this approach is similar in its realization to a finite-differences gradient estimator (Richardson, 1911; Spall, 1992), wherein evaluating *tiny* perturbations of a parameter vector contribute to a direct calculation of the gradient of performance. As a result, it may be attractive to interpret the results of Salimans et al. (2017) solely through the lens of finite differences (e.g. as in Ebrahimi et al. 2017), concluding that the method is interesting or effective only because it is approximating the gradient of performance with respect to the parameters. However, such a hypothesis ignores that ES's objective function is interestingly different from traditional finite differences, which this paper argues grants it additional properties of interest. In particular, ES optimizes the performance of *any draw* from the learned *distribution* of parameters (called the search distribution), while finite differences optimizes the performance of *one particular* setting of the domain parameters. The main contribution of this paper is to support the hypothesis that this subtle distinction may in fact be important to understanding the behavior of ES (and future NES-like approaches), by conducting experiments that highlight how ES is driven to more *robust* areas of the search space than either finite differences or a more traditional evolutionary approach. The push towards robustness carries potential implications for RL and other applications of deep learning that could be missed without highlighting it specifically.

Note that the core interest of this paper is to clarify a subtle but interesting possible misconception, and not merely to debate the semantics of what exactly does or does not qualify as a finite-difference approximator. The framing here is that a traditional finite-difference gradient approximator makes tiny perturbations of domain parameters to estimate the gradient of improvement for the current *point* in the search space. While it is true that ES also stochastically follows a gradient (i.e. the *search gradient* of how to improve expected performance across the search distribution representing a *cloud* in the parameter space), it does not do so by standard application of common finite-differences methods. In any case, the most important distinction is that ES optimizes the expected value of a distribution of parameters with fixed variance, while traditional finite differences optimizes a singular optimal parameter vector.

To explore the empirical effects of this distinction, this paper first uses simple two-dimensional fitness landscapes to highlight systematic differences between the behavior of ES and other optimization approaches. The results from these theoretical experiments are then validated in the Humanoid Locomotion RL benchmark domain, showing that ES's drive towards robustness manifests also in complex domains. Interestingly, parameter vectors resulting from ES are much more robust than those of similar performance discovered by a genetic algorithm (GA) or by a non-evolutionary policy gradient approach (TRPO).

These results have implications for researchers in evolutionary computation (EC; De Jong 2002) who have long been interested in properties like robustness (Wilke et al., 2001; Wagner, 2008; Lenski et al., 2006) and evolvability (Wagner and Altenberg, 1996; Kirschner and Gerhart, 1998; Lehman and Stanley, 2013), and also for deep learning researchers seeking to more fully understand ES and how it relates to gradient-based methods.

2 Background

The method of finite differences is reviewed first, followed by the related approach of following search gradients applied by ES. Finally, methods that (like ES) encourage robustness in EC are discussed.

2.1 Finite Differences

One popular numerical approach to estimating the gradient of a function is the *finite-difference* method. The idea is that a tiny (but finite) perturbation is applied to the parameters of a scalar-valued system (i.e. a system outputting a scalar), and that evaluating the effect of such perturbations enables approximating the derivative with respect to the parameters. Such a method is useful for optimization when the system as a whole is not differentiable, e.g. in a RL context, when reward comes from a partially-observable or analytically-intractable environment. Indeed, because of its conceptual simplicity there are many policy gradient methods motivated by the finite-difference approach (Spall, 1992; Glynn, 1987).

One common finite-difference estimator of the derivative of function f with respect to the scalar x is given by:

$$f'(x) \approx \frac{f(x) + f(x+\delta)}{\delta},$$

given some small constant δ. This estimator generalizes naturally to the setting of parameter vectors, where the partial derivative with respect to each element of such a vector can be similarly calculated; however, naive finite differences scales poorly with the size of the problem, as it perturbs each parameter *individually*, making its application infeasible for large problems (like optimizing deep neural networks). However, finite-difference-based methods such as simultaneous perturbation stochastic approximation (SPSA; Spall 1992) can aggregate information from independent perturbations of all parameters simultaneously to estimate the gradient more efficiently. Indeed, SPSA is similar in implementation to ES.

However, the motivation for traditional finite-differences methods, importantly, is to use *tiny* perturbations; the larger such perturbations become, the less meaningfully it approximates the underlying gradient, which is formally defined as the slope of the function with respect to its parameters at a particular point. In other words, as perturbations become larger, finite differences becomes qualitatively disconnected from the principles motivating its construction; its estimate becomes increasingly influenced by the curvature of the reward function, and its interpretation becomes unclear. This consideration is important because ES is *not* motivated by tiny perturbations nor by approximating the gradient of performance for any singular setting of parameters, as the next section describes in more detail.

2.2 Search Gradients

Instead of searching directly for a single high-performing setting of a parameter vector, as is typical in gradient descent and finite-difference methods, a distinct (but related) approach is to optimize the *search distribution* of domain parameters to achieve high average reward when a particular parameter vector is sampled from the distribution (Berny, 2000; Wierstra et al., 2014; Sehnke et al., 2010). Doing so requires following *search gradients* (Berny, 2000; Wierstra et al., 2014), i.e. gradients of increasing expected fitness with respect to distributional parameters (e.g. the mean and variance of a Gaussian distribution).

While the procedure for following such search gradients uses mechanisms similar to a finite-difference gradient approximation (i.e. it involves aggregating fitness information from samples of domain parameters in a local neighborhood), importantly the underlying objective function from which it derives is different:

$$J(\theta) = \mathrm{E}_{\theta} f(z) = \int f(z)\pi(z|\theta)dz, \tag{1}$$

where $f(z)$ is the fitness function, and z is a sample from the search distribution $\pi(z|\theta)$ specified by parameters θ. Equation 1 formalizes the idea that ES's objective (like other search-gradient methods) is to optimize the *distributional parameters* such that the expected fitness of *domain parameters* drawn from that search distribution is maximized. In contrast, the objective function for more traditional gradient descent approaches is to find the optimal domain parameters directly: $J(\theta) = f(\theta)$.

While NES allows for adjusting both the mean and variance of a search distribution, in the ES of Salimans et al. (2017), the evolved distributional parameters control only the *mean* of a Gaussian distribution and not its *variance*. As a result, ES cannot reduce variance of potentially-sensitive parameters; importantly, the implication is that ES will be driven towards *robust* areas of the search space. For example, imagine two paths through the search space of similarly increasing reward, where one path requires precise settings of domain parameters (i.e. only a low-variance search distribution could capture such precision) while the other does not. In this scenario, ES with a sufficiently-high variance setting will only be able to follow the latter path, in which performance is generally robust to parameter perturbations. The experiments in this paper illuminate circumstances in which this robustness property of ES impacts search. Note that the relationship of *low-variance* ES (which bears stronger similarity to finite differences) to SGD is explored in more depth in Zhang et al. (2017).

2.3 Robustness in Evolutionary Computation

Researchers in EC have long been concerned with robustness in the face of mutation (Wilke et al., 2001; Wagner, 2008; Lenski et al., 2006), i.e. the idea that randomly mutating a genotype will not devastate its functionality. In particular, evolved genotypes in EC often *lack* the apparent robustness of natural organisms (Lehman and Stanley, 2011b), which can hinder progress in an evolutionary algorithm (EA). In other words, robustness is important for its link to *evolvability* (Kirschner and Gerhart, 1998; Wagner and Altenberg, 1996), or the ability of evolution to generate productive heritable variation.

As a result, EC researchers have introduced mechanisms useful to encouraging robustness, such as self-adaptation (Meyer-Nieberg and Beyer, 2007), wherein evolution can modify or control aspects of generating variation. Notably, however, such mechanisms can emphasize robustness *over* evolvability depending on selection pressure (Clune et al., 2008; Lehman and Stanley, 2011b), i.e. robustness can be trivially maximized when a genotype encodes that it should be subjected only to trivial perturbations. ES avoids this potential pathology because the variance of its distribution is fixed, although in a full implementation of NES variance is subject to optimization and the robustness-evolvability trade-off would likely re-emerge.

While the experiments in this paper show that ES is drawn to robust areas of the search space as a direct consequence of its objective (i.e. to maximize expected fitness across its search distribution), in more traditional EAs healthy robustness is often a second-order effect (Lehman and Stanley, 2011b; Kounios et al., 2016; Wilke et al., 2001). For example, if an EA lacks elitism and mutation rates are high, evolution favors more robust optima although it is not a direct objective of search (Wilke et al., 2001); similarly, when selection pressure rewards phenotypic or behavioral divergence, self-adaptation can serve to balance robustness and evolvability (Lehman and Stanley, 2011b).

It is important to note that the relationship between ES's robustness drive and evolvability is nuanced and likely domain-dependent. For example, some domains may indeed require certain NN weights to be precisely specified, and evolvability may be hindered by prohibiting such specificity. Thus an interesting open question is whether ES's mechanism for generating robustness can be enhanced to better seek evolvability in a domain-independent way, and additionally, whether its mechanism can be abstracted such that its direct search for robustness can also benefit more traditional EAs.

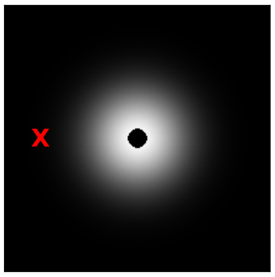
3 Experiments

The approach of this paper is to demonstrate empirically the way in which the ES of Salimans et al. (2017) systematically differs from a more traditional gradient-following approach. First, in a series of toy landscapes, a finite-differences approximator of domain parameter improvement is contrasted with ES's approximator of distributional parameter improvement, to highlight similarities and differences. Additionally, in the limit of decreasing variance, the convergence of ES's distributional gradient to the domain parameter gradient is also empirically validated.

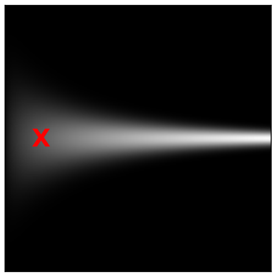
Then, to explore whether these differences also manifest in real world domains, the robustness property of ES is investigated in a popular RL benchmark, i.e. the Humanoid Locomotion task (Brockman et al., 2016). Policies from ES are compared with those from a genetic algorithm (GA) and a representative policy gradient RL algorithm called trust region policy optimization (TRPO; Schulman et al. 2015), to explore whether ES is drawn to qualitatively different areas of the parameter space.

3.1 Fitness Landscapes

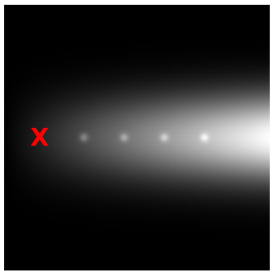
This section introduces a series of illustrative fitness landscapes (shown in figure 1), in which the behavior of ES and finite differences can easily be contrasted. In each landscape, performance is a deterministic function of two variables. For ES, the distribution over variables is an isotropic Gaussian with fixed variance as in Salimans et al. (2017), i.e. each dimension varies independently from the other. That is, ES optimizes two distributional parameters that encode the location of the distribution's *mean*. In contrast, while the finite-difference gradient-follower also optimizes two parameters, these represent a *single* instantiation of domain parameters, and consequently its function thus depends only on $f(\theta)$ at that singular position.



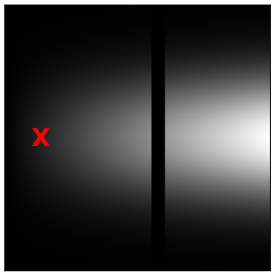
(a) Donut Landscape



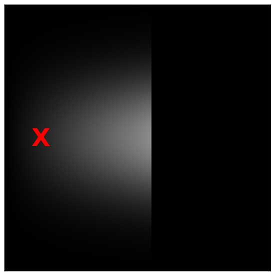
(b) Narrowing Path Landscape



(c) Fleeting Peaks Landscape



(d) Gradient Gap Landscape



(e) Gradient Cliff Landscape

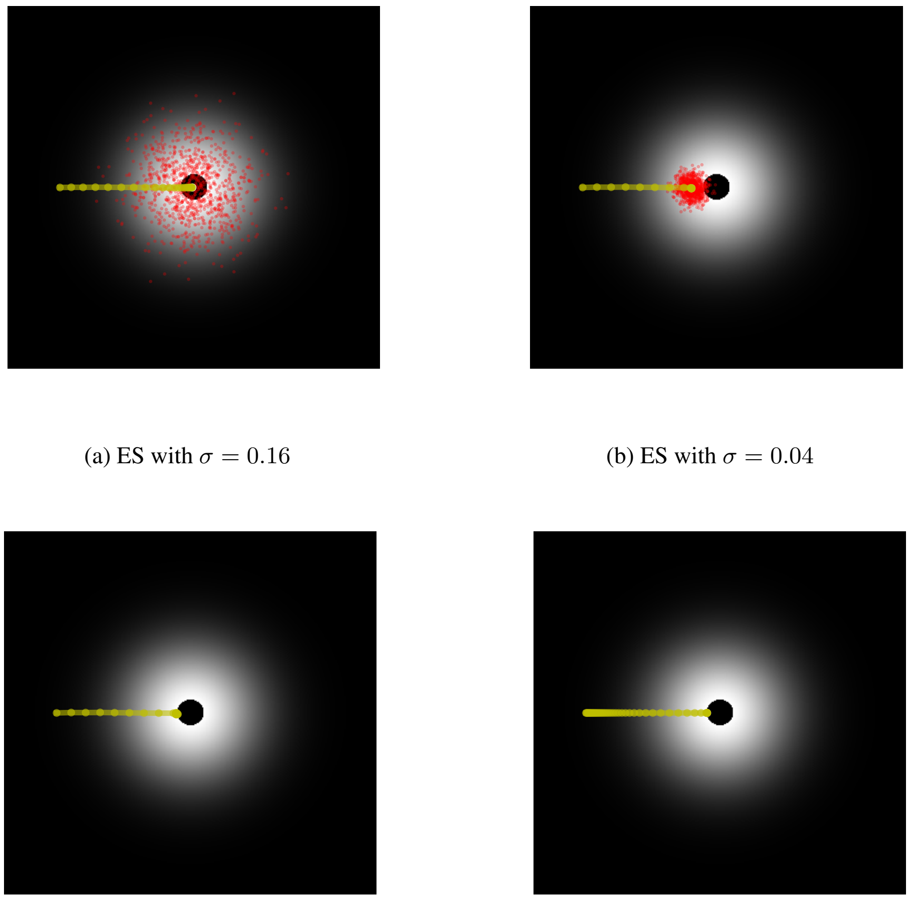
Figure 1: **Illustrative fitness landscapes.** A series of five fitness landscapes highlight divergences between the behavior of ES and finite differences. In all landscapes, darker colors indicate lower fitness and the red X indicates the starting point of search. In the (a) Donut landscape, there is a single Gaussian fitness peak, but the small neighborhood immediately around and including the peak has zero reward. In the (b) Narrowing Path landscape, fitness increases to the right, but the peak's spread increasingly narrows, testing an optimizer's ability to follow a narrow path. In the (c) Fleeting Peaks landscape, fitness increases to the right, but optimization to the true peak is complicated by a series of small local optima. The (d) Gradient Gap landscape is complicated by a gradient-free zero-reward gap in an otherwise smooth landscape, highlighting ES's ability to cross fitness plateaus (i.e. escape areas of the landscape where there is no local gradient). A control for the Gradient Gap landscape is the (e) Gradient Cliff landscape, wherein there is no promising area beyond the gap.

In the Donut landscape (figure 1a), when the variance of ES's Gaussian is high enough (i.e. σ of the search distribution is set to 0.16, shown in figure 2a), ES maximizes distributional reward by centering the mean of its domain parameter distribution at the middle of the donut where fitness is *lowest*; figure 3a further illuminates this divergence. When ES's variance is smaller ($\sigma = 0.04$), ES instead positions itself such that the tail of its distribution avoids the donut hole (figure 2b). Finally, when ES's variance becomes tiny ($\sigma = 0.002$), the distribution becomes tightly distributed along the edge of the donut-hole (figure 2c). This final ES behavior is qualitatively similar to following a finite-difference approximation of the domain parameter performance gradient (figure 2d).

In the Narrowing Path landscape (figure 1b), when ES is applied with high variance ($\sigma = 0.12$) it is unable to progress far along the narrowing path to higher fitness (figure 4a), because expected value is highest when a significant portion of the distribution remains on the path. As variance declines (figures 4b and 4c), ES proceeds further along the path. Finite-difference gradient descent is able to easily traverse the entire path (figure 4d).

In the Fleeting Peaks landscape (figure 1c), when ES is applied with high variance ($\sigma = 0.16$) the search distribution has sufficient spread to ignore the local optima and proceeds to the maximal-fitness area (figure 5a). With medium variance ($\sigma = 0.048$; figure 5b), ES gravitates to each local optima before leaping to the next one, and ultimately becomes stuck on the last local optimum (see also figure 3b). With low variance ($\sigma = 0.002$; figure 5c), ES latches onto the first local optimum and remains stuck there indefinitely. Finite-difference gradient descent becomes stuck on the same local optimum (figure 5d).

Finally, in the Gradient Gap landscape (figure 1d), ES with high variance ($\sigma = 0.18$) can traverse a zero-fitness non-diffentiable gap in the landscape (figure 6a), demonstrating ES's ability to "look



(a) ES with $\sigma = 0.16$

(b) ES with $\sigma = 0.04$

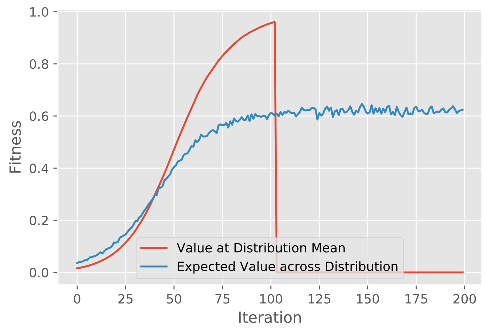
(c) ES with $\sigma = 0.002$

(d) Finite Differences with $\epsilon = 1e - 7$

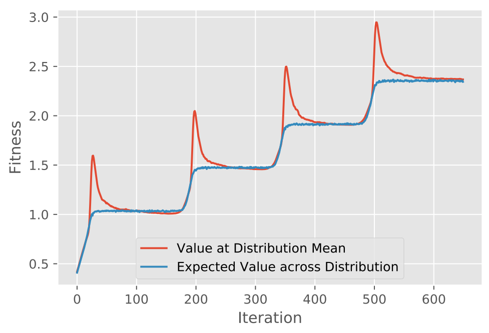
Figure 2: **Search trajectory comparison in the Donut landscape.** The plots compare representative trajectories of ES with decreasing variance in its search distribution to finite-differences gradient descent. With (a) high variance, ES maximizes expected fitness by moving the distribution's mean into a low-fitness area. With (b,c) decreasing variance, ES is drawn closer to the edge of the low-fitness area, qualitatively converging to the behavior of (d) finite-difference gradient descent.

ahead" in parameter space to cross fitness valleys between local optima (see also figure 3c). Lower variance ES (not shown) and finite differences cannot cross the gap (figure 6b). Highlighting that ES is informed by samples at the tail of the search distribution and is not blindly pushing forward, ES with high variance in the Gradient Cliff landscape (figure 6c) does not leap into the cliff, and lower variance ES (not shown) and finite differences (figure 6d) behave no different then they do in the Gradient Gap landscape.

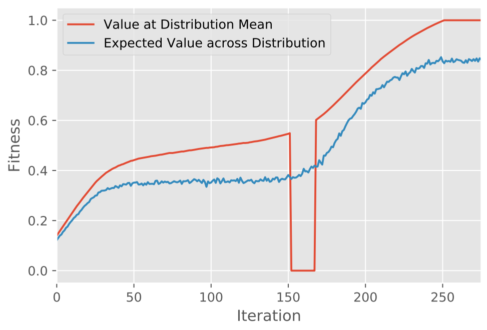
Overall, these landscapes, while simple, help to demonstrate that there are indeed systematic differences between ES and traditional gradient descent. They also show that no particular treatment is ideal in all cases, so the utility of the optimizing over a fixed-variance search distribution, at least for finding the global optimum, is (as would be expected) domain-dependent. The next section describes results in the Humanoid Locomotion domain that provide a proof-of-concept that these differences also manifest themselves when applying ES to modern deep RL benchmarks.



(a) Reward of ES on the Donut Landscape



(b) Reward of ES on the Fleeting Peaks Landscape



(c) Reward of ES on the Gradient Gap Landscape

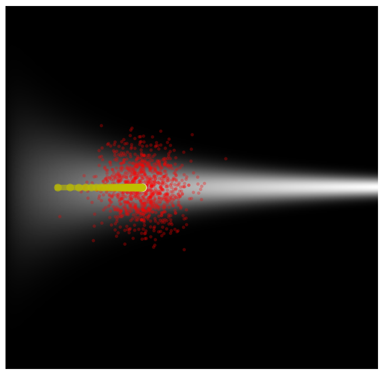
Figure 3: **ES maximizes expected value over the search distribution.** These plots show how the expected value of fitness and the fitness value evaluated at the distribution's mean can diverge in representative runs of ES. This divergence is shown on (a) the Donut landscape with *high* variance ($\sigma = 0.16$), (b) the Fleeting Peaks landscape with *medium* variance ($\sigma = 0.048$), and (c) the Gradient Gap landscape with *high* variance ($\sigma = 0.18$).

4 Humanoid Locomotion

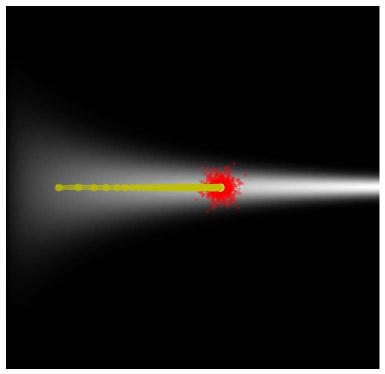
In the Humanoid Locomotion domain, a simulated humanoid robot is controlled by an NN controller with the objective of producing a fast energy-efficient gait (Tassa et al., 2012; Brockman et al., 2016), implemented in the Mujoco physics simulator (Todorov et al., 2012). Many RL methods are able to produce competent gaits, which this paper considers as achieving a fitness score of 6,000 averaged across many independent evaluations, following the threshold score in Salimans et al. (2017); averaging is necessary because the domain is stochastic. The purpose of this experiment is not to compare *performance* across methods as is typical in RL, but instead to examine the *robustness* of solutions, as defined by the distribution of performance in the neighborhood of solutions.

Three methods are compared in this experiment: ES, GA, and TRPO. Both ES and GA are methods that directly search through the parameter space for solutions, while TRPO uses gradient descent to modify policies directedly to more often take actions resulting in higher reward. All methods optimize the same underlying NN architecture, which is a feedforward NN with two hidden layers of 256 Tanh units, comprising approximately 167,000 weight parameters (recall that ES optimizes the same number of parameters, but that they represent the mean of a search distribution over domain parameters). This NN architecture is taken from the configuration file released with the source code from Salimans et al. (2017). The architecture described in their paper is similar, but smaller, having 64 neurons per layer (Salimans et al., 2017).

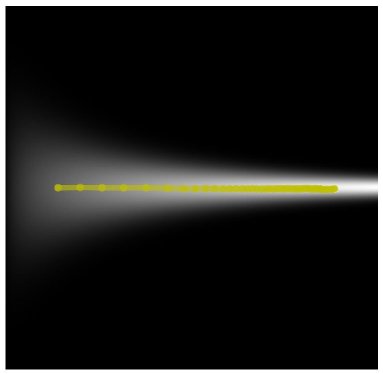
The hypothesis is that ES policies will be more robust to policy perturbations than policies of similar performance generated by either GA or TRPO. The GA of Petroski Such et al. (2017) provides a natural control, because its mutation operator is the same that generates variation within ES,



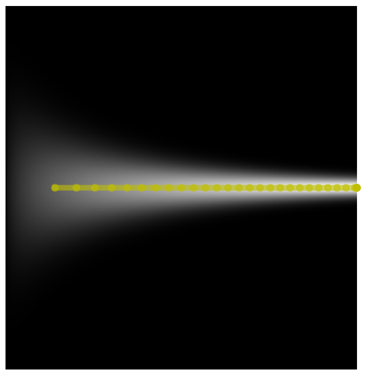
(a) ES with $\sigma = 0.12$



(b) ES with $\sigma = 0.04$



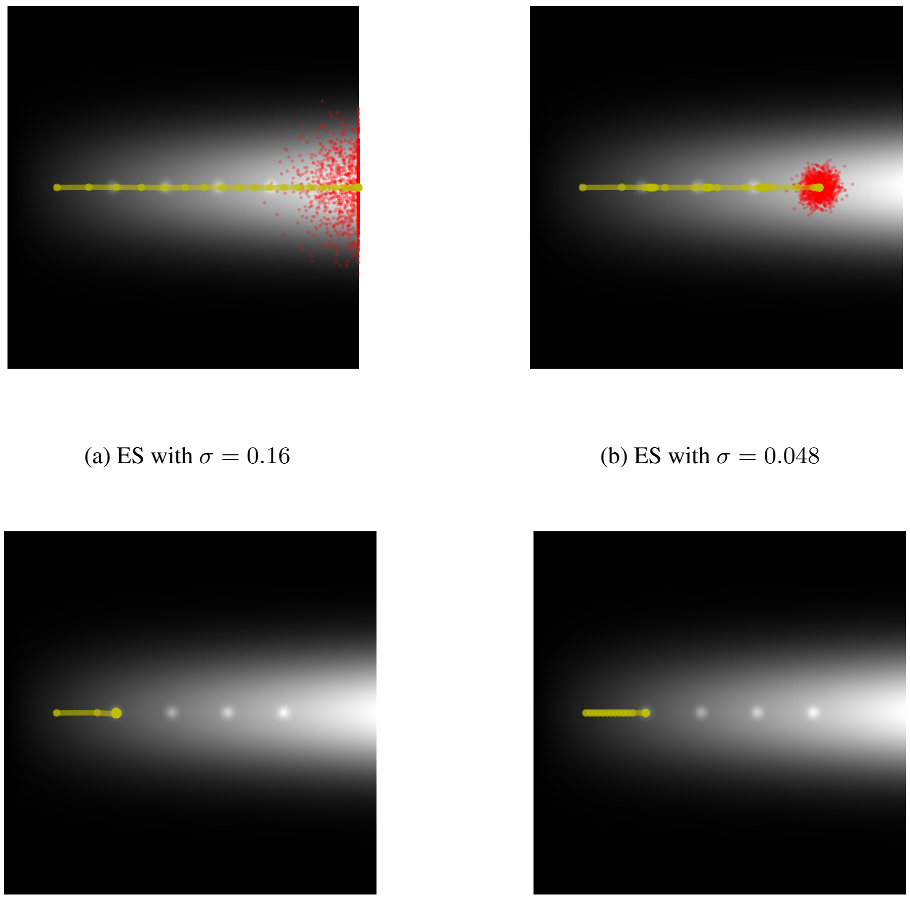
(c) ES with $\sigma = 0.0005$



(d) Finite Differences with $\epsilon = 1e-7$

Figure 4: **Search trajectory comparison in the Narrowing Path landscape.** With (a) high variance, ES maximizes expected fitness by staying on the wider part of the path, meaning it does not traverse the entire path. Importantly, if ES is being used to ultimately discover a single high-value policy, as is often the case (Salimans et al., 2017), this method will not discover the superior solutions further down the path. With (b,c) decreasing variance, ES is able to traverse further along the narrowing path. (d) Finite-difference gradient descent traverses the entire path.

but its objective function does not directly reward robustness. Note that ES is trained with policy perturbations from a Gaussian distribution with $\sigma = 0.02$ while the GA required a much narrower distribution ($\sigma = 0.00224$) for successful training (Petroski Such et al., 2017); training the GA with a larger mutational distribution destabilized evolution, as mutation too rarely would preserve or improve performance to support adaptation. Interestingly, this destabilization itself supports the idea that robustness to high variance perturbations is not pervasive throughout this search space. TRPO provides another useful control, because it follows the gradient of increasing performance without generating any random parameter perturbations; thus if the robustness of ES solutions is higher than that of those from TRPO, it also provides evidence that ES's behavior is distinct, i.e. it is *not* best understood as simply following the gradient of improving performance with respect to domain parameters (as TRPO does); note that this argument does not imply that TRPO is deficient if its policies are less robust to random parameter perturbations than ES, as such random perturbations are not part of its search process.


(a) ES with $\sigma = 0.16$

(b) ES with $\sigma = 0.048$

(c) ES with $\sigma = 0.002$

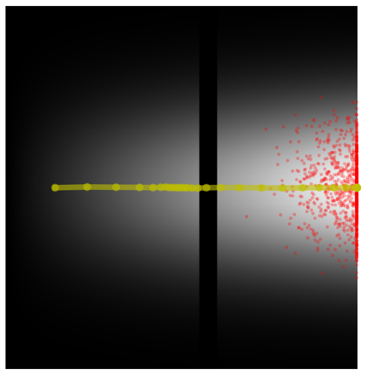
(d) Finite differences with $\epsilon = 1e-7$

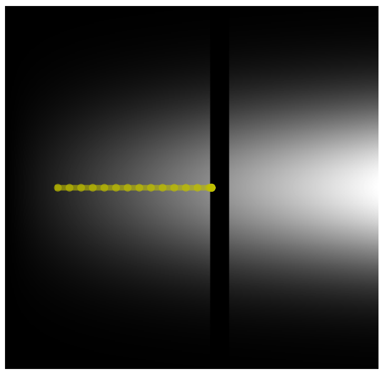
Figure 5: **Search trajectory comparison in the Fleeting Peaks landscape.** With (a) high variance, ES can bypass the local optima because its contribution to expected fitness across the distribution is small. With (b) medium variance, ES hops between local optima, and with (c) low variance, ES converges to a local optimum, similarly to (d) finite-difference gradient descent.

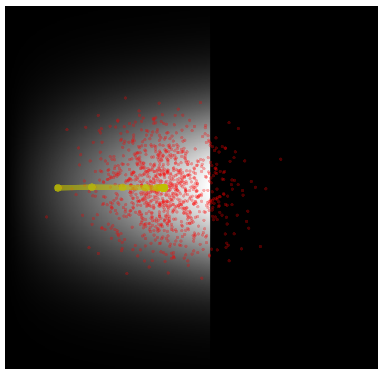
The experimental methodology is to take solutions from different methods and examine the distribution of resulting performance when policies are perturbed with the perturbation size of ES and of GA. In particular, policies are taken from generation 1,000 of the GA, from iteration 100 of ES, and from iteration 350 of TRPO, where methods have approximately evolved a solution of ≈6,000 fitness. The ES is run with hyperparameters according to Salimans et al. (2017), the GA is taken from Petroski Such et al. (2017), and TRPO is based on OpenAI's baselines package (Dhariwal et al., 2017). Exact hyperparameters are listed in the appendix.

4.1 Results

Figure 7 shows a representative example of the stark difference between the robustness of ES solutions and those from the GA or TRPO, even when the GA is subject only to the lower-variance perturbations that were applied during evolution. Qualitatively we observed that this result appears consistent across independently trained models. A video comparing perturbed policies of ES and TRPO can be viewed at the following URL (along with other videos showing selected fitness landscape animations): `https://goo.gl/yz1MeM`.


(a) ES on Gradient Gap with $\sigma = 0.18$


(b) Finite differences on Gradient Gap with $\epsilon = 1e-7$


(c) ES on Gradient Cliff with $\sigma = 0.18$

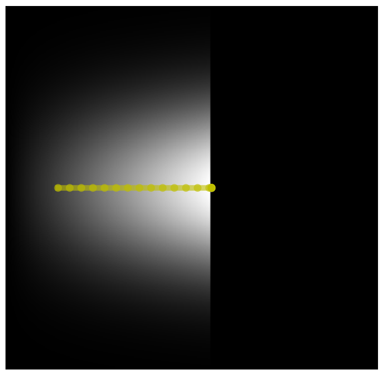
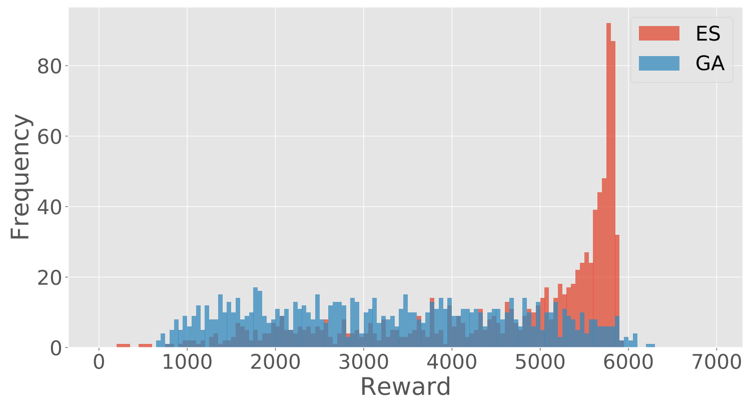
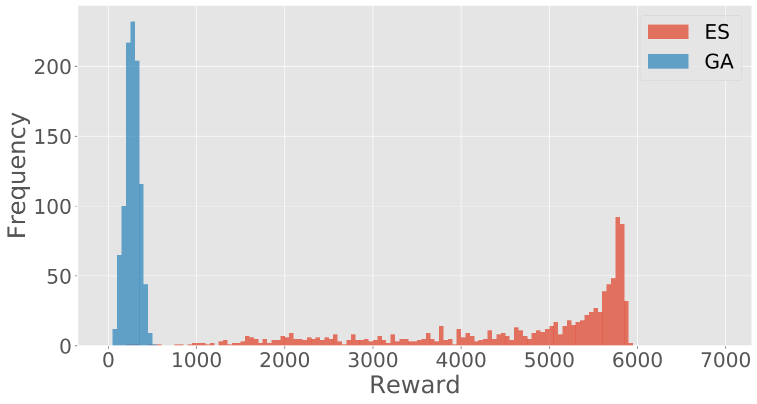

(d) Finite differences on Gradient Cliff with $\epsilon = 1e-7$

Figure 6: **Search trajectory comparison in the Gradient Gap and Gradient Cliff landscapes.** With (a) high variance, ES can bypass the gradient-free gap because its distribution can span the gap; with lower-variance ES or (b) finite differences, search cannot cross the gap. In the control Gradient Cliff landscape, (c) ES with high variance remains rooted in the high-fitness area, and the performance of (d) finite differences is unchanged from the Gradient Gap landscape.

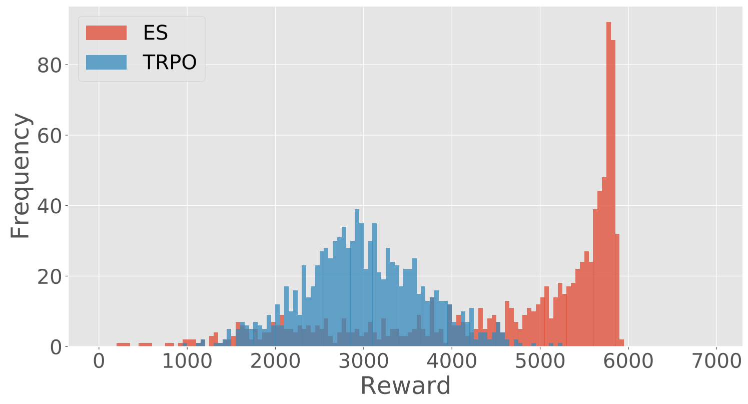
To further explore this robustness difference, a quantitative measure of robustness was also applied. In particular, for each model, the original parameter vector's reward was calculated by averaging its performance over 1,000 trials in the environment. Then, 1,000 perturbations were generated for each model, and each perturbation's performance was averaged over 100 trials in the environment. Finally, a robustness score is calculated for each model as the ratio of the perturbations' median performance to the unperturbed policy's performance, i.e. a robustness score of 0.5 indicates that the median perturbation performs half as well as the unperturbed model. The results (shown in figure 8) indicate that indeed by this measure ES is significantly more robust than the GA or TRPO (Mann-Whitney U-test; $p < 0.01$). The conclusion is that the robustness-seeking property of ES demonstrated in the simple landscapes also manifests itself in this more challenging and high-dimensional domain. Interestingly, TRPO is significantly more robust than both GA treatments (Mann-Whitney U-test; $p < 0.01$) even though it is not driven by random perturbations; future work could probe the relationship between the SGD updates of policy gradient methods and the random perturbations applied by ES and the GA.



(a) ES ($\sigma = 0.02$) vs GA ($\sigma = 0.002$)



(b) ES ($\sigma = 0.02$) vs GA ($\sigma = 0.02$)



(c) ES ($\sigma = 0.02$) vs TRPO ($\sigma = 0.02$)

Figure 7: **ES is more robust to parameter perturbation in the Humanoid Locomotion task.** The distribution of reward is shown from perturbing models trained by ES, GA, and TRPO. Models were trained to a fitness value of 6,000 reward, and robustness is evaluated by generating perturbations with Gaussian noise (with the specified variance) and evaluating perturbed policies in the domain. High-variance perturbations of ES produce a healthier distribution of reward than do perturbations of GA or TRPO.

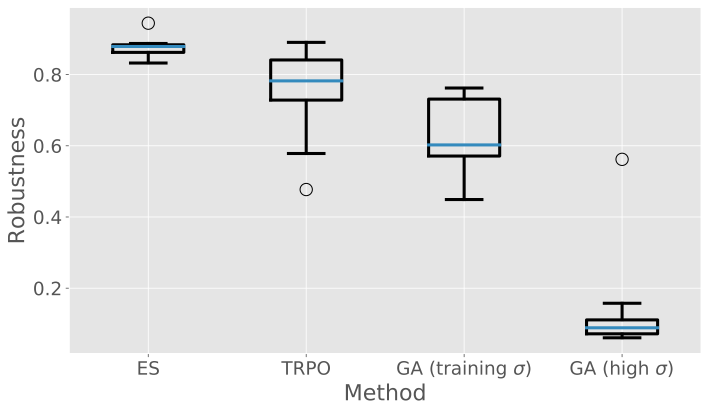


Figure 8: **Quantitative measure of robustness across independent runs of ES, GA, and TRPO.** The distribution of reward is shown from perturbing 10 independent models for each of ES, GA, and TRPO under the high-variance perturbations used to train ES ($\sigma = 0.02$). Results from GA are shown also for perturbations drawn from the lower-variance distribution it experienced during training ($\sigma = 0.00224$). The conclusion is that high-variance perturbations of ES retain significantly higher performance than do perturbations of GA or TRPO (Student's t-test; $p < 0.01$).

5 Discussion and Conclusion

An important contribution of this paper is to ensure that awareness of the robustness-seeking property of ES, especially with higher σ, is not lost – which is a risk when ES is described as simply performing stochastic finite differences. In effect, when σ is above some threshold, it is not accurate to interpret ES as merely an approximation of SGD, nor as a traditional finite-differences-based approximator. Rather, it becomes a gradient approximator coupled with a compass that seeks areas of the search space robust to parameter perturbations. This latter property is not easily available to point-based gradient methods, as highlighted dramatically in the Humanoid Locomotion experiments in this paper. On the other hand, if one wants ES to better mimic finite differences and SGD, that option is still feasible simply by reducing σ.

The extent to which seeking robustness to parameter perturbation is important remains open to further research. As shown in the landscape experiments, when it comes to finding optima, it clearly depends on the domain. If the search space is reminiscent of Fleeting Peaks, then ES is likely an attractive option for reaching the global optimum. However, if it is more like the Narrowing Path landscape, especially if the ultimate goal is a single solution (and there is no concern about its robustness), then high-sigma ES is less attractive (and the lower-sigma ES explored in Zhang et al. (2017) would be more appropriate). It would be interesting to better understand whether and under what conditions domains more often resemble Fleeting Peaks as opposed to the Narrowing Path.

An intriguing question that remains open is when and why such robustness might be desirable even for reasons outside of global optimality. For example, it is possible that policies encoded by networks in robust regions of the search space (i.e. where perturbing parameters leads to networks of similar performance) are also robust to other kinds of noise, such as domain noise. It is interesting to speculate on this possibility, but at present it remains a topic for future investigation. Perhaps parameter robustness also correlates to robustness to new opponents in coevolution or self-play (Popovici et al., 2012), but that again cannot yet be answered. Another open question is how robustness interacts with divergent search techniques like novelty search (Lehman and Stanley, 2011a) or quality diversity methods (Pugh et al., 2016); follow-up experiments to Conti et al. (2017), which combines ES with novelty search, could explore this issue. Of course, the degree to which the implications of robustness matter likely varies by domain as well. For example, in the Humanoid Locomotion task the level of domain noise means that there is little choice but to choose a higher σ during evolution (because otherwise the effects of perturbations could be drowned out by noise), but in a domain like MNIST there is no obvious need for anything but an SGD-like process (Zhang et al., 2017).

Another possible benefit of robustness is that it could be an indicator of compressibility: If small mutations tend not to impact functionality (as is the case for robust NNs), then less numerical precision is required to specify an effective set of network weights (i.e. fewer bits are required to encode them). This issue too is presently unexplored.

This study focused on ES, but it raises new questions about other related algorithms. For instance, non-evolutionary methods may be modified to include a drive towards robustness or may already share abstract connections with ES. For example, stochastic gradient Langevin dynamics (Welling and Teh, 2011), a Bayesian approach to SGD, approximates a distribution of solutions over iterations of training by adding Gaussian noise to SGD updates, in effect also producing a solution cloud. Additionally, it is possible that methods combining parameter-space exploration with policy gradients (such as Plappert et al. 2017) could be modified to include robustness pressure.

A related question is, do all evolutionary algorithms, which are generally population-based, possess at least the potential for the same tendency towards robustness? Perhaps some such algorithms have a different means of turning the knob between gradient following and robustness seeking, but nevertheless in effect leave room for the same dual tendencies. One particularly interesting relative of ES is the NES (Wierstra et al., 2014), which adjusts σ dynamically over the run. Given that σ seems instrumental in the extent to which robustness becomes paramount, characterizing the tendency of NES in this respect is also important future work.

We hope ultimately that the brief demonstration in this work can serve as a reminder that the analogy between ES and finite differences only goes so far, and there are therefore other intriguing properties of the algorithm that remain to be investigated.

Acknowledgements

We thank all of the members of Uber AI Labs, in particular Thomas Miconi, Martin Jankowiak, Rui Wang, Xingwen Zhang, and Zoubin Ghahramani for helpful discussions; Felipe Such for his GA implementation and Edoardo Conti for his ES implementation, both of which were used in this paper's experiments. We also thank Justin Pinkul, Mike Deats, Cody Yancey, Joel Snow, Leon Rosenshein and the entire OpusStack Team inside Uber for providing our computing platform and for technical support.

References

Berny, A. (2000). Statistical machine learning and combinatorial optimization. In *Theoretical Aspects of Evolutionary Computing*, pages 287–306. Springer.

Beyer, H.-G. and Schwefel, H.-P. (2002). Evolution strategies: A comprehensive introduction. *Natural Computing*, 1:3–52.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). OpenAI gym.

Clune, J., Misevic, D., Ofria, C., Lenski, R. E., Elena, S. F., and Sanjuán, R. (2008). Natural selection fails to optimize mutation rates for long-term adaptation on rugged fitness landscapes. *PLoS Computational Biology*, 4(9):e1000187.

Conti, E., Madhavan, V., Petroski Such, F., Lehman, J., Stanley, K. O., and Clune, J. (2017). Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents. *arXiv preprint to appear*.

De Jong, K. A. (2002). *Evolutionary Computation: A Unified Perspective*. MIT Press, Cambridge, MA.

Dhariwal, P., Hesse, C., Plappert, M., Radford, A., Schulman, J., Sidor, S., and Wu, Y. (2017). Openai baselines. `https://github.com/openai/baselines`.

Ebrahimi, S., Rohrbach, A., and Darrell, T. (2017). Gradient-free policy architecture search and adaptation. *ArXiv e-prints*, 1710.05958.

Glynn, P. W. (1987). Likelilood ratio gradient estimation: an overview. In *Proceedings of the 19th conference on Winter simulation*, pages 366–375. ACM.

Kirschner, M. and Gerhart, J. (1998). Evolvability. *Proceedings of the National Academy of Sciences*, 95(15):8420–8427.

Kounios, L., Clune, J., Kouvaris, K., Wagner, G. P., Pavlicev, M., Weinreich, D. M., and Watson, R. A. (2016). Resolving the paradox of evolvability with learning theory: How evolution learns to improve evolvability on rugged fitness landscapes. *arXiv preprint arXiv:1612.05955*.

Lehman, J. and Stanley, K. O. (2011a). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223.

Lehman, J. and Stanley, K. O. (2011b). Improving evolvability through novelty search and self-adaptation. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 2693–2700. IEEE.

Lehman, J. and Stanley, K. O. (2013). Evolvability is inevitable: Increasing evolvability without the pressure to adapt. *PLoS ONE*, 8(4):e62186.

Lenski, R. E., Barrick, J. E., and Ofria, C. (2006). Balancing robustness and evolvability. *PLoS biology*, 4(12):e428.

Meyer-Nieberg, S. and Beyer, H.-G. (2007). Self-adaptation in evolutionary algorithms. *Parameter setting in evolutionary algorithms*, pages 47–75.

Petroski Such, F., Madhavan, V., Conti, E., Lehman, J., Stanley, K. O., and Clune, J. (2017). Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. *arXiv preprint to appear*.

Plappert, M., Houthooft, R., Dhariwal, P., Sidor, S., Chen, R. Y., Chen, X., Asfour, T., Abbeel, P., and Andrychowicz, M. (2017). Parameter space noise for exploration. *arXiv preprint arXiv:1706.01905*.

Popovici, E., Bucci, A., Wiegand, R. P., and De Jong, E. D. (2012). *Coevolutionary Principles*, pages 987–1033. Springer Berlin Heidelberg, Berlin, Heidelberg.

Pugh, J. K., Soros, L. B., and Stanley, K. O. (2016). Quality diversity: A new frontier for evolutionary computation. 3(40).

Richardson, L. F. (1911). The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 210:307–357.

Salimans, T., Ho, J., Chen, X., and Sutskever, I. (2017). Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1889–1897.

Schwefel, H.-P. P. (1993). *Evolution and optimum seeking: the sixth generation*. John Wiley & Sons, Inc.

Sehnke, F., Osendorfer, C., Rückstieß, T., Graves, A., Peters, J., and Schmidhuber, J. (2010). Parameter-exploring policy gradients. *Neural Networks*, 23(4):551–559.

Spall, J. C. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control*, 37(3):332–341.

Tassa, Y., Erez, T., and Todorov, E. (2012). Synthesis and stabilization of complex behaviors through online trajectory optimization. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4906–4913. IEEE.

Todorov, E., Erez, T., and Tassa, Y. (2012). Mujoco: A physics engine for model-based control. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 5026–5033. IEEE.

Wagner, A. (2008). Robustness and evolvability: a paradox resolved. *Proceedings of the Royal Society of London B: Biological Sciences*, 275(1630):91–100.

Wagner, G. P. and Altenberg, L. (1996). Perspective: complex adaptations and the evolution of evolvability. *Evolution*, 50(3):967–976.

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 681–688.

Wierstra, D., Schaul, T., Glasmachers, T., Sun, Y., Peters, J., and Schmidhuber, J. (2014). Natural evolution strategies. *Journal of Machine Learning Research*, 15(1):949–980.

Wilke, C. O., nad Charles Ofria, J. L. W., Lenski, R. E., and Adami, C. (2001). Evolution of digital organisms at high mutation rates leads to survival of the flattest. *Nature*, 412:331–333.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Zhang, X., Clune, J., and Stanley, K. O. (2017). On the relationship between the openai evolution strategy and stochastic gradient descent. *arXiv preprint to appear*.

Appendix A Hyperparameters

This section describes the relevant hyperparameters for the search methods (ES, GA, and TRPO) applied in the Humanoid Walker experiments.

A.1 ES

The ES algorithm was based on Salimans et al. (2017) and uses the same hyperparameters as in their Humanoid Walker experiment. In particular, 10,000 roll-outs were used per iteration of the algorithm, with a fixed σ of the parameter distribution set to 0.02. The ADAM optimizer was applied with a step-size of 0.01.

A.2 GA

The GA was based on Petroski Such et al. (2017). The population size was set to 12,501, and σ of the normal distribution used to generate mutation perturbations was set to 0.00224. Truncation selection was performed, and only the highest-performing 5% of the population survived.

A.3 TRPO

The TRPO (Schulman et al., 2015) implementation was taken from the OpenAI baselines package (Dhariwal et al., 2017). The maximum KL divergence was set to 0.1 and 10 iterations of conjugate gradients were conducted per batch of training data. Discount rate (γ) was set to 0.99.