Enhanced POET: Open-ended Reinforcement Learning through Unbounded Invention of Learning Challenges and their Solutions

Rui Wang [1] **Joel Lehman** [1] **Aditya Rawal** [1] **Jiale Zhi** [1] **Yulun Li** [1] **Jeff Clune** [*2] **Kenneth O. Stanley** [*1]

Abstract

Creating *open-ended algorithms*, which generate their own never-ending stream of novel and appropriately challenging learning opportunities, could help to automate and accelerate progress in machine learning. A recent step in this direction is the Paired Open-Ended Trailblazer (POET), an algorithm that generates and solves its own challenges, and allows solutions to *goal-switch* between challenges to avoid local optima. However, the original POET was unable to demonstrate its full creative potential because of limitations of the algorithm itself and because of external issues including a limited problem space and lack of a universal progress measure. Importantly, both limitations pose impediments not only for POET, but for the pursuit of open-endedness in general. Here we introduce and empirically validate two new innovations to the original algorithm, as well as two external innovations designed to help elucidate its full potential. Together, these four advances enable the most open-ended algorithmic demonstration to date. The algorithmic innovations are (1) a domain-general measure of how meaningfully novel new challenges are, enabling the system to potentially create and solve *interesting* challenges endlessly, and (2) an efficient heuristic for determining when agents should goal-switch from one problem to another (helping open-ended search better scale). Outside the algorithm itself, to enable a more definitive demonstration of open-endedness, we introduce (3) a novel, more flexible way to encode environmental challenges, and (4) a generic measure of the extent to which a system continues to exhibit open-ended innovation. Enhanced POET produces a diverse range of sophisticated behaviors that solve a wide range of environmental challenges, many of which cannot be solved through other means.

1. Introduction

The progress of machine learning so far mostly relies upon a series of challenges and benchmarks that are manually conceived by the community (e.g. MNIST (LeCun et al., 1998), ImageNet (Deng et al., 2009), pole balancing (Anderson, 1989), and Atari (Bellemare et al., 2013)). Once a learning algorithm converges, or solves a task, there is nothing to gain by running it longer in that domain. Sometimes, learned parameters are transferred between challenges (Yosinski et al., 2014). However, in such cases a human manually chooses which task to transfer from and to, slowing the process and limiting the opportunities to harness such transfer to cases where humans recognize its value.

A fundamentally different approach is to create *open-ended* algorithms (Standish, 2003; Langdon, 2005; Bedau, 2008; Taylor et al., 2016; Stanley et al., 2017; Schmidhuber, 2013; Forestier et al., 2017) that propel *themselves* forward by conceiving simultaneously *both* challenges and solutions, thereby creating a never-ending stream of learning opportunities across expanding and sometimes circuitous webs of stepping stones. Such an algorithm also need not rely on our intuitions to determine either the right stepping stones or in what order they should be traversed to learn complex tasks, both notoriously difficult decisions (Stanley & Lehman, 2015). Instead, it could continually invent environments that pose novel challenges of appropriate difficulty, to stimulate further capabilities without being so difficult that all gradient of improvement is lost. The environments need not arrive in a strict sequence either; they can be discovered in parallel and asynchronously, in an ever-expanding tree of diverse challenges and their solutions.

The concept of *open-endedness* takes inspiration from natural evolution, which creates problems (also known as challenges, niches, environments, learning opportunities, etc.), such as reaching and eating the leaves of trees for nutrition, *and* their solutions, such as giraffes and caterpillars, in an ongoing process that has avoided stagnation and continued

[*]Co-senior authors [1]Uber AI [2]OpenAI. Work done at Uber AI. Correspondence to: Rui Wang <ruiwang@uber.com>, Jeff Clune <jclune@gmail.com>, Kenneth O. Stanley <kennethostanley@gmail.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

to produce novel artifacts for billions of years (and still continues). Open-endedness is also reflected in human innovation within art and science, which almost never unfold as a single linear progression of optimization aiming towards a given objective (Stanley & Lehman, 2015). Rather, they generate innumerable parallel and interacting branches of ideas, radiating continually in producing divergent outputs. New discoveries continue to extrapolate from their predecessors with no unified endpoint in mind. Open-endedness as a field of study encompasses all kinds of processes that have these properties (Stanley et al., 2017; Taylor et al., 2016). A fascinating, challenging research question is how we can create algorithms that exhibit such open-endedness; that is, can we ignite a process that unboundedly produces and solves increasingly diverse and complex challenges (given sufficient computation)?

The quest to achieve such open-endedness in computation has so far proven vexing (Taylor et al., 2016; Dolson et al., 2018). First, algorithms need to maintain a delicate balance between diversity (e.g. pursuing different kinds of solutions simultaneously) and optimization (e.g. giving one arguably "best" solution) (Brant & Stanley, 2017; Soros & Stanley, 2014; Pugh et al., 2016; Lehman & Stanley, 2011b; Mouret & Clune, 2015), as those solely focusing on optimization often lead to convergence. Second, the domain has to sustain endless opportunities to explore and learn something new. In a sense, there is a need for self-generated curricula that can continue to unfold indefinitely. Such curriculum building has begun to emerge as its own field of study in reinforcement learning (RL), as reviewed in Section 2. Finally, (unbounded) innovation needs to be measured quantitatively, a problem that still lacks a satisfying solution despite some thought-provoking efforts in the past (Bedau, 1992).

Recently, the *Paired Open-Ended Trailblazer* (POET) algorithm (Wang et al., 2019a;b) took a step towards tackling some of these challenges (and thus towards open-ended algorithms) by simultaneously creating problems (i.e. learning environments) while also learning to solve them. However, while it lays a foundation for open-ended computation, the original demonstration of POET (called *original POET* from here onward) still grapples with the field's longstanding challenges with balancing creativity and optimization. In addition, limitations of the domain on which it was tested, and the lack of a general measure of open-ended progress further complicated establishing its potential. Nevertheless, the fundamental insights behind POET come tantalizingly close to pushing past the limitations of convergent optimization, which could open up a new experimental paradigm in open-ended computation.

With this aim in mind, this paper first enhances the POET algorithm to more effectively generate and exploit diversity through two key innovations: (1) Instead of a hand-designed, domain-specific metric to decide the novelty of an environment, the first fully-generic method for identifying *meaningfully* novel environments is formulated based on the insight that what makes an environment interesting is how agents behave in it, and novel environments are those that provide new information about how the behaviors of agents within them differ; (2) a more computationally efficient heuristic is formalized for determining when agents should goal-switch from one environment to another. It also introduces two innovations that are external to the algorithm, but still crucial to demonstrating the potential of open-ended algorithms like POET: (3) a novel environmental encoding generates much more complex and diverse environments than what was used in original POET experiments, and (4) a novel measure for quantifying open-endedness allows the claim of enhanced open-endedness to be validated objectively. As shown by experiments in this paper, the result of these four innovations is a definitive demonstration of open-endedness, a phenomenon rarely observed in learning algorithms.

2. Related Work

The balance between diversity and optimization figures prominently in the field called *quality diversity* (QD) (Pugh et al., 2016; Lehman & Stanley, 2011b; Mouret & Clune, 2015), in which the aim is to collect a diversity of high-quality solutions. Results from QD algorithms have shown that simultaneously optimizing solutions to many different problems and allowing goal-switching between tasks (i.e. allowing a copy of a solution being optimized for one task to switch to start being optimized to solve a different task if it looks promising for that other task) dramatically improves performance, including solving previously unsolvable problems like Montezuma's Revenge or rapid damage recovery in robots (Cully et al., 2015; Nguyen et al., 2016; Ecoffet et al., 2019; Lehman & Stanley, 2011b; Mouret & Clune, 2015). Though closely related to open-endedness, QD does not involve the continual invention of new *problems*.

Other longstanding threads of research into *self-play* (Samuel, 1967; Bansal et al., 2018; Silver et al., 2018; OpenAI et al., 2019; Balduzzi et al., 2019) and Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) (both related to *coevolution* (Ficici & Pollack, 1998; Wiegand et al., 2001; Popovici et al., 2012)) have shown the benefit of optimizing against constantly changing, increasingly difficult challenges (e.g. against oneself or an opponent that also learns). Some recent exciting research also exists at the intersection of self-play and QD, e.g. AlphaStar (Vinyals et al., 2019) applies extensions of population-based training (Jaderberg et al., 2017) to maintain a diversity of high-quality strategies (Arulkumaran et al., 2019).

Recognition of the importance of self-generated curricula is also reflected in recent advances in automatic curriculum building for RL, where the intermediate goals of curricula towards a given, final objective are automatically generated via approaches such as goal generation (Florensa et al., 2018), reverse curriculum generation (Florensa et al., 2017), intrinsically motivated goal exploration processes (Forestier et al., 2017), teacher-student curriculum learning (Matiisen et al., 2017), or procedural content generation methods (usually focused on gaming) (Togelius et al., 2011; Shaker et al., 2016; Justesen et al., 2018). Generating training environments is also important for meta-learning (Finn et al., 2017; Duan et al., 2016; Wang et al., 2016), and AI-generating algorithms (Clune, 2019).

3. Methods

This section first describes the original POET framework (Wang et al., 2019a;b), and then details the two enhancements that help POET reach its potential of producing general open-ended innovation.

3.1. The Original POET Framework

To facilitate an open-ended process of discovery within a single run, POET grows and maintains a *population* of environment-agent pairs, where each agent is optimized to solve its paired environment (Figure 1). POET typically starts with a trivial environment and a randomly-initialized agent, then gradually creates new environments and searches for their solutions by performing three key steps: **(1)** Every M iterations POET generates new environments by applying mutations (i.e. random perturbations) to the *encoding*[1] of active environments whose paired agents have exhibited sufficient performance, signaling that perturbations of such environments are likely to be useful for encouraging learning progress. Once generated, new environments are filtered by a *minimal criterion* (Brant & Stanley, 2017) that ensures that they are neither too hard nor too easy for existing agents in the current population, i.e. that they are likely to provide a promising environment for learning progress. From those that meet this minimal criterion, only the most *novel* are added to the population, which pushes the environments towards capturing meaningfully *diverse* learning opportunities. Finally, when the population size reaches a preset threshold (set in accordance with available computational resources), adding a new environment results also in moving the oldest active one from the population into an inactive *archive*. **(2)** POET *continually* optimizes every agent in the population within its paired environment with a variant of the evolution strategies (ES) algorithm popularized by Salimans et al. (2017) (in principle, any RL algorithm could be used). **(3)** Every N iterations POET tests whether a copy

[1] In original POET, the environment is encoded as a small set of hand-picked parameters. A less limited and more sophisticated encoding is introduced later in this paper.

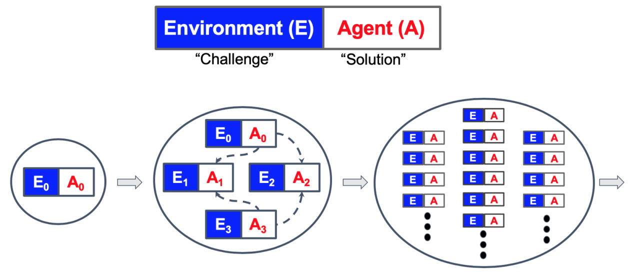


Figure 1. **POET maintains and grows a population of environment-agent pairs.** Each environment is paired with an agent being optimized to solve it. The system typically starts with a simple environment and then gradually creates and adds new environments (and their paired agents) with increasing diversity and complexity. POET harnesses goal-switching by periodically testing whether the current best solution to one challenge is also better than an incumbent on another challenge and, if so, replacing the incumbent with a copy of the better agent (dashed arrows).

of any agent should be transferred from one environment to another within the population to replace the target environment's existing paired agent (i.e. "goal-switching" to solve a different task), if the transferred agent either immediately (through *direct transfer*) or after one optimization step (through *fine-tuning transfer*) outperforms the incumbent. In original POET (Wang et al., 2019b;a), one of its main findings was that transfer is essential to finding solutions to increasingly complex and difficult environments.

3.2. Enhancing POET

A central aspiration of POET is to make open-ended discovery of new problems (environments) and agents that solve them as domain-independent and efficient as possible. The first enhancement in this section makes this kind of domain independence significantly more realistic than in the original POET. After that, we identify and fix an inefficiency in the original POET transfer mechanism.

3.2.1. Domain-General Environmental Characterization

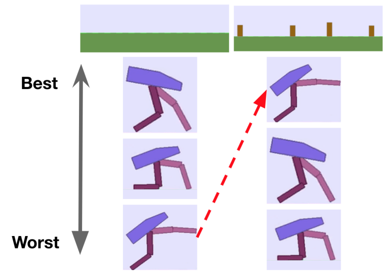
When POET is applied to a particular domain, such as the obstacle courses in this paper, two important concepts are essential to the search through environments: the *environmental encoding* (EE), which is a mapping from a parameter vector to an instance of an environment, creating an environmental search space, and the *environment characterization* (EC), which describes key attributes of an environment that thereby facilitate calculating *distances* between environments. POET harnesses this distance information to encourage the production of *novel*[2] environments. In

[2] The novelty of an environment is calculated as the average distance to its k nearest neighbors ($k = 5$ in this paper) among the

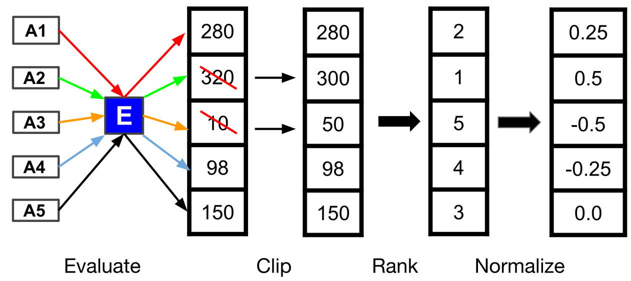
original POET, the EE and EC are both derived from the same set of static, hand-coded features that directly tie to the domain itself (e.g. the roughness of the terrain and the ranges of stump heights and gap sizes). This conflation of EE and EC seems convenient, but is also a key limitation to the system's creative potential: if the EC is itself hand-coded to fit the specific domain by e.g. specifying fixed, preconceived properties such as a terrain's smoothness or its vertical span, then the system's output will be bound to exploration only within such prescripted possibilities. A key contribution of this paper is thus to formulate an EC that is both domain-independent and principled from the perspective of open-ended innovation.

Our proposed domain-general EC, the *Performance of All Transferred Agents EC* (PATA-EC), is grounded by how all agents (in the population and archive) perform in that environment. The key insight motivating the PATA-EC is that a novel and useful challenge should make novel *distinctions* among agents in the system (de Jong & Pollack, 2004): if a newly-generated environment induces a significantly distinct *ordering* on how agents perform within it (relative to other environments), it likely poses a qualitatively new kind of challenge. For example, as illustrated in Figure 2a, the emergence of a new landscape with stumps may induce a new ordering on agents, as agents with different walking gaits may differ on their ability to step over protruding obstacles. An important aspect of this insight, based on how environments order agents, is that it does not rely upon any domain-specific information at all.

Figure 2b illustrates the steps to calculate the PATA-EC for any given environment: (1) **Evaluate**: Each environment evaluates all agents and stores their raw scores in a vector. Note that the required computation already occurs incidentally in the course of POET for active environments as a result of the transfer mechanism (which tries agents in their non-native environments). (2) **Clip**: Each score in the vector is clipped between a lower bound and an upper bound. The intuition is that both extreme scenarios are irrelevant for learning progress: a score that is too low indicates outright failure of an agent, while a score that is too high hints that an agent is already competent. This intuition is similar to that behind the "minimal criteria" in original POET; hence we adopt the same values of those bounds. (3) **Rank-normalize**: The PATA-EC is assembled by replacing the scores with rankings, and then normalizing each rank to the range of $[-0.5, 0.5]$ (similar techniques have been adopted in the ES literature, e.g. (Wierstra et al., 2014; Salimans et al., 2017)). Performing this normalization allows direct use of the Euclidean distance metric to measure the distance between PATA-ECs, which worked empirically better than



(a) The emergence of stumps induces different orderings of agents.



(b) Steps to calculate the PATA-EC for an environment (E).

Figure 2. **PATA-EC, a domain-general distance metric for measuring meaningfully different environments.** (a) An agent that walks with one leg raised is not energy-efficient on flat ground and thus ranks last, but that gaits enables it to step over high stumps and thus ranks highest in a more stumpy environment. (b) The calculation of the PATA-EC for environment (E) based on the rank of performance of five agents (A1–A5).

rank-correlation-based distance metrics directly defined on vectors of rankings in preliminary tests. Note that because of score clipping in Step 2, there could potentially be ties in the rank of specific agent ranks within a set of ranked agents. However, overall there is almost never a tie in all rankings. In the unlikely event there was an exact tie in all rankings, PATA-EC for both environments would be the same (ties broken randomly), but that outcome is still principled because the assumption is that both candidates are equally promising with respect to diversity.

PATA-EC is domain-general in the sense that it does not require any domain-specific information except for a predefined score function (just as most domains in RL assume such a reward or score function). This scoring function injects some human consideration into the system, but nevertheless it is generally much easier to define success than to define an open-ended measure of novelty. Measuring the performance of agents in an environment also helps ensure that an open-ended system generates tasks we want the system to solve. The implication of the PATA-EC is significant: We can now measure and reward environmental novelty completely independently of any domain-specific information, opening up POET to almost any conceivable domain.

active population and archive of environments (Lehman & Stanley, 2011a; Wang et al., 2019a;b).

3.2.2. Improved Transfer Strategy

Finally, POET's *transfer mechanism* enables innovations from solutions for one environment to aid progress in other environments by periodically attempting to replace an incumbent agent (for a target environment) with another agent in the population that performs better in that environment. While critical for the overall performance, the transfer mechanism in original POET also creates two problems: it is (1) computationally expensive because it involves an optimization step to compute the fine-tuning transfer score for each transfer evaluation, and (2) prone to "false positives" due to stochasticity in RL optimization and a low bar for replacing more proven incumbents. To effectively remedy both pitfalls, we introduce a more stringent threshold (i.e. the maximum of the 5 most recent scores of the incumbent) that *both* direct and fine-tuning transfer scores (instead of either one, as in original POET) of a candidate agent must exceed to qualify as an incoming transfer (Algorithm 1 in Appendix A.1). This simplification not only smooths out noise from the stochasticity of ES optimization, but also saves computation because the fine-tuning step is only performed if the direct transfer test is passed.

Now with the enhanced algorithm at hand, uncovering its full potential will require two additional innovations extrinsic to the algorithm itself.

4. More Expressive Environment Encoding

Even with the right algorithm, innovation will eventually grind to a halt if the domain itself is limited. The challenge lies in how to formalize an encoding that can sustain an environmental space with possibilities beyond the imagination of its designer. In original POET, the 2-D bipedal walking environments are encoded by a fixed, small set of hand-picked parameters (e.g. ranges of stump height and gap width, surface roughness, etc.) that can only support a finite number of obstacle types with predefined regular shapes and limited variations (e.g. Figure 4a). While this encoding expresses sufficient possibilities for POET to demonstrate an initial period of innovation, such innovation by necessity will eventually peter out as possible novel environments to explore gradually run out. To overcome this limitation, a desired encoding should be highly *expressive*, i.e. able to express environmental details with a high degree of granularity and precision to capture ever-more-intricate detail.

A class of neural networks known as compositional pattern-producing networks (CPPNs) (Stanley, 2007) are a candidate for a general encoding mechanism that respects this requirement. CPPNs take as input geometric coordinates (e.g. x and y), and when queried across such coordinates produce a geometric pattern (e.g. 2-D images). Figure 3 illustrates how to generate the landscape of a 2-D bipedal walking environment from a single-input, single-output CPPN, which is queried across the space of x coordinates that compose the landscape. Its output is interpreted as the height of the landscape at that point. (A selection of more complex CPPN landscapes from POET runs are later shown in Figures 4b and 8.) Because the obstacles in original POET are all simply deviations in the elevation of the ground at specific x locations, the CPPN encoding includes/subsumes all possible obstacle courses that can be generated by the original encoding (though with a different bias on obstacle courses produced), while also including many other configurations not possible in the original encoding.

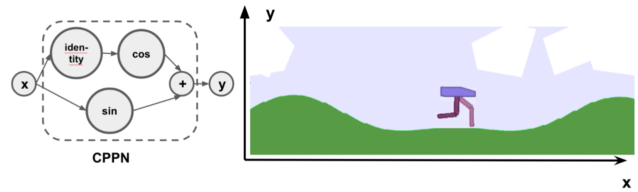


Figure 3. **A sample CPPN (left) and its generated landscape (right).** The CPPN produces y coordinates, given each x coordinate, which are then rendered into a bipedal walker environment for the Bipedal Walker environment in OpenAI Gym (Brockman et al., 2016). An agent, shown in the right figure, is controlled by a different agent neural network to navigate through the generated landscape and is rewarded for quickly moving from left to right.

As an encoding mechanism, CPPNs offer desirable properties for open-endedness: (1) They are typically initialized with simple topologies (e.g. no hidden nodes), and are trained with NEAT (Stanley & Miikkulainen, 2002), a neuroevolution (Stanley et al., 2019) algorithm that learns both the topology and the weights of CPPNs (details in Appendix A.2). As a result, simple (e.g. flat or sloped) landscapes are often produced in the beginning of a POET run, while more complex (and often more challenging) landscapes gradually emerge as NEAT's topology-altering mutations (e.g. adding a node or a connection) gradually elaborate the neural architecture of the CPPNs. (2) Because CPPNs can evolve arbitrarily complex architectures, in this domain they can in theory express any possible landscape at any conceivable resolution or size. More importantly, as shown in Section 6, the CPPN-based encoding provides a working example to demonstrate that the domain-general PATA-EC described in the previous section indeed can work in much richer environmental contexts, completely independently of their encoding mechanisms. Researchers who later apply POET to domains beyond what is shown in this paper can thus be fully creative in designing the encoding of their domain by enjoying the freedom to substitute any other encoding for CPPNs should that be more appropriate for their domain.

5. The ANNECS Measure of Progress

Measuring progress in open-ended systems has long presented a challenge to pursuing open-endedness: As there is no a priori expected outcome against which progress can be measured, how can we tell whether a system continues to generate *interesting* new things? The new idea here is that measuring progress can be based on the idea that if an existing set of agents are able to solve all of the new challenges generated by a system in the future, then the system has not generated any meaningfully new challenges. The system also should not generate problems with no hope of being solved. Therefore, we propose to track the *accumulated number of novel environments created and solved* (ANNECS) across the duration of a run of an open-ended system. Specifically, to be counted in ANNECS, an environment created at a particular iteration (1) must pass the minimal criterion (i.e. that it is neither too hard nor too easy) measured against all the agents (including ones currently in the active population and in the archive) generated over the entire current run so far, and, (2) must be eventually solved by the system (which means that the system does not receive credit for producing unsolvable challenges). This proposed metric ties directly to the overall effectiveness of an open-ended process: As the run proceeds, the ANNECS metric consistently going up indicates that the underlying algorithm is constantly creating meaningfully new environments.

6. Experiments and Results

With an enhanced algorithm, a more open-ended environment encoding, and a new means for measuring open-ended innovation over time, the question now is whether a definitive improvement in open-ended computation can be demonstrated. The aim in this section is to attack this question from several angles, both to show why open-endedness remains unique in its potential among all the methods in machine learning, and also how the enhancements to POET genuinely improve its tendency towards continual innovation.

For this purpose, the experimental approach is to empirically evaluate Enhanced POET in a domain adapted from the 2-D bipedal walking environment used in the original POET, which itself was based on the "Bipedal Walker Hardcore" environment of OpenAI Gym (Brockman et al., 2016). An instance of this experimental domain consists of a bipedal walking agent and an obstacle course that the agent attempts to navigate from left to right (Figure 3). The agent in this work has the same configuration as in the original POET, while the obstacle courses now can be encoded and generated by the CPPN-based EE (recall that the EE here is an environmental encoding mechanism that maps from a CPPN to an instance of a 2-D bipedal walking environment as described in Section 4). The experiments are organized to demonstrate the values of the four main contributions: we first evaluate the performance of the new EC (recall that the EC here is a domain-general characterization of key attributes of an environment to facilitate calculating distances between environments as described in Section 3.2.1) and improved transfer strategy, respectively, and then test the overall performance of the Enhanced POET with the CPPN-based EE. Lastly, the new ANNECS metric is put to the test, measuring progress in Enhanced POET and contrasting it with the original POET (Section 5). Unless noted otherwise, a POET run takes 60,000 POET iterations with a population size of 40 active environments. Because POET consists of many independent operations, such as agents optimizing within their paired environments, as well as transfer attempts, it is feasible and favorable to distribute the computations over many processors. Our software implementation, available at `https://github.com/uber-research/poet`, completes a 60,000-iteration POET run in about 12 days with 750 CPU cores. Further details about the domain and experiment setup are in Appendix A.3.

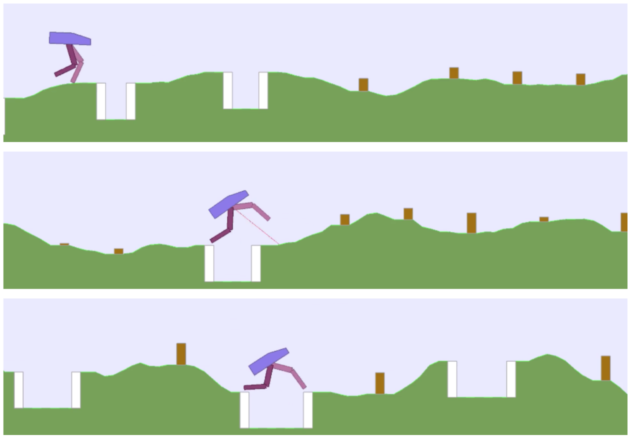
6.1. Empirical Evaluation of PATA-EC, Improved Transfer Strategy, and CPPN-based EE

The first experiment tests whether the proposed PATA-EC indeed encourages creating and solving a diverse set of challenges. When applied to the domain in the original POET (still with the original, hand-crafted EE), we find that PATA-EC can produce the same diversity and challenge levels of environments[3] as the original hand-designed EC, although it requires $82.4 \pm 7.31\%$ more computation, measured in ES steps (details in Appendix A.4.2). Because the original EC was hand-designed for this specific domain and encoding, its performance is the best we could reasonably expect from any EC in this domain. It is therefore promising that adopting PATA-EC with full generality only carries the price of less than a factor of two slowdown, but frees us to incorporate much richer EEs (such as CPPNs) into POET, thus powering the exploration of novel, unanticipated terrains.

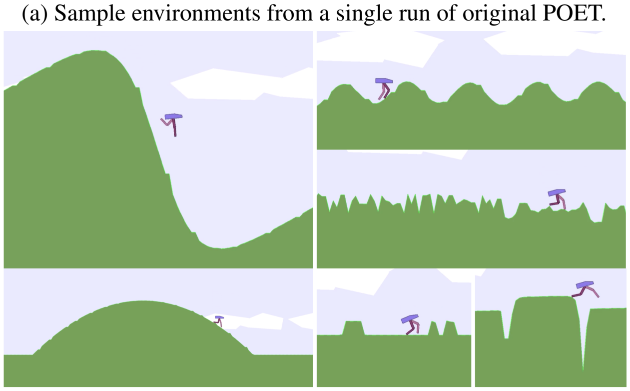
The second experiment evaluates the proposed improved transfer strategy. With the same setup as in original POET (Wang et al., 2019a;b) but with the improved transfer strategy, POET can create (and solve) the same diversity and challenge levels of environments with only $79.7 \pm 1.67\%$ of the computation (measured in number of ES steps) (details in Appendix A.4.3). This result suggests that the improved transfer strategy successfully reduces the cost of goal-switching in original POET without sacrificing its benefits with respect to solution discovery.

The next set of experiments leverage all four contributions of

[3] The definition of the three challenge levels follows that in Wang et al. (2019a;b), and is also explained in Appendix A.4.1.



(a) Sample environments from a single run of original POET.



(b) Sample environments from a single run of Enhanced POET.

Figure 4. **With the CPPN-based EE and other innovations, Enhanced POET is able to generate (and solve) a wide diversity of environments within a single run.** In contrast, the original POET can only generate environments with limited types of regularly-shaped obstacles (e.g. stumps and gaps).

this work. We first show that Enhanced POET with the new CPPN-based EE is able to create and solve a large diversity of environments within a single run. These are qualitatively different than those produced by the original POET with the simple, hand-designed EE that supports only a few types of simple obstacles (e.g., stumps and gaps) (Figure 4a; more in Figure 9 in Appendix A.5). The CPPN-encoded environments produced by the Enhanced POET exhibit a wide variety of obstacles vastly different in their overall shapes, heights, fine details, and subtle variations (Figure 4b; more in Figure 8 and links to videos of agents in Appendix A.5). Such diversity is also reflected in phylogenetic trees (also known as family trees) of the environments it has created, which exhibit a clear signature of open-ended algorithms: multiple, deep, hierarchically nested branches, resembling those from natural phylogenies (Figure 10 in Appendix A.6). It is also interesting to see that POET agents tend to be specialized to particular environments that pose very different challenges, as illustrated in the matrix formed by the vectors of scores of agents across all the first 80 environments created and solved in a POET run (Figure 11 in Appendix A.7).

6.2. Control Experiments with Direct Optimization and the Ground-Interpolation Curriculum

We next test an intriguing hypothesis that was also investigated for the original POET: Is it the case that some of the environments Enhanced POET generates are challenging enough that the curriculum it self-generates is *necessary* to solve them? This question is interesting because it implies that modern learning algorithms on their own may be able to achieve much more than is currently known if only they were embedded into an open-ended process like POET. Our approach is to sample environments created and solved throughout Enhanced POET runs and attempt to solve them with control algorithms. Specifically, we sort all the environments generated and eventually solved in a POET run in the order of when they are solved, and select one from the first 10%, one from the middle $(45\% - 55\%)$, and one from the last 10% of the run, in each case choosing the environment with the lowest initial score from that part of the run (indicating difficulty). These are referred to as *early stage*, *middle stage*, and *late stage* environments, respectively. The process was repeated for 5 independent POET runs (each with a different random seed) to obtain in total 15 environment targets. For each target environment, two different types of controls are attempted: One is direct optimizations by ES (with the same hyperparameters as in POET) and by the Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017) (hyperparameters in Appendix A.8.1), respectively. The other, stronger control is to manually create an explicit curriculum by introducing a scaling factor that multiplies the height of the ground at each position from left to right, and increase the scaling factor from 0.0 to 1.0 at a step size of 0.02. Doing so smoothly morphs the perfectly flat environment to a given target environment, yielding a natural curriculum (referred to later as the *ground-interpolation curriculum*) that is analogous to the direct-path curriculum in the original POET paper.

When given an equivalent computational budget to what POET spent to solve each target (details in Appendix A.8.2), the two types of controls can solve target environments selected at the earlier stages of POET runs (when the produced environments are often less challenging), but both significantly underperform POET in solving middle and late stage target environments ($p < 0.01$; Wilcoxon signed rank test). (The percentage of target environments solved by the controls and more results are given in Appendix A.8.3.) The result that neither direct optimization nor manually created curricula come close to producing the level of success in challenging environments that POET achieves via its self-generated implicit curriculum, confirms the result of the original POET paper in a new setting (and now also with PPO). Interestingly, much research effort is spent attempting to design or learn single-path curricula to help an agent learn a complex task (Gomez & Miikkulainen, 1997; Bengio et al., 2009; Karpathy & Van De Panne, 2012; Heess et al., 2017; Justesen et al., 2018). Here, we see (again) that such efforts often do not work. POET, however, is not trying to create any specific curriculum, but ends up producing many effective curricula (within one run) to solve many different challenging tasks. It does so because it collects an ever-expanding set of stepping stones (in the form of challenges and solutions) and allows goal-switching between them, which captures serendipitous discoveries as they occur (Stanley & Lehman, 2015; Nguyen et al., 2016; Lehman & Stanley, 2011b).

6.3. ANNECS Comparison

Finally, Figure 5 compares the ANNECS metric of progress proposed in Section 5 between the Enhanced POET with the new EE, and the original POET with the original (fixed) EE, a comparison that demonstrates the overall impact of both algorithmic innovations and the enhanced encoding. The original POET initially exhibits comparable performance to Enhanced POET, but eventually loses its ability to innovate, as shown by its ANNECS curve plateauing after 20,000 iterations. Such stagnation occurs because the EE for original POET can only sustain a finite number of obstacle types with predefined regular shapes and limited variations, so it gradually runs out of possible novel environments to explore. In intriguing contrast, innovation within Enhanced POET continues almost linearly throughout the experiments, though at a slightly slower speed beyond 30,000 iterations. This slight slowdown reflects that as environments generally become more challenging, it requires more optimization steps for environment-agent pairs to reach the score threshold for generating new environments (data not shown). Despite that, new environments that can pass the MC continue to be consistently discovered, significantly exceeding the duration of time that the original POET can continuously innovate. The result validates the ANNECS approach by aligning with our expectation that a limited encoding cannot support long-term innovation, while the longer chain of innovation of Enhanced POET is achieved because the CPPN-encoded environmental space offers significantly more potential for meaningful diversity. Furthermore, the domain-general PATA-EC and improved transfer strategy make it possible and efficient to explore, create and solve novel environments in such a space.

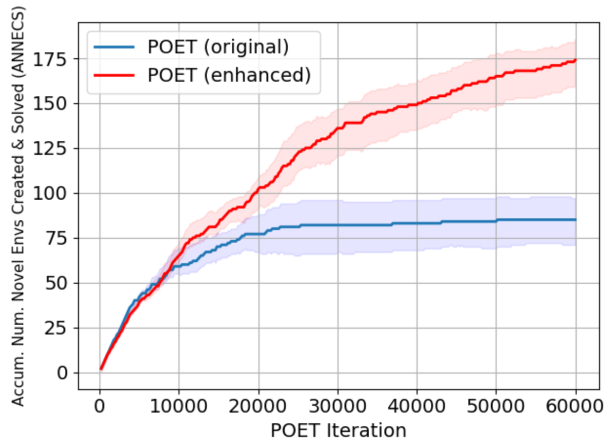


Figure 5. **A comparison of the ANNECS metric across iterations between Enhanced POET and original POET.** Solid lines denotes the median across 5 runs and shading denotes the 95% bootstrapped confidence intervals. Original POET runs gradually lose steam and plateaus after around 20,000 iterations; in sharp contrast, Enhanced POET runs maintain momentum with the ANNECS consistently going up without much sign of slowing down.

7. Discussion, Conclusion, and Future Work

The reason open-endedness is so compelling and so important to the future of machine learning is suggested by an intriguing result in this paper: the very same optimization algorithm, i.e. ES (and PPO too), that *cannot* solve any late-stage environment from POET runs, actually *can* solve them, but only if it is embedded within an open-ended *algorithmic context* (in this case, POET). This result, perhaps counterintuitive at first glance, rests on the insight that we cannot know in advance the stepping stones that must be crossed to reach a far-off achievement. Science's history repeatedly confirms this kind of lesson: Microwaves were invented not by food-heating researchers but by those studying radar; and computers were not invented by optimizing the abacus to increase computations per second, but because scientists invented vacuum tubes and electricity for entirely unrelated purposes (Stanley & Lehman, 2015). Open-ended processes *embrace* this lesson by collecting stepping stones from innumerable *divergent branching* paths through the search space, many climbing towards higher complexity and challenge simultaneously, towards otherwise inconceivable future achievements.

This divergent branching in Enhanced POET is enabled by the newly-introduced PATA-EC, which resonates with "behavior characterizations" aimed at encouraging divergence and exploration, e.g. in novelty search (Lehman & Stanley, 2011a), QD (Pugh et al., 2016) and similarly-oriented work in intrinsic motivation in RL (Oudeyer & Kaplan, 2009; Schmidhuber, 2010; Bellemare et al., 2016). However, unlike previous such characterizations that struggle with the problem of domain generality, PATA-EC is an entirely general characterization, an interesting side-effect of coevolving both environments and agents together. It is precisely because we now have a palette of environments from which to sample that we can begin to construct a profile of behavior (for both environments and agents!) based on their interactions without knowing anything about the inner workings of those environments or agents. Thus, in a sense, the push for divergence in learning (and ultimately, towards open-endedness) becomes fundamentally more tractable when environments are not predefined but instead being learned as agents are being optimized.

Yet despite all these new possibilities, a shadow of doubt lingers at the heart of open-endedness: Because we have no way to know what it may find or what the future of any given run may bring, a skeptic regarding this *uncertainty* might interpret a system like POET as a kind of meandering walk through problem space dangerously close to randomness. Yet the qualitative results conflict with such a pessimistic interpretation – indeed, these agents have gained the ability to traverse extreme irregularity underfoot (reminiscent of dried lava flows near volcanoes), to walk swiftly in efficient alternating bipedal fashion on flat ground, and even to brace remarkably for landing after a fall from great heights. Not only that, but there seems to be no other viable method to learn such skills from scratch. They are not arbitrary skills, but genuinely meaningful, sometimes beyond what we might even expect possible. If we embrace such uncertainty in algorithms that will take us to amazing places, but will not tell us our destinations ahead of time, we might harness the power and reap the rewards of powerful, open-ended search processes.

Finally, how long might it endure? Is even a tiny sliver of the multi-billion-year saga of unfolding life on Earth even conceivable in computation? Looking at Figure 5, though clearly more enduring than its predecessor, the curve for Enhanced POET appears to surrender to slightly more modest growth after 30,000 iterations. Is it petering out, though just more slowly? In fact (as noted also in Section 6), the slower growth in ANNECS is the result of the environments becoming increasingly difficult, and thus each challenge requiring more time to optimize before more can be generated. That is different from a case where there is simply no more room for discovery in the space of the domain itself. Yet even so, it seems inevitable in such a relatively simple world that the time will come where nothing more can be invented that is physically possible to traverse for our agent. The ANNECS curve might be expected to flatline then.

However, that fate is not inevitable by virtue of the algorithm itself. Rather, it seems an artifact of the domain, even when enhanced with CPPNs – somehow, the idea of obstacle courses will succumb to its own finitude in a way that life on Earth has not. There is a sense though in which this realization is exciting – Enhanced POET itself seems prepared to push onward as long as there is ground left to discover. Therefore, an important next step for future work is to apply it to different domains. The *algorithm* is arguably unbounded. If we can conceive a domain without bounds, or at least with bounds beyond our conception, we may now have the possibility to see something far beyond our imagination borne out of computation alone. That is the exciting promise of open-endedness.

Acknowledgements

We thank all of the members of Uber AI Labs, in particular Joost Huizinga, Thomas Miconi, Paul Szerlip, Lawrence Murray, and Jane Hung for helpful discussions. We are grateful to Leon Rosenshein, Joel Snow, Thaxton Beesley, the Colorado Data Center team and the entire Opus Team at Uber for providing our computing platform and for technical support. Finally, we thank the anonymous reviewers for their helpful comments.

References

Anderson, C. W. Learning to control an inverted pendulum using neural networks. *IEEE Control Systems Magazine*, 9(3):31–37, 1989.

Arulkumaran, K., Cully, A., and Togelius, J. Alphastar: An evolutionary computation perspective. *arXiv preprint arXiv:1902.01724*, 2019.

Balduzzi, D., Garnelo, M., Bachrach, Y., Czarnecki, W., Pérolat, J., Jaderberg, M., and Graepel, T. Open-ended learning in symmetric zero-sum games. *ArXiv*, abs/1901.08106, 2019.

Bansal, T., Pachocki, J., Sidor, S., Sutskever, I., and Mordatch, I. Emergent complexity via multi-agent competition. *arXiv preprint arXiv:1710.03748*, 2018.

Bedau, M. Measurement of evolutionary activity. In *Proceedings of the 2nd International Conference on the Synthesis and Simulation of Living Systems (ALIFE 2)*, 1992.

Bedau, M. The arrow of complexity hypothesis (abstract). In Bullock, S., Noble, J., Watson, R., and Bedau, M. (eds.), *Proceedings of the Eleventh International Conference on Artificial Life (Alife XI)*, pp. 750, Cambridge, MA, 2008. MIT Press. URL http://www.alifexi.org/papers/ALIFExi-abstracts-0010.pdf.

Bellemare, M., Naddaf, Y., Veness, J., and Bowling, M. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. Unifying count-based exploration and intrinsic motivation. In *NIPS*, pp. 1471–1479, 2016.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48. ACM, 2009.

Brant, J. C. and Stanley, K. O. Minimal criterion coevolution: A new approach to open-ended search. In *Proceedings of the 2017 on Genetic and Evolutionary Computation Conference (GECCO)*, pp. 67–74, 2017.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Clune, J. AI-GAs: AI-generating algorithms, an alternate paradigm for producing general artificial intelligence. *arXiv preprint arXiv:1905.10985*, 2019.

Cully, A., Clune, J., Tarapore, D., and Mouret, J.-B. Robots that can adapt like animals. *Nature*, 521:503–507, 2015. doi: 10.1038/nature14422.

de Jong, E. D. and Pollack, J. B. Ideal evaluation from coevolution. *Evolutionary Computation*, 12:159–192, 2004.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. IEEE, 2009.

Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., Schulman, J., Sidor, S., Wu, Y., and Zhokhov, P. Openai baselines. *GitHub, GitHub repository*, 2017.

Dolson, E., Vostinar, A., and Ofria, C. What's holding artificial life back from open-ended evolution? *The Winnower*, 04 2018. doi: 10.15200/winn.152418.86598. URL https://dx.doi.org/10.15200/winn.152418.86598.

Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I., and Abbeel, P. Rl 2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

Ecoffet, A., Huizinga, J., Lehman, J., Stanley, K. O., and Clune, J. Go-explore: a new approach for hard-exploration problems. *arXiv preprint arXiv:1901.10995*, 2019.

Ficici, S. and Pollack, J. Challenges in coevolutionary learning: Arms-race dynamics, open-endedness, and mediocre stable states. *Artificial life VI*, pp. 238, 1998.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1126–1135. JMLR. org, 2017.

Florensa, C., Held, D., Wulfmeier, M., Zhang, M., and Abbeel, P. Reverse curriculum generation for reinforcement learning. In *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 482–495, 2017.

Florensa, C., Held, D., Geng, X., and Abbeel, P. Automatic goal generation for reinforcement learning agents. In *International Conference on Machine Learning*, pp. 1514–1523, 2018.

Forestier, S., Mollard, Y., and Oudeyer, P.-Y. Intrinsically motivated goal exploration processes with automatic curriculum learning. *arXiv preprint arXiv:1708.02190*, 2017.

Gomez, F. and Miikkulainen, R. Incremental evolution of complex general behavior. *Adaptive Behavior*, 5(3-4): 317–342, 1997.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Heess, N., Sriram, S., Lemmon, J., Merel, J., Wayne, G., Tassa, Y., Erez, T., Wang, Z., Eslami, S., Riedmiller, M., et al. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*, 2017.

Jaderberg, M., Dalibard, V., Osindero, S., Czarnecki, W. M., Donahue, J., Razavi, A., Vinyals, O., Green, T., Dunning, I., Simonyan, K., et al. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017.

Justesen, N., Torrado, R. R., Bontrager, P., Khalifa, A., Togelius, J., and Risi, S. Illuminating generalization in deep reinforcement learning through procedural level generation. *arXiv preprint arXiv:1806.10729*, 2018.

Karpathy, A. and Van De Panne, M. Curriculum learning for motor skills. In *Canadian Conference on Artificial Intelligence*, pp. 325–330. Springer, 2012.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Klimov, O. Bipedalwalkerhardcore-v2. https://gym.openai.com, 2016.

Langdon, W. B. Pfeiffer – A distributed open-ended evolutionary system. In Edmonds, B., Gilbert, N., Gustafson, S., Hales, D., and Krasnogor, N. (eds.), *AISB'05: Proceedings of the Joint Symposium on Socially Inspired Computing (METAS 2005)*, pp. 7–13, 2005.

URL http://www.cs.ucl.ac.uk/staff/W.Langdon/ftp/papers/wbl_metas2005.pdf.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

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

Lehman, J. and Stanley, K. O. Evolving a diversity of virtual creatures through novelty search and local competition. In *GECCO '11: Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pp. 211–218, 2011b.

Lenski, R. E., Ofria, C., Pennock, R. T., and Adami, C. The evolutionary origin of complex features. *Nature*, 423 (6936):139, 2003.

Matiisen, T., Oliver, A., Cohen, T., and Schulman, J. Teacher-student curriculum learning. *arXiv preprint arXiv:1707.00183*, 2017.

McIntyre, A., Kallada, M., Miguel, C. G., and da Silva, C. F. neat-python. https://github.com/CodeReclaimers/neat-python.

Mouret, J. and Clune, J. Illuminating search spaces by mapping elites. *ArXiv e-prints*, abs/1504.04909, 2015. URL http://arxiv.org/abs/1504.04909.

Nguyen, A., Yosinski, J., and Clune, J. Understanding innovation engines: Automated creativity and improved stochastic optimization via deep learning. *Evolutionary computation*, 24(3):545–572, 2016.

OpenAI, Berner, C., Brockman, G., Chan, B., Cheung, V., Debiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., Józefowicz, R., Gray, S., Olsson, C., Pachocki, J., Petrov, M., de Oliveira Pinto, H. P., Raiman, J., Salimans, T., Schlatter, J., Schneider, J., Sidor, S., Sutskever, I., Tang, J., Wolski, F., and Zhang, S. Dota 2 with large scale deep reinforcement learning. 2019. URL https://arxiv.org/abs/1912.06680.

Oudeyer, P.-Y. and Kaplan, F. What is intrinsic motivation? a typology of computational approaches. *Frontiers in Neurorobotics*, 1:6, 2009.

Popovici, E., Bucci, A., Wiegand, R. P., and De Jong, E. D. *Coevolutionary Principles*, pp. 987–1033. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-540-92910-9. doi: 10.1007/978-3-540-92910-9_31. URL http://dx.doi.org/10.1007/978-3-540-92910-9_31.

Pugh, J. K., Soros, L. B., and Stanley, K. O. Quality diversity: A new frontier for evolutionary computation. 3(40), 2016. ISSN 2296-9144.

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

Samuel, A. L. Some studies in machine learning using the game of checkers. ii—recent progress. *IBM Journal of research and development*, 11(6):601–617, 1967.

Schmidhuber, J. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010.

Schmidhuber, J. POWERPLAY: Training an increasingly general problem solver by continually searching for the simplest still unsolvable problem. *Frontiers in psychology*, 4:313, 2013.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Shaker, N., Togelius, J., and Nelson, M. J. *Procedural content generation in games*. Springer, 2016.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

Soros, L. and Stanley, K. O. Identifying necessary conditions for open-ended evolution through the artificial life world of chromaria. In *ALIFE 14: The Fourteenth International Conference on the Synthesis and Simulation of Living Systems*, pp. 793–800, 2014.

Standish, R. K. Open-ended artificial evolution. *International Journal of Computational Intelligence and Applications*, 3(02):167–175, 2003.

Stanley, K. O. Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 8(2):131–162, 2007.

Stanley, K. O. and Lehman, J. Why greatness cannot be planned. 2015.

Stanley, K. O. and Miikkulainen, R. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 2002. URL http://nn.cs.utexas.edu/?stanley:ec02.

Stanley, K. O., Lehman, J., and Soros, L. Open-endedness: The last grand challenge you've never heard of. *O'Reilly Online*, 2017.

Stanley, K. O., Clune, J., Lehman, J., and Miikkulainen, R. Designing neural networks through neuroevolution. *Nature Machine Intelligence*, 1(1):24–35, 2019.

Taylor, T., Bedau, M., Channon, A., Ackley, D., Banzhaf, W., Beslon, G., Dolson, E., Froese, T., Hickinbotham, S., Ikegami, T., McMullin, B., Packard, N., Rasmussen, S., Virgo, N., Agmon, E., Clark, E., McGregor, S., Ofria, C., Ropella, G., Spector, L., Stanley, K. O., Stanton, A., Timperley, C., Vostinar, A., and Wiser, M. Open-ended evolution: Perspectives from the OEE workshop in York. *Artificial life*, 22(3):408–423, 2016.

Togelius, J., Yannakakis, G. N., Stanley, K. O., and Browne, C. Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3):172–186, 2011.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575 (7782):350–354, 2019.

Wang, J. X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J. Z., Munos, R., Blundell, C., Kumaran, D., and Botvinick, M. Learning to reinforcement learn. *CoRR*, abs/1611.05763, 2016. URL http://arxiv.org/abs/1611.05763.

Wang, R., Lehman, J., Clune, J., and Stanley, K. O. Paired Open-Ended Trailblazer (POET): Endlessly generating increasingly complex and diverse learning environments and their solutions. *arXiv preprint arXiv:1901.01753*, 2019a.

Wang, R., Lehman, J., Clune, J., and Stanley, K. O. POET: open-ended coevolution of environments and their optimized solutions. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 142–151. ACM, 2019b.

Wiegand, R. P., Liles, W. C., and Jong, K. A. D. An empirical analysis of collaboration methods in cooperative coevolutionary algorithms. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, GECCO'01, pp. 1235–1242, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-774-9. URL http://dl.acm.org/citation.cfm?id=2955239.2955458.

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

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pp. 3320–3328, 2014.