Curious POET: Intrinsic Motivation Improves Exploration Efficiency

Robert Mash[1], Gregory Castañón[1], Jared Culbertson[2]
[1]Systems and Technology Research
[2]ACT3 - Air Force Research Laboratory
robert.mash@str.us, gregory.castanon@str.us, jared.culbertson@afrl.af.mil

Abstract

Paired Open-Ended Trailblazer (POET) and its variants represent the state of the art in auto-curriculum generation wherein environments are co-evolved with agents to simultaneously explore the space of possible problems and their solutions. However, we observe that distinct POET agents often explore similar behavior spaces. To address this, we present Curious POET, in which an intrinsically curious oracle tracks an evolving Enhanced POET (ePOET) population and rewards agents for novel behavior, leading to more efficient behavior exploration. To fairly evaluate agent populations, we introduce a training-independent strategy for environment generation and define a coverage metric over these environments. We demonstrate our approach on the enhanced Bipedal Walker environment and find that Curious POET outperforms ePOET at environment coverage and population cross-evaluation. Our study explores how a curious oracle can bias individual agent evolution in such a way as to speed up behavioral exploration at the population level. Our implementation is available at `https://github.com/act3-ace/Curious-POET`.

1 Introduction

Open-ended learning, as realized in POET (Wang et al., 2019) and developed further in ePOET (Wang et al., 2020), is an effective method for simultaneously evolving agent and environment populations. Open-ended learning has been shown to be effective in 2D and 3D (Zhou and Vanschoren, 2022) robotic proprioceptive environments, as well as Atari/Zelda-like map games (Dharna et al., 2020, 2022). POET and its derivatives present an elegant alternative to traditional curricular approaches in reinforcement learning of explicit curricula and hand-crafted rewards.

However, a potential problem with this class of algorithms is a lack of mechanisms beyond environment selection to drive diversity in behavioral exploration. Consequently, agents in an evolving POET population are likely to co-explore some of the same behavioral strategies as they evolve independently. To explore this problem space, we introduce an intrinsically curious oracle that individually alters the agents' fitness functions to incentivize behaviors that are novel to the population. We attempt to illustrate this conceptually in Figure 1. Formed around the Intrinsic Curiosity Module (ICM) from Pathak et al. (2017), the curious oracle is trained to predict the actions and next-state-embeddings of the evolving population of Curious POET agents. In addition to extrinsic reward from the environment, Curious POET agents receive intrinsic reward for exhibiting behavior the oracle cannot predict, leading to a more diverse population of agents. We contend that the effect of the curious oracle is to mitigate redundant behavioral exploration in a population of otherwise independently evolving agents. We compare evolved populations with varying amounts of intrinsic reward by performing cross-run evaluation of agent and environment populations. Additionally, to compare evolved agent populations efficiently, we propose a method for randomly sampling environments from the space of realizable environments given a generation approach. We use this method to generate an independent test set of environments and define a coverage metric for an agent population over that set of environments. This metric rewards populations creating diverse, specialized agents that solve many different environments as opposed to similar agents which solve a few.

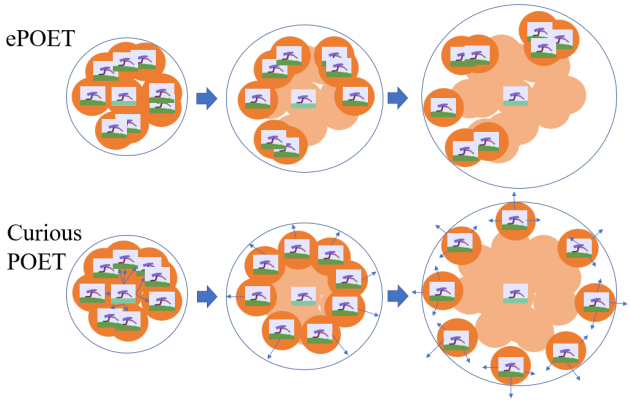

Figure 1: Curious Poet's population-aware curious oracle directly incentivizes agents (orange circles) to prefer novel behaviors resulting in enhanced behavior space (white background) exploration efficiency when compared to ePOET.

We present the following contributions:

- Curious POET, an approach to open-ended learning which employs a curious oracle to evolve diverse agent populations which outperform ePOET agent populations on multiple performance measures.
- A method for sampling independent environments from the space of potential environments visited by an open-ended learning process and a coverage metric over these environments that rewards agent specialization.

2 Background & Related Work

2.1 Open-Ended Learning

In open-ended learning, agents are typically realized as neural networks, their weights and possibly architecture represented by a genome, as selected and passed down to children via an evolutionary process as in Salimans et al. (2017). There are various strategies for inducing genetic diversity in a population; sexual pairing and reproduction, random genetic mutation, etc. The neural architecture of agents can be fixed, or can also evolve. Neural Evolution of Augmenting Topologies (NEAT) (Stanley and Miikkulainen, 2002) is an important development enabling not just the weights of a neural network architecture to evolve, but also its structure. A Compositional Pattern Producing Network (CPPN) is a population implementation of NEAT used to breed pictures in PicBreeder (Secretan, 2011), to realize an environment genome in ePOET (Wang et al., 2020), and the current work.

Soros and Stanley (2014) explore convergent population stagnation, which can result in a single species dominating an evolving population. Novelty search (Lehman and Stanley, 2011b) attempts to address this problem by redefining fitness as behavioral novelty rather than an explicitly defined environmental objective. This is the conceptual beginning of open-ended learning as the evolutionary progress of a population is decoupled from any defined objective short of novelty. Quality Diversity (QD) algorithms such as Lehman and Stanley (2011a); Mouret and Clune (2015) seek to manage and promote behavioral diversity by keeping track of multiple niches and agents and swapping the niche-agent pairing on some basis. Go-explore (Ecoffet et al., 2019) is a notable variation on QD with extremely sparse rewards. Another important development is Multi-Criterion Coevolution (MCC) (Brant and Stanley, 2017) which evolves both environmental niches as well as agent populations by imbuing environments with a defining genome which can itself be evolved. MCC allows the difficulty of a task or environment to evolve in complexity beyond a fixed environment enabling open ended exploration rather than limiting a population of agents to finding the best solution to a fixed problem.

Wang et al. (2019), integrates these ideas together in the POET algorithm, which co-evolves a population of paired environments and agents using MCC. Also, POET represents a significant philosophical rethinking of objectives, specifically in terms of rejecting objectives altogether as in Lehman (2011) and Lehman (2008). Wang et al. (2019) describes the POET evolutionary process as automatically discovering "stepping stones" in terms of agent behavior, which we submit, is equivalent to discovery of trajectories through behavior space, or automatic curriculum discovery. POET has been adapted to map-type environments such as dZelda: (Dharna et al., 2020, 2022) and robotic traversal over terrain3D (Zhou and Vanschoren, 2022). Nasir et al. (2022) have extended ePOET by instantiating not only the environment topology with a CPPN, but the agent architecture as well. Curious POET is based on the ePOET architecture, which we refer to as "baseline," and proceed to augment with a centralized form of intrinsic motivation.

2.2 Intrinsic motivation

Intrinsic motivation, considered a subset of a broader mammalian motivation system (White, 1959), is described by Barto (Barto et al., 2004) in these terms:

> An agent's activity is said to be intrinsically motivated if the agent engages in it for its own sake rather than as a step toward solving a specific problem.

Silver et al. (2021) suggests that in the context of reinforcement learning, hand-crafted extrinsic rewards are insufficient for an agent to develop broadly applicable skills useful for solving a diverse set of problems encountered later in life. There have been many works in the field of intrinsic motivation, largely along the lines of recognizing when a novel experience is happening and rewarding an agent according to how novel the experience is. Burda et al. (2019a) provide an excellent summary of prediction error, prediction uncertainty and model refinement, all of which realize a non-stationary intrinsic reward as learning occurs over experiential time.

Note that there is an interesting parallel with novelty search above, in that novelty search can be viewed as a population level intrinsic motivation. Indeed, this is the central thesis of this work: **explicitly rewarding individuals in a POET population with intrinsic motivation directly induces a broader variety of agent behaviors.** The ICM was introduced in Pathak et al. (2017) and evaluated in the Mario and VizDoom environments. Later the ICM and other intrinsic reward architectures were more fully explored in Burda et al. (2019a) in a suite of Atari environments. In the next section we detail the operation of the ICM in the context of Curious POET, as we chose the ICM architecture to realize intrinsic motivation, or "curiosity," in Curious POET's oracle. We suspect, however, that other novelty-based methods such as Random Network Distillation (RND) (Burda et al., 2019b) or prediction-error-based methods such

as Bring Your Own Latent-Explore (BYOL-Explore) (Guo, 2022) could be used to similar effect.

3 Methods

In this section we review the POET algorithm, the intrinsic curiosity module, and the integration of the two, forming Curious POET. Also, we describe the formulation of a population coverage metric used to compare the performance of agent populations on a set of freely evolved environments.

3.1 POET and ePOET

POET (Wang et al., 2019) and its updated version Enhanced POET (ePOET) (Wang et al., 2020) are multiple-criteria co-evolution (Brant and Stanley, 2017) algorithms that co-evolve a population of agent-environment pairs, where agents train on a paired environment. Environments that are either too challenging or too easy for the current agent population are discarded. They also employ a quality diversity mechanism to transfer agent genetic information across the agent population when one agent's skill is useful in another agent's environment. This continuing balance of increasing agent skill and increasingly difficult environments is an example of the open-ended discovery (Stanley et al., 2017; Stanley, 2019; Lehman, 2008). In this context, patterns of co-evolved environments represent a discovered curriculum of stepping stones for continuous agent improvement in terms of a fitness function, which in POET and ePOET are simply the extrinsic reward returned from an environment. In Curious POET, we build on ePOET by augmenting the extrinsic reward from the environment with intrinsic motivation, the effect of which is an increase in exploration efficiency over that of ePOET.

Enhanced POET is organized around a repeating three-step process called an ePOET iteration:

1. **Generate new environments.** Given an ePOET population of size M consisting of a set of agent-environment pairs $\{(\theta_m, E_m)\}_{1\leq m\leq M}$, each environment is mutated a fixed number K times using the NEAT evolutionary mechanism (Stanley and Miikkulainen, 2002), resulting in a set of potential child environments $\mathcal{E}_m^c = \{E_m^{c_i}\}_{1\leq i\leq K}$ for each $m \in \{1, .., M\}$. Next, each agent[1] θ_m in the set of active agents Θ is evaluated on each $E_n^{c_i}$ evolved from E_n in the set of active environments $\mathcal{E}$ and the scores are compared to a set of MCC thresholds ρ_h and ρ_e, representing minimum and maximum scores that the population as a whole must achieve. Fitness in ePOET, $f_E(\theta)$, is defined for a given agent $\theta \in \Theta$ and environment $E \in \mathcal{E}$ as the non-discounted sum of individual extrinsic rewards $R^e(t)$ (Wang et al., 2020) over the length T of an episode:

$$f_E(\theta) = \sum_{t=0}^{T-1} R^e(t). \tag{1}$$

If there exists $\theta_m \in \Theta$ such that $\rho_h \leq f_{E_n^{c_i}}(\theta_m) \leq \rho_e$, then $E_n^{c_i}$ is neither too easy nor too hard and it is placed in a list of potential child environments ranked by novelty according to the PATA-EC measure (Wang et al., 2020). The top ranking child environments for each $n \in \{1, .., M\}$ is added to $\mathcal{E}$ and the oldest environments are removed to keep the size of $\mathcal{E}$ below a fixed maximum number of environments.

2. **Optimize paired agents within their respective environments.** To optimize each agent θ_m within its paired environment E_m, POET generates a population $\overline{\Theta}_m$ of size S of mutated parameter vectors for θ_m by adding Gaussian noise $\epsilon_s \sim \mathcal{N}(0, I)$ to an agent's parameter vector θ_m as $\theta_m^s = \theta_m + \sigma\epsilon_s$. We then compute the fitness $f_{E_m}(\theta_m^s)$ for each member of this population $\theta_m^s \in \overline{\Theta}_m$ with respect to θ_m's paired environment E_m and average over the population to compute the fitness for a mutated population $\overline{\Theta}_m$ as

$$J(\overline{\Theta}_m) = \frac{1}{S}\sum_{s=1}^{S} f_{E_m}(\theta_m^s). \tag{2}$$

Then the approximate gradient of fitness $J(\overline{\Theta}_m)$ with respect to the parameter vector θ_m is

$$\nabla_{\theta_m} J(\overline{\Theta}_m) \approx \frac{1}{S\sigma}\sum_{j=1}^{S} f_{E_m}(\theta_m^s)\epsilon_s. \tag{3}$$

Finally, the agent's weight vector θ_m is updated according to the learning rate α:

$$\theta_m = \theta_m + \alpha\nabla_{\theta_m} J(\overline{\Theta}_m). \tag{4}$$

3. **Attempt agent transfers.** In order to mitigate situations where agents are stuck in local minima while training in their paired environments, each iteration POET assesses whether for each environment another agent performs better than the environment's current agent. For example, if $f_{E_n}(\theta_m) > f_{E_n}(\theta_n)$, then the parameter vector for agent θ_n is overwritten by agent θ_m. This process is referred to in Wang et al. (2019) as cross-pollination; wherein progress in one environment can end up helping in another environment. We leave this portion of the POET algorithm unmodified.

[1]Note that θ_m represents the parameter vector associated with the agent m, but for ease of notation we also refer to the agent itself as θ_m.

3.2 Curious POET

We extend the Enhanced POET architecture with a curious oracle realized with an ICM. The ICM observes the POET agent population and injects intrinsic motivation into POET's agent optimization step in order to bias individual agents' exploration away from behaviors it can predict using the existing population. The objective of the ICM is to deconflict agent behaviors, biasing evolutionary pressure toward a diverse population of specialists rather than a homogenous population of generalist agents.

We implemented the Intrinsic Curiosity Module (ICM) as a shared resource—an oracle. This choice enables population awareness and encourages agents to explore beyond the behavioral origin. Alternative architectures such as equipping each agent with its own ICM are also possible, but this would limit their awareness to their own (or perhaps their ancestors') experiences. Alternatively, ICM-equipped agents could observe their peers' behaviors, but this would lead to each agents' ICM redundantly observing and learning the same behaviors. For the sake of computational efficiency, we chose to train a single ICM as an oracle, effectively allowing all members of the population to observe and learn from each other, facilitating the acquisition of shared knowledge.

Intrinsic Curiosity Module In a reinforcement learning or agent evolution setting, the ICM (Pathak et al., 2017) takes as input tuples of state, action, next-state (s_t, a_t, s_{t+1}) from an agent episode of length T, and computes an intrinsic reward R_t^i at each time-step t. This intrinsic reward R_t^i can then be used (Pathak et al., 2017; Burda et al., 2019a) in combination with extrinsic reward R_t^e from the environment E to bias the agent toward behavioral exploration while optimizing for maximum performance in the environment E.

Internally, the intrinsic curiosity module, shown in Figure 2, is composed of three neural networks trained via self-supervised learning: an embedding network ϕ, an inverse dynamics model g, and a forward dynamics model f. First, ϕ computes feature embeddings $\phi(s_t)$ and $\phi(s_{t+1})$ from states s_t and s_{t+1}. The inverse dynamics model g takes these embeddings $\phi(s_t)$ and $\phi(s_{t+1})$ as input and predicts the action a_t that caused, or at least accompanied, a transition from state s_t to state s_{t+1}. The forward model f takes as input the current action a_t and embedding of the current state $\phi(s_t)$ and predicts the embedding of the next state $\phi(s_{t+1})$.

Loss terms L_I and L_F for self-supervised training of the ICM seek to minimize prediction error of current action $\hat{a}_t$ by the inverse dynamics model g and the embedding of next state $\hat{\phi}(s_{t+1})$ by the forward model f.

$$L_I = \frac{1}{N_a} \sum_{i=0}^{N_a - 1} (a_i - \hat{a}_i)^2, \tag{5}$$

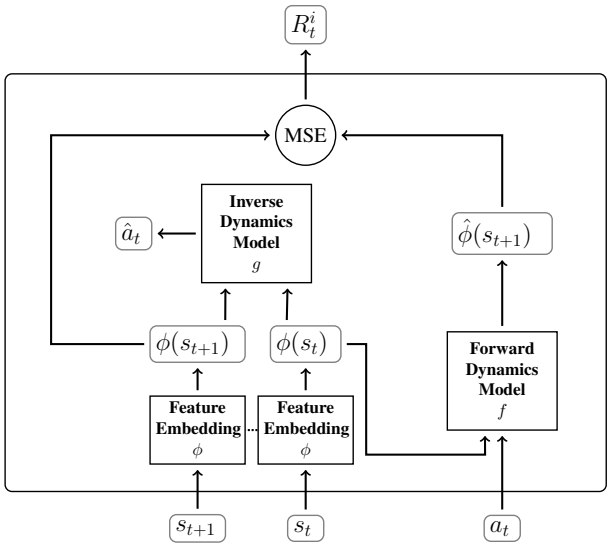

Figure 2: The Intrinsic Curiosity module learns an embedding which supports both an inverse dynamics model to predict the action that causes the state transition from s_t to s_{t+1}, as well as a forward model to predict the next embedded state $\hat{\phi}(s_{t+1})$. We use the ICM's prediction error to reward novel behavior.

and

$$L_F = \frac{1}{N_\phi} \sum_{i=0}^{N_\phi - 1} (\phi(s_i) - \hat{\phi}(s_i))^2, \tag{6}$$

where N_a is the length of the action vector a and N_ϕ is the length of the embedding vector $\phi(s)$. The overall loss term used to optimize the curious oracle's ICM is specified as

$$L_{ICM} = (1 - \beta) L_I + \beta (L_F), \tag{7}$$

where we chose $\beta = 0.5$ for Curious POET, although other values may be advantageous.

Note that there is no reconstruction objective. Rather, the embedding is task oriented or agent-centric, only concerned with embedding into a feature vector $\phi(s_t)$ those aspects of the state s_t that are relevant for predicting actions a_t, and the next state embedding $\phi(s_{t+1})$, ignoring all other aspects of the state s_t. The argument presented in Pathak et al. (2017) is that this architecture avoids a curious agent getting stuck in a "curiosity trap," forever observing some meaningless stochastic process, e.g., static on a television.

Integration with POET In step two of the ePOET algorithm described in Section 3.1 above, each agent θ_m is optimized in its paired environments E_m. During evaluation of candidate child agent θ_m^s in the parent agent's paired environment E_m, episode rollouts are collected as a batch, and transmitted to the curious oracle. Here a rollout batch B_m is defined as a set of S trajectories of length T sequences of tuples (s_t, a_t), where the batch size S is the size of $\overline{\Theta}_m$ (the

Algorithm 1 ICM Training & Inference

Require:
ICM model
Rollout store R
Observation Normalization Buffer N_o
Rewards Normalization Buffer N_r
while Loop Forever **do**
`// Train ICM`
$B_{train} \leftarrow sample(R)$ `// sample batch`
Update N_o with B_{train}
$B_{train} \leftarrow StackObs((B_{train} - N_o^{\mu})/N_o^{\sigma}, 4)$
Train ICM on B_{train} `// ICM weight update`

`// Compute intrinsic reward`
Receive new rollout batch B_{new} from POET node
$R \leftarrow R + B_{new}$ `// add new batch to store`
$B_{new} \leftarrow StackObs((B_{new} - N_o^{\mu})/N_o^{\sigma}, 4)$
$R_i = ICM(B_{new})$ `// Infer` R_i
Update N_r with R_i
$R_i \leftarrow R_i/N_r^{\sigma}$
Transmit batch R_i to POET node
end while

set of candidate children agents of θ_m) and the dimension T is zero-padded to the length of the longest trajectory in the batch.

Rollouts B_m are transmitted to the curious oracle, wherein they are saved into a rollout buffer. The curious oracle performs two tasks in alternation:

1. Performs ICM forward model inference on incoming rollouts and transmits intrinsic reward R^i back to the POET trainer node.

2. Samples rollout buffer and performs self-supervised training of ICM.

Here we augment Equation (1) of the ePOET algorithm to include both extrinsic reward R^e from the environment and intrinsic reward R^i from the curious oracle:

$$f_E(\theta) = \sum_{t=0}^{T-1} R^e(t) + \zeta R^i(t). \tag{8}$$

where ζ is the intrinsic motivation coefficient.

We apply the learned lessons in Burda et al. (2019a) to realize a practical ICM trained and operated according to Algorithm 1.

3.3 Population Coverage Metric

As discussed in Wang et al. (2019), it is generally challenging to measure performance in open-endedness due to the inherent lack of objective. Wang et al. (2019) proposes the Accumulated Number of Novel Environments Created and Solved (ANNECS) measure as a reasonable way to track progress in open-ended exploration by counting the number of unique environments created and solved by ePOET. However, ANNECS does not measure the quality of a population of agents; it only measures how many unique environments were solved, not how difficult or different they were from each other. It is often not clear if a high ANNECS score indicates a population of agents that has a higher likelihood of solving a given environment. To address this question, we present a coverage metric that leverages environments generated by the environment evolution mechanism in the ePOET framework. Enhanced POET's Generate New Environments step in Section 3.1 utilizes MCC to prevent candidate child environments from being either too easy or hard for the current agent population. This step filters environments to produce bespoke environments for the existing agent population; environments are chosen to be not-too-hard and not-too-easy, driving the agent solve rate to the middle by definition. Our environment cover generator removes this filter; we do not prune environments that are not-too-hard or not-too-easy for the existing agent population. Experimentally we find that a small collection of 100 environments from freely evolved sequences comprise an effective cover of those environments selected by the MCC filter and used in training an ePOET population. (See `https://github.com/act3-ace/Curious-POET` for additional covering set selection and analysis.)

More formally, given a population Θ of M ePOET agents and a set $\mathcal{E}_{\text{cov}}$ of N covering environments, the coverage metric is computed as

$$S_{\mathcal{E}_{\text{cov}}}(\Theta) = \frac{|\{E \in \mathcal{E}_{\text{cov}} \mid \exists \theta \in \Theta, f_E(\theta) > \rho\}|}{N}, \tag{9}$$

where $|\cdot|$ denotes the size of the set and ρ is a domain specific threshold score. This provides us with a metric for evaluating the performance of a population of agents that is independent of any particular population of agent-environment pairs, allowing for fair cross-population comparisons.

3.4 Cross Evaluation

Cross evaluation of evolved agents and environments was previously performed in Wang et al. (2020), but only between co-evolved agents and environments from one POET run. We perform comprehensive cross evaluation of agents and environments across multiple POET runs in order to evaluate the aggregate effect of intrinsic motivation on population evolution in terms of both agent capability and environment difficulty. We perform pairwise cross evaluation of two ePOET populations of sizes M and N consisting of sets of agent-environment pairs $\{(\theta_m^{\zeta_a}, E_m^{\zeta_a})\}_{1 \le m \le M}$ and $\{(\theta_n^{\zeta_e}, E_n^{\zeta_e})\}_{1 \le n \le N}$, with intrinsic motivation coefficients ζ_a and ζ_e for agent and environment respectively. Then the cross pairs of environments $\mathcal{E}^{\zeta_a}$ and agents Θ^{ζ_e}, along with the counterparts Θ^{ζ_a} and $\mathcal{E}^{\zeta_e}$ are used for cross evaluation.

For simplicity, we refer to these pairs of sets generically as Θ and $\mathcal{E}$. We compute cross evaluation score

$$S_{\mathcal{E}(\Theta)} = \frac{|\{E \in \mathcal{E} \mid \exists \theta \in \Theta \text{ with } g_E(\theta) > \rho_{CE}\}|}{N}, \quad (10)$$

where $g_E(\theta)$ is the proportion of environments solved over the evaluation seeds (10 seeds were used in this experiment and solving means that $f_E(\theta) > 230$), we use $|\cdot|$ to denote the size of the set, and $\rho_{CE} = 0.9$.

As shown in Figure 3, we compute and plot the means and standard deviations over the sets of scores $S_{\mathcal{E}(\Theta)}$ represented by each pair of ζ_{agent} and $\zeta_{environment}$.

4 Experimental Setup

We compare Curious POET to an Enhanced POET (ePOET) baseline in the bipedal walker environment. Baseline populations are evolved using only extrinsic reward from the bipedal walker environment as in Equation (8) with intrinsic motivation coefficient $\zeta = 0$. We train each population for 2000 ePOET iterations, with the maximum active population size set to ten. As in ePOET, the oldest agents are removed to the archived population as the agent population grows. We see populations typically grow to around 18-20 total individuals.

In our main experiment, we independently train at least three and as many as six Curious POET populations for each intrinsic motivation coefficient ζ equal to $0.0, 5.0, 7.5, 10.0, 15.0$ and 20.0 (non-uniform number of populations due to computational constraints). We artificially set ζ equal to 0.0 for the first 200 Curious POET iterations in order to allow ICM training to stabilize.

We then evaluate both enhanced POET and Curious POET populations using both the cross-population and population coverage metric techniques as described in Methods. We use the same ePOET hyperparameters for Curious POET as for baseline ePOET. These hyperparameters were derived mainly from the ePOET codebase (`https://github.com/uber-research/poet`) and are not optimized for Curious POET in any way.

5 Results and Discussion

In order to understand the population-level effect of a curious oracle's influence on individual agents, we take a two-pronged approach.

First we cross-evaluate agents and environments from populations evolved with varying levels of curiosity to assess their relative performance. Then we evaluate the same populations against one another independently using a coverage metric.

5.1 Cross Population Evaluation

We comprehensively cross evaluate each agent population against each environment population. As in eq. (8), each Curious POET population utilizes a single ζ coefficent.

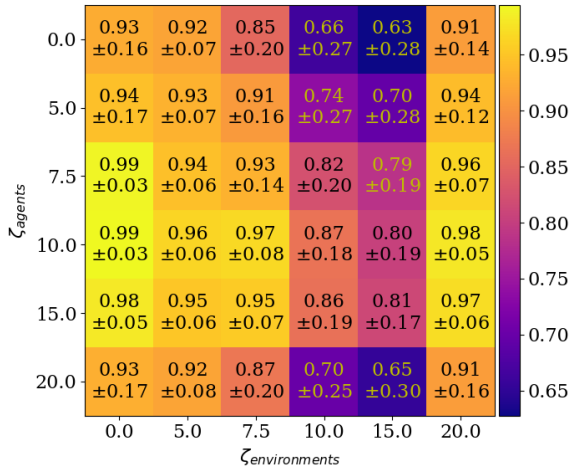

Figure 3: Cross evaluation scores of POET agent populations vs co-evolved environment populations ($\uparrow \pm \downarrow$ is better). Indicated scores are the fraction of environment populations solvable by any agent in an agent population in at least 90% of evaluation seeds. Agent performance peaks at $\zeta_{agents} \approx 10$ which corresponds to more difficult environments (indicated by lower scores) at $\zeta_{environments} \approx 10$.

However, for clarity of analysis we refer to Curious POET agent populations and their associated environments according to their intrinsic motivation coefficient ζ_{agents} and $\zeta_{environments}$.

Intuitively, we might expect that performance would be highest when a population of agents is evaluated against their co-evolved environments since the evolutionary process for the agents is guided by their fitness in these paired environments. Instead, we find that setting the curiosity parameter appropriately (in this case $\zeta \approx 10$) leads to near-universally more capable agents. This is reflected in Figure 3, where we see higher values in the row $\zeta_{agents} = 10$ compared to other values in each column (except the $\zeta_{environments} = 15$ column, where the $\zeta_{agents} = 15$ populations score slightly higher).

We also observe that **co-evolving agents and environments tightly couples agent performance with environment difficulty**. Because performant agents are able to tackle more difficult obstacles, they tend to make their paired environments more difficult for all populations of agents, as reflected by lower agent performance scores. These trends are apparent in Figure 3. Note the horizontal region of higher agent scores centered around $\zeta_{agents} \approx 10$ (comparing to other values in each column) corresponds to reduced agent scores centered around $\zeta_{environments} \approx 10$ for all populations. In other words, the bright horizontal band at $\zeta_{agents} \in \{10, 15\}$ *causes* the dark vertical bars at $\zeta_{environments} \in \{10, 15\}$. It appears that, at least for bipedal walker, $\zeta = 10$ strikes an optimal balance between exploration and exploitation, resulting in a set of diverse *and*

Agent ζ	Mean Score over all Environments $\mu \pm \sigma$
0.0	0.81 ± 0.24
5.0	0.86 ± 0.23
7.5	0.91 ± 0.16
10.0	$\mathbf{0.93 \pm 0.14}$
15.0	0.92 ± 0.13
20.0	0.83 ± 0.24

Table 1: Agent population cross evaluation performance vs intrinsic motivation coefficient ζ. Agent populations with $\zeta = 10$ score 15% higher than baseline ePOET. Notably, smaller standard deviations correspond with higher scores.

useful behaviors. Too low ζ values, as in ePOET, tend toward useful, but less diverse behaviors, resulting in poorer agent performance for a given investment of training iterations. While too high ζ values tend to over-prioritize behavioral diversity over usefulness resulting in reduced agent performance. In the bipedal walker environment we observe that with too much curiosity, agents are rewarded for moving their appendages in novel ways, but don't necessarily learn to stand and walk and thereby progress past the initial "flat" environment. Consequently, the resulting agents are relatively unskilled and their paired, co-evolved environments are relatively easy to solve. A striking feature in the first column of Figure 3 is that Curious POET agents with $\zeta_{agents} \in \{5, 7.5, 10, 15\}$ achieve higher scores on ePOET environments ($\zeta_{environments} = 0$) than do the ePOET agents ($\zeta_{agents} = 0$) paired with, and specialized in, those environments.

Table 1 summarizes agent cross evaluation performance as a function of intrinsic motivation coefficient ζ. Peak performance (0.93) occurs at $\zeta = 10$, representing a 15% increase over ePOET (0.81). Notably, variation in agent performance (across repeated Curious POET seeds) reaches a minimum as mean performance reaches a maximum at $\zeta \approx 10$. We hypothesize that the space of useful behaviors as represented conceptually in Figure 1 is more efficiently explored and therefore more likely to be usefully covered by agent populations with greater behavioral diversity.

An issue with comprehensive cross evaluation is $O(N^2)$ quadratic complexity. Consequently, we evaluate and compare Curious POET to ePOET using an alternative coverage metric designed to be both independent of any particular population and that scales with $O(N)$ linear complexity.

5.2 Coverage Metric Evaluation

We now utilize the coverage metric introduced in Section 3.3 to independently evaluate populations of agents on a covering set of environments. Figure 4 and Table 2 detail performance of Curious POET agent populations as a function of intrinsic motivation coefficient ζ. Agent population performance peaks at $\zeta = 10.0$ with a 29.6% improvement over the ePOET baseline represented by $\zeta = 0$. This result is consistent with cross population evaluation in Section 5.1 in that mean agent population performance increases as variance decreases for $7.5 \leq \zeta \leq 15.0$. Using the Mann Whitney U-test to compute statistical significance, we report $p = 0.045$ for the case $\zeta = 10.0$ with $N = 6$. As illustrated in Figure 5, "curious" populations tend to solve not only the environments solved by baseline ePOET populations ("both solved"), but also go on to solve an additional set of environments ("Curious POET Solved") that baseline populations fail to solve.

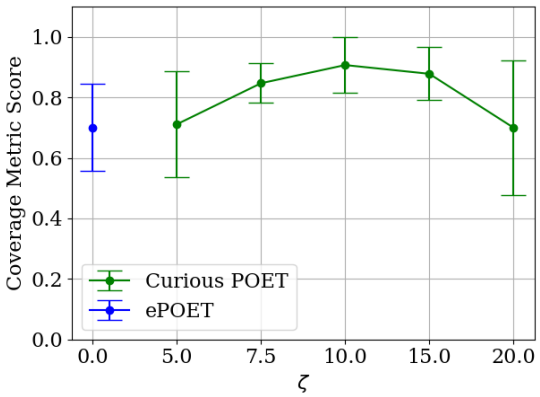

Figure 4: Coverage metric score statistics vs intrinsic motivation coefficient ζ at POET iteration 2000, over the covering set.

Intrinsic Motivation Coefficient ζ	Coverage Metric Score $\mu \pm \sigma$
0.0	70.0 ± 14.4
5.0	71.1 ± 17.5
7.5	84.7 ± 6.5
10.0	$\mathbf{90.7 \pm 9.3}$
15.0	87.8 ± 8.7
20.0	70.0 ± 22.2

Table 2: Coverage metric score statistics vs intrinsic motivation coefficient ζ at POET iteration 2000, over the covering set.

5.3 Discussion

We argue that a central curious oracle tends to bias individual agent evolution in such a way as to speed up behavioral exploration at the population level. To further support this argument, we extend training of an ePOET population and

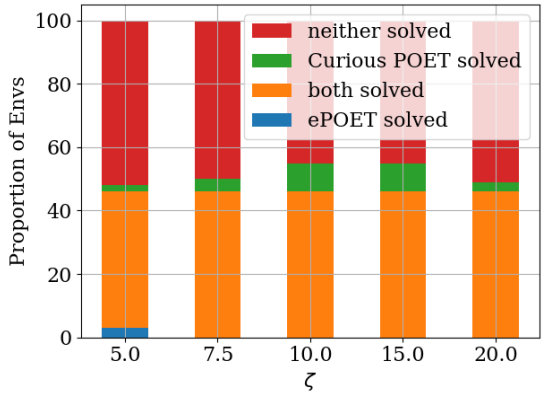

Figure 5: Covering environments partitioned by solution group. Curious populations tend to solve the set of environments which are solved by baseline ePOET (orange), and go on to solve additional environments (green) in the covering set. The relative prominence of the green area over the blue area showcases the ability of Curious POET to not only evolve agent populations that are more diverse than ePOET populations, but also more capable.

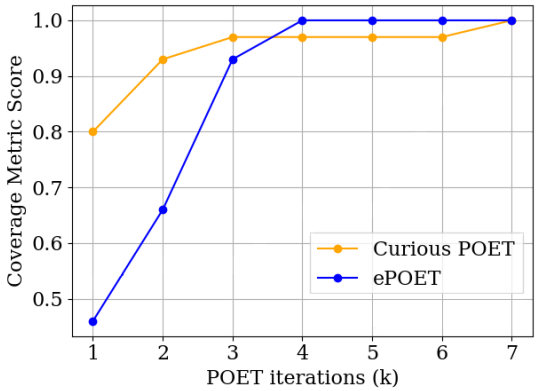

Figure 6: The covering set of environments $\mathcal{A}$ is generated to evaluate the main 2k-POET-iteration experiment. In order to show that these environments are actually solvable, we extend training of one Curious POET and one ePOET population to 7k iterations and see that they eventually solve all of the covering set $\mathcal{A}$.

evaluate using the 2000-POET-iterations covering set $\mathcal{A}$. We see in Figure 6 that after training for $\geq$4k POET iterations, the baseline ePOET populations solve all or most of the environments in the covering set $\mathcal{A}$. From this we conclude that **Curious POET populations learn the same behaviors as ePOET populations, but in fewer POET iterations.** We suspect the underlying mechanism is simply mitigation of redundant behavioral exploration, or equivalently, a population biased toward greater behavioral diversity.

We observe in Figure 5 that curious populations tend to solve a broader distribution than those environments which are solved by baseline populations. This observation is initially hard to understand in the context of previous work (Wang et al., 2020) which suggests that ePOET tends to evolve a population of specialist agents, the behaviors of which are specific to solving their paired environments, but not generally appropriate for solving other environments. We theorize that as Curious POET's objective is both novelty seeking and extrinsic reward driven, evolved agents must produce behaviors that are both novel and useful, i.e., an ensemble of non-overlapping useful skills. In summary, it seems that a population bias toward behavioral diversity coupled with extrinsic reward, can result in improved skill acquisition efficiency.

5.4 Limitations, Fairness, & Future

Many of the limitations of our approach are inherent to evolutionary population-based training. As discussed in Wang et al. (2020), POET populations are relatively compute expensive, which limits our ability to experiment across a diversity of game environments or other mechanisms for realizing intrinsic motivation. Further, other game environments may be more challenging to encode into the needed phenotype of the genotype stored and evolved in the CPPN, a requirement for both POET-based approaches and our population coverage metric for evaluating their performance in a method agnostic to training. While we employ (Pathak et al., 2017) to realize a curious oracle, we expect recent works such as RND (Burda et al., 2019b) or BYOL-explore (Guo, 2022) would provide similar functionality.

Future work should seek to establish both practical and theoretical advances beyond Curious POET. Practical work could address the current necessity of training multiple populations over a range of intrinsic motivation coefficients ζ in order to find the optimal value. Ideally, ζ Scheduling and/or POET-specific hyperparameter optimization would lead to approaches to population-based training that generalize across multiple diverse environments with minimal user input. Given our success generating high-performing and diverse agent populations, we anticipate that work which selects agents from a population for specific tasks, or that seeks to intelligently compose multiple agent skills are interesting avenues of future research.

Acknowledgments

We acknowledge the insights provided by Hamilton Clouse, Christian Manasseh, Nathaniel Bade, Jared Bennett, David Ackerman, Karleigh Pine, Joel Klipfel, and Nathaniel Hamilton who provided valuable insights.

Additionally, we would like to express our particular gratitude to Nathaniel Bade and Jared Bennett of Mobius Logic for their contributions to this research through the development of the Marco Polo codebase used throughout our experimentation.

References

Barto, A. G., Singh, S., Chentanez, N., et al. (2004). Intrinsically motivated learning of hierarchical collections of skills. In *Proceedings of the 3rd International Conference on Development and Learning*, volume 112, page 19, La Jolla, CA. The Salk Institute for Biological Studies.

Brant and Stanley (2017). Minimal criterion coevolution: a new approach to open-ended search. In *Proceedings of the Genetic and Evolutionary Computation Conference*, New York, NY. Association for Computing Machinery.

Burda, Y., Edwards, H., Pathak, D., Storkey, A., Darrell, T., and Efros, A. A. (2019a). Large-scale study of curiosity-driven learning. In *Seventh International Conference on Learning Representations*, Appleton, WI. ICLR.

Burda, Y., Edwards, H., Storkey, A., and Klimov, O. (2019b). Exploration by random network distillation. In *Seventh International Conference on Learning Representations*, pages 1–17, Appleton, WI. ICLR.

Dharna, A., Hoover, A. K., Togelius, J., and Soros, L. (2022). Transfer dynamics in emergent evolutionary curricula. *IEEE Transactions on Games*, 15:157–170.

Dharna, A., Togelius, J., and Soros, L. B. (2020). Co-generation of game levels and game-playing agents. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 16, pages 203–209, Washington, DC. AAAI Press.

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

Guo, e. a. (2022). Byol-explore: Exploration by bootstrapped prediction. In *Neurips 2022*, Red Hook, NY. Advances in Neural Information Processing Systems.

Lehman and Stanley (2011a). Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the Genetic and Evolutionary Computation Conference, 2011, pp. 211–218*, New York, NY. Association for Computing Machinery.

Lehman, J. and Stanley, K. O. (2011b). *Novelty Search and the Problem with Objectives*. Springer New York, New York, NY.

Lehman, S. (2008). Exploiting openendedness to solve problems through the search for novelty. In *ALIFE, 2008, pp. 329-336*, Cambridge, MA. MIT Press.

Lehman, S. (2011). Abandoning objectives: Evolution through the search for novelty alone. In *Evolutionary Computation, Volume 19, Issue 2, Pages 189-223*, Cambridge, MA. MIT Press.

Mouret, J.-B. and Clune, J. (2015). Illuminating search spaces by mapping elites.

Nasir, M. U., Beukman, M., James, S., and Cleghorn, C. W. (2022). Augmentative topology agents for open-ended learning.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning (ICML)*, New York, NY. Association for Computing Machinery.

Salimans, T., Ho, J., Chen, X., Sidor, S., and Sutskever, I. (2017). Evolution strategies as a scalable alternative to reinforcement learning.

Secretan, e. a. (2011). Picbreeder: A case study of collaborative evolutionary exploration of design space. *Evolutionary Computation 19*, pages 373–403.

Silver, D., Singh, S., Precup, D., and Sutton, R. S. (2021). Reward is enough. *Artificial Intelligence*, 299:103–535.

Soros, L. and Stanley, K. (2014). Identifying Necessary Conditions for Open-Ended Evolution through the Artificial Life World of Chromaria. volume ALIFE 14: The Fourteenth International Conference on the Synthesis and Simulation of Living Systems, pages 793–800, Manhattan, New York. ASME.

Stanley, K. O. (2019). Why Open-Endedness Matters. *Artificial Life*, 25(3):232–235.

Stanley, K. O., Lehman, J., and Soros, L. (2017). Open-endedness: The last grand challenge you've never heard of. https://www.oreilly.com/ideas/open-endednessthe-last-grand-challenge-youve-never-heard-of. Accessed on April 27, 2023.

Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127.

Wang, R., Lehman, J., Clune, J., and Stanley, K. O. (2019). Paired open-ended trailblazer (poet): Endlessly generating increasingly complex and diverse learning environments and their solutions.

Wang, R., Lehman, J., Rawal, A., Zhi, J., Li, Y., Clune, J., and Stanley, K. O. (2020). Enhanced poet: Open-ended reinforcement learning through unbounded invention of learning challenges and their solutions.

White (1959). Motivation reconsidered: The concept of competence. *Psychological Review 66*, pages 297–333.

Zhou, F. and Vanschoren, J. (2022). Open-ended learning strategies for learning complex locomotion skills.