Learning General World Models in a Handful of Reward-Free Deployments

Anonymous Author(s)
Affiliation
Address
email

Abstract

Building generally capable agents is a grand challenge for deep reinforcement learning (RL). To approach this challenge practically, we outline two key desiderata: 1) to facilitate generalization, exploration should be task agnostic; 2) to facilitate scalability, exploration policies should collect large quantities of data without costly centralized retraining. Combining these two properties, we introduce the *reward-free deployment efficiency* setting, a new paradigm for RL research. We then present CASCADE, a novel approach for self-supervised exploration in this new setting. CASCADE seeks to learn a world model by collecting data with a population of agents, using an information theoretic objective inspired by Bayesian Active Learning. We show theoretical guarantees on a tabular analogue to CASCADE. We also demonstrate that CASCADE collects diverse task-agnostic datasets and learns agents that generalize zero-shot to novel, unseen downstream tasks on Atari, MiniGrid and the DM Control Suite.

1 Introduction

Reinforcement learning (RL, [100]) has achieved a number of impressive feats over the past decade, with successes in games [66, 12, 95, 109] robotics [73], and the emergence of real world applications [10, 23]. Indeed, now that RL has successfully mastered a host of individual tasks, the community has begun to focus on the grand challenge of building *generally capable* agents [86, 104, 65, 4].

In this work, we take steps towards building *generalist agents at scale*, where we outline two key desiderata. First, in order for agents to become *generalists* that can adapt to novel tasks, we eschew the notion of restricting agent learning to task-specific reward functions. Instead, we believe it is vital to focus on the *reward-free* problem setting [79, 26], whereby agents must discover novel skills and behaviors for themselves.[1] Consider the problem of learning to control robotic arms: we may already have some expert offline data to learn from, which will likely cover only a subset of the entire range of possible behaviors. Therefore, to learn additional *general* skills, it is imperative to further collect novel and diverse data *without* a pre-specified reward function.

Second, to ensure scalability, we should have access to a large fleet of robots that we can *deploy* to gather this data for a large number of timesteps [57], without costly and lengthy centralized retraining during this crucial phase [64]. This has recently been referred to as *deployment efficiency*, falling between the typical online/offline RL dichotomy (see Figure 1). Limiting deployments not only reduces the overhead in retraining exploration policies, but also limits the potential costs and risks present when deploying new policies [103, 51], an important consideration in many real world settings, such as robotics [46], education [62] and healthcare [27]. Combining these two desiderata, we introduce the *reward-free deployment efficiency* setting, a new paradigm for deep RL research.

[1] Indeed, designing reward functions to learn behaviors can have unintended consequences [70, 5].

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

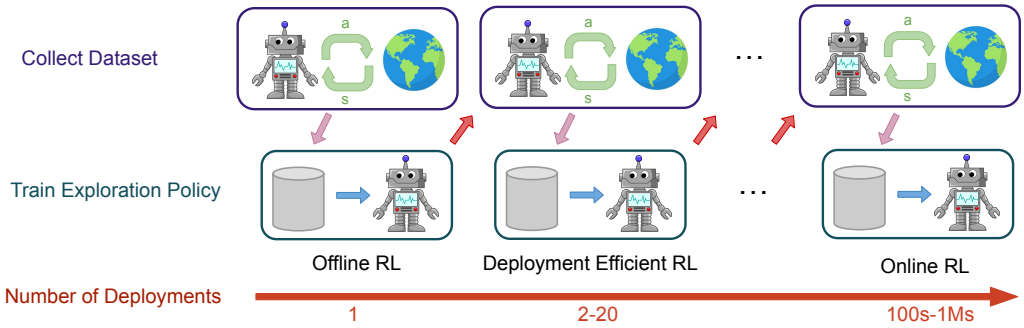

Figure 1: **Reward-Free Deployment Efficiency**: We seek to train agents using data collected by a handful of exploration policies (2-20). This is different from offline RL which trains from a single dataset, or online RL which often retrains as frequently as every tiemstep and deploys millions of different policies in an environment. Unlike previous deployment efficiency work (Figure 1 in [64]), our data collection is reward-free, and we evaluate our resulting dataset in a zero-shot fashion on unseen tasks.

To maximize deployment efficiency we now must train exploration policies capable of collecting large quantities of interesting and, crucially, *useful* data. To tackle this new problem we use *world models* [32], which allow agents to plan and/or train policies without interacting with the true environment. World models have already been shown to be highly effective for deployment efficiency [64], offline RL [114, 47, 7, 84] and self-supervised exploration [94, 92]. Furthermore, world models offer the potential for increasing agent generalization capabilities [6, 35, 65, 15, 7, 19, 9], one of the frontiers of RL research [76, 49]. However, since existing self-supervised methods for learning world models are designed to collect only a few transitions with a single exploration policy, they likely produce a *homogenous dataset* when deployed at scale, which does not optimally improve the model.

Instead, drawing analogies from Bayesian Active Learning [40, 50], we introduce a new information theoretic objective that maximizes the information gain from an *entire dataset* collected by a *population* of exploration agents (see Figure 2). We call our method ***C**oordinated **A**ctive **S**ample **C**ollection vi**a** **D**iverse **E**xplorers* or CASCADE. We provide theoretical justification for CASCADE, which emphasizes the importance of collecting data with diverse agents. In addition, we provide a rigorous empirical evaluation across three challenging domains that shows CASCADE can discover a rich dataset from a handful of deployments. We see that CASCADE produces general exploration strategies that are equally adept at both "deep" exploration problems and diverse behavior discovery. This makes it possible to train agents capable of zero-shot transfer when rewards are provided at test time in a variety of different settings.

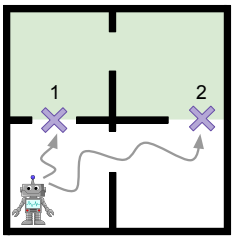

Figure 2: Motivation for CASCADE: Green areas represent high expected information gain. If we train a population of agents independently, at deployment time they will all follow the trajectory to #1, producing a homogenous dataset. However, if we consider the diversity of the data then we will produce agents that reach both #1 and #2.

To summarize, our contributions are as follows: 1) We introduce a novel problem setting, *Reward-Free Deployment Efficiency*, designed to train generalist agents in a scalable fashion; 2) We propose CASCADE, a theoretically motivated model-based RL agent designed to gather diverse, highly informative data, inspired by Bayesian Active Learning; 3) We demonstrate that CASCADE is capable of *improved zero-shot transfer*, and provide analysis that shows CASCADE theoretically guarantees improved sample efficiency over other naïve methods that do not ensure sample diversity.

2 Problem Statement

Reinforcement learning (RL) considers training an agent to solve a Markov Decision Process (MDP), represented as a tuple $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, P, R, \rho, \gamma\}$, where $s \in \mathcal{S}$ and $a \in \mathcal{A}$ are the set of states and actions respectively, $P(s'|s,a)$ is a probability distribution over next states given a previous state and action, $R(s,a,s') \to r$ is a reward function mapping a transition to a scalar reward, ρ is an initial state distribution and γ is a discount factor. A policy π acting in the environment produces a trajectory $\tau = \{s_1, a_1, \ldots, s_H, a_H\}$ for an episode with horizon H. Since actions in the trajectory

are sampled from a policy, we can then define the RL problem as finding a policy π that maximizes expected returns in the environment, i.e. $\pi^\star = \arg\max_\pi \mathbb{E}_{\tau\sim\pi}[R(\tau)]$.

We seek to learn policies that can transfer to *any* MDP within a family of MDPs. This can be formalized as a *Contextual* MDP [49], where the observations and dynamics are consistent but the reward function is decided by the context. If the context is unknown at training time, we must collect data that *sufficiently covers the space of possible reward functions*. Finally, to facilitate scalability, we operate in the deployment efficient paradigm [64], whereby policy learning and exploration are completely separate, and during a given *deployment*, we gather a large quantity of data without further policy retraining (c.f. online approaches like DER [107], which take multiple gradient steps *per* exploration timestep in the real environment). Taken together, we consider the *reward-free deployment efficiency* problem. This differs from previous work as follows: 1) unlike previous deployment efficiency work, our exploration is task agnostic; 2) unlike previous reward-free RL work, we cannot update our exploration policy π_{EXP} during deployment. Thus, the focus of our work is on how to train π_{EXP} offline such that it gathers heterogeneous and informative data which facilitate zero-shot transfer to unknown tasks.

In this paper we make use of model-based RL (MBRL), where the goal is to learn a model of the environment (or *world model* [91]) and then use it to subsequently train policies to solve downstream tasks. To do this, the world model needs to approximate both P and R. Typically, the model will be a neural network, parameterized by ψ, hence we denote the approximate dynamics and reward functions as P_ψ and R_ψ, which produces a new "imaginary" MDP, $\mathcal{M}_\psi = (\mathcal{S}, \mathcal{A}, P_\psi, R_\psi, \rho_0)$. We focus on Dyna-style MBRL [99], whereby we train a policy (π_θ parameterized by θ) solely using "imagined" transitions inside $\mathcal{M}_\psi$. We can use any model-free algorithms to optimize the policy. Furthermore, we can train the policy on a single GPU with parallelized rollouts since the simulator is a neural network [52]. The general form of all methods in this paper is shown in Algorithm 1, with the key difference being step 5: We aim to update π_{EXP} in the new imaginary MDP $\mathcal{M}_\psi$ such that it continues to collect a large, diverse quantity of reward-free data.

Algorithm 1 Reward-Free Deployment Efficiency via World Models

1: **Input:** Initial exploration policy π_{EXP}
2: **while** True **do**
3: Deploy π_{EXP} to collect a *large quantity* of *reward-free* data.
4: Train world model on all existing data.
5: Update π_{EXP} in new imaginary MDP $\mathcal{M}_\psi$.
6: **end while**

We focus on learning world models from high dimensional sensory inputs such as pixels [32, 72, 45], where at each timestep we are given access to an observation o_t rather than a state s_t. A series of recent works have shown tremendous success by mapping the observation to a compact latent state z_t [37, 36, 38]. In this paper we will make use of the model from DreamerV2 [38], which has been shown to produce highly effective policies in a variety of high dimensional environments. The primary component of DreamerV2 is a Recurrent State Space Model (RSSM) that uses a learned latent state to predict the image reconstruction, reward r_t and discount factor γ_t. Aside from the reward head, all components of the model are trained jointly, in similar fashion to variational encoders (VAEs, [48, 88]). For zero-shot evaluation, we follow [92] and only train the reward head at test time when provided with labels for our pre-collected data, which is then used to train a behavior policy. Thus, it is critical that our dataset is sufficiently diverse to enable learning novel, unseen behaviors.

3 Coordinated Active Sample Collection

The primary aim of this work is to train a population of B exploration policies $\{\pi^{(i)}_{\text{EXP}}\}_{i=1}^B$ such that they collectively acquire data which maximally improves the accuracy of a *world model*. To achieve this, we take inspiration from the information theoretic approach in Plan2Explore [92]:

$$\pi_{\text{EXP}} = \arg\max_\pi \mathcal{I}\left(d^\pi_{\mathcal{M}_\psi}; \mathcal{M}_\psi\right) = \mathcal{H}(d^\pi_{\mathcal{M}_\psi}) - \mathcal{H}(d^\pi_{\mathcal{M}_\psi}|\mathcal{M}_\psi) \tag{1}$$

where $d^\pi_{\mathcal{M}_\psi}$ is the distribution of states visited by the policy π in the imaginary MDP $\mathcal{M}_\psi$. This objective produces π_{EXP}, a policy whose visitation distribution has a high entropy when computed over

model samples, but has low entropy for each individual MDP model (i.e., high epistemic/reducable uncertainty). To make this objective more general, we represent the trajectory data collected by a policy with a "summary" embedding space [75, 68]. Let $\Phi : \Gamma \to \Omega$ be a summary function mapping trajectories into this embedding space. $\mathbb{P}^{\Phi}_{\pi}[\mathcal{M}_\psi]$ denotes the embedding distribution generated by policy π in imaginary MDP $\mathcal{M}_\psi$. We can now write the objective from Eq. 1 as follows:

$$\pi_{\text{EXP}} = \arg\max_{\pi} \mathcal{I}\left(\mathbb{P}^{\Phi}_{\pi}[\mathcal{M}_\psi]; \mathcal{M}_\psi\right) = \mathcal{H}(\mathbb{P}^{\Phi}_{\pi}[\mathcal{M}_\psi]) - \mathcal{H}(\mathbb{P}^{\Phi}_{\pi}[\mathcal{M}_\psi]|\mathcal{M}_\psi) \tag{2}$$

This more general framework allows us to consider multiple representations for trajectories, in a similar fashion to behavioral characterizations in Quality Diversity algorithms [81]. For the rest of this discussion, we will use the final state embedding as our summary representation, whereby $\Phi(\tau) = h_H$, since in the case of the RSSM, the final latent is a compact representation of the entire trajectory collected by the policy, analogous to the final hidden state in an RNN [90, 98].

3.1 A Cascading Objective with Diverse Explorers

We now consider a population-based version of Equation. 2, using B agents:

$$\{\pi^{(i)}_{\text{EXP}}\}_{i=1}^{B} = \arg\max_{\boldsymbol{\pi}^B \in \Pi^B} \mathcal{I}\left(\prod_{i=1}^{B} \mathbb{P}^{\Phi}_{\pi^{(i)}}[\mathcal{M}_\psi]; \mathcal{M}_\psi\right) = \mathcal{H}\left(\prod_{i=1}^{B} \mathbb{P}^{\Phi}_{\pi^{(i)}}[\mathcal{M}_\psi]\right) - \mathcal{H}\left(\prod_{i=1}^{B} \mathbb{P}^{\Phi}_{\pi^{(i)}}[\mathcal{M}_\psi]\middle|\mathcal{M}_\psi\right) \tag{3}$$

where $\boldsymbol{\pi}^B = \pi^{(1)}, \cdots, \pi^{(B)}$ and $\prod_{i=1}^{B} \mathbb{P}^{\Phi}_{\pi^{(i)}}[\mathcal{M}_\psi]$ is the product measure of the policies' embedding distributions in $\mathcal{M}_\psi$. By definition, the conditional entropy factorizes as:

$$\mathcal{H}\left(\prod_{i=1}^{B} \mathbb{P}^{\Phi}_{\pi^{(i)}}[\mathcal{M}_\psi]\middle|\mathcal{M}_\psi\right) = \sum_{i=1}^{B} \mathcal{H}\left(\mathbb{P}^{\Phi}_{\pi^{(i)}}[\mathcal{M}_\psi]\middle|\mathcal{M}_\psi\right). \tag{4}$$

It is now possible to show that maximum information gain is achieved with an *diverse* set of agents:

Lemma 1. *When all models $\mathcal{M}_\psi$ in the support of the model posterior are deterministic and tabular, and the space of policies Π consists only of deterministic policies, there always exists a solution $\{\pi^{(i)}_{\text{EXP}}\}_{i=1}^{B}$ satisfying $\pi^{(i)}_{\text{EXP}} \neq \pi^{(j)}_{\text{EXP}} \forall i \neq j$. Moreover, there exists a family of tabular MDP models, such that the maximum cannot be achieved by setting $\pi_{\text{EXP}}(i) = \pi$ for a fixed π.*

The proof of Lemma 1 is in Appendix D.2.

Since the mutual information objective 3 is submodular, a greedy algorithm yields a $(1 - \frac{1}{e})$ approximation of the optimum [69]. Leveraging this insight, let us assume that we already have a set of policies $\pi^{(1)}, \cdots, \pi^{(i-1)}$; we then select the next policy $\pi^{(i)}$ based on the following greedy objective:

$$\begin{aligned}\pi^{(i)} &= \arg\max_{\tilde{\pi}^{(i)} \in \Pi} \mathcal{I}\left(\prod_{j=1}^{i} \mathbb{P}^{\Phi}_{\tilde{\pi}^{(j)}}[\mathcal{M}_\psi]; \mathcal{M}_\psi \middle| \tilde{\pi}^{(j)} = \pi^{(j)} \ \ \forall j \leq i-1\right) \\ &= \mathcal{H}\left(\prod_{j=1}^{i} \mathbb{P}^{\Phi}_{\tilde{\pi}^{(j)}}[\mathcal{M}_\psi] \middle| \tilde{\pi}^{(j)} = \pi^{(j)} \ \ \forall j \leq i-1\right) - \mathcal{H}\left(\prod_{j=1}^{i} \mathbb{P}^{\Phi}_{\pi^{(j)}}[\mathcal{M}_\psi] \middle| \mathcal{M}_\psi, \tilde{\pi}^{(j)} = \pi^{(j)} \ \ \forall j \leq i-1\right)\end{aligned}$$

Which can be factorized in similar fashion to Equation 4 (See Appendix D).

3.2 A Tractable Objective for Deep RL

Inspired by [92], we make a couple of approximations to derive a tractable objective for $\{\pi_{\text{EXP}}\}_{i=1}^{B}$ in the deep RL setting. First, we assume that the embedding distributions are Gaussian with means that depend on the policies and sampled worlds, and variances that depend on the worlds, i.e. $\mathbb{P}^{\Phi}_{\pi}[\mathcal{M}_\psi = w] = \mathcal{N}(\mu(w, \pi), \Sigma(w))$. In this case, $\mathcal{H}(\mathbb{P}^{\Phi}_{\pi}[\mathcal{M}_\psi]|\mathcal{M}_\psi = w) = \rho(w)$, and Eq. 5 reduces to solving $\pi^{(i)} = \arg\max_{\tilde{\pi}^{(i)} \in \Pi} \mathcal{H}\left(\prod_{j=1}^{i} \mathbb{P}^{\Phi}_{\tilde{\pi}^{(j)}}[\mathcal{M}_\psi] \middle| \tilde{\pi}^{(j)} = \pi^{(j)} \ \ \forall j \leq i-1\right)$ for a policy that maximizes the resulting joint entropy of the embedding distribution when added to the policy population. This produces the following surrogate objective, maximizing a quadratic

cascading disagreement:

$$\text{ImagDiv}^{\Phi}(\pi|\pi^{(1)},\cdots,\pi^{(i-1)}) = \mathbb{E}_{\tau\sim\mathbb{P}^{\pi}[\mathcal{M}_{\psi}]}\left[\frac{1}{|\mathcal{D}^{(i-1)}|}\sum_{\tilde{\tau}\in\mathcal{D}^{(i-1)}}\|\Phi(\tau)-\Phi(\tilde{\tau})\|^2\right]$$

where $\mathcal{D}^{(i-1)}$ is a dataset of imagined trajectories sampled from policies $\pi^{(1)},\cdots,\pi^{(i-1)}$ in the model, and ImagDiv is short for Imagined Diversity. Finally, following [92, 8], we also add a per state model-disagreement/information gain component to each policy's reward to encourage a richer landscape for data acquisition: $\text{InfoGain}(\pi) = \mathbb{E}_{\tau\sim\mathbb{P}^{\pi}[\mathcal{M}_{\psi}]}\left[\sum_{(s,a)\in\tau}\sigma(s,a)\right]$ where $\sigma(\cdot,\cdot)$ is the variance across the ensemble latent state predictions (for details see App. B). Taken together, these objectives form our approach, which we call ***C**oordinated **A**ctive **S**ample **C**ollection vi**a** **D**iverse **E**xplorers* or CASCADE. CASCADE trains agents to optimize: 1) a diversity term (ImagDiv) that takes into account the behaviors of the other agents in the population; 2) an information gain term (InfoGain) that encourages an individual agent to sample states that maximally improve the model:

$$\pi^{(i)} = \arg\max_{\pi\in\Pi}\left[\lambda\text{ImagDiv}^{\Phi}(\pi|\{\pi^{(j)}_{\text{EXP}}\}_{j=1}^{i-1}) + (1-\lambda)\text{InfoGain}(\pi)\right] \tag{6}$$

where λ is a weighting hyperparameter that trades off whether we favor individual model information gain or population diversity. Finally, we train the B agents in *parallel* using (policy) gradient descent over θ, which makes it possible to achieve the same wall clock time as training a single agent [21].

3.3 Theoretical Motivation

We now seek to provide a tabular analogue to CASCADE which provides theoretical guarantees. In Appendix C we outline the pseudo-code of CASCADE-TS, a greedy Thompson Sampling algorithm [3] that produces the i-th exploration policy in a tabular enviornment using "imaginary" data gathered by running policies $\pi^{(1)},\cdots,\pi^{(i-1)}$ in the *model*. We can then show the following:

Lemma 2. *For the class of Binary Tree MDPs, the* CASCADE-TS *algorithm satisfies,*

$$T(\epsilon,\text{Sequential}) \leq T(\epsilon,\text{CASCADE-TS}) \leq T(\epsilon,\text{SinglePolicyBatch})$$

where $T(\epsilon,\cdot)$ are the expected number of rounds of deploying a population of B policies necessary to learn the true model up to ϵ accuracy; SinglePolicyBatch *plays a fixed policy B times in each round;* Sequential *does not have a population, and instead interleaves updates and executions of a single policy B times within each round.*

The proof is in App. C. We see that CASCADE-TS achieves provable efficiency gains over a naïve sampling approach that does not ensure diversity in its deployed agents. This provable gain is achieved by discouraging the i-th policy away from imaginary state-action pairs sampled by the previous $i-1$ policies by using imaginary counts.

Now returning to CASCADE, we can see the importance of leveraging the imaginary data gathered in the model by the previous $i-1$ policies when training the i-th exploration policy. Concretely, encouraging policy $\pi^{(i)}$ to induce high disagreement with the embeddings produced by $\{\pi^{(1)},\cdots,\pi^{(i-1)}\}$ is analogous to the imaginary count bonus term of CASCADE-TS in Line 10 of Alg. 2, which avoids redundant data collection during deployment.

4 Experiments

In our experiments we test whether CASCADE facilitates the learning of generalist agents by evaluating their zero-shot transfer to unseen tasks, given a limited number of reward-free deployments. In all experiments, agents cannot train online in the environment, and instead execute a fixed exploration policy to gather new data during each deployment. We test against 3 baselines. The first is random exploration, which is used in the majority of RL publications. Our next baseline is Plan2Explore [92] (P2E), which trains a single exploration policy that optimizes for expected information gain inside a world model. We then include a population-based version of P2E, which we call *Population Plan2Explore* (PP2E). PP2E trains a set of randomly initialized agents that independently seek to maximize expected information gain. All methods make use of a DreamerV2 world model [38] and use the same hyperparameters for model and agent training (more details in App. B).

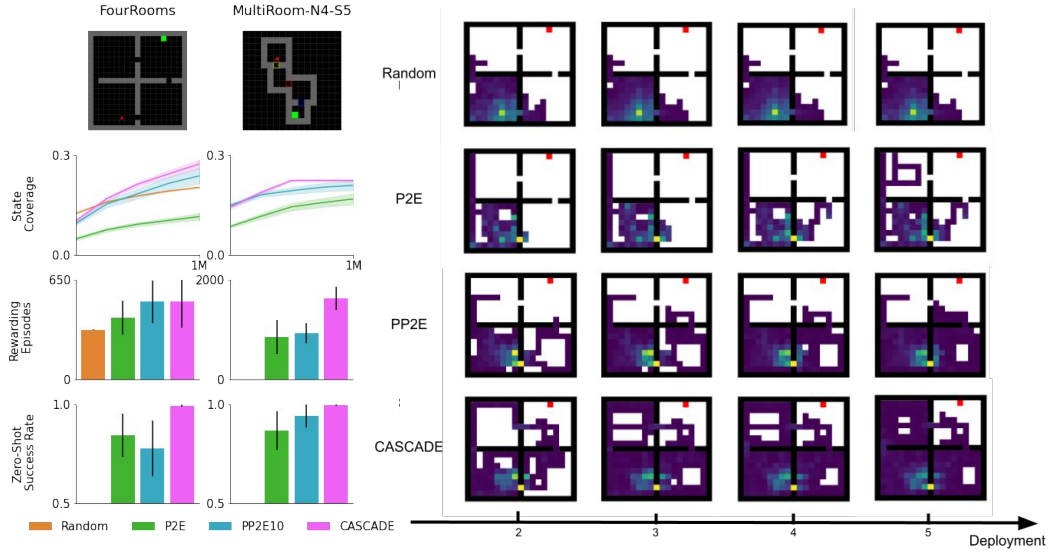

Figure 3: **MiniGrid Results**. Left: Performance statistics for both `FourRooms` and `MultiRoom`. For reward-free exploration we show both "state coverage", which corresponds to the the percentage of states visited on a fixed set of levels, and "rewarding episodes" which is the count of total solved episodes found after 1M steps. Finally for zero-shot transfer we show the success rate of a task-specific behavior policy trained with labels provided only at test time. All plots show the mean of ten seeds with SEM shaded. (Random achieves 0 zero-shot success rate). Right: we show the state coverage on a fixed level in `FourRooms`, using the exploration policy π_{EXP} for each deployment (Test time goal state indicated by the red dot).

We test all methods in two separate settings: *Exploring Worlds* (Sec 4.1) where we seek to collect data in challenging exploration environments. We test how well the explorers cover the state space and discover sparse rewards (also known as "deep exploration" [74, 71]). Next we consider *Exploring Behaviors* (Sec 4.2) where we consider collecting data in continuous control environments. In both settings we test zero-shot transfer as follows: 1) we provide reward labels to the model; 2) we train a separate reward head; 3) we train a task specific behavior policy and test it in the environment. This tests whether our model is general enough to facilitate learning downstream policies for previously unknown tasks [92, 53, 112].

4.1 Exploring Worlds

In many cases we may not know a priori which states in an environment are highly rewarding. Thus, if we wish to use our world model to subsequently train agents to solve tasks, it is crucial that we accurately model as much of state space as possible. Here we consider exploring worlds in two different environments: MiniGrid [18], and Atari [11]. In both settings, we start with a *random* deployment before beginning reward-free exploration.

We begin with MiniGrid, a set of sparse-reward, partially observable navigation environments commonly used to test state-of-the-art exploration methods [85, 16, 29, 117]. MiniGrid environments are procedurally generated, which provides an additional generalization challenge [44, 89]. We consider the canonical `FourRooms` and the more challenging `MultiRoom`. In each case we are limited to 5 deployments of 200k transitions, for a total of 1M environment interactions.[2] We score the reward-free exploration in two ways: 1) state coverage, where we evaluate the percentage of the states visited by the exploration policies after each deployment on the same set of held out test levels; and 2) rewarding episodes, where we show the number of rewarding episodes collected during the training process. Note that these two objectives are not useful in isolation, as solved levels may only include a narrow set of observations, while state coverage alone could be maximized without performing deep, goal-seeking exploration in the environment.

We show the results in Figure 3, where we see the CASCADE exploration policies cover the state space far more effectively than all baselines. In `FourRooms`, both CASCADE and PP2E find the goal

[2]Zero-shot success rates of P2E, PP2E and CASCADE increase with more training steps, and CASCADE achieves a near 100% success rate at 1M steps in both `FourRooms` and `MultiRoom`.

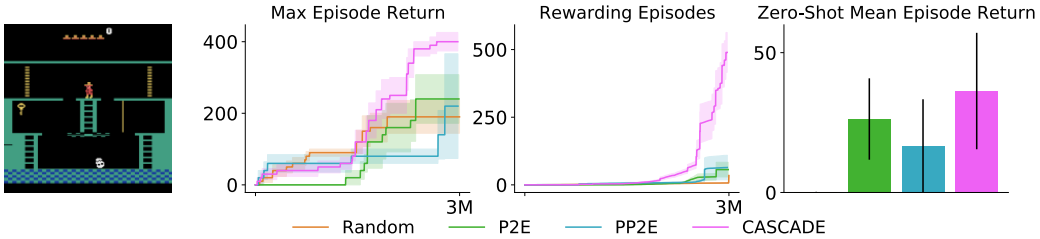

Figure 4: **Montezuma's Revenge Results**. From left to right we show a frame from the game, a plot of the cumulative maximum episode reward, the total number of rewarding episodes discovered and zero-shot average return, from 3M training steps (15 deployments). Plots show mean and SEM over 10 seeds.

an equal number of times, while CASCADE finds almost double the number of rewarding episodes in the `MultiRoom` environment. We also test zero-shot performance, where CASCADE achieves almost 100% success rate. Interestingly, in both settings the Random policy does cover a large proportion of the state space, but finds fewer rewarding episodes and subsequently does not have a sufficient quantity to train a policy to solve the task (and accordingly achieves 0% success rate).

Next we consider the Atari game `Montezuma's Revenge`, one of the hardest exploration challenges in the Arcade Learning Environment [11], often used as a barometer for exploration capabilities [25]. Many online RL methods achieve zero return in this environment, despite deploying millions of exploration policies trained with the extrinsic rewards. Instead we use 15 reward-free deployments,[3] each consisting of 200k steps, with results shown in Figure 4. Once again we see that CASCADE not only finds higher rewarding episodes (almost double the next best baseline), but discovers an order of magnitude more rewarding episodes on average. This translates to stronger zero-shot performance than all baselines. We further see that the Random baseline again does find an occasional high reward episode (Max Episode Return), but it does not find a sufficient quantity (Rewarding Episodes) to produce a positive zero-shot return. Figure 8 in the Appendix visualizes the diversity of data collected by each method.

4.2 Exploring Behaviors

Now we have shown our agents are capable of exploring the state space in challenging navigation settings, the next thing we wish to test is whether they sufficiently explore a range of behaviors. To test this, we move from discrete to continuous control, using the DeepMind Control Suite [106]. In these experiments we operate from pixels, and assume access to an initial dataset of 200k transitions. We then conduct two further reward-free deployments, each collecting 200k additional transitions. To test the ability of our methods to improve the generality of a model when given access to prior data, we test three different initial datasets commonly used in the offline RL literature: `random`, `medium` and `expert`. We focus on the `walker` environment and evaluate our agents on the four tasks proposed in URLB [54].

In Figure 5 we show the performance for each downstream task, averaged over all three initial datasets. In each plot we show the mean performance per deployment, for a single task. As expected, performance increases with each deployment, as the model has more data. However, this improvement is drastically larger for CASCADE, which outperforms all baselines. The next best method is PP2E, while both P2E and Random provide only modest gains. See Figure 7 in the Appendix for a full breakdown of the environment-dataset performance.

To test the overall significance of our results in this benchmark, we use aggregated statistics from the RLiable library [2]. To do this, we combine the results across all datasets (`random`, `medium`, `expert`) and all downstream tasks (`stand`, `walk`, `run`, `flip`) and compute the following: 1) the Inter-Quartile Mean (IQM) of the normalized scores (the robust statistic recommended in [2]); 2) the Ranking, which shows the ordering of the methods by task; 3) the Probability of Improvement, the likelihood of one method outperforming another on a new, unseen task. We show the results for each deployment in Figure 6. As we see, after a single deployment CASCADE shows clear gains over the next best baseline (PP2E), ranking first more than any other method and showing statistical significance under the Mann-Whitney U test [63]. Crucially, after a second deployment these gains compound, with the

[3]The max episode returns of all methods stabilize after 15 deployments (3M training steps in total).

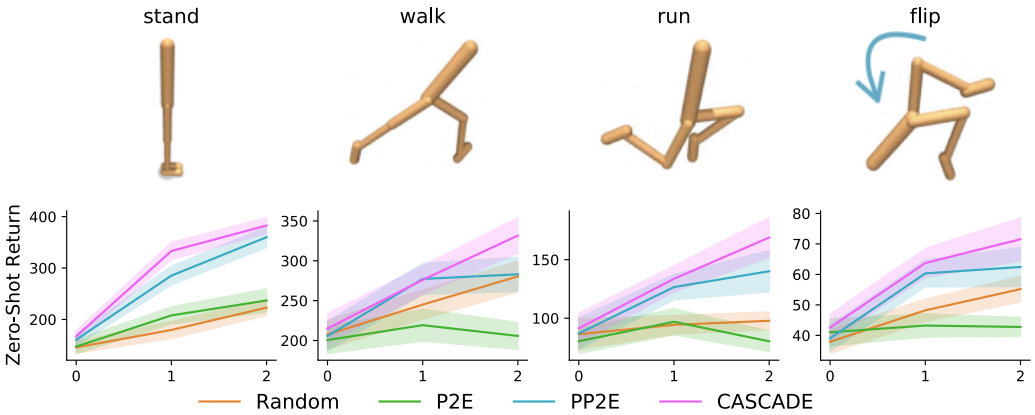

Figure 5: **DMC Zero-Shot Results by Task**: Aggregate performance across all three initial datasets (`random`, `medium`, `expert`), and show the mean performance across a total of 30 seeds, with SEM shaded.

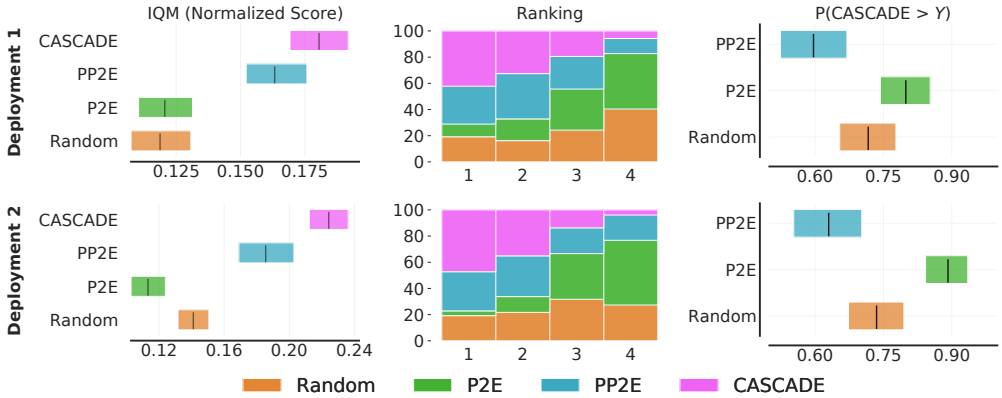

Figure 6: **DMC Zero-Shot Aggregate Results**: Plots show RLiable metric performance aggregated across all three initial datasets (`random`, `medium`, `expert`) for a total of 30 seeds. Note that the 'Probability of Improvement' lower CIs all exceed 0.5, indicating CASCADE provides a statistically significant improvement (under the Mann-Whitney U test) over all baseline methods [2], even after a single deployment.

probability of improvement increasing to over 60%. Note that here the single agent P2E performs poorly, likely due to only collecting data with a single mode of exploratory behavior.

4.3 Discussion and Limitations

Across a variety of environments we have shown that CASCADE is the most effective method in training generally capable agents. This is because CASCADE provides the best of both worlds: deep exploration by seeking the frontier of the current knowledge (via InfoGain), while covering the space of possible behaviors (via ImagDiv). Each of our baselines lacked in one of these two key properties. For example, Random exploration fared poorly when exploring worlds, which requires deep exploration. However, it can be a highly effective method for exploring behaviors. This is likely why boosting entropy in the action space is so effective for robotics tasks [34], but less so in deep exploration tasks [71]. Furthermore, the information theoretic baselines were more effective than Random in Exploring Worlds, but the disappointing performance for P2E in DMC is likely caused by a lack of data diversity, while despite diversity from random initialization, the PP2E populations were still not sufficiently heterogeneous to match the coverage of CASCADE.

In terms of limitations, the main weakness in current methods is the absolute level of performance. Despite outperforming baselines in DMC, the policies learned by CASCADE are far from optimal. This is likely due to the challenge of the setting (reward-free deployment efficiency) and we believe that future work in the community will be crucial to further improve these methods. Finally, CASCADE does require training additional agents that may provide additional computational expense. In this setting we assume the time between deployments is arbitrary, but this may not be the case in all settings and future work could consider ways to make this process more efficient and scalable.

5 Related Work

At the core of our paper is the notion of training generalist agents, which have recently been of increased interest in RL [86, 65, 104, 105]. In particular, we focus on the *reward-free* paradigm, since in many cases it may not be possible to know a priori all possible tasks we may wish our agent to solve [43]. With this in mind, there has been a surge of interest in methods that can learn in a self-supervised fashion, to build representations that lead to fast (or even zero-shot) adaptation to future tasks. These methods range from competence-based (i.e. learning "skills") [1, 31, 26, 56, 59, 96] to data-based that aim to maximize data diversity [60, 113].

In this work we take inspiration from *knowledge-based* approaches, which typically seek to learn a world model [8, 79, 80, 28, 102, 14, 93]. In particular we build on Plan2Explore [92] which trains an exploration agent by maximizing an objective resembling information gain inside the world model. Plan2Explore was the first work that showed it is possible to transfer a model learned without rewards in a zero-shot fashion. However, it differed from our work in that Plan2Explore solely considers the online RL setting, deploying a new exploration policy every timestep. Other recent work examined the effectiveness of reward-free RL for collecting datasets for subsequent offline RL [112, 53]. Once again however, this work trains the exploration agent online. By contrast, we only deploy a handful of exploration agents and collect a larger quantity of data with each one.

Our setting resembles a reward-free version of the *deployment efficiency* paradigm, first introduced in [64]. By contrast, their method focuses on policy optimization in the *supervised* RL paradigm, whereas we focus on exploration and zero-shot generalization in the unsupervised RL setting. Follow up work has remained focused on supervised policy optimization [97], whereas we are focused on training generalist agents in the reward-free paradigm.

To address the challenges of this novel problem setting, we draw inspiration from a variety of fields. In particular, several works have considered exploration with a population of agents [101, 24, 61], but these have only considered task-specific online RL. Our choice of a diverse population of agents takes inspiration from the field of Quality Diversity (QD, [81, 22]), where it has been shown that diversity can boost exploration in RL [20]. Recently, [58] showed it is possible to train diverse populations inside a world model to acquire new skills. As far as we are aware, we are the first to consider a cascading objective for training diverse populations, which is typically done either synchronously [78] or sequentially [118, 115].

We make use of an information theoretic objective, which has been shown to be highly effective for exploration in RL [67, 41, 39]. Most similar to our work [116] consider the reward free RL setting, and show that maximizing entropy can lead to improved exploration that can facilitate transfer to arbitrary reward functions. By contrast we consider the limited deployment setting and use model-based RL. Finally, our work relates to collective intelligence [33, 111], since each member of the population contributes a small piece to the puzzle that produces a general world model.

6 Conclusion

We introduced the *reward-free deployment efficiency* paradigm—an important problem setting for learning generally capable agents in a scalable fashion. To address the challenges of this setting, we proposed CASCADE, a theoretically motivated approach that leverages a diverse population of self-supervised exploration agents. We showed in a variety of experiments that CASCADE produces *general exploration strategies*, adept at both deep exploration *and* gathering sufficiently hetereogeneous datasets that facilitate the learning of diverse downstream behaviors. For future work, we could consider alternative means for computing diversity, for example using optimization landscapes [77] or explicit behavioral diversity [78]. These could also be combined with representations that directly incorporate the behaviors of the policy [13] to further improve the design of the "summary" embedding space. Furthermore, we would be interested in considering transformer models [108] for trajectory modelling [17, 42] that ought to scale more gracefully with larger batches of data [83]. Recent work has shown that this class of models can compose unseen complex multi-task behaviors at test time from fixed offline datasets [30], as well as leverage significantly large and diverse offline datasets to improve behavioral modeling [87], highly relevant to both aforementioned desiderata for the reward-free deployment efficiency paradigm.

References

[1] J. Achiam, H. Edwards, D. Amodei, and P. Abbeel. Variational option discovery algorithms. *CoRR*, abs/1807.10299, 2018.

[2] R. Agarwal, M. Schwarzer, P. S. Castro, A. C. Courville, and M. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in Neural Information Processing Systems*, 34, 2021.

[3] S. Agrawal and N. Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on learning theory*, pages 39–1. JMLR Workshop and Conference Proceedings, 2012.

[4] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R. J. Ruano, K. Jeffrey, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, and M. Yan. Do as i can and not as i say: Grounding language in robotic affordances. In *arXiv preprint arXiv:2204.01691*, 2022.

[5] D. Amodei and J. Clark. Faulty Reward Functions in the Wild. https://blog.openai.com/faulty-reward-functions/, 2016.

[6] A. Anand, J. C. Walker, Y. Li, E. Vértes, J. Schrittwieser, S. Ozair, T. Weber, and J. B. Hamrick. Procedural generalization by planning with self-supervised world models. In *International Conference on Learning Representations*, 2022.

[7] A. Argenson and G. Dulac-Arnold. Model-based offline planning. In *International Conference on Learning Representations*, 2021.

[8] P. Ball, J. Parker-Holder, A. Pacchiano, K. Choromanski, and S. Roberts. Ready policy one: World building through active learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML*, 2020.

[9] P. J. Ball, C. Lu, J. Parker-Holder, and S. Roberts. Augmented world models facilitate zero-shot dynamics generalization from a single offline environment. In *Proceedings of the 38th International Conference on Machine Learning*, pages 619–629, 2021.

[10] M. Bellemare, S. Candido, P. Castro, J. Gong, M. Machado, S. Moitra, S. Ponda, and Z. Wang. Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature*, 588:77–82, December 2020.

[11] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The Arcade Learning Environment: An Evaluation Platform for General Agents. *CoRR*, abs/1207.4708, 2012.

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

[13] H. Bharadhwaj, M. Babaeizadeh, D. Erhan, and S. Levine. Information prioritization through empowerment in visual model-based RL. In *International Conference on Learning Representations*, 2022.

[14] Y. Burda, H. Edwards, A. Storkey, and O. Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019.

[15] A. Byravan, J. T. Springenberg, A. Abdolmaleki, R. Hafner, M. Neunert, T. Lampe, N. Siegel, N. M. O. Heess, and M. A. Riedmiller. Imagined value gradients: Model-based policy optimization with transferable latent dynamics models. In *CoRL*, 2019.

[16] A. Campero, R. Raileanu, H. Kuttler, J. B. Tenenbaum, T. Rocktäschel, and E. Grefenstette. Learning with AMIGo: Adversarially motivated intrinsic goals. In *International Conference on Learning Representations*, 2021.

[17] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[18] M. Chevalier-Boisvert, L. Willems, and S. Pal. Minimalistic gridworld environment for OpenAI Gym. https://github.com/maximecb/gym-minigrid, 2018.

[19] I. Clavera, A. Nagabandi, S. Liu, R. S. Fearing, P. Abbeel, S. Levine, and C. Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. In *International Conference on Learning Representations*, 2019.

[20] E. Conti, V. Madhavan, F. P. Such, J. Lehman, K. O. Stanley, and J. Clune. Improving exploration in Evolution Strategies for deep reinforcement learning via a population of novelty-seeking agents. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, pages 5032–5043, USA, 2018. Curran Associates Inc.

[21] B. Cui, H. Hu, L. Pineda, and J. N. Foerster. K-level reasoning for zero-shot coordination in hanabi. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[22] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret. Robots that can adapt like animals. *Nature*, 521:503–507, 2015.

[23] J. Degrave, F. Felici, J. Buchli, M. Neunert, B. Tracey, F. Carpanese, T. Ewalds, R. Hafner, A. Abdolmaleki, D. de las Casas, C. Donner, L. Fritz, C. Galperti, A. Huber, J. Keeling, M. Tsimpoukelli, J. Kay, A. Merle, J.-M. Moret, S. Noury, F. Pesamosca, D. Pfau, O. Sauter, C. Sommariva, S. Coda, B. Duval, A. Fasoli, P. Kohli, K. Kavukcuoglu, D. Hassabis, and M. Riedmiller. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602:414–419, February 2022.

[24] M. Dimakopoulou, I. Osband, and B. Van Roy. Scalable coordinated exploration in concurrent reinforcement learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[25] A. Ecoffet, J. Huizinga, J. Lehman, K. O. Stanley, and J. Clune. First return then explore. *Nature*, 590 7847:580–586, 2021.

[26] B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2019.

[27] M. Fatemi, T. W. Killian, J. Subramanian, and M. Ghassemi. Medical dead-ends and learning to identify high-risk states and treatments. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 4856–4870. Curran Associates, Inc., 2021.

[28] A. Filos, E. Vértes, Z. Marinho, G. Farquhar, D. Borsa, A. Friesen, F. Behbahani, T. Schaul, A. Barreto, and S. Osindero. Model-value inconsistency as a signal for epistemic uncertainty. *arXiv preprint arXiv:2112.04153*, 2021.

[29] Y. Flet-Berliac, J. Ferret, O. Pietquin, P. Preux, and M. Geist. Adversarially guided actor-critic. In *International Conference on Learning Representations*, 2021.

[30] H. Furuta, Y. Matsuo, and S. S. Gu. Generalized decision transformer for offline hindsight information matching. In *International Conference on Learning Representations*, 2022.

[31] K. Gregor, D. J. Rezende, and D. Wierstra. Variational intrinsic control. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*, 2017.

[32] D. Ha and J. Schmidhuber. Recurrent world models facilitate policy evolution. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, NeurIPS'18, pages 2455–2467, 2018.

[33] D. Ha and Y. Tang. Collective intelligence for deep learning: A survey of recent developments. *CoRR*, abs/2111.14377, 2021.

[34] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine. Soft actor-critic algorithms and applications. *CoRR*, abs/1812.05905, 2018.

[35] D. Hafner. Benchmarking the spectrum of agent capabilities. In *International Conference on Learning Representations*, 2022.

[36] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

[37] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, pages 2555–2565, 2019.

[38] D. Hafner, T. P. Lillicrap, M. Norouzi, and J. Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021.

[39] E. Hazan, S. Kakade, K. Singh, and A. Van Soest. Provably efficient maximum entropy exploration. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2681–2691. PMLR, 09–15 Jun 2019.

[40] N. Houlsby, F. Huszar, Z. Ghahramani, and M. Lengyel. Bayesian active learning for classification and preference learning. *CoRR*, abs/1112.5745, 2011.

[41] R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel. Vime: Variational information maximizing exploration. *Advances in neural information processing systems*, 29, 2016.

[42] M. Janner, Q. Li, and S. Levine. Offline reinforcement learning as one big sequence modeling problem. In *Advances in Neural Information Processing Systems*, 2021.

[43] C. Jin, A. Krishnamurthy, M. Simchowitz, and T. Yu. Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, pages 4870–4879. PMLR, 2020.

[44] N. Justesen, R. R. Torrado, P. Bontrager, A. Khalifa, J. Togelius, and S. Risi. Procedural level generation improves generality of deep reinforcement learning. *CoRR*, abs/1806.10729, 2018.

[45] L. Kaiser, M. Babaeizadeh, P. Milos, B. Osiński, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine, A. Mohiuddin, R. Sepassi, G. Tucker, and H. Michalewski. Model based reinforcement learning for Atari. In *International Conference on Learning Representations*, 2020.

[46] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on Robot Learning*, pages 651–673. PMLR, 2018.

[47] R. Kidambi, A. Rajeswaran, P. Netrapalli, and T. Joachims. Morel : Model-based offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020.

[48] D. P. Kingma and M. Welling. Auto-encoding variational bayes, 2014.

[49] R. Kirk, A. Zhang, E. Grefenstette, and T. Rocktäschel. A survey of generalisation in deep reinforcement learning. *arXiv preprint arXiv:2111.09794*, 2021.

[50] A. Kirsch, J. Van Amersfoort, and Y. Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems*, 32, 2019.

[51] J. Kober, J. A. Bagnell, and J. Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[52] T. Kurutach, I. Clavera, Y. Duan, A. Tamar, and P. Abbeel. Model-ensemble trust-region policy optimization. In *International Conference on Learning Representations*, 2018.

[53] N. Lambert, M. Wulfmeier, W. F. Whitney, A. Byravan, M. Bloesch, V. Dasagi, T. Hertweck, and M. A. Riedmiller. The challenges of exploration for offline reinforcement learning. *CoRR*, 2022.

[54] M. Laskin, D. Yarats, H. Liu, K. Lee, A. Zhan, K. Lu, C. Cang, L. Pinto, and P. Abbeel. URLB: Unsupervised reinforcement learning benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[55] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition, 1989.

[56] L. Lee, B. Eysenbach, E. Parisotto, E. Xing, S. Levine, and R. Salakhutdinov. Efficient exploration via state marginal matching. 2019.

[57] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *Int. J. Robotics Res.*, 37(4-5):421–436, 2018.

[58] B. Lim, L. Grillotti, L. Bernasconi, and A. Cully. Dynamics-aware quality-diversity for efficient learning of skill repertoires. In *ICRA*, 2022.

[59] H. Liu and P. Abbeel. Aps: Active pretraining with successor features. In *Proceedings of the 38th International Conference on Machine Learning*, pages 6736–6747, 2021.

[60] H. Liu and P. Abbeel. Behavior from the void: Unsupervised active pre-training. In *Advances in Neural Information Processing Systems*. 2021.

[61] I.-J. Liu, U. Jain, R. A. Yeh, and A. Schwing. Cooperative exploration for multi-agent deep reinforcement learning. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6826–6836. PMLR, 18–24 Jul 2021.

[62] T. Mandel, Y.-E. Liu, S. Levine, E. Brunskill, and Z. Popovic. Offline policy evaluation across representations with applications to educational games. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*, AAMAS '14, page 1077–1084, Richland, SC, 2014. International Foundation for Autonomous Agents and Multiagent Systems.

[63] H. B. Mann and D. R. Whitney. On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *The Annals of Mathematical Statistics*, 18(1):50 – 60, 1947.

[64] T. Matsushima, H. Furuta, Y. Matsuo, O. Nachum, and S. Gu. Deployment-efficient reinforcement learning via model-based offline optimization. In *International Conference on Learning Representations*, 2021.

[65] R. Mendonca, O. Rybkin, K. Daniilidis, D. Hafner, and D. Pathak. Discovering and achieving goals via world models. In *Advances in Neural Information Processing Systems*. 2021.

[66] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–33, 02 2015.

[67] S. Mohamed and D. J. Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 2125–2133, Cambridge, MA, USA, 2015. MIT Press.

[68] T. Moskovitz, M. Arbel, F. Huszar, and A. Gretton. Efficient wasserstein natural gradients for reinforcement learning. In *International Conference on Learning Representations*, 2021.

[69] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

[70] A. Y. Ng, D. Harada, and S. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *In Proceedings of the Sixteenth International Conference on Machine Learning*, pages 278–287. Morgan Kaufmann, 1999.

[71] B. O'Donoghue, I. Osband, and C. Ionescu. Making sense of reinforcement learning and probabilistic inference. In *International Conference on Learning Representations*, 2020.

[72] J. Oh, X. Guo, H. Lee, R. Lewis, and S. Singh. Action-conditional video prediction using deep networks in atari games. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 2863–2871, Cambridge, MA, USA, 2015. MIT Press.

[73] OpenAI, M. Andrychowicz, B. Baker, M. Chociej, R. Józefowicz, B. McGrew, J. W. Pachocki, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba. Learning dexterous in-hand manipulation. *CoRR*, abs/1808.00177, 2018.

[74] I. Osband, B. V. Roy, D. J. Russo, and Z. Wen. Deep exploration via randomized value functions. *Journal of Machine Learning Research*, 20(124):1–62, 2019.

[75] A. Pacchiano, J. Parker-Holder, Y. Tang, A. Choromanska, K. Choromanski, and M. I. Jordan. Learning to score behaviors for guided policy optimization. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[76] C. Packer, K. Gao, J. Kos, P. Krähenbühl, V. Koltun, and D. Song. Assessing generalization in deep reinforcement learning. *arXiv preprint arXiv:1810.12282*, 2018.

[77] J. Parker-Holder, L. Metz, C. Resnick, H. Hu, A. Lerer, A. Letcher, A. Peysakhovich, A. Pacchiano, and J. Foerster. Ridge rider: Finding diverse solutions by following eigenvectors of the hessian. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 753–765. Curran Associates, Inc., 2020.

[78] J. Parker-Holder, A. Pacchiano, K. Choromanski, and S. Roberts. Effective diversity in population-based reinforcement learning. In *Advances in Neural Information Processing Systems 33*, 2020.

[79] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 2778–2787. JMLR.org, 2017.

[80] D. Pathak, D. Gandhi, and A. Gupta. Self-supervised exploration via disagreement. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5062–5071. PMLR, 09–15 Jun 2019.

[81] J. K. Pugh, L. B. Soros, and K. O. Stanley. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3:40, 2016.

[82] S. Qiu, J. Ye, Z. Wang, and Z. Yang. On reward-free rl with kernel and neural function approximations: Single-agent mdp and markov game. In *International Conference on Machine Learning*, pages 8737–8747. PMLR, 2021.

[83] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. 2019.

[84] R. Rafailov, T. Yu, A. Rajeswaran, and C. Finn. Offline reinforcement learning from images with latent space models. In *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, volume 144 of *Proceedings of Machine Learning Research*, pages 1154–1168. PMLR, 2021.

[85] R. Raileanu and T. Rocktäschel. Ride: Rewarding impact-driven exploration for procedurally-generated environments. In *International Conference on Learning Representations*, 2020.

[86] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, T. Eccles, J. Bruce, A. Razavi, A. Edwards, N. Heess, Y. Chen, R. Hadsell, O. Vinyals, M. Bordbar, and N. de Freitas. A generalist agent, 2022.

[87] M. Reid, Y. Yamada, and S. S. Gu. Can wikipedia help offline reinforcement learning?, 2022.

[88] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

[89] S. Risi and J. Togelius. Increasing generality in machine learning through procedural content generation. *Nature Machine Intelligence*, 2, 08 2020.

[90] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. , 323(6088):533–536, Oct. 1986.

[91] J. Schmidhuber. Making the world differentiable: On using self-supervised fully recurrent neural networks for dynamic reinforcement learning and planning in non-stationary environments. Technical report, 1990.

[92] R. Sekar, O. Rybkin, K. Daniilidis, P. Abbeel, D. Hafner, and D. Pathak. Planning to explore via self-supervised world models. In *International Conference on Machine Learning*, pages 8583–8592. PMLR, 2020.

[93] A. Sharma, S. Gu, S. Levine, V. Kumar, and K. Hausman. Dynamics-aware unsupervised discovery of skills. In *International Conference on Learning Representations*, 2020.

[94] P. Shyam, W. Jaśkowski, and F. Gomez. Model-based active exploration. In *International conference on machine learning*, pages 5779–5788. PMLR, 2019.

[95] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. P. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529:484–489, 2016.

[96] D. Strouse, K. Baumli, D. Warde-Farley, V. Mnih, and S. S. Hansen. Learning more skills through optimistic exploration. In *International Conference on Learning Representations*, 2022.

[97] D. Su, J. D. Lee, J. Mulvey, and H. V. Poor. MURO: Deployment constrained reinforcement learning with model-based uncertainty regularized batch optimization, 2022.

[98] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[99] R. S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *SIGART Bull.*, 2(4):160–163, July 1991.

[100] R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

[101] R. S. Sutton, J. Modayil, M. Delp, T. Degris, P. M. Pilarski, A. White, and D. Precup. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '11, page 761–768, Richland, SC, 2011. International Foundation for Autonomous Agents and Multiagent Systems.

[102] O. Svidchenko and A. Shpilman. Maximum entropy model-based reinforcement learning. *CoRR*, abs/2112.01195, 2021.

[103] M. Tan. Cost-sensitive learning of classification knowledge and its applications in robotics. *Mach. Learn.*, 13(1):7–33, oct 1993.

[104] O. E. L. Team, A. Stooke, A. Mahajan, C. Barros, C. Deck, J. Bauer, J. Sygnowski, M. Trebacz, M. Jaderberg, M. Mathieu, N. McAleese, N. Bradley-Schmieg, N. Wong, N. Porcel, R. Raileanu, S. Hughes-Fitt, V. Dalibard, and W. M. Czarnecki. Open-ended learning leads to generally capable agents. *CoRR*, abs/2107.12808, 2021.

[105] A. Touati and Y. Ollivier. Learning one representation to optimize all rewards. In *Advances in Neural Information Processing Systems*, volume 34, pages 13–23, 2021.

[106] S. Tunyasuvunakool, A. Muldal, Y. Doron, S. Liu, S. Bohez, J. Merel, T. Erez, T. Lillicrap, N. Heess, and Y. Tassa. dm control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020.

[107] H. van Hasselt, M. Hessel, and J. Aslanides. When to use parametric models in reinforcement learning? In *NeurIPS*, pages 14322–14333, 2019.

[108] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[109] O. Vinyals, I. Babuschkin, W. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. Agapiou, M. Jaderberg, and D. Silver. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575, 2019.

[110] R. Wang, S. S. Du, L. Yang, and R. R. Salakhutdinov. On reward-free reinforcement learning with linear function approximation. *Advances in neural information processing systems*, 33:17816–17826, 2020.

[111] A. Yahya, A. Li, M. Kalakrishnan, Y. Chebotar, and S. Levine. Collective robot reinforcement learning with distributed asynchronous guided policy search. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 79–86, 2017.

[112] D. Yarats, D. Brandfonbrener, H. Liu, M. Laskin, P. Abbeel, A. Lazaric, and L. Pinto. Don't change the algorithm, change the data: Exploratory data for offline reinforcement learning. *CoRR*, abs/2201.13425, 2022.

[113] D. Yarats, R. Fergus, A. Lazaric, and L. Pinto. Reinforcement learning with prototypical representations. In *Proceedings of the 38th International Conference on Machine Learning*, pages 11920–11931, 2021.

[114] T. Yu, G. Thomas, L. Yu, S. Ermon, J. Zou, S. Levine, C. Finn, and T. Ma. Mopo: Model-based offline policy optimization. In *Advances in Neural Information Processing Systems*, 2020.

[115] T. Zahavy, B. O'Donoghue, A. Barreto, V. Mnih, S. Flennerhag, and S. Singh. Discovering diverse nearly optimal policies with successor features. *arXiv preprint arXiv:2106.00669*, 2021.

[116] C. Zhang, Y. Cai, L. Huang, and J. Li. Exploration by maximizing renyi entropy for reward-free RL framework. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 10859–10867. AAAI Press, 2021.

[117] T. Zhang, H. Xu, X. Wang, Y. Wu, K. Keutzer, J. E. Gonzalez, and Y. Tian. Noveld: A simple yet effective exploration criterion. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[118] Z. Zhou, W. Fu, B. Zhang, and Y. Wu. Continuously discovering novel strategies via reward-switching policy optimization. In *International Conference on Learning Representations*, 2022.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 4.3
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]