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

Yingchen Xu[*] UCL, FAIR

Jack Parker-Holder[*] University of Oxford

Aldo Pacchiano[*] Microsoft Research

Philip J. Ball[*] University of Oxford

Oleh Rybkin UPenn

Stephen J. Roberts University of Oxford

Tim Rocktäschel UCL

Edward Grefenstette UCL, Cohere

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. CASCADE achieves this by specifically maximizing the *diversity of trajectories* sampled by the population through a novel *cascading* objective. We provide theoretical intuition for CASCADE which we show in a tabular setting improves upon naïve approaches that do not account for population diversity. We then demonstrate that CASCADE collects diverse task-agnostic datasets and learns agents that generalize zero-shot to novel, unseen downstream tasks on Atari, MiniGrid, Crafter and the DM Control Suite. Code and videos are available at https://ycxuyingchen.github.io/cascade/

1 Introduction

Reinforcement learning (RL, [105]) has achieved a number of impressive feats over the past decade, with successes in games [69, 13, 100], robotics [48, 77], and the emergence of real world applications [11, 25]. 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 [90, 109, 68, 5].

In this work, we take steps towards building *generalist agents at scale*, where we outline two key desiderata. First, for agents to become generalists that can adapt to novel tasks, we eschew the notion of restricting agent learning to task-specific reward functions and focus on the *reward-free* problem setting instead [83, 28], whereby agents must discover novel skills and behaviors without supervision.[2] Consider the problem of learning to control robotic arms, where we may already have some expert offline data to learn from. In many cases this data will cover only a subset of the entire range of possible behaviors. Therefore, to learn additional general skills, it is imperative to collect additional novel and diverse data, and to do so 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 [60], without costly and lengthy centralized

[*] Equal contribution. Correspondence to ycxu@meta.com.

[2] Indeed, designing reward functions to learn behaviors can have unintended consequences [74, 6].

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

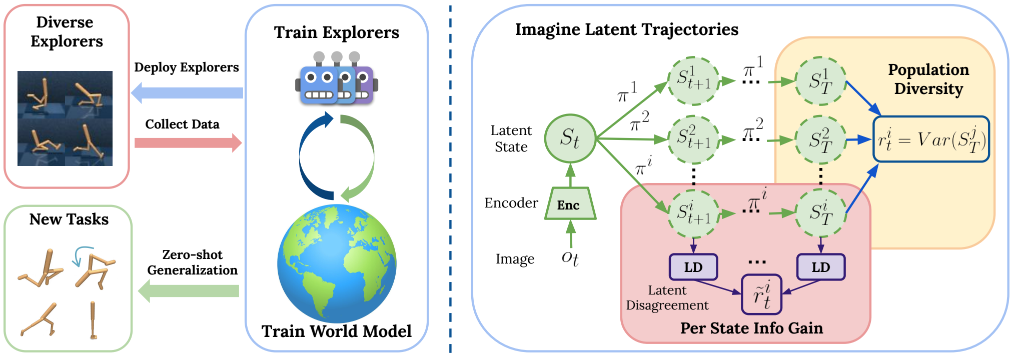

Figure 1: **Overview.** Left: CASCADE trains a population of diverse explorers and uses them to collect large batches of reward-free trajectories for learning a general world model that facilitates zero-generalization to novel tasks. Right: To train B exploration agents in parallel, at each training step t, CASCADE first infers a latent state s_t from image observation o_t. It then rolls out latent trajectories $\tau^1, \ldots, \tau^B$ in imagination using the current exploration policies $\pi^1, \ldots, \pi^B$. The training objective for each policy π^i is to optimize 1) the population diversity estimated by the disagreement of the final states of imagined trajectories sampled from policies $\pi^1, \ldots, \pi^i$; 2) the expected per state information gain over all future timesteps $t+1, \ldots, T$, computed as the disagreement of an ensemble of dynamics models.

retraining during this crucial phase [67]. This has recently been referred to as *deployment efficiency*, falling between the typical online/offline RL dichotomy. Limiting deployments not only reduces the overhead in retraining exploration policies, but also limits the potential costs and risks present when deploying new policies [108, 53], an important consideration in many real world settings, such as robotics [48], education [65] and healthcare [29].

Combining these two desiderata, we introduce the *reward-free deployment efficiency* setting, a new paradigm for deep RL research. To tackle this new problem we train a population of exploration policies to collect large quantities of useful data via *world models* [20, 34, 39]. World models allow agents to plan and/or train policies without interacting with the true environment. They have already been shown to be highly effective for deployment efficiency [67], offline RL [118, 49, 8, 88] and self-supervised exploration [99, 97]. Furthermore, world models offer the potential for increasing agent generalization capabilities [7, 37, 68, 16, 8, 21, 10, 58], one of the frontiers of RL research [80, 51]. 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 [42, 52], 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 (Figure 1). 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 four 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.

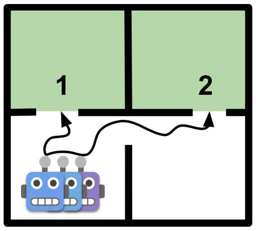

Figure 2: Motivation for CASCADE: Green areas represent high expected information gain. If we train a population of agents independently, at deployment time they will all follow the trajectory to #1, producing a homogenous dataset. However, if we consider the diversity of the data then we will produce 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 provide analysis that shows CASCADE theoretically improves sample efficiency over other naïve methods that do not ensure sample diversity, and demonstrate that CASCADE is capable of *improved zero-shot transfer* in four distinct settings, ranging from procedurally generated worlds to continuous control from pixels.

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, $R(s,a,s') \rightarrow r$ 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 [51], where observations, dynamics and rewards can vary given a context. In this paper we consider settings where only the reward varies, thus, if the test-time 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 [67], 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 [112], 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* [96]) 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, $\mathcal{M}_\psi = (\mathcal{S}, \mathcal{A}, P_\psi, R_\psi, \rho)$. We focus on Dyna-style MBRL [104], whereby we train a policy (π_θ parameterized by θ) with model-free RL solely using "imagined" transitions inside $\mathcal{M}_\psi$. Furthermore, we can train the policy on a single GPU with parallelized rollouts since the simulator is a neural network [54]. 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. Note that π_{EXP} need not be a single policy, but could also refer to a collection of policies that we can deploy (either in parallel or in series), such that $\pi \in \pi_{\text{EXP}}$.

Algorithm 1 Reward-Free Deployment Efficiency via World Models

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

We focus on learning world models from high dimensional sensory inputs such as pixels [34, 76, 47], 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 [39, 38, 40]. In this paper we will make use of the model from DreamerV2 [40], 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, [50, 92]). For zero-shot evaluation, we follow [97] 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 offine. Thus, it is critical that our dataset is sufficiently diverse to enable learning novel, unseen behaviors.

3 Coordinated Active Sample Collection

The aim of this work is to train a population of B exploration policies $\{\pi_{\text{EXP}}^{(i)}\}_{i=1}^{B}$ 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 [97], but crucially focus on maximizing information gain over *entire trajectories* rather than per state-action, and hence drop the conditional dependence on state and action (see App. C.1 for why this distinction is important):

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

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). We think of each model $\mathcal{M}_\psi$ as sampled from a posterior distribution over models given the data. A good exploration policy has low entropy on individual models but large entropy across models, i.e. it is intent in visiting regions of the space where there is large uncertainty about the model transitions. To make this objective more general, we represent the trajectory data collected by a policy with a "summary" embedding space [79, 72]. 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 [85]. 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 [94, 103].

3.1 A Cascading Objective with Diverse Explorers

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

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

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:

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

It is now possible to show that maximum information gain is achieved with a *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 $\{\pi_{\text{EXP}}^{(i)}\}_{i=1}^{B}$ satisfying $\pi_{\text{EXP}}^{(i)} \neq \pi_{\text{EXP}}^{(j)} \forall i \neq j$. 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 C.1.1. Since the mutual information objective 3 is submodular, a greedy algorithm yields a $(1 - \frac{1}{e})$ approximation of the optimum (where e is Euler's number) [73]. 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:

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

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

3.2 A Tractable Objective for Deep RL

Inspired by [97], 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 final state embedding distributions are Gaussian with means that depend on the policies and sampled worlds, and variances that depend on the worlds, i.e. $\mathbb{P}_{\pi}^{\Phi}[\mathcal{M}_{\psi} = w] = \mathcal{N}(\mu(w, \pi), \Sigma(w))$. In this case, $\mathcal{H}(\mathbb{P}_{\pi}^{\Phi}[\mathcal{M}_{\psi}] | \mathcal{M}_{\psi} = w) = \rho(w)$, and Eq. 5 reduces to solving $\pi^{(i)} = \arg\max_{\tilde{\pi}^{(i)} \in \Pi} \mathcal{H}\left(\prod_{j=1}^{i} \mathbb{P}_{\tilde{\pi}^{(j)}}^{\Phi}[\mathcal{M}_{\psi}] \middle| \tilde{\pi}^{(j)} = \pi^{(j)} \ \forall j \leq i-1\right)$ 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:

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

where $\mathcal{D}^{(i-1)}$ is a dataset of imagined trajectories sampled from policies $\pi^{(1)}, \cdots, \pi^{(i-1)}$ in the model, and PopDiv is short for Population Diversity. Finally, following [97, 9], we also add a per state 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 (PopDiv) 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:

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

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 [23].

3.3 Theoretical Motivation

We now seek to provide a tabular analogue to CASCADE which provides a theoretical grounding for our approach. In App. D we outline the pseudo-code of CASCADE-TS, a greedy Thompson Sampling algorithm [4] 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. D. Indeed, 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 $\{\pi^{(1)}, \cdots, \pi^{(i-1)}\}$ (i.e. the PopDiv term) 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 three baselines. The first is random exploration, which is used in the majority of RL publications. Our next baseline is Plan2Explore [97] (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 [40] and use the same hyperparameters for model and agent training (more details in App. B).

We test all methods in two separate settings: Firstly *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" [78, 75]). Secondly, 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 [97, 55, 116].

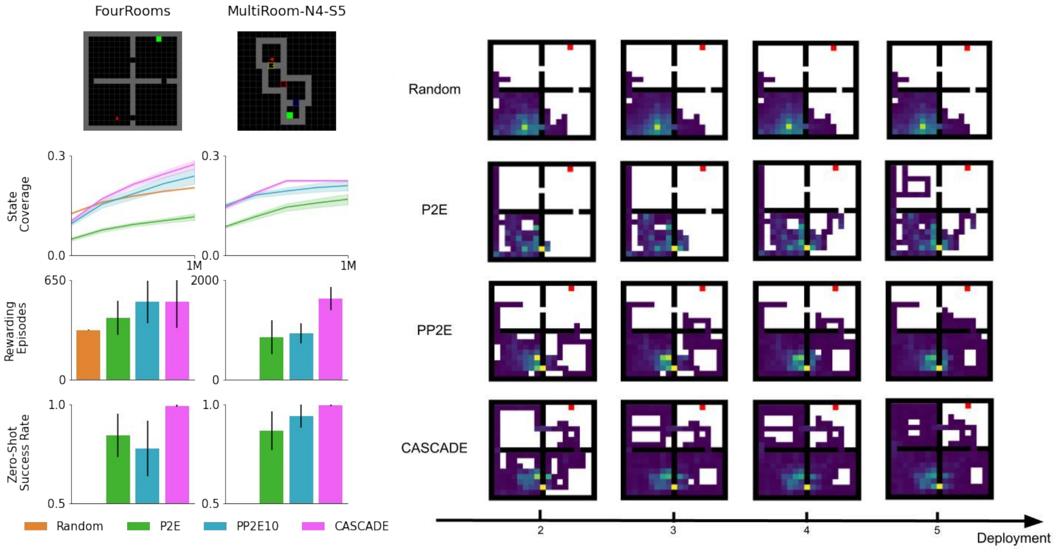

Figure 3: **MiniGrid Results**. Left: Performance statistics for both FourRooms and MultiRoom. For reward-free exploration we show both "state coverage", which corresponds to the the percentage of states visited on a fixed set of levels, and "rewarding episodes" which is the count of total solved episodes 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).

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 three different environments: MiniGrid [19], Atari [12] and Crafter [37]. In all three 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 [89, 17, 31, 121]. MiniGrid environments are procedurally generated, which provides an additional generalization challenge [46, 93]. 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.[3] In order to gauge the effectiveness of the reward-free exploration, we use two metrics: 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. However, in combination they provide a proxy for exploration.

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 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 four Atari games: Montezuma's Revenge, Frostbite, Hero and Freeway, often used as barometers for exploration capabilities [27]. We use 15 reward-free deployments, each consisting of 200k steps, and use the final model to train a behavior policy with task reward. We test these agents in a zero-shot fashion, with results shown in the first three columns of Figure 4. We see that CASCADE gets statistically significantly stronger zero-shot returns compared to the baselines. Furthermore, the baseline methods display no clear trend, with random exploration sometimes offering the strongest performance. On the other hand, we see that CASCADE performs consistently well, providing further evidence of it being a strong general exploration strategy. See App. A.2 for more detailed analysis.

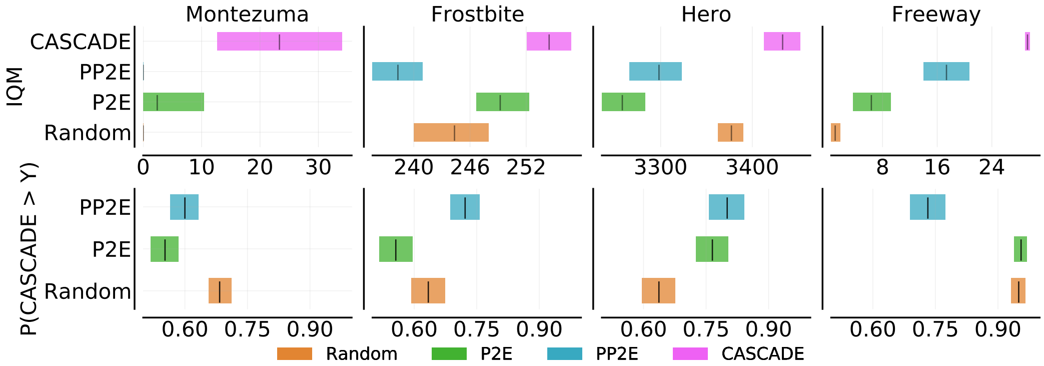

Figure 4: **Atari Zero-Shot Transfer**: Plots show RLiable metric performance aggregated across all seeds. Note that the 'Probability of Improvement' lower CIs all exceed 0.5 for Atari, indicating CASCADE provides a statistically significant improvement (under the Mann-Whitney U test) over all baseline methods [3]. We show five seeds of zero-shot test performance.

Finally, we consider the Crafter environment [37], a procedurally generated grid world based on the game of Minecraft. This more complex setting requires the agent to master a variety of compositional skills to fully explore all possible behaviors. Online P2E represents the state of the art reward-free agent, achieving a "Crafter Score" of 2.1. Making the problem more challenging, we deploy only 20 distinct exploration policies, each collecting 50k timesteps, which leads to weaker performance for P2E vs. the reported result in [40]. However, as we see in Figure 5, adding a population of diverse explorers recovers the majority of this performance. We find this result encouraging as the behaviors required to achieve a higher Crafter score are non-trivial. Moreover, as shown in Figure 6, CASCADE achieves the highest average success rate and unlocks the most number of unique Crafter skills among all baselines, including Making a Stone Pickaxe. We believe this opens the door to further gains with more active tuning of the behavioral representation used in CASCADE.

[3] 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.

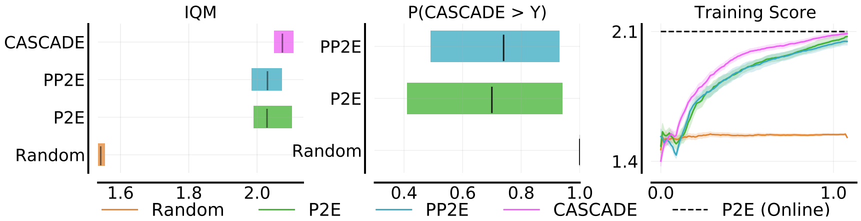

Figure 5: **Crafter Score**: Plots show final RLiable metric performance and training curve over 1M steps aggregated across all seeds. We show ten seeds of Crafter Score, the geomean of skills discovered in a purely reward-free fashion. Again, see that the 'Probability of Improvement' lower CIs all exceed 0.5, indicating CASCADE provides a statistically significant improvement (under the Mann-Whitney U test) over all baseline methods [3].

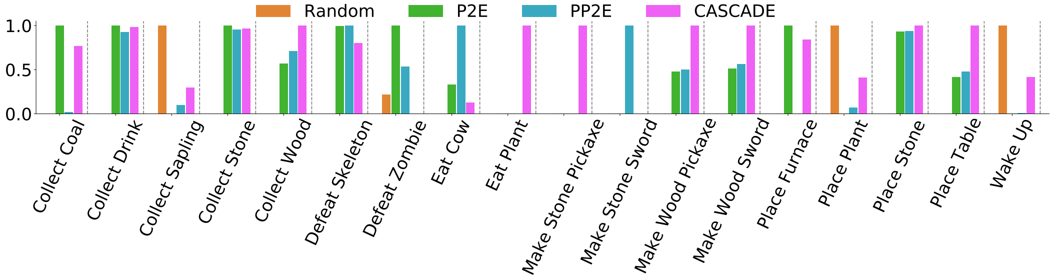

Figure 6: **Crafter Skills**: Plots show normalized success rate per skill. Note that CASCADE not only achieves the highest average success rate but also unlocks the most (16 out of 22) unique skills among all baselines. Skills that are not unlocked by any of the baselines are excluded from the plot.

4.2 Exploring Behaviors

Next we wish to test is whether our agents sufficiently explore a range of behaviors in a continuous control setting, using the `walker` environment from the DM Control Suite [111] (DMC). In these experiments, we operate from pixels and assume access to an initial dataset of 200k transitions. Then, we conduct 14 further reward-free deployments, each collecting 200k transitions. We consider the possibility of improving generality from arbitrary offline data, using common datasets from the offline RL literature: `random`, `medium` and `expert`. Using random data represents learning from scratch. In Figure 7 we show the behaviors deployed by CASCADE and PP2E in a single deployment. The CASCADE explorers all display diverse, useful behaviors, while PP2E agents are homogenous.

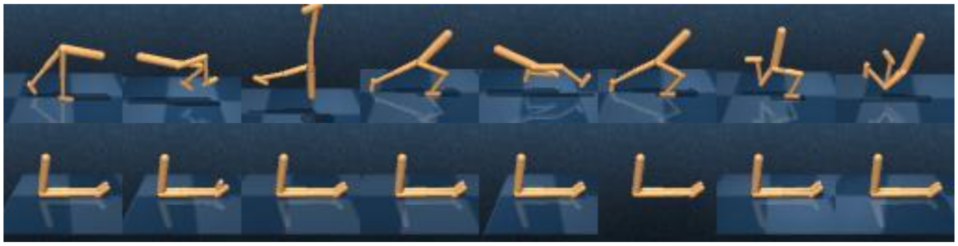

Figure 7: **DMC Qualitative Results**: Each row visualizes behaviors of a population of explorers trained after 15 total deployments by the same world model, deployed at the same time in environments with the same initial state. We can see that CASCADE explorers (top) collect data with a diversity of behaviors while PP2E agents (bottom) only show less interesting homogenous behaviors.

We then test the generality of the world model after each deployment by training a separate agent from scratch for each of the four individual tasks: `stand`, `walk`, `run`, `flip` proposed in URLB [56]. This demonstrates whether the world model is capable of producing an imaginary MDP that facilitates learning specific behaviors. To compare the performance of CASCADE and the baselines, in Figure 8 we show the aggregated statistics from the RLiable library [3] for a given number of deployments. 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 [3]); 2) the Probability of Improvement, the likelihood of one method outperforming another on a new, unseen task. As we see, CASCADE shows clear and consistent gains over the next best baseline (PP2E) throughout 15 deployments, showing

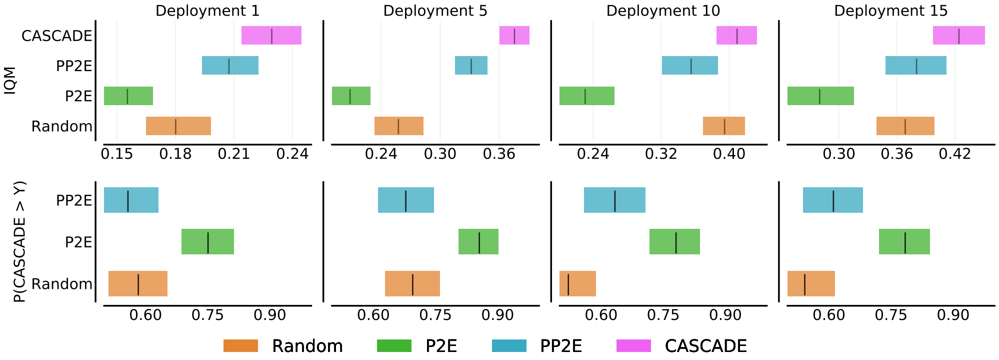

Figure 8: **DMC Zero-Shot Aggregate Results**: Plots show RLiable metric performance aggregated across all three initial datasets (`random`, `medium`, `expert`) for a total of 30 seeds. Note that the 'Probability of Improvement' lower CIs all exceed 0.5, indicating CASCADE provides a statistically significant improvement (under the Mann-Whitney U test) over all baseline methods [3], even after a single deployment.

statistical significance under the Mann-Whitney U test [66]. Note that here the single agent P2E performs poorly, likely due to only collecting data with a single mode of exploratory behavior. See Figure 9 in the Appendix for a full breakdown of the performance.

4.3 Discussion and Limitations

We have shown that CASCADE is the most effective method in training generally capable agents across a wide range of environments. Indeed, 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 PopDiv). Each of our baselines lacked in one of these two key properties. 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 [36], but less so in deep exploration tasks [75]. Furthermore, the information theoretic baselines were effective in Exploring Worlds, but the disappointing performance for P2E in DMC is likely caused by a lack of data diversity, while the PP2E populations were 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 [90, 68, 109, 110]. 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 [45]. 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, 33, 28, 59, 62, 101] to data-based that aim to maximize data diversity [63, 117].

In this work we take inspiration from *knowledge-based* approaches, which typically seek to learn a world model [9, 83, 84, 30, 107, 15, 98]. In particular we build on Plan2Explore [97] 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 [116, 55]. 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 [67]. 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 [102], whereas we are focused on training generalist agents in the reward-free paradigm. Recent work has also shown the importance of frequent policy updates in ensuring diverse exploration [95], further supporting the need for a diverse population of explorers when operating in the deployment-efficient setting.

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 [106, 26, 64], 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, [85, 24]), where it has been shown that diversity can boost exploration in RL [22]. Recently, [61] 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 [82] or sequentially [122, 119].

Also related are works that specifically train policies that are able to maximally cover the state space of a given MDP [2, 70]. However, these works focus on the more theoretically amenable (and less general) low-rank MDP setting, and hence do not demonstrate any experiments on larger scale problems. Furthermore, we aim to achieve state-space coverage by encouraging *policy diversity*, which is distinct from the approaches taken in these works. We make use of an information theoretic objective, which has been shown to be highly effective for exploration in RL [71, 43, 41]. Most similar to our work [120] 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 [35, 115], 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. We believe this work provides a new approach for collecting large, diverse datasets in a self-supervised fashion, inducing a potentially open-ended process for training general world models at scale.

For future work, we could consider alternative means for computing diversity, for example using optimization landscapes [81] or explicit behavioral diversity [28, 82]. These could also be combined with representations that directly incorporate the behaviors of the policy [14] to further improve the design of the "summary" embedding space. Furthermore, we would be interested in considering transformer models [113] for trajectory modelling [18, 44] that ought to scale more gracefully with larger batches of data [87]. Recent work has shown that this class of models can compose unseen complex multi-task behaviors at test time from fixed offline datasets [32], as well as leverage significantly large and diverse offline datasets to improve behavioral modeling [91], highly relevant to both aforementioned desiderata for the reward-free deployment efficiency paradigm.

Acknowledgments

The authors would like to thank Cong Lu for help with Offline DreamerV2 code, and Minqi Jiang and Robert Kirk for useful discussions. This work was funded by Meta AI.

References

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

[2] A. Agarwal, S. Kakade, A. Krishnamurthy, and W. Sun. Flambe: Structural complexity and representation learning of low rank mdps. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20095–20107. Curran Associates, Inc., 2020.

[3] 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.

[4] 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.

[5] 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.

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

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

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

[20] K. Chua, R. Calandra, R. McAllister, and S. Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems 31*, pages 4754–4765, 2018.

[21] 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.

[22] 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.

[23] 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.

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

[25] 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.

[26] 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.

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

[28] 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.

[29] 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.

[30] 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.

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

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

[33] 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.

[34] 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.

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

[36] 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.

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

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

[39] 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.

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

[41] 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.

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

[43] 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.

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

[45] 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.

[46] 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.

[47] 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.

[48] 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.

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

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

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

[52] 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.

[53] 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.

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

[55] 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.

[56] 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.

[57] 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.

[58] K. Lee, Y. Seo, S. Lee, H. Lee, and J. Shin. Context-aware dynamics model for generalization in model-based reinforcement learning. In *International Conference on Machine Learning*, 2020.

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

[60] 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.

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

[62] 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.

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

[64] 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.

[65] 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.

[66] 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.

[67] 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.

[68] 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.

[69] 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.

[70] A. Modi, J. Chen, A. Krishnamurthy, N. Jiang, and A. Agarwal. Model-free representation learning and exploration in low-rank mdps, 2021.

[71] 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.

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

[73] 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.

[74] 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.

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

[76] 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.

[77] 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.

[78] 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.

[79] 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.

[80] 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.

[81] 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.

[82] 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.

[83] 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.

[84] 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.

[85] 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.

[86] 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.

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

[88] 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.

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

[90] 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.

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

[92] 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.

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

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

[95] T. Schaul, A. Barreto, J. Quan, and G. Ostrovski. The phenomenon of policy churn, 2022.

[96] 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.

[97] 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.

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

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

[100] 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.

[101] 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.

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

[103] 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.

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

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

[106] 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.

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

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

[109] 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.

[110] 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.

[111] 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.

[112] H. van Hasselt, M. Hessel, and J. Aslanides. When to use parametric models in reinforcement learning? In *Advances in Neural Information Processing Systems*, pages 14322–14333, 2019.

[113] 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.

[114] 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.

[115] 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.

[116] 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.

[117] 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.

[118] 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.

[119] T. Zahavy, Y. Schroecker, F. Behbahani, K. Baumli, S. Flennerhag, S. Hou, and S. Singh. Discovering policies with domino: Diversity optimization maintaining near optimality. *arXiv preprint arXiv:2205.13521*, 2022.

[120] 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.

[121] 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.

[122] 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]

Appendix

A Additional Experimental Results

A.1 DMC

Here we show additional experimental results, beginning with a full breakdown of the DMC results for each dataset. In Figure 9 we see the zero-shot transfer performance for models trained with `random`, `medium` and `expert` initial datasets, and 14 subsequent deployments. Interestingly, we see that the `medium` dataset proves to be the most effective for achieving high performance, this may be due to being more diverse than the `expert` dataset, which is a narrow distribution of high performing episodes.

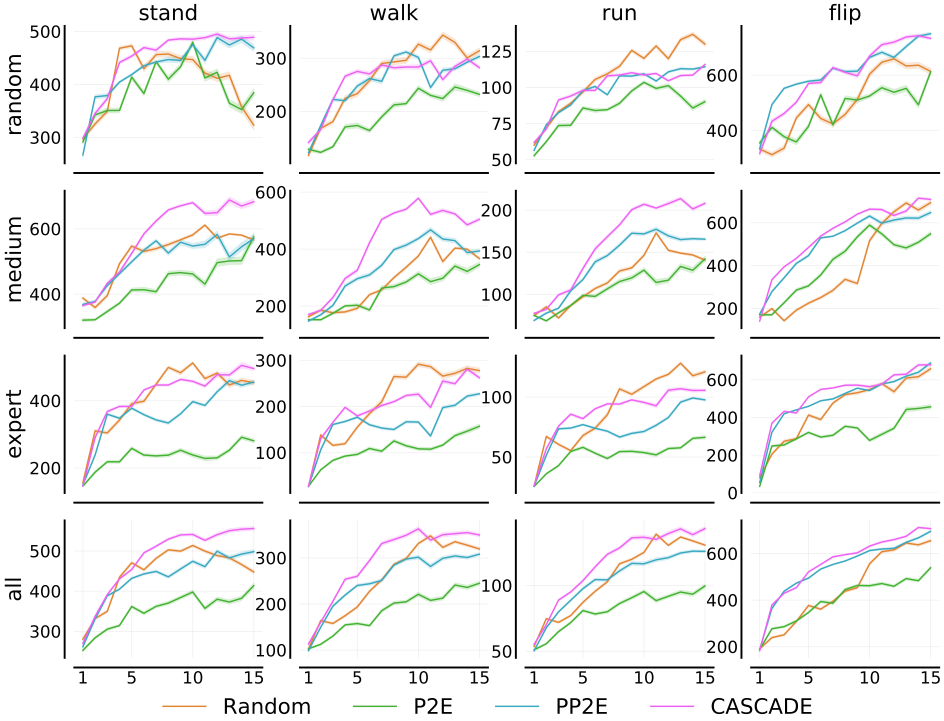

Figure 9: **DMC Zero-Shot Results**: Plots show the performance by environment and dataset, with the last row being the task performance average over all three initial datasets. Plots show the mean and SEM over 10 seeds.

A.2 Atari Tasks

Next we show detailed results in four Atari games: Montezuma's Revenge, Frostbite, Hero and Freeway in Fig. 10. We see that generally CASCADE discovers higher rewarding episodes (almost double the next best baseline in Montezuma's Revenge), and also discovers more rewarding episodes on average. This translates to stronger zero-shot performance than all baselines in all four environments. On the other hand, the other baselines achieve worse zero-shot performance because they either (1) find an occasional high reward episode (Max Episode Return), but not in a sufficient quantity (Rewarding Episodes), or (2) collect abundant low rewarding episodes that are less helpful for learning good behavior polices. For instance, the random baseline is able to find Rewarding Episodes more frequently in Frostbite, but these are relatively low quality, hence the lower curve when assessing its Max Episode Return.

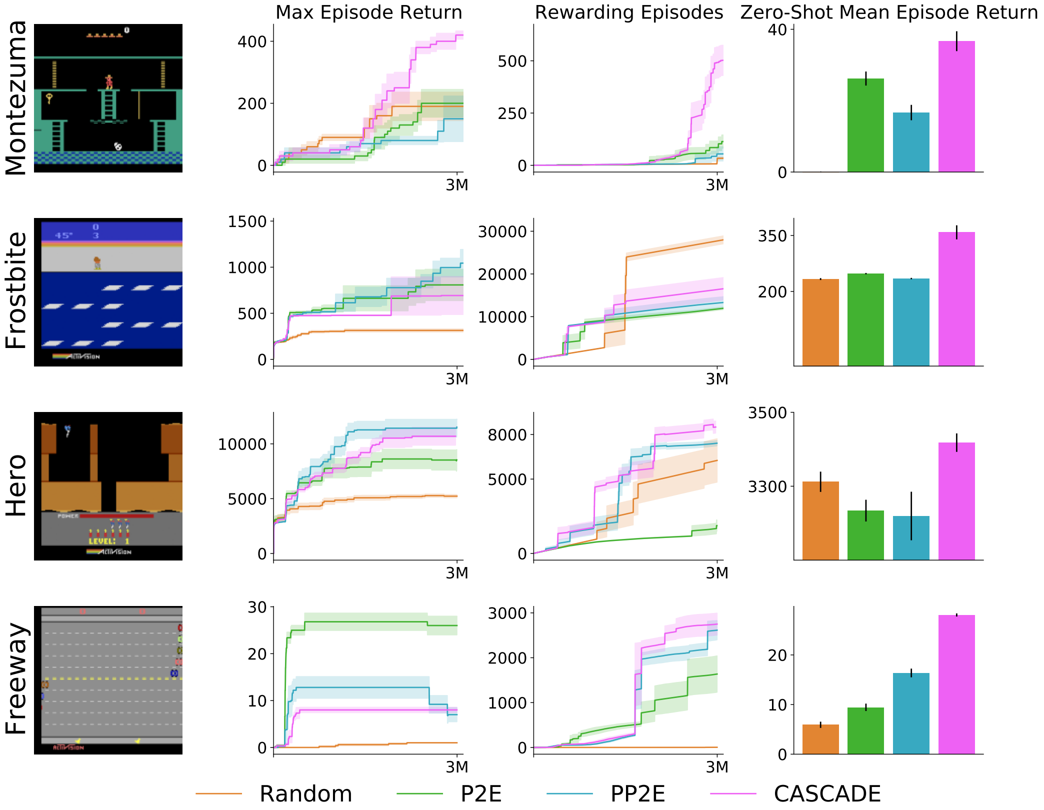

Figure 10: From left to right on each row we show a frame from the game, a plot of the cumulative maximum episode reward, the total number of rewarding episodes discovered and zero-shot average return, from 3M training steps (15 deployments). Plots show mean and SEM over 10 seeds. Note that the max episode returns of all methods stabilize after 15 deployments (3M training steps in total).

To understand why CASCADE performs well, in Figure 11 we plot trajectories from Montezuma's Revenge. We see that the inclusion of a diversity-inducing objective does indeed lead to more diverse behaviors in the environment, with each policy exploring different regions of the room. In contrast, for PP2E, agents 3,4,5,6,7,9 all exhibit very similar behavior, likely indicating that random initialization alone is not sufficient to induce diversity when all agents optimize for the same objective.

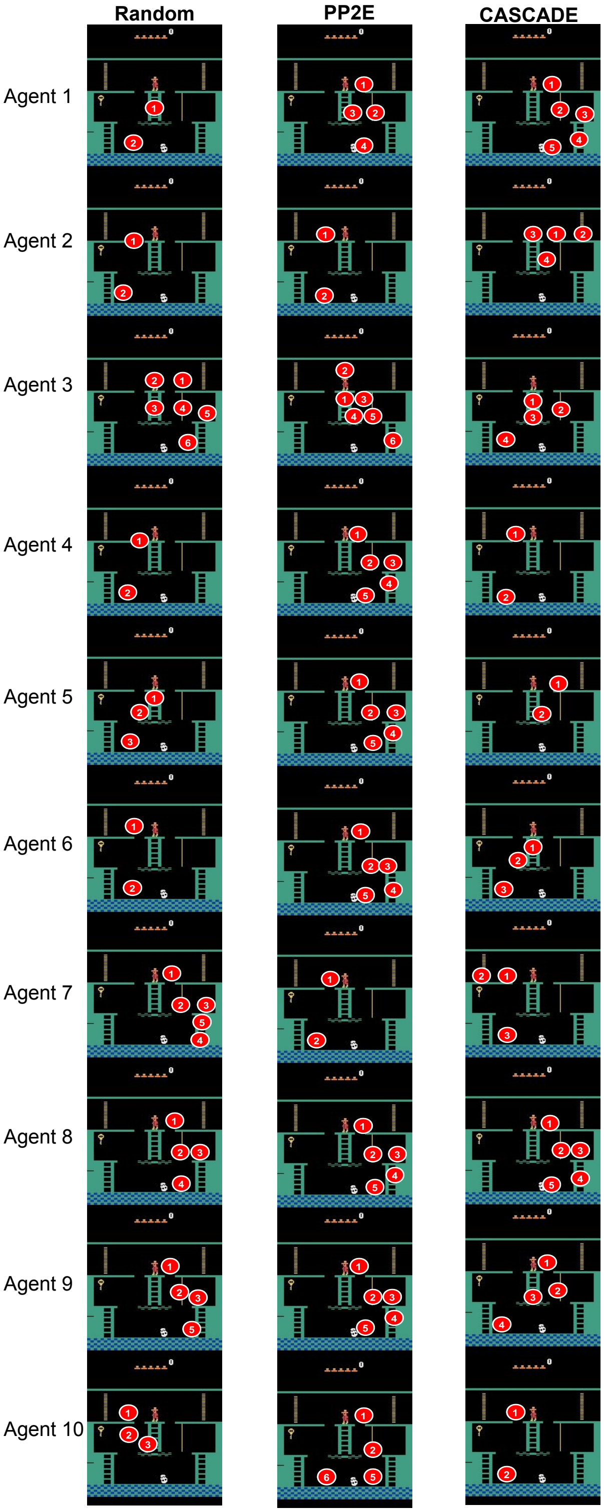

Figure 11: **Trajectories in Montezuma's Revenge**: Each row shows a trajectory from one of the 10 exploration agents of a model after 3M training timesteps (P2E is omitted because it only has 1 agent.) We can see that PP2E's agents exhibit the most homogeneous behaviors, and result in trajectories that focus on the bottom right of the room, while CASCADE agents manage to explore more of the room collectively.

B Implementation Details

DreamerV2 consists of an image encoder that uses a Convolutional Neural Network (CNN, [57]), a Recurrent State-Space Model (RSSM [39]) that learns the dynamics, and predictors for the image, reward, and discount factor. The RSSM uses a sequence of deterministic recurrent states h_t. At each step, it computes a posterior state z_t conditioned on the current image x_t, as well as a prior state $\hat{z}_t$ without the current image. During world model training, the concatenation of h_t and z_t is used to reconstruct the current image x_t, and predict the reward r_t and discount factor γ_t. Once the world model is trained, it can be used to roll out "imaginary" trajectories where the model state is instead the concatenation of the deterministic state h_t and prior stochastic state $\hat{z}_t$.

$$\text{RSSM}\begin{cases}\text{Recurrent model:} & h_t = f_\psi(h_{t-1}, z_{t-1}, a_{t-1})\\ \text{Representation model:} & z_t \sim q_\psi(z_t | h_t, x_t)\\ \text{Transition predictor:} & \hat{z}_t \sim p_\psi(\hat{z}_t | h_t)\end{cases} \tag{7}$$

$$\begin{aligned}&\text{Image predictor:} && \hat{x}_t \sim p_\psi(\hat{x}_t | h_t, z_t)\\ &\text{Reward predictor:} && \hat{r}_t \sim p_\psi(\hat{r}_t | h_t, z_t)\\ &\text{Discount predictor:} && \hat{\gamma}_t \sim p_\psi(\hat{\gamma}_t | h_t, z_t).\end{aligned} \tag{8}$$

Our implementation was built on top of the official DreamerV2 repository `https://github.com/danijar/dreamerv2`. We used the default hyperparameter values in the DreamerV2 repository. Table 1 lists the additional hyperparameters used in our experiments.

We also incorporate a latent disagreement ensemble, following Plan2Explore [97]. This involves training, alongside the RSSM model, an MLP ensemble with 10 members (each having 4 hidden layers, 400 units per layer, and different parameter initializations). These one-step models take action a_t and latents z_t, h_t as inputs, and try to predict the next stochastic latent z_{t+1}. The variance across these outputs during imagined rollouts is then used as the reward signal that forms the Information Gain (InfoGain) objective. Note that this ensemble is otherwise unused, and is solely trained for the purposes of the exploration objective.

Table 1: CASCADE hyperparameters

Environment	Parameter	Value
MiniGrid-FourRooms	CASCADE weight (λ)	0.1
	batch size	200k
	deployments	5
	explorer train steps	10k
	offline model train steps	20k
MiniGrid-MultiRoom-N4S5	CASCADE weight (λ)	0.3
	batch size	200k
	deployments	5
	explorer train steps	10k
	offline model train steps	20k
Montezuma's Revenge	CASCADE weight (λ)	0.8
	batch size	200k
	deployments	15
	explorer train steps	10k
	offline model train steps	5k
Walker	CASCADE weight(λ)	0.1 (Random) 0.3 (Medium/Expert)
	batch size	200k
	deployments	2
	explorer train steps	250
	offline model train steps	5k

C Deriving the Objective

C.1 An Information Theoretic Perspective

Inspired by [97], we derive a data acquisition objective based on the mutual information between the collected data and parameters of the world model $\mathcal{M}_\psi$. $\mathcal{M}_\psi$ can be either stochastic or deterministic, and can also be an ensemble of empirical models. Crucially however, we eschew the reliance on one-step information gain (as performed in [97]), and instead aim to maximize diversity directly in the space of trajectories. To do this, let $\Phi : \Gamma \to \Omega$ be an summary function mapping trajectories into an embedding 'summary' space. For any model $\mathcal{M}_\psi$ we use the notation $\mathbb{P}_\pi[\mathcal{M}_\psi]$ to denote the distribution over trajectories generated by policy π in $\mathcal{M}_\psi$ and use $\mathcal{M}_\psi \sim \mathbb{P}_{\mathcal{M}_\psi}$ to denote the posterior distribution over models. The later depends on the data collected so far.

In this work we consider different embedding functions Φ,

1. **Final State Embedding**. In this setting we define $\Phi(\tau) = h_H$ where h_H corresponds to the $H-$th (and last) state embedding in the trajectory τ. Since we use an RNN world model, the use of this summary embedding is analogous to the final encoder embedding in $\mathrm{seq2seq}$ language models [103].
2. **Visitation Frequencies**. In the tabular setting if we define $\Phi(\tau)$ as a discounted count statistic for states in τ, we recover $\mathbb{P}^\Phi_\pi[\mathcal{M}_\psi] = d^\pi_{\mathcal{M}_\psi}$ where $d^\pi_{\mathcal{M}_\psi}$ corresponds to the discounted visitation distribution of policy π in model $\mathcal{M}_\psi$.

The authors of [97] study a per state-action mutual information objective that informs the construction of a greedy mutual information maximizing policy. Implicitly, this per-step objective assumes the model to factor in *independent* transition operators pertaining to each state-action pair. This is certainly not the case when using powerful function approximators such as neural networks. In this case, the dynamics model corresponding to the state transitions of a specific state action pair is correlated with the state transition model of *other* state-actions. This is not captured accurately by the objective in [97]. Instead, we consider the following choice for an exploratory policy in the single policy setting studied in [97]:

$$\pi_{\mathrm{EXP}} = \max_\pi \mathcal{I}\left(\mathbb{P}^\Phi_\pi[\mathcal{M}_\psi]; \mathcal{M}_\psi\right) = \mathcal{H}(\mathbb{P}^\Phi_\pi[\mathcal{M}_\psi]) - \mathcal{H}(\mathbb{P}^\Phi_\pi[\mathcal{M}_\psi]|\mathcal{M}_\psi). \tag{9}$$

π_{EXP} is a policy whose distribution over summaries $\Phi(\tau)$ with $\tau \sim \mathbb{P}_\pi[\mathcal{M}_\psi]$ and $\mathcal{M}_\psi \sim \mathbb{P}_{\mathcal{M}_\psi}$ has large entropy, but such that the average entropy of the summaries in every individual model is small. The term $\mathcal{H}(\mathbb{P}^\Phi_\pi[\mathcal{M}_\psi])$ captures the total uncertainty (epistemic + aleatoric) while subtracting the conditional entropy removes the aleatoric uncertainty resulting from noise within the model and the policy.

To gain intuition about this objective, consider the case where Φ equals the *final state embedding*. For simplicity, we consider the scenario when all models $\mathcal{M}_\psi \sim \mathbb{P}$ and all policies $\pi \sim \Pi$ are such that for any realization of $\mathcal{M}_\psi = m_\psi$, the conditional entropy $\mathcal{H}(\mathbb{P}^\Phi_\pi[\mathcal{M}_\psi]|\mathcal{M}_\psi = m_\psi) = \sigma(m_\psi)$ is a function of the world m_ψ and not of the policy. In this case, the policy π_{EXP} is the entropy maximizing policy:

$$\pi_{\mathrm{EXP}} = \max_\pi \mathcal{H}(\mathbb{P}^\Phi_\pi[\mathcal{M}_\psi]).$$

The scenario above is realized for example when all models $\mathcal{M}_\psi \sim \mathbb{P}_{\mathcal{M}_\psi}$ are deterministic and all policies in Π are deterministic. In this case, for every realization of $\mathcal{M}_\psi = m_\psi$, the conditional entropy $\mathcal{H}(\mathbb{P}^\Phi_\pi[\mathcal{M}_\psi]|\mathcal{M}_\psi = m_\psi) = 0$. The assumption $\mathcal{H}(\mathbb{P}^\Phi_\pi[\mathcal{M}_\psi]|\mathcal{M}_\psi = m_\psi) = \sigma(w)$ is also realized when the distribution $\Phi(\tau) \sim \mathbb{P}^\Phi_\pi[m_\psi]$ is approximated as a Gaussian distribution $\mathbb{P}^\Phi_\pi[m_\psi] \approx \mathcal{N}(\mu(m_\psi, \pi), \Sigma(m_\psi))$ with policy dependent mean $\mu(m_\psi, \pi)$ and policy independent covariance $\Sigma(m_\psi)$.

In these cases π_{EXP} is the policy that produces the most diverse distribution over final states across the posterior distribution over models $\mathbb{P}_{\mathcal{M}_\psi}$. Since access to $\mathcal{H}(\mathbb{P}^\Phi_\pi[\mathcal{M}_\psi])$ may not be possible, in our experiments we make further approximations inspired by [97]. Under the same gaussian approximation $\mathbb{P}^\Phi_\pi[m_\psi] \approx \mathcal{N}(\mu(m_\psi, \pi), \Sigma(m_\psi))$, optimizing $\max_\pi \mathcal{H}(\mathbb{P}^\Phi_\pi[\mathcal{M}_\psi])$ is achieved by finding the policy π that makes the ensemble means $\mu(m_\psi, \pi)$ as far apart as possible. A suitable surrogate for this objective is to maximize the empirical variance over means,

$$\text{Var}(\pi) = \frac{1}{|\mathcal{M}_\psi| - 1} \sum_{m_\psi} \|\mu(m_\psi, \pi) - \mu'(\pi)\|^2$$

where $|\mathcal{M}_\psi|$ denotes the number of samples m_ψ, and $\mu'(\pi) = \frac{1}{|\mathcal{M}_\psi|} \sum_{m_\psi} \mu(m_\psi, \pi)$. We now consider a "batch" version of Eq. 9 involving a population of B agents:

$$\begin{aligned}\{\pi^{(i)}_{\text{EXP}}\}_{i=1}^{B} = \max_{\pi^{(1)},\cdots,\pi^{(B)} \in \Pi^B} \mathcal{I}\left(\prod_{i=1}^{B} \mathbb{P}^{\Phi}_{\pi^{(i)}}[\mathcal{M}_\psi]; \mathcal{M}_\psi\right) = \\ \mathcal{H}\left(\prod_{i=1}^{B} \mathbb{P}^{\Phi}_{\pi^{(i)}}[\mathcal{M}_\psi]\right) - \mathcal{H}\left(\prod_{i=1}^{B} \mathbb{P}^{\Phi}_{\pi^{(i)}}[\mathcal{M}_\psi] \middle| \mathcal{M}_\psi\right) \qquad (10)\end{aligned}$$

where $\prod_{i=1}^{B} \mathbb{P}^{\Phi}_{\pi^{(i)}}[\mathcal{M}_\psi]$ is the product measure of the candidate policies embedding distributions over the model $\mathcal{M}_\psi$. By definition the conditional entropy factors as,

$$\mathcal{H}\left(\prod_{i=1}^{B} \mathbb{P}^{\Phi}_{\pi^{(i)}}[\mathcal{M}_\psi] \middle| \mathcal{M}_\psi\right) = \sum_{i=1}^{B} \mathcal{H}\left(\mathbb{P}^{\Phi}_{\pi^{(i)}}[\mathcal{M}_\psi] \middle| \mathcal{M}_\psi\right). \qquad (11)$$

The objective in Eq. 10 has a similar interpretation as in the single policy case. We are hoping to find a set of policies whose average conditional entropies are small (Eq. 4), but whose total entropies are large. Intuitively, we see this objective is more amenable to our population of policies. Concretely, by considering distributions directly over the space of trajectories, we ensure that each agent does not 'double count' the explored states under Eq. 10, whereas applying the same principal to a one-step information gain objective would simply ensure diversity *conditioned* on that state and action, and doesn't explicitly ensure diversity in the visited states by the population.

C.1.1 Proof of Lemma 1

In the proof of Lemma 1 we will make use of the following supporting result,

Lemma 3. *Let $p \in (0, 1]$ and $p_1, p_2 > 0$ satisfying $p_1 + p_2 = p$ then,*

$$p \log(1/p) < p_1 \log(1/p_1) + p_2 \log(1/p_2).$$

Proof. Let $p_1 = \alpha p$ with $\alpha \neq 0, 1$, then,

$$\begin{aligned}p_1 \log(1/p_1) + p_2 \log(1/p_2) &= \alpha p \log(1/(\alpha p)) + (1-\alpha)p \log(1/((1-\alpha)p)) \\ &= p \log(1/p) + p \left(\underbrace{\alpha \log(1/\alpha) + (1-\alpha)\log(1/(1-\alpha))}_{>0}\right) \\ &> p \log(1/p).\end{aligned}$$

□

Lemma 3 implies the following result,

Lemma 4. *Let $\mathbf{p} \in [0, 1]^K$ be a vector satisfying $\sum_{i=1}^{K} \mathbf{p}_i = 1$ and let $\mathcal{C}$ be a partition of $[K]$ such that $|\mathcal{C}| \leq K - 1$. For all $C \in \mathcal{C}$ denote $\mathbf{p}(C) = \sum_{i \in C} \mathbf{p}_i$. The following inequality holds,*

$$\sum_{C \in \mathcal{C}} \mathbf{p}(C) \log(1/\mathbf{p}(C)) \leq \sum_{i=1}^{K} \mathbf{p}_i \log(1/\mathbf{p}_i).$$

If $\mathbf{p}_i > 0$ *the inequality is strict,*

$$\sum_{C \in \mathcal{C}} \mathbf{p}(C) \log(1/\mathbf{p}(C)) < \sum_{i=1}^{K} \mathbf{p}_i \log(1/\mathbf{p}_i).$$

Proof. W.l.o.g we call C_1 the partition set containing 1 and assume $\mathbf{p}_1 > 0$. If $\mathbf{p}_i > 0$ for only one element of C_1, it must be the case that $\mathbf{p}_1 \log(1/\mathbf{p}_1) = \mathbf{p}_{C_1} \log(1/\mathbf{p}_{C_1})$.

We now assume that $|C_1| > 1$ and that C_1 has at least two indices $i \neq j$ such that $\mathbf{p}_i, \mathbf{p}_j > 0$. Let l denote the size of C_1 and w.l.o.g. let's say $C_1 = \{1, 2, \cdots, l\}$. Lemma 3 implies the following inequalities,

$$\begin{aligned}
\mathbf{p}(C_1)\log(1/\mathbf{p}(C_1)) &< \mathbf{p}_1 \log(1/\mathbf{p}_1) + \mathbf{p}(C_1 \backslash \{1\}) \log(1/\mathbf{p}(C_1 \backslash \{1\})) \\
&< \cdots \\
&< \sum_{i \in C_1} \mathbf{p}_i \log(1/p_i).
\end{aligned}$$

Applying this reasoning to each element in the partition $\mathcal{C}$ yields the desired result.

□

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.*

Proof. First let's observe that when the models and policies are all deterministic, the conditional entropies satisfy,

$$\mathcal{H}\left(\prod_{i=1}^{B} \mathbb{P}^{\Phi}_{\pi^{(i)}}[\mathcal{M}_\psi] \Big| \mathcal{M}_\psi \right) = \sum_{i=1}^{B} \mathcal{H}\left(\mathbb{P}^{\Phi}_{\pi^{(i)}}[\mathcal{M}_\psi] \Big| \mathcal{M}_\psi \right) = 0.$$

We assume the set of policies Π is of size at least B. Otherwise, the result cannot be true. The first result follows immediately by Lemma 4. For any fixed π, the product distribution $\prod_{i=1}^{B} \mathbb{P}^{\Phi}_{\pi}[\mathcal{M}_\psi]$ is a distribution over 'diagonal' tuples of the form $\underbrace{(b, \cdots, b)}_{\text{size } B}$ where b is an embedding.

Consider an arbitrary set of policies $\pi^{(2)}, \cdots, \pi^{(B)}$ satisfying $\pi^{(i)} \neq \pi^{(j)} \neq \pi$ for all $i, j \in \{2, \cdots, K\}$. Call $\pi^{(1)} = \pi$. The distribution over embeddings induced by the product distribution $\prod_{i=1}^{B} \mathbb{P}^{\Phi}_{\pi^{(i)}}[\mathcal{M}_\psi]$ is made of tuples of the form $(b_1, \cdots, b_B)$. Call $\mathbf{p}(b_1, \cdots, b_B)$ the probability under the product measure $\prod_{i=1}^{B} \mathbb{P}^{\Phi}_{\pi^{(i)}}[\mathcal{M}_\psi]$ of the tuple $(b_1, \cdots, b_B)$. Notice the 'projection measure' onto the first coordinate satisfies

$$\prod_{i=1}^{B} \mathbb{P}^{\Phi}_{\pi^{(i)}}[\mathcal{M}_\psi](b_1) = \prod_{i=1}^{B} \mathbb{P}^{\Phi}_{\pi}[\mathcal{M}_\psi](b_1, \cdots, b_1) := \mathbf{p}_1(b_1)$$

and that $\sum_{b_2, \cdots, b_K} \mathbf{p}(b_1, b_2, \cdots, b_K) = \mathbf{p}_1(b_1)$. This induces a partition over the outcome space $(b_1, \cdots, b_K)$ corresponding to $C(b_1) = \{(b_1, b_2, \cdots, b_K)\}_{b_2, \cdots, b_K}$.

A direct use of Lemma 4 implies the entropy of the product distribution over finer grained tuples is larger than the entropy over the diagonal of the product distribution having all coordinates equal to each other. In fact this result also informs when the inequality will be strict. This happens when (for example the measure over $\mathcal{M}_\psi$ is the counting measure) there exists two worlds $\mathcal{M}^{(1)}_\psi$ and $\mathcal{M}^{(2)}_\psi$ such that the embedding tuples $(b^{(1)}_1, \cdots, b^{(1)}_B)$ and $(b^{(2)}_1, \cdots, b^{(2)}_B)$ satisfy $b^{(1)}_1 = b^{(2)}_1$ and $(b^{(1)}_2, \cdots, b^{(1)}_B) \neq (b^{(2)}_2, \cdots, b^{(B)}_B)$.

We will use this observation to prove the second claim. Consider a family of MDPs formed of depth L binary trees. In this family, the paths leading to the $L-1$ layer are the same, but the connections from layer $L-1$ to layer L are unknown. The 'posterior' distribution is assumed to be uniform over all plausible trees. Layer $L-1$ has size 2^{L-1} and layer L has size 2^L. W.l.o.g. we assume the set of nodes in layer L is labeled as $[1, 2, \cdots, 2^L]$. The distribution over models is supported over the set of partitions of size 2^{L-1} of $[1, \cdots, 2^L]$ where each partition set has size 2. We assume $2^L \geq B$ so that there are at least B distinct policies.

Let π be a fixed policy. It is enough to show there exist two worlds $\mathcal{M}_\psi^{(1)}$ and $\mathcal{M}_\psi^{(2)}$ such that policy π ends in the same state for both $\mathcal{M}_\psi^{(1)}$ and $\mathcal{M}_\psi^{(2)}$ but that there exist distinct policies $\pi^{(2)}, \cdots, \pi^{(B)}$ such that their endpoints $(b_2^{(1)}, \cdots, b_B^{(1)})$ and $(b_2^{(2)}, \cdots, b_B^{(2)})$ in worlds $\mathcal{M}_\psi^{(1)}$ and $\mathcal{M}_\psi^{(2)}$ satisfy $(b_2^{(1)}, \cdots, b_B^{(1)}) \neq (b_2^{(2)}, \cdots, b_B^{(2)})$. The latter always holds because among the set of models that maintain the same endpoint for π, there is a pair of models that has different endpoints for $\pi^{(2)}$ for any arbitrary $\pi^{(2)} \neq \pi$. This suffices to show the entropy of the ensemble of distinct policies is strictly larger than the entropy of the 'diagonal choice' $\underbrace{(\pi, \cdots, \pi)}_{\text{size } B}$.

□

C.1.2 InfoCascade

Since the mutual information objective in Eq. 3 is submodular, a simple greedy algorithm yields a $(1 - \frac{1}{e})$ approximation of the optimum [73]. This is the same observation that gives rise to the greedy algorithm underlying other batch exploration objectives, such as in BatchBALD [52].

Let's start by assuming we have candidate policies $\pi^{(1)}, \cdots, \pi^{(i-1)}$. InfoCascade selects $\pi^{(i)}$ based on the following greedy objective,

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

Equation 4 implies

$$\mathcal{H}\left(\prod_{j=1}^{i} \mathbb{P}^{\Phi}_{\pi^{(j)}}[\mathcal{M}_\psi] \middle| \mathcal{M}_\psi, , \tilde{\pi}^{(j)} = \pi^{(j)} \;\; \forall j \leq i-1\right) = \sum_{j=1}^{i} \mathcal{H}\left(\mathbb{P}^{\Phi}_{\pi^{(i)}}[\mathcal{M}_\psi] \middle| \mathcal{M}_\psi\right) \tag{12}$$

and therefore if we approximate $\mathbb{P}^{\Phi}_{\pi}[m_\psi]$ by a Gaussian $\mathbb{P}^{\Phi}_{\pi}[m_\psi] \approx \mathcal{N}(\mu(m_\psi, \pi), \Sigma(m_\psi))$, the conditional entropy becomes a policy independent term. In this case finding $\pi^{(i)}$ boils down to solving for the policy that maximizes $\mathcal{H}\left(\prod_{j=1}^{i} \mathbb{P}^{\Phi}_{\pi^{(j)}}[\mathcal{M}_\psi] \middle| \pi^{(1)}, \cdots, \pi^{(i-1)}\right)$. Using the same approximations described for the single policy objective, a suitable surrogate for this objective is to maximize $\max_\pi \mathrm{Var}(\pi | \pi^{(1)}, \cdots, \pi^{(i-1)})$, the empirical variance over means with respect to the policies found so far $\{\pi^{(j)}\}_{j=1}^{i-1}$,

$$\begin{aligned}&\mathrm{Var}(\pi | \pi^{(1)}, \cdots, \pi^{(i-1)}) = \\ &\qquad \frac{1}{|\mathcal{M}_\psi||i| - 1} \sum_{m_\psi} \sum_{\tilde{\pi} \in \{\pi^{(1)}, \cdots, \pi^{(i-1)}, \pi\}} \|\mu(m_\psi, \tilde{\pi}) - \mu'(\pi, \pi^{(1)}, \cdots, \pi^{(i-1)})\|^2\end{aligned}$$

Where $\mu'(\pi, \pi^{(1)}, \cdots, \pi^{(i-1)}) = \frac{1}{|\mathcal{M}_\psi||i|} \sum_{m_\psi} \sum_{\tilde{\pi} \in \{\pi^{(1)}, \cdots, \pi^{(i-1)}, \pi\}} \mu(m_\psi, \tilde{\pi})$.

D Theoretical RL Intuition

The problem we study in this work can be thought of as the "batch" version of the Reward Free Exploration formalism [45]. In this setting, the learner interacts with an MDP in two phases: 1) a training phase, where it is allowed to collect data from the environment; 2) a test phase, where the learner is presented with an arbitrary task (parametrized by a reward function unseen during training) and it must produce a near optimal policy. When faced with this problem, the learner is required to design a careful exploration strategy that permits them to build an accurate model in all areas of the state-space that can be reached with sufficient probability. There have been multiple follow up works that have also studied this problem in the context of Linear MDPs [114] and neural function approximation [86]. Nonetheless, there has been limited focus on the batch setting, where the learner is required to collect data via parallel data gathering policies in each training phase.

Due to the adaptive nature of the algorithm (every single environment query uses all information collected so far), there is an inevitable drop in performance when moving from the sequential (fully online) setting to the batch (deployment efficient) setting, as measured by the total number of environment interactions (in our case, $T \times B$ where T is the number of training rounds and B the number of parallel collection policies). To illustrate how CASCADE mitigates this loss in sample-efficiency, we outline the pseudo-code of CASCADE-TS, a greedy Thompson Sampling algorithm (see [4]) that produces the $i-$th batch exploration policy in a tabular enviornment by incorporating fake count data from rolling out policies $\pi^{(1)}, \cdots, \pi^{(i-1)}$ in the model. In the algorithms we present in the main body, we incorporate data gathered in the model by previous $i-1$ selected policies into the batch when building the $i-$th exploration policy; this is directly related to the approach behind CASCADE-TS. Concretely, encouraging policy $\pi^{(i)}$ to induce a high variance among the embeddings produced by $\{\pi^{(1)}, \cdots, \pi^{(i-1)}\}$ can be thought as adding a bonus to encourage $\pi^{(i)}$ to visit regions of the space that have a low embedding visitation count by the previous policies in $\{\pi^{(1)}, \cdots, \pi^{(i-1)}\}$.

Algorithm 2 CASCADE-TS

```
Input: Exploration batch size $B$. Fake samples parameter $M$.
Initialize model $\hat{\mathbb{P}}_0$ over $\mathcal{S} \times \mathcal{A}$ state actions.
Initialize the batch collection policies as $\{\pi_0^{(i)}\}_{i=1}^B$ to uniform.
for time in $k = 0, 1, 2, \dots$ do
    Collect trajectory data $\{\tau_k^{(i)}\}_{i=1}^B$ from $\pi_k^{(i)}$ for all $i = 1, \cdots, B$. Update $\mathcal{D}^{k+1} \leftarrow \mathcal{D}^k \cup \{\tau_k^{(i)}\}_{i=1}^B$.
    Initialize the fake data buffer $\mathcal{D}_{\text{fake}}^{k+1} = \emptyset$.
    for $i = 1, \cdots, B$ do
        Sample MDP model from posterior $\widetilde{\mathbb{P}}_{k+1}^{(i)} \sim \mathbb{P}(\cdot | \mathcal{D}^{k+1} \cup \mathcal{D}_{\text{fake}}^{k+1})$.
        Compute fake counts $N_{k+1}^{(i)}(s,a) = \sum_{(\tilde{s},\tilde{a},\tilde{s}') \in \mathcal{D}^{k+1} \cup \mathcal{D}_{\text{fake}}^{k+1}} \mathbf{1}(\tilde{s} = s, \tilde{a} = a)$.
        Compute fake uncertainty bonuses $b_{k+1}^{(i)}(s,a) \propto \sqrt{\frac{1}{N_{k+1}^{(i)}(s,a)}}$
        Solve for $\pi_{k+1}^{(i)}$ by solving,
            $$\pi_{k+1}^{(i)} = \max_{\pi} \mathbb{E}_{\tau \sim \widetilde{\mathbb{P}}_{k+1}^{(i),\pi}} \left[ \sum_{h=1}^{H} b_{k+1}^{(i)}(s_h, a_h) \right]$$
        Collect fake trajectory data $\{\tau_{\ell,\text{fake}}^{(i)}\}_{\ell=1}^M$ from posterior sampled model $\widetilde{\mathbb{P}}_{k+1}^{(i)}$.
        Update fake data buffer $\mathcal{D}_{\text{fake}}^{k+1} \leftarrow \mathcal{D}_{\text{fake}}^{k+1} \cup \{\tau_{\ell,\text{fake}}^{(i)}\}_{\ell=1}^M$.
    end for
end for
```

Algorithm 2 works by sampling a model from a model posterior, solving for an optimal uncertainty seeking policy and updating the model with 'fake' data corresponding to trajectories collected in the model from this uncertainty seeking policy. After producing the $i-$th exploration policy in the batch, the recomputed uncertainty bonuses of the model are reduced in the areas of the state space most visited by the $i-$th policy. This will encourage subsequent uncertainty seeking policies (i.e., $\pi_{k+1}^{(i+1)}, \pi_{k+1}^{(i+2)}, \dots, \pi_{k+1}^{(B)}$) to visit parts of the space not yet explored by previous policies.

Why Thompson Sampling? The reader may wonder why Algorithm 2 samples a model $\widetilde{\mathbb{P}}_{k+1}^{(i)}$ from a TS posterior instead of using the empirical model resulting from fake and true data. The answer is that the raw empirical model may ascribe a probability of zero to certain transitions, which means we may miss exploring parts of the true state-action space. Having a TS prior that ascribes non-zero probabilities to all possible transitions fixes this potential issue.

D.1 Proof of Lemma 2

Let's consider the class of deterministic MDPs with $S = |\mathcal{S}|$ states and $A = |\mathcal{A}|$ actions. It is clear that playing the same deterministic policy multiple times does not provide us with any more information than playing it once. We will then show that running Thompson TS, the learner will end up with a nonzero probability of producing at least two distinct policies within a batch of size $B > 2$.

We will further assume the posterior is aware of the deterministic nature of the MDP family so that any sample MDP model $\widetilde{\mathbb{P}}_{k+1}^{(i)}$ is deterministic. For $i > 1$, the model $\widetilde{\mathbb{P}}_{k+1}^{(i)}$ is one whose transitions are consistent with the fake data generated by $\pi_{k+1}^{(1)}, \cdots, \pi_{k+1}^{(i-1)}$ (and the true data in $\mathcal{D}^k$).

A deterministic MDPs can be encoded as a set of triplets (s, a, s'). We say a model $\{(s, a, s')\}_{s\in\mathcal{S}, a\in\mathcal{A}}$ is $\epsilon-$accurate if at least a fraction of $1 - \epsilon$ of the triplets are correct.

We now illustrate how the CASCADE-TS algorithm evolves in this setting. To simplify our task, we will further restrict our attention to the family of deterministic MDPs made of binary trees of height L. The number of nodes in any of such trees equals $2^{L+1} - 1$.We will define the counts to be a large value ($2L$) when no data has been collected of a given state action pair. When data has been collected, we define the bonus terms to be of order at most 1.

To figure out the connectivity of any such trees it is necessary to try out $2^L - 1$ distinct sequences of L actions (the nature and connectivity of the remaining leaf nodes can be inferred once all the other ones have been decoded). To build an $\epsilon-$accurate model, it is enough to know the connectivity structure of $1 - \epsilon$ proportion of paths corresponding to $(1 - \epsilon)(2^L - 1)$ leaf nodes.

To prove the result of Lemma 2, we observe that any batch strategy can be simulated by a fully sequential learner, so it immediately follows that $T(\epsilon, \text{Sequential}) \leq T(\epsilon, \text{AnyBatchStrategy})$. To show that CASCADE-TS will be better than SinglePolicyBatch in this tree example, as long as B is smaller than the remaining number of paths the learner has not tried out, it will produce a set of B policies different from any policy played so far and also different from each other.

To see why the second inequality is true, it is enough to see that complete paths (real or imagined) have a total reward score of at most L, since the counts are always at most one for each edge that is present. Nonetheless, the counts over any path (sequence of L actions) that has not been visited nor has been imagined to be visited will be at least $2L$. Thus, every step in the sequential batch construction mechanism will produce a new policy (sequence of L actions) not existent neither in the true nor the fake data so far. This finalizes the proof. Notice that the SinglePolicyBatch strategy will have tried only K unique paths after K batches have been collected, whereas the CASCADE-TS strategy will have tried BK, resulting in a potentially substantial improvement in sample efficiency.