Plan2Explore Again

Kerim Erekmen
kerimerekmen@informatik.uni-hamburg.de
Hauke Bünning
haukebuenning@informatik.uni-hamburg.de

***Abstract*—This paper introduces a significant modification to the Plan2Explore algorithm, focusing on a novel approach to intrinsic reward computation. Unlike traditional methods that measure uncertainty in predictions of latent states, our modified algorithm quantifies uncertainty based on a sample of action predictions. This shift emphasizes the maximization of action disagreement under the learned world model, fostering more effective exploration strategies. Our exploration policy targets actions with high uncertainty, facilitating the discovery of interesting and rewarding states. We enhance the baseline methodology by integrating the concept of Latent Action Disagreement, which utilizes the inverse model from the Intrinsic Curiosity Module (ICM) [3] within a planned latent space. This integration improves the efficiency of the learning process compared to the baseline model of the Plan2Explore algorithm in a zero-shot configuration. This approach shows that actions with high uncertainty are a suitable approach to promote the agents curiosity. The world model is continually refined using data from new trajectories, leading to reduced uncertainty and more precise action prediction. Code: https://github.com/wabbajack1/plan2explore-again**

I. Introduction

To enable agents to become universal problem solvers, it is crucial to use a broad exploration method. One of the biggest challenges in reinforcement learning is the effective use of samples to quickly learn new tasks. Central to this is the use of an exploration strategy, which is integral to the concept of task-agnostic learning. This paradigm highlights the ability of agents to learn without a specific tasks in mind [2]. A universal agent is designed to learn in a task-agnostic manner and is thus able to formulate strategies without direct interaction with the real world or a simulator [1]. In this approach often through a mechanism similar to curiosity in humans, the agent should plan future interaction with the dyamics of its environment with a generic model [4]. This intrinsic motivation to explore and plan is not driven by external rewards, but by an internal desire to discover and understand new aspects of its environment. Such a system highlights the potential to create more autonomous and efficient learning system that is not reliant on predefined reward structures, but are self-motivated and self-sustaining in their pursuit of knowledge and skill acquisition. This type of learning is closely related to unsupervised learning, which maximizes the use of an abundance of unlabeled data [5].

Planning in unknown environments continues to be a major and ongoing challenge. For an agent to navigate these environments effectively, it must acquire knowledge of the dynamics through practical experience. Thus, the key question is: Is it possible to agnostically learn a general model of the environment and within the model, maximizing imagined future novelty, and thus guide exploration in important bounded domains without directly interacting with the environment and later use this model for task adaptation in a zero- or few-shot fashion? Learning of generic dynamics models that have sufficient accuracy for planning purposes while addressing this aspect of exploration is a persistent challenge in the field of model-based reinforcement learning, especially in the context of task-agnostic learning.

[4] presents an approach to tackle the key question. Plan2Explore is differentiated by a unique approach to self-supervised exploration and rapid adaptation to new tasks that do not need to be predefined during the exploration phase. Unlike previous methods that retrospectively calculate the novelty of observations, the exploration policy in Plan2Explore is optimized from trajectories imagined under the learned world model to maximize the its intrinsic rewards. The agent is able to measure expected novelty by imagining future states that have not been visited yet. This methods is able to solve downstream tasks with zero or a few shot interaction with the environment.

In this paper, in particular in Section III, we present a modification of the Plan2Explore algorithm in which the agent computes its intrinsic rewards under the learned world model by quantifying the uncertainty about a sample of action predictions instead of the uncertainty about its predictions for different latent states [4]. This suggests that the agent strives to maximize the disagreement of its actions under the world model. An exploration policy then looks for actions with high uncertainty. By evaluating different outcomes from different models, an agent can estimate which actions are most likely to lead to interesting and rewarding states. The world model is then trained using the newly acquired trajectories and reduces its uncertainty.

II. Related Work

Plan2Explore for learning process [4] method consists of two stages. First, the agent acquires environmental knowledge through self-supervised exploration and encapsulates this experience in a parametric world model. After exploration, the agent is assigned a downstream task, defined by a reward function, to which it must adapt with minimal or no further interaction with the environment. The exploration phase begins with the agent developing a comprehensive model of the world based on the collected data. This model then guides the agent in its exploration to collect further data, as described in A (a.). This process involves training an exploration strategy within the world model to follow unexplored states. The novelty of states is determined by the degree of disagreement between latent predictions generated by a 1-step transition ensemble. During the adaptation phase, as illustrated in A (b.), a task policy can be effectively optimized by imagination within the world model. Since our self-supervised model is trained without preference for a particular task, a single model can be adeptly used to solve different downstream tasks. The Figure 1 depicts the modified architecuture of Plan2Explore [4].

PlaNet for latent dynamics model To plan efficiently in the latent state, [4] uses PlaNets latent dynamics model [1], which learns the environment dynamics from pixels and selects actions through online planning in a compact latent space. To learn the dynamics model a transition model with both stochastic and deterministic components are used.

Dreamer for learning in latent dynamics model Instead of online planning, [4] uses Dreamer [1] to efficiently learn a parametric policy inside the world model that considers long-term rewards. Specifically Dreamer is background planning, i.e. before an action is selected for a current state s_t, planning has played a role in improving the function approximation parameters needed to select actions for many states, including s_t. Specifically, the planning approach consists of learning two neural networks that operate on latent states of the model, similar shown in figure 1.

Computing Intrinsic Reward Curiosity-driven exploration in the paper involves formulating curiosity as the error in an agent's ability to predict the consequences of its actions in a given environmental state [3]. This is achieved through a self-supervised learning framework where the agent learns to predict the outcome of its actions in a feature space learned by a neural network. The prediction error, representing the agent's surprise, serves as an intrinsic reward, motivating the agent to explore actions that lead to new or less predictable states.

III. Approaches

A. Components of the learning agent

Intrinsic reward model To compute the Intrinsic Reward a ensemble of one-step models which are trained alongside the world model. They aim to help explore how two states are connected and which actions may be taken to move from one to the other. Our contribution lies right here, specifically that our one-step models mimic the ICM's Inverse Model [3] instead of the forward model. The difference to Pathak et al's inverse model however is that we predict our action in latent space, whereas the original ICM uses a simulator.

(a)

Algorithm 1 Planning to Explore Again via Latent Action Disagreement

```
1: initialize: Dataset $D$ from a few random episodes.
2: World model $M$.
3: Latent disagreement ensemble $E$.
4: Exploration actor-critic $\pi_{LAD}$.
5: while exploring do
6:     Train $M$ on $D$.
7:     Train $E$ on $D$.
8:     Train $\pi_{LAD}$ on LAD reward in imagination of $M$.
9:     Execute $\pi_{LAD}$ in the environment to expand $D$.
10: end while
11: return Task-agnostic $D$ and $M$.
```

(b)

Algorithm 2 Zero and Few-Shot Task Adaptation

```
1: input: World model $M$.
2: Dataset $D$ without rewards.
3: Reward function $R$.
4: initialize: Latent-space reward predictor $\hat{R}$.
5: Task actor-critic $\pi_R$.
6: while adapting do
7:     Train $\pi_R$ on $\hat{R}$ in imagination of $M$.
8:     Execute $\pi_R$ for the task and report performance.
9:     Optionally, add task-specific episode to $D$ and repeat.
10: end while
11: return Task actor-critic $\pi_R$.
```

Unlike Plan2Explore, our ensemble model utilizes the feature encoding from the latent space of imagined consecutive states, s_t and s_{t+1}, as inputs. The loss is computed as follows:

$$\min_{\theta} L_2(\hat{a}_t, a_t) = \|\hat{a}_t - a_t\|_2^2 \tag{1}$$

It employs a neural network to predict the action $\hat{a}_t$ that the agent should execute to transition from s_t to s_{t+1}. It is important to note that these states are not real but imagined by the model, which also includes imagining the actions that would lead to the next state.

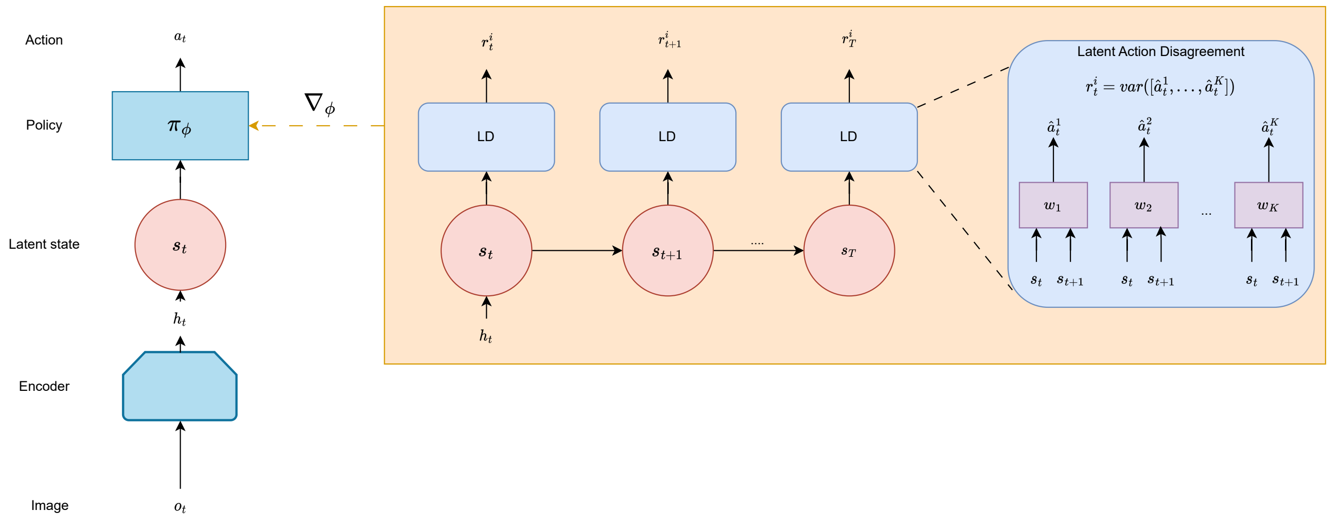

Figure 1. In Plan2Explore-Again, each observation o_t at time t is encoded into features h_t to infer a recurrent latent state s_t. During training, the agent plans using the current world model to maximize expected novelty r_t^i over future time steps. This planning objective, based on the disagreement in predicted actions from learned transition dynamics w_k, is backpropagated to improve the exploration policy π_ϕ. The learned model is then utilized for latent space exploration, with collected data iteratively enhancing the model. At test time, the refined world model plans for novel tasks by shifting from novelty to task-specific rewards.

Dynamic models

$$\begin{aligned}
\text{Image encoder:} \quad & h_t = e_\theta(o_t) && (2)\\
\text{Posterior dynamics:} \quad & q\theta(s_t|s_{t-1}, a_{t-1}, h_t) && (3)\\
\text{Prior dynamics:} \quad & p\theta(s_t|s_{t-1}, a_{t-1}) && (4)\\
\text{Reward predictor:} \quad & p\theta(r_t|s_t) && (5)\\
\text{Image decoder:} \quad & p\theta(o_t|s_t) && (6)\\
\text{Actor:} \quad & a_t \sim \pi_\phi(a_t \mid s_t) && (7)\\
\text{Value estimator:} \quad & v_\psi(s_t) \approx \mathbb{E}_{\pi(a_t|s_t)}\left(\sum_{\tau=t}^{t+H} \gamma^{\tau-t} r_\tau^i\right). && (8)
\end{aligned}$$

As in [4] we use the same models to learn the dynamics of the environement. We derive an action selection strategy from the learned world model and learn new behaviors. Here we use an actor-critic architecture, specifically two neural networks interact with the learned world model, i.e. the state-value estimates $v_\beta(s_t)$ the sum of future rewards and the actor $\pi_\phi(a_t|s_t)$ tries to maximize these predicted values. The value model estimates the expected imagined rewards that the action model achieves from each state.

B. Latent Action Disagreement

[4] are using a ensemble disagreement of different predictions for latent states as an empirically successful method for quantifying uncertainty and hence computing the instrinsic reward signal. In our method, we instead use an ensamble of action models, as an exploration strategy where actions are chosen to maximize expected novelty. The variance measures the disagreement between the actions predicted by multiple models $w_{k:K}$ given the same state, encouraging the agent to explore states where its models disagree about what action to take, see Figure 1. Lets imagine a scenario where a explorer is faced with several paths in the jungle (action to take), each leading to unknown territories. By exploring these less certain paths, the explorer gains a wealth of diverse experiences, learning about different parts of the jungle that were not well understood. The idea is that if the models do not match the action, the system is uncertain about what will happen, indicating a lack of knowledge about that part of the state space, therefore high disagreement about the best action to take means that there is uncertainty or lack of knowledge about the consequences of actions in that state, causing the agent to explore to resolve this uncertainty. Therefore, action disagreement captures uncertainty at the level of decision making for immediate actions, while latent state disagreement captures uncertainty about the next state of the environment.

C. Learning Process

The learning process described in Figure 11 generally follows the one outlined for Plan2Explore [4] but differs specifically in the Latent Action Disagreement, as can be seen in Algorithm 1II in line 7 and III-B.

D. Experimental setup

Hyperparamters The hyperparameters, environment details (e.g. in the Deep Mind Control Suite [7]) and implementation details are explained here. As in Plan2Explore we use visual observations and an episode length of 1000 steps. But due to time constraints our testing will focus on a zero shot approach and a comparatively lower number of episodes (1500 compared to 3000). Equally, in this draft only four runs will be examined for both our approach and the baseline, due to time constraints.

Environment As mentioned above we use the DM Control Suite benchmark and specifically the walker-walk task, in which the algorithm controls a simple bipedal agent that is tasked with walking as far to the right as possible.

IV. Results

Evaluation Metric To evaluate both our and the baseline approach we will be using the test reward. The test reward is computed by evaluating the zero shot performance of the approaches on unknown data every 10 episodes. This metric will be used as it is also used in the Plan2Explore paper[4].

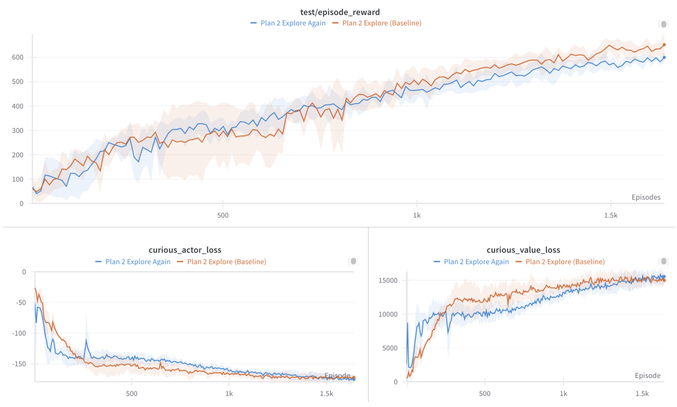

Figure 2. The graph depicts a performance comparison between two methods, Plan2Explore and Plan2Explore-Again, on the walker-walk task based on zero-shot learning from raw pixels.. The x-axis displays episodes and the y-axis displays the test reward (see the video). These graphs are averaged over four runs per approach.

Analysis In Section IV, both methods demonstrate an improvement in rewards over time, indicating effective learning from the environment through the curiosity signal. A notable distinction, as shown by the upper graph, is that Plan2Explore-Again exhibits more consistent variance, suggesting that its use of an ensemble of action prediction as intrinsic reward might offer better, more stable performance compared to an ensemble of latent state prediction. This could be due to Latent Action Disagreement (LAD) fostering exploration more directly linked to rewards. The trend reveals that while both methods improve over time, Plan2Explore-Again frequently attains higher rewards at comparable episode intervals. However, there are instances, especially in the latter half of the experiment, where its rewards dip close to or below the Plan2Explore line. Overall, Plan2Explore-Again appears to have an advantage in exploring the state space during the initial 500 episodes but eventually shows similar results to the baseline at the experiment's end. This suggests that the agent trained with LAD becomes more decisive in its actions, as evidenced by the reduced variance in the predicted actions array, showing greater certainty about which action will lead to the next imagined state. This is also reflected in the actor loss, where Plan2Explore-Again displays a narrower variance than the baseline, and both methods see a decrease in actor loss, indicating policy improvement. Plan2Explore-Again's slight advantage may imply that focusing on action-based uncertainty aligns more closely with the reward structure of the environment. The stagnations cause is unclear, but the actor loss suggests that Plan2Explore-Agains agent gains confidence in its actions more quickly than the baseline, reducing exploration. The faster decrease in LAD compared to the baseline might contribute to this. The value loss also supports this, showing greater stability for the Plan2Explore-Again agent compared to the baseline approach.

V. Conclusion

The paper introduces a new method, Plan2Explore-Again, for computing intrinsic rewards in the Plan2Explore framework. It focuses on action prediction disagreement as a means to promote exploration. The modified algorithm demonstrates promise by utilizing latent action disagreement to enhance the efficiency of the learning algorithm. The results indicate that Plan2Explore-Again outperforms the baseline Plan2Explore method, with respect to its stability in zero-shot manner, suggesting that using an ensemble of action predictions as intrinsic rewards yields better performance than using an ensemble of latent state predictions. This is observed in a task where the algorithm controls a bipedal agent in the DM Control Suite benchmark, aiming to maximize its distance walked to the right. Plan2Explore-Again appears to increase the agent's confidence in selecting certain actions by reducing the variance of predicted actions. However, after an initial exploration phase, the performance of Plan2Explore-Again plateaus and becomes similar to the baseline. The paper acknowledges this and further exploration is recommended to understand the reasons behind this.

References

[1] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

[2] Simone Parisi, Victoria Dean, Deepak Pathak, and Abhinav Gupta. Interesting object, curious agent: Learning task-agnostic exploration. *Advances in Neural Information Processing Systems*, 34:20516–20530, 2021.

[3] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. pages 2778–2787, 2017.

[4] Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. 2020.

[5] Ravid Shwartz-Ziv and Yann LeCun. To compress or not to compress–self-supervised learning and information theory: A review. *arXiv preprint arXiv:2304.09355*, 2023.

[6] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[7] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy P. Lillicrap, and Martin A. Riedmiller. Deepmind control suite. *CoRR*, abs/1801.00690, 2018.

APPENDIX

GENERIC VIEW OF MODEL BASED LEARNING

Figure 3 escribes a process where an agent learns to make decisions by interacting with an environment. The model represents the environment. Planning uses the model to predict outcomes. Value/Policy guides decision-making to achieve goals. Experience is the data gained from actions taken. Model learning improves the model with new data, and Direct RL updates the Value/Policy directly from experiences without a model. Arrows indicate the flow from one stage to the next in a cyclical pattern.

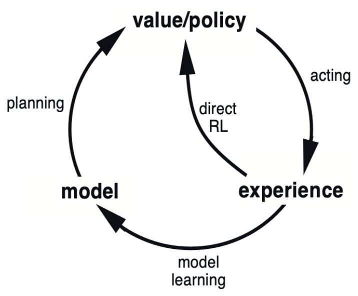

Figure 3. Generic process of planning and learning. ([6])

ADDITIONAL ALGORITHMS

This is the algorithm of the baseline of Plan2Explore [4], which was compared to our Plan2Explore-Again approach. Shown here is (a.) the first phase of the algorithm, in which a generic model is learned and further exploration is guided to fill the experience buffer, and (b.) the second phase, in which the adaptation phase is performed to learn downstream tasks.

(a)

Algorithm 3 Planning to Explore via Latent Disagreement

```
1: initialize: Dataset $D$ from a few random episodes.
2: World model $M$.
3: Latent disagreement ensemble $E$.
4: Exploration actor-critic $\pi_{LD}$.
5: while exploring do
6:    Train $M$ on $D$.
7:    Train $E$ on $D$.
8:    Train $\pi_{LD}$ on LD reward in imagination of $M$.
9:    Execute $\pi_{LD}$ in the environment to expand $D$.
10: end while
11: return Task-agnostic $D$ and $M$.
```

(b)

Algorithm 4 Zero and Few-Shot Task Adaptation

```
1: input: World model $M$.
2: Dataset $D$ without rewards.
3: Reward function $R$.
4: initialize: Latent-space reward predictor $\hat{R}$.
5: Task actor-critic $\pi_R$.
6: while adapting do
7:    Distill $R$ into $\hat{R}$ for sequences in $D$.
8:    Train $\pi_R$ on $\hat{R}$ in imagination of $M$.
9:    Execute $\pi_R$ for the task and report performance.
10:   Optionally, add task-specific episode to $D$ and repeat.
11: end while
12: return Task actor-critic $\pi_R$.
```