Model Based Reinforcement Learning Pre-Trained with Various State Data

1st Masaaki Ono
SOKENDAI, The Graduate University for Advanced Studies
National Institute of Informatics
Tokyo, Japan
onom@nii.ac.jp

2nd Ryutaro Ichise
Tokyo Institute of Technology
National Institute of Informatics
Tokyo, Japan
ichise@iee.e.titech.ac.jp

***Abstract*—Reinforcement Learning (RL) has shown remarkable capabilities in various domains, yet struggles in environments with sparse rewards. A significant challenge in such environments is the exploration depth and the robustness of performance. This paper introduces WODID framework, aiming to enhance exploration in Model-Based Reinforcement Learning (MBRL) without relying heavily on initial or early-stage trajectory data. We identify one primary issue of the transition model of MBRL: trained with random policy when forming the transition model, which hinders exploration and causes high dependency on the success of dataset collection by random policy. By pre-training world models using diverse state data, WODID improves the quality of the transition model, leading to deeper exploration and stabilizing its performance. Our empirical studies, particularly in the challenging sparse reward environment: Montezuma's Revenge, demonstrate that WODID outperforms the baseline methods, achieving more profound exploration with fewer environmental steps. Furthermore, our approach offers a human-free method to feed trajectory data, promoting less dependency on initial samples and paving the way for more robust and efficient RL agents.**

***Index Terms*—artificial intelligence, reinforcement learning, model-based reinforcement learning, neural networks**

I. Introduction

Reinforcement Learning (RL) has witnessed transformative advancements, enabling computers to achieve remarkable feats, from mastering intricate board games like GO to outperforming humans in multifaceted games such as DOTA [1], [2]. At the core of RL lies the principle of learning through interaction, whereby agents continuously adapt their actions based on the rewards received from an environment. Despite these celebrated successes, RL has its share of challenges that hinder its broader applicability, especially in complex real-world scenarios.

One of the most formidable challenges in RL is the notorious "sparse reward" problem [3]. In many real-world tasks, rewards are not frequently encountered, making it challenging for the agent to discern the optimal sequence of actions leading to the desired outcome. Imagine a scenario where an RL agent is tasked with searching for survivors in a disaster-stricken area. The goal is abstract: "find survivors and report their location". This high-level directive lacks the step-by-step guidance an agent might need. Consequently, the agent might explore vast areas without any feedback, making learning slow and inefficient. Such tasks exemplify the challenge of sparse rewards, where the reward signals are few and far between. Furthermore, these sparse reward functions often lead to unintended local optima, where the agent might get stuck performing suboptimal actions believing them to be the best [4]–[7].

The implications of sparse rewards are manifold. Firstly, they lead to poor sample efficiency, implying that agents require a massive amount of interaction data before they can learn anything meaningful. This limitation is especially detrimental in real-world applications where obtaining such data is either expensive, time-consuming, or both. Secondly, due to the inherent difficulty in exploring the environment efficiently, the stableness of exploration of an agent in the environment varies. It is significant that an agent can achieve a stable performance regardless of the initial setting or a lucky exploration especially when the application is in reality.

To circumvent the sample efficiency problem, researchers have turned to Model-Based Reinforcement Learning (MBRL). Unlike its model-free counterpart, which learns purely from trial and error, MBRL leverages "World Models" [8]–[10] that capture and generalize the dynamics of the environment.

These models, which often learn compact representations of sensory, can predict the outcomes of potential actions, thereby enabling more informed planning. By capturing essential characteristics of the environment, world models have proven effective in tasks ranging from virtual environments like Minecraft [11] and Crafter [12] to real-world robotic tasks [13]. While MBRL has made strides in addressing the sample efficiency problem, it introduces its own set of challenges. A pivotal concern is the deviation of the transition model from real-world transitions, which can lead to policies that, although optimal in the model, perform poorly in the actual environment. Also, since the training data of world models are dependent on its RL agent, the stability problem remains in MBRL too.

In light of these challenges, this paper introduces WOrld Model pretrained with DIverse Data (WODID), a novel ap-

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00169

proach that harnesses the power of state data to pre-train world models in MBRL. Our primary aim is to enhance the depth of exploration and stabilize it, thereby addressing the challenges posed by sparse rewards and model inaccuracies. Through WODID, we hope to bridge the gap between theoretical RL successes and its practical, real-world applications.

The key Contributions of this paper are summarized as follows:

- Enhanced Exploration: We propose methods that significantly improve exploration in the RL domain, ensuring agents can effectively navigate environments, especially those with sparse rewards.
- Reduced Dependency on Prefills: The performance of traditional MBRL methods heavily relies on initial prefills under random policy. Our approach minimizes this dependency, paving the way for more adaptive and resilient RL agents.
- Human-free Training Approach: Recognizing the challenges in preparing expert data, we introduce a new training paradigm that is both human-free and efficient. This method not only simplifies the training process but also boosts both exploration and its stability.
- Framework-Centric Advancement: While algorithmic innovations are abundant in the field of RL, our work uniquely emphasizes the overarching framework. We believe that robust frameworks can amplify the benefits of even simple algorithms, leading to some notable results.

In this paper, we introduce related research in Section Ⅱ. Section Ⅲ introduce the core research to our work. Section Ⅳ discusses how our system is built and more detail about how the data is collected and used. we also present the application of MBRL to our system. Section V sets up the metrics to see the performance of our system and the comparison with the baseline. We also introduce setups for hyperparameters and environmental setups in the section.

II. Related Work

In this section, we introduce research works on three main topics: the World Model in MBRL, curiosity and intrinsic motivations, and the application of expert data. We first introduce the background of world models and their challenges, especially with their exploration and stable results. Then, we introduce intrinsic motivation which is one of the dominant approaches for exploring rare states in an environment and hence it is a critical topic for finding unseen/not-so-much-seen state trajectories. The application of expert data in RL is a widely used approach for improving the policy of RL. We think it is important to present the application of data as our work also employs datasets not collected from the RL in MBRL. The MBRL is using an internal model to train its RL agent instead of the actual environments. The internal model is widely called the "World Model" and it is the model that we want to enhance its representation of transitions.

World Models in MBRL One of the pivotal advances in the RL domain has been the integration of world models in MBRL [8], [10]. Distinct from the traditional model-free methods that predominantly rely on trial and error, MBRL leverages internal representations of the environment to anticipate outcomes of potential actions. These representations, commonly termed "World Models", are pivotal in enabling agents to generalize across varied scenarios. Notably, these world models have found applicability in diverse environments ranging from virtual realms like Minecraft [11] to real-world robotic challenges [13]. Nevertheless, challenges persist. The fidelity of these world models is paramount; even slight deviations from real-world dynamics can render derived policies ineffective to explore deeply and perform stably, highlighted by prior works [8], [9], [14]–[18] and these problems should be solved in order to apply MBRL in more complicated real-world tasks.

Curiosity and Intrinsic Motivation Approach Beyond world models and expert data, there has been a burgeoning interest in endowing RL agents with intrinsic motivations to bolster exploration [3], [19]–[23], especially in sparse reward settings. One such approach is the integration of curiosity-driven mechanisms [24], [25]. By fostering an intrinsic desire to explore unfamiliar states or actions, agents can navigate environments more effectively, even in the absence of frequent external rewards. This approach is particularly pertinent in challenging real-world problems characterized by hard-exploration settings, where traditional reward signals might be insufficient to guide effective learning. However, the "detachment" and "derailment" problems [26] hinder Intrinsic Motivation approaches from further investigating states worth exploring, resulting in some states and their transitions being unable to be discovered.

Application of Expert Data in RL The realm of RL has also seen a surge in methods that employ expert data. Traditionally, RL methods, especially in complex scenarios, have benefited from expert-guided trajectories, aiding agents in navigating environments more effectively and with higher stability of performance [27]–[31]. One notable approach is Behavior Cloning [32]. Although it is a classic approach in the field of utilizing demonstration, the applications vary from complicated games such as Minecraft to computer control [33], [34]. GAIL [35] is an approach to model reward by discriminating expert trajectories from agents' trajectories. Such discrimination method (discriminator) is further extended to masking action from the trajectories [36], [37]. Expert data are also used to annotate datasets. MineDojo [38] and VPT [33] add information of action based on external information to the existing dataset. However, preparing the expert data can be cumbersome and demanding especially when it is human data. Also, reliance on expert data may improve the policy of an agent in RL but it does not necessarily broaden the diversity and adaptability of the transition model, limiting the policy of the agent only to specific situations and not to further improve its

policy. Moreover, though some approaches are learning from demonstration to benefit its action or reward, not many works focus on improving the transition model of MBRL to stabilize the performance.

III. Preliminaries

Here, we demonstrate core algorithms to the framework of WODID. we first briefly explain the state-of-the-art (SOTA) algorithms in hard exploration, Go-Explore [26]. It is critical to have a mechanism to collect useful state data for training the transition model of WODID and Go-Explore provides a hint for such data collection. In Section III-B, we summarize the Recurrent State Space Model (RSSM), the World Model mechanism of the Dreamer series. The MBRL part of our architecture is based on RSSM and hence it is crucial to understand RSSM in order to understand our system.

A. Go-Explore

Deep exploration, especially in sparse reward environments, necessitates innovative techniques. Here, we integrate the Go-Explore architecture, which is unique for its architecture memorizing states ripe for exploration and subsequently returning to these promising states.

To collect important state transitions that can refine the transition model within MBRL, we leverage the Go-Explore architecture. This architecture utilizes memorizing promising states and its transitions to resolve detachment (prioritizing easier access to intrinsic rewards that hinder intrinsic rewards in deeper exploration) and derailment (not being able to return to a promising state) problems in sparse reward environments.

The promising states and its sequence of transitions are saved in the system called "Archive" where the states are downsampled and called cells. The cell selection in Go-Explore plays a crucial role in determining states considered as "important". This selection uses a heuristic approach to score the importance of each state. Cells are chosen at each iteration by first assigning them a score, which is then normalized across all cells in the Archive, determining the probability of each cell being selected. The score of a cell is the cumulative sum of separate subscores. A significant set of these subscores termed the count subscores, are derived from attributes representing the frequency of interactions with a cell. These interactions include the number of times a cell has been chosen, the number of times a cell was visited during the exploration phase, and the number of times a cell has been selected since its last productive exploration. For each of these attributes, a lower count typically indicates a more promising cell to explore from.

$$\text{CellScore}(c, a) = w_a \cdot \left(\frac{1}{v(c, a) + \epsilon_1} \right)^{p_a} + \epsilon_2 \qquad (1)$$

In the equation, c represents the specific cell for which we're determining the score. The function $v(c, a)$ provides the value of the attribute associated with cell c. Each attribute a has an associated weight, denoted as w_a, and a power hyperparameter, represented as p_a. Also, ϵ_1 and ϵ_2 are hyperparameters set as 0.001 and 0.00001 respectively in Go-Explore.

B. RSSM

High-dimensional environments, especially those with image inputs, present unique challenges to MBRL. To navigate these, some state-of-the-art MBRL models employ the Recurrent State Space Model(RSSM) [9] as its world model system. The RSSM, with its ability to predict using compact model states, stands out as a formidable tool for planning [14], [39]. Unlike traditional prediction methods in image space, RSSM facilitates efficient parallel planning. This not only allows handling large batch sizes but also minimizes accumulating errors [40]. The following components formulate RSSM:

$$\begin{aligned} \text{Encoder:} \quad & e_t = \text{enc}_\phi(x_t) \\ \text{Posterior:} \quad & q_\phi(s_t | s_{t-1}, a_{t-1}, e_t) \\ \text{Dynamics:} \quad & p_\phi(s_t | s_{t-1}, a_{t-1}) \\ \text{Image Decoder:} \quad & p_\phi(x_t | s_t) \end{aligned} \qquad (2)$$

The Encoder and The Image Decoder leverage the prowess of convolutional neural networks (CNNs). Encoder (enc_ϕ) represents encoding the image input x_t. Image Decoder (p_ϕ) plays as a generative model of input x_t given s_t which is the latent variable of state at time step t. s_t also composes a deterministic component that uses the recurrent state of a Gated Recurrent Unit (GRU) [41]. The two functions Encoder and Decoder work to reduce the dimension of inputs and make it able to generate the next input.

The Posterior (q_ϕ) and the Dynamics (p_ϕ) are the models that utilize the latent variable of previous input s_{t-1} and the previous action a_{t-1} to generate the next latent variable s_t. During the training, the Dynamics is trained under the Posterior. Instead of the Posterior, the Dynamics model is used as the transition model of the agent since the Posterior takes e_t as input which goes against the POMDP. The training is formulated using multi-layer perceptrons. An end-to-end training paradigm is adopted, optimizing the evidence lower bound using stochastic backpropagation [42], [43] with the assistance of the Adam optimizer [44].

IV. Method

Deep exploration in MBRL has garnered significant attention due to the potential rewards it promises. However, this is not without its challenges. A central issue lies in the collection of trajectories rich in diverse state transitions, which in turn can fortify the transition model intrinsic to MBRL. This paper presents a novel method, termed the WODID, tailored to address this very challenge. In this section, we first state the problem of the transition model in MBRL and illustrate

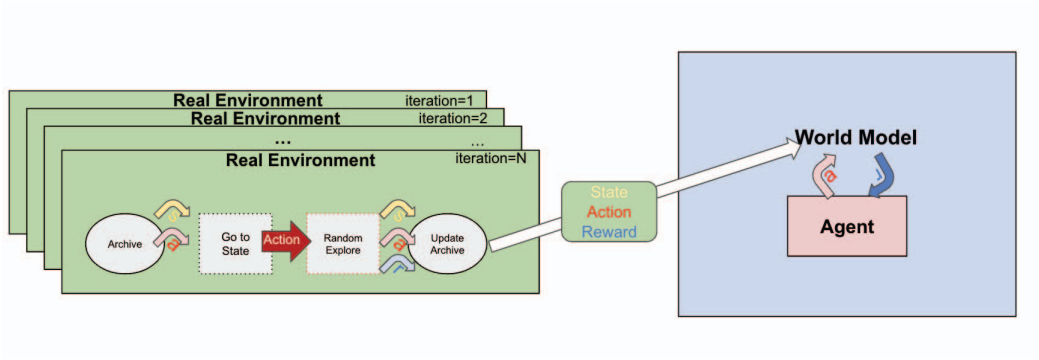

Fig. 1. The overview of WODID. The world model takes the trajectory data from the Archive that saves the trajectory (state, reward, action) of different states to train itself. The agent in the world model learns its policy from trajectories of compact representations predicted by the world model.

the design concept of our work. After that, we present what is needed for the desired MBRL. Then, we define how our system can overcome the presented issue and how WODID is composed. Section IV-C discusses the dataset size required for the pretraining data of WODID. Lastly, in Section IV-D we note the application of WODID in terms of MBRL to clarify how the WODID may be used for various scenes.

A. Design Concept

Before introducing WODID, we would like to state how Basic MBRLs are structured and identify what limitations they have. The following are the training process of MBRL:

1) Collect N steps of the trajectory of states (s, a, r) in the real environment.
2) Use the collected trajectories to train the transition model.
3) Use the transition model as an environment for an agent to update its policy.

From the structure of this MBRL, there are two main shortages: random sampling and the quality of representation of the transition model.

Random Sampling: MBRLs without specific usage of demonstration data are normally trained with random policy when first forming the transition model in MBRL. This is because there are no trained policies to be used to collect the dataset for training the transition model at the beginning. However, such a pretraining method for the transition model may be highly dependent on the dataset collected. Training the transition model with random policy in the hard exploration environment with various transitions can lead to transitions not as realistic as the real environment. The random policy lacks deep exploration which is necessary to discover new states (transitions) and thus can feed deficient kinds of state transitions to the transition model. An agent learning the wrong sequence of transitions will make the learned policy hardly applicable in the real world. The policy will fail to explore enough in the real environment and end up with another poor transition model and policy. In addition, although there may be some chance that random exploration may succeed, it is very unlikely in hard-exploration tasks and the succeeded result is hardly reproducible. In other words, such MBRLs have high variance in results which is not the agent we desire.

The Quality of Representation: Unlike Model-Free Reinforcement Learning (MFRL) MBRL trains its policy in the learned transition model as its nature. Although the nature enables MBRL to do the planning, it also forces the success of learning policies in the transition model dependent on the quality of representation. Indeed, MBRL in the real environment where it is under few transitions such as GO, shogi, and chess has superhuman achievement [27] whereas in the environments where they have diverse and complicated transitions even SOTA MBRL models cannot beat average human score [17], [45]. These contrasting results demonstrate the representation of the transition model plays a key role in MBRL.

Criteria for better Transition Model: For MBRL to achieve profound exploration, the transition model's accuracy is paramount. It is this accuracy that dictates the quality of policy learning. Consequently, we posit that feeding the model with pivotal states enhances its accuracy. But what constitutes a "better transition model"? By our definition, it is a model that predicts transitions with heightened accuracy. Given that transition models learn from collected state trajectories, initial stages of training often grapple with unseen or rarely observed state transitions. Although certain transitions can be generalized from the trained data, a model necessitates a threshold volume of data for effective generalization.

To counteract this, we emphasize training with infrequently observed state transitions. The "Archive" in Go-Explore proves invaluable here. It not only retains rarely observed states but also their associated sequences of trajectory, ensuring that our transition model can be both comprehensive and robust.

B. WODID

In the previous section, we clarify the two drawbacks of MBRL: random sampling and the quality of representation of the transition model. To overcome these challenges, our approach hinges on two foundational principles: the strategic collection of a dataset and the subsequent pretraining of a world model. The dataset is meticulously curated, with states archived over the course of interactions. Unlike conventional methods that rely on random trajectory collections, our archive-driven method ensures a richer variety of states and associated trajectories. Such a dataset, by virtue of its diversity, equips the transition model to encapsulate a broader spectrum of state transitions. Fig.1. is the overview of our system where the left-hand side represents the pretraining process and the right-hand side shows the MBRL structure.

The pretraining phase of our transition model in MBRL is transformative. Instead of the ubiquitous random prefill, which is contingent on a random policy for dataset collection and widely used in general MBRL, we employ the Archive for the prefill collection. This ensures that the model remains immune to the exploration tendencies of a random policy, thereby enhancing the quality of the transition model with more robustness.

The number of steps to run in Go-Explore (iterations) is denoted as N stated in Fig.1. Then, a dataset is created from the Archive. It is constituted by the trajectory of states - sequences of state, reward, and action referred to as s, a, and r respectively.

In WODID, it saves "important" states in the Archive for N steps shown in the left-hand side of Figure 1. We employ "Go-Explore" to run it. At each step, the algorithm first selects a state s saved in the Archive and goes to the state selected. The agent then explores randomly from the selected state s and the destination state s' will be updated to the Archive. Then, we modify the Archive after N steps to a dataset depicted in the middle of Figure 1. This modification depends on what MBRL is used in WODID but the common dataset D contains sequences of state s_n, action a_n, and reward r_n. On the right-hand side of the figure, it represents the pre-training phase of MBRL to construct its transition model and the MBRL used in WODID follows the original cycle of its algorithm.

C. Dataset size for pretraining

Deciding the dataset size is difficult since there are no benchmarks for the size. However, there are some heuristic ideas for the requirements. Here we set two conditions for determining the size of the dataset needed to meet the following rule:

1) Hold more than enough trajectory of states to improve the transition model of MBRL.
2) The total steps of states in Archive are equivalent to the steps the original MBRL collects using random policy.

Since 1 is dependent on tasks an agent is in and 2 is dependent on hyperparameter, we make a heuristic determination for deciding the dataset size to fulfill both 1 and 2 according to the task we use. The detail of heuristic determination is stated in Section V-B.

D. Application on MBRL

Since our framework focuses on the collection of useful data for MBRL and the dataset is used for training the transition model of MBRL, WODID applies to all MBRLs that follow the steps in Section IV-A. MBRLs require further modification when there are system changes in 2) and 3). WODID acts as an external dataset supplier to 1) and therefore the modifications in the algorithms of MBRL are very few. Such a simple-to-apply system to MBRL for its further improvement in exploration and narrowing down the variance can play an easy option for both research and industries to apply on their system.

V. Evaluation

Our evaluation focuses on the following questions.

1) Does WODID have a more stable transition model?
2) Is WODID able to do deeper exploration in a sparse reward environment?

To evaluate these research questions, we first set the condition of environments and the hyperparameters.

A. Environment

We evaluate WODID on a hard-exploration benchmark from Arcade Learning Environment (ALE) [27], [46]: Montezuma's Revenge. It is necessary for an algorithm to make correct sequences of more than tens of actions in order to receive rewards in Montezuma's Revenge. The requirements of these actions make Montezuma's Revenge a hard exploration task and a crucial benchmark for evaluating how deep an algorithm can explore.

Evaluating in a Deterministic Setup: The Atari environment, particularly Montezuma's Revenge, operates under deterministic state transitions. The actions and states are predictably interlinked, ensuring the absence of randomness in transitions. While such an environment simplifies our evaluation, it's worth noting that evaluating in stochastic environments would necessitate a different approach. The task of determining the importance of states would become multifaceted.

B. Experimental Setup

Type of Go-Explore: To collect the dataset pretraining the transition model in WODID, we employ Go-Explore without domain knowledge and goal-conditioned reinforcement learning [47]. Although Go-Explore with demonstration and restore system is the one that achieved a superhuman score in Montezuma's Revenge the model we run uses goal-conditioned reinforcement learning to go to the state selected from the Archive. We used this Go-Explore because we wanted to eliminate human intervention in the framework of WODID so that it can be a human-free system.

Dataset size for the experiment: As presented in Section IV-C, we take a heuristic approach to determining the size of the dataset feeding to MBRL in WODID. To fulfill the conditions we set in 1 and 2 in Section IV-C, we decided to run Go-Explore 10 million (M) steps in Montezuma's Revenge to collect various states in the Archive. This Archive has the highest score of 500, the mean score is around 300, and there are 198 kinds of trajectory of states. We selected this size of Archive for 2 reasons. The first reason is that the total steps of trajectories in the Archive are equivalent to the number of randomly sampled steps from the baseline (Dreamer V2 [18] prefills 50,000 steps). We want to equalize these steps so that we can merely compare the differences between the dataset from the Archive and the dataset from random prefill. Another reason is that heuristically there are sufficient transitions in the Archive considering the highest score and average score it has scored. An agent can score 100, 300, and 100 (sum to 500) in the first few rooms. Transitions in those rooms are composed of dying from touching skulls, dying from jumping from a certain height, climbing ladders, etc. Many transitions in Montezuma's Revenge are variants of the transitions seen in the early rooms. Thus, learning various transitions from the first few rooms not only be useful to improve the quality of representation in the transition model in the early rooms but may also be helpful to represent the transitions in later rooms. Also, from a score perspective, since the Archive after 10M steps has reached to score of 500 and the average is around 300, meaning the agent has some experiences and exploration after reaching the score of 500, the dataset seems to satisfy the conditions.

Overall, our size of the dataset set to fulfill both conditions 1 and 2 and thus we used the Archive after 10M environmental steps.

Model Configuration For the MBRL configuration within WODID, we intentionally employed the Dreamer V2 algorithm. It is a SOTA MBRL algorithm in Atari and we used it to see whether WODID is capable of overtaking it. For the hyperparameters of WODID, we kept the same parameters for the sake of mere comparison with the baseline in exploration and the consistency of exploration. Furthermore, a variance of actor-critic-based RL [48], [49] algorithm that is used in Dreamer V2 is also used in WODID to keep the condition of MBRL identical.

C. Metrics

To answer the stated research questions, our evaluation hinges on two main metrics:

Average Results: We use this metric to assess research question 1. This metric acts as a beacon of the RL agent's exploration prowess within the transition model, as it offers insights into the model's effectiveness in terms of the resilience of the algorithms in exploration. We compare this metric under reward gain, variance, and standard deviation for both WODID and the baseline to evaluate their stability.

Average Environmental Steps: To answer research question 2, we used Average Environmental Steps as its metric. This metric elucidates the model's exploration capabilities and also serves as a reflection of the transition model's representational quality. Since the RL algorithm between our model and the baseline are the same, the number of environmental steps can be a reflection of the quality of the transition model in both architectures. In this metric, we compare the succeeding experiment with our work and the baseline to evaluate the effect of using the pretraining data under our collection mechanism.

TABLE I
COMPARISON OF RESULTS BETWEEN WODID AND THE BASELINE

	Avg Reward	Steps to 2500	Variance	SD
baseline	1080	4,000,000	1,707,000	1306
WODID	2080	6,000,000	882,000	939

D. Results

Average Results As we stated in the metrics, we use average results to evaluate research question 1. Our comparison of WODID against baseline algorithms averaged over five experiments, confirmed WODID's dominance over Dreamer V2 in Table 1 showing the statistics of the results. The numbers speak for themselves: WODID registered a mean score of 2080, dwarfing Dreamer V2's 1080 — more than 90% performance enhancement. This data doesn't just highlight numbers but narrates a story of WODID's unmatched ability to enable the Dreamer V2 agent to navigate deeper states with an unparalleled level of stability. The Variance and Standard Deviation (SD) WODID are nearly 50% and 66% of the baseline respectively also displaying the robustness of policy learning in WODID. Furthermore, Figure 2 and Figure 3 show 5 runs of WODID and the baseline in different seeds where the y-axis is the number of reward gains and the x-axis is the number of environmental steps taken. 4 out of 5 experiments of WODID are able to reach a reward of 2500 whereas only 2 of the baseline experiments were able to reach it and 2 experiments are unable to receive rewards, showing an unstable performance.

These results illustrate that our proposed model, WODID can not only improve the world model to be able to fasten the deep exploration but also improve the Dreamer V2 to have more stable results with a human-free dataset.

Average Environmental Steps Answering research question 2, the evaluation is shown in the reward gain of WODID in Montezuma's Revenge over environmental steps in Figure 4. In the comparison of reward gain over the environmental steps, our findings paint a compelling picture. WODID, in its performance, surpassed the benchmarks set by Dreamer V2. Specifically, WODID achieved a reward of 2500 on average at around 40M steps, contrasting with Dreamer V2's performance, which clocked in at around 60M steps. The result shows that WODID requires only 66% environmental steps to reach its highest score, meaning its transition model is more representative and allows the agent to do deeper exploration. Additionally, WODID still requires fewer

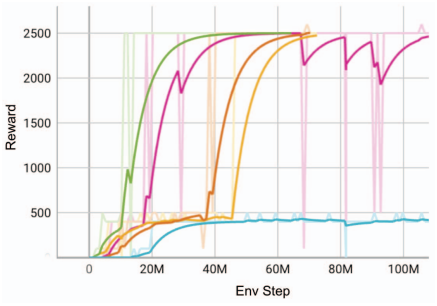

Fig. 2. 5 experiments of WODID in Montezuma's Revenge. The y-axis is the gained reward and the x-axis is the number of environmental steps.

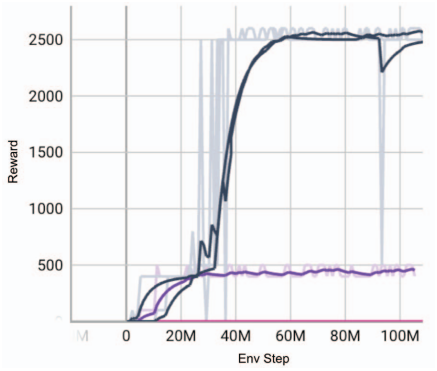

Fig. 3. 5 experiments of the baseline in Montezuma's Revenge.

environmental steps to achieve a reward of 2500 even if we count 10M environmental steps from the preparation of the pretraining dataset (50M in total) although WODID is not fully trained with all transitions of 10M exploration from Go-Explore.

Our evaluation stands as a testament to WODID's potential to explore deep with higher stability, especially when benchmarked against stalwarts like Dreamer V2. The empirical data, gleaned from our rigorous testing, underscores WODID's promise as an innovative approach, especially potent in environments like Montezuma's Revenge, characterized by their sparse rewards and demands for accurate transition to explore deeply. The results, in their clarity and depth, validate our hypotheses and lay a strong foundation for future explorations in this domain.

VI. Conclusion

In this research endeavor, we introduced WODID, a model-based reinforcement learning approach, distinguished by its unique pretraining framework for the transition model. To actualize this, we delineated the features of importance within a transition model and subsequently architected a framework tailored to accrue such pivotal data. The empirical outcomes reaffirmed the potential of WODID, showcasing a marked improvement over the established Dreamer V2 algorithm.

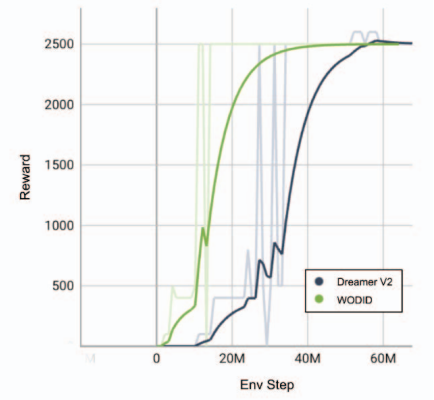

Fig. 4. Comparison of WODID and the baseline in terms of required environmental steps for exploring and performing in Montezuma's Revenge. Results are smoothed.

However, every research endeavor has its constraints, and ours is no exception. A notable limitation of WODID is its singular data collection paradigm. The one-dimensional data collection approach, while effective, offers a ripe avenue for further enhancement. Moreover, our current methodology is underpinned by a specific definition of "important" data and its acquisition mechanism. An enticing direction for future research is to broaden this scope. By generalizing the concept of "important data" for transition models in model-based reinforcement learning, it can potentially create a more versatile framework that isn't bound by the constraints of specific data types or environments such as stochastic environments.

In sum, while WODID stands as a testament to the potential of targeted pretraining in model-based reinforcement learning, the horizon ahead is replete with opportunities for further refinement and innovation. We are optimistic that the insights gleaned from this research will catalyze future endeavors in the realm of reinforcement learning.

Acknowledgment

The authors would like to thank NTT DATA Automobiligence Research Center, Ltd for useful discussions.

References

[1] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, and Sifre Laurent et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[2] OpenAI. Openai five. https://openai.com/research/openai-five, 2018. (Accessed: 2023-12-14).

[3] Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, and David Saxton et al. Unifying count-based exploration and intrinsic motivation. In *Proceedings of Neural Information Processing Systems*, pages 1471–1479, 2016.

[4] Dario Amodei, Christopher Olah, Jacob Steinhardt, Paul Francis Christiano, and John Schulman et al. Concrete problems in AI safety. *Computing Research Repository*, abs/1606.06565, 2016.

[5] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. Back to basics: Benchmarking canonical evolution strategies for playing atari. In *Proceedings of International Joint Conference on Artificial Intelligence*, page 1419–1426, 2018.

[6] Joel Lehman and Kenneth O. Stanley. Novelty search and the problem with objectives. *Genetic Programming Theory and Practice*, pages 37–56, 2011.

[7] Joel Lehman, Jeff Clune, Dusan Misevic, Christoph Adami, and Altenberg et al. The surprising creativity of digital evolution: A collection of anecdotes from the evolutionary computation and artificial life research communities. *Artificial Life*, 26:274–306, 2020.

[8] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Proceedings of Neural Information Processing Systems*, pages 2450–2462, 2018.

[9] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, and Ha David et al. Learning latent dynamics for planning from pixels. In *Proceedings of International Conference on Machine Learning*, pages 2555–2565, 2019.

[10] Richard S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART*, 2:160–163, 1990.

[11] William H. Guss, Mario Ynocente Castro, Sam Devlin, Brandon Houghton, and Noboru Sean Kuno et al. The MineRL competition on sample efficient reinforcement learning using human priors. *Computing Research Repository*, abs/2101.11071, 2021.

[12] Danijar Hafner. Benchmarking the spectrum of agent capabilities. In *Proceedings of International Conference on Learning Representations*, 2022.

[13] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Ken Goldberg, and Pieter Abbeel. Daydreamer: World models for physical robot learning. In *Proceedings of Conference on Robot Learning*, pages 2226–2240, 2022.

[14] Manuel Watter, Jost Tobias Springenberg, Joschka Boedecker, and Martin A. Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In *Proceedings of Neural Infomation Processing Systems*, pages 2746–2754, 2015.

[15] F Ebert, C Finn, AX Lee, and S Levine. Self-supervised visual planning with temporal skip connections. In *Proceedings of Conference on Robot Learning*, pages 344–356, 2017.

[16] Marvin Zhang, Sharad Vikram, Laura M. Smith, P. Abbeel, Matthew J. Johnson, and Sergey Levine. Solar: Deep structured representations for model-based reinforcement learning. In *Proceedings of International Conference on Machine Learning*, pages 7444–7453, 2019.

[17] Danijar Hafner, Timothy P. Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *Proceedings of International Conference on Learning Representations*, 2020.

[18] Danijar Hafner, Timothy P. Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *Proceedings of International Conference on Learning Representations*, 2020.

[19] Jürgen Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proceedings of Conference on Simulation of Adaptive Behavior on From Animals to Animats*, page 222–227. MIT Press, 1991.

[20] Pierre-Yves Oudeyer and F. Kaplan. What is intrinsic motivation? a typology of computational approaches. *Frontiers in Neurorobotics*, 1:6, 2007.

[21] Andrew G. Barto. Intrinsic motivation and reinforcement learning. In *Intrinsically Motivated Learning in Natural and Artificial Systems*, pages 17–47. Springer, 2013.

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

[23] Nicolas Bougie and Ryutaro Ichise. Exploration via progress-driven intrinsic rewards. In *Proceedings of International Conference on Artificial Neural Networks*, pages 269–281, 2020.

[24] Jürgen Schmidhuber. Developmental robotics, optimal artificial curiosity, creativity, music, and the fine arts. *Connection Science*, 18:173–187, 2006.

[25] Jürgen Schmidhuber. Curious model-building control systems. In *Proceedings of International Joint Conference on Neural Networks*, volume 2, pages 1458–1463, 1991.

[26] Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. First return, then explore. *Nature*, 590(7847):580–586, 2021.

[27] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, and Sifre Laurent et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

[28] Tim Salimans and Richard Chen. Learning Montezuma's Revenge from a single demonstration. *Computing Research Repository*, abs/1812.03381, 2018.

[29] Xue Bin Peng, P. Abbeel, Sergey Levine, and Michiel van de Panne. Deepmimic. *ACM Transactions on Graphics*, 37:1–14, 2018.

[30] Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and P. Abbeel. Overcoming exploration in reinforcement learning with demonstrations. In *Proceedings of International Conference on Robotics and Automation*, pages 6292–6299, 2017.

[31] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, and Alexander Novikov et al. A generalist agent. *Transactions on Machine Learning Research*, 2022.

[32] Pomerleau D. *Neural Network Perception for Mobile Robot Guidance*. Springer New York, 1993.

[33] Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, and Tang Jie et al. Video pretraining (vpt): Learning to act by watching unlabeled online videos. In *Proceedings of Neural Information Processing Systems*, pages 24639–24654, 2022.

[34] Peter C. Humphreys, David Raposo, Tobias Pohlen, Gregory Thornton, and Rachita Chhaparia et al. A data-driven approach for learning to control computers. In *Proceedings of Machine Learning Research*, volume 162, pages 9466–9482, 2022.

[35] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Proceedings of Neural Information Processing Systems*, page 4572–4580, 2016.

[36] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In *Proceedings of International Joint Conference on Artificial Intelligence*, page 4950–4957, 2018.

[37] Xin Cao and Xiu Li. Generative adversarial negative imitation learning from noisy demonstrations. In *Proceedings of International Conference on Neural Information Processing*, page 405–416, 2021.

[38] Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, and Yuncong Yang et al. Minedojo: Building open-ended embodied agents with internet-scale knowledge. In *Proceedings of Neural Information Processing Systems*, pages 18343–18362, 2022.

[39] L. Buesing, T. Weber, S. Racaniere, S. Eslami, and D. et al. Rezende. Learning and querying fast generative models for reinforcement learning. *Computing Research Repository*, abs/1802.03006, 2018.

[40] Vaibhav Saxena, Jimmy Ba, and Danijar Hafner. Clockwork variational autoencoders. In *Proceedings of Neural Information Processing Systems*, pages 29246–29257, 2021.

[41] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, and Fethi Bougares et al. Learning phrase representations using RNN Encoder–Decoder for statistical machine translation. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734, 2014.

[42] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedigns of International Conference on Learning Representations*, 2014.

[43] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of International Conference on Machine Learning*, pages 1278–1286, 2014.

[44] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of International Conference on Learning Representations*, 2015.

[45] Danijar Hafner, J. Paukonis, Jimmy Ba, and Timothy P. Lillicrap. Mastering diverse domains through world models. *Computing Research Repository*, abs/2301.04104, 2023.

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

[47] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *Computing Research Repository*, abs/1707.06347, 2017.

[48] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, and Lillicrap Timothy et al. Asynchronous methods for deep reinforcement learning. In *Proceedings of International Conference on Machine Learning*, pages 1928–1937, 2016.

[49] Vijaymohan Konda and John N. Tsitsiklis. *Actor-Critic Algorithms*. Ph.D thesis, Massachusetts Institute of Technology, 2002.