Knowledge Retention in Continual Model-Based Reinforcement Learning

Haotian Fu[*], **Yixiang Sun**[*], **Michael Littman, George Konidaris**
Brown University

Abstract

We propose DRAGO, a novel approach for continual model-based reinforcement learning aimed at improving the incremental development of world models across a sequence of tasks that differ in their reward functions but not the state space or dynamics. DRAGO comprises two key components: *Synthetic Experience Rehearsal*, which leverages generative models to create synthetic experiences from past tasks, allowing the agent to reinforce previously learned dynamics without storing data, and *Regaining Memories Through Exploration*, which introduces an intrinsic reward mechanism to guide the agent toward revisiting relevant states from prior tasks. Together, these components enable the agent to maintain a comprehensive and continually developing world model, facilitating more effective learning and adaptation across diverse environments. Empirical evaluations demonstrate that DRAGO is able to preserve knowledge across tasks, achieving superior performance in various continual learning scenarios.

1 Introduction

Model-based Reinforcement Learning (MBRL) aims to enhance decision-making by developing a world model that captures the underlying dynamics of the environment. A robust world model allows an agent to predict future states, plan actions, and adapt to new situations with minimal real-world trial and error. For MBRL to be effective in dynamic, real-world applications, the world model must incrementally learn and adapt, continually integrating new information as the agent encounters diverse environments and tasks.

Imagine an agent initially exploring a small, confined part of a complex world, like a robot navigating a single room in a large building. At first, the robot learns the dynamics specific to that room, such as the layout of obstacles and how to maneuver around them. As it moves to different rooms and floors, it must learn new dynamics (i.e., new layouts, different lighting conditions, varying types of obstacles), while retaining its understanding of the previously explored areas. Over time, as the robot encounters more and more distinct environments, it becomes familiar with a broader range of settings, eventually developing a comprehensive understanding of the building's overall structure. This incremental learning process aligns with the principles of *continual learning*, where the agent must progressively acquire new knowledge across a sequence of tasks without forgetting earlier experiences. Developing world models that can grow their understanding from one small part of the world toward encompassing an ever broader array of different environments remains a critical and underexplored area in MBRL.

In principle, continual MBRL would allow agents to learn a generalizable model that captures the dynamics needed to support a universal set of tasks. If data from all previous tasks are available, this problem could be tackled effectively using multitask learning strategies (Fu et al., 2022). The agent could leverage the shared structure and learn a comprehensive model that generalizes across

[*]Equal Contribution

Accepted at the Intrinsically Motivated Open-ended Learning workshop at NeurIPS 2024.

tasks. However, in real-world scenarios, agents often **do not have access to the data collected from earlier tasks** due to storage constraints, privacy concerns, or the evolving nature of the environment. In such cases, standard MBRL methods struggle to maintain performance across tasks; as illustrated in Figure 1 and shown in the experiment section, naive model-based RL approaches tend to suffer from catastrophic forgetting, where knowledge acquired from earlier tasks is lost when encoding new experiences. Ideally, as the agent encounters more tasks and diverse environments, its world model should become increasingly complete, accumulating a richer understanding of the dynamics across different scenarios. To achieve this goal, we require a strategy that retains the essential knowledge from prior environments, ensuring that the model builds upon its past experiences even when direct access to earlier data is no longer available.

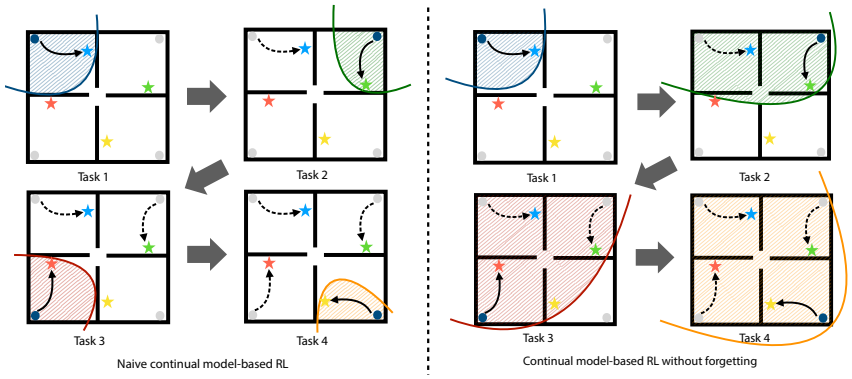

Figure 1: Comparison between the world model learned by naive continual MBRL and MBRL without forgetting. Each task requires the agent to move from the corner of one room to a specific point in the same room. Shaded areas represent the world model's coverage after finishing each task. Naively continually training MBRL (*Left*) tends to suffer the catastrophic forgetting problem—the agent forgets almost everything about the first room after training in the second room. (Our experimental results support this claim.) Our project identifies a continual MBRL method (*Right*) that helps the world model preserve the knowledge of previous tasks even when the old data is no longer available.

Specifically, we propose DRAGO, a novel continual model-based reinforcement learning approach designed to address the challenges of catastrophic forgetting and incomplete world models in the absence of prior task data. DRAGO consists of two key components: Synthetic Experience Rehearsal and Regaining Memories Through Exploration. *Synthetic Experience Rehearsal* uses a continually learned generative model to enable the agent to internally simulate and learn from synthetic experiences that resemble those from prior tasks. This process allows the agent to synthesize representative transitions that resemble prior experience, reinforcing its understanding of previously learned dynamics without requiring access to past data. In the *Regaining Memories Through Exploration* component, we introduce an intrinsic reward mechanism that encourages the agent to actively explore states where the previous transition model performs well. This exploration bridges the gap between different tasks by discovering connections within the environment, leading to a more comprehensive and cohesive world model. By integrating these two strategies, DRAGO enables the agent to incrementally build a complete understanding of the environment's dynamics across a sequence of tasks while effectively mitigating catastrophic forgetting. Our empirical results clearly demonstrate that DRAGO achieves superior performance on challenging continual learning scenarios **without retaining any data from prior tasks**.

2 Dynamics-learning while RegAinG MemOries

2.1 Synthetic Experience Rehearsal

The concept of *Synthetic Experience Rehearsal* draws inspiration from how humans and animals replay and consolidate memories during sleep (Wilson & McNaughton, 1994). We refer to this process as *dreaming* because the agent simulates experiences from its past internally, without direct interaction with the environment. Imagine a robot that has navigated through several rooms in a building. As it progresses to new rooms, it may begin to forget the layouts and navigation strategies

of earlier ones due to limited memory capacity and the inability to revisit those rooms. By internally generating and rehearsing synthetic experiences that mimic its interactions in earlier rooms, the robot can maintain and reinforce its knowledge of how to navigate them. This internal rehearsal helps the robot integrate past experiences with new ones, ensuring a more comprehensive understanding of the entire environment.

Our method leverages a generative model to produce synthetic data that aids in training the dynamics model, thereby preventing forgetting of previously learned dynamics. Specifically, we employ a Variational Autoencoder (VAE) (Kingma & Welling, 2014) that encodes and decodes both states and actions, capturing the joint distribution of state-action pairs encountered in previous tasks. Including actions is crucial, especially in continuous action spaces where randomly sampled actions may not correspond to meaningful behaviors.

When training dynamics model on the current task $\mathcal{T}_i$, we generate synthetic state-action pairs using the VAE trained up to task $\mathcal{T}_{i-1}$. Sampling latent variables z from the prior distribution $p(z)$, we obtain synthetic state-action pairs: $(\hat{s}, \hat{a}) = G_{i-1}(z), \quad z \sim p(z)$, where G_{i-1} represents the generative model from previous tasks. We then use the previous dynamics model T_{i-1} to predict the next states for these pairs: $\hat{s}' = T_{i-1}(\hat{s}, \hat{a})$. This process yields synthetic transitions $(\hat{s}, \hat{a}, \hat{s}')$ that simulate experiences from prior tasks.

We integrate these synthetic transitions directly into the training batches when updating the current dynamics model T_ψ (**the synthetic transitions are not directly added to the current tasks's replay buffer**). By combining synthetic transitions with real transitions from the current task $\mathcal{D}_i$, we form a training dataset: $\mathcal{D}_{\text{train}} = \mathcal{D}_i \cup \hat{\mathcal{D}}$, where $\hat{\mathcal{D}} = \{(\hat{s}, \hat{a}, \hat{s}')\}$. The dynamics model is then trained by minimizing the prediction loss over this combined dataset.

To prevent forgetting within the generative model itself, we adopt a continual training strategy. We generate synthetic state-action pairs using the previous generative model G_{i-1}: $(\tilde{s}, \tilde{a}) = G_{i-1}(\tilde{z}), \quad \tilde{z} \sim p(z)$, and combine these with real data from the current task to form the training dataset for the new generative model: $\mathcal{D}_{\text{gen}} = \mathcal{D}_i \cup \tilde{\mathcal{D}}$, where $\tilde{\mathcal{D}} = \{(\tilde{s}, \tilde{a})\}$. The new generative model G_i is then trained by minimizing the VAE loss over $\mathcal{D}_{\text{gen}}$:

$$\mathcal{L}_{\text{gen}}(\phi_i, \theta_i) = \mathbb{E}_{(s,a)\sim\mathcal{D}_{\text{gen}}} \left[-\mathbb{E}_{z\sim q_{\phi_i}(z|s,a)} \left[\log p_{\theta_i}(s, a \mid z) \right] + \mathrm{KL}\left(q_{\phi_i}(z \mid s, a) \,\|\, p(z) \right) \right]. \quad (1)$$

This continual learning procedure ensures that the generative model retains its ability to produce state-action pairs representative of all previous tasks.

Our method is general and can be applied with other types of generative models. While we use a VAE for its effectiveness and simplicity, alternative models like diffusion models (Ho et al., 2020) or generative adversarial networks (GANs) (Goodfellow et al., 2014) could also be employed to generate synthetic state-action pairs.

2.2 Regaining Memories Through Exploration

While generating synthetic data via a generative model helps mitigate forgetting, it may not fully capture the richness of real experiences, and the agent might still benefit from revisiting areas of the environment related to previous tasks. To further enhance the agent's retention of prior knowledge, we propose an intrinsic reward mechanism that encourages the agent to actively explore states where the previous transition model performs well, effectively "regaining" forgotten memories through real interaction with the environment.

Our approach is inspired by the need to complement the generation-based "dreaming" method with actual exploration that bridges the **gap** between different tasks. The generative model can produce states from prior tasks, but these imagined states might not be naturally encountered or connected within the current task's environment. Consider the earlier example of a robot exploring different rooms within a building. The "Dreaming" method introduced in the last section can generate imagined states from previously visited rooms, but without actual exploration, the robot might not find the doorways or corridors connecting these rooms to its current location. Our intrinsic reward incentivizes the robot to search for these connections, enabling it to discover pathways that link the new room to the old ones. Without exploring the actual environment to find these connections, the agent's world model remains fragmented, lacking a cohesive understanding of how different regions relate.

Specifically, during training on task $\mathcal{T}_i$, we introduce an intrinsic reward r^i_{cont} designed to guide the agent towards states that are familiar to the previous transition model T_{i-1} but less familiar to the current model T_i. The intrinsic reward is defined as:

$$r^i_{\text{cont}}(s_t, a_t, s_{t+1}) := \sigma\left(-\log|T_{i-1}(s_t, a_t) - s_{t+1}|\right) - \alpha \cdot \sigma\left(-\log|T_i(s_t, a_t) - s_{t+1}|\right), \quad (2)$$

where σ denotes the sigmoid function, and α is a weighting coefficient that balances the two terms.

Intuitively the first term assigns higher rewards when the previous transition model T_{i-1} predicts the next state s_{t+1} accurately. This incentivizes the agent to revisit states that were well-understood in previous tasks. The second term penalizes the agent for visiting states where the current model T_i already has low prediction error. This encourages the agent to explore less familiar areas to improve the current model's understanding.

By actively exploring and connecting different regions, the agent's world model becomes more comprehensive, capturing the dynamics across tasks more effectively. Revisiting familiar states reinforces prior knowledge, reducing the tendency of the model to forget previously learned information. This approach complements the synthetic data generation in Section 2.1 by providing actual experience that reinforces the agent's knowledge. Compared to pure novelty-seeking exploration strategies (Pathak et al.), our method emphasizes revisiting and reinforcing previously learned dynamics.

3 Experiments

We evaluated DRAGO on three continual learning domains. For each domain, we let the agent train on a sequence of tasks, where the tasks share the same transition dynamics but different reward functions. Although the transition dynamics are the same, the training tasks are designed in a way such that to solve each task only part of the state space's transition dynamics needs to be learned and different tasks involve learning transition dynamics corresponding to different parts of the state space **with a small overlap**. We evaluate the agent's continual learning performance on test tasks that requires the combination of knowledge from more than one previously learned tasks. For example, to better transfer on *Cheetah jump2run* the agent is expected to still remember the knowledge learned in *Cheetah run* even after continual training on *Cheetah jump*. These transfer tasks are designed to test the agent's ability to retain knowledge from previous tasks, as solving them requires understanding multiple tasks.

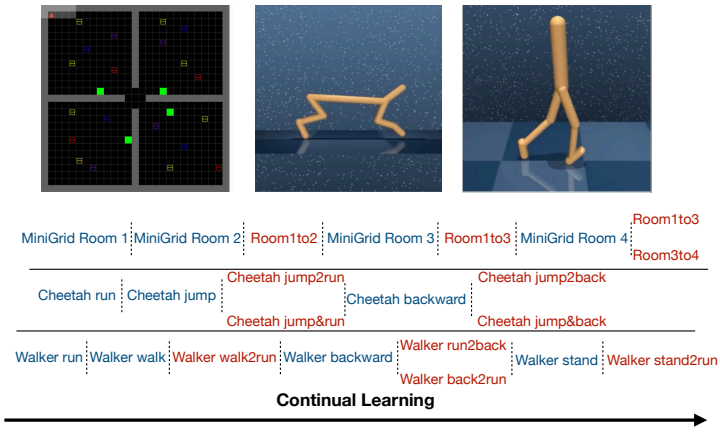

Figure 2: Visualization of the evaluated domains. Task names in Blue denote the continual **training** tasks; Task names in Red denote the **test** tasks. More details about the tasks and environments can be found in the appendix.

As shown in Figure 3, we find that the proposed method DRAGO achieves the best overall performance compared to all the other approaches across three domains. The results demonstrate its advantage in continual learning settings by effectively retaining knowledge from previous tasks and transferring it to new ones. We can also see that naively continual Model-based RL may suffer from severe plasticity loss: Continual TDMPC constantly performs worse than learning from scratch baseline. Equipped with EWC, it can achieve better overall performance but still not as good as DRAGO. But DRAGO does not fully alleviate the plasticity loss, in *Cheetah Jump and runbackward* (Last plot in the mid row of Figure 3), learning from scratch still has the best performance, but we can see that DRAGO still improves a lot compared to Continual TDMPC — the Continual MBRL baseline it is built on.

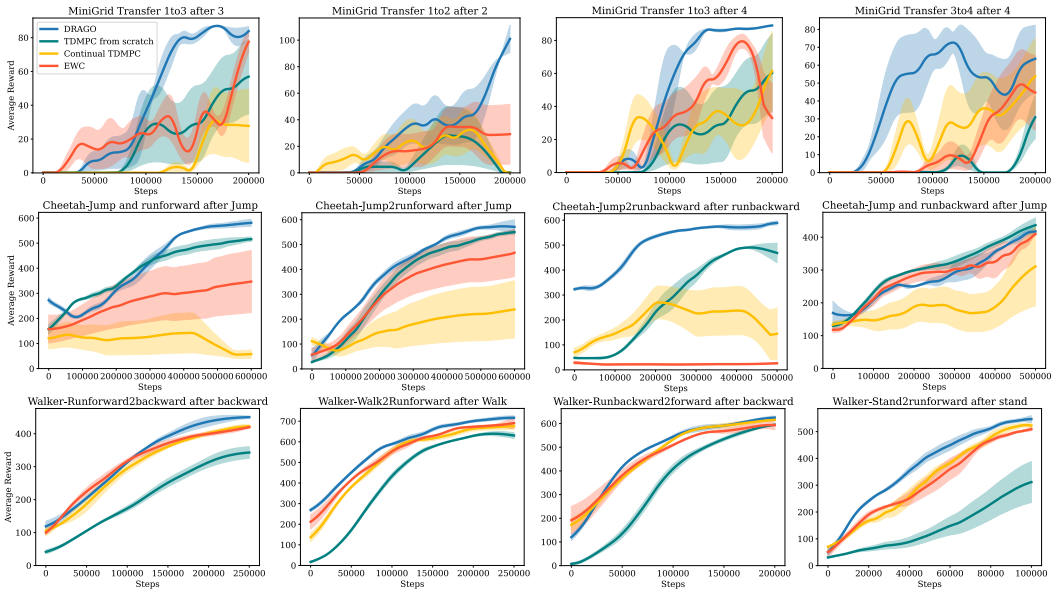

Figure 3: We evaluate the continual learning transfer performance on 12 tasks (3 domains, 4 tasks each) that are not seen during the agent's previous training. For each test task of MiniGrid, the agent starts in one room and have to move to the goal in another room. E.g., *Transfer 3to4 after 4* means that after sequentially training on four tasks, the agent is tested on a new task where it starts in room 3 and the target position is in room 4. For each test task of Cheetah and Walker, the agent has to start from a state in one locomotion mode and the goal is to switch to another mode. E.g., *Jump2runforward after Jump* means that after training on Cheetah-Jump, the agent is tested on a new task where it starts in one state of the jumping mode, and the goal is to run forward.

In Figure 4, we also visualize the prediction accuracy of the learned world models across the whole gridworld, comparing just naively continually training TDMPC and our method. The results are aligned with our intuition. Without other counter-forgetting techniques, world models easily forget almost everything it has learned in previous tasks and are only accurate in the transition space that is related to the current task. In contrast, DRAGO is able to retain most of the knowledge learned in previous tasks and have a more and more complete world model as it continually trains on different tasks, which leads to the performance gain of transferring on new tasks as we show in Figure 3.

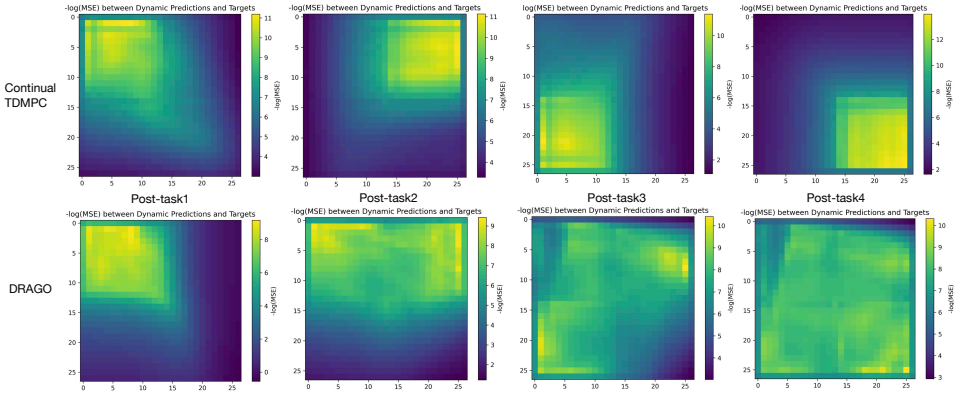

Figure 4: Prediction accuracy of the learned world models across the entire gridworld after each task. The heatmaps compare the performance of naive continual training of TDMPC (top row) with our proposed DRAGO method (bottom row) after Tasks 1 to 4. The results show that continual MBRL suffers from significant forgetting, maintaining accuracy only in regions relevant to the current task, whereas DRAGO effectively retains knowledge from previous tasks, leading to a more comprehensive world model and improved performance in new tasks.

References

Haotian Fu, Shangqun Yu, Michael Littman, and George Konidaris. Model-based lifelong reinforcement learning with bayesian exploration. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pp. 2672–2680, 2014.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2778–2787. PMLR.

Mathew Wilson and Bruce L. McNaughton. Reactivation of hippocampal ensemble memories during sleep. *Science*, 265 5172:676–9, 1994. URL `https://api.semanticscholar.org/CorpusID:890257`.