Dream On: DreamerV3 for Continual Reinforcement Learning via Permanent-Transient Critic Decoupling

Zihan Wang, Alex Koran, Thomas Lai
260825558, 260918404, 260864949

Abstract

Reinforcement Learning (RL) has emerged as a potent framework within Machine Learning, enabling agents to learn optimal behaviors through environmental interactions. The versatility of RL is evident in its adaptation to Continual Reinforcement Learning (CRL), where agents learn across a sequence of tasks, leveraging past knowledge to master new challenges. In our project, we adapted the state-of-the-art DreamerV3 for the CRL paradigm. We propose Continual-DreamerV3, which decouples the traditional critic function into a transient and a permanent critic, aiming to address the stability-plasticity dilemma inherent in CRL. Our experimental analysis on the DeepMind Control Suite demonstrates that our approach mitigates catastrophic forgetting and facilitates rapid adaptation to new tasks by resetting specific components when task boundaries are encountered. The results show that our method outperforms the original DreamerV3 model in retaining prior knowledge and adapting to new tasks, with selective reinitialization of the value function emerging as a particularly effective strategy. The findings suggest promising avenues for future research in developing more versatile and robust CRL systems.

1 Introduction

Within the realm of Machine Learning, Reinforcement Learning has become a very powerful technique for training agents to interact optimally with reward-based task environments through trial and error. RL has the benefit over other ML techniques of being able to learn directly through environmental interactions, making it particularly well suited for problems where explicit supervision is unavailable. Current RL algorithms can be categorized into model-free and model-based. Model-based RL algorithms consist of a model of the environment dynamics, which can be used to simulate data to train value or policy functions. DreamerV3 Hafner et al. (2023) is an example of a state-of-the-art model-based RL agent that also uses the actor-critic framework.

One promising subfield of RL is Continual Reinforcement Learning (CRL). In this more recent domain, agents attempt to train sequentially on various similar tasks and environments, utilizing previously attained knowledge. This addresses the concept of lifelong learning in RL, allowing agents to generalize their actions and policies throughout a large range of different environments and tasks. The main challenge in CRL is the stability-plasticity dilemma, which is the trade-off between stability, i.e., the ability of an agent to retain its previous knowledge, and plasticity, i.e., the ability of an agent to adapt to new changes. One potential solution to this dilemma is the decoupling of the value function into a transient and a permanent one, which was proposed by Anand and Precup (2024).

In our project, we adapted the DreamerV3 agent to the continual reinforcement learning problem by decoupling its critic function into a transient and a permanent critic. We then tested our Continual-DreamerV3 on the DMC Proprio environment. For our experiments, we assumed that the task boundaries are known, and investigated the effect on performance of DreamerV3 and Continual-

Preprint. Under review.

DreamerV3 when different components of the agents are reset when the task changes. We found that by decoupling the critic function, DreamerV3 can be modified to forget less and adapt better in the continual setting, compared to the original critic function.

2 Background

Dreamer V3 Hafner et al. (2023) is a reinforcement learning algorithm which accomplishes multiple complex tasks in various environments without requiring domain specific tuning. It is based on several components, including actor and critic components, and a world model, which are concurrently trained on replayed experiences. Actor-critic methods consist of an actor that learns a policy function that maximizes the expected cumulative reward, and a critic that evaluates the potential actions based on the estimated action value function. The world model component is implemented as a Recurrent State Space Model (RSSM). This consists of an encoder which maps sensory inputs to stochastic representations, a sequence model which predicts the sequence of these representations based on past actions, and a Dynamics predictor which predicts the future state of the environment. In addition, there is a reward predictor, a continue predictor for episode continuation flags, and decoder.

Hafner et al. (2023) proposes several novelties such as symlog predictions to aid with reconstructing inputs and predicting rewards, binning to transform continuous value predictions into a discrete set of outcomes, and a normalization in their loss function for exponential decay based on batch percentile.

The paper "Prediction and Control in Continual Reinforcement Learning" Anand and Precup (2024) offers a new potential solution to the Stability-Plasticity dilemma in reinforcement learning. The proposed solution suggests a two component system. The value function is broken into a transient component, which has a recency bias capturing current information into the policy, and a permanent component, which focuses on retaining past experiences and general knowledge.

3 Methodology

Inspired by Anand and Precup (2024), we decoupled the critic function of DreamerV3 into a transient critic and $V_w^{(T)}(s)$ a permanent critic $V_\theta^{(P)}(s)$, which are parameterized by w and θ, respectively. The overall value function estimation is given by $V^{(PT)}(s) = V_\theta^{(P)}(s) + V_w^{(T)}(s)$. The architecture of the transient critic is

$$V_w^{(T)}(s) = \texttt{symexp}(p_w^{(T)}(s)^T B), \tag{1}$$

where `symexp` is the symexp function described in Hafner et al. (2023), $p_w^{(T)}(s)$ is a neural network parameterization, and $B := [-20, ..., +20]$ are bins that discretize the state space Hafner et al. (2023). Similarly, the permanent critic is $V_\theta^{(P)}(s) = \texttt{symexp}(p_\theta^{(T)}(s)^T B)$. We train the transient critic network with the original episodic critic loss of DreamerV3:

$$\mathcal{L}_w = \sum_{t=1}^{T} \texttt{sg}(\texttt{two-hot}(\texttt{symlog}(R_t^\lambda)))^T \log p_w(s_t), \tag{2}$$

where t denotes time steps in an episode of length T, `sg` denotes "stop-gradient", `two-hot` denotes two-hot encoding, `symlog` denotes the symlog function described in Hafner et al. (2023), and R_t^λ is a bootstrapped λ-return, i.e.,

$$R_t^\lambda = r_t + \gamma c_t((1-\lambda)V^{(PT)}(s_{t+1}) + \lambda R_{t+1}^\lambda), \tag{3}$$

where c_t is a binary stop signal, and λ is the TD-λ hyperparameter. In this setup, the transient critic parameterization $p_w(s)$ is trained with a target determined by the overall value function $V^{(PT)}(s)$, and gradient descent on $p_w(s)$ is performed at every critic training step. On the other hand, gradient descent on $p_\theta(s)$ is only performed at every k critic training steps, in which the objective is to make the permanent critic learn the overall value function. To achieve this, we train $V_\theta^{(P)}(s)$ via mean-squared error, i.e.,

$$\mathcal{L}_\theta = \frac{1}{T}\sum_{t=1}^{T} ||V_\theta^{(P)}(s_t) - \texttt{sg}(V^{(PT)}(s_t))||^2. \tag{4}$$

Note that $\mathcal{L}_\theta$ is differentiable w.r.t. θ and its gradient can be written as

$$\nabla_\theta \mathcal{L} = \sum_{t=1}^{T} (V_\theta^{(P)}(s_t) - \mathtt{sg}(V^{(PT)}(s_t)))\nabla_\theta V_\theta^{(P)}(s_t) \tag{5}$$

$$= \sum_{t=1}^{T} (V_\theta^{(P)}(s_t) - \mathtt{sg}(V^{(PT)}(s_t)))\nabla \mathtt{symexp}(p_\theta^{(T)}(s)^T B) B^T \nabla_\theta p_\theta^{(P)}(s_t). \tag{6}$$

The rest of our method is consistent with the original DreamerV3.

4 Experiments

We tested Continual-DreamerV3 and the original DreamerV3 on the DeepMind Control Suite environment Tassa et al. (2018). Specifically, we ran both agents on the "Walker Stand" and "Walker Walk" tasks, which are alternated every 100,000 steps. To ensure a fair comparison with DreamerV3, reduced the number of parameters in our transitent and permanent critics by half. We also used two separate Adam optimizers for the transient and permanent critics. We also conducted an experiment in which, using the task boundary information, we explored re-initializing various networks and their optimizers for both Continual-DreamerV3 and the original DreamerV3. The networks that were re-initialized include the Recurrent State Space Model (RSSM), the reward predictor MLP, the continuation predictor MLP, the actor network, and the critic network (including permanent and/or transient critics). The agents' policies are evaluated per 2000 training steps, where we compare each method based on the mean return over 10 testing episodes.

4.1 Comparison with original DreamerV3 on Continual Tasks

Comparative analysis of average returns demonstrated our model's ability to retain prior knowledge compared to the original DreamerV3, as illustrated in Figure 1. We also included the original DreamerV3 framework with complete reinitialization. Results indicate that while the original DreamerV3 model fails to learn the 'walk' task following task changes, it maintains proficiency in the 'stand' task. In contrast, our method showed superior adaptability in the second task, effectively learning to 'walk'. Moreover, our model sustained knowledge for subsequent 'stand' and 'walk' tasks, showcasing its robustness in continual learning scenarios.



Figure 1: Results on DMC.

4.2 Further Analysis of Reinitialization Strategies

We further explore the reinitialization strategies, as illustrated in Figure 2. Our findings indicate that reinitializing the value function yields the best performance upon task transitions. Specifically, from the second 'walk' task, the model swiftly recovers its average return, indicating retention of the 'walk' knowledge. Additionally, the transient value model appears to be particularly beneficial for reinitialization as it consistently shows improved performance when involved in the combination, whereas reinitializing the world model also proves effective but at a higher cost and slower convergence compared to the value-only reinitialization.

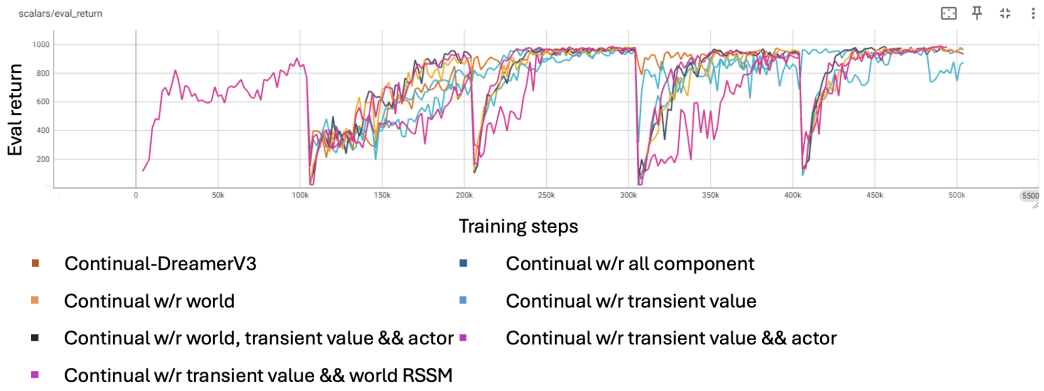


Figure 2: Combination of Reinitialization Strategies Across Different Model Components

Surprisingly, models that do not undergo any reinitialization perform better in the second 'walk' task. A deeper examination of the agent's behavior at a human-perceivable level reveals insights into this phenomenon. This can be attributed to the inherent design of the 'stand' and 'walk' tasks. In the 'stand' task, the reward structure combines terms that encourage maintaining an upright torso and achieving a minimal torso height. Consequently, the model learns a trivial solution to maintain an upright torso while still capable of walking.

In the evaluation of hyper-parameter tuning, depicted in Figure 3, the training performance under different configurations of λ and k is presented. The setting with $\lambda = 0.5$ and $k = 10$, associated with the blue line, demonstrates a consistent upward trend in evaluation returns, achieving stability beyond 200K steps. Altering the hyper-parameters to $\lambda = 0.3$ and $k = 25$, represented by the green line, introduces higher variance during training, with a gradual approach to convergence noticeable after 300K steps. These findings indicate that the system exhibits a degree of resilience to fluctuations in λ and k values. Nonetheless, the subtle distinctions in trajectory between the two lines underscore the importance of fine-tuning these parameters for optimal long-term learning performance.

5 Conclusion

In conclusion, our adaptation of the DreamerV3 model to continual reinforcement learning scenarios demonstrates significant potential for enhancing the stability-plasticity balance through the decoupling of the critic function into transient and permanent components. Our experimental results confirm that this novel approach enables the Continual-DreamerV3 to not only retain prior knowledge more effectively but also to adapt more adeptly to new tasks compared to the original DreamerV3 setup. Particularly, the reinitialization strategies implemented show a pronounced impact on performance, with selective reinitialization of the value function proving most effective.

In future work, we aim to extend our investigations to encompass a broader array of environments, such as DMC Vision, MiniAtar, and Crafter. Additionally, exploring different action spaces will be a priority, with the goal of designing novel algorithms that are task-agnostic. These developments will help to enhance the adaptability and generalizability of our models across diverse and complex scenarios.

References

Anand, N. and Precup, D. (2024). Prediction and control in continual reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Hafner, D., Pasukonis, J., Ba, J., and Lillicrap, T. (2023). Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*.

Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., de Las Casas, D., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., Lillicrap, T., and Riedmiller, M. (2018). Deepmind control suite.

A Adjustment of hyper-parameters

In this appendix, we detail the importance of tuning the hyperparameters λ and k for optimizing our model's performance. Tuning these parameters is crucial as it helps in refining the model to better suit specific performance metrics, making such adjustments worthwhile for enhancing model efficacy.

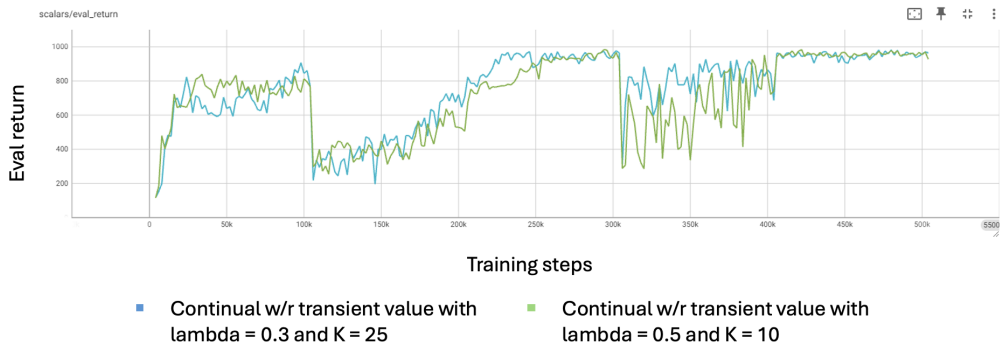


Figure 3: Adjustment of hyper-parameters λ and k across two settings, with λ values set at 0.5 and 0.3, and k values at 10 and 25, corresponding to the blue and green lines, respectively.