Do Transformer World Models Give Better Policy Gradients?

Michel Ma [*1 2] **Tianwei Ni** [1 2] **Clement Gehring** [1 2] **Pierluca D'Oro** [*1 2] **Pierre-Luc Bacon** [1 2]

Abstract

A natural approach for reinforcement learning is to predict future rewards by unrolling a neural network world model, and to backpropagate through the resulting computational graph to learn a policy. However, this method often becomes impractical for long horizons since typical world models induce hard-to-optimize loss landscapes. Transformers are known to efficiently propagate gradients over long horizons: could they be the solution to this problem? Surprisingly, we show that commonly-used transformer world models produce *circuitous gradient paths*, which can be detrimental to long-range policy gradients. To tackle this challenge, we propose a class of world models called Action-conditioned World Models (AWMs), designed to provide more direct routes for gradient propagation. We integrate such AWMs into a policy gradient framework that underscores the relationship between network architectures and the policy gradient updates they inherently represent. We demonstrate that AWMs can generate optimization landscapes that are easier to navigate even when compared to those from the simulator itself. This property allows transformer AWMs to produce better policies than competitive baselines in realistic long-horizon tasks.

1. Introduction

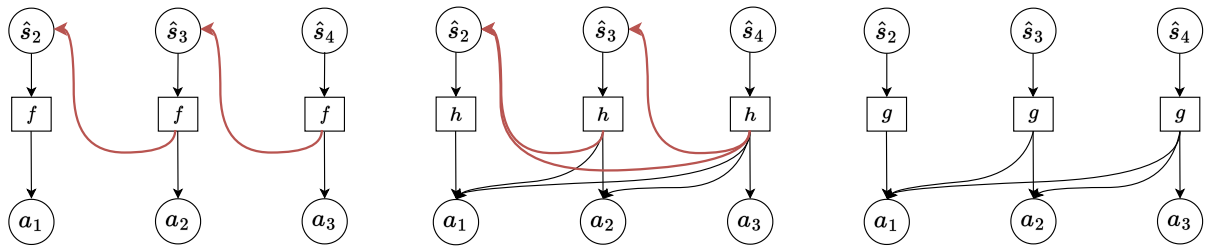
Given a class of parameterized policies, policy optimization methods aim to find parameters $\boldsymbol{\theta}$ that maximize a performance criterion J – typically the expected sum of rewards over the episodes of experiences generated by an agent in an environment. When the environment dynamics are known and differentiable, the gradient of the objective J can be directly evaluated through automatic differentiation and used to optimize J by gradient ascent. If the dynamics are unknown, a differentiable world model can be trained on interaction data, whose predicted states will be backpropagated through. This model-based approach to reinforcement learning (RL) has been explored early on in the field with Werbos (1974); Schmidhuber (1990); Miller et al. (1995) and led to contemporary work in deep RL such as Hafner et al. (2021; 2023). While backpropagation through time (Werbos, 1990) using a world model is conceptually appealing and readily compatible with deep learning practice, most attempts to scale it up have been unsuccessful beyond a few dozen steps (Hafner et al., 2021; Xu et al., 2022; Ghugare et al., 2023). Indeed, even in the presence of perfectly accurate differentiable simulators, policy gradients obtained by backpropagation through time can quickly become unstable over long horizons (Parmas et al., 2018; Suh et al., 2022; Metz et al., 2021).

This problem of stability in differentiating through world models over long horizons echoes similar challenges observed in sequence modeling. The latter field has experienced numerous advances in deep network architectures specifically aimed at alleviating this challenge. As a notable example, transformers were designed to directly propagate gradients from any output to any input arbitrarily far in the past, within their context length (Vaswani et al., 2017). It is, therefore, natural to ponder whether using a transformer as a world model and differentiating through it could overcome the difficulties encountered in model-based policy optimization. Or, in other words, to ask the question: *do transformer world models give better policy gradients?* In this paper, we provide a new answer to this question: surprisingly, transformers do not necessarily confer the same benefit for policy optimization when used as world models conditioned on the full history of states and actions. Instead, conditioning solely on a sequence of actions tends to yield better-behaved policy gradients. This, in turn, leads to improved temporal credit assignment and policy performance.

We first show that backpropagating naively through a transformer predicting the next state conditioned on the full history (Micheli et al., 2022; Chen et al., 2022; Robine et al., 2023) does not necessarily lead to better policy gradients in long-horizon tasks. We demonstrate, conceptually and theoretically, that this phenomenon can be attributed to the

Code is available at https://github.com/micklethepickle/action-conditioned-world-models. [*]Equal contribution [1]Mila, Montreal, Canada [2]Department of Computer Science, University of Montreal, Canada. Correspondence to: Michel Ma <michel.ma@mila.quebec>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).



(a) Markovian world model (b) History-conditioned World Model (c) Action-conditioned World Model

Figure 1. **Diagram illustrating gradient flows through different world model types from states to actions.** Circuitous (longer than necessary) gradient paths go through connections highlighted in red. An Action-conditioned World Model has no circuitous gradient paths, allowing gradients to directly flow from states to actions through a single application of a world model.

existence of *circuitous gradient paths*: long state-to-state gradient paths generated during the autoregressive unrolling of the model (see Figure 1).

To avoid the creation of those longer-than-necessary gradient paths, we propose an alternative class of world models, which we call *Action-conditioned World Models* (AWMs), of the form $\hat{s}_{t+1} = g(s_1, a_1, \ldots, a_{t-1}, a_t)$ conditioned solely on the history of past actions and an initial state. An AWM is forced to create its own internal dynamics, potentially very different from the one of the environment, while being able to predict future states. This inductive bias constrains the temporal accumulation of gradients to only happen within the neural network of the AWM, fully harnessing its gradient properties.

We show theoretically that an AWM directly inherits the gradient propagation properties of the underlying network architecture. Thus, AWMs yield policy gradients subject to bounds similar to the ones previously derived in deep sequence models (Pascanu et al., 2013; Kerg et al., 2020). Additionally, our theoretical framework provides a unified perspective on many backpropagation-based policy optimization methods by exposing the relationship between the network architecture and the policy gradient updates that they induce. For instance, it allows us to cast backpropagation through unrolled Markovian dynamics (Heess et al., 2015) as a specific AWM instantiated with a vanilla recurrent neural network. Our framework paves the way for new algorithmic developments by fostering novel synergies between the study of network architectures in deep learning and the policy optimization updates they implicitly express.

Finally, through a series of experiments, we showcase the remarkable empirical properties of backpropagation-based policy optimization with AWMs. We demonstrate in illustrative domains with chaotic or non-differentiable dynamics that AWMs can give better policy gradients than the simulator itself, by forcing the policy optimization landscape to be easy to optimize. This leads to successful policy optimization via backpropagation even when the same procedure would fail with the true underlying model. On a testbed of realistic long-horizon tasks (Howe et al., 2022), we benchmark policy optimization with transformer AWMs against other competitive model-free and model-based methods, showing superior performance.

2. Background

Problem definition We are interested in deterministic[1] discrete-time finite-horizon Markov Decision Processes (MDPs) (Fairbank, 2014), defined as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, f, r, H, s_1)$, where $\mathcal{S} \subseteq \mathbb{R}^n$ is the state space, $\mathcal{A} \subseteq \mathbb{R}^m$ is the action space, $f : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the differentiable transition dynamics, $r : \mathcal{S} \to \mathbb{R}$ is the known differentiable reward function, H is the horizon and $s_1 \in \mathcal{S}$ is the initial state. The behavior of an agent in the environment is described by a policy $\pi_{\boldsymbol{\theta}} : \mathcal{S} \to \mathcal{A}$, belonging to a space of parameterized differentiable deterministic[2] stationary Markov policies $\Pi = \{\pi_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \mathbb{R}^d\}$. For clarity, we focus on stationary rewards and dynamics, but our results could readily be adapted to the non-stationary case.

Backpropagation-based Policy Optimization The goal of the agent is to maximize the cumulative episodic rewards. In the rest of the paper, we consider the following problem:

$$\begin{aligned} \text{maximize} \quad & J^f(\boldsymbol{\theta}; H) := \sum_{t=1}^{H} r(s_t), \\ \text{subject to} \quad & s_{t+1} = f(s_t, a_t), \\ & a_t = \pi_{\boldsymbol{\theta}}(\mathrm{sg}[s_t]),\ t = 1, \ldots, H-1 \\ \text{given} \quad & s_1. \end{aligned} \tag{1}$$

where $\mathrm{sg}[\cdot]$ is the stop-gradient operator. Model-based policy gradient methods solve this problem by unrolling (simulating) the dynamics forward and computing the gradient of

[1] See Laidlaw et al. (2023) for a discussion on the importance of the study of deterministic MDPs.

[2] This assumption can be relaxed to a stochastic policy using the reparameterization trick (Kingma & Welling, 2014).

Algorithm 1 Backpropagation-based Policy Optimization (BPO)

Input: Initial buffer $\mathcal{B}$, initial policy parameters $\boldsymbol{\theta}$, initial model parameters $\boldsymbol{\psi}$, learning rates $\{\alpha_{\boldsymbol{\theta}}, \alpha_{\boldsymbol{\psi}}\}$, world model class $\mathcal{W}$.

while not exceeding training steps **do**
  Collect an episode with $\pi_{\boldsymbol{\theta}}$ and add it to $\mathcal{B}$
  for each world model learning step **do**
    $\boldsymbol{\psi} \leftarrow \boldsymbol{\psi} - \alpha_{\boldsymbol{\psi}} \nabla_{\boldsymbol{\psi}} \ell^{\mathcal{W}}(\tau; \boldsymbol{\psi}), \quad \tau \sim \mathcal{B}$
  end for
  for each policy learning step **do**
    Compute $J^{\mathcal{W}}(\boldsymbol{\theta}; \boldsymbol{\psi})$ by unrolling the world model
    Compute $\nabla_{\theta} J^{\mathcal{W}}(\boldsymbol{\theta}; \boldsymbol{\psi})$ by backpropagation
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} J^{\mathcal{W}}(\boldsymbol{\theta}; \boldsymbol{\psi})$
  end for
end while

the corresponding unconstrained problem. The presence of the stop gradient operator in our formulation aligns with the contemporary practice (Hafner et al., 2021; 2023; Ghugare et al., 2023). Its inclusion in our formulation does not impact performance (see Appendix C.1 for an ablation) and simplifies our theoretical analysis.

When the true dynamics f are unknown, a transition function $\hat{f}_{\boldsymbol{\psi}}(s_t, a_t)$ belonging to a space of parameterized differentiable functions $\mathcal{F} = \{f_{\boldsymbol{\psi}} : \boldsymbol{\psi} \in \mathbb{R}^{d_\psi}\}$ must be learned to perform policy optimization. Given transitions s_t, a_t, s_{t+1} sampled from the environment, an approximate Markovian world model $\hat{f}_{\boldsymbol{\psi}}$ can be learned with the loss $\ell^{\hat{f}}(\tau; \boldsymbol{\psi}) = \sum_{t=1}^{H-1} \|s^{\tau}_{t+1} - \hat{f}_{\boldsymbol{\psi}}(s^{\tau}_t, a^{\tau}_t)\|^2$, where $\tau = (s^{\tau}_1, a^{\tau}_1, \ldots, s^{\tau}_H)$ is a trajectory. For the remainder of this paper, consider $\| \bullet \|$ to denote the L^2 norm. Given a learned model, policy and initial state, the corresponding return $J^{\hat{f}}(\boldsymbol{\theta}; \boldsymbol{\psi})$ can be computed through rollouts and the gradient $\nabla_{\boldsymbol{\theta}} J^{\hat{f}}(\boldsymbol{\theta}; \boldsymbol{\psi})$ can be obtained by backpropagation. We call this basic algorithm, outlined in Algorithm 1, *Backpropagation-based Policy Optimization* (BPO).

3. Policy Gradient Computation with History-conditioned World Models

When running Algorithm 1, the performance of the resulting policy depends on the quality of the gradient approximation. Effective long-term credit assignment capabilities are required for computing the gradient $\nabla_{a_t} r(s_{t+k})$, as provided by the world model, for potentially very large values of k in challenging control tasks. This means understanding how a reward at a temporally distant time changes in response to variations in a specific action taken at an earlier moment. Therefore, the long-term credit assignment properties of a model-based policy optimization algorithm are intimately tied to the structure of the unrolled world model.

One fundamental fact about backpropagation is that its success in learning long-term dependencies is intimately linked to the length of the paths involved in the neural network's gradient computation (Pascanu et al., 2014; Zhang et al., 2016). Indeed, the success of transformers is often attributed to this phenomenon: "one key factor affecting the ability to learn such dependencies is the length of the paths forward and backward signals have to traverse in the network" (Vaswani et al., 2017). When trained for sequence modeling tasks, transformers stand out compared to other architectures because the length of these paths does not linearly increase with the sequence length.

Recent model-based RL approaches have introduced the use of transformer History-conditioned World Models (HWMs), which predict the next state as $\hat{s}_t = h(s_{1:t-1}, a_{1:t-1})$ based on the full history of states and actions (Micheli et al., 2022; Robine et al., 2023). Their use leads to improved performance due to better prediction abilities but, surprisingly, these approaches were not able to take advantage of the gradient propagation properties of the transformer architecture, as exemplified by previous work which only unrolls them for a few steps (Chen et al., 2022).

In practice, an HWM is unrolled in this manner to compute the cumulative reward for a given policy:

$$\begin{aligned} \text{maximize} \quad & J^h(\boldsymbol{\theta}; H) := \sum_{t=1}^{H} r(\hat{s}_t), \\ \text{subject to} \quad & \hat{s}_{t+1} = h(s_1, a_1, \hat{s}_2, a_2, \ldots, \hat{s}_t, a_t), \\ & a_t = \pi_{\boldsymbol{\theta}}(\text{sg}[\hat{s}_t]),\ t = 1, \ldots, H-1 \\ \text{given} \quad & s_1. \end{aligned} \tag{2}$$

HWMs are typically learned by employing a prediction objective $\ell^h(\tau; \boldsymbol{\psi}) = \sum_{t=1}^{H-1} \|s^{\tau}_{t+1} - h_{\boldsymbol{\psi}}(s^{\tau}_{1:t}, a^{\tau}_{1:t})\|^2$, similar to the one employed for a Markovian model, where $s^{\tau}_{1:t}$ and $a^{\tau}_{1:t}$ denote subsequences of states and actions from an episode τ. While history-conditioned dynamics are in principle not necessary in an MDP[3], one may hope that the gradient properties of transformers might manifest themselves positively in the policy gradient. It stands to reason that the policy gradients through a transformer may be able to more effectively capture long-term dependencies than a Markovian model would. Nonetheless, this apparently natural usage of a transformer has unintended consequences for policy optimization.

Visually, we show in Figure 1 the gradient paths induced by unrolling a *state-based* world model, whether Markovian (Figure 1a) or history-dependent (Figure 1b), from state predictions to actions. The impact of an action on a

[3]Policy gradients can be computed through equation 2 similarly in partially observable MDPs (POMDP), where states are replaced with observations. The resulting analysis would, therefore, still hold in a POMDP.

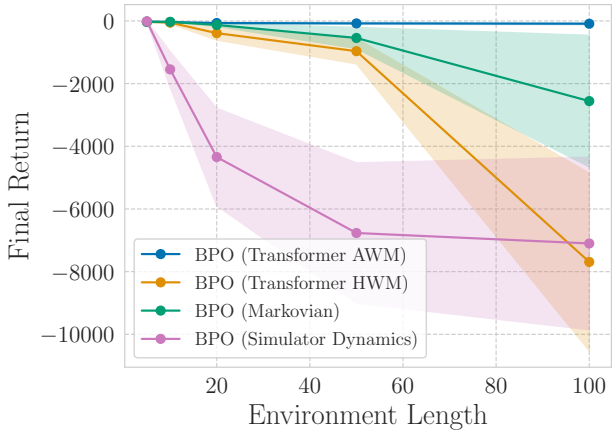


Figure 2. **Policy optimization with transformer AWMs outperforms all BPO baselines in chaotic environments.** Final performance of BPO using different world models and the differentiable dynamics on the double-pendulum environment (10 seeds ± std).

future state is measured by navigating the paths connecting the two generated by a world model. Some of these are *circuitous gradient paths*: these paths do not go from state to action directly but instead include auto-regressive state predictions from the model itself. Such paths are perilous for long-horizon policy optimization; errors, noise, and gradient behaviors that may be good internally can rapidly degrade outside the model, reducing the possible advantages of backpropagating through transformers.

In particular, the following theorem shows that even if the gradient of a transformer HWM is bounded, the resulting policy gradient can still grow exponentially. The proof of this and the rest of our results can be found in Appendix A.

Theorem 3.1. *Let the gradient norm of h with respect to its inputs be bounded by L_a and L_s: $\|\frac{\partial h(\hat{s}_{1:t},a_{1:t})}{\partial a_k}\| \leq L_a$ and $\|\frac{\partial h(\hat{s}_{1:t},a_{1:t})}{\partial \hat{s}_i}\| \leq L_s$ for all $s_{1:t}, a_{1:t}, k, i$. Let r be the L_r-Lipschitz reward function from a Markov Decision Process $\mathcal{M}$, $\Pi_{\boldsymbol{\theta}}$ a parametric space of differentiable deterministic L_π-policies. Given $\pi_{\boldsymbol{\theta}} \in \Pi_{\boldsymbol{\theta}}$, the norm of the policy gradient $\nabla_{\boldsymbol{\theta}} J^h(\boldsymbol{\theta}; H)$ of $\pi_{\boldsymbol{\theta}}$ under a History-conditioned World Model h grows asymptotically as a function of the horizon H as:*

$$\begin{aligned}\|\nabla_\theta J^h(\boldsymbol{\theta}; H)\| &= O(HL_r + H^2 L_\pi + H^2 L_a + H^2 L_s^H) \\ &= O(L_s^H) \ .\end{aligned}$$

The exponential dependency, related to exploding gradients in RNNs, is apparent in Figure 1b, where the longest path from any given state to action is no shorter than the longest path through a Markovian model. The bound is shown to be tight in Appendix A.6. In the one-dimensional case, when all gradients are positive scalars, the above bound can be turned into a lower bound.

To illustrate the negative effects of circuitous gradient paths in practice, we conduct experiments in a double-pendulum environment where the agent must move the double-pendulum to a target position. We describe and visualize the task more precisely in Section 5. The results in Figure 2 show that BPO does not lead to successful policies with both Markovian models and HWMs in this notoriously chaotic environment. This result demonstrates that state-conditioned transformers do not provide any benefits to policy gradients due to circuitous gradient paths, unlike in the traditional supervised learning setting where they were initially introduced. In the rest of the paper, we will see how using transformers as a different type of world model allows us not only to overcome the limitations of Markovian models and HWMs, but also to outperform policy optimization performed by differentiating through the dynamics of the simulator, as well as competitive model-free approaches.

4. Policy Gradient Computation with Action-conditioned World Models

Policy gradients obtained from state-based world models, whether Markovian models or HWMs, are necessarily computed by backpropagating through sequences of states generated by autoregressively unrolling the models. Each state prediction uses as input previously generated states; consequently, the longest path between any action and reward traverses through circuitous gradient paths in the state space, *regardless of the underlying world model architecture*. Through careful examination of Figure 1 and Theorem 3.1, it quickly becomes apparent that the state dependency in h or f is at the root cause of such paths.

Inspired by this observation, we propose to remove all circuitous gradient paths in the simplest and most direct way: conditioning the world model on actions only. We thus consider a model which predicts a future state given an initial state and a sequence of actions. The resulting model, which we call an *Action-conditioned World Model* (AWM), is of the form $\hat{s}_{t+1} = g(s_1, a_1, a_2, \ldots, a_t)$. This structure for the world model constrains gradients to pass directly from rewards to actions through the network parameterizing the AWM. Note that while actions are theoretically sufficient to predict future states in a deterministic MDPs, we show in Appendix C.3 that they can also be sufficient for stochastic environments.

Given an AWM, a policy is learned by computing and employing (as in Algorithm 1) the gradient of the objective:

$$\begin{aligned} \text{maximize} \quad & J^g(\boldsymbol{\theta}; H) := \sum_{t=1}^{H} r(\hat{s}_t), \\ \text{subject to} \quad & \hat{s}_t = g(s_1, a_1, a_2, \ldots, a_{t-1}), \\ & a_t = \pi_{\boldsymbol{\theta}}(\text{sg}[\hat{s}_t]),\ t = 1, \ldots, H-1 \\ \text{given} \quad & s_1. \end{aligned} \tag{3}$$

An AWM can be trained in a similar manner to a Markovian model or an HWM, by using the loss $\ell^g(\tau;\boldsymbol{\psi}) = \sum_{t=1}^{H-1} \|s^\tau_{t+1} - g_{\boldsymbol{\psi}}(s_1, a^\tau_{1:t})\|^2$, computed likewise using states from an episode τ.

We will now show that, despite the seemingly counter-intuitive removal of state inputs, AWMs naturally connect the theory of policy gradients and sequence modeling with neural networks and provide a framework in which transformers can give high-quality policy gradients.

4.1. Markovian Models are RNN Action-conditioned World Models

The intuition that policy gradients computed through an unrolled Markovian model prescribed by equation 1 resembles the process of backpropagating through a recurrent neural network has been pointed out in multiple prior works (Heess et al., 2015; Metz et al., 2021; Parmas & Seno, 2022; Suh et al., 2022; Zhang et al., 2023). Yet, a formal account of this relationship has never been shown. Via the framework provided by AWMs, the next proposition shows how recurrent neural networks and policy gradients are connected, offering a first glimpse at the connection between neural network architectures, AWMs and policy gradient computation in MDPs.

Proposition 4.1. *Let f-RNN be a recurrent network with its recurrent cell being the dynamics f of the MDP $\mathcal{M}$, and $g_{f\text{-RNN}}$ denote an AWM instantiated with f-RNN. Then,*

$$\nabla_{\boldsymbol{\theta}} J^{g_{f\text{-RNN}}}(\boldsymbol{\theta}; H) = \nabla_{\boldsymbol{\theta}} J^{f}(\boldsymbol{\theta}; H).$$

The above proposition tells us that the policy gradient computed through a Markovian model is, in fact, equivalent to the one computed through an AWM when instantiating g as a recurrent neural network with a specific recurrent cell. Crucially, this not only provides grounding for gradient estimation with AWMs but also solidifies a fundamental fact that will be analyzed in-depth in this section: policy gradient computation by differentiating through unrolled Markovian models can be understood to be fundamentally ill-behaved due to its correspondence to an RNN structure.

4.2. Theoretical Properties of Action-conditioned World Models

Through the concept of Action-conditioned World Models, we have established a direct connection between deep sequence models and policy gradient computation. We will now exploit this connection to characterize the asymptotic behavior of the policy gradient depending on the underlying neural network architecture employed as an AWM. To do so, we leverage the following theorem.

Theorem 4.1. *Let r be the L_r-Lipschitz reward function from a Markov Decision Process $\mathcal{M}$, $\Pi_{\boldsymbol{\theta}}$ a parametric space of differentiable deterministic L_π-policies. Given $\pi_{\boldsymbol{\theta}} \in \Pi_{\boldsymbol{\theta}}$, the norm of the policy gradient $\nabla_{\boldsymbol{\theta}} J^g(\boldsymbol{\theta}; H)$ of $\pi_{\boldsymbol{\theta}}$ under an Action-conditioned World Model g as a function of the horizon H can upper bounded as:*

$$\|\nabla_{\boldsymbol{\theta}} J^g(\boldsymbol{\theta}; H)\| \leq L_r L_\pi \sum_{t=1}^{H} \sum_{k=1}^{t-1} \left\| \frac{\partial g(s_1, a_{1:t-1})}{\partial a_k} \right\|.$$

This theorem holds for any differentiable AWM g, and establishes a worst-case relationship between the Jacobian of the AWM w.r.t. its inputs (i.e., actions) and the policy gradient. Notably, the result implies that the policy gradient does not explode if the Jacobian of the AWM does not explode. In contrast, this may not be true even for well-behaved HWM.

We will now leverage the generality of the previous result to characterize the asymptotic behavior of the policy gradient computed using different neural network architectures for an AWM. First, let us consider an AWM instantiated with a simple recurrent neural network:

$$(g_{\text{RNN}}) \quad x_{t+1} = \sigma(W_x x_t) + W_a a_t + b; \quad \hat{s}_{t+1} = W_o x_{t+1}, \tag{4}$$

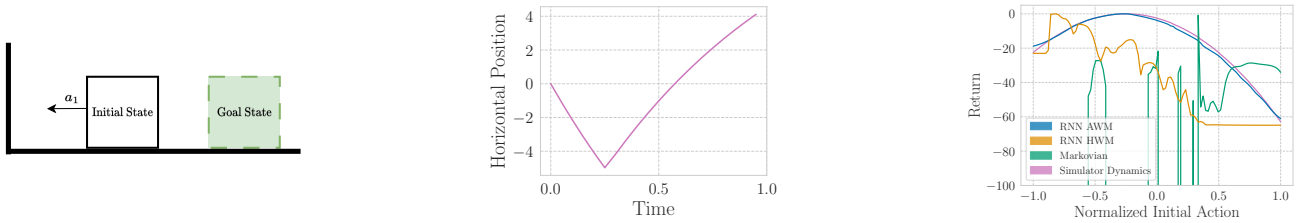
where σ is an activation function with gradient norm bounded by $\|diag(\sigma'(x))\| \leq \frac{1}{\beta}$ for some constant β. Then, the following result holds.

Corollary 4.1. *Let g_{RNN} be an Action-conditioned World Model instantiated with a recurrent neural network as in Equation 4 and $\eta = \|W_x^T\|\frac{1}{\beta}$. The asymptotic behavior of the norm of the policy gradient $\nabla_{\boldsymbol{\theta}} J^{g_{\text{RNN}}}(\boldsymbol{\theta}; H)$ as a function of the horizon H can be described as:*

$$\|\nabla_{\boldsymbol{\theta}} J^{g_{\text{RNN}}}(\boldsymbol{\theta}; H)\| = O\left(\eta^H\right).$$

Corollary 4.1 shows that, in the worst case, the policy gradient computed with an AWM instantiated with an RNN backbone can explode exponentially fast with respect to the problem horizon if the spectral radius of W_x exceeds β. Just like Theorem 3.1, the above bound can be written as a lower bound when all gradients are positive scalars, therefore it is tight. This result will allow us to compare the gradient provided by an RNN AWM with the one provided by a transformer AWM, but it is also connected with gradient computation with Markovian models: as seen in Proposition 4.1, using a Markovian model to compute the policy gradient, either given or learned, implies an RNN-structure for the AWM. Theorem 4.1 is a consequence of foundational theory of RNNs (Bengio et al., 1994; Pascanu et al., 2013), thus formalizing the intuition that the recurrent nature of policy gradients through Markovian models make both exploiting differentiable simulators f and learned models $\hat{f}_{\boldsymbol{\psi}}$ difficult (Metz et al., 2021; Heess et al., 2015).

Let us now apply Theorem 4.1 to a layer of self-attention AWM (Vaswani et al., 2017). Given $Q \in \mathbb{R}^{1\times d_z}$, $K \in$



(a) One-bounce environment overview.

(b) Example trajectory with non-differentiable point at $t \approx 0.25$.

(c) Final return with respect to the initial action for different models.

Figure 3. **AWMs ignore non-differentiable points in the state space.** (a) After the block is pushed with some initial action, it bounces off the wall, instantaneously reversing its velocity. (b) Visualization of the point of non-differentiability in the state space. (c) Learning a Markovian model or a HWM causes catastrophic compounding errors, but an AWM can still accurately model the final reward when varying the initial action. Learned dynamics are trained offline on a dataset collected using random actions.

$\mathbb{R}^{n\times d_z}$, $V \in \mathbb{R}^{n\times d_o}$, we define $\text{Attention}(Q,K,V) := \text{softmax}(QK^T)V$. Using a set of weight matrices, a self-attention AWM predicts the next state as follows[4]:

$$(g_{\text{ATT}}) \quad \hat{s}_{t+1} = \text{Attention}(a_{1:t}W_q^T, a_{1:t}W_k^T, a_{1:t}W_v^T) \ . \tag{5}$$

We can now characterize the behavior of the policy gradient computed with a self-attention AWM.

Corollary 4.2. *Let g_{ATT} be an attention-based Action-conditioned World Model instantiated with self-attention as in equation 5. The asymptotic behavior of the norm of the policy gradient $\nabla_{\boldsymbol{\theta}} J^{g_{\text{ATT}}}(\boldsymbol{\theta}; H)$ as a function of the horizon H can be described as:*

$$\|\nabla_{\boldsymbol{\theta}} J^{g_{\text{ATT}}}(\boldsymbol{\theta}; H)\| = O\left(H^3\right).$$

This result shows that the norm of the policy gradient computed through a self-attention AWM has a worst-case polynomial dependency on the horizon instead of an exponential one shown in Theorem 3.1 and Corollary 4.1. Since the output of a transformer model only depends on the sequence length through the attention mechanism, this characterization captures the gradient dynamics of the transformer model itself. It shows that, by eliminating all circuitous gradient paths, AWMs provide a framework for transformers to translate their favorable gradient properties towards better policy gradients, unlike state-conditioned models.

5. Experiments

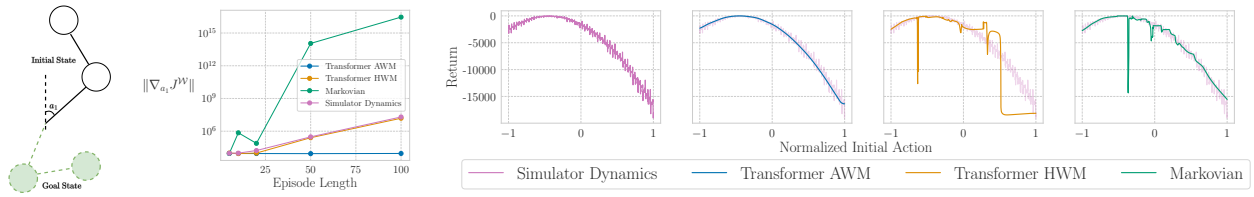
Backpropagation through Action-conditioned World Models produces policy gradients with no circuitous gradient paths. Consequently, the longest gradient path from an action to any reward strictly depends on the AWM internals, while the path through state-based world models scales with the horizon, regardless of the model architecture. This section empirically explores how this phenomenon benefits the policy optimization process (see Appendix B.1 for hyperparameters of all experiments).

5.1. Empirical Properties of Action-conditioned World Models

By removing the need to explicitly model the state dynamics auto-regressively, AWMs can internally build a more appropriate world model for policy optimization. Using environments inspired by real-world physical systems, we highlight two particular properties of AWMs: skipping points of complex state-space dynamics and smoothing out ill-behaved policy optimization landscapes.

AWMs overcome non-differentiable dynamics Unlike state-based world models, AWMs directly map actions to future states without explicitly predicting intermediate states. This mapping allows AWMs to skip points of potentially complex dynamics in the state space, which are not always necessary for policy optimization. For example, consider points of non-differentiability (e.g., contact points) in real-world systems (e.g., physical systems). These points make directly using the simulator dynamics for BPO impossible and can cause catastrophic compounding errors for state-conditioned world models. However, these points often exist even if the dependency of rewards on actions is smooth and well-behaved everywhere. In such cases, AWMs can be used to side-step the non-differentiability of the state dynamics. To illustrate this point, we examine a *one-bounce* environment, shown in Figure 3. In this task, an agent must push a block towards a wall with some initial action, such that the block bounces off the wall and ends up at some predetermined goal state after H steps. A single terminal reward is given at $t = H$ measuring the distance of the block to the goal state. Even though the state trajectory is non-differentiable due to the wall bounce, the final state of the

[4]Without loss of generality, the dependency on the initial state is not included in the self-attention layer.



(a) Double-pendulum. (b) Gradient norms. (c) Final returns with respect to the initial action for different models when $H = 100$.

Figure 4. **Transformer AWMs smooths out chaotic dynamics.** (a) A double-pendulum environment where an initial position must be chosen in order to achieve some pre-determined goal state after H steps. Different transition models are learned on a data set of random trajectories. (b) The mean gradient norm of the final state with respect to the initial action for each model is computed over 50 different random actions for different horizons. (c) Final return according to different models with respect to different initial actions for $H = 100$.

block is actually well-behaved with respect to initial actions (Figure 3b and Figure 3c). Consequently, we show in Figure 3c that AWMs can accurately predict the final reward while an unrolled Markovian or HWM cannot. Notably, the resulting AWM thus allows to easily use backpropagation-based policy optimization even in the presence of an underlying non-differentiable environment dynamics.

AWMs overcome chaotic dynamics In many real-world systems (e.g., robots), the chaotic dynamics of an environment can create a remarkably complex mapping from actions to rewards over time or a highly non-smooth return landscape (Rahn et al., 2023). In such systems, even when a differentiable simulator is available, it gives ill-behaved gradients and generates an optimization landscape that is hard to navigate (Parmas et al., 2018; Suh et al., 2022). We now show that transformer AWMs are able to take full advantage of the inductive biases of the underlying neural networks, naturally generating policy optimization landscapes that are easy to navigate. This happens regardless of the complexity of the real dynamics while preserving the global problem structure and allowing good policies to be found more easily. To demonstrate this, we use a prototypical chaotic system, a double-pendulum task depicted in Figure 4a. In this environment, the agent's initial action determines the initial angular position of the inner pendulum. Afterwards, the system is rolled out for H steps. The objective is to get the pendulums to be in some predefined goal state after H steps. A terminal reward is given measuring the distance between the final observed state and the desired goal state. We illustrate in Figure 4b and Figure 4c the chaotic nature of the task, showing respectively that the norm of the true gradient of the return with respect to the initial action grows exponentially with the episode length H, and that the true return landscape is extremely difficult to navigate for long horizons. Figure 4b shows that transformers provide stable gradients when used as AWMs, while the gradient provided by HWMs explodes similarly to the true gradients. AWMs generate a smooth and accurate approximation of the return landscape (Figure 4c), which gradient ascent can easily navigate, leading to better policies when used for backpropagation-based policy optimization compared to all the other world models (Figure 2).

5.2. Benchmarking Credit Assignment with AWMs

We now show that transformers employed as AWMs can give better policies in environments which require long-horizon planning. We focus on eight tasks from the Myriad testbed (Howe et al., 2022). Myriad provides continuous MDPs inspired by real-world problems (Lenhart & Workman, 2007) with configurable horizons. In Myriad, the agent requires better temporal credit assignment capabilities to solve problems with longer episode lengths, in contrast to other RL benchmarks (e.g., focused on simple robotics locomotion), which typically feature a short effective horizon (Laidlaw et al., 2023; Ni et al., 2023) regardless of the episode length. Additional information about Myriad and its experiments are provided in Appendix B.4. All aggregate results use the IQM (Agarwal et al., 2021).

Figure 5 shows the final performance of various world models used in BPO with increasing episode lengths. The transformer AWM is the only model to perform reliably well for increasingly long horizons, while other methods struggle when the horizon exceeds 100 steps. Indeed, when used as AWMs, transformer world models give better policy gradients than standard Markovian models and RNN world models. However, when transformer world models are conditioned on states, their long-term policy gradients are no better than the gradients induced by Markovian models as measured by performance in Figure 5.

We also compare our method with other relevant baselines in Figure 6, namely model-free Soft Actor-Critic (SAC) (Haarnoja et al., 2018) and the recently popularized Online Decision Transformer (Online-DT) (Zheng et al., 2022). In addition to BPO with transformer AWMs scaling better with the problem horizon, BPO is also significantly more sample efficient than the two baselines.

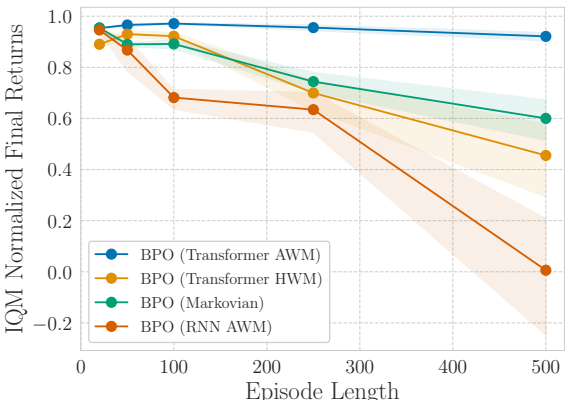


Figure 5. **Policy optimization with transformer AWMs gives better policies for long horizons.** Final performance of BPO with different world models on Myriad (10 seeds ± 95% C.I.).

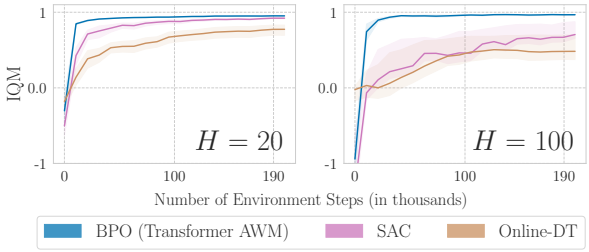


Figure 6. **BPO with transformer AWMs is more sample efficient and better performing than baselines.** Learning curves of BPO through a transformer AWM, a SAC agent, and an Online-DT agent on 20 and 100 length horizons (10 seeds ± 95% C.I.).

6. Related Work

Backpropagation-based Policy Optimization The concept of learning a policy by differentiating through the environment dynamics has been a foundational aspect of early RL methods (Werbos, 1974; Schmidhuber, 1990). Estimators for such a policy gradient have been referred to with different names, including pathwise derivatives (Clavera et al., 2020; Hafner et al., 2021; 2023), model-based reparameterized policy gradients (Ruiz et al., 2016; Zhang et al., 2023), and value gradients (Fairbank, 2014; Heess et al., 2015; Amos et al., 2021). Different solutions have been explored to address long horizon policy optimization with these methods, such as gradient mixing (Parmas, 2020; Parmas et al., 2023; Suh et al., 2022), bootstrapping with a critic (Heess et al., 2015; Xu et al., 2022), and gradient normalization (Zhang et al., 2023). Conversely, AWMs connect this literature to the one on sequence modeling with neural networks, with a focus on gradient propagation (Bengio et al., 1994; Hochreiter, 1998; Pascanu et al., 2013) and on architectures that allow for improved credit assignment (Hochreiter & Schmidhuber, 1997; Vaswani et al., 2017; Kerg et al., 2020). Finally, our work can be seen as an instance of gradient-aware model-based RL (D'Oro et al., 2020; D'Oro & Jaskowski, 2020; Abachi et al., 2020), in which we modify the architecture of a model of the dynamics to obtain better policy gradients (Ma et al., 2021).

Multi-step state prediction with actions Prior works used multi-step models related to AWMs, analyzing their role as partial models (Rezende et al., 2020), in the context of tree search (Schrittwieser et al., 2019), or training them with a multi-step latent prediction loss (Gregor et al., 2019; Schwarzer et al., 2020), without however ever analyzing their gradient properties. Notably, models used in these prior works are rarely shown to work beyond a few dozen steps of unrolling or imagination, while AWMs demonstrably scale favorably for horizons well beyond established limitations.

Sequence models in RL In partially observable MDPs, sequence models have been extensively used as history encoders to maximize returns with or without world models (Hausknecht & Stone, 2015; Ni et al., 2022; Hafner et al., 2023; Samsami et al., 2024). Other works have recently emerged treating MDPs as sequence modeling (Chen et al., 2021; Zheng et al., 2022; Janner et al., 2021), in imitation learning or offline RL problem settings, based on return-conditioned models. Sequence models have also been used to reshape rewards in sparse reward settings (Hung et al., 2019; Arjona-Medina et al., 2019; Liu et al., 2019). In contrast to all of these, our framework uses an *action-only* conditioned model to directly improve long-term policy gradients and naturally assigns credit by backpropagation as opposed to reward-shaping heuristics. Appendix D contains a more detailed discussion of how all these methods differ from ours.

7. Conclusions and Limitations

We presented a model-based policy optimization framework based on backpropagation through Action-conditioned World Models: models of the dynamics of the environment which only take an initial state and a sequence of actions as inputs. We highlighted that, in contrast to the more common world models that compute the gradient by autoregressively unrolling their state predictions, AWMs do not suffer from the existence of circuitous gradient paths where the gradient can flow outside of the underlying network architecture. We showed that their structure allows AWMs to fully exploit transformers' favorable properties, taking advantage, both in theory and in practice, of their ability to perform long-term temporal credit assignment by leveraging short gradient paths.

Theoretically, we demonstrated how transformer AWMs do not suffer from ill-behaved gradients coming from circuitous gradient paths which harm the quality of the gra-

dients provided by more common Markovian models and HWMs. Empirically, we showed that BPO using AWMs can be successful even when BPO using the underlying simulator fails, due to how AWMs leverage transformers' inductive biases to obtain an easier-to-navigate policy optimization landscape. We demonstrated that policies trained with transformer AWMs outperform existing model-based and model-free approaches, including recently proposed algorithms based on transformers, in realistic domains requiring long-term planning.

While our theoretical analysis is developed within a deterministic setting, and our experiments are conducted in low-dimensional yet realistic domains, our framework establishes a new connection between gradient propagation techniques for sequence modeling and backpropagation-based policy gradients. Future work could focus on scaling our approach, making RL even more relevant to real-world applications where long-horizon capabilities are crucial.

Acknowledgements

The authors thank Nathan Rahn, Jürgen Schmidhuber and Paavo Parmas for insightful discussions and useful suggestions on the paper, the Mila community for creating a stimulating research environment, Digital Research Alliance of Canada and Nvidia for computational resources. This work was partially supported by CIFAR, Google, Facebook CIFAR AI, IVADO, and Gruppo Ermenegildo Zegna.

Impact Statement

Enhancing model-based RL with transformers increases agent autonomy, posing risks of unintended actions in complex environments. More powerful autonomous agents can also increase the potential for misuse in sensitive areas like finance and cybersecurity. Advances in this field can exacerbate challenges in controlling and ensuring accountability of these more capable AI agents.

References

Abachi, R., Ghavamzadeh, M., and massoud Farahmand, A. Policy-aware model learning for policy gradient methods. *ArXiv*, abs/2003.00030, 2020.

Agarwal, R., Schwarzer, M., Castro, P. S., Courville, A. C., and Bellemare, M. G. Deep reinforcement learning at the edge of the statistical precipice. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 29304–29320, 2021.

Amos, B., Stanton, S., Yarats, D., and Wilson, A. G. On the model-based stochastic value gradient for continuous reinforcement learning learning. In Jadbabaie, A., Lygeros, J., Pappas, G. J., A.;Parrilo, P., Recht, B., Tomlin, C. J., and Zeilinger, M. N. (eds.), *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, volume 144 of *Proceedings of Machine Learning Research*, pp. 6–20. PMLR, 07 – 08 June 2021.

Arjona-Medina, J. A., Gillhofer, M., Widrich, M., Unterthiner, T., Brandstetter, J., and Hochreiter, S. Rudder: Return decomposition for delayed rewards. *Advances in Neural Information Processing Systems*, 32, 2019.

Bengio, Y., Simard, P. Y., and Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Networks*, 5(2):157–166, 1994. doi: 10.1109/72.279181.

Chen, C., Wu, Y.-F., Yoon, J., and Ahn, S. Transdreamer: Reinforcement learning with transformer world models. *ArXiv preprint*, abs/2202.09481, 2022.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 15084–15097, 2021.

Clavera, I., Fu, Y., and Abbeel, P. Model-augmented actor-critic: Backpropagating through paths. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

D'Oro, P. and Jaskowski, W. How to learn a useful critic? model-based action-gradient-estimator policy optimization. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

D'Oro, P., Metelli, A. M., Tirinzoni, A., Papini, M., and Restelli, M. Gradient-aware model-based policy search. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 3801–3808. AAAI Press, 2020. doi: 10.1609/aaai.v34i04.5791.

Fairbank, M. *Value-gradient learning*. PhD thesis, City University London, 2014.

Ghugare, R., Bharadhwaj, H., Eysenbach, B., Levine, S., and Salakhutdinov, R. Simplifying model-based rl: Learning representations, latent-space models, and policies with one objective, 2023.

Gregor, K., Rezende, D. J., Besse, F., Wu, Y., Merzic, H., and van den Oord, A. Shaping belief states with generative environment models for RL. *CoRR*, abs/1906.09237, 2019.

Gu, A., Goel, K., and Ré, C. Efficiently modeling long sequences with structured state spaces. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *CoRR*, abs/1801.01290, 2018.

Hafner, D., Lillicrap, T. P., Norouzi, M., and Ba, J. Mastering atari with discrete world models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Hafner, D., Pasukonis, J., Ba, J., and Lillicrap, T. Mastering diverse domains through world models, 2023.

Hausknecht, M. and Stone, P. Deep recurrent q-learning for partially observable mdps. In *2015 aaai fall symposium series*, 2015.

Heess, N., Wayne, G., Silver, D., Lillicrap, T. P., Erez, T., and Tassa, Y. Learning continuous control policies by stochastic value gradients. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 2944–2952, 2015.

Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Computation*, 9(8):1735–1780, November 1997. doi: 10.1162/neco.1997.9.8.1735.

Howe, N., Dufort-Labbé, S., Rajkumar, N., and Bacon, P.-L. Myriad: a real-world testbed to bridge trajectory optimization and deep learning. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 29801–29815. Curran Associates, Inc., 2022.

Hung, C.-C., Lillicrap, T., Abramson, J., Wu, Y., Mirza, M., Carnevale, F., Ahuja, A., and Wayne, G. Optimizing agent behavior over long time scales by transporting value. *Nature communications*, 10(1):5223, 2019.

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

Kerg, G., Kanuparthi, B., Goyal, A., Goyette, K., Bengio, Y., and Lajoie, G. Untangling tradeoffs between recurrence and self-attention in artificial neural networks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

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

Laidlaw, C., Russell, S., and Dragan, A. Bridging RL theory and practice with the effective horizon. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Lenhart, S. and Workman, J. Optimal control applied to biological models. 01 2007.

Liu, Y., Luo, Y., Zhong, Y., Chen, X., Liu, Q., and Peng, J. Sequence modeling of temporal credit assignment for episodic reinforcement learning. *arXiv preprint arXiv:1905.13420*, 2019.

Lu, C., Schroecker, Y., Gu, A., Parisotto, E., Foerster, J., Singh, S., and Behbahani, F. Structured state space models for in-context reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

Ma, M., D'Oro, P., Bengio, Y., and Bacon, P.-L. Long-term credit assignment via model-based temporal shortcuts. In *Deep RL Workshop NeurIPS 2021*, 2021.

Metz, L., Freeman, C. D., Schoenholz, S. S., and Kachman, T. Gradients are not all you need. *arXiv preprint arXiv:2111.05803*, 2021.

Micheli, V., Alonso, E., and Fleuret, F. Transformers are sample efficient world models. *arXiv preprint arXiv:2209.00588*, 2022.

Miller, W., Sutton, R., and Werbos, P. *Neural Networks for Control*. A Bradford book. MIT Press, 1995. ISBN 9780262631617.

Ni, T., Eysenbach, B., and Salakhutdinov, R. Recurrent model-free RL can be a strong baseline for many pomdps. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16691–16723. PMLR, 2022.

Ni, T., Ma, M., Eysenbach, B., and Bacon, P.-L. When do transformers shine in rl? decoupling memory from credit assignment. *arXiv preprint arXiv:2307.03864*, 2023.

Ni, T., Eysenbach, B., Seyedsalehi, E., Ma, M., Gehring, C., Mahajan, A., and Bacon, P.-L. Bridging state and history representations: Understanding self-predictive rl. *arXiv preprint arXiv:2401.08898*, 2024.

Parisotto, E., Song, F., Rae, J., Pascanu, R., Gulcehre, C., Jayakumar, S., Jaderberg, M., Kaufman, R. L., Clark, A., Noury, S., et al. Stabilizing transformers for reinforcement learning. In *International conference on machine learning*, pp. 7487–7498. PMLR, 2020.

Parmas, P. *Total Stochastic Gradient Algorithms and Applications to Model-based Reinforcement Learning*. PhD thesis, Okinawa Institute of Science and Technology Graduate University, 2020-02-04 2020.

Parmas, P. and Seno, T. Proppo: a message passing framework for customizable and composable learning algorithms. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 29152–29165. Curran Associates, Inc., 2022.

Parmas, P., Rasmussen, C. E., Peters, J., and Doya, K. Pipps: Flexible model-based policy search robust to the curse of chaos. In *International Conference on Machine Learning*, pp. 4065–4074. PMLR, 2018.

Parmas, P., Seno, T., and Aoki, Y. Model-based reinforcement learning with scalable composite policy gradient estimators. In *International Conference on Machine Learning*, pp. 27346–27377. PMLR, 2023.

Pascanu, R., Mikolov, T., and Bengio, Y. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pp. 1310–1318. JMLR.org, 2013.

Pascanu, R., Gülçehre, Ç., Cho, K., and Bengio, Y. How to construct deep recurrent neural networks. In Bengio, Y. and LeCun, Y. (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Rahn, N., D'Oro, P., Wiltzer, H., Bacon, P.-L., and Bellemare, M. G. Policy optimization in a noisy neighborhood: On return landscapes in continuous control. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Rezende, D. J., Danihelka, I., Papamakarios, G., Ke, N. R., Jiang, R., Weber, T., Gregor, K., Merzic, H., Viola, F., Wang, J., Mitrovic, J., Besse, F., Antonoglou, I., and Buesing, L. Causally correct partial models for reinforcement learning. *CoRR*, abs/2002.02836, 2020.

Robine, J., Höftmann, M., Uelwer, T., and Harmeling, S. Transformer-based world models are happy with 100k interactions. *arXiv preprint arXiv:2303.07109*, 2023.

Ruiz, F. R., AUEB, T. R., Blei, D., et al. The generalized reparameterization gradient. *Advances in neural information processing systems*, 29, 2016.

Samsami, M. R., Zholus, A., Rajendran, J., and Chandar, S. Mastering memory tasks with world models. *arXiv preprint arXiv:2403.04253*, 2024.

Schmidhuber, J. Making the world differentiable: On using self-supervised fully recurrent neural networks for dynamic reinforcement learning and planning in non-stationary environments, 1990.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., Lillicrap, T. P., and Silver, D. Mastering atari, go, chess and shogi by planning with a learned model. *CoRR*, abs/1911.08265, 2019.

Schwarzer, M., Anand, A., Goel, R., Hjelm, R. D., Courville, A. C., and Bachman, P. Data-efficient reinforcement learning with momentum predictive representations. *CoRR*, abs/2007.05929, 2020.

Suh, H. J., Simchowitz, M., Zhang, K., and Tedrake, R. Do differentiable simulators give better policy gradients? In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 20668–20696. PMLR, 2022.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017.

Werbos, P. J. *Beyond regression: New tools for prediction and analysis in the behavioral sciences*. PhD thesis, Harvard University, Cambridge, MA, 1974.

Werbos, P. J. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

Williams, R. J. and Zipser, D. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280, 1989. doi: 10.1162/neco.1989.1.2.270.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Xu, J., Makoviychuk, V., Narang, Y. S., Ramos, F., Matusik, W., Garg, A., and Macklin, M. Accelerated policy learning with parallel differentiable simulation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Zhang, A., McAllister, R. T., Calandra, R., Gal, Y., and Levine, S. Learning invariant representations for reinforcement learning without reconstruction. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Zhang, S., Wu, Y., Che, T., Lin, Z., Memisevic, R., Salakhutdinov, R. R., and Bengio, Y. Architectural complexity measures of recurrent neural networks. *Advances in neural information processing systems*, 29, 2016.

Zhang, S., Liu, B., Wang, Z., and Zhao, T. Model-based reparameterization policy gradient methods: Theory and practical algorithms, 2023.

Zheng, Q., Zhang, A., and Grover, A. Online decision transformer. In *international conference on machine learning*, pp. 27042–27059. PMLR, 2022.

A. Proofs and Results for Sections 3 and 4

A.1. Proof of Theorem 3.1

Theorem 3.1. *Let the gradient norm of h with respect to its inputs be bounded by L_a and L_s: $\|\frac{\partial h(\hat{s}_{1:t},a_{1:t})}{\partial a_k}\| \leq L_a$ and $\|\frac{\partial h(\hat{s}_{1:t},a_{1:t})}{\partial \hat{s}_i}\| \leq L_s$ for all $s_{1:t}, a_{1:t}, k, i$. Let r be the L_r-Lipschitz reward function from a Markov Decision Process $\mathcal{M}$, $\Pi_{\boldsymbol{\theta}}$ a parametric space of differentiable deterministic L_π-policies. Given $\pi_{\boldsymbol{\theta}} \in \Pi_{\boldsymbol{\theta}}$, the norm of the policy gradient $\nabla_{\boldsymbol{\theta}} J^h(\boldsymbol{\theta}; H)$ of $\pi_{\boldsymbol{\theta}}$ under a History-conditioned World Model h grows asymptotically as a function of the horizon H as:*

$$\begin{aligned}\|\nabla_\theta J^h(\boldsymbol{\theta}; H)\| &= O(HL_r + H^2 L_\pi + H^2 L_a + H^2 L_s^H) \\ &= O(L_s^H) \ .\end{aligned}$$

Proof. The policy gradient computed through equation 2 can be expanded using the chain rule into the following expression:

$$\nabla_{\boldsymbol{\theta}} J^h(\boldsymbol{\theta}; H) = \sum_{t=1}^{H} \frac{\partial r(\hat{s}_t)}{\partial \hat{s}_t} \sum_{k=1}^{t-1} \frac{dh(\hat{s}_{1:t-1}, a_{1:t-1})}{da_k} \frac{\partial \pi_{\boldsymbol{\theta}}(\hat{s}_k)}{\partial \boldsymbol{\theta}}$$

$$\frac{dh(\hat{s}_{1:t-1}, a_{1:t-1})}{da_k} = \frac{\partial h(\hat{s}_{1:t-1}, a_{1:t-1})}{\partial a_k} + \sum_{i=k+1}^{t-1} \frac{\partial h(\hat{s}_{1:t-1}, a_{1:t-1})}{\partial \hat{s}_i} \frac{dh(\hat{s}_{1:i-1}, a_{1:i-1})}{da_k}$$

We begin by showing how the norm of the policy gradient depends on $\frac{dh(\hat{s}_{1:t},a_{1:t})}{da_k}$.

$$\begin{aligned}\|\nabla_{\boldsymbol{\theta}} J^h(\boldsymbol{\theta}; H)\| &= \left\| \sum_{t=1}^{H} \frac{\partial r(\hat{s}_t)}{\partial \hat{s}_t} \sum_{k=1}^{t-1} \frac{dh(\hat{s}_{1:t-1}, a_{1:t-1})}{da_k} \frac{\partial \pi_{\boldsymbol{\theta}}(\hat{s}_k)}{\partial \boldsymbol{\theta}} \right\| \\ \text{By Lipschitz assumption} \quad &\leq \left\| \sum_{t=1}^{H} L_r \sum_{k=1}^{t-1} \frac{dh(\hat{s}_{1:t-1}, a_{1:t-1})}{da_k} L_\pi \right\| \\ \text{Triangle inequality} \quad &\leq \sum_{t=1}^{H} L_r \sum_{k=1}^{t-1} \left\| \frac{dh(\hat{s}_{1:t-1}, a_{1:t-1})}{da_k} \right\| L_\pi \ .\end{aligned}$$

Now, we show how $\|\frac{dh(\hat{s}_{1:t-1},a_{1:t-1})}{da_k}\|$ grows asymptotically with t and k.

$$\begin{aligned}\left\| \frac{dh(\hat{s}_{1:t-1}, a_{1:t-1})}{da_k} \right\| &= \left\| \frac{\partial h(\hat{s}_{1:t-1}, a_{1:t-1})}{\partial a_k} + \sum_{i=k+1}^{t-1} \frac{\partial \hat{s}_t}{\partial \hat{s}_i} \frac{dh(\hat{s}_{1:i-1}, a_{1:i-1})}{da_k} \right\| \\ \text{Triangle inequality} \quad &\leq \left\| \frac{\partial h(\hat{s}_{1:t-1}, a_{1:t-1})}{\partial a_k} \right\| + \sum_{i=k+1}^{t-1} \left\| \frac{\partial h(\hat{s}_{1:t-1}, a_{1:t})}{\partial \hat{s}_i} \frac{dh(\hat{s}_{1:i-1}, a_{1:i-1})}{da_k} \right\| \\ \text{Cauchy-Schwarz inequality} \quad &\leq \left\| \frac{\partial h(\hat{s}_{1:t-1}, a_{1:t-1})}{\partial a_k} \right\| + \sum_{i=k+1}^{t-1} \left\| \frac{\partial h(\hat{s}_{1:t-1}, a_{1:t-1})}{\partial \hat{s}_i} \right\| \left\| \frac{dh(\hat{s}_{1:i-1}, a_{1:i-1})}{da_k} \right\| \\ \text{By assumption} \leq L_a + \sum_{i=k+1}^{t-1} L_s \left\| \frac{dh(\hat{s}_{1:i-1}, a_{1:i-1})}{da_k} \right\| \ .\end{aligned}$$

In the recursive expansion of the previous line, notice the linear dependency w.r.t. L_a, and exponential dependency w.r.t. L_s due to the compounding products. Therefore, $\|\frac{dh(\hat{s}_{1:t},a_{1:t})}{da_k}\| = O(L_a + L_s^{t-k})$, and we may complete the proof.

$$\begin{aligned}\|\nabla_{\boldsymbol{\theta}} J^h(\boldsymbol{\theta}; H)\| &\leq \sum_{t=1}^{H} L_r \sum_{k=1}^{t-1} \left\| \frac{dh(\hat{s}_{1:t-1}, a_{1:t-1})}{da_k} \right\| L_\pi \\ &= O\left(HL_r + H^2 L_\pi + H^2 \left\| \frac{dh(\hat{s}_{1:H-1}, a_{1:H-1})}{da_1} \right\| \right) \\ &= O(HL_r + H^2 L_\pi + H^2 L_a + H^2 L_s^H)\end{aligned}$$

□

A.2. Proof of Proposition 4.1

Proposition 4.1. *Let f-RNN be a recurrent network with its recurrent cell being the dynamics f of the MDP $\mathcal{M}$, and $g_{f\text{-RNN}}$ denote an AWM instantiated with f-RNN. Then,*

$$\nabla_{\boldsymbol{\theta}} J^{g_{f\text{-RNN}}}(\boldsymbol{\theta}; H) = \nabla_{\boldsymbol{\theta}} J^{f}(\boldsymbol{\theta}; H).$$

Proof. We begin by providing a formal definition of the recurrent network with its recurrent cell being the dynamics f of the MDP as:

$$\begin{aligned} g_{f\text{-RNN}}(s_1, a_{1:t}) &:= f(s_t, a_t) \\ s_t &:= f(s_{t-1}, a_{t-1}) = f(g_{f\text{-RNN}}(s_1, a_{1:t-1}), a_t) \end{aligned}$$

Begin by developing the LHS of the equation.

$$\nabla_{\boldsymbol{\theta}}^{g_{f\text{-RNN}}} J(\boldsymbol{\theta}; H) = \sum_{t=1}^{H} \frac{\partial r(s_t)}{\partial s_t} \sum_{k=1}^{t-1} \frac{\partial s_t}{\partial a_k} \frac{\partial \pi_{\boldsymbol{\theta}}(s_k)}{\partial \boldsymbol{\theta}}$$

where $s_t = f(s_{t-1}, a_{t-1})$ by construction (RNN assumption). Therefore, apply the chain rule $\frac{\partial s_t}{\partial a_k}$:

$$\nabla_{\boldsymbol{\theta}}^{g_{f\text{-RNN}}} J(\boldsymbol{\theta}; H) = \sum_{t=1}^{H} \frac{\partial r(s_t)}{\partial s_t} \sum_{k=1}^{t-1} \frac{\partial s_{k+1}}{\partial a_k} \left(\prod_{i=k+1}^{t-1} \frac{\partial s_{i+1}}{\partial s_i} \right) \frac{\partial \pi_{\boldsymbol{\theta}}(s_k)}{\partial \boldsymbol{\theta}}$$

Now, for the RHS of the equation. The policy gradient through equations 1 can be written as:

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} J^{f}(\boldsymbol{\theta}; H) &= \sum_{t=1}^{H} \frac{\partial r(s_t)}{\partial s_t} \sum_{k=1}^{t-1} \frac{\partial s_{k+1}}{\partial a_k} \frac{\partial \pi_{\boldsymbol{\theta}}(s_k)}{\partial \boldsymbol{\theta}} \left(\prod_{i=k+1}^{t-1} \frac{\partial s_{i+1}}{\partial s_i} \right). \\ \nabla_{\boldsymbol{\theta}}^{g_{f\text{-RNN}}} J(\boldsymbol{\theta}; H) &= \nabla_{\boldsymbol{\theta}} J^{f}(\boldsymbol{\theta}; H). \end{aligned}$$

□

A.3. Proof of Theorem 4.1

Theorem 4.1. *Let r be the L_r-Lipschitz reward function from a Markov Decision Process $\mathcal{M}$, $\Pi_{\boldsymbol{\theta}}$ a parametric space of differentiable deterministic L_π-policies. Given $\pi_{\boldsymbol{\theta}} \in \Pi_{\boldsymbol{\theta}}$, the norm of the policy gradient $\nabla_{\boldsymbol{\theta}} J^{g}(\boldsymbol{\theta}; H)$ of $\pi_{\boldsymbol{\theta}}$ under an Action-conditioned World Model g as a function of the horizon H can upper bounded as:*

$$\|\nabla_{\boldsymbol{\theta}} J^{g}(\boldsymbol{\theta}; H)\| \leq L_r L_\pi \sum_{t=1}^{H} \sum_{k=1}^{t-1} \left\| \frac{\partial g(s_1, a_{1:t-1})}{\partial a_k} \right\|.$$

Proof. With a slight abuse of notation, we use g_t to replace $g(s_1, a_{1:t-1})$.

$$\nabla_{\boldsymbol{\theta}} J^{g}(\boldsymbol{\theta}; H) = \left\| \sum_{t=1}^{H} \frac{\partial r(\hat{s}_t)}{\partial \hat{s}_t} \sum_{k=1}^{t-1} \frac{\partial g_t}{\partial a_k} \frac{\partial \pi(\hat{s}_k)}{\partial \boldsymbol{\theta}} \right\| \leq L_r L_\pi \left\| \sum_{t=1}^{H} \sum_{k=1}^{t-1} \frac{\partial g_t}{\partial a_k} \right\| \leq L_r L_\pi \sum_{t=1}^{H} \sum_{k=1}^{t-1} \left\| \frac{\partial g_t}{\partial a_k} \right\|.$$

□

A.4. Proof of Corollary 4.1

Corollary 4.1. *Let g_{RNN} be an Action-conditioned World Model instantiated with a recurrent neural network as in Equation 4 and $\eta = \|W_x^T\| \frac{1}{\beta}$. The asymptotic behavior of the norm of the policy gradient $\nabla_{\boldsymbol{\theta}} J^{g_{RNN}}(\boldsymbol{\theta}; H)$ as a function of the horizon H can be described as:*

$$\|\nabla_{\boldsymbol{\theta}} J^{g_{RNN}}(\boldsymbol{\theta}; H)\| = O\left(\eta^{H}\right).$$

Proof. We begin by defining the transition function of the recurrent network as $x_t = \sigma(W_x x_{t-1}) + W_a a_t + b$, where σ is some activation function with gradient norm bounded by $\|diag(\sigma'(x))\| \leq \frac{1}{\beta}$. Further, consider a linear output cell $s_{t+1} = W_o x_t$. Now we begin by showing that $\left\|\frac{\partial x_t}{\partial a_k}\right\| \leq ||W_a^T||(||W_x^T||\frac{1}{\beta})^{t-k}$:

$$
\begin{aligned}
\frac{\partial x_t}{\partial a_k} &= \frac{\partial x_k}{\partial a_k} \prod_{i=k}^{t-1} \frac{\partial x_{i+1}}{\partial x_i} \\
\left\|\frac{\partial x_t}{\partial a_k}\right\| &= \left\| W_a^T \prod_{i=k}^{t-1} \frac{\partial x_{i+1}}{\partial x_i} \right\| \\
\text{Cauchy-Schwarz Inequality} \quad &\leq ||W_a^T|| \prod_{i=k}^{t-1} \left\|\frac{\partial x_{i+1}}{\partial x_i}\right\| = ||W_a^T|| \prod_{i=k}^{t-1} \left\|W_x^T diag(\sigma'(x_i))\right\| \\
\text{Cauchy-Schwarz Inequality} \quad &\leq ||W_a^T|| \prod_{i=k}^{t-1} \left\|W_x^T\right\| \|diag(\sigma'(x_i))\| \\
\text{By assumption} \quad &\leq ||W_a^T|| \prod_{i=k}^{t-1} \left\|W_x^T\right\| \frac{1}{\beta} = ||W_a^T|| \left(||W_x^T||\frac{1}{\beta}\right)^{t-k} \quad .
\end{aligned}
$$

We develop the expression $\frac{\partial \hat{s}_t}{\partial a_k} = \frac{\partial \hat{s}_t}{\partial x_{t-1}} \frac{\partial x_{t-1}}{\partial a_k}$, applying the norm and plugging in the above result we obtain:

$$
\left\|\frac{\partial \hat{s}_t}{\partial a_k}\right\| \leq \|W_o\| \left\|W_a^T\right\| \left(||W_x^T||\frac{1}{\beta}\right)^{t-k-1}
$$

We then use the bound found in Theorem 1:

$$
\begin{aligned}
\|\nabla_{\boldsymbol{\theta}} J^g(\boldsymbol{\theta}; H)\| &\leq L_r L_\pi \sum_{t=1}^{H} \sum_{k=1}^{t-1} \left\|\frac{\partial g_t}{\partial a_k}\right\| = L_r L_\pi \sum_{t=1}^{H} \sum_{k=1}^{t-1} \left\|\frac{\partial \hat{s}_t}{\partial a_k}\right\| \\
&\leq L_r L_\pi \sum_{t=1}^{H} \sum_{k=1}^{t-1} \|W_o\| \left\|W_a^T\right\| \left(||W_x^T||\frac{1}{\beta}\right)^{t-k-1} = L_r L_\pi \sum_{t=1}^{H} \sum_{k=1}^{t-1} \|W_o\| \left\|W_a^T\right\| \eta^{t-k-1} \quad ,
\end{aligned}
$$

where $\eta = ||W_x^T||\frac{1}{\beta}$.

$$
\begin{aligned}
\|\nabla_{\boldsymbol{\theta}}^g J(\boldsymbol{\theta}; H)\| &\leq L_r L_\pi \sum_{t=1}^{H} \sum_{k=1}^{t-1} \|W_o\| \left\|W_a^T\right\| \eta^{t-k-1} \\
&= O(H^2 ||W_o|| \left\|W_a^T\right\| + H^2 \eta^H) \\
&= O(\eta^H) \quad .
\end{aligned}
$$

□

A.5. Proof of Corollary 4.2

Corollary 4.2. *Let g_{ATT} be an attention-based Action-conditioned World Model instantiated with self-attention as in equation 5. The asymptotic behavior of the norm of the policy gradient $\nabla_{\boldsymbol{\theta}} J^{g_{\text{ATT}}}(\boldsymbol{\theta}; H)$ as a function of the horizon H can be described as:*

$$
\|\nabla_{\boldsymbol{\theta}} J^{g_{\text{ATT}}}(\boldsymbol{\theta}; H)\| = O\left(H^3\right).
$$

Proof. We begin with the definition of the self-attention Action-conditioned World Model. Let $Q \in \mathbb{R}^{1 \times d_z}$, $K \in \mathbb{R}^{n \times d_z}$, $V \in \mathbb{R}^{n \times d_o}$, attention can be defined as :

$$
\text{Attention}(Q, K, V) := \text{softmax}(QK^T)V
$$

A self-attention AWM can then be defined in the following way with weight matrices $W_q \in \mathbb{R}^{d_z \times d_a}$, $W_k \in \mathbb{R}^{d_z \times d_a}$, $W_v \in \mathbb{R}^{d_s \times d_a}$.

$$\texttt{ATT}_t(a_1, ...a_{t-1}) := \text{Attention}(a_{t-1}W_q^T, a_{1:t-1}W_k^T, a_{1:t-1}W_v^T) = \sum_{i=1}^{t-1} c_i(a_i W_v^T) \quad = \hat{s}_t,$$

where $c_i = \text{softmax}_i(a_{t-1}W_q^T W_k a_{1:t-1}^T)$. The subscript on the softmax operator indicates the ith index. Now, we begin by showing the expression for $\frac{\partial g_t}{\partial a_k}$:

$$\begin{aligned}
\frac{\partial g_t}{\partial a_k} &= \sum_{i=1}^{t-1} \frac{\partial}{\partial a_k} c_i(a_i W_v^T) \\
&= \sum_{i=1}^{t-1} \frac{\partial c_i}{\partial a_k}(a_i W_v^T) + c_i \frac{\partial (a_i W_v^T)}{\partial a_k} \\
\text{By Softmax derivative} \quad &= c_k W_v^T + \sum_{i=1}^{t-1} \Big(c_i(1\{i=k\} - c_k)(a_i W_v^T) \Big) \ .
\end{aligned}$$

Then, we take the norm:

$$\begin{aligned}
\left\| \frac{\partial g_t}{\partial a_k} \right\| &\leq \|c_k W_v^T\| + \sum_{i=1}^{t-1} \left\| c_i(1\{i=k\} - c_k)(a_i W_v^T) \right\| \quad \text{Triangle Inequality} \\
\text{Cauchy-Schwarz Inequality} \quad &\leq \|c_k W_v^T\| + \sum_{i=1}^{t-1} \|c_i(1\{i=k\} - c_k)\| \left\| (a_i W_v^T) \right\| \\
\text{By definition of Softmax} \quad &\leq \|W_v^T\| + \sum_{i=1}^{t-1} \left\| (a_i W_v^T) \right\| \\
\text{Assuming bounded actions} \quad &\leq \|W_v^T\| + \alpha \sum_{i=1}^{t-1} \left\| W_v^T \right\| \quad \text{, where } a_i \leq \alpha \ \forall \ i \ .
\end{aligned}$$

Finally, we use the bound derived in Theorem 1 to finalize the proof:

$$\begin{aligned}
\|\nabla_{\boldsymbol{\theta}} J^g(\boldsymbol{\theta}; H)\| &\leq L_r L_\pi \sum_{t=1}^{H} \sum_{k=1}^{t-1} \left(\|W_v^T\| + \alpha \sum_{i=1}^{t-1} \|W_v^T\| \right) \\
&= O(H^3 \alpha \|W_v^T\|) \\
&= O(H^3) \ .
\end{aligned}$$

□

A.6. Tightness of Theorem 3.1

The result of Theorem 3.1 argues that policy gradients computed through equation 2 may explode as the horizon grows. To make this claim, we comment on the tightness of the upper bound provided in Theorem 3.1. Consider the one-dimensional case, where all the gradients are positive scalars, and $\|\frac{\partial h(\hat{s}_{1:t}, a_{1:t})}{\partial a_k}\| = L_a$ and $\|\frac{\partial h(\hat{s}_{1:t}, a_{1:t})}{\partial \hat{s}_i}\| = L_s$ for all $\hat{s}_{1:t}, a_{1:t}, k, i$. Then, we show that

$$\|\nabla_{\boldsymbol{\theta}} J^h(\boldsymbol{\theta}; H)\| = \Omega(L_s^H)$$

Proof. We begin the proof of this lower bound in a similar way, by considering only one of the terms of the summation.

$$\begin{aligned}
\|\nabla_{\boldsymbol{\theta}} J^h(\boldsymbol{\theta}; H)\| &= \left\| \sum_{t=1}^{H} \frac{\partial r(\hat{s}_t)}{\partial \hat{s}_t} \sum_{k=1}^{t-1} \frac{dh(\hat{s}_{1:t-1}, a_{1:t-1})}{da_k} \frac{\partial \pi_{\boldsymbol{\theta}}(\hat{s}_k)}{\partial \boldsymbol{\theta}} \right\| \\
&\geq \left\| \frac{\partial r(\hat{s}_H)}{\partial \hat{s}_H} \frac{dh(\hat{s}_{1:H-1}, a_{1:H-1})}{da_1} \frac{\partial \pi_{\boldsymbol{\theta}}(\hat{s}_1)}{\partial \boldsymbol{\theta}} \right\| \\
&= \left\| \frac{\partial r(\hat{s}_H)}{\partial \hat{s}_H} \right\| \left\| \frac{dh(\hat{s}_{1:H-1}, a_{1:H-1})}{da_1} \right\| \left\| \frac{\partial \pi_{\boldsymbol{\theta}}(\hat{s}_1)}{\partial \boldsymbol{\theta}} \right\| \\
&\geq \left\| \frac{\partial r(\hat{s}_H)}{\partial \hat{s}_H} \right\| \left\| \frac{\partial \pi_{\boldsymbol{\theta}}(\hat{s}_1)}{\partial \boldsymbol{\theta}} \right\| \left\| \left(\frac{\partial h(\hat{s}_{1:H-1}, a_{1:H-1})}{\partial a_1} + \sum_{i=2}^{H-1} \frac{dh(\hat{s}_{1:H-1}, a_{1:H-1})}{\partial s_i} \frac{dh(\hat{s}_{1:i-1}, a_{1:i-1})}{da_1} \right) \right\| \ .
\end{aligned}$$

Consider now only the longest chain of products in the summation term of the last line. This corresponds to the longest path of the policy gradient through the History-conditioned World Model.

$$\begin{aligned}
\|\nabla_{\boldsymbol{\theta}} J^h(\boldsymbol{\theta}; H)\| &\geq \left\| \frac{\partial r(\hat{s}_H)}{\partial \hat{s}_H} \right\| \left\| \frac{\partial \pi_{\boldsymbol{\theta}}(\hat{s}_1)}{\partial \boldsymbol{\theta}} \right\| \left\| \left(\frac{\partial h(\hat{s}_{1:H-1}, a_{1:H-1})}{\partial a_1} + \frac{dh(\hat{s}_{1:2}, a_{1:2})}{\partial a_1} \prod_{i=2}^{H-1} \frac{dh(\hat{s}_{1:i-1}, a_{1:i-1})}{\partial s_i} \right) \right\| \\
&= \Omega(L_a + L_s^H) \ .
\end{aligned}$$

□

A.7. Tightness of Corollary 4.1

The result of Corollary 4.1 argues that policy gradients through Markovian models may explode as the horizon grows. In order to make this claim, we comment on the tightness of the bound. Just like for Theorem 3.1, consider the one-dimensional case, where all the gradients are positive scalars and assume further the gradient norm of the activation function σ is lower bounded by $\gamma > 0$ (which is the case for popular activation functions such as sigmoid). We will show that, given these conditions, the bound can be written as a lower bound:

$$\|\nabla_{\boldsymbol{\theta}} J^{g_{\text{RNN}}}(\boldsymbol{\theta}; H)\| = \Omega(\eta^H) \ .$$

Proof. We begin the proof by lower bounding the expression for the recurrent cell $\frac{\partial x_t}{\partial a_k}$:

$$\begin{aligned}
\frac{\partial x_t}{\partial a_k} &\geq ||W_a^T||(||W_x^T||\gamma)^{t-k} \\
\left\| \frac{\partial \hat{s}_t}{\partial a_k} \right\| &\geq \|W_o\| ||W_a^T||(||W_x^T||\gamma)^{t-k} \ .
\end{aligned}$$

The bound found in Theorem 4.1 can also be rewritten as an equality given these additional assumptions, since the Cauchy-Schwarz and triangle inequalities become equalities in the one-dimensional positive case: $\|\nabla_{\boldsymbol{\theta}} J^g(\boldsymbol{\theta}; H)\| = L_r L_\pi \sum_{t=1}^{H} \sum_{k=1}^{t-1} \left\| \frac{\partial g_t}{\partial a_k} \right\|$. Finally, putting these new results together, we find that

$$\begin{aligned}
\|\nabla_{\boldsymbol{\theta}} J^{g_{\text{RNN}}}(\boldsymbol{\theta}; H)\| &\geq L_r L_\pi \sum_{t=1} \sum_{k=1}^{t-1} \|W_o\| \|W_a^T\| (\|W_x^T\|\gamma)^{t-k} \\
&\geq L_r L_\pi \|W_o\| \|W_a^T\| (\|W_x^T\|\gamma)^{H-1} \\
&= \Omega(\eta^H) \ ,
\end{aligned}$$

where $\eta = \|W_x^T\|\gamma$. □

A.8. Action-conditioned and Markovian Models Equivalency

The policy gradient framework provided by Action-conditioned World Models can also be used to interpret training design choices used in RL through the lens of sequence modeling. As a concrete example, recall that the transition function of a Markovian model can be trained with the following loss:

$$\ell^{\hat{f}}(\tau; \boldsymbol{\psi}) = \sum_{t=1}^{H-1} \|s_{t+1}^{\tau} - \hat{f}_{\boldsymbol{\psi}}(s_t^{\tau}, a_t^{\tau})\|^2 \ . \tag{6}$$

We present the following remark.

Remark A.1. *Training an AWM instantiated with an RNN to predict states using teacher forcing is equivalent to training its recurrent cell with the loss function from Equation 6.*

Teacher forcing (Williams & Zipser, 1989) is an algorithm used to train recurrent neural networks. Instead of autoregressively unrolling them during training, it enforces the outputs (i.e., in this case, the environment/RNN states) to be the ones coming from a ground-truth sequence. In other words, when training an AWM as an RNN with teacher forcing, we are essentially training its recurrent cell as a one-step model, as traditionally done in model-based RL applied to a Markovian setting.

B. Additional Experimental Details for Section 5

B.1. Hyperparameters

Model-based Hyperparameters. All model-based backpropagation-based policy optimization methods, including the Markovian agent, AWMs and HWMs were sweeped on three entropy constants for exploration: $[0.1, 0.01, 0.001]$, and the best performing results are reported. Indeed, in addition to the objective function J, the policies are optimized with an additional entropy maximization factor as done in Amos et al. (2021) for exploration. All actions, and states (for History-conditioned World Models) are embedded with a linear layer with an output size of 72. The RNN is initialized with two hidden layers, with a hidden layer size of (64, 64). The self-attention transformers use a similar architecture to the GPT-2 model (Radford et al., 2019) implemented by the Hugging Face Transformer library (Wolf et al., 2019). Our transformers stack 2 layers and 3 heads of self-attention modules, with hidden layers of size 64. Timesteps are added as an input to every input of all world models, since we are in a finite-horizon setting. The Markovian world models predicts transitions as a difference function: $\hat{s}_t = \hat{f}(s_{t-1}, a_{t-1}) + s_{t-1}$, using two hidden layers of size (64, 64), and ReLU activation functions. Gradient norms are clipped at a value of 100 for all policy gradients. All other hyper-parameters relating to the policy learning algorithm are shown in Table 1.

Hyperparameter	Value
Number of Environment steps	200000
Dynamics replay ratio	2
Policy replay ratio	16
Dynamics batch size	64
Policy batch size	16
Dynamics learning rate	0.001
Policy learning rate	0.0001
Replay buffer size	1e6
Warmup steps	1500

Table 1. Hyper-parameters for all BPO algorithms that do not pertain to the world model hyper-parameters.

Model-free Hyperparameters. We use a model-free soft-actor critic (Haarnoja et al., 2018) as a benchmark for the myriad experiments. The critic of this agent is modeled as a 2-layer MLP with 256 hidden units each. Entropy regularization is done using a constant entropy constant. Results in Figure 6 show the best performing results after doing a grid search over the following hyper-parameters: learning rate $= [0.001, 0.0001, 0.00001]$ and entropy constant $= [1., 0.01, 0.001]$. The remaining hyperparameters are shown in Table 2.

Hyperparameter	Value
Number of environment steps	200000
Critic replay ratio	2
Policy replay ratio	16
Batch size	128
Discount factor	0.995
Polyak averaging factor	0.995
Replay buffer size	1e6
Warmup steps	1500

Table 2. Hyper-parameters for all model-free results on the Myriad environments.

B.2. Offline Experiments

The experiments showed in Figure 3 and Figure 4 are in the offline setting. In both settings, 100000 transitions are collected for all experiments using a uniform random policy. The world models are then trained on data sampled from these transitions for 1000 steps. The batch size and learning rate are set to the same values as shown in Table 1.

B.3. Online Double-pendulum Experiments

The goal position of the environment is calculated using some fixed initial angular position (initial action) for all episode lengths. In our experiments, the optimal action is -0.4, which corresponds to an initial angle of $-0.4 \times 180°$. The results in the double-pendulum experiments use the same hype-parameters as the Myriad experiments, with two exceptions: the number of environment steps used for optimization is 100000 instead of 200000, and the leaerning rate is sweeped for values in [0.01, 0.001, 0.0001, 0.00001, 0.000001] due to the unstable gradients. The complete learning curves are shown in Figure 7.

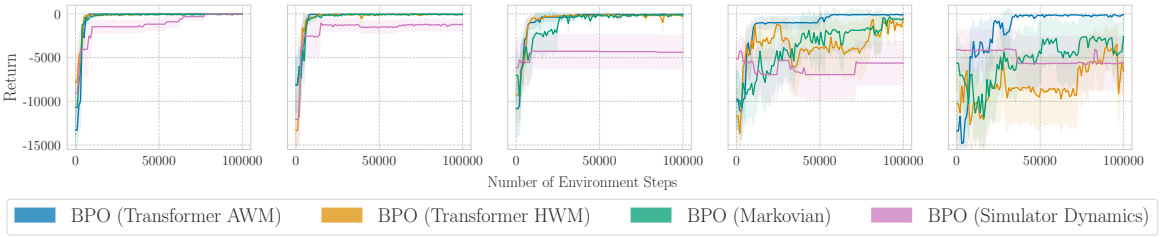


Figure 7. Learning curves of different world models used for BPO on the double-pendulum environment for different horizons. The horizons presented are $H = [5, 10, 20, 50, 100]$ in order (10 seeds $\pm$ std).

B.4. Myriad Environments

We give a short overview of each of the Myriad environments used in this work. For more details, refer to (Howe et al., 2022) and (Lenhart & Workman, 2007).

The underlying dynamics of each of these environments are described by a set of ordinary differential equations, which are then discretized using Euler's method for discrete-time optimal control. In practice, to change a problem's horizon, we fix the duration of the experiment to ensure it is still interesting and simply discretize using a smaller or larger step size. We normalize the returns, where 0 is the expected performance of a random policy and 1 is the performance of an optimal policy provided by (Howe et al., 2022)[5]. Our experiments are conducted on eight environments: *cancer treatment, bioreactor, mould fungicide, bacteria, harvest, invasive plant, HIV treatment,* and *timber harvest*. Each environment's state and action space varies between one and five dimensions.

Cancer treatment follows the normalized density of a cancerous tumour undergoing chemotherapy. The actions at every time step correspond to the strength of the chemotherapy drug at a given time. The goal is to minimize the size of the tumour

[5]Optimal policies are computed using trajectory optimization on the underlying differential equations.

over a set fixed duration while also minimizing the amount of drugs administered to the patient.

Bioreactor seeks to minimize the total amount of a chemical contaminant that naturally degrades in the presence of bacteria. The actions here allow the agent to feed the bacteria, increasing its population and increasing the rate of the contaminants' degradation. However, a cost is associated with feeding the bacteria.

Mould fungicide models the concentration of a mould population. The goal is to minimize its population by applying a fungicide, which has an associated cost to apply.

Bacteria looks to maximize a bacteria population through the application of a chemical nutrient that stimulates growth. On top of the associated cost of applying the chemical nutrient, the chemical also produces a byproduct that might, in turn, hinder bacterial growth.

Harvest models the growing population of some vegetables, and the goal is to maximize the harvested yield of this population. While harvesting directly contributes to the reward, it consequently slows down the population's exponential growth.

Invasive plant seeks to minimize the presence of an invasive plant species through interventions that remove a proportion of the invasive population. These actions have an associated cost.

HIV treatment follows the evolution of uninfected and infected cells in the presence of a virus. The actions correspond to a drug administered that affects the virus' rate of infection. The use of the drug must also be minimized.

Timber harvest is similar to the harvest environment, except the harvested population is infinite. Instead, harvested timber can be converted into capital, which can then be re-invested in the harvesting operation, stimulating company growth. The goal is to maximize revenue.

To the best of our knowledge, this paper presents the first results using reinforcement learning for the environments in Myriad. The final performances of all BPO methods and the SAC agent for each of the eight environments are shown in Figures 8. Note that not only does our method (BPO with transformer AWM) have the best aggregate scores for long horizons, but it also achieves the best performance for the longest horizon of 500 steps for each environment.

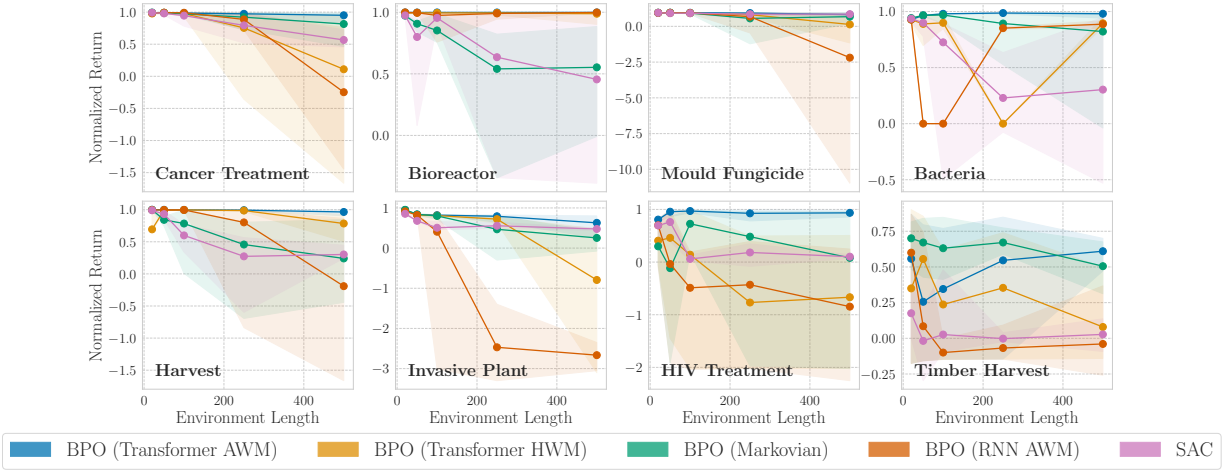


Figure 8. **Policy gradients through transformer AWMs give better policies for long horizons for every environment.** Learning curves of BPO with different world models on eight of the Myriad environments (10 seeds ± min/max)

.

B.5. Online Decision Transformer Results

Recently, a sequence modeling perspective of reinforcement learning has shown promising results on various continuous control tasks in an offline RL setting (Chen et al., 2021; Zheng et al., 2022; Janner et al., 2021). Although our method differs significantly both conceptually and in the problems they solve (see Appendix D), we show experimentally, for completeness, that the online decision transformer (Zheng et al., 2022) performs poorly on the low dimensional Myriad suite. We use

the code and hyperparameters provided by (Zheng et al., 2022) in a purely online setup, with a sweep on the number of trajectories gathered per iteration due to the online nature of our problem setup. Importantly, decision transformers (DT) must be conditioned on the *return-to-go* (RTG) to derive desired policies. Prior works (Chen et al., 2021; Zheng et al., 2022; Janner et al., 2021) have shown that DTs are robust to these hyperparameters and can achieve good and sometimes better performance even when the RTG is set to an out-of-distribution return that is impossible to attain.

We show in Figure 9 that DT performs poorly on the Myriad tasks in an online setting. We summarize some of the takeaways from these experiments:

- Decision transformers are still not well suited for a purely online training regime, exhibiting worse performance than simple one-step model-based methods. Moreover, the action-sequence model outperforms the decision transformer in every single environment.
- Decision transformers require some expert knowledge in many domains, and simply overshooting the return-to-go can yield sub-optimal and sometimes catastrophic results.
- These methods do not scale well with the horizon unlike BPO with Action-conditioned World Models. Even on relatively short horizons such as 100 time steps, the online decision transformer's performance quickly drops when compared to its performance for 20 time steps.

Figure 9. Aggregate final performance of an online decision transformer on the Myriad environments, trained on 200000 environment steps. The *Expert RTG* is conditioned on the optimal return for each environment, while the *Overshot RTG* is conditioned on a return two times higher than the optimal return (10 seeds $\pm$ 95% C.I.).

C. Additional Experiments

C.1. Ablation on Stop Gradient

Most of the analysis in this work relies on the assumption that the gradient is stopped through the policy input, as seen in equation 1, equation 2 and equation 4. As discussed in Section 2, we perform an ablation on whether the stop-gradient operator on the policy input has any major impact on the final performance on the Myriad benchmark, much like . The results in Figure 10 confirm previous remarks (Hafner et al., 2021; 2023; Ghugare et al., 2023) that detaching the policy input from the policy gradient has little to no effect on final performance.

C.2. Ablation on LSTMs

Our analysis suggests that policy gradients through an AWM inherit certain gradient properties of the underlying world model architecture. Long-short-term memory networks (LSTMs) (Hochreiter, 1998) were also suggested as an alternative to vanilla RNNs due to their favorable long-term gradients. Indeed, we show in an ablation study that LSTM AWMs also perform quite well on the Myriad benchmark in Figure 11. Thus, the Action-conditioned World Model is agnostic to the

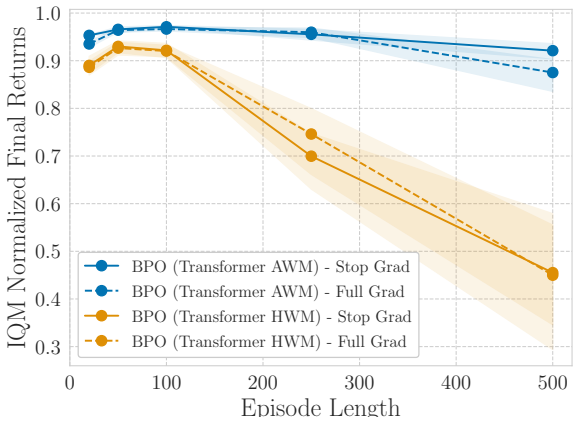


Figure 10. **Stop gradients on the policy input do not affect final performances on the Myriad benchmark.** Final performances of BPO with a transformer AWM and a transformer HWM, with and without the full policy gradient (10 seeds ± 95% C.I.). *Stop Grad* denotes the policy gradient used throughout the main text of this paper, while *Full Grad* denotes the full unbiased policy gradient without any stop gradient operators.

specific neural network architecture, allowing any modern sequence models to produce favorable policy gradients, which could be an interesting direction to explore other more advanced sequence models such as state space models (Gu et al., 2022). We hypothesize that LSTMs are quite effective on environments from Myriad due to the recurrent inductive bias that is compatible with all MDPs. We suspect transformer AWMs show greater promise for large-scale experiments, especially in POMDPs.

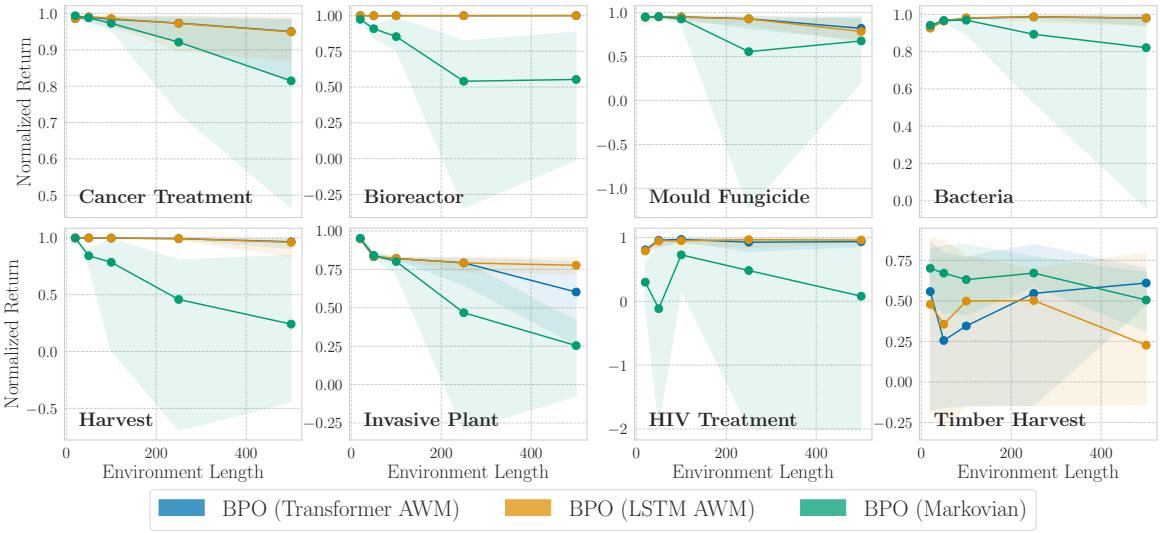


Figure 11. **Ablation with LSTM Action-conditioned World Models on Myriad**. Learning curves of BPO with a transformer AWM, LSTM AWM, and a Markovian world model on eight of the Myriad environments (10 seeds ± min/max).

C.3. Stochastic Environments

In this section, we conduct additional experiments to show how AWMs can also be robust to various forms of stochastic environments. First, we show that action-conditioned world models are actually more robust to stochasticity in the form of distracted MDPs when compared to the history-conditioned models, and just as robust as Markovian models. Second, we consider a stochastic environment specifically designed to be unsuitable for action-conditioned world models (Rezende

et al., 2020), and show that AWMs are still effective in such environments under a small change in sampling procedure.

Random Noise in Distracted Myriad. An important challenge that arises from stochastic environments is to ensure our method is robust to noise. A common way to test for this is to inject noise in so-called *distractor* dimensions (Zhang et al., 2021; Ni et al., 2024). In these problems, an environment is made more difficult by adding dimensions to the observations of an environment which contain noise sampled independently from the states or actions. We evaluate the performance of AWMs in a distracted variant of Myriad, where N dimensions with noise sampled from $\mathcal{N}(0, 1)$ are appended to all observations. We vary the number of distractions in Myriad, and show the final performance of different world models for different horizons in Figure 12. These results demonstrate that AWMs still outperform both Markovian and History-conditoned world models for long horizons, and are especially more robust to noise than HWMs.

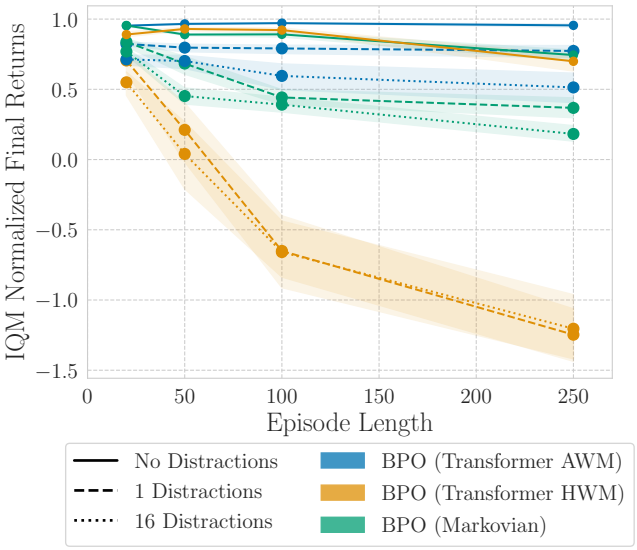


Figure 12. **Action-conditioned World Models still outperform Markovian and History-conditoned World Models under stochasticity in the form of distractor dimensions**. Aggregate final performance of Backpropagation-based policy optimization using different models on a stochastic variant of the Myriad suite with 1 and 16 distractor dimensions.

Stochastic Transitions in FuzzyBear. Rezende et al. (2020) note that models conditioned only on actions, without intermediate states, are not suitable for stochastic environments. They illustrate this point with a simple MDP named *FuzzyBear*, where a stochastic transition in an intermediate state presents new information which needs to be considered for an agent to act optimally. Concretely, consider a continuous alternative to the one-dimensional *FuzzyBear* environment with the following dynamics and reward function.

$$
\begin{aligned}
&s_0 := 0 \\
&s_1 \sim \mathcal{N}(0, 1) \\
&s_2 := s_1 \times a_1 \\
&r_t := s_t \ \text{ if } \ t = 2 \ \text{ else } \ 0 \ ,
\end{aligned}
$$

where actions can be taken from the interval $[-1, 1]$, s_2 is a terminal state and all states include timestep information to preserve the Markov property. In this environment, the optimal action is -1 if $s_1 < 0$ and $+1$ if $s_1 \geq 0$, thus a world model conditioned only on actions will not be able to appropriately predict final rewards without intermediate state information. To circumvent this problem, we propose performing rollouts from any randomly sampled intermediate state, instead of only performing rollouts from the initial state. This is common practice in model-based algorithms (Ghugare et al., 2023; Hafner et al., 2021), and allows AWMs to capture intermediate state information while maintaining constant gradient path lengths. We demonstrate the failure case of AWMs in the FuzzyBear environment described above in Figure 13, but also show that performing random intermediate state rollouts addresses this problem. This simple change lets AWMs perform just as well as Markovian models in a stochastic environment specifically designed to break action-conditioned models. These

experiments suggest that Action-conditioned World Models have a promising future in both deterministic and stochastic environments alike.

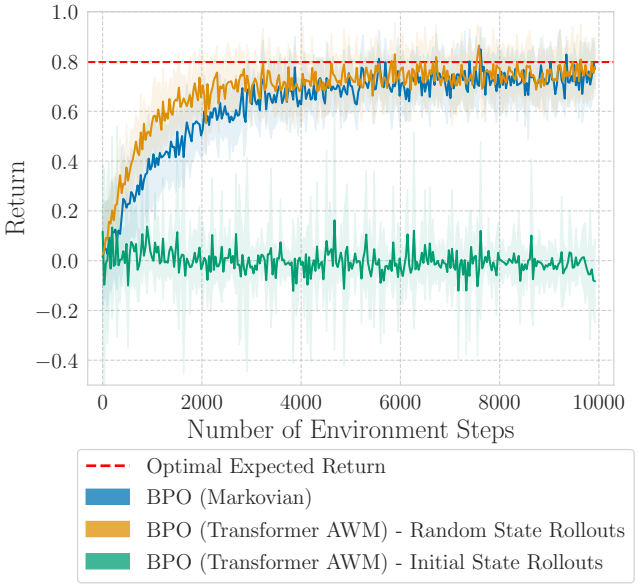


Figure 13. **Action-conditioned World Models can overcome adversarial stochastic environments by performing rollouts from any intermediate state.** Learning curves of Backpropagation-based policy optimization with a Markovian model, and a Transformer AWM with different sampling procedures for rollouts (10 seeds $\pm$ std). To account for stochastic transitions, all world models are modeled as a gaussian distribution with standard deviation fixed to 1.

D. Extended Related Work on Sequence Models in RL

Sequence models in RL have primarily been used in one of three ways. First, sequence models can be used as history encoders in RL algorithms to maximize returns in partially observable MDPs (POMDPs) (Hausknecht & Stone, 2015; Ni et al., 2022), sometimes through a history-dependent world model (Hafner et al., 2021; 2023). Second, sequence models have recently shown promise in an imitation learning or offline reinforcement learning setting by treating MDPs as a sequence modeling problem (Chen et al., 2021; Zheng et al., 2022; Janner et al., 2021), usually conditioning on returns to derive desired policies. Lastly, a separate line of work has used sequence models to reshape the reward landscape for improved temporal credit assignment (Hung et al., 2019; Arjona-Medina et al., 2019; Liu et al., 2019). In contrast to all of these, our framework is the only one to use an *action-only* conditioned sequence model to directly improve long-term policy gradients in MDPs with no intermediate step. Below, we go into a detailed comparison with each area.

Comparison with history-conditioned RL methods for POMDPs. History-conditioned encoders, modeled as sequence models, are often used in POMDPs in both model-free (Hausknecht & Stone, 2015; Parisotto et al., 2020; Lu et al., 2024; Ni et al., 2022) and model-based methods (Hafner et al., 2021; 2023; Samsami et al., 2024). The latter is typically also concerned with predicting observations, they are conditioned on entire history information, while our method includes only actions. The problem setup is also different; such methods are usually concerned with memory in POMDPs (Ni et al., 2023), while ours seeks to improve credit assignment in MDPs.

Comparison with decision and trajectory transformers. Decision transformers (Chen et al., 2021; Zheng et al., 2022) and trajectory transformers (Janner et al., 2021) take a more extreme approach, casting the entire reinforcement learning problem as a sequence modeling one. Conversely, we specifically draw a parallel between policy gradients and sequence models. Their sequence models are conditioned on entire trajectories, which include states, actions, rewards and returns. While trajectory transformers predict states just like our Action-conditioned World Models, both trajectory and decision transformers also predict actions. In either case, their sequence models must first be trained on an offline dataset in an imitation learning (Janner et al., 2021), offline RL (Chen et al., 2021), or pre-training framework (Zheng et al., 2022), while AWMs work for an online RL setting.

Comparison with reward reshaping methods. Perhaps more relevant, prior works have already tried harnessing advanced sequence models for improved temporal credit assignment in RL (Hung et al., 2019; Arjona-Medina et al., 2019; Liu et al., 2019). In these cases, the predictive power of sequence models, either an LSTM (Arjona-Medina et al., 2019) or transformer (Hung et al., 2019; Liu et al., 2019) are used to redistribute or augment the given reward function to produce a surrogate reward function. This surrogate reward is then used in a more traditional model-free RL algorithm. Again, we stress a fundamental difference in the inputs of the sequence models, which all include state information. Our framework also establishes a more direct path between sequence models and credit assignment, avoiding any intermediate steps like reward reshaping.