

# Differentiable Information Enhanced Model-Based Reinforcement Learning

Xiaoyuan Zhang<sup>1,2,3</sup>, Xinyan Cai<sup>4</sup>, Bo Liu<sup>1</sup>, Weidong Huang<sup>3</sup>,  
Song-Chun Zhu<sup>3,1,2</sup>, Siyuan Qi<sup>3</sup>, Yaodong Yang<sup>1,2\*</sup>

<sup>1</sup> Institute for Artificial Intelligence, Peking University

<sup>2</sup> State Key Laboratory of General Artificial Intelligence, Peking University, Beijing, China

<sup>3</sup> State Key Laboratory of General Artificial Intelligence, BIGAI, Beijing, China

<sup>4</sup> Institute of automation, Chinese Academy of Sciences

## Abstract

Differentiable environments have heralded new possibilities for learning control policies by offering rich differentiable information that facilitates gradient-based methods. In comparison to prevailing model-free reinforcement learning approaches, model-based reinforcement learning (MBRL) methods exhibit the potential to effectively harness the power of differentiable information for recovering the underlying physical dynamics. However, this presents two primary challenges: effectively utilizing differentiable information to 1) construct models with more accurate dynamic prediction and 2) enhance the stability of policy training. In this paper, we propose a Differentiable Information Enhanced MBRL method, MB-MIX, to address both challenges. Firstly, we adopt a Sobolev model training approach that penalizes incorrect model gradient outputs, enhancing prediction accuracy and yielding more precise models that faithfully capture system dynamics. Secondly, we introduce mixing lengths of truncated learning windows to reduce the variance in policy gradient estimation, resulting in improved stability during policy learning. To validate the effectiveness of our approach in differentiable environments, we provide theoretical analysis and empirical results. Notably, our approach outperforms previous model-based and model-free methods, in multiple challenging tasks involving controllable rigid robots such as humanoid robots' motion control and deformable object manipulation.

## 1 Introduction

Robot control aims to develop effective control inputs to guide robots in accomplishing assigned tasks. Differentiable environments present a promising opportunity, enabling precise computation of first-order gradients of task rewards with respect to control inputs (Xu et al. 2021). This gradient information facilitates gradient-based policy optimization methods (Antonova et al. 2023; Freeman et al. 2021). Model-based reinforcement learning (MBRL) methods show greater potential than model-free approaches in leveraging environmental differentiable information to construct accurate models, thereby improving performance (Moerland et al. 2023).

However, the design of MBRL methods in differentiable environments presents new challenges. Firstly, ensuring the stability of policy training in a differentiable environment is crucial. Gradient-based methods have been developed, with the most representative one being SHAC (Xu et al. 2021) (which we will also refer to as a differentiable-based method in the following text). These algorithms make better use of gradient information and often achieve better experimental performance compared to non-differentiable-based approaches (such as PPO (Schulman et al. 2017), SAC (Haarnoja et al. 2018), etc.) in differentiable environments. However, due to the presence of collisions (discontinuities in gradients) and the issue of vanishing or exploding gradients during long trajectory back-propagation, differentiable-based methods often suffer from instability of training and are highly sensitive to trajectory lengths (Suh et al. 2022). This can be particularly detrimental to robot control. For various types of robot control, especially in environments such as humanoid robots, the stability of policy training is crucial for performance (Andrychowicz et al. 2020; Abeyruwan et al. 2023). Second, in terms of model learning, a critical challenge arises from the accumulation of model errors (Plaat, Kusters, and Preuss 2023), which manifest in both trajectory prediction and first-order gradient prediction (Li et al. 2022).

In this paper, we present a novel MBRL approach, **MB-MIX**, to address these challenges. Our method leverages path derivatives (Clavera, Fu, and Abbeel 2020) for policy optimization. To enhance the stability of policy training, we introduce trajectory length mixing (**MIX**) in subsection 3.2. This technique combines both long and short trajectories, reducing variance in policy gradient estimation. We conduct a theoretical analysis of MIX approach in subsection 3.4, demonstrating its effectiveness in achieving low-variance policy gradient estimation. To address the problem of model errors, we propose **Sobolev model training method** (Czarnecki et al. 2017; Parag et al. 2022) for learning dynamics models that make effective use of differentiable information. This approach trains the model by matching its predictions of environmental dynamics and their first-order gradients. We also analyze the internal connection between these two methods in 3.3. By combining trajectory length mixing and Sobolev training, we mitigate the impact of cumulative prediction errors of long trajectory models on policy training.

\*Corresponding author

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Finally, we validated the effectiveness of our algorithm in multiple scenarios. We first demonstrated the effectiveness of the MIX approach in a designed simple tabular case experiment. We then conducted experiments on two benchmarks, DiffRL (Xu et al. 2021) and Brax (Freeman et al. 2021), which contain classic robot control problems. In DiffRL, our algorithm achieved performance surpassing all state-of-the-art methods. Humanoid robots present a challenging problem in robotics, with bipedal interaction playing a crucial role. These bipedal robots are well-suited for collaboration and work with humans in everyday life and work environments. We innovatively trained a real bipedal humanoid robot, **Bruce** (Liu et al. 2022b), using differentiable simulation (DiffRL), successfully validating the effectiveness of our algorithm and opening up significant possibilities for future real-robot deployments. In Brax, we proved that utilizing a Sobolev training method for the dynamics model provided greater benefits compared to traditional models. Moreover, in DaXBench (Chen et al. 2022), we demonstrated the effectiveness of our method in differentiable deformable object environments with large state and action spaces.

In summary, our work makes three main contributions to MBRL in differentiable environments by effectively utilizing differentiable information. **First**, we propose an MBRL method with trajectory length mixing to reduce variance in policy gradient estimation, thereby achieving more stable training performance. **Second**, we employ Sobolev training to reduce model errors. This training approach is consistent and coherent with gradient-based policy training methods. **Third**, our approach achieves better performance compared to state-of-the-art model-free and model-based reinforcement learning methods in multiple robot-control tasks.

## 2 Related Work

**Model-Based Reinforcement Learning** methods learn dynamic models to guide policy optimization, reducing sample complexity while maintaining performance. Learned dynamics models fall into two categories: First is enhancing model-free methods with the learned model, enhancing policy optimization through path derivatives (Amos et al. 2021; D’Oro and Jaśkowski 2020), where policy gradients are computed via model back-propagation. For example, SVG (Heess et al. 2015) introduced a framework for learning continuous control policies using model-based back-propagation. MAAC (Clavera, Fu, and Abbeel 2020) incorporated learned terminal Q-functions to estimate long-term rewards. DDPPPO (Li et al. 2022) highlights the importance of accurate gradient predictions by the model in influencing policy training. Model-enhanced data methods include MBPO (Janner et al. 2019), which trains the SAC algorithm using generated and real trajectories. Similar ideas have been extended to offline model-based RL settings (Yu et al. 2020; Lee, Lee, and Kim 2021). Impressive advancements have also been made in learning dynamic changes in latent variable spaces (Hafner et al. 2019, 2020, 2021), and the latest development is DreamerV3 (Hafner et al. 2023). Furthermore, the application of transformers as world models (Micheli, Alonso, and Fleuret 2022; Robine et al. 2023;

Ma et al. 2024) has demonstrated robust performance in real humanoid robots (Radosavovic et al. 2023). The second way is to use the model for planning. LOOP (Sikchi, Zhou, and Held 2022), TD-MPC (Hansen, Wang, and Su 2022; Hansen, Su, and Wang 2024) incorporate terminal value estimates for long-term reward estimates. Sobolev training is a method that improves neural network training by using both function values and derivatives as supervision signals (Czarnecki et al. 2017). This can enhance prediction accuracy and generalization (Parag et al. 2022). In this paper, we suggest using the Sobolev training method along with differentiable information from the environment to improve the accuracy of the dynamic model.

**Differentiable Simulators** are physics engines that can compute gradients of physical quantities concerning simulation parameters or inputs (Newbury et al. 2024), categorized into rigid-body and soft-body environments (Hu et al. 2019; Huang et al. 2021). Soft-body simulations, like fluid simulations (Xian et al. 2023) and deformable objects (Li et al. 2023; Chen et al. 2023), face challenges like local optima, while rigid-body simulations (Freeman et al. 2021; Xu et al. 2021) confront discontinuous gradients due to collisions (Zhong, Han, and Brikis 2022). Despite improvements in environment design (Werling et al. 2021; Geilinger et al. 2020), challenges like collision-induced losses persist (Suh et al. 2022). In this paper, we have conducted extensive experiments on both rigid robots and soft deformable differentiable environments. The proposed trajectory length mixing method alleviates the gradient impact caused by collisions to some extent.

**Policy Gradient Estimation** is a technique for optimizing stochastic policies by calculating gradient estimates of expected returns with respect to policy parameters using environment samples (Grondman et al. 2012; Schulman et al. 2015, 2017; Haarnoja et al. 2018). Differentiable simulators support gradient-based policy optimization by providing gradient information (Du et al. 2021; Mora et al. 2021; Freeman et al. 2021; Xu et al. 2021). Our approach use a hybrid of trajectory lengths and effectively reduce the variance of the policy gradient estimation, improving the stability of policy training.

## 3 Method

We propose a differentiable information enhanced Model-Based Reinforcement Learning method, MB-MIX, outlined in Figure 1 and Algorithm 1. In subsection 3.2, we present a model-based method that mixes trajectories of different lengths, effectively reducing variance in policy gradient estimation, and mitigating the issue of accumulated model errors in long trajectory predictions in MBRL. In subsection 3.3, we employ the Sobolev training method, which utilizes state transition function gradients in differentiable environments to optimize dynamics model training, and analyze its consistency with MIX methods. In subsection 3.4, we provide theoretical analysis demonstrating the MIX method’s effectiveness in reducing policy gradient estimation variance.

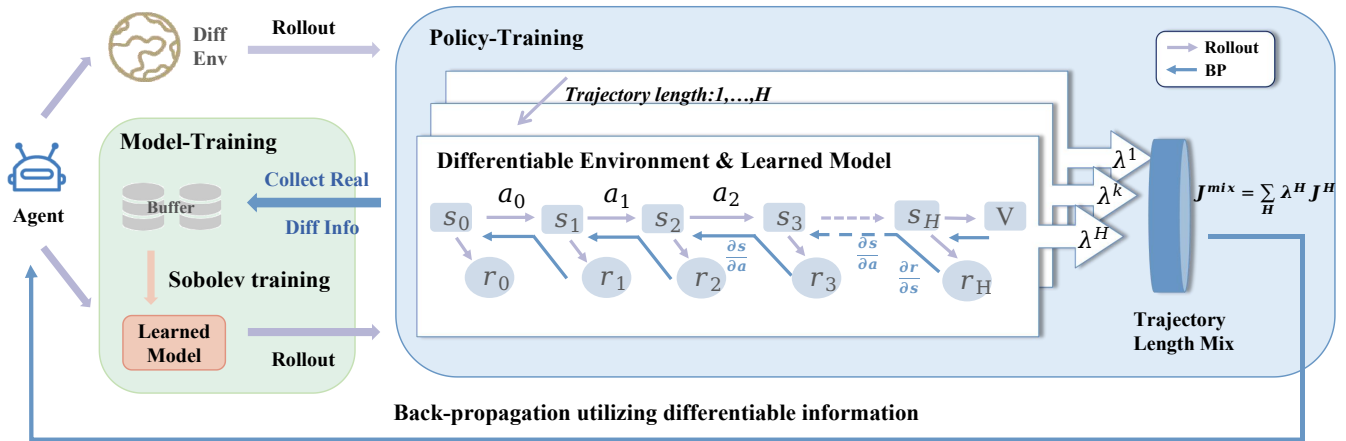


Figure 1: Algorithm diagram. We propose a differentiable information enhanced model-based reinforcement learning approach, **MB-MIX**, which uses **Sobolev model training** method to learn a dynamics model that leverages gradient information(Diff Info) from the differentiable environment. We perform rollouts in the differentiable environment as well as in the learned model, and employ **Trajectory Length Mix** to weight and sum the optimization functions. Policy updates are then performed through Back-propagation. It is worth noting that in our method, the **Model-Training** with Diff Info and the gradient-based **Policy-Training** method are consistent.

### 3.1 Preliminaries

We consider a discrete-time infinite-horizon Markov decision process (MDP) defined by a tuple  $(\mathcal{S}, \mathcal{A}, p_0, f, r, \gamma)$ , where  $\mathcal{S} \subseteq \mathbb{R}^{d_s}$  is the space of states,  $\mathcal{A} \subseteq \mathbb{R}^{d_a}$  the space of actions,  $p_0(s_0)$  the distribution over initial states  $s_0$ ,  $s_{t+1} \sim f(s_t, a_t)$  the transition function,  $r(s_t, a_t)$  the reward function, and  $\gamma \in (0, 1]$  a discount factor. A trajectory  $\tau = (s_0, a_0, s_1, a_1, \dots, s_H, a_H) \in \mathcal{T}$ , is a sequence of states and actions of length  $H$  sampled from the dynamics defined by the MDP and the policy, where the horizon  $H$  may be infinite.  $\mathcal{T} \subseteq \mathbb{R}^{(d_s+d_a) \times (H+1)}$  is the space of trajectories and  $\pi(a_t|s_t; \theta)$  is the policy parameterized by  $\theta \in \mathbb{R}^{d_\theta}$ . The objective of an RL algorithm is to train a policy that maximizes the expected sum of discounted rewards:  $J(\theta) = \mathbb{E}_{\tau \sim p(\tau; \theta)} \left[ \sum_{t=0}^H \gamma^t r(s_t, a_t) \right]$ . A value function  $V(s)$  is often learned to approximate  $J(\theta)$ . The input-output relationship of dynamic model  $M$  is defined as  $\hat{s}_t = M(s_{t-1}, a_{t-1})$ .

### 3.2 Model-Based Approach with Mixed Trajectory Length

Commonly used policy update approaches using path derivatives often collect trajectories of a specific length for reward aggregation, which is used to propagate updates to the policy. In particular, the general form of the objective function can be written as:

$$J_\pi^H(\theta) = E \left[ \sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \gamma^H V(s_H) \right]. \quad (1)$$

In practice, varying trajectory lengths  $H$  significantly impact policy gradient estimation. Moreover, model-based methods can face increased cumulative error with longer trajectory predictions. To address these challenges, we propose trajectory length mixing. This technique blends trajectories of different lengths to reduce policy gradient estima-

tion variance and improve model training stability. Specifically, it computes the optimization function by weighted averaging expected reward aggregation from trajectories of varying lengths:

$$J_\pi^{\text{mix}}(\theta) = (1 - \lambda) \sum_{H=1}^{\infty} \lambda^{H-1} J_\pi^H(\theta). \quad (2)$$

In practice, the trajectory length is upper-bounded by the upper limit of a specific task or simply by choosing a higher length. Additionally, the interval of mixing trajectories of different lengths is adjustable. By substituting Equation 1 into Equation 2, we have:

$$J_\pi^{\text{mix}}(\theta) = E \left[ \sum_{t=0}^{\infty} ((\gamma\lambda)^t (r(s_t, a_t) + (1 - \lambda)\gamma V(s_{t+1})) \right] \quad (3)$$

The transition of the state function is predicted by a learned dynamics model. Here, we adopt the idea of branch rollout (Janner et al. 2019), starting from states in the real environment and rolling out in the model to control the impact of model errors. For policy gradient estimation, the model accuracy has a greater impact on long trajectories, whereas the accuracy of the value function plays a more substantial role for shorter trajectories. Furthermore, Equation 3 represents a general form of the optimization function for gradient-based policy training using path derivatives. When  $\lambda = 1$ ,  $J_\pi^{\text{mix}}(\theta) = E \left[ \sum_{t=0}^{\infty} (\gamma)^t (r(s_t, a_t)) \right]$  equivalent to  $SVG(\infty)$ (Heess et al. 2015). When  $\lambda = 0$ ,  $J_\pi^{\text{mix}}(\theta) = E \left[ (r(s_t, a_t) + \gamma V(s_{t+1})) \right]$  equivalent to  $SVG(1)$ .

### 3.3 Sobolev Model Training Method

To leverage the rich gradient information in differentiable environments, we propose using the Sobolev training method to learn the environment dynamics model. Sobolev training incorporates the gradient information into the loss function, thereby encouraging the model to learn more about the gradients of the input values during the training process.

The loss function is:

$$J_M(\varphi) = E \left[ \|\hat{s}_{t+1} - s_{t+1}\|_2 + \alpha \left[ \left\| \frac{\partial \hat{s}_{t+1}}{\partial s_t} - \frac{\partial s_{t+1}}{\partial s_t} \right\|_2 + \left\| \frac{\partial \hat{s}_{t+1}}{\partial a_t} - \frac{\partial s_{t+1}}{\partial a_t} \right\|_2 \right] \right]. \quad (4)$$

where  $\varphi$  represents the parameters of the dynamics model  $M$ , and  $\hat{s}$  represents the predicted state. The dynamics model trained using the Sobolev method maintains consistency with the path gradient-based policy training approach described in subsection 3.2. This can be observed from the chain rule formula for gradient back-propagation in Equation 5, a detailed expansion of  $J^{\text{mix}}$  is included in Appendix:

$$\frac{\partial J_\pi}{\partial \theta} = \mathbb{E} \left[ \sum_{i=0}^{\infty} \frac{\partial J_\pi}{\partial a_i} \frac{\partial a_i}{\partial \theta} \right] = \mathbb{E} \sum_{i=0}^{\infty} \left( \gamma^i \frac{\partial r_i}{\partial a_i} + \sum_{k=i+1}^{\infty} \gamma^k \frac{\partial r_k}{\partial s_k} \frac{\partial s_k}{\partial s_{i+1}} \frac{\partial s_{i+1}}{\partial a_i} \right) \frac{\partial \pi_\theta(s_i)}{\partial \theta}. \quad (5)$$

**Consistency between model training and policy training: MIX** (Using trajectories of different lengths) is helpful for stability, but in the model-based setting, longer trajectories suffer from accumulated errors. In model-based policy training methods utilizing path derivatives, the dynamics model must accurately predict the zeroth-order values of  $(s, a)$  and provide precise predictions for the gradients of the transition function. Previous model-based approaches prioritized accurate zeroth-order values over first-order gradients. Dynamics models trained using Sobolev training methods emphasize the accuracy of both zeroth-order and first-order values. Therefore, to utilize longer trajectories in the MBRL setting, **Sobolev Model training method** is needed to reduce the error in gradient predictions. We list the pseudo code in Algorithm 1.

### 3.4 Theoretical Analysis

In the subsequent analysis, we delve into the Variance for the MIX estimator and the SHAC method (a representative differentiable based method) using a differentiable environment scenario. Based on some common assumptions detailed in Appendix Section A (We consider these assumptions to be common constraints for proving convergence and bounds as shown in (Fallah et al. 2021; Clavera, Fu, and Abbeel 2020)), we deduce an upper bound for the Variance of these estimators and provide the necessary conditions for this bound to hold. A key finding from our analysis is that the MIX estimator of the policy gradient has an existing Variance bound, and this bound is lower than that of the SHAC method for any trajectory length under balance factor  $\lambda$  and discount factor  $\gamma$  less than 1. This result suggests a performance enhancement for the MIX estimator.

**Theorem 1.** *Suppose Assumptions 1, 2, 3, and 4 hold (please see appendix). Then in a differentiable environment, the Variance of the MIX policy gradient estimate ( $A^{\text{MIX}}$ ) will be equal to or less than the Variance of the SHAC policy gradient estimate ( $A^{\text{SHAC}}$ ):*

$$\text{Variance}(A^{\text{MIX}}) \leq \text{Variance}(A^{\text{SHAC}}) \quad (6)$$

---

### Algorithm 1: MB-MIX

---

- 1: **Input:** policy  $\pi_\theta$ , Model  $M_\varphi$ , Value function  $V_\psi$ , Environment buffer  $D_{env}$
  - 2: **for**  $i = 0, \dots, K - 1$  **do**
  - 3:   Sample trajectories  $s_t, a_t, r_t, s_{t+1}, \frac{\partial s_{t+1}}{\partial s_t}, \frac{\partial s_{t+1}}{\partial a_t}$  from real environment with  $\pi_\theta$  add to  $D_{env}$ .
  - 4:   Train  $M_\varphi$  using data from  $D_{env}$  to minimize  $J_M(\varphi)$
  - 5:   **for**  $t = 0, 1, 2, \dots, T$  **do**
  - 6:     sample state, action, reward trajectories through the interaction of policy  $\pi_\theta$  and  $M_\varphi$
  - 7:     compute  $J_\pi^{\text{mix}}(\theta)$  for all time steps.
  - 8:      $\theta = \theta - \alpha \nabla_\theta J_\pi^{\text{mix}}(\theta)$ .
  - 9:      $\hat{V}(s_t) = (1 - \lambda) \left( \sum_{k=1}^{h-t-1} \lambda^{k-1} G_t^k \right) + \lambda^{h-t-1} G_t^{h-t}$ ,  
       where  $G_t^K = \sum_{l=0}^{K-1} \gamma^l r(s_{t+l}) + \gamma^K V(s_{t+K})$
  - 10:   Train  $V_\psi$  to minimize  $L = E_{s \in \tau_i} [\|V_\psi(s) - \hat{V}(s)\|]$ .
  - 11:   **end for**
  - 12: **end for**
  - 13: **return** Optimal policy  $\pi_\theta$
- 

Here,  $A^{\text{MIX}}$  and  $A^{\text{SHAC}}$  respectively denote the MIX and SHAC policy gradient estimates, the balance factor  $\lambda$ , the discount factor  $\gamma$  are positive real numbers less than 1.

Using all of the assumptions, we bound the variance term for MIX and SHAC. The complete proof with all steps and mathematical details can be found in Appendix Section A.

## 4 Experimental Results

Our experiment aims to investigate the following questions: (1) Can trajectory length mixing improve gradient-based policy optimization methods? Especially in terms of training stability. (2) How does the proposed MB-MIX approach perform compared to state-of-the-art reinforcement learning methods? (3) Does the use of a dynamics model trained with Sobolev improve the performance of model-based reinforcement learning methods? We conducted experiments in the environment depicted in Figure 2 to address the aforementioned questions:

1. We developed a straightforward tabular case environment with discrete state and action spaces. Additionally, experimental evaluations were performed on DaXBench, a sophisticated deformable object manipulation environment (Chen et al. 2022). Through these experiments, we provide empirical evidence for the efficacy of trajectory length mixing.
2. We conducted extensive experiments in the differentiable and parallelizable robot control environment, DiffRL (Xu et al. 2021). Our approach was compared against state-of-the-art gradient-based, model-free, and model-based methods, outperforming them across various tasks. Ablation experiments were also performed to analyze the individual effects of each component. Excitingly, we introduced Bruce (Liu et al. 2022b), a real bipedal humanoid robot, into the DiffRL environment. Bruce’s real-world nature intro-

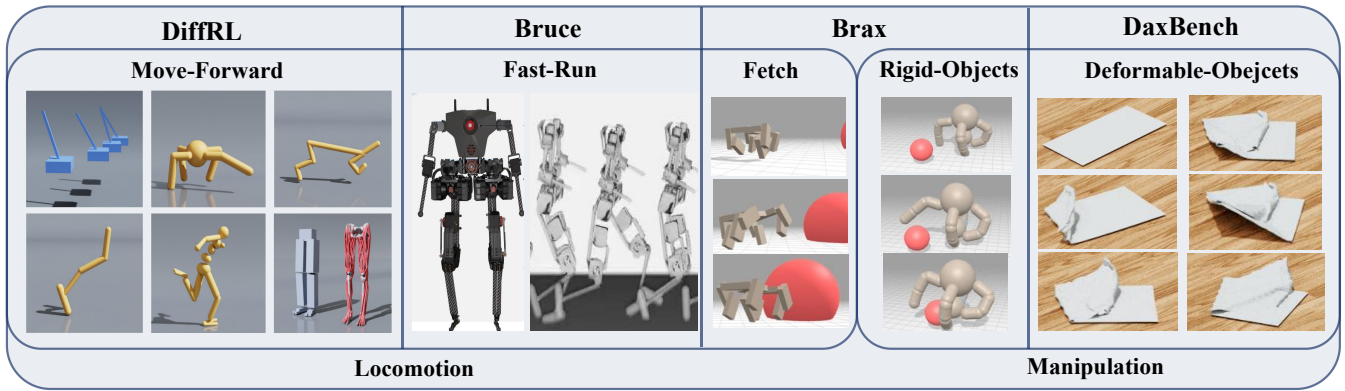


Figure 2: **Task Gallery.** We assessed the algorithm’s effectiveness in four differentiable environments, encompassing various control tasks. (a) **DiffRL:** The agent controls a range of robots, including those with muscles. (b) **Bruce, Humanoid Robot:** Designed for tasks like Fast-Run, was introduced into the DiffRL environment to extend its real-world applications. (c) **Brax:** Involving advanced tasks such as Fetch and Grasp (d) **DaxBench:** Entailing a series of tasks related to deformable objects manipulation.

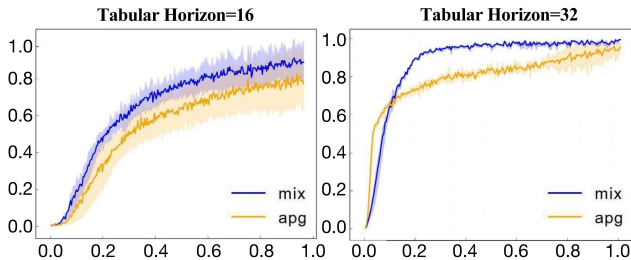


Figure 3: Experiment results in Tabular case. We show the effectiveness of mixing trajectory length in a designed simple tabular case environment. The legend “reward/step” on the y-axis denotes the average reward. The right end of the horizontal axis represents  $1e6$  environment steps.

duced additional constraints and collisions, demanding robustness and posing new challenges in policy training. Remarkably, our algorithm achieved performance surpassing all SOTA baselines, greatly enhancing the potential for deploying it on real robots.

3. We validated the effectiveness of our proposed Sobolev model training approach in the Brax environment, as mentioned in subsection 3.3. Models trained with the Sobolev training approach exhibited better performance in assisting gradient-based policy training.

#### 4.1 Tabular Case: Effectiveness of MIX

**Training Setting:** We consider a tabular random Markov Decision Process setting as described in (Liu et al. 2022a). The state and action spaces have dimensions 20 and 5, respectively, yielding a reward matrix  $R \in \mathbb{R}^{20 \times 5}$ . The transition probability matrix is generated from independent Dirichlet distributions. The initial policy is a matrix  $\theta^0 \in \mathbb{R}^{20 \times 5}$ , and the final policy  $\pi_\theta$  is obtained via softmax activation:  $\pi_\theta(a|s) = \exp(\theta(s, a)) / \sum_b \exp(\theta(s, b))$ . The tabular MDP provides a controlled setting to evaluate the effec-

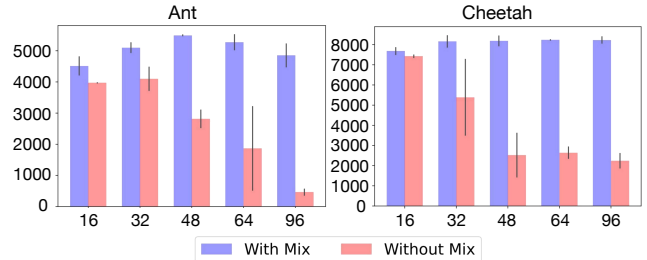


Figure 4: Impact of different trajectory lengths on training. The vertical axis of the figure represents reward, while the horizontal axis represents the maximum length of trajectories. Our Mix method enhances policy training stability.

tiveness of trajectory length mixing. The training objective was to optimize the policy through improved gradient-based methods. The experimental settings involved varying trajectory lengths, using a  $\lambda = 0.98$  discount rate for reward-to-go calculations, and adjusting the mix-interval accordingly.

**Main Result:** As shown in Figure 3, our MIX approach outperforms baseline methods in terms of convergence speed and final policy performance. The baseline APG (Wiedemann et al. 2023) uses complete trajectory lengths for policy training. Without additional noise factors, this experiment indicates that trajectory length mixing can enhance the performance of gradient-based policy optimization.

#### 4.2 DiffRL: Comparison with SOTA methods

**Training Setting:** We conduct experiments in the differentiable parallel environment DiffRL (Xu et al. 2021), which includes tasks such as Ant, Humanoid, and SNU-Humanoid. The objective was reward maximization. Lower parallel environments (4 and 8) were used to highlight sample efficiency of model-based methods. In the experiment, our MB-MIX algorithm was trained on all six tasks, with a  $\lambda = 0.98$  and mix-interval set to 1 or 2 depending on the task.

	Model-Free Method			Model-Based Method			
	SHAC	PPO	SAC	DreamerV3	LOOP	MAAC	<b>MB-MIX(ours)</b>
Ant	5174 $\pm$ 349	1213 $\pm$ 217	1278 $\pm$ 39	1989 $\pm$ 374	2995 $\pm$ 249	5289 $\pm$ 651	<b>6363<math>\pm</math> 27</b>
Cheetah	<b>8366<math>\pm</math> 259</b>	5552 $\pm$ 661	6392 $\pm$ 215	8034 $\pm$ 142	7029 $\pm$ 468	7832 $\pm$ 239	<b>8483<math>\pm</math> 80</b>
Hopper	3638 $\pm$ 222	382 $\pm$ 71	384 $\pm$ 34	3010 $\pm$ 722	542 $\pm$ 297	2056 $\pm$ 317	<b>4048<math>\pm</math> 127</b>
Cartpole	-622 $\pm$ 25	-1794 $\pm$ 108	-1607 $\pm$ 76	-666 $\pm$ 59	-801 $\pm$ 76	-629 $\pm$ 32	<b>-604<math>\pm</math> 9</b>
Humanoid	3321 $\pm$ 968	279 $\pm$ 104	889 $\pm$ 41	294 $\pm$ 103	369 $\pm$ 38	1297 $\pm$ 387	<b>4955<math>\pm</math> 197</b>
SNU-Humanoid	3722 $\pm$ 395	88 $\pm$ 2	49 $\pm$ 7	35 $\pm$ 5	26 $\pm$ 10	2774 $\pm$ 444	<b>3907<math>\pm</math> 94</b>

Table 1: **Experiment results in DiffRL.** In a differentiable robot control environment, our method outperforms state-of-the-art model-free and model-based methods, demonstrating the overall effectiveness.

**Compared Baseline:** The sample efficiency and numerical performance of the proposed MB-MIX algorithm were compared with state-of-the-art model-free and model-based RL methods, including path-derivative-based (e.g., SHAC (Xu et al. 2021)) and non-path-derivative-based (e.g., PPO (Schulman et al. 2017), SAC (Hafner et al. 2018)) model-free methods, as well as model-based methods like DreamerV3 (Hafner et al. 2023), LOOP (Sikchi, Zhou, and Held 2022), and MAAC (Clavera, Fu, and Abbeel 2020).

**Main Result:** Table 1 explicitly demonstrates that our MB-MIX method outperforms all baselines on all six tasks. Particularly, we achieve remarkably impressive performance on the Humanoid and ant tasks. It is worth noting that our algorithm, compared to the state-of-the-art gradient-based method SHAC in differentiable environments, not only achieves better performance but also demonstrates higher stability across all tasks.

**Ablation Experiment:** We conducted experiments on the stability of the trajectory mixing methods. We compared the impact of different trajectory lengths on training in the Ant and Cheetah environments, as illustrated in Figure 4. The x-axis of the graph represents the maximum trajectory length used for each update (the length of interaction between the agent and the environment). The results showed that, under different trajectory lengths, Our **Mix** Method achieved more stable and better policy training performance.

### 4.3 Bruce: Humanoid Robotic Control

**Training Setting:** Bruce is a miniature bipedal robot with five degrees of freedom (DoF) per leg, including a spherical hip joint, a knee joint, and an ankle joint. To reduce leg inertia, Bruce employs cable-driven differential pulley systems and link mechanisms for the hip and ankle joints, respectively, enabling a human-like range of motion (Liu et al. 2022b). "Fast Run" task evaluates robot's forward speed.

**Main Result:** The algorithm's performance on the "Fast Run" task is shown in Figure 5. Our MB-MIX and MIX methods outperform other approaches, exhibiting higher reward and greater stability, consistent with theoretical analysis. Notably, **MB-MIX** demonstrates significant improvement over **MIX**, indicating the learned Dynamic Model effectively aids policy learning, serving as an ablation experiment for the dynamic model. The right side of Figure 5 visualizes the policy performance of **MB-MIX** versus the SHAC algorithm. Starting from the same initial position and

Task	APG	SHAC	<b>MIX</b>
Fold-Cloth-1	0.36 $\pm$ 0.06	0.34 $\pm$ 0.07	<b>0.52<math>\pm</math> 0.02</b>
Fold-Cloth-3	<b>0.19<math>\pm</math> 0.09</b>	<b>0.22<math>\pm</math> 0.22</b>	<b>0.23<math>\pm</math> 0.08</b>
Unfold-Cloth-1	0.42 $\pm$ 0.02	0.50 $\pm$ 0.03	<b>0.73<math>\pm</math> 0.01</b>
Unfold-Cloth-3	0.39 $\pm$ 0.02	<b>0.48<math>\pm</math> 0.03</b>	<b>0.51<math>\pm</math> 0.03</b>

Table 2: **DaXBench.** In the deformable objects environment, our proposed MIX approach achieved better performance compared to gradient-based algorithms

duration, MB-MIX enables the robot to travel farther and perform better in the "Fast Run" task. The MB-MIX-trained robot exhibits alternating fast forward motion, which is more "human-like" and efficient compared to SHAC. Figure 6 demonstrates the learned model's effectiveness in state prediction, with the dynamic model accurately aligning with the ground truth. This provides strong support for reducing wear and tear during real-world robot interactions. Experiment indicates our algorithm's tremendous potential for bipedal robot applications.

### 4.4 Brax: Model Training Performance

**Training Setting:** Brax is a environment with interactive objects. Tasks in Brax include Fetch and Grasp, which have continuous state and action spaces, providing us with more challenging test beds. We used a dynamics model trained with Sobolev training, which allows us to compare the effectiveness of this method against traditional ones.

**Main Result:** The results illustrated in Figure 7 demonstrate the superior performance of our approach, which combines Sobolev model training method for the dynamics model, surpassing the benchmark methods. Notably, the performance gap becomes more pronounced with increasing task complexity, underscoring the effectiveness and scalability of our sobolev model training approach. These findings affirm that the integration of a Sobolev-trained dynamics model can indeed enhance the performance of MBRL methods.

### 4.5 DaXBench: Efficiency on Deformable Objects with Large State-Action Spaces

**Training Setting:** DaXBench (Chen et al. 2022) is a high-performance differentiable simulation platform, ideal for

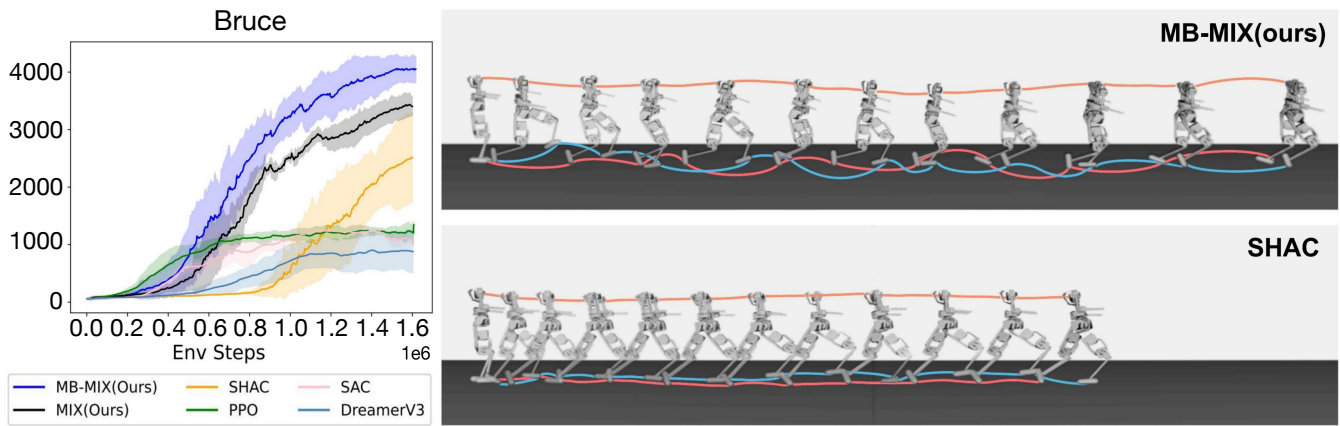


Figure 5: Experiments on Bruce, humanoid robot. The top-left figure demonstrates that our proposed MB-MIX method surpasses all model-free and model-based algorithms, the vertical axis represents the rewards. In the top-right figure, we visualize the performance in the “Fast Run” task. Starting from the same position and after the same amount of time, our MB-MIX method enables the trained robot to move further. Judging from the alternation of the red and blue lines, which track the foot-steps, our MB-MIX algorithm achieves better alternating leg movements.

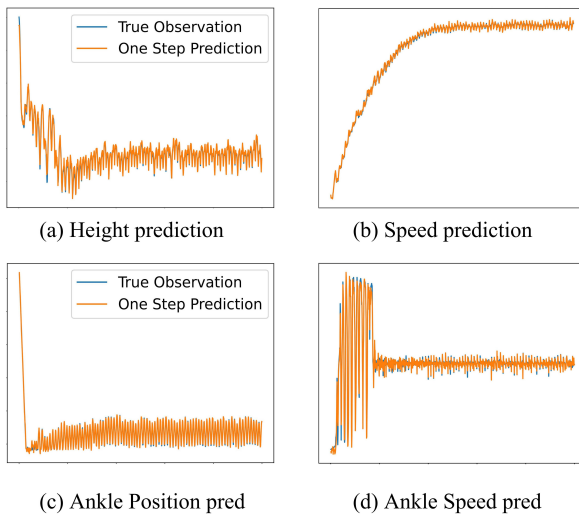


Figure 6: Learned model predictions on Bruce: The state predicted by the model and the actual state. The horizontal axis represents the environmental steps. The vertical axis represents the corresponding predicted values: Height, Speed, Ankle Position, Ankle Speed.

deformable object manipulation (DOM) research. Tasks in DaXBench cover deformable objects and manipulation tasks, providing our method with more challenging test beds. We conducted experiments to evaluate the stability of the trajectory mixing methods within this environment.

**Main Result:** The results in Table 2 demonstrate MIX approach leads to high performance and low variance across all tasks, outperforming the benchmark methods. Indicating the effectiveness and scalability of our approach in diverse and complex tasks. These findings confirm that the MIX approach can indeed enhance the performance of DOM methods, providing a stable and efficient solution.

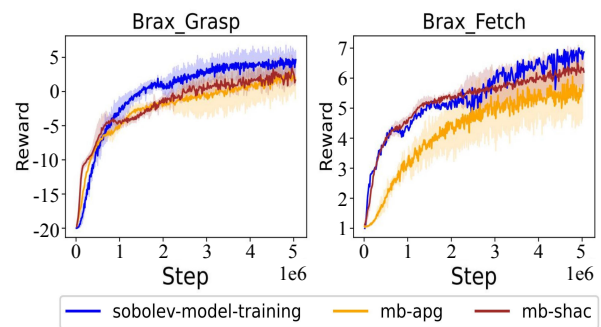


Figure 7: Experiments on the Brax benchmark. In a more complex robot control environment with Fetch and Grasp tasks, sobolev-model-training method enhances the overall performance.

## 5 Conclusion

We propose a novel MBRL framework for differentiable environments. The method introduces a trajectory length mixing technique to mitigate the impact of varying trajectory lengths on policy gradient estimation, thereby enhancing the stability of policy training. Additionally, we innovatively leverage the Sobolev method to learn accurate dynamics models that align with the gradient-based policy training. Theoretical analysis validates the advantages of the trajectory length mixing technique. Experimental evaluation on rigid robot control and deformable object manipulation tasks demonstrates superior performance over prior model-based and model-free approaches.

## 6 Limitations and Future Work

The theoretical analysis is limited. We focus on the MIX method’s impact on policy gradient estimation, yet have not extended the analysis to the model-based setting. Future research could explore the model-based configuration.

## References

- Abeyruwan, S. W.; Graesser, L.; D’Ambrosio, D. B.; Singh, A.; Shankar, A.; Bewley, A.; Jain, D.; Choromanski, K. M.; and Sanketi, P. R. 2023. i-sim2real: Reinforcement learning of robotic policies in tight human-robot interaction loops. In *Conference on Robot Learning*, 212–224. PMLR.
- Amos, B.; Stanton, S.; Yarats, D.; and Wilson, A. G. 2021. On the model-based stochastic value gradient for continuous reinforcement learning. In *Learning for Dynamics and Control*, 6–20. PMLR.
- Andrychowicz, M.; Raichuk, A.; Stańczyk, P.; Orsini, M.; Girgin, S.; Marinier, R.; Hussenot, L.; Geist, M.; Pietquin, O.; Michalski, M.; et al. 2020. What matters in on-policy reinforcement learning? a large-scale empirical study. *arXiv preprint arXiv:2006.05990*.
- Antonova, R.; Yang, J.; Jatavallabhula, K. M.; and Bohg, J. 2023. Rethinking optimization with differentiable simulation from a global perspective. In *Conference on Robot Learning*, 276–286. PMLR.
- Chen, S.; Xu, Y.; Yu, C.; Li, L.; Ma, X.; Xu, Z.; and Hsu, D. 2023. DaxBench: Benchmarking Deformable Object Manipulation with Differentiable Physics. In *The Eleventh International Conference on Learning Representations*.
- Chen, S.; Yu, C.; Xu, Y.; Li, L.; Ma, X.; Xu, Z.; and Hsu, D. 2022. Benchmarking Deformable Object Manipulation with Differentiable Physics. *arXiv preprint arXiv:2210.13066*.
- Clavera, I.; Fu, V.; and Abbeel, P. 2020. Model-augmented actor-critic: Backpropagating through paths. *ICLR*.
- Czarnecki, W. M.; Osindero, S.; Jaderberg, M.; Swirszcz, G.; and Pascanu, R. 2017. Sobolev training for neural networks. *Advances in neural information processing systems*, 30.
- D’Oro, P.; and Jaśkowski, W. 2020. How to learn a useful critic? Model-based action-gradient-estimator policy optimization. *Advances in Neural Information Processing Systems*, 33: 313–324.
- Du, T.; Wu, K.; Ma, P.; Wah, S.; Spielberg, A.; Rus, D.; and Matusik, W. 2021. Diffpd: Differentiable projective dynamics. *ACM Transactions on Graphics (TOG)*, 41(2): 1–21.
- Fallah, A.; Georgiev, K.; Mokhtari, A.; and Ozdaglar, A. 2021. On the convergence theory of debiased model-agnostic meta-reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 3096–3107.
- Freeman, C. D.; Frey, E.; Raichuk, A.; Girgin, S.; Mordatch, I.; and Bachem, O. 2021. Brax—A Differentiable Physics Engine for Large Scale Rigid Body Simulation. *arXiv preprint arXiv:2106.13281*.
- Geilinger, M.; Hahn, D.; Zehnder, J.; Bächer, M.; Thomaszewski, B.; and Coros, S. 2020. Add: Analytically differentiable dynamics for multi-body systems with frictional contact. *ACM Transactions on Graphics (TOG)*, 39(6): 1–15.
- Grondman, I.; Busoniu, L.; Lopes, G. A.; and Babuska, R. 2012. A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6): 1291–1307.
- Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, 1861–1870. PMLR.
- Hafner, D.; Lillicrap, T.; Ba, J.; and Norouzi, M. 2020. Dream to control: Learning behaviors by latent imagination. *ICLR*.
- Hafner, D.; Lillicrap, T.; Fischer, I.; Villegas, R.; Ha, D.; Lee, H.; and Davidson, J. 2019. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, 2555–2565. PMLR.
- Hafner, D.; Lillicrap, T.; Norouzi, M.; and Ba, J. 2021. Mastering atari with discrete world models. *ICLR*.
- Hafner, D.; Pasukonis, J.; Ba, J.; and Lillicrap, T. 2023. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*.
- Hansen, N.; Su, H.; and Wang, X. 2024. TD-MPC2: Scalable, Robust World Models for Continuous Control.
- Hansen, N.; Wang, X.; and Su, H. 2022. Temporal Difference Learning for Model Predictive Control. In *ICML*.
- Heess, N.; Wayne, G.; Silver, D.; Lillicrap, T.; Erez, T.; and Tassa, Y. 2015. Learning continuous control policies by stochastic value gradients. *Advances in neural information processing systems*, 28.
- Hu, Y.; Li, T.-M.; Anderson, L.; Ragan-Kelley, J.; and Durand, F. 2019. Taichi: a language for high-performance computation on spatially sparse data structures. *ACM Transactions on Graphics (TOG)*, 38(6): 201.
- Huang, Z.; Hu, Y.; Du, T.; Zhou, S.; Su, H.; Tenenbaum, J. B.; and Gan, C. 2021. Plasticinelab: A soft-body manipulation benchmark with differentiable physics. *ICLR*.
- Janner, M.; Fu, J.; Zhang, M.; and Levine, S. 2019. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32.
- Lee, B.-J.; Lee, J.; and Kim, K.-E. 2021. Representation balancing offline model-based reinforcement learning. *International Conference on Learning Representations*.
- Li, C.; Wang, Y.; Chen, W.; Liu, Y.; Ma, Z.-M.; and Liu, T.-Y. 2022. Gradient information matters in policy optimization by back-propagating through model. In *International Conference on Learning Representations*.
- Li, S.; Huang, Z.; Chen, T.; Du, T.; Su, H.; Tenenbaum, J. B.; and Gan, C. 2023. DexDeform: Dexterous Deformable Object Manipulation with Human Demonstrations and Differentiable Physics. *The Eleventh International Conference on Learning Representations*.
- Liu, B.; Feng, X.; Ren, J.; Mai, L.; Zhu, R.; Zhang, H.; Wang, J.; and Yang, Y. 2022a. A theoretical understanding of gradient bias in meta-reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 31059–31072.
- Liu, Y.; Shen, J.; Zhang, J.; Zhang, X.; Zhu, T.; and Hong, D. 2022b. Design and control of a miniature bipedal robot with proprioceptive actuation for dynamic behaviors. In *2022 International Conference on Robotics and Automation (ICRA)*, 8547–8553. IEEE.

- Ma, M.; Ni, T.; Gehring, C.; D’Oro, P.; and Bacon, P.-L. 2024. Do Transformer World Models Give Better Policy Gradients? *arXiv preprint arXiv:2402.05290*.
- Micheli, V.; Alonso, E.; and Fleuret, F. 2022. Transformers are sample efficient world models. *arXiv preprint arXiv:2209.00588*.
- Moerland, T. M.; Broekens, J.; Plaat, A.; Jonker, C. M.; et al. 2023. Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 16(1): 1–118.
- Mora, M. A. Z.; Peychev, M.; Ha, S.; Vechev, M.; and Coros, S. 2021. Pods: Policy optimization via differentiable simulation. In *International Conference on Machine Learning*, 7805–7817. PMLR.
- Newbury, R.; Collins, J.; He, K.; Pan, J.; Posner, I.; Howard, D.; and Cosgun, A. 2024. A Review of Differentiable Simulators. *IEEE Access*.
- Parag, A.; Kleff, S.; Saci, L.; Mansard, N.; and Stasse, O. 2022. Value learning from trajectory optimization and Sobolev descent: A step toward reinforcement learning with superlinear convergence properties. In *2022 International Conference on Robotics and Automation (ICRA)*, 01–07. IEEE.
- Plaat, A.; Kusters, W.; and Preuss, M. 2023. High-accuracy model-based reinforcement learning, a survey. *Artificial Intelligence Review*, 1–33.
- Radosavovic, I.; Xiao, T.; Zhang, B.; Darrell, T.; Malik, J.; and Sreenath, K. 2023. Learning Humanoid Locomotion with Transformers. *arXiv preprint arXiv:2303.03381*.
- Robine, J.; Höftmann, M.; Uelwer, T.; and Harmeling, S. 2023. Transformer-based World Models Are Happy With 100k Interactions. *arXiv preprint arXiv:2303.07109*.
- Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015. Trust region policy optimization. In *International conference on machine learning*, 1889–1897. PMLR.
- Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Sikchi, H.; Zhou, W.; and Held, D. 2022. Learning off-policy with online planning. In *Conference on Robot Learning*, 1622–1633. PMLR.
- Suh, H. J.; Simchowicz, M.; Zhang, K.; and Tedrake, R. 2022. Do differentiable simulators give better policy gradients? In *International Conference on Machine Learning*, 20668–20696. PMLR.
- Werling, K.; Omens, D.; Lee, J.; Exarchos, I.; and Liu, C. K. 2021. Fast and feature-complete differentiable physics engine for articulated rigid bodies with contact constraints. In *Robotics: Science and Systems*.
- Wiedemann, N.; Wüest, V.; Loquercio, A.; Müller, M.; Floreano, D.; and Scaramuzza, D. 2023. Training Efficient Controllers via Analytic Policy Gradient.
- Xian, Z.; Zhu, B.; Xu, Z.; Tung, H.-Y.; Torralba, A.; Fragkiadaki, K.; and Gan, C. 2023. Fluidlab: A differentiable environment for benchmarking complex fluid manipulation. *The Eleventh International Conference on Learning Representations*.
- Xu, J.; Makoviychuk, V.; Narang, Y.; Ramos, F.; Matusik, W.; Garg, A.; and Macklin, M. 2021. Accelerated Policy Learning with Parallel Differentiable Simulation. In *International Conference on Learning Representations*.
- Yu, T.; Thomas, G.; Yu, L.; Ermon, S.; Zou, J. Y.; Levine, S.; Finn, C.; and Ma, T. 2020. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33: 14129–14142.
- Zhong, Y. D.; Han, J.; and Brikis, G. O. 2022. Differentiable physics simulations with contacts: Do they have correct gradients wrt position, velocity and control? *arXiv preprint arXiv:2207.05060*.