Dual-Enhanced Model-Based Policy Optimization: Dynamic Bias-Shift Tradeoff and Adaptive Bidirectional Rollout

Yuetian Wang
Shanghai Jiao Tong University
Shanghai, China
echo_yt@sjtu.edu.cn

Dianxi Shi
Intelligent Game and Decision Lab
Beijing, China
dxshi@nudt.edu.cn

Huanhuan Yang
Beijing Academy of Science and Technology
Beijing, China
yanghh94@126.com

Yuanze Wang
Shanghai Jiao Tong University
Shanghai, China
yz.wang@sjtu.edu.cn

Shiming Song
Intelligent Game and Decision Lab
Beijing, China
shiming_js@foxmail.com

Chunping Qiu
Intelligent Game and Decision Lab
Beijing, China
chunping.qiu@aliyun.com

ABSTRACT

Model-based reinforcement learning (MBRL) achieves superior sample efficiency compared to model-free approaches, but its performance is fundamentally limited by both the accuracy of the learned dynamics model and the effectiveness of model utilization. Most existing MBRL methods optimize either model learning or model utilization in isolation, often resulting in compounding errors and suboptimal policy performance. In this paper, we propose Dual-Enhanced Model-Based Policy Optimization (DEMBPO), a unified framework that jointly optimizes model learning and utilization. DEMBPO introduces a dynamic bias-shift weighting mechanism and an adaptive bidirectional rollout strategy to simultaneously mitigate model bias and model shift while actively suppressing long-horizon error accumulation. Furthermore, we incorporate the Wasserstein distance as a principled performance metric, establishing formal theoretical guarantees for policy improvement. Extensive experiments on standard MuJoCo benchmarks demonstrate that DEMBPO consistently outperforms state-of-the-art MBRL methods in sample efficiency while achieving asymptotic performance on par with leading model-free approaches.

KEYWORDS

Model-Based Reinforcement Learning, Model bias and shift, Theoretical guarantees

ACM Reference Format:
Yuetian Wang, Dianxi Shi, Huanhuan Yang, Yuanze Wang, Shiming Song, and Chunping Qiu. 2026. Dual-Enhanced Model-Based Policy Optimization: Dynamic Bias-Shift Tradeoff and Adaptive Bidirectional Rollout. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/PHVU5630

1 INTRODUCTION

Recent advances in deep learning and computational infrastructure have driven substantial progress in reinforcement learning (RL), enabling breakthroughs in both academic research and real-world applications [4, 7, 8, 11]. A central challenge in RL is balancing asymptotic performance with sample efficiency, particularly in domains where environment interactions are costly or time-consuming. Broadly, RL algorithms fall into two paradigms: *model-free* (MFRL) and *model-based* reinforcement learning (MBRL) [24]. While MFRL methods have demonstrated strong asymptotic performance in complex, high-dimensional control tasks [6, 8], they often require extensive interaction with the environment, limiting their practicality in data-constrained settings [17].

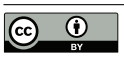
MBRL addresses this limitation by learning environment dynamics models to generate synthetic trajectories—enabling rapid policy improvement with fewer real interactions [12]. This makes MBRL attractive for real-world and safety-critical applications. However, its performance fundamentally hinges on the fidelity and stability of the learned model. Two key error sources—*model bias* (the gap between learned and true dynamics) and *model shift* (distributional drift during updates)—compound over long rollouts, resulting in inaccurate value estimates and degraded policies [20].

A major limitation of existing MBRL methods is the decoupling of model learning and utilization. Many approaches alternate between model fitting and policy optimization [3, 12], often neglecting the dynamic interplay of model bias and shift. Some attempt to correct for model error via return discrepancy [27], but overlook state transition inaccuracies. Others impose fixed update thresholds [13], sacrificing adaptability and robustness across training phases.

On the utilization side, most MBRL algorithms use static, fixed-length forward rollouts [12] or non-adaptive bidirectional simulation [16], which cannot respond to evolving model errors. Consequently, compounding prediction errors in long-horizon rollouts undermines the reliability of synthetic data and destabilizes policy learning.

To overcome these limitations, we propose *Dual-Enhanced Model-Based Policy Optimization* (DEMBPO), a unified MBRL framework that jointly optimizes model learning and utilization. DEMBPO enhances both model learning and utilization to enable robust, sample-efficient policy improvement. Our main innovations are as follows:

- **Theoretical guarantee:** We develop a principled objective based on the second-order Wasserstein distance, which unifies



Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/PHVU5630

model bias and shift in a single framework and yields a formal lower bound on policy improvement.

- **Dynamic joint constraint:** Unlike USB-PO [27], which uses fixed weights, DEMBPO employs an adaptive weighting mechanism that dynamically balances model bias and model shift throughout training, enabling stable, high-fidelity environment modeling.

- **Adaptive bidirectional rollout:** We introduce a flexible rollout mechanism that dynamically chooses trajectory direction and length according to error accumulation, significantly mitigating long-horizon error propagation.

Extensive experiments on five MuJoCo continuous control tasks show that DEMBPO achieves superior sample efficiency and final performance, matching or exceeding the best existing model-based and model-free methods. Overall, DEMBPO provides a unified, theoretically principled, and practically robust solution for model-based RL, bridging the gap between sample efficiency and asymptotic performance.

2 RELATED WORK

Model-based reinforcement learning (MBRL) has been extensively studied due to its potential for high sample efficiency in sequential decision-making [12, 24]. Classical frameworks such as Dyna [25] first leveraged learned dynamics models to generate synthetic data, significantly reducing environment interactions required for policy improvement. Building on this foundation, many recent MBRL approaches have integrated advanced model-free RL algorithms—including SAC [8] and TD3 [6]—as policy optimizers, further improving empirical performance.

A persistent challenge for MBRL is providing theoretical guarantees for policy improvement. While model-free methods such as TRPO [23] and conservative policy iteration (CPI) [14] deliver monotonic improvement, most MBRL algorithms focus on minimizing "return discrepancy" [12, 26], which often ignores the impact of model shift across iterations. To address this, CMLO [13] imposes a fixed threshold on model updates to control policy divergence, but such static regularization lacks adaptability and can lead to suboptimal performance as training progresses. USB-PO [27] takes a step further by jointly regularizing model bias and shift, but relies on manually chosen weights, limiting its flexibility under diverse learning dynamics.

Recent work highlights *primacy bias* in model-based RL, where learning can be overly shaped by early data and early model errors [21]. This is especially relevant in online MBRL with non-stationary data. DEMBPO addresses this by explicitly regularizing both bias and shift during model updates, using dynamic weighting to avoid a fixed early-stage tradeoff, and adapting bidirectional rollouts to reduce harmful error accumulation when the model is unreliable.

On the model utilization side, a key limitation in MBRL is the compounding error that arises from long-horizon rollouts with imperfect models [1]. Methods such as model ensembles [3, 15, 22] and uncertainty-aware rollout truncation [26] have been proposed to alleviate this issue. BMPO [16] explores bidirectional rollouts, allowing the model to generate samples from both forward and backward perspectives, but uses static rollout directions that do not adapt to error accumulation during training.

Confidence-aware bidirectional imagination in offline MBRL [18] also emphasizes not trusting unreliable model states; our online approach differs but shares the motivation of coupling bidirectional rollouts with error-aware control.

Generative data augmentation methods aim to expand coverage or reshape the data distribution by synthesizing additional transitions/trajectories. This is complementary to our focus on controlling the bias-shift tradeoff in model updates and the error accumulation in rollout usage; thus, generative augmentation and DEMBPO are largely orthogonal and potentially composable.

In parallel, recent advances in latent dynamics and sequence modeling, such as Dreamer [9, 10] and Transformer-based MBRL [2, 5], have demonstrated strong sample efficiency and asymptotic performance, especially in high-dimensional or pixel-based environments. However, these methods predominantly rely on empirical heuristics and still lack explicit theoretical guarantees for policy improvement or error control.

Despite these advances, few prior works offer a unified, adaptive framework that addresses both the theoretical bias-shift tradeoff in model learning and the practical challenge of error accumulation in model utilization. Our work bridges this gap by introducing a theoretically grounded dynamic weighting mechanism, together with adaptive bidirectional rollouts, resulting in robust and efficient MBRL across diverse tasks.

3 PRELIMINARIES

Consider a Markov Decision Process (MDP) defined by the tuple $M = \langle \mathcal{S}, \mathcal{A}, p, r, \gamma, \rho_0 \rangle$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $\gamma \in (0, 1)$ is the discount factor, $p(s'|s, a)$ represents the transition dynamics, and $r(s, a)$ is the reward function. We denote $p_{M^*}(s'|s, a)$ as the true environment dynamics with initial state distribution $\rho_0(s)$.

The optimal policy π^* maximizes the expected return under the true environment M^*:

$$\pi^* = \arg\max_{\pi} V^{\pi}_{M^*}$$
$$\text{where } V^{\pi}_{M^*} = \mathbb{E}_{\substack{a_t \sim \pi(\cdot|s_t) \\ s_{t+1} \sim p_{M^*}(\cdot|s_t, a_t) \\ s_0 \sim \rho_0(s)}} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 \right], \tag{1}$$

MBRL algorithms learn an approximate transition model $p_M(s'|s, a)$. We define:

V^{π}_M: Expected return of policy π under model M.

$V^{\pi|M}$: Expected return of policy π derived from model M in the real environment.

r_M: Model-predicted reward function.

Let $\mathcal{M}$ and Π denote parameterized families of models and policies. The normalized discounted state-action distribution under model M and policy π is:

$$d^{\pi}_M(s, a) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t p^{\pi}_{M,t}(s) \pi(a|s), \tag{2}$$

where $p^{\pi}_{M,t}(s) = \mathbb{P}(s_t = s \mid \pi, M)$ is the state distribution at timestep t.

4 DEMBPO: DUAL-ENHANCED MODEL-BASED POLICY OPTIMIZATION

This section presents our *Dual-Enhanced Model-Based Policy Optimization* (DEMBPO), a unified framework for model-based reinforcement learning that jointly optimizes model generation and model utilization. Unlike conventional MBRL methods, which decouple model training from policy optimization—often leading to compounding prediction errors and unstable learning—DEMBPO integrates both components into a single, theory-guided pipeline. Our approach enables dual enhancement: improving model fidelity through adaptive error control, and enhancing data utility via intelligent rollout strategies.

Specifically, Section 4.1 outlines the overall architecture, emphasizing the integration of model learning and policy updates within a coherent optimization loop. In Section 4.2, we introduce the second-order Wasserstein distance as a principled performance metric, unifying model bias and model shift into a composite error framework. This analysis yields a formal lower bound on policy improvement, providing theoretical guarantees absent in most existing MBRL methods. Building on this foundation, Section 4.3 proposes a dynamic weighting mechanism that adaptively balances bias reduction and shift regularization over training, implemented without hand-tuned schedules. Finally, Section 4.4 introduces an adaptive bidirectional rollout strategy: by learning both forward and backward dynamics models, DEMBPO dynamically selects the trajectory direction with minimal predicted error accumulation, effectively mitigating long-horizon inaccuracies.

Together, these components form a complete MBRL framework that bridges theoretical rigor and practical efficiency, achieving robust and sample-efficient policy optimization through synergistic model design.

The algorithm's pseudocode, main code and detailed derivation process are provided in the supplementary material.

4.1 The Overall Algorithm

Figure 1 illustrates the DEMBPO framework. Unlike conventional MBRL methods that decouple model learning from policy optimization—often leading to compounding errors and unstable training—DEMBPO unifies both components into a single, synergistic pipeline. Our approach enables dual enhancement: improving model generation through adaptive error control, and enhancing model utilization via intelligent rollout strategies.

We adopt a two-stage modeling strategy to balance model fidelity and stability. In the first stage, we fit the dynamics model via maximum likelihood estimation (MLE), minimizing prediction error to reduce model bias. However, aggressive bias reduction alone can induce large distributional shifts when the policy updates, destabilizing value estimation and degrading performance. To address this, in the second stage, we introduce a joint optimization objective that simultaneously reduces bias and constrains model update magnitude:

$$p^*_{M_2} = \arg\min_{p_{M_2}} \mathbb{E}_{d^{\pi}_{M_1}} \Big[\lambda(t) \cdot W_2(p_{M_2}, p_{M^*}) + (1-\lambda(t)) \cdot W_2(p_{M_1}, p_{M_2}) \Big] \tag{3}$$

This objective uses the second-order Wasserstein distance W_2 to jointly measure model bias (between the updated model M_2 and the true environment M^*) and model shift (between M_2 and the previous model M_1), quantified via their state-action distributions under the current policy. The weighting coefficient $\lambda(t)$ adapts throughout training to balance these two terms: high in early stages to prioritize model accuracy, and low in later stages to suppress excessive updates, thereby enforcing stable and reliable model learning.

In the model utilization phase, we further enhance the quality and reliability of synthetic data. Inspired by BMPO [16], DEMBPO learns both a forward dynamics model and a backward transition model, each paired with its corresponding policy network. During virtual trajectory generation, the algorithm adaptively selects between forward and backward rollout paths based on estimated cumulative prediction error, favoring the direction with slower error growth. This mechanism effectively mitigates long-horizon error propagation and significantly improves trajectory fidelity.

Finally, DEMBPO maintains a highly modular design in the policy optimization stage, enabling seamless integration with advanced model-free reinforcement learning (MFRL) algorithms. In this work, we adopt Soft Actor-Critic (SAC) [8] as the policy optimizer within the MBPO [12] framework. MBPO [12] provides a principled and stable foundation for model-based policy updates, balancing sample efficiency and performance, making it an ideal host for our dual-enhancement mechanisms.

4.2 Theoretical Analysis

In model-based reinforcement learning (MBRL), policy improvement hinges on both the accuracy and stability of the learned dynamics model. Yet conventional methods typically focus only on minimizing prediction error—i.e., model bias—while overlooking the distributional shift induced by iterative model updates under evolving policies. This shift can destabilize value estimation and lead to compounding errors in long-horizon rollouts, even when the model is accurate on past data.

To establish a verifiable performance guarantee, we introduce the Wasserstein distance as a unified metric that jointly captures model bias and model shift. Building on this foundation, we derive a lower bound on policy improvement that explicitly depends on a composite error—the sum of expected bias and shift under the current policy distribution. This theoretical result provides a principled basis for algorithm design, linking model learning directly to reliable policy optimization.

Wasserstein distance as a principled metric. The Wasserstein distance, also known as the Earth Mover's Distance, provides a geometrically meaningful measure between probability distributions even when their supports are non-overlapping—a critical advantage over traditional metrics like KL divergence. Formally:

Definition 4.1 (First-order Wasserstein Distance). Given a metric space $(\mathcal{S}, d)$, the first-order Wasserstein distance between two probability distributions $\mu_1, \mu_2 \in \mathbb{P}(\mathcal{S})$ is defined as:

$$W_1(\mu_1, \mu_2) := \inf_{j \in \Pi(\mu_1, \mu_2)} \mathbb{E}_{(s_1, s_2) \sim j}[d(s_1, s_2)] \tag{4}$$

where $\Pi(\mu_1, \mu_2)$ denotes the set of all joint distributions with marginals μ_1 and μ_2.

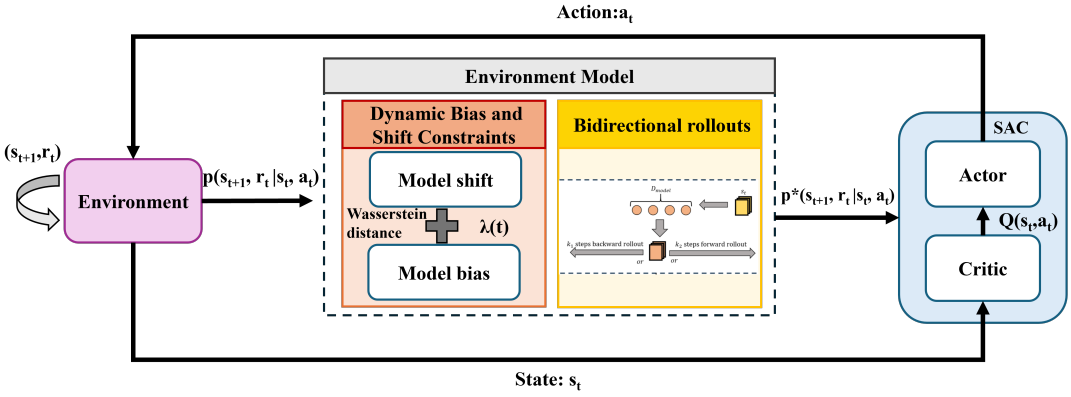


Figure 1: The DEMBPO framework. DEMBPO unifies model learning and utilization within a single optimization pipeline. During model generation, we implement a two-stage modeling strategy with dynamic bias-shift balancing. During model utilization, our adaptive bidirectional rollout mechanism selectively generates trajectories from forward and backward models based on error accumulation.

This geometric property makes Wasserstein distance particularly suitable for MBRL, where state distributions often lie on low-dimensional manifolds. For Gaussian distributions $p \sim \mathcal{N}(\mu_1, \Sigma_1)$ and $q \sim \mathcal{N}(\mu_2, \Sigma_2)$, the second-order Wasserstein distance has a closed-form solution:

$$W_2(p,q) = \sqrt{\|\mu_1 - \mu_2\|_2^2 + \operatorname{tr}\Sigma_1 + \operatorname{tr}\Sigma_2 - 2\operatorname{tr}\left(\Sigma_2^{\frac{1}{2}}\Sigma_1\Sigma_2^{\frac{1}{2}}\right)^{\frac{1}{2}}} \tag{5}$$

This closed-form expression is crucial for practical implementation, as it allows direct computation from model predictions.

Lipschitz continuity assumption. We note that if the value function is L-Lipschitz continuous with respect to the state metric (a property satisfied by many neural approximators and entropy-regularized policies), then the integral probability metric (IPM) between transition dynamics can be directly bounded by the Wasserstein distance:

Lemma 4.2 (IPM-Wasserstein Equivalence). *For any two models $M, M' \in \mathcal{M}$, the IPM between their transition dynamics satisfies:*

$$\sup_{f \in \mathcal{F}_L} \left|\mathbb{E}_{s' \sim p_M}[f(s')] - \mathbb{E}_{s' \sim p_{M'}}[f(s')]\right| = L \cdot W_1(p_M, p_{M'}) \tag{6}$$

where $\mathcal{F}_L$ denotes the set of all L-Lipschitz functions.

Performance improvement guarantee. Our theoretical framework centers on establishing a lower bound for policy improvement in the true environment. We define the performance difference between consecutive iterations as:

Definition 4.3 (Performance Difference Bound). Let $V^{\pi|M}$ denote the expected return of policy π derived from model M in the true environment M^*. A non-negative lower bound $C \geq 0$ on the performance difference:

$$V^{\pi_2|M_2} - V^{\pi_1|M_1} \geq C \tag{7}$$

guarantees monotonic policy improvement when $C > 0$.

To derive this bound, we decompose the performance difference into three components:

Theorem 4.4 (Performance Difference Decomposition). *For models $M_1, M_2 \in \mathcal{M}$ and corresponding policies $\pi_1, \pi_2 \in \Pi$:*

$$V^{\pi_2|M_2} - V^{\pi_1|M_1} = \underbrace{(V^{\pi_2|M_2} - V^{\pi_2}_{M_2})}_{\text{Model error at } \pi_2} - \underbrace{(V^{\pi_1|M_1} - V^{\pi_1}_{M_1})}_{\text{Model error at } \pi_1} + \underbrace{(V^{\pi_2}_{M_2} - V^{\pi_1}_{M_1})}_{\text{Model improvement}} \tag{8}$$

The first two terms represent model error at different policy iterations, while the third term captures improvement within the model. We now bound the third term:

Theorem 4.5 (Model Return Bound). *Suppose the value function V^{π}_{M} is L-Lipschitz continuous with respect to the state metric $d(\cdot,\cdot)$. Then for any policies π_1, π_2 and models M_1, M_2,*

$$V^{\pi_2}_{M_2} - V^{\pi_1}_{M_1} \geq -\frac{2L}{1-\gamma}\left(\max_s W_1(\pi_1(s), \pi_2(s)) + \frac{\gamma}{1-\gamma}\mathbb{E}_{(s,a)\sim d^{\pi_1}_{M_1}}[W_1(p_{M_1}, p_{M_2})]\right) \tag{9}$$

Combining these results, we derive the complete performance bound:

Theorem 4.6 (Composite Error Performance Bound). *Let $\epsilon^{\pi_i}_{M_i} = \mathbb{E}_{(s,a)\sim d^{\pi_i}_{M_i}}[W_1(p_{M_i}, p_{M^*})]$ denote the model bias and $\epsilon^{M_2}_{M_1} = \mathbb{E}_{(s,a)\sim d^{\pi_1}_{M_1}}[W_1(p_{M_1}, p_{M_2})]$ denote the model shift. Then:*

$$V^{\pi_2|M_2} - V^{\pi_1|M_1} \geq \frac{2L\gamma}{1-\gamma}\left(\epsilon^{\pi_1}_{M_1} - \epsilon^{\pi_2}_{M_2} - \epsilon^{M_2}_{M_1} - \frac{1}{\gamma}\max_s W_1(\pi_1(s), \pi_2(s))\right) \tag{10}$$

Practical implications. Equation 10 reveals that performance improvement depends on three critical factors: 1. Reduction in

model bias ($\epsilon_{M_1}^{\pi_1} - \epsilon_{M_2}^{\pi_2}$) 2. Controlled model shift ($-\epsilon_{M_1}^{M_2}$) 3. Small policy updates ($\max_s W_1(\pi_1(s), \pi_2(s))$)

Notably, the cross-term $\mathbb{E}_{d_{M_1}^{\pi_1}}[W_1(p_{M_2}, p_{M^*})] - \mathbb{E}_{d_{M_2}^{\pi_2}}[W_1(p_{M_2}, p_{M^*})]$ is of higher order and can be safely ignored under standard assumptions that policy updates are sufficiently small [12]. Returning to our objective, to ensure a performance improvement guarantee, it is imperative to maximize the term C. This implies that we should aim to minimize $W_1(p_{M_1}, p_{M_2}) + W_1(p_{M_2}, p_{M_*})$. It is important to highlight that for $1 \le p \le q$, the Wasserstein distance $W_p(p_M, p_{M'}) \le W_q(p_M, p_{M'})$ [19]. This implies that $W_1(p_M, p_{M'}) \le W_2(p_M, p_{M'})$.

This leads to our key insight:

Corollary 4.7 (Optimization Objective). *To guarantee policy improvement, we should minimize the composite error:*

$$\begin{aligned}\mathcal{E}_{composite} =& \lambda \cdot \mathbb{E}_{d_{M_1}^{\pi}}[W_2(p_{M_2}, p_{M^*})] \\ &+ (1-\lambda) \cdot \mathbb{E}_{d_{M_1}^{\pi}}[W_2(p_{M_1}, p_{M_2})]\end{aligned} \tag{11}$$

where $\lambda \in [0,1]$ balances model bias and model shift.

This composite error formulation directly informs our algorithm design in the following two sections, establishing a theoretically grounded connection between model learning and policy improvement. All proofs and detailed derivations are deferred to the supplementary material.

4.3 Dynamic Bias and Shift Constraints

This section focuses on the **model learning** component of MBRL. Building on the preceding theoretical analysis, we observe that the policy improvement bound in MBRL depends on both model bias and model shift, whose relative importance changes throughout training. Early on, severe underfitting makes bias the main bottleneck as the model struggles to capture environment dynamics. As the model improves, large updates can cause model shift, degrading performance by disrupting previously learned dynamics. This evolving balance of error poses a challenge: static bias-shift trade-offs (e.g., USB-PO [27]) cannot jointly optimize fast early learning and stable late-stage performance, often leading to suboptimal results.

To address this limitation, we introduce a dynamic weighting mechanism that adaptively balances model bias and model shift throughout training. Building upon the joint optimization objective in Equation 3, we parameterize $\lambda(t)$ as an exponential decay function:

$$\lambda(t) = \lambda_{\max} \cdot e^{-kt} + \lambda_{\min} \cdot (1 - e^{-kt}) \tag{12}$$

where $\lambda_{\max} = 0.8$, $\lambda_{\min} = 0.2$, and $k > 0$ controls the decay rate (empirically set to $k = 0.005$). This functional form was selected for its smooth transition properties and alignment with the natural progression of model learning.

This design embodies clear phase-dependent semantics with theoretically grounded parameter choices:

Early training ($t \to 0$): $\lambda(t) \to \lambda_{\max}$, prioritizing bias reduction through larger parameter updates to rapidly fit environment dynamics. This aggressive phase enables the model to quickly overcome initial underfitting, capturing the most salient features of environment dynamics.

Late training ($t \to \infty$): $\lambda(t) \to \lambda_{\min}$, emphasizing shift regularization by effectively tightening the "trust region" to preserve high-performing solutions. This conservative phase prevents destructive model updates that would otherwise degrade previously learned accurate dynamics.

The specific values of $\lambda_{\max}$ and $\lambda_{\min}$ were determined through theoretical analysis of the performance bound (Equation 10) and empirical validation across multiple environments. Setting $\lambda_{\max} = 0.8$ ensures sufficient emphasis on bias reduction without completely ignoring model shift even in early stages, while $\lambda_{\min} = 0.2$ maintains some focus on continued model refinement even during stabilization.

Integrating this schedule into our theoretical framework, the performance improvement bound extends to:

$$\begin{aligned}V^{\pi_2|M_2} - V^{\pi_1|M_1} \ge& \frac{2L\gamma}{1-\gamma}\Big(\lambda(t) \cdot (\epsilon_{M_1}^{\pi_1} - \epsilon_{M_2}^{\pi_2}) \\ &- (1-\lambda(t)) \cdot \epsilon_{M_1}^{M_2}\Big) - O(\Delta\pi)\end{aligned} \tag{13}$$

where $O(\Delta\pi)$ represents policy difference terms as in Equation 10.

Theoretically, as long as $\lambda(t)$ varies smoothly and converges to a stable value, our framework preserves fundamental convergence properties while adapting to training dynamics. This mechanism implements an adaptive error suppression strategy with practical implementation considerations:

When model bias dominates ($\epsilon_{M_1}^{\pi_1} \gg \epsilon_{M_1}^{M_2}$), high $\lambda(t)$ encourages exploration to rapidly improve model accuracy through larger gradient steps

When model shift becomes critical ($\epsilon_{M_1}^{M_2} \gg \epsilon_{M_1}^{\pi_1}$), low $\lambda(t)$ enhances stability by constraining update magnitudes through smaller effective learning rates

This "coarse-to-fine" learning strategy effectively mitigates the problem of single-error dominance by dynamically allocating optimization resources to the most pressing error source at each training stage. The resulting training dynamics ensure model learning remains both efficient and robust throughout the entire training trajectory, with empirical evidence showing faster convergence to high-fidelity models and more stable policy learning compared to static weighting approaches.

4.4 Adaptive Bidirectional Rollout

We introduce an adaptive bidirectional rollout mechanism as the core of DEMBPO's **model utilization**. Unlike conventional MBRL methods that tolerate error accumulation in long rollouts, DEMBPO actively selects the most reliable trajectory direction, substantially improving the quality of synthetic data for policy optimization.

Error accumulation challenge. Even high-fidelity models can yield minor prediction errors that compound over multi-step rollouts, ultimately producing trajectories that deviate from true dynamics. Previous approaches rely on either single forward rollouts (MBPO [12]) or fixed-length bidirectional rollouts (BMPO [16]), without dynamic adaptation to changing error profiles.

Forward and backward models. DEMBPO employs both forward and backward dynamics models within a probabilistic ensemble. The forward model $p_M(s'|s,a)$ is parameterized as a Gaussian:

$$p_M(s'|s,a) = \mathcal{N}(\mu_M(s,a), \Sigma_M(s,a)) \tag{14}$$

where the diagonal covariance $\Sigma_M(s,a)$ captures transition uncertainty. This ensemble models both aleatoric and epistemic uncertainties [3], leading to improved performance as shown in [12].

The forward model is trained via maximum likelihood estimation (MLE) on real environment data:

$$\mathcal{L}_M(\theta) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\Big[\|\mu_M(s,a) - s'\|^2_{\Sigma_M^{-1}} + \log\det\Sigma_M(s,a)\Big] \tag{15}$$

Similarly, we construct a backward dynamics model $p_{M^{-1}}(s|s',a)$ with identical architecture.

Backward rollouts require generating plausible actions without policy guidance. DEMBPO addresses this by introducing a reverse policy network $\pi_{M^{-1}}(a|s')$, which infers actions from future states and is trained via maximum likelihood on real trajectories:

$$\mathcal{L}_\pi(\phi) = -\mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[\log\pi_{M^{-1}}(a|s')\right] \tag{16}$$

Adaptive rollout strategy. The key innovation of our approach lies in dynamically selecting between forward and backward rollout paths based on estimated cumulative prediction error. While theoretically optimal to compute error accumulation directly (e.g., via negative log-likelihood or Wasserstein distance), we propose a computationally efficient approximation: using rollout length as a proxy for error accumulation.

Given a maximum rollout length H, we allow forward and backward paths of lengths k_1 and k_2 with $k_1 + k_2 \le H$, and select the shorter path as the output trajectory. This is justified by the error-length correlation: in probabilistic models, prediction error grows exponentially with rollout steps, so shorter rollouts are more reliable due to less compounded uncertainty in both directions. Moreover, Theorem 4.8 shows that the return deviation bound depends on $\min(k_1,k_2)$, meaning the minimum rollout length directly determines the error upper bound and selecting the shorter path effectively approximates the optimal error-minimizing strategy.

Theoretical advantage. To quantify our strategy's superiority, we analyze the return deviation bound under branched rollouts:

Theorem 4.8 (Branch Return Bound). *Let $\eta[\pi]$ denote the expected return of policy π in the true environment M^*, and $\eta^{branch}[\pi]$ the return under branched rollouts. Define ϵ_m as the model error upper bound and ϵ_π as the policy difference upper bound. Under rollout lengths k_1 (forward) and k_2 (backward), the return deviation satisfies:*

$$|\eta[\pi] - \eta^{branch}[\pi]| \le \frac{2r_{\max}}{1-\gamma}\left[\frac{\gamma^{k_1+k_2+1}\epsilon_\pi}{1-\gamma} + \min(k_1,k_2)\cdot\epsilon_m\right] \tag{17}$$

The dominant term $\min(k_1,k_2)\cdot\epsilon_m$ reveals the fundamental advantage of our approach: - Single-direction rollout (e.g., MBPO): error accumulation $\propto H\cdot\epsilon_m$ - Fixed bidirectional rollout (e.g., BMPO): error accumulation $\propto \frac{H}{2}\cdot\epsilon_m$ - Adaptive bidirectional rollout (DEMBPO): error accumulation $\propto \min(k_1,k_2)\cdot\epsilon_m$

This demonstrates that DEMBPO reduces error accumulation to the theoretical minimum by adaptively selecting the path with slower error growth. The mechanism effectively transforms model utilization from passive error acceptance to active error mitigation, significantly enhancing the reliability of synthetic data and the stability of policy learning. [1]

5 EXPERIMENTS

We evaluate DEMBPO across five MuJoCo continuous control benchmarks (Walker2d, Ant, Humanoid, HalfCheetah, and Hopper) employing the standard 1000-step setup with uniform environment configurations. Our experimental protocol systematically addresses three core questions:

(1) Does DEMBPO achieve state-of-the-art sample efficiency and asymptotic performance compared to MBRL baselines?
(2) How do the dual-enhancement modules—composite error optimization and adaptive bidirectional rollouts—individually and jointly impact performance?
(3) To what extent does the dynamic weighting mechanism resolve the bias-shift tradeoff as predicted by our theoretical framework?

5.1 Comparison with State-of-the-Art Methods

We compare DEMBPO with strong model-based and model-free baselines that incorporate principled mechanisms for controlling model error and/or improving rollout usage. Our goal is to evaluate performance under matched environment-interaction budgets (Fig. 2), while also reporting SAC trained for a longer budget as a ceiling reference.

MBRL baselines. For model acquisition, we include USB-PO [27], which balances model bias and model shift via a fixed weighting, and CMLO [13], which uses environment-specific constraints to limit harmful model shift. For model utilization, we include BMPO [16], which generates short-horizon bidirectional rollouts using both forward and backward dynamics models. We also compare against MBPO [12], which serves as the underlying branched-rollout framework that our method builds upon.

MFRL baseline. We report SAC [8]. In Fig. 2, the solid SAC curve corresponds to training SAC under the same interaction budget as the model-based methods, while the dashed line indicates SAC's converged performance when trained for 3M steps (ceiling reference).

Rollout settings. For MBPO, we follow the default configuration in the original paper and generate only forward rollouts with a fixed horizon. For bidirectional methods, the forward/backward rollout lengths are sampled within a predefined range following prior work [13].

Figure 2 summarizes learning curves across tasks. Overall, DEMBPO tends to learn faster than the compared MBRL methods and achieves competitive final performance within the same interaction budgets. The gains over the strongest baselines (e.g., USB-PO) are task-dependent: in several environments DEMBPO provides clear improvements, while in others the final performance is comparable. When compared to SAC, DEMBPO substantially improves sample efficiency relative to SAC trained under the same limited budget, and its final performance is generally close to the SAC ceiling on some tasks but can remain below it on others.

[1] Appendices

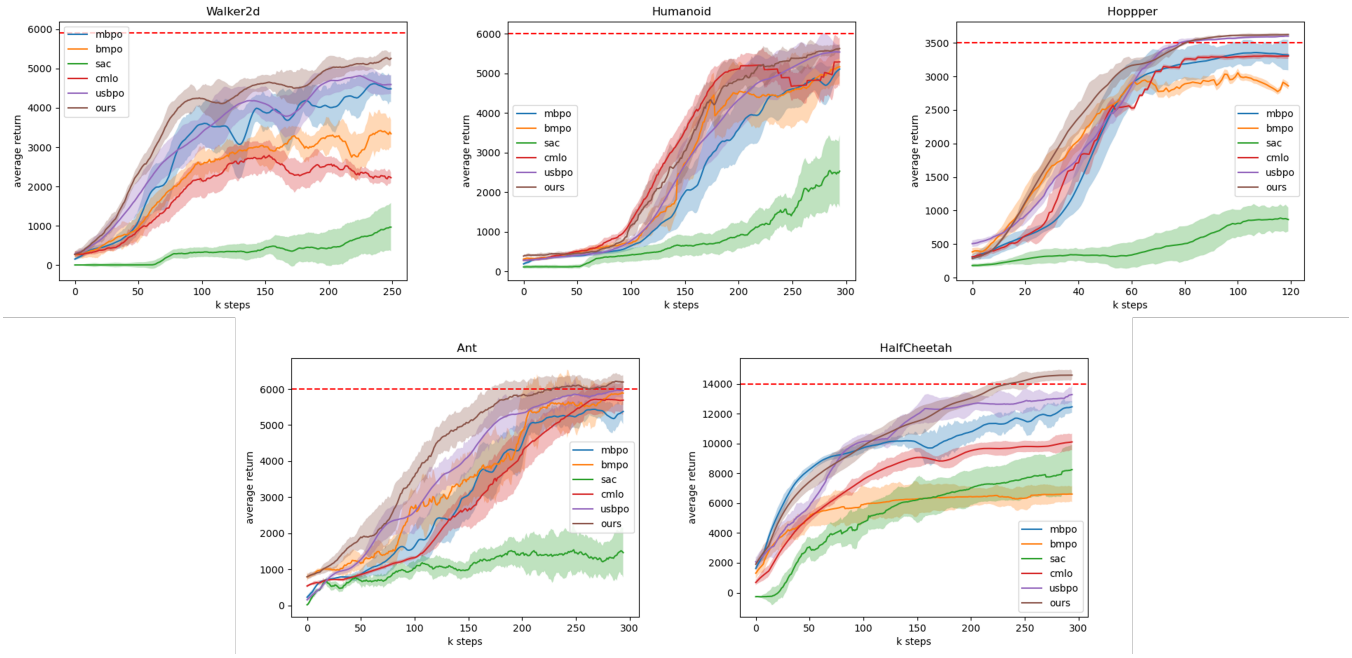


Figure 2: Comparison against baselines on continuous control benchmarks. Solid curves refer to the mean performance of trials over different random seeds, and the shaded area refers to the standard deviation of these trials. Dashed lines refer to the asymptotic performance of SAC (at 3M steps).

Sample efficiency. Across tasks, DEMBPO typically reaches strong returns earlier than the MBRL baselines in Fig. 2, especially on Walker2d, Ant, and HalfCheetah.

Final performance. Under the matched MBRL budgets, DEMBPO achieves consistently strong final returns and is competitive with the best-performing MBRL baselines. Relative to the SAC ceiling (3M steps), DEMBPO is close on some tasks, though it does not uniformly match SAC across all environments.

Stability. We observe that some baselines show larger variability and occasional oscillations on challenging tasks (e.g., Humanoid), whereas DEMBPO generally exhibits smoother learning curves. BMPO can be competitive on simpler tasks but may lag behind on complex environments, suggesting that bidirectional rollouts benefit from being coupled with explicit error control.

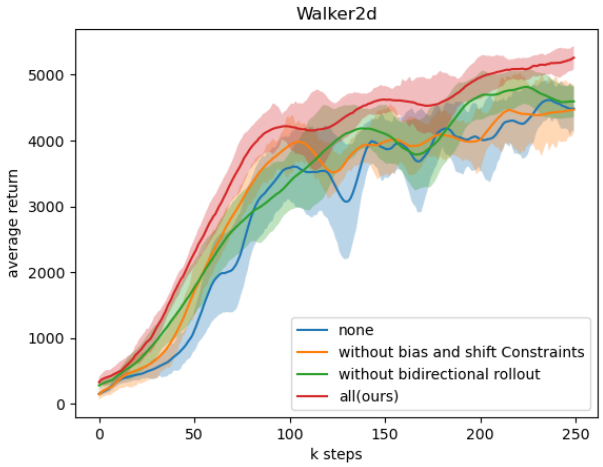


Figure 3: Ablation study showing the necessity of dual-enhancement modules.

5.2 Ablation Studies

We analyze the contribution of the two enhancements in DEMBPO by evaluating four variants:

None: Standard MBPO with return discrepancy. This variant uses neither the composite bias-shift constraints during model acquisition nor adaptive bidirectional rollouts during model utilization.

Without bias/shift constraints: A variant that uses adaptive bidirectional rollouts but removes the composite bias-shift constraints in model learning.

Without bidirectional rollouts: A variant that retains the composite bias-shift constraints in model learning but uses forward-only rollouts (the same rollout setting as MBPO) during policy optimization.

All (Ours): Full DEMBPO implementation.

Figure 3 shows that removing either enhancement leads to a clear degradation on Walker2d, suggesting that the two components are complementary in practice.

The *Without bidirectional rollouts* variant (forward-only rollout) underperforms the full method, which indicates that incorporating backward rollouts can provide additional useful synthetic transitions and improve learning when the learned dynamics is sufficiently reliable.

The *Without bias/shift constraints* variant also falls short of the full method, suggesting that explicit control of model error is particularly important when training the policy with bidirectional model-generated data.

Overall, the full DEMBPO achieves the most favorable learning curve in this task, consistent with our design goal of coupling constrained model learning with bidirectional rollouts.

5.3 Dynamic Weighting Analysis

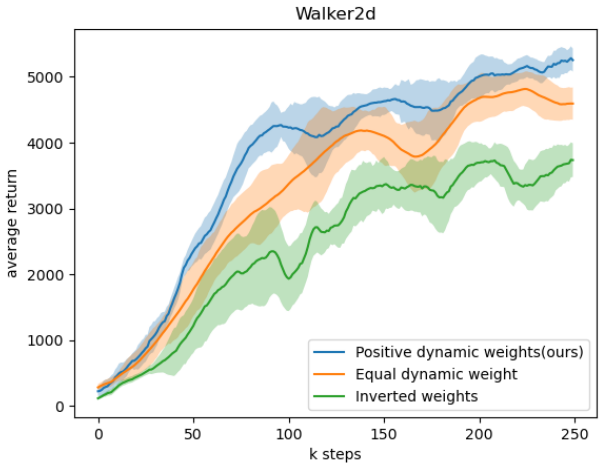


Figure 4: Dynamic weighting strategy comparison. Our temporally adaptive approach outperforms static alternatives by balancing exploration and stability.

To validate the effectiveness of the dynamic weighting mechanism, we compare three different weighting schemes, and compares three weighting strategies on Walker2d:

Positive dynamic weights (Ours): $\lambda(t)$ decays exponentially from 0.8 to 0.2, emphasizing bias early and stability later.

Equal dynamic weight: $\lambda(t)$ is fixed at 0.5, representing a simplified version of USB-PO[27] that applies equal weighting to bias and shift.

Inverted weights: $\lambda(t)$ increasing from 0.2 to 0.8, where the weight schedule is reversed, representing a "stable-first" approach.

Empirical results align with theoretical predictions: Figure 4 shows that the choice of schedule can noticeably affect learning. The decaying schedule yields the strongest curve among the three in this task, while the fixed and inverted schedules are less effective under the same interaction budget. A plausible explanation is that the relative importance of controlling model bias and model shift changes over training: placing more emphasis on bias early can help learning when the model is still inaccurate, whereas gradually increasing emphasis on shift later can improve stability as rollouts become longer and policy updates become more sensitive to distribution shift. We view these results as supportive evidence that a time-varying schedule can be beneficial, although the best schedule may depend on the task and training budget.

5.4 Summary of Experimental Findings

Overall, our experiments provide consistent evidence that DEMBPO is a strong online MBRL approach on standard MuJoCo benchmarks: **Sample efficiency**: DEMBPO often reaches strong returns earlier than MBPO and other MBRL baselines, indicating improved learning speed under the same interaction budgets. **Training stability**: On challenging environments (e.g., Humanoid), DEMBPO tends to exhibit smoother learning curves and fewer pronounced oscillations than several baselines, aligning with the intended effect of dynamically balancing bias and shift during training. **Final performance**: DEMBPO achieves competitive end-of-budget returns across tasks and, on some environments, approaches the long-budget SAC ceiling, while still operating in the sample-efficient MBRL regime.

The ablation results show that both composite error optimization and adaptive bidirectional rollouts contribute to the gains, and the dynamic weighting analysis further indicates that the schedule choice can materially affect learning behavior. Taken together, these results support DEMBPO as a practically effective and theoretically motivated framework that improves the reliability of online model-based policy optimization. Detailed settings and additional results are provided in the supplementary material.

6 CONCLUSION

In this work, we introduced DEMBPO, a principled model-based reinforcement learning framework that systematically tackles both model learning and model utilization challenges in MBRL. By dynamically balancing model shift and model bias via adaptive weighting and bidirectional rollouts, DEMBPO effectively mitigates error accumulation and delivers robust policy improvement. Our incorporation of the Wasserstein distance not only grounds policy optimization in solid theory but also strengthens the robustness of model learning.

Extensive experiments on standard continuous control benchmarks demonstrate that DEMBPO consistently achieves superior sample efficiency and asymptotic performance, outperforming existing MBRL baselines and approaching or even exceeding the performance of model-free RL in several tasks. These findings validate our dual-optimization strategy and underline the necessity of coordinated model learning and utilization for scalable and reliable MBRL.

Looking forward, we plan to extend the adaptive bidirectional modeling paradigm to broader MBRL settings, deepen theoretical understanding of bias and shift dynamics, and further explore principled strategies for robust and efficient model-based policy optimization.

ACKNOWLEDGMENTS

This work was jointly supported by the National Natural Science Foundation of China (No. 42201513), and China Postdoctoral Science Foundation (No. 2022M723902 and 2023T160789).

REFERENCES

[1] Kavosh Asadi, Dipendra Misra, Seungchan Kim, and Michel L Littman. 2019. Combating the compounding-error problem with a multi-step model. *arXiv preprint arXiv:1905.13320* (2019).

[2] Annie S. Chen, Archit Sharma, Sergey Levine, and Chelsea Finn. 2022. You Only Live Once: Single-Life Reinforcement Learning. In *Advances in Neural Information Processing Systems*. arXiv:2210.08863 https://proceedings.neurips.cc/paper_files/paper/2022/file/5ec4e93f2cec19d47ef852a0e1fb2c48-Paper-Conference.pdf

[3] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. 2018. Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models. In *Advances in Neural Information Processing Systems*. arXiv:1805.12114 https://proceedings.neurips.cc/paper_files/paper/2018/file/3de568f8597b94bda53149c7d7f5958c-Paper.pdf

[4] Oguzhan Dogru, Junyao Xie, Om Prakash, Ranjith Chiplunkar, Jansen Soesanto, Hongtian Chen, Kirubakaran Velswamy, Fadi Ibrahim, and Biao Huang. 2024. Reinforcement learning in process industries: Review and perspective. *IEEE/CAA Journal of Automatica Sinica* 11, 2 (2024), 283–300.

[5] Daniel Freeman, David Ha, and Luke Metz. 2019. Learning to predict without looking ahead: World models without forward prediction. *Advances in Neural Information Processing Systems* 32 (2019).

[6] Scott Fujimoto, Herke Hoof, and David Meger. 2018. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*. PMLR, 1587–1596.

[7] Chengqian Gao, Ke Xu, Kuangqi Zhou, Lanqing Li, Xueqian Wang, Bo Yuan, and Peilin Zhao. 2022. Value Penalized Q-Learning for Recommender Systems. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. arXiv:2110.07923 https://arxiv.org/abs/2110.07923

[8] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*. PMLR, 1861–1870.

[9] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. 2019. Learning latent dynamics for planning from pixels. In *International conference on machine learning*. PMLR, 2555–2565.

[10] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. 2023. Mastering Diverse Domains through World Models. *arXiv preprint* (2023). arXiv:2301.04104 https://arxiv.org/abs/2301.04104

[11] Nicolas Heess, Gregory Wayne, David Silver, Timothy Lillicrap, Tom Erez, and Yuval Tassa. 2015. Learning continuous control policies by stochastic value gradients. *Advances in neural information processing systems* 28 (2015).

[12] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. 2019. When to Trust Your Model: Model-Based Policy Optimization. In *Advances in Neural Information Processing Systems*. arXiv:1906.08253 https://proceedings.neurips.cc/paper/2019/hash/5faf461eff3099671ad63c6f3f094f7f-Abstract.html

[13] Tianying Ji, Yu Luo, Fuchun Sun, Mingxuan Jing, Fengxiang He, and Wenbing Huang. 2022. When to update your model: Constrained model-based reinforcement learning. *Advances in Neural Information Processing Systems* 35 (2022), 23150–23163.

[14] Sham Kakade and John Langford. 2002. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*. 267–274.

[15] Thanard Kurutach, Ignasi Clavera, Yan Duan, Aviv Tamar, and Pieter Abbeel. 2018. Model-ensemble trust-region policy optimization. *arXiv preprint arXiv:1802.10592* (2018).

[16] Hang Lai, Jian Shen, Weinan Zhang, and Yong Yu. 2020. Bidirectional model-based policy optimization. In *International Conference on Machine Learning*. PMLR, 5618–5627.

[17] Yongshuai Liu, Avishai Halev, and Xin Liu. 2021. Policy learning with constraints in model-free reinforcement learning: A survey. In *The 30th international joint conference on artificial intelligence (ijcai)*.

[18] Jiafei Lyu, Xiu Li, and Zongqing Lu. 2022. Double check your state before trusting it: Confidence-aware bidirectional offline model-based imagination. *Advances in Neural Information Processing Systems* 35 (2022), 38218–38231.

[19] Ester Mariucci and Markus Reiß. 2018. Wasserstein and total variation distance between marginals of Lévy processes. (2018).

[20] Aske Plaat, Walter Kosters, and Mike Preuss. 2023. High-accuracy model-based reinforcement learning, a survey. *Artificial Intelligence Review* 56, 9 (2023), 9541–9573.

[21] Zhongjian Qiao, Jiafei Lyu, and Xiu Li. 2023. Mind the model, not the agent: the primacy bias in model-based RL. *arXiv preprint arXiv:2310.15017* (2023).

[22] Aravind Rajeswaran, Sarvjeet Ghotra, Balaraman Ravindran, and Sergey Levine. 2016. Epopt: Learning robust neural network policies using model ensembles. *arXiv preprint arXiv:1610.01283* (2016).

[23] John Schulman. 2015. Trust Region Policy Optimization. *arXiv preprint arXiv:1502.05477* (2015).

[24] Richard S Sutton. 1990. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine learning proceedings 1990*. Elsevier, 216–224.

[25] Richard S Sutton. 1991. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin* 2, 4 (1991), 160–163.

[26] Chenjun Xiao, Yifan Wu, Chen Ma, Dale Schuurmans, and Martin Müller. 2019. Learning to combat compounding-error in model-based reinforcement learning. *arXiv preprint arXiv:1912.11206* (2019).

[27] Hai Zhang, Hang Yu, Junqiao Zhao, Di Zhang, Hongtu Zhou, Chang Huang, Chen Ye, et al. 2023. How to fine-tune the model: unified model shift and model bias policy optimization. *Advances in Neural Information Processing Systems* 36 (2023), 59252–59272.