Bootstrap Off-policy with World Model

Guojian Zhan[1,2], **Likun Wang**[1], **Xiangteng Zhang**[1], **Jiaxin Gao**[1],
Masayoshi Tomizuka[2], **Shengbo Eben Li**[1*]

[1] College of AI & School of Vehicle and Mobility, Tsinghua University
[2] Berkeley AI Research (BAIR), UC Berkeley

Abstract

Online planning has proven effective in reinforcement learning (RL) for improving sample efficiency and final performance. However, using planning for environment interaction inevitably introduces a divergence between the collected data and the policy's actual behaviors, degrading both model learning and policy improvement. To address this, we propose BOOM (Bootstrap Off-policy with WOrld Model), a framework that tightly integrates planning and off-policy learning through a bootstrap loop: the policy initializes the planner, and the planner refines actions to bootstrap the policy through behavior alignment. This loop is supported by a jointly learned world model, which enables the planner to simulate future trajectories and provides value targets to facilitate policy improvement. The core of BOOM is a likelihood-free alignment loss that bootstraps the policy using the planner's non-parametric action distribution, combined with a soft value-weighted mechanism that prioritizes high-return behaviors and mitigates variability in the planner's action quality within the replay buffer. Experiments on the high-dimensional DeepMind Control Suite and Humanoid-Bench show that BOOM achieves state-of-the-art results in both training stability and final performance. The code is accessible at `https://github.com/molumitu/BOOM_MBRL`.

1 Introduction

Reinforcement learning (RL) has achieved impressive performance in a wide range of domains, from industrial automation to autonomous driving and embodied intelligence [43, 36, 49]. Among the various techniques developed to enhance RL, online planning stands out for its predictive optimization ability to improve control performance using learned dynamics [29, 3, 50]. By performing look-ahead rollouts, it enables agents to anticipate future consequences and iteratively refine actions [42, 37]. Compared to model-free approaches, which rely solely on trial-and-error learning, model-based planning offers an effective tool to generate high-quality actions for environment interaction [6, 5].

A growing body of research has explored how to more effectively integrate planning into RL [17, 29]. Early methods such as PETS [4] and PlaNet [13] have proven that planning with learned dynamics and reward signals can achieve impressive control performance. To further improve performance in high-dimensional tasks, recent approaches have combined online planning with policy learning, where the policy can provide a good initial solution to speed up planning [1]. For example, LOOP [39] builds on SAC [11], an off-policy model-free algorithm, and integrates a planner guided by a learned dynamics model to enable higher-quality environment interaction. Recently, TD-MPC and TD-MPC2 [19, 18] jointly learn the dynamics, reward and value functions through temporal differential (TD) learning, achieving strong performance through both algorithmic innovations and

*Corresponding author. `lishbo@tsinghua.edu.cn`.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

implementation advances. These methods successfully deliver substantial gains over model-free baselines such as SAC, particularly on high-dimensional benchmarks.

However, this class of planning-driven model-based RL algorithms inevitably suffers from a fundamental issue known as *actor divergence*: the data used for learning is collected by the planner, which acts as a different actor from the policy network [39]. Under the paradigm framework of off-policy RL, this issue leads to two problems: (1) *Distribution shift in value learning*: The value function is trained on the data collected by the planner rather than the policy itself. As a result, it learns accurately within the planner's state-action distribution but tends to overestimate values in out-of-distribution regions that are rarely visited [23]. (2) *Unreliable policy improvement*: The policy network is updated using value estimates from the value network, which are influenced by the distributional shift. These biased estimates may mislead the policy, impairing its ability to distinguish between good and bad actions, ultimately leading to severe performance degradation [34]. To conclude, the misalignment between the training data and the policy's actual behavior can severely hinder the learning process, particularly in complex, high-dimensional environments where accurate value estimation is already challenging [32]. The resulting value bias not only misguides policy updates but also risks destabilizing the entire training process [27]. More critically, since online planning algorithms typically rely on sample-based optimization, the resulting action distributions are non-parametric and difficult to access. This means that they cannot be explicitly represented and the likelihood is intractable [47].

To address these challenges, we propose **B**ootstrap **O**ff-policy with W**O**rld **M**odel (BOOM), a novel framework that seamlessly integrates online planning with off-policy RL, effectively mitigating the negative impact of data distribution shifts caused by actor divergence. This is accomplished through a bootstrap loop: the policy initializes the planner, and the planner refines actions to bootstrap the policy via behavior alignment, alleviating the *actor divergence* issue. This loop is supported by a jointly learned world model, which enables the planner to simulate future trajectories and provides value targets that facilitate policy improvement. We refer to it as bootstrap alignment because the planner typically generates higher-quality actions via model predictive optimization. Aligning the policy with these actions also offers strong guidance for improvement and accelerates learning.

In this paper, we introduce three key contributions: **(1)** To facilitate alignment with the online planner's sample-based non-parametric distribution, we adopt a *likelihood-free alignment loss* that measures the divergence between the policy and the planner without requiring explicit likelihoods of actions from the online planner. **(2)** We introduce a *soft value-weighted mechanism* that prioritizes high-return behaviors, driven by the planner's value-guided action selection principle. Additionally, to maintain training efficiency, we align the policy with the stored planner actions in the replay buffer. Since these historical actions may vary in quality, our value-weighting mechanism ensures the policy prioritizes the high-valued promising experiences, accelerating learning while handling the variability in the planner's past actions. **(3)** BOOM combines ease of implementation with state-of-the-art (SOTA) performance on high-dimensional continuous control benchmarks, including the DeepMind Control Suite [41] and Humanoid-Bench [38].

2 Preliminaries

2.1 Reinforcement Learning

Reinforcement learning (RL) offers a powerful framework for sequential decision-making [24], formalized as a Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $P(s'|s,a)$ is the transition dynamics, $r(s,a)$ is the reward function, and $\gamma \in [0,1)$ is the discount factor. At each timestep, an agent observes state s_t, selects action a_t, receives reward r_t, and transitions to the next state s_{t+1} according to P. A fundamental concept in RL is the action-value function: $Q^\pi(s,a) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t) \,\middle|\, s_0 = s, a_0 = a\right]$, which captures the expected return of taking action a in state s, followed by policy π. Off-policy RL aims to discover the optimal policy π^* by maximizing the Q-value function using transitions generated from a different behavior policy β. By decoupling data collection from policy improvement, off-policy methods can achieve high sample efficiency—primarily due to the effective reuse of past transitions stored in a replay buffer. However, when the behavior policy β deviates too far from π, the resulting *distributional shift* can severely undermine value estimation, a delicate yet critical challenge that continues to motivate much of the recent progress in off-policy learning [8].

2.2 Online Planner

Online planning optimizes action sequences at each step by simulating future trajectories under a predictive model, enabling informed action selection [48]. This approach typically yields high-quality decisions by leveraging foresight over potential future outcomes. As a result, it is widely adopted in model-based RL for environment interaction, generating higher-quality interaction samples that facilitate more efficient learning and policy improvement.

Among various planning methods, Model Predictive Path Integral (MPPI) is a widely adopted sampling-based planner for continuous control tasks due to its high efficiency [47]. Formally, at each planning step, with the help of a dynamics model, N_p action trajectories are sampled from a factorized Gaussian: $a_t^i \sim \mathcal{N}(\mu_t, \sigma_t^2), \quad$ for $t = 0, \ldots, H{-}1$ and $i = 1, \ldots, N_p$, where H is the predictive horizon. Each trajectory is evaluated using a reward and value model to estimate the future return: $G^i = \sum_{t=0}^{H-1} r_t^i + \hat{v}_H^i$, where r_t^i denotes predicted rewards and $\hat{v}_H^i$ is the terminal value estimate. To update the trajectory distribution, MPPI reweights the N samples using a softmax over the returns: $w^i = \exp(G^i - \max_j G^j)/\sum_k \exp(G^k - \max_j G^j)$. Next, the parameters of the trajectory distribution are updated using weighted statistics: $\mu_t \leftarrow \sum_i w^i a_t^i$. This process is repeated over several iterations, gradually converging toward an optimal solution. At test time, the planner executes the first action μ_0, while during training, Gaussian noise is added for exploration.

Despite its effectiveness, MPPI and similar planners suffer from a key limitation: the resulting planned action is obtained via weighted averaging over sampled candidates and does not directly arise from a parameterized probabilistic policy. Although Gaussian noise is used during the sampling stage, the final action distribution no longer adheres to a true Gaussian form because of the reweighting and resampling process. As a result, computing the precise likelihood of MPPI-generated actions is intractable in practice.

3 Method

3.1 Inevitable Actor Divergence When Off-policy RL Meets Online Planning

A common approach to integrating online planning with off-policy RL is to jointly learn a latent world model and a policy from replay buffer data, as exemplified by TD-MPC2. Specifically, the total world model trains an encoder $z = h(s)$, a latent dynamics model $z' = f(z, a)$, a reward predictor $R(z, a)$, and a value function $Q(z, a)$, where z denotes the latent state. Throughout the learning process, all states s are first encoded into latent representations z. For notational simplicity, we continue to denote the inputs to the policy and value functions as s, although they implicitly refer to the encoded latent states. The components h, f, R, and Q are jointly trained using the TD loss:

$$\mathcal{L}_{\text{model}} = \mathbb{E}_{(s,a,r,s')_{0:H}} \left[\sum_{t=0}^{H} \gamma^t \left(\|f(z_t, a_t) - \text{sg}(h(s'_t))\|_2^2 + \text{CE}(R_t, r_t) + \text{CE}(Q_t, q_t) \right) \right], \quad (1)$$

where q_t is the target value computed using the policy network π, sg denotes the stop-gradient operator and CE denotes the cross entropy loss function. Then following standard off-policy RL, the policy π is updated by maximizing predicted Q-values, i.e., $\mathcal{L}_{\text{policy}} = -\mathbb{E}_{(s,a,r,s')_{0:H}} \sum_{t=0}^{H-1} Q(s, \pi(s))$.

Despite being trained purely on off-policy data, this world model empowers the planner, such as MPPI, to perform effective predictive optimization. We refer to the resulting behavior policy executed at each timestep as β, which can be understood as π+MPPI with world model. By seamlessly integrating model-based planning into the off-policy learning pipeline, this approach yields high-quality training interactions and enhances sample efficiency.

However, this paradigm inevitably suffers from a phenomenon known as *actor divergence*. During data collection, the agent interacts with the environment using a planner-augmented policy $\beta = \pi +$ MPPI, which refines policy actions using model rollouts. This behavior policy β may exhibit highly deterministic or multi-modal behavior, resulting in a state-action distribution $d^\beta(s, a)$ that may significantly diverges from that of the network policy π. This mismatch breaks the assumptions of standard off-policy learning and can lead to instability in training process. Specifically, such an actor divergence leads to following two major challenges:

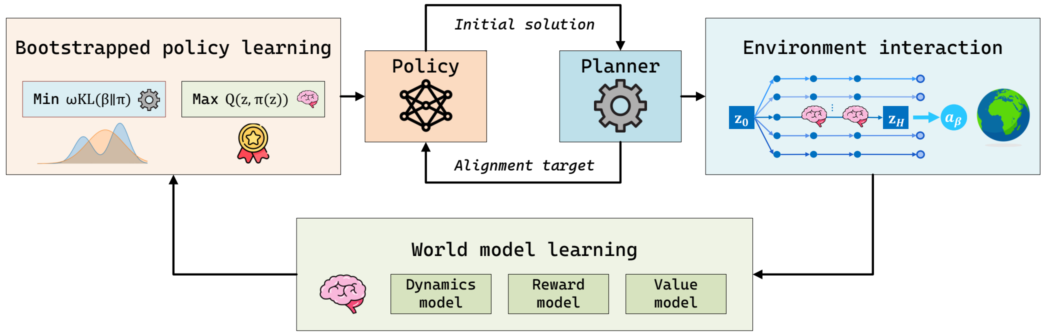


Figure 1: **Overview of the proposed BOOM algorithm.** BOOM consists of three key components: a policy, a planner, and a world model. The policy and planner form a bootstrap loop where the policy provides the planner with an initial solution, and the planner in turn guides the policy via alignment. The world model plays a dual role: it enables the planner to perform receding horizon control for collecting high-quality trajectories, and it allows the policy to utilize Q-values for effective performance improvement.

(1) Distribution shift in value learning. The value function is trained to minimize the Bellman error over samples drawn from the behavior distribution: $\mathbb{E}_{(s,a)\sim d^\beta}\left[Q(s,a) - \mathcal{T}^\pi Q(s,a)\right]^2$, where $\mathcal{T}^\pi Q(s,a) = r(s,a) + \gamma \mathbb{E}_{a' \sim \pi(s')}[Q(s', a')]$ denotes the Bellman backup under policy π. However, when d^β assigns little probability mass to regions where π places significant density, the value function is only optimized on a narrow subset of the state-action space. This mismatch can result in biased and generally overconfident estimates in out-of-distribution regions, undermining the accuracy and reliability of value learning.

(2) Unreliable policy improvement. The policy π is typically optimized by maximizing the expected value under Q. However, due to the distribution shift mentioned above, Q may be inaccurate, especially in regions where π assigns high probability but are poorly represented in the behavior distribution β. This bias will result in unstable policy updates and poor performance.

In summary, *actor divergence* inevitably occurs when online planning meets off-policy RL, which undermines the core assumption of distributional consistency in off-policy learning paradigm and degrades both model accuracy and policy improvement.

3.2 Bootstrap Off-policy with World Model

To mitigate actor divergence caused by collecting data with a planning-augmented behavior policy (e.g., $\beta = \pi + \text{MPPI}$), we propose a simple yet effective framework: **BOOM** (Bootstrap Off-policy with World Model) as shown in Figure 1. It comprises three tightly coupled components—a policy, a planner, and a world model. At the heart of BOOM is a bootstrap loop: the policy provides an initialization for the planner, while the planner refines this initialization through model predictive optimization and in turn bootstraps the policy via behavior alignment. The world model, trained in the TD-MPC2 style, serves a dual purpose: it enables the planner to simulate future trajectories for better control, and it supports the policy with value estimates for improvement.

The core of BOOM is the *Bootstrap Alignment* objective—a likelihood-free regularization term that encourages the policy π to align with planner-generated actions without requiring the likelihoods of planner's non-parametric distribution. This objective is further enhanced by a soft value-weighted mechanism that prioritizes high-return behaviors. We describe these two components in detail below.

Likelihood-free alignment metric. Since all actions stored in the replay buffer are generated by the planner policy β, we can directly imitate them to align the planner behaviors without impairing training efficiency. However, β is a non-parametric sample-based planner whose action likelihood is intractable. This makes typical imitation learning metric, such as reverse KL divergence, theoretically inapplicable, as they rely on knowing $\beta(a \mid s)$. To avoid this, we adopt a *likelihood-free* approach by minimizing the forward KL divergence:

$$\mathrm{KL}(\beta \,\|\, \pi) = \mathbb{E}_{a \sim \beta(\cdot|s)} \left[\log \frac{\beta(a \mid s)}{\pi(a \mid s)} \right] = \mathbb{E}_{a \sim \beta(\cdot|s)} \left[\log \beta(a \mid s) \right] - \mathbb{E}_{a \sim \beta(\cdot|s)} \left[\log \pi(a \mid s) \right]. \quad (2)$$

The first term depends only on β and is constant with respect to the parameters of π. Therefore, we discard it during optimization, resulting in the following simplified loss:

$$\mathcal{L}_{\text{align}} = \mathbb{E}_{(s,a,r,s')_{0:H}} \left[-\log \pi(a \mid s) \right], \quad (3)$$

which encourages π to assign higher probability to actions chosen by the planner, without requiring any access to $\beta(a \mid s)$. This likelihood-free formulation provides a simple, principled mechanism to distill the strengths of an online planner—yielding non-parametric action distributions—into a parametric policy.

Soft Q-weighted mechanism. To further enhance policy learning, we introduce a value-guided alignment objective that prioritizes high-return behaviors, drawing inspiration from the planner's action selection principle—where candidate actions are first sampled and then selected based on their value-weighted probabilities. Given a batch of off-policy transitions $\{(s_i, a_i)\}_{i=1}^{N}$ replayed from buffer, we define a soft target distribution over actions using the current Q-function: $p \propto \exp\left(Q/\tau\right)$, where $\tau > 0$ controls the distribution sharpness and is set to 1 by default. Normalization across the batch yields weights $w_i = \exp(Q_i/\tau) / \sum_{j=1}^{N} \exp(Q_i/\tau)$. Now the soft Q-weighted alignment loss becomes

$$\mathcal{L}_{\text{align}} = \mathbb{E}_{(s,a,r,s')_{0:H}} \sum_{t=0}^{H-1} \sum_{i=1}^{N} w_i \left[-\log \pi(a_i \mid s_i) \right]. \quad (4)$$

This mechanism ensures that the policy assigns higher probability to high-value actions, effectively guiding it toward promising regions. Besides, since these actions stored in the replay buffer may vary in quality, this value-weighting ensures that the policy prioritizes the most beneficial experiences, thus accelerating learning while accommodating the variability in the planner's past actions. The Q function here can also be replaced by other critics, such as state value V or advantage A, here we think Q is the most convenient one to access.

Bootstrapped policy objective. By combining the above two techniques, we integrate the alignment term into the standard policy loss to obtain the final bootstrapped policy objective:

$$\mathcal{L}_{\text{policy}} = -\mathbb{E}_{(s,a,r,s')_{0:H}} \sum_{t=0}^{H-1} \Big[Q(s, \pi(s)) + \lambda_{\text{align}} \cdot \mathcal{L}_{\text{align}} \Big], \quad (5)$$

where λ_{align} is a tunable coefficient. This bootstrapped objective encourages the policy not only to improve with respect to its own value estimates ($\max\ Q$) but also to stay aligned with the high-quality planner actions found in the replay buffer ($\min\ \mathcal{L}_{\text{q-align}}$). The complete pseudocode of our BOOM is presented in Algorithm 1.

Discussion on why the world model learning is improved. The reasons are two-fold. *(1) Improved value learning.* By aligning the policy with the planner, we reduce the discrepancy between the collected data and the actual policy behavior. This improves the distributional matching during training, enabling the value estimator to learn from more consistent and policy-relevant trajectories. As a result, the predicted values used in planning become more accurate and reliable. *(2) Improved representation, reward, and dynamics learning.* We adopt a TD-style learning objective that jointly optimizes the value function along with the other components of the world model as shown in (1). As the value predictions become more accurate due to better distributional match, the gradients flowing into the encoder, dynamics model, and reward predictor become more informative, leading to improved overall model quality.

3.3 Theoretical Analysis

We provide theoretical guarantees for bootstrap alignment in addressing *actor divergence*—the mismatch between the planner collected data and the policy actual behavior. By minimizing the divergence $\mathrm{KL}(\beta \| \pi)$, we establish theoretical bounds on the return gap and Q-value deviation, ensuring stable and efficient policy learning.

Algorithm 1 BOOM: Bootstrap Off-policy with World Model

Input: Policy π_θ, encoder h_ξ, dynamics f_ψ, reward R_ω, value Q_ϕ, planner $\mathcal{P}$
Initialize: π_θ, h_ξ, f_ψ, R_ω, Q_ϕ

`// Warmup (World Model Pretraining)`
Interact with environment using random actions: $(r, s', \texttt{done}) \leftarrow \texttt{env.step}(a_{\text{rand}})$
Store random transitions $(s, a_{\text{rand}}, r, s')$ into replay buffer $\mathcal{D}$
Update h_ξ, f_ψ, R_ω, Q_ϕ by minimizing model loss $\mathcal{L}_{\text{model}}$ in (1)

for each iteration **do**
    `// Data Collection (Using Planner)`
    Encode current observation: $z = h_\xi(s)$
    Plan: $a_\beta \sim \beta = \mathcal{P}(\pi_\theta, f_\psi, R_\omega, Q_\phi, z)$
    Interact with environment: $(r, s', \texttt{done}) \leftarrow \texttt{env.step}(a_\beta)$
    Store transition (s, a_β, r, s') into replay buffer $\mathcal{D}$
    `// World Model and Policy Learning`
    Replay rollout batch $\{(s_t, a_\beta, r_t, s_{t+1})_{t=0}^{H-1}\} \sim \mathcal{D}$
    Update h_ξ, f_ψ, R_ω, Q_ϕ by minimizing model loss $\mathcal{L}_{\text{model}}$ in (1)
    Update π_θ by minimizing bootstrapped policy loss $\mathcal{L}_{\text{policy}}$ in (5)
end for

Theorem 1 (Bootstrap Alignment Controls Return Gap). *Let $\beta(a \mid s)$ be the behavior policy, $\pi(a \mid s)$ be the learned policy, and $d^\beta(s)$ the state distribution induced by β. Assume the per-step reward satisfies $|r(s,a)| \leq R_{\max}$ and the discount factor $\gamma \in [0, 1)$. Then for any state s, if $\mathrm{KL}(\beta \| \pi) \leq \varepsilon$, the following return gap bound holds:*

$$\big|J(\beta) - J(\pi)\big| \leq \frac{R_{\max}}{1-\gamma}\sqrt{2\varepsilon}. \tag{6}$$

Proof. See Appendix A.2 ☐

The first theorem shows that bootstrap alignment ensures a small return gap between β and π, meaning staying close to the planner avoids performance drops from distribution mismatch.

Theorem 2 (Bootstrap Alignment Controls Q-Value Gap). *Assume that for any state s, the learned policy $\pi(a \mid s)$ and the planner $\beta(a \mid s)$ satisfy $\mathrm{KL}(\beta \| \pi) \leq \varepsilon$, and that $Q(s, a)$ is L_Q-Lipschitz continuous in a. Then for any state s, the expected Q-value difference is bounded as:*

$$|Q(s, a_\beta) - Q(s, a_\pi)| \leq L_Q \cdot \|a_\beta - a_\pi\|_2 \leq L_Q \cdot D(\varepsilon), \tag{7}$$

where $D(\varepsilon)$ is an upper bound on the 2-norm distance $\|a_\beta - a_\pi\|_2$ with $a_\beta \sim \beta(s)$ and $a_\pi \sim \pi(s)$. In general, the action distribution of MPPI can be approximated by a Gaussian Mixture Model (GMM) as $\beta(s) = \sum_{i=1}^K w_i \mathcal{N}(\mu_i, \Sigma_i)$ with weights $w_i \geq 0$, $\sum_{i=1}^K w_i = 1$, and $\pi(s) = \mathcal{N}(\mu_\pi, \Sigma_\pi)$ is a Gaussian policy. Denote the maximum eigenvalue of each Σ_i as $\Lambda_i := \Lambda(\Sigma_i)$ and that of Σ_π as $\Lambda(\Sigma_\pi)$. Then for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ over $a_\beta \sim \beta(s)$ and $a_\pi \sim \pi(s)$, the deviation bound satisfies:

$$D(\varepsilon) \leq \min\left(2\sqrt{d},\ \max_i\left(\sqrt{2\Lambda_i \log\tfrac{K}{\delta}} + \sqrt{2\varepsilon\Lambda(\Sigma_\pi)/w_i}\right) + \sqrt{2\Lambda(\Sigma_\pi)\log\tfrac{1}{\delta}}\right). \tag{8}$$

Here, d is the dimensionality of the normalized action space.

Proof. See Appendix A.3 ☐

The second theorem bounds the Q-value difference under bootstrap alignment, showing that value overestimation—commonly seen when the policy strays into poorly covered regions—is effectively controlled by alignment. This avoids misleading policy updates and stabilizes training.

Together, these insights justify that bootstrap alignment mitigates *actor divergence* by keeping the learned policy close to the behavior policy, preventing large discrepancies in value estimates. It retains the benefits of off-policy learning without requiring access to behavior policy likelihoods, making it compatible with modern planners and applicable in complex, high-dimensional settings.

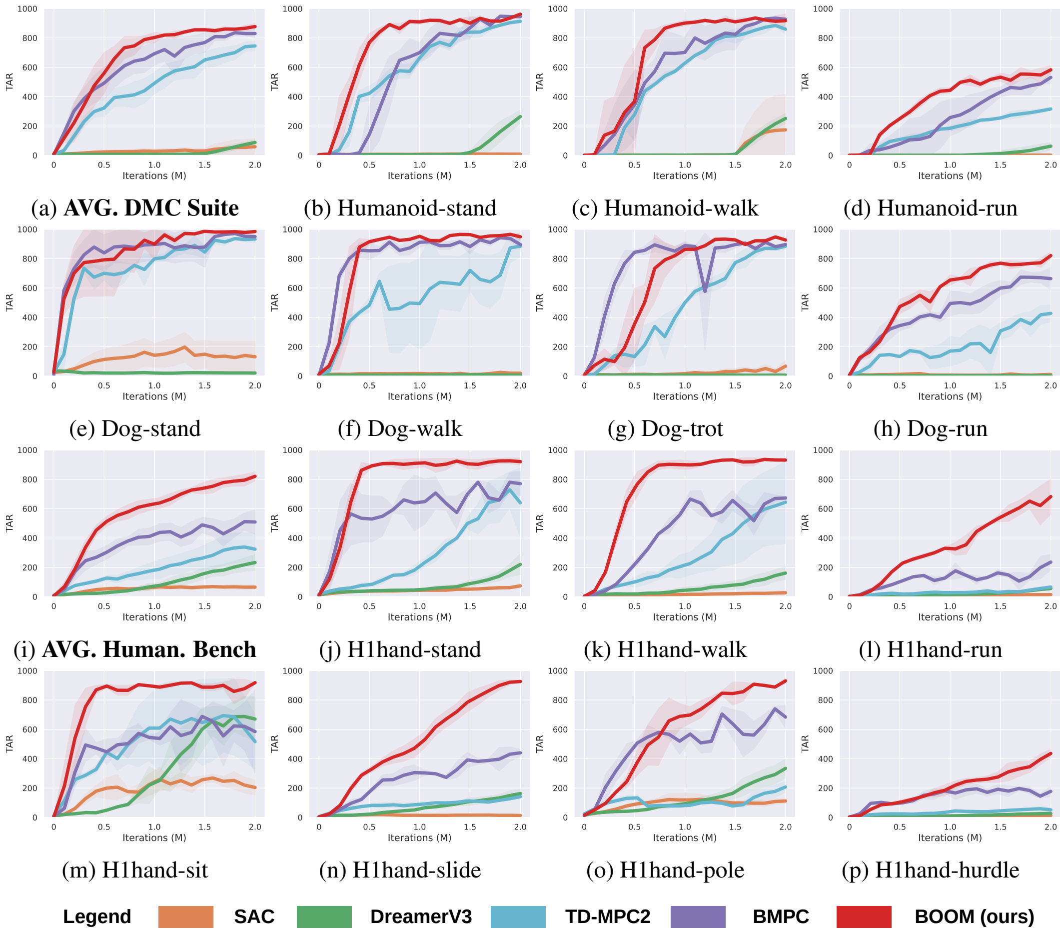


Figure 2: **Training curves on benchmarks.** The solid lines represent the mean, while the shaded regions indicate the confidence interval over three runs. The average performance curves for the two benchmarks appear at the left corner of the 1st and 3rd rows, respectively, highlighted in **bold**.

4 Experiments

4.1 Experimental Setup

Baselines. We choose four representative online RL algorithms as our baselines: (1) **SAC** [11]: the state-of-the-art model-free off-policy RL algorithm under the maximum entropy framework; (2) **DreamerV3** [16]: the state-of-the-art imagination-driven model-based RL algorithm that learns from imaginary rollouts generated by a world model. To enable a comprehensive comparison, we evaluate DreamerV3 under two different interaction budgets: 2M and 10M iterations; (3) **TD-MPC2** [18]: a planning-driven model-based RL algorithm that performs online planning via an MPPI planner; (4) **BMPC** [45]: an improved variant of TD-MPC2, where the policy is trained solely to imitate the actions generated by the MPPI with an extra relabeling mechanism to update the historical actions in the replay buffer with the latest planner.

Benchmarks. We evaluate our method on a challenging benchmark of 14 high-dimensional locomotion tasks drawn from the DeepMind Control Suite (DMC) [41] and the recently proposed Humanoid Bench (H-Bench) [38]. The chosen 7 DMC tasks feature two most complex agents—humanoid (67/21 state/action dims) and dog (223/38)—that demand sophisticated balance and coordination. The other 7 H-Bench tasks raise the difficulty further with long-horizon, goal-directed tasks on the Unitree H1hand robot (151/61), such as walking over slides, traveling over a pole forest without collision, and continuously crossing hurdles. Detailed descriptions are listed in Appendix B.1.

Implementation details. The detailed hyperparameters and reproducibility statement of other baselines are documented in Appendix B.2.

4.2 Experimental Results

All the training curves are shown in Figure 2 and the detailed numerical results are listed in Table 1. Our method, BOOM, consistently delivers the best Total Average Return (TAR) across all 14 high-dimensional locomotion tasks. These tasks pose significant challenges due to their large state and action spaces, yet BOOM demonstrates remarkable stability and effectiveness.

Results on the DMC Suite. Our BOOM achieves an average TAR of **877.7**, outperforming the previous best BMPC (835.8) by a substantial **+5.0%**, and exceeding TD-MPC2 by an even larger margin of **+17.7%**. Notably, BOOM achieves new best results on all humanoid and dog tasks. In *Humanoid-run*, BOOM outperforms the second-best method by **+9.7%**, and in *Dog-run*, it leads by a staggering **+21.8%**. While SAC and DreamerV3 often fail on these high-dimensional control tasks and achieve near-zero performance, planning-based methods like TD-MPC2 and BMPC perform better but still suffer from instability and limited final returns. In contrast, BOOM learns faster and achieves much stronger final performance across all tasks.

Results on the Humanoid Bench. Our BOOM again dominates with an average TAR of **820.6**. This marks a dramatic **+47.7%** improvement over DreamerV3 (10M), which is (555.6), and an even more impressive **+60.5%** gain over BMPC. BOOM sets new records on every single task in the Humanoid Bench. For example, in *H1hand-slide*, BOOM improves over the second-best method by **+110.5%**, in *H1hand-pole*, by **+25.8%**, and in *H1hand-hurdle*, by **+121.0%**, .

In summary, compared to SAC's difficulty in scaling to complex control, DreamerV3's limitations in sample efficiency, and BMPC/TD-MPC2's occasional instability, our BOOM demonstrates clear and consistent advantages across the board.

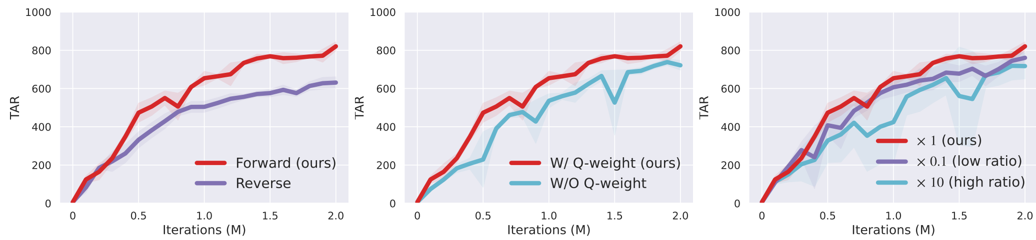
Table 1: Total Average Return (TAR) on 7 DMC Suite tasks and 7 Humanoid Benchmark (H-Bench) tasks. Mean ± Std over 3 seeds. **Bold** = best, underlined = second-best. Higher is better.

Task	SAC	DreamerV3 (2M & 10M iters)		TD-MPC2	BMPC	BOOM (ours)
Humanoid-stand	9.0 ± 0.7	264.5 ± 44.1	717.0 ± 21.2	913.3 ± 14.7	947.9 ± 4.4	**962.1** ± 10.7
Humanoid-walk	173.8 ± 242.2	251.5 ± 35.2	755.6 ± 25.2	884.8 ± 8.3	935.1 ± 4.0	**936.1** ± 3.3
Humanoid-run	1.6 ± 0.1	62.5 ± 25.8	353.5 ± 33.2	316.2 ± 9.2	531.2 ± 42.0	**582.8** ± 26.0
Dog-stand	197.6 ± 102.4	35.4 ± 10.8	35.4 ± 10.8	936.4 ± 7.6	971.3 ± 11.0	**986.8** ± 1.8
Dog-walk	24.7 ± 11.3	9.1 ± 0.6	9.1 ± 0.6	885.0 ± 74.8	942.9 ± 9.6	**965.4** ± 0.3
Dog-trot	67.1 ± 39.9	7.9 ± 0.7	8.4 ± 0.9	884.4 ± 22.2	911.3 ± 18.2	**947.9** ± 4.7
Dog-run	16.5 ± 8.5	4.3 ± 3.2	4.3 ± 3.2	427.0 ± 57.9	673.7 ± 50.2	**820.7** ± 23.0
AVG. DMC Suite	58.8 ± 57.7	87.9 ± 15.8	269.0 ± 13.6	745.6 ± 34.1	835.8 ± 20.7	**877.7** ± 20.3
H1hand-stand	74.1 ± 17.5	220.3 ± 73.5	845.4 ± 27.3	728.7 ± 121.9	780.0 ± 65.8	**926.1** ± 19.2
H1hand-walk	27.0 ± 13.8	161.3 ± 44.5	744.0 ± 28.7	644.2 ± 281.1	672.6 ± 10.4	**935.4** ± 7.3
H1hand-run	14.1 ± 1.4	55.8 ± 10.5	622.4 ± 66.7	66.1 ± 8.1	236.0 ± 53.9	**682.2** ± 120.6
H1hand-sit	268.4 ± 26.1	687.3 ± 138.0	699.1 ± 177.2	693.7 ± 249.9	688.2 ± 46.3	**918.1** ± 4.2
H1hand-slide	19.0 ± 5.9	162.6 ± 29.5	367.6 ± 29.7	141.3 ± 15.6	440.1 ± 25.4	**926.1** ± 8.0
H1hand-pole	122.5 ± 33.5	334.3 ± 65.1	577.4 ± 62.3	207.5 ± 35.6	739.9 ± 18.0	**930.5** ± 18.9
H1hand-hurdle	12.9 ± 2.7	26.6 ± 3.0	135.7 ± 6.1	59.0 ± 19.3	197.1 ± 12.1	**435.6** ± 29.8
AVG. H-Bench.	68.5 ± 9.2	233.0 ± 53.9	555.6 ± 49.5	338.8 ± 98.6	511.7 ± 59.2	**820.6** ± 31.0

4.3 Ablation Study

We conduct three ablation studies to assess the contribution of key components in our framework:

Bootstrap alignment metric. We compare the common reverse KL divergence with our proposed likelihood-free forward KL. As shown in Figure 3a, forward KL consistently shows higher returns, highlighting its strength in capturing the planner's non-parametric distribution without requiring likelihood estimation. Reverse KL, in contrast, relies on approximating the planner's likelihood,



(a) Ablation study on the alignment metric

(b) Ablation study on the Q-weight mechanism

(c) Ablation study on the alignment coefficient

Figure 3: **Ablation study curves.** We select the *Dog-run* task (223/38 state/action dims) in DMC Suite with the highest dimensionality to perform all ablation experiments.

which is not directly accessible. We estimate it using a Gaussian surrogate based on the value-weighted mean and variance of planner actions. The weaker performance also suggests that such approximations may be inaccurate and detrimental. Further discussion is provided in Appendix C.1.

Soft Q-weight mechanism. We replace the soft Q-weighting with uniform weighting to assess the benefit of leveraging the learned Q-function. As shown in Figure 3b, incorporating Q-weights consistently accelerates training and improves final performance. This improvement stems from the ability of Q-weighting to handle the variability in action quality within the replay buffer by prioritizing high-value actions, thereby enabling more focused and efficient policy updates.

Alignment coefficient. We vary the alignment coefficient by testing $0.1\times$ and $10\times$ the default setting ($\lambda_{\text{align}} = \dim(\mathcal{A})/1000$ for DMC and $\dim(\mathcal{A})/50$ for Humanoid-Bench). Results in Figure 3c show stable performance across this range, suggesting that BOOM is robust to this hyperparameter and does not require very sensitive tuning.

5 Related Work

Model-based RL, i.e., MBRL can be broadly categorized into planning-driven and imagination-driven approaches, depending on how the learned model is utilized during training.

Planning-driven MBRL. Planning-driven methods use planner rather than policy itself to generate high-quality actions for environment interaction [5]. Early work has shown that solely learning the value and then combining it with the planner can achieve good control performance [31, 22, 7, 26]. To further improve performance in high-dimensional tasks, recent approaches have combined online planning with policy learning, where the policy can provide a good initial solution to speed up planning[20, 44, 33, 2, 25]. LOOP [39] takes SAC [11] as the backbone, and employs a planner under the policy behavior constraint for collecting samples. TD-MPC family [19, 18] jointly learns model and value function through TD-learning, achieving strong performance through both algorithmic innovations and implementation advances. It successfully delivers substantial gains over model-free baselines, particularly on complex benchmarks. However, these methods inevitably encounter *actor divergence*—a mismatch between the planner and the policy. Our approach, **BOOM**, addresses this challenge by tightly coupling planning and off-policy learning through a bootstrap loop that aligns the policy with the planner's non-parametric action distribution via a Q-weighted likelihood-free alignment loss, preserving distributional consistency. A recent method BMPC [45] simplifies the pipeline by discarding explicit policy optimization and directly imitating the planner to avoid *actor divergence*. However, BMPC ignores the Q-function during training, resulting in lower policy learning efficiency and sensitivity to the variability of historical planner actions in the buffer. This leads to unstable learning, as reflected by its oscillatory training curves in our experiments. We acknowledge a concurrent and close work, TDM(PC)^2 [25], which similarly found that aligning the policy with the planner is beneficial. The major distinction lies in the design of the alignment objective: their approach follows a TD3+BC style using reverse KL, whereas ours is closer to AWAC using critic-guided weights and forward KL.

Imagination-driven MBRL. Imagination-driven methods leverage a learned model to generate synthetic rollouts for policy and value updates [40]. Modern approaches such as SimPLe [21], IRIS [28], IDM [30], and the Dreamer family [12, 14, 15] train latent dynamics models to support actor-critic learning entirely in imagination. These methods offer fast test-time execution and relatively high sample efficiency, but their performance is often limited by compounding model errors over long imagined rollouts. Among them, DreamerV3 [16] stands out as a leading representative, demonstrating strong performance across a range of tasks. Unlike planning-based MBRL methods, DreamerV3 interacts by directly sampling actions from its learned policy, typically requiring more iterations to converge. For comprehensive comparison, we evaluate DreamerV3 under 2M and 10M iterations. While it improves with more interaction budget and outperforms TD-MPC2 and BMPC on certain tasks, our BOOM consistently outperforms all baselines across all tasks.

Compared to Offline RL. One might notice the similarity between BOOM and offline RL; however, the fundamental paradigms and practical implications of these two settings differ significantly. *(1) Learning paradigm.* Offline RL centers on a fixed dataset, relying heavily on behavior cloning (BC) to restrict the policy within the dataset's support [10]. To cautiously improve policy performance, offline RL relaxes BC for poor actions, but always aims to keep the policy close to known data to avoid extrapolation error [34, 35]. BOOM centers on the policy itself as the optimization target. It seeks to maximize Q-values and align with planner-generated actions, both aimed at improving policy performance. Ultimately, the policy and planner co-adapt and converge to the optimal solution through online interaction. *(2) Learning objective.* In offline RL, maximizing Q-values and BC often conflict [9]: Q-values outside the dataset distribution cannot be accurately estimated, so maximizing Q risks pushing the policy toward unsupported actions; BC pulls the policy back toward known actions. This tension forces a conservative balance [46]. In BOOM, the two objectives are largely complementary: the planner generally produces higher-quality actions than the policy [51]. Aligning policy with the planner improves performance and strengthens policy-planner consistency, which in turn leads to more accurate Q estimates. More accurate Q-values then enable better policy improvement and allow the planner to generate higher-quality actions. This positive feedback loop bootstraps policy and planner consistently toward faster convergence to the optimal solution.

6 Conclusion

We introduce BOOM, a model-based RL method that enhances the integration of planning and off-policy learning through bootstrap alignment. By leveraging the non-parametric planner actions not only for environment interaction but also for bootstrapping policy behavior via a Q-weighted likelihood-free alignment loss, BOOM mitigates the inevitably actor divergence issue in planning-driven model-based RL methods, improving both training stability and final performance while maintaining high time-efficiency and flexibility of off-policy learning paradigm. Experiments on tens of high-dimensional locomotion tasks show that BOOM consistently outperforms existing planning-driven and imagination-driven baselines. We believe BOOM establishes a strong foundation with ample room for further improvement, such as more adaptive integration of max-Q and bootstrap alignment objectives, or the adoption of more expressive policy classes beyond diagonal Gaussians like diffusion models to unlock even higher performance and broader applicability.

Our work underscores two critical directions for advancing planning-driven MBRL. First, we emphasize that the learned value function remains the fundamental bottleneck; reliable improvement in its estimation accuracy consistently translates to enhanced planning and superior final asymptotic performance. Second, a promising avenue for future research concerns the principled adjustment of the maximum entropy temperature coefficient (α). In planning-driven MBRL, the sampling is dictated by the planner's search, yielding an intricate, non-trivial relationship between the planner's intrinsic exploration entropy and the network policy's entropy. Developing a automatic tuning mechanism for α based on these entropy signals is key to stabilizing the inherent tension between exploration and exploitation—a challenge that continues to define the frontier of effective RL algorithm design.

7 Acknowledgment

This study is supported by the Tsinghua University-Toyota Joint Research Center for AI Technology of Automated Vehicle, Beijing Natural Science Foundation (L257002), and SunRisingAI Lab.

References

[1] Arthur Argenson and Gabriel Dulac-Arnold. Model-based offline planning. *arXiv preprint arXiv:2008.05556*, 2020.

[2] Jacob Buckman, Danijar Hafner, George Tucker, Eugene Brevdo, and Honglak Lee. Sample-efficient reinforcement learning with stochastic ensemble value expansion. *Advances in neural information processing systems*, 31, 2018.

[3] Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Afroz Mohiuddin, Ryan Sepassi, George Tucker, and Henryk Michalewski. Model based reinforcement learning for atari. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 6–9, 2019.

[4] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.

[5] Sebastian Curi, Felix Berkenkamp, and Andreas Krause. Efficient model-based reinforcement learning through optimistic policy search and planning. *Advances in Neural Information Processing Systems*, 33:14156–14170, 2020.

[6] Marc Deisenroth and Carl E Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pages 465–472, 2011.

[7] Marc Peter Deisenroth, Dieter Fox, and Carl Edward Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE transactions on pattern analysis and machine intelligence*, 37(2):408–423, 2013.

[8] Jingliang Duan, Wenxuan Wang, Liming Xiao, Jiaxin Gao, and Shengbo Eben Li. Dsac-t: Distributional soft actor-critic with three refinements. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[9] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in Neural Information Processing Systems*, 34:20132–20145, 2021.

[10] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pages 2052–2062. PMLR, 2019.

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

[12] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

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

[14] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

[15] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[16] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, pages 1–7, 2025.

[17] Jessica B Hamrick, Abram L Friesen, Feryal Behbahani, Arthur Guez, Fabio Viola, Sims Witherspoon, Thomas Anthony, Lars Holger Buesing, Petar Veličković, and Theophane Weber. On the role of planning in model-based deep reinforcement learning. In *International Conference on Learning Representations*, 2020.

[18] Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. *arXiv preprint arXiv:2310.16828*, 2023.

[19] Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. *arXiv preprint arXiv:2203.04955*, 2022.

[20] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32, 2019.

[21] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*, 2019.

[22] Jus Kocijan, Roderick Murray-Smith, Carl E Rasmussen, and Agathe Girard. Gaussian process model based predictive control. In *Proceedings of the 2004 American control conference*, volume 3, pages 2214–2219. IEEE, 2004.

[23] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191, 2020.

[24] Shengbo Eben Li. *Reinforcement Learning for Sequential Decision and Optimal Control*. Springer Verlag, Singapore, 2023.

[25] Haotian Lin, Pengcheng Wang, Jeff Schneider, and Guanya Shi. Td-mpc^2: Improving temporal difference mpc through policy constraint. *arXiv preprint arXiv:2502.03550*, 2025.

[26] Kendall Lowrey, Aravind Rajeswaran, Sham Kakade, Emanuel Todorov, and Igor Mordatch. Plan online, learn offline: Efficient learning and exploration via model-based control. In *International Conference on Learning Representations*, 2019.

[27] Jared Markowitz, Jesse Silverberg, and Gary Lynn Collins. Avoiding value estimation error in off-policy deep reinforcement learning. In *I Can't Believe It's Not Better Workshop: Failure Modes of Sequential Decision-Making in Practice (RLC 2024)*, 2024.

[28] Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. *arXiv preprint arXiv:2209.00588*, 2022.

[29] Thomas M Moerland, Joost Broekens, Aske Plaat, Catholijn M Jonker, et al. Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 16(1):1–118, 2023.

[30] Yao Mu, Yuzheng Zhuang, Bin Wang, Guangxiang Zhu, Wulong Liu, Jianyu Chen, Ping Luo, Shengbo Li, Chongjie Zhang, and Jianye Hao. Model-based reinforcement learning via imagination with derived memory. *Advances in Neural Information Processing Systems*, 34:9493–9505, 2021.

[31] Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE international conference on robotics and automation ()*, pages 7559–7566. IEEE, 2018.

[32] Michal Nauman, Michał Bortkiewicz, Piotr Miłoś, Tomasz Trzcinski, Mateusz Ostaszewski, and Marek Cygan. Overestimation, overfitting, and plasticity in actor-critic: the bitter lesson of reinforcement learning. In *International Conference on Machine Learning*, pages 37342–37364. PMLR, 2024.

[33] Tung D Nguyen, Rui Shu, Tuan Pham, Hung Bui, and Stefano Ermon. Temporal predictive coding for model-based planning in latent space. In *International Conference on Machine Learning*, pages 8130–8139. PMLR, 2021.

[34] Rafael Figueiredo Prudencio, Marcos ROA Maximo, and Esther Luna Colombini. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[35] Yuhang Ran, Yi-Chen Li, Fuxiang Zhang, Zongzhang Zhang, and Yang Yu. Policy regularization with dataset constraint for offline reinforcement learning. In *International Conference on Machine Learning*, pages 28701–28717. PMLR, 2023.

[36] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

[37] Julian Schrittwieser, Thomas Hubert, Amol Mandhane, Mohammadamin Barekatain, Ioannis Antonoglou, and David Silver. Online and offline reinforcement learning by planning with a learned model. *Advances in Neural Information Processing Systems*, 34:27580–27591, 2021.

[38] Carmelo Sferrazza, Dun-Ming Huang, Xingyu Lin, Youngwoon Lee, and Pieter Abbeel. Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation. *arXiv preprint arXiv:2403.10506*, 2024.

[39] Harshit Sikchi, Wenxuan Zhou, and David Held. Learning off-policy with online planning. In *Conference on Robot Learning*, pages 1622–1633. PMLR, 2022.

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

[41] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

[42] Yuval Tassa, Tom Erez, and Emanuel Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4906–4913. IEEE, 2012.

[43] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *nature*, 575(7782):350–354, 2019.

[44] Tingwu Wang and Jimmy Ba. Exploring model-based planning with policy networks. *arXiv preprint arXiv:1906.08649*, 2019.

[45] Yuhang Wang, Hanwei Guo, Sizhe Wang, Long Qian, and Xuguang Lan. Bootstrapped model predictive control. *arXiv preprint arXiv:2503.18871*, 2025.

[46] Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *International Conference on Learning Representations*, 2022.

[47] Grady Williams, Andrew Aldrich, and Evangelos A Theodorou. Model predictive path integral control: From theory to parallel computation. *Journal of Guidance, Control, and Dynamics*, 40(2):344–357, 2017.

[48] Guojian Zhan, Qiang Ge, Haoyu Gao, Yuming Yin, Bin Zhao, and Shengbo Eben Li. An explicit discrete-time dynamic vehicle model with assured numerical stability. *Vehicle System Dynamics*, pages 1–24, 2024.

[49] Guojian Zhan, Yuxuan Jiang, Shengbo Eben Li, Yao Lyu, Xiangteng Zhang, and Yuming Yin. A transformation-aggregation framework for state representation of autonomous driving systems. *IEEE Transactions on Intelligent Transportation Systems*, 25(7):7311–7322, 2024.

[50] Guojian Zhan, Yao Lyu, Shengbo Eben Li, Yuxuan Jiang, Xiangteng Zhang, and Letian Tao. Enhance generality by model-based reinforcement learning and domain randomization. In *2023 7th CAA International Conference on Vehicular Control and Intelligence (CVCI)*, pages 1–6. IEEE, 2023.

[51] Guojian Zhan, Xiangteng Zhang, Feihong Zhang, Letian Tao, and Shengbo Eben Li. Bicriteria policy optimization for high-accuracy reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

A Theoretical Analysis

A.1 Useful Lemmas

Lemma 1 (Triangle Inequality). *For any vectors $x, y \in \mathbb{R}^n$, the triangle inequality states that the norm of the sum of two vectors is less than or equal to the sum of their norms:*

$$\|x + y\| \leq \|x\| + \|y\|,$$

where $\|\cdot\|$ denotes the standard Euclidean norm.

Lemma 2 (Pinsker's Inequality). *Let p and q be two probability distributions over a measurable space $(\mathcal{X}, \mathcal{F})$. Denote the total variation distance between p and q as*

$$\mathrm{TV}(p, q) := \sup_{A \in \mathcal{F}} |p(A) - q(A)| = \frac{1}{2} \int_{\mathcal{X}} |p(x) - q(x)| \, dx,$$

where p and q are the probability density functions of p and q, respectively. The forward Kullback–Leibler (KL) divergence from p to q is defined as

$$\mathrm{KL}(p \| q) := \int_{\mathcal{X}} p(x) \log \frac{p(x)}{q(x)} \, dx.$$

Then the total variation distance is upper bounded by the square root of the forward KL divergence:

$$\mathrm{TV}(p, q) \leq \sqrt{\tfrac{1}{2} \, \mathrm{KL}(p \| q)}.$$

Lemma 3 (Total Variation Bound on Expectation Difference). *Let p and q be two probability densities over a common measurable space $\mathcal{X}$, and let $f : \mathcal{X} \to \mathbb{R}$ be a measurable function such that $\|f\|_\infty = \sup_{x \in \mathcal{X}} |f(x)| < \infty$. Then the difference in expectations is bounded by*

$$|\mathbb{E}_{x \sim p}[f(x)] - \mathbb{E}_{x \sim q}[f(x)]| \leq 2\|f\|_\infty \cdot \mathrm{TV}(p, q),$$

Proof. We begin by expressing the difference in expectations:

$$|\mathbb{E}_p[f(x)] - \mathbb{E}_q[f(x)]| = \left| \int_{\mathcal{X}} f(x)(p(x) - q(x)) \, dx \right|.$$

By the triangle inequality,

$$\left| \int_{\mathcal{X}} f(x)(p(x) - q(x)) \, dx \right| \leq \int_{\mathcal{X}} |f(x)||p(x) - q(x)| \, dx.$$

Using the fact that $|f(x)| \leq \|f\|_\infty$, we obtain

$$\int_{\mathcal{X}} |f(x)||p(x) - q(x)| \, dx \leq \|f\|_\infty \int_{\mathcal{X}} |p(x) - q(x)| \, dx.$$

Recalling the definition of total variation distance,

$$\int_{\mathcal{X}} |p(x) - q(x)| \, dx = 2 \, \mathrm{TV}(p, q),$$

we conclude that

$$|\mathbb{E}_p[f(x)] - \mathbb{E}_q[f(x)]| \leq 2\|f\|_\infty \cdot \mathrm{TV}(p, q).$$

□

Lemma 4 (Concentration of Gaussian Policy Samples). *Let $a_\pi \sim \mathcal{N}(\mu_\pi, \Sigma_\pi)$ be a sample from a multivariate Gaussian distribution with mean $\mu_\pi \in \mathbb{R}^d$ and covariance matrix $\Sigma_\pi \in \mathbb{R}^{d \times d}$. Define*

$$\Lambda(\Sigma_\pi) := \sup_{\|v\|_2 = 1} v^\top \Sigma_\pi v$$

as the largest directional variance (i.e., the spectral norm of Σ_π). Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$,

$$\|a_\pi - \mu_\pi\|_2 \leq \sqrt{2\Lambda(\Sigma_\pi) \log \tfrac{1}{\delta}}.$$

Proof. Let $z \sim \mathcal{N}(0, I_d)$, and let L be a matrix such that $\Sigma_\pi = LL^\top$. Then the policy sample can be written as $a_\pi = \mu_\pi + Lz$, and we have

$$\|a_\pi - \mu_\pi\|_2^2 = \|Lz\|_2^2 = z^\top \Sigma_\pi z.$$

Using a standard concentration bound for sub-Gaussian quadratic forms (e.g., the Laurent–Massart inequality), for any $\delta \in (0, 1)$,

$$\Pr\left(z^\top \Sigma_\pi z \geq \mathbb{E}[z^\top \Sigma_\pi z] + 2\Lambda(\Sigma_\pi) \log \tfrac{1}{\delta}\right) \leq \delta.$$

Since $\mathbb{E}[z^\top \Sigma_\pi z] = \mathrm{Tr}(\Sigma_\pi)$, we obtain

$$\Pr\left(\|a_\pi - \mu_\pi\|_2^2 \geq \mathrm{Tr}(\Sigma_\pi) + 2\Lambda(\Sigma_\pi) \log \tfrac{1}{\delta}\right) \leq \delta.$$

By omitting the trace term, we get a looser but simpler bound:

$$\|a_\pi - \mu_\pi\|_2 \leq \sqrt{2\Lambda(\Sigma_\pi) \log \tfrac{1}{\delta}}, \quad \text{with probability at least } 1 - \delta.$$

□

Lemma 5 (Mean Bound via KL Between Gaussians)**.** *Let $p = \mathcal{N}(\mu_p, \Sigma_p)$ and $q = \mathcal{N}(\mu_q, \Sigma_q)$ be two multivariate Gaussian distributions of dimension d. Then,*

$$\mathrm{KL}(p\|q) \geq \frac{1}{2}\|\mu_p - \mu_q\|_{\Sigma_q^{-1}}^2.$$

Consequently, if $\mathrm{KL}(p\|q) \leq \varepsilon$, then

$$\|\mu_p - \mu_q\|_2 \leq \sqrt{2\varepsilon\Lambda(\Sigma_q)},$$

where $\Lambda(\Sigma_q)$ denotes the largest eigenvalue of Σ_q.

Proof. The KL divergence between Gaussians is given by:

$$\mathrm{KL}(p\|q) = \frac{1}{2}\left(\mathrm{tr}(\Sigma_q^{-1}\Sigma_p) + (\mu_q - \mu_p)^\top \Sigma_q^{-1}(\mu_q - \mu_p) - d + \log \frac{\det \Sigma_q}{\det \Sigma_p}\right).$$

Dropping the non-negative trace and log-determinant terms, we obtain the lower bound:

$$\mathrm{KL}(p\|q) \geq \frac{1}{2}\|\mu_p - \mu_q\|_{\Sigma_q^{-1}}^2.$$

Applying the spectral norm inequality $\|v\|_2^2 \leq \Lambda(\Sigma_q)\|v\|_{\Sigma_q^{-1}}^2$, we get:

$$\|\mu_p - \mu_q\|_2^2 \leq 2\varepsilon\Lambda(\Sigma_q).$$

This completes the proof. □

A.2 Proof of Theorem 1

Proof. We aim to bound the difference between the expected returns under two policies π and β, based on the divergence of their induced trajectory distributions.

By definition, the expected return under a policy π is:

$$J(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right],$$

where $\tau = (s_0, a_0, s_1, a_1, \ldots)$ denotes a full trajectory, and similarly for β, we obtain

$$J(\beta) = \mathbb{E}_{\tau\sim\beta}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right],$$

We compute the difference:

$$|J(\beta) - J(\pi)| = \left| \sum_{t=0}^{\infty} \gamma^t \left(\mathbb{E}_{(s_t,a_t)\sim\beta}[r(s_t,a_t)] - \mathbb{E}_{(s_t,a_t)\sim\pi}[r(s_t,a_t)] \right) \right|.$$

Using the triangle inequality in Lemma 1, this is upper bounded by

$$\sum_{t=0}^{\infty} \gamma^t \left| \mathbb{E}_{(s_t,a_t)\sim\beta}[r(s_t,a_t)] - \mathbb{E}_{(s_t,a_t)\sim\pi}[r(s_t,a_t)] \right|.$$

Assuming the reward function is uniformly bounded as $|r(s,a)| \leq R_{\max}$, Lemma 3 implies the following bound:

$$\left| \mathbb{E}_{(s_t,a_t)\sim\beta}[r(s_t,a_t)] - \mathbb{E}_{(s_t,a_t)\sim\pi}[r(s_t,a_t)] \right| \leq 2R_{\max} \cdot \mathrm{TV}(p_t^{\beta}, p_t^{\pi}),$$

where p_t^{β} and p_t^{π} denote the marginal distributions of (s_t, a_t) under policies π and β, respectively.

Substituting this into the previous expression, we obtain:

$$|J(\beta) - J(\pi)| \leq 2R_{\max} \sum_{t=0}^{\infty} \gamma^t \cdot \mathrm{TV}(p_t^{\beta}, p_t^{\pi}).$$

Now, applying Pinsker's inequality in Lemma 2, we further have

$$\mathrm{TV}(p_t^{\beta}, p_t^{\pi}) \leq \sqrt{\frac{1}{2}\mathrm{KL}(p_t^{\beta} \| p_t^{\pi})}.$$

Assuming that at every timestep, the KL divergence is bounded as $\mathrm{KL}(p_t^{\beta} \| p_t^{\pi}) \leq \varepsilon$, we get:

$$|J(\beta) - J(\pi)| \leq 2R_{\max} \sum_{t=0}^{\infty} \gamma^t \sqrt{\frac{\varepsilon}{2}} = \frac{2R_{\max}}{1-\gamma}\sqrt{\frac{\varepsilon}{2}} = \frac{R_{\max}}{1-\gamma}\sqrt{2\varepsilon}.$$

This completes the proof. □

A.3 Proof of Theorem 2

Proof. Let $a_\pi \sim \pi(s)$ and $a_\beta \sim \beta(s)$, we begin by applying the Lipschitz continuity of $Q(s,a)$:

$$|Q(s, a_\beta) - Q(s, a_\pi)| \leq L_Q \cdot \|a_\beta - a_\pi\|_2.$$

The difference in the RHS is typically bounded by the forward KL divergence between these two distributions, i.e., $\|a_\beta - a_\pi\|_2 \leq D(\epsilon)$, where ϵ is the forward KL divergence bound, i.e., $\mathrm{KL}(\beta \| \pi) \leq \epsilon$. This bound provides an upper limit on the discrepancy between the sampled actions from both policies, reflecting how much the policies differ in terms of their action distributions.

We consider the case that $\beta(s) = \sum_{i=1}^{K} w_i \mathcal{N}(\mu_i, \Sigma_i)$, where the mixture weights satisfy $w_i \geq 0$ and $\sum_{i=1}^{K} w_i = 1$. Policy $\pi(s) = \mathcal{N}(\mu_\pi, \Sigma_\pi)$ is a Gaussian. Let $\Lambda_i := \Lambda(\Sigma_i)$ denote the largest eigenvalue of each component covariance Σ_i, and let $\Lambda(\Sigma_\pi)$ denote the largest eigenvalue of Σ_π.

Consider the decomposition using triangle inequality in Lemma 1:

$$\|a_\beta - a_\pi\|_2 \leq \max_i \left(\|a_\beta - \mu_i\|_2 + \|\mu_i - \mu_\pi\|_2 \right) + \|a_\pi - \mu_\pi\|_2.$$

(1) Sampling deviation of GMM: Conditional on sampling component $i \sim w$, the action sample $a_\beta \sim \mathcal{N}(\mu_i, \Sigma_i)$. For each component, we apply Lemma 4 to obtain the following bound:

$$\|a_\beta - \mu_i\|_2 \leq \sqrt{2\Lambda_i \log \tfrac{K}{\delta}} \quad \text{with probability at least } 1-\delta.$$

(2) Mean shift of GMM components: We now bound the deviation between the GMM component means μ_i and the target policy mean μ_π. From the definition of KL divergence between Gaussians:

$$\mathrm{KL}\left(\mathcal{N}(\mu_i, \Sigma_i) \| \mathcal{N}(\mu_\pi, \Sigma_\pi)\right) \geq \frac{1}{2} \|\mu_i - \mu_\pi\|^2_{\Sigma_\pi^{-1}}.$$

Therefore, if we know that the total mixture KL satisfies $\mathrm{KL}(\beta \| \pi) \leq \varepsilon$, then by invoking Lemma 5, we have: for each component i, it must be that

$$\|\mu_i - \mu_\pi\|^2_{\Sigma_\pi^{-1}} \leq 2 \cdot \frac{\varepsilon}{w_i} \quad \Rightarrow \quad \|\mu_i - \mu_\pi\|_2 \leq \sqrt{2\varepsilon \Lambda(\Sigma_\pi)/w_i}.$$

Hence, plugging this into the deviation bound, we now obtain:

$$\|a_\beta - a_\pi\|_2 \leq \max_i \left(\sqrt{2\Lambda_i \log \tfrac{K}{\delta}} + \sqrt{2\varepsilon \Lambda(\Sigma_\pi)/w_i} \right) + \|a_\pi - \mu_\pi\|_2,$$

with high probability at least $1 - \delta$.

(3) Sampling deviation of π: As $a_\pi \sim \mathcal{N}(\mu_\pi, \Sigma_\pi)$, by invoking Lemma 4 again, we have:

$$\|a_\pi - \mu_\pi\|_2 \leq \sqrt{2\Lambda(\Sigma_\pi) \log \tfrac{1}{\delta}}.$$

Putting everything together, with probability at least $1 - \delta$, we have:

$$\|a_\beta - a_\pi\|_2 \leq \max_i \left(\sqrt{2\Lambda_i \log \tfrac{K}{\delta}} + \sqrt{2\varepsilon \Lambda(\Sigma_\pi)/w_i} \right) + \sqrt{2\Lambda(\Sigma_\pi) \log \tfrac{1}{\delta}}.$$

Since both a_β and a_π are supported on the normalized action space $[-1, 1]^d$, the maximum possible distance is bounded by $2\sqrt{d}$. Hence:

$$\|a_\beta - a_\pi\|_2 \leq \min \left(2\sqrt{d},\ \max_i \left(\sqrt{2\Lambda_i \log \tfrac{K}{\delta}} + \sqrt{2\varepsilon \Lambda(\Sigma_\pi)/w_i} \right) + \sqrt{2\Lambda(\Sigma_\pi) \log \tfrac{1}{\delta}} \right).$$

Conclusion. For any $\delta \in (0, 1)$, with probability at least $1 - \delta$, we obtain the bound:

$$D(\varepsilon) \leq \min \left(2\sqrt{d},\ \max_i \left(\sqrt{2\Lambda_i \log \tfrac{K}{\delta}} + \sqrt{2\varepsilon \Lambda(\Sigma_\pi)/w_i} \right) + \sqrt{2\Lambda(\Sigma_\pi) \log \tfrac{1}{\delta}} \right).$$

This completes the proof. $\square$

B Environmental Details

B.1 Benchmark Introduction

DeepMind Control Suite. We evaluate on the 7 most challenging tasks involving the `dog` and `humanoid` agents. These tasks fall into two categories: (1) *Standing tasks*, where the agent must maintain upright balance, and (2) *Moving tasks*, which additionally require the agent to move at a target velocity. For moving tasks, the reward is defined as the product of the standing reward and the forward velocity reward, i.e., ***Reward*** $=$ (***Standing reward***) $\times$ (***Forward velocity reward***).

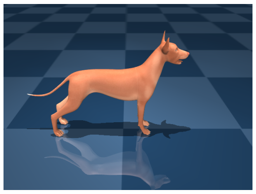


Figure 4: Dog

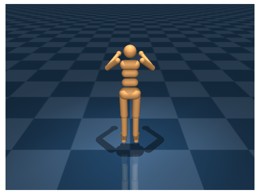


Figure 5: Humanoid

Standing reward: Encourages the agent to maintain an upright posture.

Forward velocity reward: Ensures the agent moves at the target speed ($1\,\mathrm{m/s}$ for dog-walk, $3\,\mathrm{m/s}$ for dog-trot, $9\,\mathrm{m/s}$ for dog-run, $1\,\mathrm{m/s}$ for humanoid-walk and $10\,\mathrm{m/s}$ for humanoid-run).

Humanoid Bench. We consider 7 typical locomotion tasks involving a Unitree H1hand robot. This robot is initialized to a standing position, with random noise added to all joint positions during each episode reset. Their specific goals are presented below.

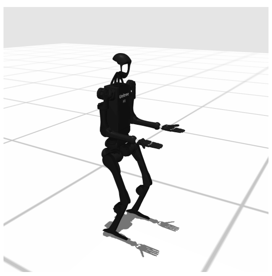


Figure 6: Stand

Objective. Maintain a standing pose.

Reward: $R(s,a) = \texttt{stable} \times (0.5 \times \texttt{still}_x + 0.5 \times \texttt{still}_y)$, where the `still` terms penalize non-zero velocities to encourage stationary balance. `stable` favors maintaining a stable and energy-efficient standing status.

Termination. 1000 steps, or when $z_{\text{pelvis}} < 0.2$.

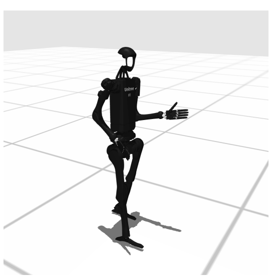


Figure 7: Walk

Objective. Keep forward velocity close to $1\,\mathrm{m/s}$ without falling to the ground.

Reward: $R(s,a) = \texttt{stable} \times \texttt{tol}(v_x, (1, \infty), 1)$, where `tol` encourages the agent to maintain a forward velocity v_x above $1\,\mathrm{m/s}$, thereby promoting low-speed locomotion.

Termination. 1000 steps, or when $z_{\text{pelvis}} < 0.2$.

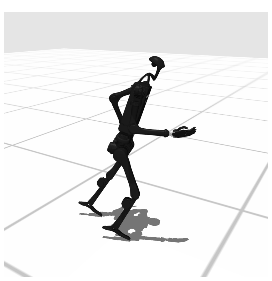


Figure 8: Run

Objective. Keep forward velocity close to $5\,\mathrm{m/s}$ without falling to the ground.

Reward: $R(s,a) = \texttt{stable} \times \texttt{tol}(v_x, (5, \infty), 5)$, where `tol` encourages the agent to maintain a forward velocity v_x above $5\,\mathrm{m/s}$, thereby promoting high-speed locomotion.

Termination. 1000 steps, or when $z_{\text{pelvis}} < 0.2$.

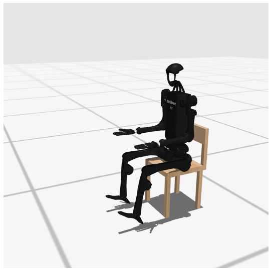


Figure 9: Sit

Objective. Sit onto a chair situated closely behind.

Reward: $R(s, a) = (0.5 \cdot \texttt{sitting\_z} + 0.5 \cdot \texttt{sitting\_x} \cdot \texttt{sitting\_y}) \times \texttt{upright} \times \texttt{posture} \times e \times \texttt{mean}(\texttt{still\_x}, \texttt{still\_y})$, where e is an energy penalty term, `sitting_x`, `sitting_y`, and `sitting_z` measure the robot's positional tolerance relative to the chair.

Termination. 1000 steps, or when $z_{\text{pelvis}} < 0.5$.

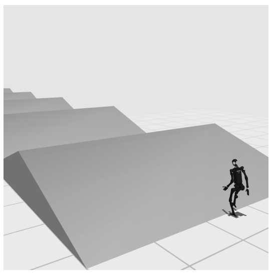


Figure 10: Slide

Objective. Walk over an iterating sequence of upward and downward slides at $1\,\mathrm{m/s}$.

Reward: $R(s, a) = e \times \texttt{tol}(v_x, (1, +\infty), 1) \times \texttt{upright} \times \big(\texttt{foot\_left} \times \texttt{foot\_right}\big)$, where `foot_left` and `foot_right` measure the vertical distance between the head and left/right foot respectively, ensuring proper foot positioning.

Termination. 1000 steps, or when $z_{\text{proj}} < 0.6$.

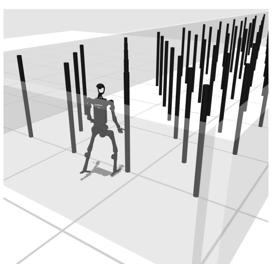


Figure 11: Pole

Objective. Travel forward over a dense forest of high thin poles, without colliding with them.

Reward: $R(s, a) = \gamma_{\text{collision}} \times \big(0.5 \times \text{stable} + 0.5 \times \texttt{tol}(v_x, (1, +\infty), 1)\big)$, where the collision penalty $\gamma_{\text{collision}}$ equals 0.1 if the robot collides with a pole, and 1 otherwise.

Termination. 1000 steps, or when $z_{\text{pelvis}} < 0.6$.

Figure 12: Hurdle

Objective. Keep forward velocity close to $5\,\mathrm{m/s}$ without falling to the ground.

Reward: $R(s, a) = \text{stable} \times \texttt{tol}(v_x, (5, \infty), 5) \times \gamma_{\text{collision}}$, which penalizes colliding with hurdle.

Termination. 1000 steps.

B.2 Reproducibility Statement & Detailed Hyperparameters

We base all our experiments on the released official TD-MPC2 codebase `https://github.com/nicklashansen/tdmpc`. We adopt their hyperparameter settings without additional tuning and use the same configuration across all previously demonstrated tasks. The details are listed in Table 2. Our core algorithm file and video demos for the most challenging hurdle, pole and slide tasks are accessible at `https://anonymous.4open.science/r/NeurIPS_BOOM-C587`.

In this paper, we evaluate each algorithm for each tasks over three random seeds. The CPU used is the AMD Ryzen Threadripper 3960X 24-Core Processor, and the GPU used is NVIDIA GeForce

RTX 3090Ti. Taking *Dog-run* task in the DMC Suite as an example, the time taken to train 2M iterations is around 50 hours.

For SAC and DreamerV3, we report the baseline results released by TD-MPC2, which are obtained from the official repositories of `https://github.com/denisyarats/pytorch_sac` and `https://github.com/danijar/dreamerv3`, respectively. For BMPC, we use the official implementation `https://github.com/wertyuilife2/bmpc` and align the settings such as number of iterations and evaluation protocol for a fair comparison.

Table 2: Hyperparameter settings.

Hyperparameter	**Value**	**Hyperparameter**	**Value**
Training			
Learning rate	3×10^{-4}	Target network update rate	0.5
Encoder learning rate	1×10^{-4}	Discount factor (γ)	0.99
Sample batch size	1	Gradient Clipping Norm	20
Replay batch size	256	Optimizer	Adam
Buffer size	1_000_000	Loss norm	Moving $(5\%, 95\%)$
Steps	2_000_000	Sampling	Uniform
World Model			
Reward loss coefficient (c_r)	0.1	Dynamics loss coefficient (c_f)	20
Value loss coefficient (c_q)	0.1	Value functions esemble	5
Number of value bins	101	Warmup steps	5000
Planner			
MPPI Iterations	6 (8 if $\|\mathcal{A}\|$ > 20)	Minimum planner std	0.05
Population size	512	Maximum planner std	2
Number of elites	64	Horizon	3
Policy prior samples	24		
Actor			
Minimum policy log std	-10	Entropy coefficient (α)	1×10^{-4}
Maximum policy log std	2		
Architecture (around 5M parameters in total)			
Encoder layers	2	Latent space dimension	512
Encoder dimension	256	Task embedding dimension	96
MLP hidden layer dimension	512	Q function drop out rate	0.01
MLP activation	Mish	MLP Normalization	LayerNorm

C Supplemental Results

C.1 Illustrative Discussion: Forward KL vs. Reverse KL

To empirically illustrate the differences between forward and reverse KL objectives in aligning an actor policy with a multimodal planner distribution, we create a toy 1D problem for fitting a mixture of two Gaussians, which simulates a planner policy that captures multiple high-value regions. We initialize a unimodal Gaussian policy $q(x) = \mathcal{N}(x \mid \mu,\ \sigma^2)$, and optimize its parameters by minimizing either the **forward KL** divergence $\mathrm{KL}(p\|q)$ or the **reverse KL** divergence $\mathrm{KL}(q\|p)$.

As shown in Figure 13, the policy trained with **forward KL** consistently adjusts its mean and variance to cover both modes of the planner distribution, demonstrating a *mode-covering* behavior. In contrast, the **reverse KL** objective tends to converge to only one mode of the distribution, often ignoring the global structure. This highlights its *mode-seeking* nature and reveals the risk of it aligning with a *suboptimal peak*, potentially converging to a low-value region.

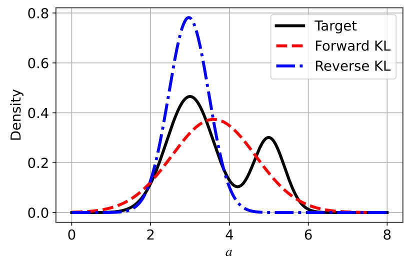
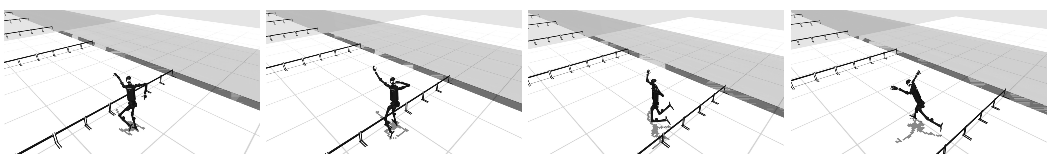


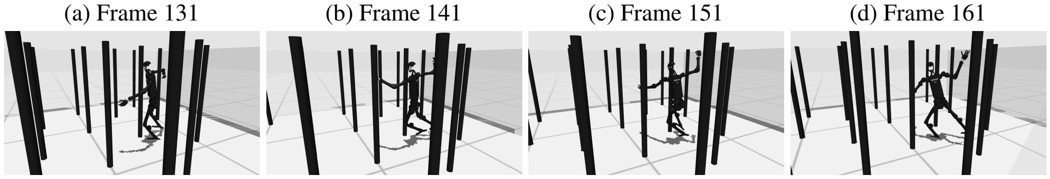
Figure 13: Comparison of forward KL and reverse KL on a fitting task. This example highlights a key issue: when the planner is multimodal but the policy is unimodal, **reverse KL may fail to sufficiently cover high-density regions of the target distribution**, still resulting in significant mismatch and poor performance. In contrast, **forward KL provides a more stable alignment strategy**, promoting broader coverage of the target distribution and preventing premature collapse.

C.2 Visualizations

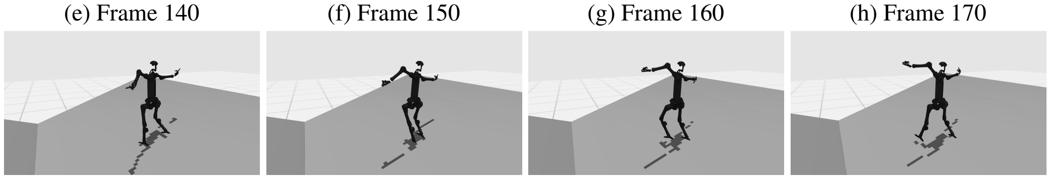
To demonstrate the effectiveness of BOOM in solving complex, high-dimensional locomotion tasks, we provide visualizations of policy control process on three of the most challenging benchmarks in the Humanoid Bench: **hurdle**, **pole**, and **slide** as shown in the following Figure 14. These tasks require precise coordination across many degrees of freedom, long-horizon reasoning, and dynamic interaction with many objects. The visualization showcase that BOOM not only achieves task success but also learns robust behaviors, highlighting its strong capabilities in difficult control scenarios.



(a) Frame 131 (b) Frame 141 (c) Frame 151 (d) Frame 161



(e) Frame 140 (f) Frame 150 (g) Frame 160 (h) Frame 170



(i) Frame 296 (j) Frame 301 (k) Frame 306 (l) Frame 311

Figure 14: Visualizations of BOOM solving three of the most challenging tasks in the Humanoid Bench—**hurdle**, **pole**, and **slide**.

D Limitation and Future Work

While BOOM demonstrates strong performance in high-dimensional continuous control, several limitations remain. First, the use of a planner during data collection introduces additional computational overhead, as generating and optimizing candidate actions through the world model is significantly slower than direct sampling in model-free methods. This can limit overall training throughput, particularly when environment interaction time is not the dominant bottleneck. Second, BOOM relies on a reasonably accurate dynamics model throughout training. When the model suffers from large prediction errors or distribution shift, especially over longer rollouts, both planning quality and value estimation may degrade, potentially harming overall performance.

Future work may explore accelerating the computation of online planning process and incorporating uncertainty-aware mechanisms to better handle multimodal or unreliable planner outputs. Additionally, extending BOOM to sparse-reward or real-world robotics settings with noisy observations also presents a promising and exciting direction.

E Positive and Negative Social Impact

Our method, BOOM, enhances the integration of online planning and off-policy learning in RL, leading to improved sample efficiency and final performance in high-dimensional control tasks. This has positive implications for real-world applications such as robotics and autonomous systems, where efficient and stable learning is crucial. By leveraging world models to reduce reliance on physical trials, our approach may also contribute to safer and more cost-effective training processes. However, like many RL technologies, BOOM could be misapplied in sensitive domains such as surveillance or autonomous weapons. Additionally, improved simulation-based efficiency might encourage premature deployment in safety-critical settings. We recommend cautious evaluation and responsible use to ensure the technology is applied ethically.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract clearly outlines the key contributions and scope of the paper. It accurately presents (1) the motivation—divergence between planner and policy in planning-driven model-based RL, (2) the proposed method—BOOM, which tightly integrates planning and policy learning via a bootstrap loop, and (3) the novel techniques—likelihood-free divergence loss and soft value-weighted mechanism. It also specifies the evaluation domains (DeepMind Control Suite and Humanoid-Bench) and the claimed results (state-of-the-art performance in sample efficiency and final return). These elements are consistent with the core contributions described in the main paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper explicitly discusses several limitations of the proposed approach in Appendix D. It acknowledges that the use of a planner during data collection increases computational overhead compared to model-free methods, potentially limiting training throughput. Additionally, the method's reliance on a reasonably accurate dynamics model is noted as a limitation—model errors or distribution shift can degrade both planning and value estimation, affecting performance. These points demonstrate a clear and honest discussion of the method's constraints.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.

- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper provides two formal theorems that analyze the theoretical guarantees of Bootstrap Alignment. For each result, we clearly state the assumptions (e.g., bounded reward, discount factor, forward KL divergence bound, Lipschitz continuity of the Q-function), and the statements are mathematically precise. While the main proofs are deferred to the appendix, the presentation in the main paper outlines the core logic and implications of the theorems.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper includes all necessary details to reproduce the main experimental results: it provides comprehensive descriptions of the benchmarks in Appendix B.1, and full hyperparameter settings in B.2. Our core algorithm file is accessible at `https://anonymous.4open.science/r/NeurIPS_BOOM-C587`. These elements together ensure that readers can independently verify and reproduce the main claims and conclusions, satisfying the reproducibility criterion.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed

instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: This paper provides training curves, and numerical results (Section 4). Additionally, it offers a public repository `https://github.com/molumitu/BOOM_MBRL` containing the core implementation of the proposed algorithm. These elements together ensure that readers can independently verify and reproduce the main claims and conclusions, satisfying the reproducibility criterion.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

Justification: The paper specifies the training and test setup in detail, including benchmark environments (Appendix B.1), full hyperparameter settings (Appendix B.2), and training curves (Figure 2). It clearly outlines choices such as the type of optimizer used, learning rates, horizon lengths for planning, and network structures. Hyperparameter values and their selection process are disclosed, allowing readers to understand and contextualize the reported performance.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper includes training curves with error bars that reflect the variability across 3 random seeds in Figure 2, which is standard and appropriate for RL benchmarks. It also reports mean and standard deviation values for final performance metrics in Table 1, providing a clear sense of the statistical reliability of the results. These practices offer sufficient information about the significance and robustness of the experimental findings.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides sufficient information on the computational setup used for the experiments in Section 4, including the type of compute workers (e.g., GPU/CPU), memory specifications, and total training time.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We made sure the code was anonymous `https://anonymous.4open.science/r/NeurIPS_BOOM-C587`.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: In Appendix E, we discuss the potential positive and negative social impacts of our work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper properly credits all external assets used, such as open-source repositories including Pytorch-RL, DreamerV3 and TD-MPC2, by citing the original authors (Appendix B.2). These assets are used in accordance with their licenses and terms of use. Proper attribution ensures ethical reuse of resources and acknowledges the contributions of prior work, fulfilling this requirement.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The paper introduces new assets, including a core implementation file and demonstration videos on three challenging tasks (`https://anonymous.4open.science/r/NeurIPS_BOOM-C587`). These assets effectively showcase the method's capabilities and support the main experimental claims. Upon acceptance, we will release the full codebase along with additional demos.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.