RETHINKING SCALABLE PLANNING PARADIGMS FOR HIGH-DIMENSIONAL CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time planning with learned world models has become a common practice in model-based reinforcement learning in recent years, yet mixed empirical results and a principled limitation arising from actor–planner divergence have also been reported. We revisit whether test-time planning should be a default component of high-dimensional control and re-examine policy improvement through background planning on imagined rollouts. By combining on-policy background planning with an annealed curiosity reward, our method outperforms the strongest prior high-dimensional control method under the matched environment interaction budget, improving mean performance by 8.57% on DMC and 9.99% on HumanoidBench. We further show that curiosity-driven exploration yields larger gains in high-dimensional and sparse-reward settings than either test-time planning or planner-action imitation. In contrast, the benefits of test-time planning and action imitation depend strongly on the task and temperature hyperparameters and are not consistently positive. We conduct an extensive hyperparameter sweep over planner and imitation temperatures. Across the 84 task–configuration cells involving a test-time planner, 21 and 15 yield relative improvements above 10% and relative degradations below -10%, respectively. Based on a correlational analysis, we formulate a causal hypothesis consistent with these observations: early intervention by the test-time planner, when the world model and value function remain underfitted, may degrade action quality and reduce experience diversity in the replay buffer. Together, these results suggest that, under our unified setting, on-policy background planning with curiosity-driven exploration provides a more reliable route to high-dimensional policy improvement than test-time planning.

1 INTRODUCTION

Random exploration allows reinforcement learning (RL) to discover behaviors beyond demonstration data, but becomes increasingly inefficient in high-dimensional continuous control. As the number of control dimensions grows, useful actions occupy a rapidly shrinking fraction of the joint action space, making high-value behaviors unlikely under maximum-entropy-driven random exploration.

Learned world models offer two ways to support policy improvement in these settings. *Test-time planning* evaluates candidate futures to select actions during environment interaction, whereas *background planning* optimizes an actor on imagined trajectories. These two give rise to two fundamental paradigms of model-based reinforcement learning (MBRL).

Planning-driven and imagination-driven agents have both achieved strong control performance, while the former has recently become the dominant paradigm for high-dimensional control. TD-MPC-style methods Hansen et al. (2022; 2024); Wang et al. (2025); Zhan et al. (2025) have shown that such planners can consistently improve action selection within the online RL policy-improvement loop for high-dimensional continuous control. However, planner-generated behavior is inherently off-policy with respect to the actor being optimized: the non-parametric planner and the parametric actor generally induce different behavior distributions and value estimates. The resulting *actor–planner divergence* has been shown to adversely affect policy improvement in BOOM Zhan et al. (2025). BOOM further mitigates this mismatch through an advantage-weighted regression (AWR)-style policy-alignment objective, leading to substantial improvements in RL performance. MRS.Q Chang et al. (2026) further shows that *actor–planner divergence* can induce substantial value overestimation,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

and that explicitly mitigating this bias yields consistent performance improvements. These findings expose *actor–planner divergence* as a principled limitation for test-time planning. Our work takes an empirical perspective and studies when and why test-time planning succeeds or fails under the actual learning dynamics of deep RL.

In contrast, *background planning* improves the policy using synthesized on-policy trajectories, avoiding *actor–planner divergence* and the high-variance importance sampling required for off-policy replay reuse. The Dreamer family Hafner et al. (2019; 2020; 2022; 2025a); Bredis et al. (2026) has shown that imagined policy gradients can achieve strong cross-domain performance with fixed hyperparameters. The limited performance of *background planning* in high-dimensional control has often been attributed to inefficient exploration by purely feedforward Gaussian policies. In contrast, we find that a primary bottleneck of Dreamer variants in high-dimensional control lies in the overly restrictive information bottleneck induced by their default categorical stochastic latent distributions. Once this bottleneck is alleviated, even planner-free feedforward policies can achieve competitive performance in high-dimensional control.

Both planning paradigms use a learned world model to improve RL sample efficiency by scaling the computation budget. However, they can still struggle in environments where rewards are highly sparse. Curiosity-based intrinsic rewards provide an exploration mechanism orthogonal to test-time planning. Inspired by prediction-error-based intrinsic motivation Pathak et al. (2017); Burda et al. (2018), we construct surprise signals from world-model dynamics- and reward-prediction errors, providing structured exploration beyond the unstructured stochasticity induced by maximum-entropy objectives. These intrinsic rewards encourage the agent to visit poorly modeled and under-covered regions of the state space. Our curiosity-driven policy improvement operates entirely in imagination, similar to Plan2Explore Sekar et al. (2020). However, instead of using ensemble disagreement to quantify novelty, we directly use the simpler world-model losses as surprise signals. Moreover, rather than separating training into two stages of purely self-supervised exploration and few-shot task adaptation, we integrate exploration into online RL through an annealing mechanism.

Test-time planning can also be combined with *background planning*, raising the question of what additional benefit action search provides to an imagination-trained actor. This work therefore asks the following question: *within a unified MBRL framework with fixed cross-task hyperparameters, which mechanism most reliably mitigates the curse of dimensionality in high-dimensional continuous control?* To avoid confounding effects in our controlled comparisons, we keep the RSSM backbone, imagined actor–critic, and major optimization settings fixed across our variants. We then evaluate our method on the main benchmarks and conduct a planning-only comparison, a planner-imitation temperature sweep, and targeted ablations.

Our main contributions are as follows:

1. **State-of-the-art (SOTA) high-dimensional control under a 4M-step interaction budget.** Across our main benchmarks, our method achieves average scores of 923.5 and 885.0 on DMC and HumanoidBench, outperforming the previous SOTA under the same 4M-step evaluation protocol, BOOM, by 8.57% and 9.99%, respectively. Recent methods including MRS.Q Chang et al. (2026) and EfficientTDMPC Evers et al. (2026) report strong results in the low-data regime ($\leq$1M environment steps). Since our study targets final performance under a 4M-interaction budget, we do not directly compare aggregate scores across these different evaluation regimes.

2. **An annealed curiosity reward for efficient exploration.** Dynamics-prediction surprise and reward-prediction surprise are normalized online and incorporated into the imagined return, with the intrinsic-reward weight annealed to zero according to extrinsic-return progress. We validate the contribution of curiosity-driven exploration through comparisons with the strongest prior planning-based baseline and targeted ablation studies.

3. **A systematic study of test-time planning and planner imitation.** The test-time planner resamples the actions during environment interaction, whereas planner imitation learning applies advantage-weighted regression to historical planner actions. We perform a systematic hyperparameter sweep, showing that the benefits of planning and imitation vary substantially across tasks and temperatures, rather than providing consistent improvements.

2 PRELIMINARIES

2.1 IMAGINATION-DRIVEN MBRL

The goal of RL is to maximize the expected cumulative return. We consider an entropy-regularized objective that augments the reward signal with the entropy of the policy distribution: $J_\pi = \mathbb{E}_{\tau=(s_{i\geq t}, a_{i\geq t})\sim p^\pi(\tau)}\left[\sum_{i=t}^{\infty}\gamma^{i-t}\left[r_i + \alpha\mathcal{H}(\pi(\cdot|s_i))\right]\right]$, where $\gamma \in (0,1)$ is the discount factor, α is the temperature coefficient, and the policy entropy $\mathcal{H}(\pi(\cdot|s)) = \mathbb{E}_{a\sim\pi(\cdot|s)}\left[-\log\pi(a|s)\right]$.

We use a Recurrent State-Space Model (RSSM) to represent the environment. The model state is defined as $s_t = \{h_t, z_t\}$, where h_t is a deterministic state recursively computed from historical states and actions, and z_t is a stochastic posterior state conditioned on the current observation. The RSSM and auxiliary prediction heads are given by

$$\begin{aligned}
&\text{RSSM} \quad \begin{cases} \text{Sequence Model:} & h_t = f_\phi(h_{t-1}, z_{t-1}, a_{t-1}) \\ \text{Causal Encoder:} & z_t \sim q_\phi(\cdot \mid h_t, x_t) \\ \text{Dynamics Predictor:} & \hat{z}_t \sim p_\phi(\cdot \mid h_t) \\ \text{Causal Decoder:} & \hat{x}_t \sim p_\phi(\cdot \mid h_t, z_t) \end{cases} \\
&\text{Heads} \quad \begin{cases} \text{Reward Predictor:} & \hat{r}_t \sim p_\phi(\cdot \mid h_t, z_t) \\ \text{Continue Predictor:} & \hat{c}_t \sim p_\phi(\cdot \mid h_t, z_t) \end{cases}
\end{aligned} \tag{1}$$

where x_t denotes the observation, a_t the action, and ϕ the world-model parameters. The posterior encoder q_ϕ incorporates the current observation, whereas the dynamics prior p_ϕ does not access future observations. The variable c_t indicates whether the episode continues. The world model is trained on replay trajectories; once sufficiently trained, its dynamics prior can recursively generate latent trajectories without access to future real observations. The main world-model losses are

$$\begin{aligned}
\mathcal{L}_{\text{pred}} &= -\log p_\phi(x_t \mid z_t, h_t) - \log p_\phi(c_t \mid z_t, h_t), \\
\mathcal{L}_{\text{rew}} &= -\log p_\phi(r_t \mid z_t, h_t), \\
\mathcal{L}_{\text{dyn}} &= \mathrm{KL}\left[\mathrm{sg}(q_\phi(z_t \mid h_t, x_t)) \,\|\, p_\phi(z_t \mid h_t)\right], \\
\mathcal{L}_{\text{rep}} &= \mathrm{KL}\left[q_\phi(z_t \mid h_t, x_t) \,\|\, \mathrm{sg}(p_\phi(z_t \mid h_t))\right].
\end{aligned} \tag{2}$$

Here, sg denotes stop-gradient: the forward value remains unchanged, while the corresponding branch receives no backward gradient. The prediction loss prevents representation collapse, the dynamics loss trains the prior to predict future latent states, and the representation loss encourages consistency between the posterior and prior.

An imagined trajectory τ starts from a latent state sampled from the replay buffer and is generated by an imagined RSSM rollout: $p(\tau) = p_{\text{buffer}}(s_0)\prod_{t=0}^{H-1}\pi_\theta(a_t \mid s_t)\, p_\phi(s_{t+1} \mid s_t, a_t)$.

Through imagination, the RSSM provides a continuous stream of on-policy trajectories for policy improvement. The actor $a_t \sim \pi_\theta(\cdot \mid s_t)$ and critic $v_t \sim v_\psi(\cdot \mid s_t)$ are updated purely on imagined latent trajectories, without requiring importance sampling for off-policy correction. The bootstrap λ-return is used as the value target:

$$R_t^\lambda = r_t + \gamma c_t\left((1-\lambda)v_t + \lambda R_{t+1}^\lambda\right), (0 \leq t < T), \quad R_T^\lambda = v_T \tag{3}$$

where v_t is the expected critic value $\mathbb{E}\left[v_\psi(\cdot|s_t)\right]$. The actor-critic losses are:

$$\begin{aligned}
\mathcal{L}_{\text{val}}(\psi) &= -\sum_{t=1}^{T}\ln p_\psi(R_t^\lambda \mid s_t), \\
\mathcal{L}_{\text{pol}}(\theta) &= -\sum_{t=1}^{T}\left[\frac{R_t^\lambda - \mathrm{sg}\left(v_\psi(s_t)\right)}{\max(1, S)}\log\pi_\theta(a_t \mid s_t) + \alpha\mathcal{H}\left[\pi_\theta(\cdot \mid s_t)\right]\right],
\end{aligned} \tag{4}$$

where $S = \mathrm{EMA}\left(\mathrm{Per}(R_t^\lambda, 95) - \mathrm{Per}(R_t^\lambda, 5)\right)$ is the return scale between the 5% and 95% quantiles, maintained by an exponential moving average. It is clipped to ≥ 1 to avoid amplifying noise.

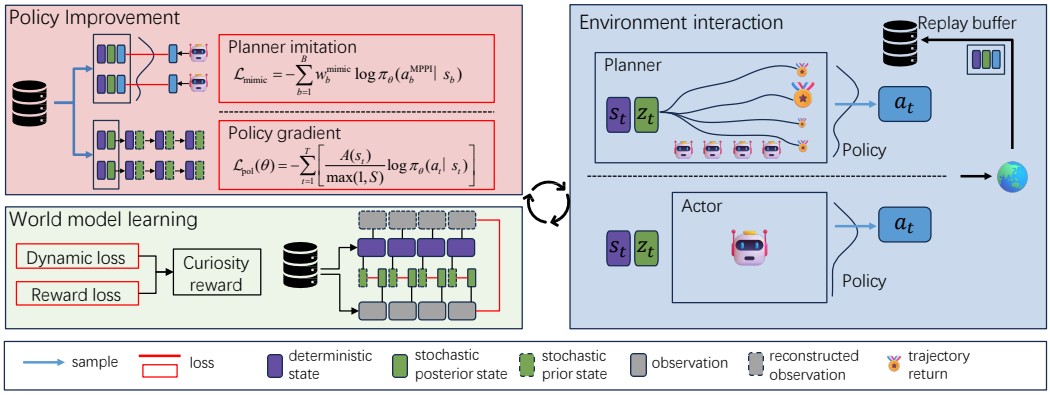

Figure 1: **Overview of the proposed framework.** Our main motivation is to evaluate two scalable policy improvement paradigms, background planning and test-time planning, under a unified setting. The world model learns environment dynamics from the replay buffer, providing the foundation for both planning paradigms. The normalized dynamics loss and reward prediction loss provide intrinsic exploration rewards for novel states and unexpected reward signals.

3 METHODS

3.1 CURIOSITY-DRIVEN EXPLORATION

Maximum-entropy-driven exploration becomes increasingly inefficient as the action space expands, because unstructured stochastic exploration suffers from the curse of dimensionality. Curiosity-driven exploration via self-supervised prediction incorporates novelty signals as intrinsic motivation through reward shaping.

For the RSSM in Eqs. 1 and 2, we derive candidate surprise signals from the dynamics, reconstruction, and reward-prediction errors. We consider $\mathcal{L}_{\text{dyn}}$, $\mathcal{L}_{\text{pred}}$, and $\mathcal{L}_{\text{rew}}$ as candidate novelty signals to drive exploration, as larger prediction errors can indicate novel or poorly modeled states. $\mathcal{L}_{\text{dyn}}$ is the latent space prediction loss, indicating the surprise of the encoded posterior. $\mathcal{L}_{\text{rew}}$ can be directly evaluated via environment feedback and contains task-related surprise signals. Since our benchmarks use proprioceptive rather than high-dimensional visual observations, the classical *noisy-TV* failure mode is less prominent in our setting. We nevertheless exclude $\mathcal{L}_{\text{pred}}$, because reconstruction errors can be sensitive to unpredictable observation noise. In contrast, $\mathcal{L}_{\text{dyn}}$ operates in a regularized latent space and is therefore less directly exposed to observation-level noise.

The surprise signal predictors $r_t^{\text{dyn}} \sim p_\phi(\cdot \mid s_t)$, $r_t^{\text{rew}} \sim p_\phi(\cdot \mid s_t)$ are trained to predict the dynamics and reward-prediction losses:

$$\mathcal{L}_{\text{surprise}}(\phi) = -\ln p_\phi(\mathcal{L}_{\text{dyn}} \mid s_t) - \ln p_\phi(\mathcal{L}_{\text{rew}} \mid s_t). \tag{5}$$

In online RL, persistent intrinsic rewards may bias policy optimization away from the extrinsic objective. We address this issue with a simple annealing mechanism that decreases the intrinsic-reward weight according to the running median of the estimated extrinsic λ-return and eventually reduces it to zero. The total shaped reward includes extrinsic task reward and normalized intrinsic exploration reward:

$$\begin{aligned} &\text{Shaped Reward:} \quad r_t^{\text{shaped}} = r_t + w_t^{\text{intrinsic}} \cdot \left(\text{Norm}^{\text{dyn}}(r_t^{\text{dyn}}) + \text{Norm}^{\text{rew}}(r_t^{\text{rew}}) \right), \\ &\text{Annealing Weights:} \quad w_t^{\text{intrinsic}} = 0.05 \cdot \text{clip}\left(1 - \frac{\text{EMA}\left(\text{Per}(R_t^\lambda, 50)\right)}{0.2 \cdot R_t^{\text{ub}}}, \quad 0, \quad 1 \right) \end{aligned} \tag{6}$$

where $\text{EMA}\left(\text{Per}(R_t^\lambda, 50)\right)$ tracks the median of the estimated λ-return. Given that all tasks feature per-step rewards normalized to $[0, 1]$ and a discount factor of $\gamma = 1 - 1/333$, the maximum return is uniformly bounded at $R_t^{\text{ub}} = 333$. The annealing weight $w_t^{\text{intrinsic}} \in [0, 0.05]$ is scaled by a

factor of $0.05 \ll 1$ to avoid the dominance of exploration reward in the middle and late stages of training. $\mathrm{Norm}^{\mathrm{dyn}}$ and $\mathrm{Norm}^{\mathrm{rew}}$ use EMA to maintain the mean and variance of their novelty signals, standardizing them to have approximately zero mean and unit variance. We provide a sensitivity analysis of the progress threshold for annealing the exploration reward to zero in Appendix A.3.

3.2 MPPI-Style Test-Time Planning

The MPPI-Style planner samples candidate trajectories of horizon H_p from a model predictive rollout: $\tau_n \sim p^\pi(\tau \mid s_0) = \prod_{t=0}^{H_p-1} \pi_\theta(a_t \mid s_t) p_\phi(s_{t+1} \mid s_t, a_t), \quad n = 1, \ldots, N$, where s_0 is the current latent state and N is the number of samples. The candidate trajectories are evaluated with the bootstrap λ-return $G(\tau_n) = R_0^\lambda$. The KL-regularized return-maximization problem can be formalized as $p^* = \arg\max_q \left[\mathbb{E}_{\tau\sim q}[G(\tau)] - \xi^{\mathrm{planner}} \mathrm{KL}(q \parallel p^\pi)\right]$. This optimization problem admits an analytical closed-form solution: $p^*(\tau_n \mid s_0) \propto p^\pi(\tau_n \mid s_0) \exp\left(G(\tau_n)/\xi^{\mathrm{planner}}\right)$.

However, directly applying the bootstrap λ-return makes the posterior weights sensitive to return scales across different tasks and training stages. To alleviate this scale sensitivity, we normalize the returns across N candidate trajectories originating from s_0:

$$p^{\mathrm{MPPI}}(\tau_n \mid s_0) \propto p^\pi(\tau_n \mid s_0) \exp\left(\frac{(G(\tau_n) - \mu_N)/\sigma_N}{\xi^{\mathrm{planner}}}\right), \tag{7}$$

where $\mu_N = \frac{1}{N}\sum_{n=1}^N G(\tau_n)$ and $\sigma_N^2 = \frac{1}{N}\sum_{n=1}^N (G(\tau_n) - \mu_N)^2$. The probability assigned to each candidate trajectory by the posterior distribution is given by:

$$p^{\mathrm{MPPI}}(\tau_n \mid s_0) = w_n^{\mathrm{planner}} = \mathrm{softmax}\left(\frac{(G(\tau_n) - \mu_N)/\sigma_N}{\xi^{\mathrm{planner}}}\right). \tag{8}$$

A planner trajectory is sampled from $p^{\mathrm{MPPI}}(\tau_n \mid s_0)$ and its initial action a_0^{MPPI} is executed. Rather than being a greedy deterministic argmax or a weighted average of actions, this process represents trajectory-level sampling from the MPPI-style posterior, thereby reconciling stochastic exploration during environment interactions with a preference for high-return trajectories.

3.3 AWR-Style Actor–Planner Alignment

For state and planner-action pairs $(s_b, a_b^{\mathrm{MPPI}}), b = 1, \ldots, B$ sampled from the replay buffer with batch size B, we evaluate the action-value function Q using a one-step model transition followed by the bootstrap λ-return. We denote the timestep of s_b as 0 and the next latent state as $s_1 = (h_1, \hat{z}_1)$, where $h_1 = f_\phi(s_b, a_b^{\mathrm{MPPI}})$ and $\hat{z}_1 \sim p_\phi(\cdot \mid h_1)$. The action value of the planner action is approximated by:

$$Q(s_b, a_b^{\mathrm{MPPI}}) \approx \hat{r}_0(s_b) + \gamma R_1^\lambda(s_1). \tag{9}$$

Applying a softmax operation across the entire batch yields the imitation weight for a_b^{MPPI}:

$$w_b^{\mathrm{mimic}} = \mathrm{softmax}\left(\frac{(A_b^{\mathrm{MPPI}} - \mu_A)/\sigma_A}{\xi^{\mathrm{mimic}}}\right), \tag{10}$$

where $A_b^{\mathrm{MPPI}} = Q(s_b, a_b^{\mathrm{MPPI}}) - R_0^\lambda(s_b)$ is the online-evaluated advantage of a_b^{MPPI}. The batch of planner actions competes for softmax weights, following the Soft Q-weighted mechanism introduced in BOOM Zhan et al. (2025). The resulting planner action imitation loss is given by:

$$\mathcal{L}_{\mathrm{mimic}} = -\sum_{b=1}^{B} w_b^{\mathrm{mimic}} \log \pi_\theta(a_b^{\mathrm{MPPI}} \mid s_b). \tag{11}$$

This objective is added on top of the original imagination-based policy-gradient objective.

4 Experiments

4.1 Main Comparison

Benchmarks. We mainly focus on high-dimensional continuous control tasks. We choose 5 tasks from the **DeepMind Control Suite (DMC)** Tassa et al. (2018) featuring its two most complex

agents—humanoid and dog—visualized in Figure 5. **HumanoidBench** Sferrazza et al. (2024) is a high-dimensional robot learning benchmark featuring a humanoid robot equipped with dexterous hands across a variety of challenging whole-body manipulation and locomotion tasks. We choose 6 locomotion tasks from HumanoidBench, including the most challenging H1hand-hurdle, where the robot must sprint and hurdle without losing balance.

Baselines. We choose four online MBRL algorithms, including BOOM Zhan et al. (2025), the previous SOTA for high-dimensional continuous control under the matched 4M-step protocol. Table 1 compares key concepts of our method and baselines. **DreamerV3** is a strong general-purpose baseline

Table 1: Conceptual comparison of recent MBRL algorithms.

	DreamerV3	TD-MPC2	BMPC	BOOM	Ours
Test-time planner	✗	✓	✓	✓	✗
Planner imitation loss	✗	✗	✓	✓	✗
Policy gradient loss	✓	✓	✗	✓	✓
Latent distribution	multi-categorical	-	-	-	Gaussian

A '-' in the Latent distribution row indicates that the latent state is deterministic.

for imagination-driven MBRL; its design is described in detail in the preliminaries. **TD-MPC2** Hansen et al. (2024) is an improved version of the classic planning-driven MBRL TD-MPC Hansen et al. (2022), which interacts with the environment using online planner behavior while optimizing the policy via max-Q learning. **BMPC** Wang et al. (2025) trains the policy solely to imitate planner behavior, using a relabeling mechanism to asynchronously update historical actions in the replay buffer with the latest planner. **BOOM** Zhan et al. (2025) combines off-policy planner imitation and policy gradients. Using its *soft value-weighted mechanism*, planner imitation prioritizes high-payoff behaviors and filters out historical low-payoff ones, akin to the weighting mechanism of AWR. Detailed hyperparameters and reproducibility statements are provided in Appendix C.

The numerical results are listed in Table 2. The training curves are shown in Figure 5 of Appendix A.1. Using a purely feedforward policy head, our method outperforms planning-based MBRL methods and consistently delivers the best or second-best score across all 11 high-dimensional tasks.

Results on the DMC Suite. Our method achieves the highest score among the compared methods on all five DMC tasks. We achieve an average score of **923.5**, outperforming the previous best BOOM (850.6) by **+8.57%**. On the most difficult humanoid-run, our method leads by a substantial **+48.5%**.

Results on the HumanoidBench. Overall, our method achieves an average score of **885.0**, leading the previous best 804.6 by **+9.99%**. Our method also achieves the highest scores among the compared methods on H1hand-run and H1hand-hurdle, improving over BOOM by **+30.30%** and **+94.93%** respectively.

Table 2: Numerical results on 5 DMC Suite tasks and 6 HumanoidBench (H-Bench) tasks. Results for our method are reported as Mean ± Std over 5 seeds. **Bold** = best, underlined = second-best. Higher is better.

Task	DreamerV3	TD-MPC2	BMPC	BOOM	Ours
Humanoid-walk	251.5 ± 35.2	884.8 ± 8.3	935.1 ± 4.0	936.1 ± 3.3	**954.1** ± 6.7
Humanoid-run	62.5 ± 25.8	316.2 ± 9.2	531.2 ± 42.0	582.8 ± 26.0	**865.5** ± 16.9
Dog-walk	9.1 ± 0.6	885.0 ± 74.8	942.9 ± 9.6	965.4 ± 0.3	**971.3** ± 4.5
Dog-trot	7.9 ± 0.7	884.4 ± 22.2	911.3 ± 18.2	947.9 ± 4.7	**962.8** ± 3.1
Dog-run	4.3 ± 3.2	427.0 ± 57.9	673.7 ± 50.2	820.7 ± 23.0	**863.3** ± 15.8
AVG. DMC Suite	67.1	679.5	798.8	850.6	**923.5**
H1hand-walk	161.3 ± 44.5	644.2 ± 281.1	672.6 ± 10.4	**935.4** ± 7.3	912.0 ± 6.3
H1hand-run	55.8 ± 10.5	66.1 ± 8.1	236.0 ± 53.9	682.2 ± 120.6	**888.9** ± 16.2
H1hand-sit	687.3 ± 138.0	693.7 ± 249.9	688.2 ± 46.3	918.1 ± 4.2	**920.0** ± 12.3
H1hand-slide	162.6 ± 29.5	141.3 ± 15.6	440.1 ± 25.4	**926.1** ± 8.0	840.6 ± 25.6
H1hand-pole	334.3 ± 65.1	207.5 ± 35.6	739.9 ± 18.0	**930.5** ± 18.9	899.0 ± 37.7
H1hand-hurdle	26.6 ± 3.0	59.0 ± 19.3	197.1 ± 12.1	435.6 ± 29.8	**849.1** ± 23.6
AVG. H-Bench.	238.0	302.0	495.7	804.6	**885.0**

4.2 Ablation Study

We use *DMC Humanoid* as a representative embodiment for ablation, as shown in Figure 2. DMC Humanoid combines high-dimensional action spaces with sparse rewards. Since the humanoid agent is initialized in a random non-standing position, the agent receives almost no effective task reward feedback before learning to stand up.

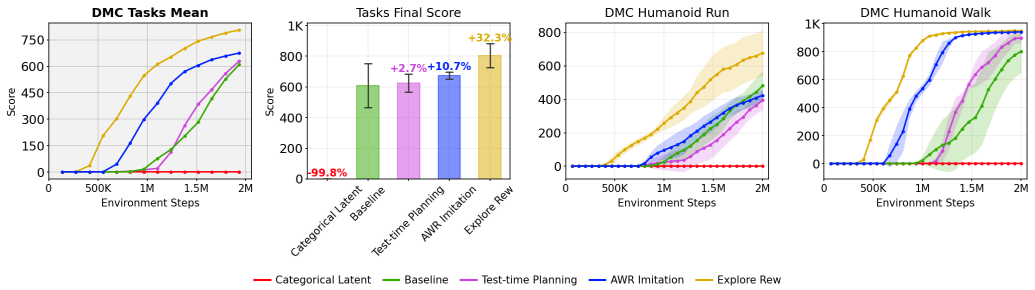

Figure 2: **Ablation Study.** We adopt the background-planning variant that replaces the categorical latent variable with a Gaussian latent variable as the baseline. For the planner and AWR imitation, we use the best temperature hyperparameters $\xi^{\text{planner}} = 0.4$ and $\xi^{\text{mimic}} = 0.5$ based on the *planner analysis* results after excluding the outlier tasks.

Notably, replacing the continuous latent variable with a multi-categorical random variable causes performance to nearly collapse to zero, despite using a slightly larger parameter budget. This sharp degradation suggests that the multi-categorical parameterization can be a major bottleneck for Dreamer-style agents in our high-dimensional control setting.

The exploration reward improves the aggregate DMC score by **+32.3%** over the Baseline. In contrast, the test-time planner and AWR imitation configurations change the aggregate performance by only **+2.7%** and **+10.7%**. These results support the use of annealed curiosity-based intrinsic rewards as an effective cold-start mechanism for overcoming sparse reward signals during early exploration.

4.3 Planner Analysis

Incorporating a test-time planner during environment interactions has the potential to improve the trajectory quality of the replay buffer with optimized planner actions. Concurrently, replay-based Advantage-Weighted Regression (AWR) on planner actions is designed to extract effective policy-improvement signals discovered by the planner while mitigating actor–planner divergence. In this section, we empirically investigate the impact of these components on sample efficiency via extensive hyperparameter sweeps across a diverse set of tasks.

We conduct a *planner analysis* across 7 DeepMind Control (DMC) suite tasks over the MPPI temperature $\xi^{\text{planner}} \in \{0.4, 0.5, 0.6\}$ and the AWR imitation temperature $\xi^{\text{mimic}} \in \{0.4, 0.5, 0.6\}$. In addition, we evaluate planner-only configurations across the same three temperatures with AWR mimic disabled, yielding a total of 84 task–configuration cells. The relative performance of each configuration is evaluated using the late-training episode return relative to the baseline.

Configurations with improvements above 10% occur on *Acrobot Swingup*, *Finger Spin*, *Hopper Hop*, and *Walker Run*, while configurations with degradations below -10% occur on *Hopper Hop*, *Quadruped Run*, and *Walker Run*. The highest gain occurs on *Finger Spin* with a **+99.5%** improvement, whereas the most severe degradation occurs on *Hopper Hop* with a **-35.0%** decline.

When using a shared temperature across all tasks, the best planner-only configuration ($\xi^{\text{planner}} = 0.5$) achieves a mean task improvement of **+9.2%**, closely matching the best mimic configuration ($\xi^{\text{planner}} = 0.5, \xi^{\text{mimic}} = 0.6$) at **+10.1%**. The aggregate improvements are predominantly driven by a small subset of tasks with large improvements such as *Finger Spin*. Furthermore, excluding the two tasks with the largest aggregate improvement and decline, the aggregated relative performance range is from **-4.5%** to **+3.7%**, precluding any claim of universal efficacy for AWR imitation alignment. As shown in Figure 8, after removing the two tasks that dominate relative performance, the best hyperparameter configuration is ($\xi^{\text{planner}} = 0.4, \xi^{\text{mimic}} = 0.5$).

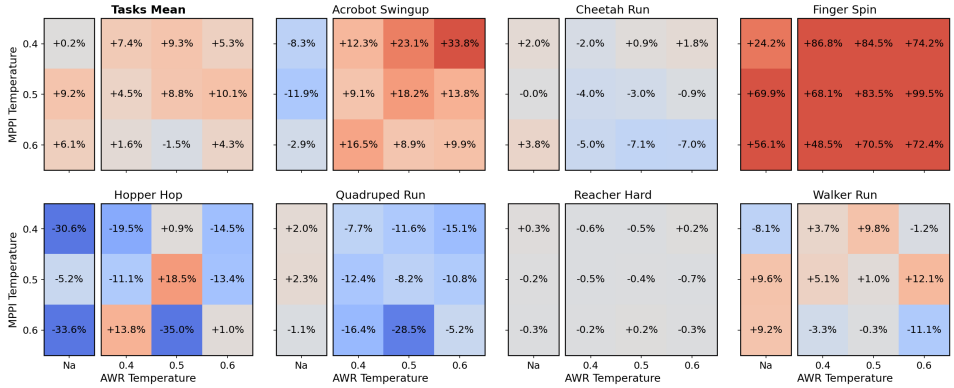

Figure 3: **Planner hyperparameters heatmap.** The heatmap shows the relative performance changes for MPPI and AWR temperature settings across 7 tasks $\times$12 hyperparameter configurations. 21 and 15 cells yield relative improvements above 10% and relative degradations below -10%, respectively. The training curves are shown in Figure 6.

To examine underlying mechanisms, we conduct an *a posteriori* stratification across 63 AWR imitation task–configuration cells into an **improved group** (18 configurations, $>$ **+10%**) and a **degraded group** (12 configurations, $<$ **-10%**). Throughout training, we track the *planner effective sample size (ESS)*, *planner-action-weighted advantage*, and the *gradient norms* of both policy objectives. For planner prior trajectories with weights $w_n, n = 1, 2, \cdots, N$, the ESS quantifies the concentration of w_n and the *planner-action-weighted advantage* quantifies the additional return of the MPPI-weighted posterior trajectory distribution compared to the prior distribution:

$$\mathrm{ESS}^{\mathrm{planner}} = \frac{1}{\sum_{n=1}^{N}(w_n^{\mathrm{planner}})^2}, \quad A^{\mathrm{planner}} = \sum_{n=1}^{N}(w_n^{\mathrm{planner}} G(\tau_n)) - \frac{1}{N}\sum_{n=1}^{N} G(\tau_n). \tag{12}$$

This correlational analysis serves to formulate the following mechanistic hypotheses:

Degraded group: Runs in this group typically exhibit denser early reward signals together with rapid trajectory differentiation within the MPPI planner, manifested as an early, sharp drop in MPPI ESS. In addition, the norm of the background-planning policy gradient spikes in the early stages, which may indicate that an under-fitted world model and value function can lead to noisy planner actions that conflict with on-policy policy improvement.

Improved group: The corresponding tasks are often characterized by sparse early rewards and relatively flat early-stage value landscapes. The planner maintains a persistently high ESS from early to mid-training, indicating less concentrated importance weights over candidate trajectories. This broader set of comparably weighted trajectories may support higher exploration entropy. Only in later phases do imitation gradients steadily increase and guide policy improvement.

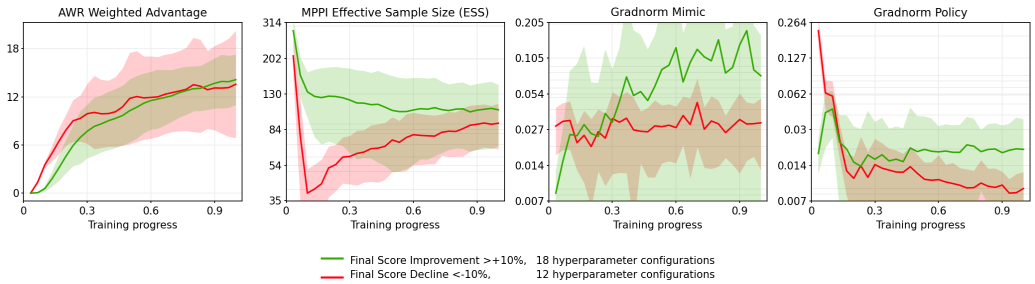

Figure 4: **Grouping analysis of AWR imitation.** We group the training metrics across 7 tasks $\times$9 AWR imitation hyperparameter $=$ 63 cells according to performance improvements $>$ **+10%** and degradations $<$ **-10%**, and average the metrics within each group.

The above performance-related factors suggest a consistent causal hypothesis. Tasks with abundant early rewards allow candidate trajectories to become distinguishable, triggering an early collapse in MPPI ESS. At this stage, the world model and value function may remain underfitted. Premature intervention by an unreliable planner may reduce action diversity and degrade replay-buffer quality. In sparse-reward tasks, the flatter early-stage value landscape is associated with less concentrated planner weights, while AWR imitation becomes stronger only later, when the model is more reliable. We leave rigorous validation of this causal mechanism to future work.

5 Related Work

Planning-Driven and Imagination-Driven MBRL. In MBRL, agents strive to acquire a *world model* capturing environment transition dynamics and reward mechanisms. This internal model enables the agent to synthesize experiences Hafner et al. (2025a;b) and perform counterfactual reasoning Hafner et al. (2019; 2022); Wang et al. (2024) beyond simple reflexive behavior. Popularized by AlphaZero Silver et al. (2017) and subsequently integrated into the MBRL ecosystem via MuZero Schrittwieser et al. (2020), Monte Carlo Tree Search (MCTS) remains a widely used planning approach in discrete domains. For continuous control, MPC Williams et al. (2017) is the predominant approach which optimizes state-action sequences over a finite horizon Hansen et al. (2022; 2024); Wang et al. (2025); Zhan et al. (2025). Compared to look-ahead search during inference, imagination-driven frameworks—epitomized by the Dreamer series Hafner et al. (2020; 2025a;b); Bredis et al. (2026)—shift the computational burden of planning entirely to *dreaming* during training. This paradigm offers significant advantages in terms of sample efficiency and deployment overhead. In this work, we identify and address bottlenecks in previous Dreamer variants, providing insights into the design of stochastic latent-state distributions for future imagination-driven policy improvement.

Curiosity-Driven Exploration. Although maximum entropy is widely regarded as a standard configuration in modern RL, its exploratory capabilities in complex tasks remain insufficient. Optimizing entropy independently across frames may simply cause the agent to dither within a local state space. To encourage sustained exploration, researchers have proposed curiosity-driven methods Pathak et al. (2017); Burda et al. (2018); Sukhija et al. (2024); Hou et al. (2025). The Intrinsic Curiosity Module Pathak et al. (2017) formulates curiosity rewards as the prediction error between a forward dynamics model's output and actual latent features. Similar to our approach of employing an independent network to predict dynamics loss, LPM Hou et al. (2025) introduces a related dual-network architecture. It uses an error model to predict the dynamics model's previous prediction error and derives intrinsic rewards from learning progress, thereby reducing sensitivity to aleatoric *noisy-TV* distractors Hou et al. (2025). We combine adaptive annealing and a dual-network architecture for curiosity-based intrinsic rewards within imagination-driven MBRL, phasing out exploration rewards during late-stage training to reduce interference with the task objective.

6 Conclusion

This work investigates whether test-time planning provides a consistent advantage over background planning for policy improvement in high-dimensional continuous control. Our results suggest three main conclusions. First, on-policy imagination combined with curiosity-driven exploration surpasses the previous SOTA under the matched 4M-step evaluation protocol in high-dimensional control. Second, the benefits of test-time planning are highly sensitive to both the task and its hyperparameters. Although planning can provide substantial gains in certain sparse-reward environments, it can also reduce sample efficiency under many configurations. Third, curiosity-driven exploration yields the strongest positive gains in our ablations, indicating that it provides a more targeted improvement for the combined difficulties of high-dimensional actions and sparse rewards.

Taken together, these findings suggest a promising direction for using computation scaling to mitigate the curse of dimensionality in high-dimensional policy improvement. Rather than treating test-time planning as a default component, we advocate using background planning together with intrinsic exploration rewards as the foundation, and further studying how to improve the signal-to-noise ratio of environment-interaction trajectories under imperfect world models and value functions. Such trajectories ultimately determine the quality of the data available for both world-model learning and subsequent policy improvement.

AI USE STATEMENT

In this work, we used generative AI tools to assist with the writing of proofs, method implementation, translation, and interpretation of results. We did not use generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, clean or reformat datasets, or support qualitative or thematic data analysis. Additionally, we used generative AI tools to create or edit software code, summarize or analyze existing literature, and improve the readability of the manuscript. We reviewed all AI-assisted work; in particular, AI-assisted code, proofs, and interpretations of results were independently verified and tested for correctness by the authors. We take responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide all implementation details and hyperparameter settings required to reproduce the results reported in this paper. Detailed hyperparameters are documented in Section C, and the complete implementation, including training and evaluation code as well as experiment configurations, is available in our anonymous repository at `https://anonymous.4open.science/r/online_imag-8C2E`. Together, these resources enable full reproduction of all experimental results presented in this work.

REFERENCES

George Bredis, Nikita Balagansky, Daniil Gavrilov, and Ruslan Rakhimov. Next embedding prediction makes world models stronger. *arXiv preprint arXiv:2603.02765*, 2026.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

Wei-Di Chang, Mikael Henaff, Brandon Amos, Gregory Dudek, and Scott Fujimoto. The surprising difficulty of search in model-based reinforcement learning. *arXiv preprint arXiv:2601.21306*, 2026.

Thomas Evers, Cristian Meo, Wendelin Bohmer, Justin Dauwels, and Yaniv Oren. Efficienttdmpc: Improved mpc objectives for sample-efficient continuous control. *arXiv preprint arXiv:2605.16692*, 2026.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/hafner19a.html`.

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

Danijar Hafner, Kuang-Huei Lee, Ian Fischer, and Pieter Abbeel. Deep hierarchical planning from pixels. *Advances in Neural Information Processing Systems*, 35:26091–26104, 2022.

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

Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training agents inside of scalable world models. *arXiv preprint arXiv:2509.24527*, 2025b.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In *ICML*, 2022.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations (ICLR)*, 2024.

Zhibo Hou, Zhiyu An, and Wan Du. Beyond noisy-tvs: Noise-robust exploration via learning progress monitoring. *arXiv preprint arXiv:2509.25438*, 2025.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787. PMLR, 2017.

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.

Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8583–8592. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/sekar20a.html.

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.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

Bhavya Sukhija, Stelian Coros, Andreas Krause, Pieter Abbeel, and Carmelo Sferrazza. Maxinforl: Boosting exploration in reinforcement learning through information gain maximization. *arXiv preprint arXiv:2412.12098*, 2024.

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.

Shengjie Wang, Shaohuai Liu, Weirui Ye, Jiacheng You, and Yang Gao. Efficientzero v2: Mastering discrete and continuous control with limited data. *arXiv preprint arXiv:2403.00564*, 2024.

Yuhang Wang, Hanwei Guo, Sizhe Wang, Long Qian, and Xuguang Lan. Bootstrapped model predictive control. In *The Thirteenth International Conference on Learning Representations*, 2025.

Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M Rehg, Byron Boots, and Evangelos A Theodorou. Information theoretic mpc for model-based reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 1714–1721. IEEE, 2017.

Guojian Zhan, Likun Wang, Xiangteng Zhang, Jiaxin Gao, Masayoshi Tomizuka, and Shengbo Eben Li. Bootstrap off-policy with world model. *arXiv preprint arXiv:2511.00423*, 2025.

A Supplemental Results

A.1 Training curves on main benchmarks.

We provide the training curves of our method and baselines in Figure 5.

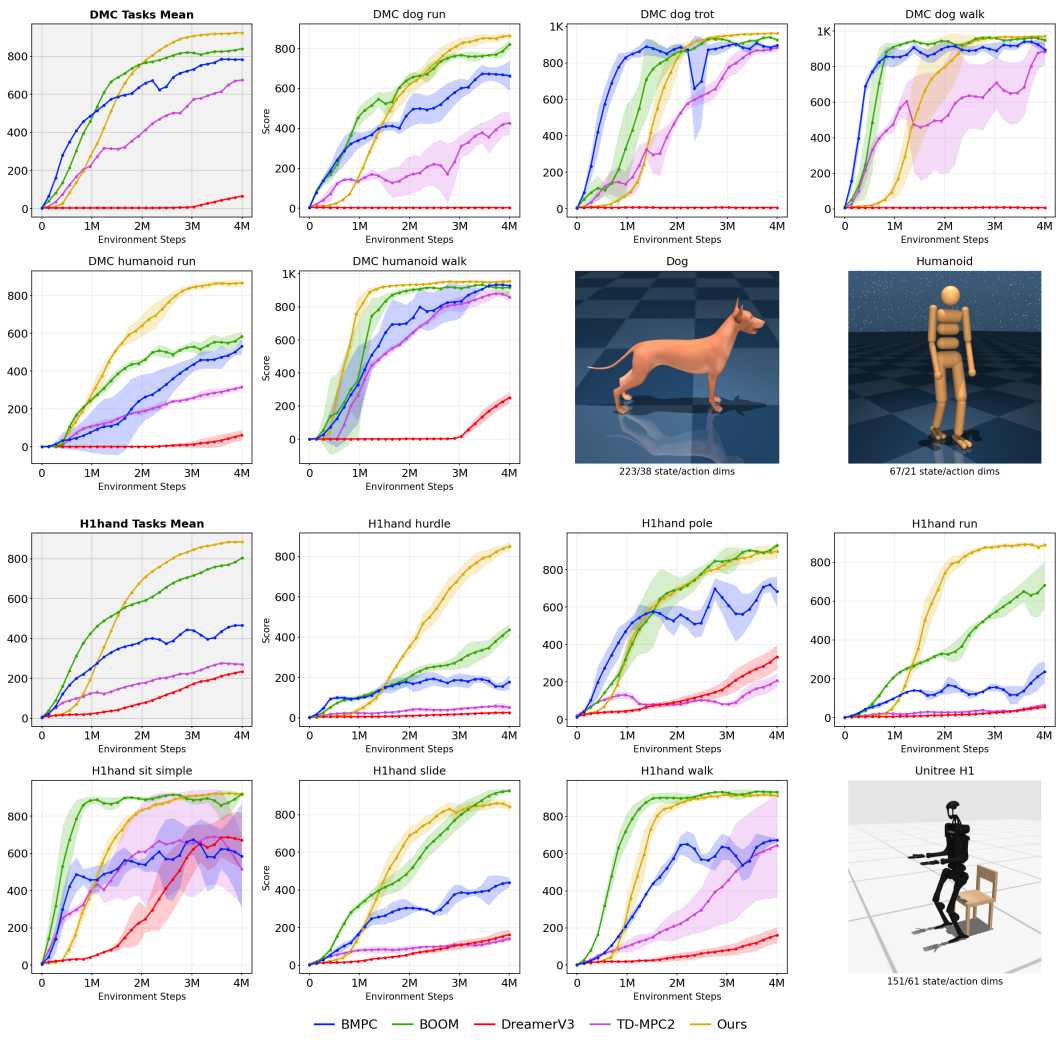

Figure 5: **Training curves on main benchmarks.** The shaded regions indicate the standard deviation over 5 runs. Additionally, the mean task-performance curves and benchmark visualizations are provided.

It is worth noting that our method exhibits strong training stability across all tasks, although it falls behind during the early stage of training compared with the TD-MPC family of methods. We hypothesize that this slow cold start primarily arises because the RSSM uses a more expressive but harder-to-optimize stochastic latent-variable model, while imagination-driven policy improvement depends strongly on the quality of synthetic trajectories, which can be inaccurate early in training. This hypothesis can be investigated in future work.

A.2 TRAINING CURVES ON PLANNER ANALYSIS.

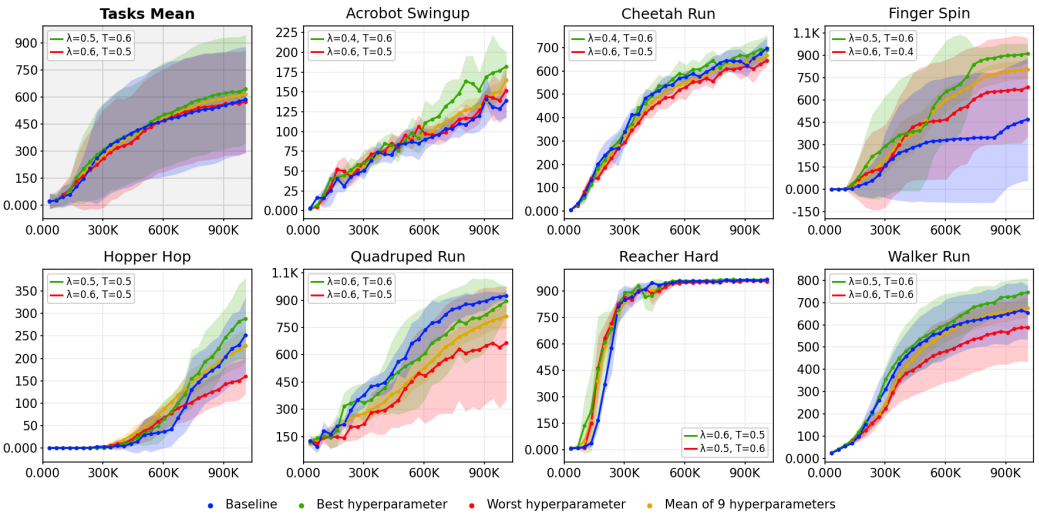

Figure 6: **Training curves on planner analysis.** The shaded regions indicate the standard deviation over 5 runs. The best and worst hyperparameter combinations for each task using the test-time planner and AWR imitation are annotated in the subfigure legend. In this figure, $\lambda = \xi^{\text{planner}}$ represents the MPPI planner temperature, and $T = \xi^{\text{mimic}}$ represents the AWR imitation temperature.

A.3 ADDITIONAL HYPERPARAMETER ANALYSIS

We posit that the model's performance exhibits low sensitivity to the curiosity annealing threshold, provided that the intrinsic reward is entirely eliminated during the final convergence phase to avoid interfering with the optimality of the task objective. To verify this, we tested an aggressive annealing threshold set to 0.8 times the return upper bound. As shown in Figure 7, the results indicate that such a substantial change in the annealing threshold only marginally affects performance.

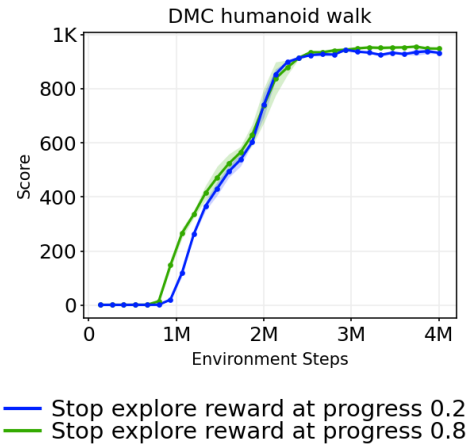

Figure 7: **Additional Hyperparameter Analysis.** We include a configuration that sets the progress threshold for annealing the curiosity exploration reward to zero at 0.8 times the return upper bound. The results indicate that the model's performance exhibits low sensitivity to substantial adjustments of this hyperparameter.

A.4 PLANNER HYPERPARAMETERS HEATMAP WITH THE LARGEST AGGREGATE IMPROVEMENT AND DECLINE REMOVED.

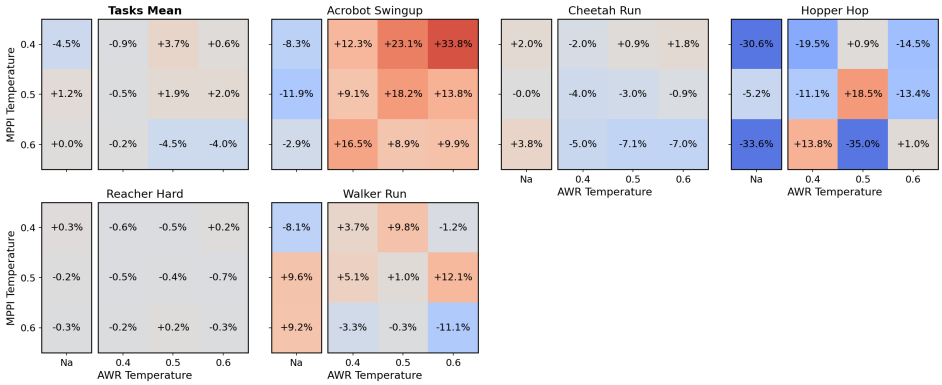

Figure 8: **Planner hyperparameters heatmap.** The heatmap shows the relative performance changes for MPPI and AWR temperature settings across 5 tasks $\times 12$ hyperparameter configurations (*Finger Spin* and *Quadruped Run* with the largest aggregate improvement and decline removed). The *Tasks Mean* panel directly aggregates performance percentages across tasks.

B THEORETICAL ANALYSIS

Consider a batch of prior trajectories $\tau \sim p(\tau)$ generated by parallel model-predictive rollouts from s_0:

$$p(\tau) = \prod_{t=0}^{H-1} \pi_\theta(a_t \mid s_t)\, p_\phi(s_{t+1} \mid s_t, a_t). \tag{13}$$

After one MPPI iteration, the trajectory distribution becomes:

$$\begin{aligned} p^*(\tau) &= \frac{1}{Z} p(\tau) \exp\left(\frac{G(\tau)}{\xi}\right) \\ Z &= \mathbb{E}_{\tau\sim p}\left[\exp\left(\frac{G(\tau)}{\xi}\right)\right] = \int_\tau p(\tau) \exp\left(\frac{G(\tau)}{\xi}\right) d\tau \end{aligned} \tag{14}$$

where $G(\tau)$ is the trajectory return.

We aim to prove that the theoretical target distribution $p^*(\tau)$ of the one-iteration MPPI planner corresponds to the exact analytical solution of the following KL-regularized objective function

$$p^* = \arg\max_q J(q) = \arg\max_q \left[\mathbb{E}_{\tau\sim q}[G(\tau)] - \xi\, \mathrm{KL}\left(q \parallel p\right)\right] \tag{15}$$

where $\xi > 0$ is the trust region regularization scale.

Proof:

First, we expand the Kullback-Leibler (KL) divergence term in the objective functional. By definition, $\mathrm{KL}\left(q \parallel p\right) = \mathbb{E}_{\tau\sim q}\left[\log \frac{q(\tau)}{p(\tau)}\right]$. Substituting this into the objective functional $J(q)$:

$$\begin{aligned} J(q) &= \mathbb{E}_{\tau\sim q}[G(\tau)] - \xi \mathbb{E}_{\tau\sim q}\left[\log \frac{q(\tau)}{p(\tau)}\right] \\ &= -\xi \mathbb{E}_{\tau\sim q}\left[\log q(\tau) - \log p(\tau) - \frac{G(\tau)}{\xi}\right] \\ &= -\xi \mathbb{E}_{\tau\sim q}\left[\log \frac{q(\tau)}{p(\tau)\exp\left(\frac{G(\tau)}{\xi}\right)}\right] \end{aligned}$$

To simplify this expression, we construct a target optimal distribution. We introduce the partition function Z to normalize the denominator so that it integrates (or sums) to 1, forming a valid probability distribution:

$$Z = \int p(\tau) \exp\left(\frac{G(\tau)}{\xi}\right) d\tau$$

Let this normalized target distribution be defined as:

$$\tilde{p}(\tau) = \frac{1}{Z} p(\tau) \exp\left(\frac{G(\tau)}{\xi}\right)$$

Rearranging this definition gives us $p(\tau) \exp\left(\frac{G(\tau)}{\xi}\right) = Z \cdot \tilde{p}(\tau)$. Substituting this equivalence back into our objective function yields:

$$J(q) = -\xi \mathbb{E}_{\tau \sim q}\left[\log \frac{q(\tau)}{Z \cdot \tilde{p}(\tau)}\right]$$

We can separate the partition constant Z from the logarithm:

$$\begin{aligned} J(q) &= -\xi \mathbb{E}_{\tau \sim q}\left[\log \frac{q(\tau)}{\tilde{p}(\tau)} - \log Z\right] \\ &= -\xi \mathbb{E}_{\tau \sim q}\left[\log \frac{q(\tau)}{\tilde{p}(\tau)}\right] + \xi \log Z \end{aligned}$$

Notice that the first term in the brackets is exactly the KL divergence between our proposal distribution q and the constructed target distribution $\tilde{p}$:

$$J(q) = -\xi \,\mathrm{KL}\left(q \parallel \tilde{p}\right) + \xi \log Z$$

Our goal is to maximize $J(q)$. According to Gibbs' inequality, the KL divergence is non-negative and attains its minimum value of 0 if and only if the two distributions are identical almost everywhere. Therefore, the unique optimal distribution that maximizes the objective is:

$$q^*(\tau) = \tilde{p}(\tau) = \frac{1}{Z} p(\tau) \exp\left(\frac{G(\tau)}{\xi}\right)$$

This matches the exponential weighting formula used by the MPPI-style update, showing that it is the closed-form optimizer of the KL-regularized expected-return objective. ■

In conclusion, the exponentially weighted formula for one-iteration MPPI is precisely the analytical solution to the problem of maximizing the expected return with KL-divergence regularization.

It is worth clarifying that the preceding derivation strictly establishes the posterior distribution update without return standardization. For a fixed candidate set with $\sigma_N > 0$, subtracting μ_N cancels after normalization, while dividing by σ_N is equivalent to using an effective temperature $\xi_{\text{eff}} = \xi\sigma_N$. Therefore, return standardization preserves the exponential-weighting form, but the standardized update is the exact optimizer of the KL-regularized objective with normalized returns (equivalently, with a state-dependent effective temperature), rather than of the original unstandardized objective with the same fixed ξ.

C Detailed Hyperparameters

Our codebase is accessible at `https://anonymous.4open.science/r/online_imag-8C2E`. We base our implementation on the official DreamerV3 codebase `https://github.com/danijar/dreamerv3`. The hyperparameter details are listed in Table 3.

The CPU used in this paper is an Intel Core i9-14900K, and the GPU is an NVIDIA GeForce RTX 5090. GPU VRAM consumption remains under 20GB across all runs. Training for 4M environment steps in the main comparison takes approximately 1.1 days.

D Visualizations

As shown in Figure 9, 10, and 11, we visualize the behaviors of our agent on the three most challenging locomotion tasks in HumanoidBench. Our agent learns to coordinate its high-degree-of-freedom body in contact-rich environments to achieve robust behaviors.

Table 3: Hyperparameter settings.

Hyperparameter	Value	Hyperparameter	Value
Training			
Learning rate	8×10^{-5}	Slow value network update rate	0.02
Environment samples	8	Value horizon	333
Replay batch size	32	Discount factor (γ)	1-1/333
Replay batch length	64	Optimizer	Adam
Buffer size	2×10^6	Optimizer steps	$256 \times 4 \times 10^6$
World Model			
Imagination length	15	Return normalization	Moving (5%, 95%)
Latent standard deviation range	$(0.001, 0.5)$	Advantage clip	Moving (5%, 95%)
Deterministic latent dimension	2048	Stochastic latent dimension	128
Reward loss scale	10	Dynamics loss scale	1
Value loss scale	1	Representation loss scale	0.1
Policy loss scale	0.5	Replay value loss scale	0.3
Reconstruction loss scale	1	Surprise loss scale	1
Planner (analysis study only; not used in the main results)			
MPPI Iterations	1	Horizon	8
MPPI samples (Population size)	512		
Actor-Critic			
Standard deviation range	(0.01, 0.33)	Entropy coefficient (α)	3×10^{-4}
bootstrap λ-return factor	0.95	Slow value regularization scale	1
Architecture (around 28M parameters in total)			
Encoder layers	3	Policy layers	3
Decoder layers	3	Value layers	3
Reward layers	1	RSSM implementation	BlockLinear
MLP Normalization	RMSNorm	RSSM imagination layers	2
MLP hidden layer dimension	512	RSSM dynamic layers	1
MLP activation	SiLU	RSSM observation layers	1

E Limitations and Future Work

While our approach provides an effective engineering framework that combines curiosity-driven intrinsic rewards and a scalable compute budget to tackle high-dimensional tasks and sparse rewards, several limitations remain. First, because imagination-driven policy improvement relies on an accurate dynamics model and training stochastic latent state variables is inherently difficult, our method generally suffers from a slow cold start during training. Second, recursively generating imagined trajectories demands a larger compute budget than the direct sampling methods used in model-free approaches. Therefore, when environment interaction is not the primary bottleneck, the trade-off between additional training compute and performance warrants further study.

F Positive and Negative Social Impact

Our method enhances the sample efficiency and final performance of MBRL in high-dimensional control tasks. This yields a positive impact on real-world applications, such as robotics, particularly in training scenarios where environment interactions are constrained. However, akin to many RL techniques, it carries the risk of misuse in sensitive domains, such as surveillance or autonomous weaponry. Furthermore, improved sample efficiency might encourage premature deployment in safety-critical environments, while the extremely high number of degrees of freedom in the agent's embodiment could introduce additional safety risks under out-of-distribution conditions. We strongly recommend comprehensive safety evaluations and responsible deployment practices to mitigate these risks.

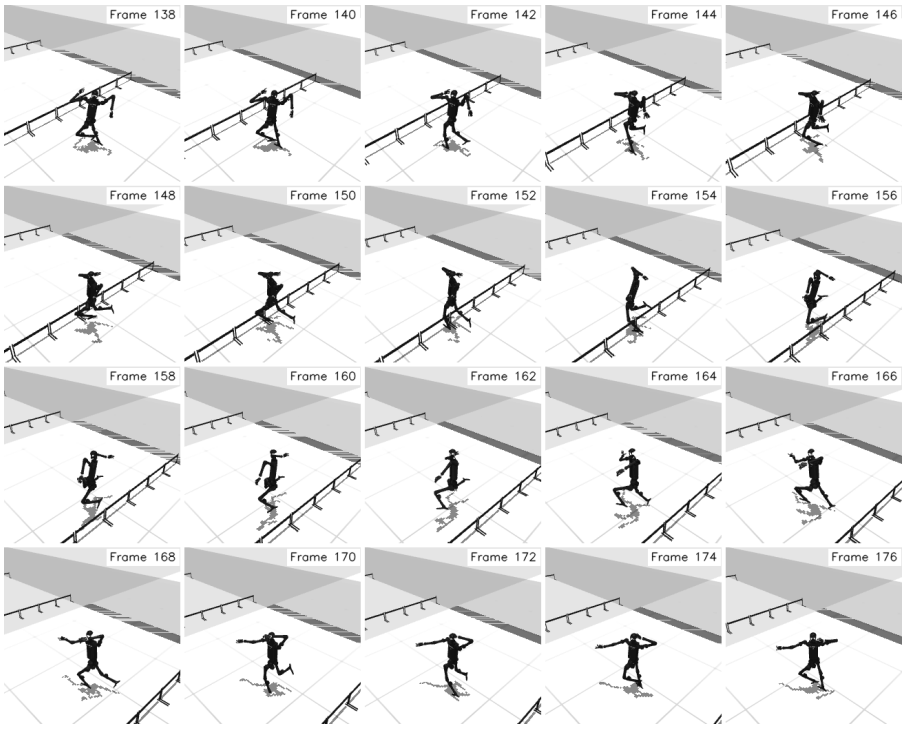

Figure 9: Visualizations of our agent solving HumanoidBench H1hand-hurdle. The agent uses only proprioceptive observations and maintains balance after clearing the hurdle while running.

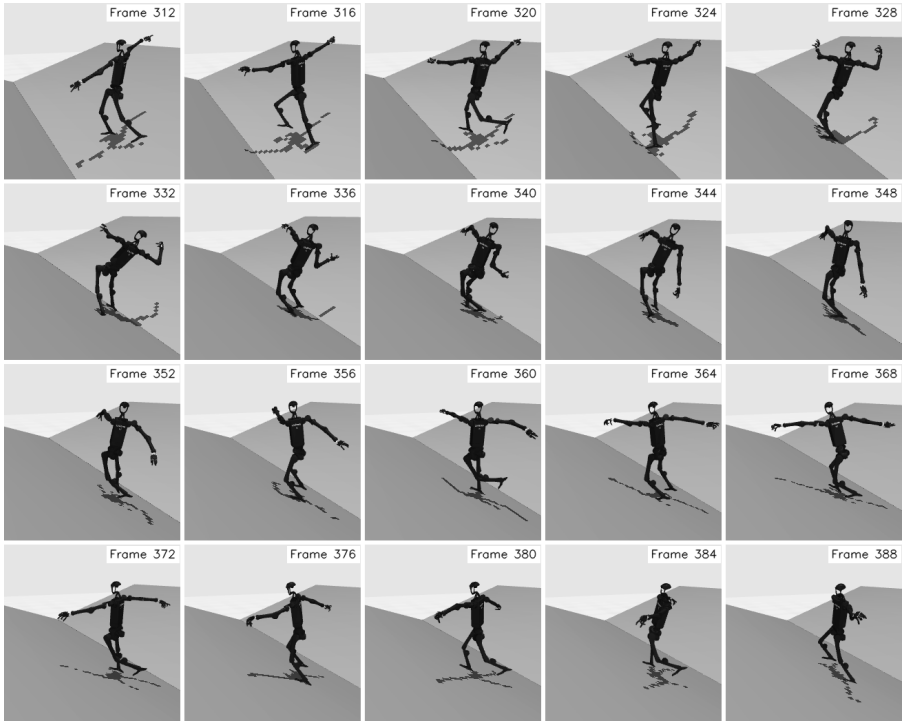

Figure 10: Visualizations of our agent solving HumanoidBench H1hand-slide. The agent appears to be balancing on a slope using outstretched arms.

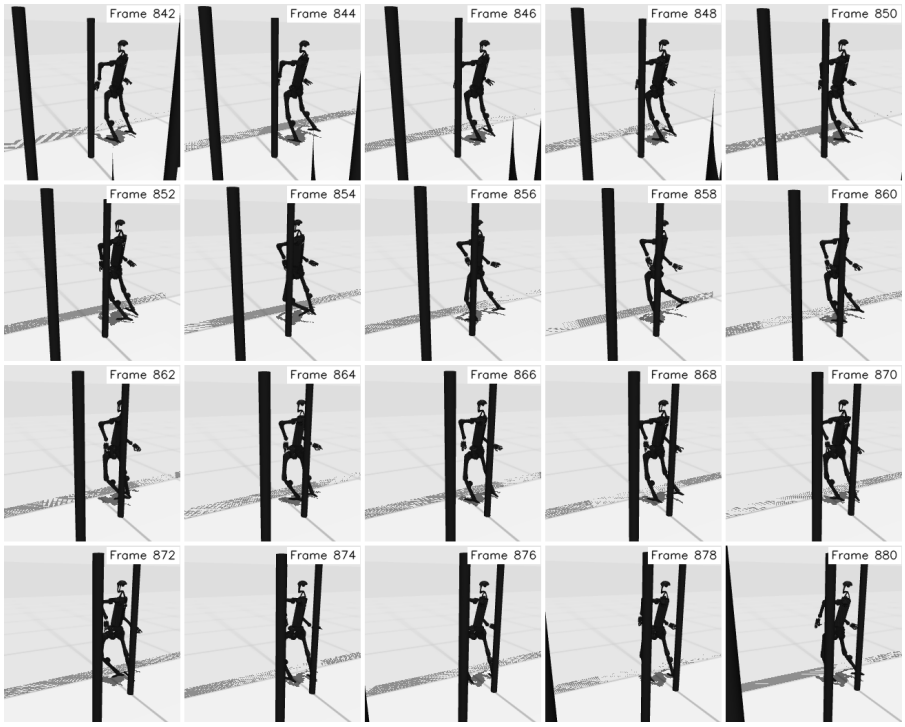

Figure 11: Visualizations of our agent solving HumanoidBench H1hand-pole. The learned policy adopts a sideways walking strategy along the wall to avoid the pole while continuing forward.