Perceiving the Knowledge Boundary: Uncertainty-Guided Exploration and Imagination for World Models

Zhenxian Liu[1], **Peixi Peng**[2,3*], **Yangru Huang**[1], **Yonghong Tian**[1,2,3*]

[1]National Engineering Research Center of Visual Technology, School of Computer Science, Peking University, China
[2]School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University, China
[3]Peng Cheng Laboratory, China
{zhenxianliu, yrhuang}@stu.pku.edu.cn, {pxpeng, yhtian}@pku.edu.cn

Abstract

World-model-based reinforcement learning achieves high sample efficiency by learning from imagined rollouts. However, its success critically depends on the accuracy of the learned world model, which is prone to producing unrealistic or hallucinated rollouts when queried beyond its domain of competence. These flawed predictions can trap the agent in a vicious cycle: by misleading exploration toward implausible or uninformative regions, they degrade the quality of collected data, which in turn corrupts policy learning with inaccurate rollouts. To break this cycle, we introduce the notion of a knowledge boundary—the region within which the world model provides reliable predictions—and propose a unified framework that both identifies and leverages this boundary. Concretely, we approximate the boundary using model uncertainty, quantified via disagreement across an ensemble of lightweight predictors, which serves as a practical proxy. This uncertainty signal is used in two complementary ways: as an intrinsic reward to guide exploration toward under-explored yet learnable regions, and as a dynamic filter to exclude unreliable imagined rollouts from policy optimization. Extensive experiments across diverse benchmarks—including CARLA, DeepMind Control Suite, Atari, and MemoryMaze—demonstrate that our approach consistently outperforms prior state-of-the-art methods.

1 Introduction

Model-Based Reinforcement Learning (MBRL) improves sample efficiency by learning a dynamics model of the environment and using it to generate synthetic rollouts for policy optimization (Sutton, Barto et al. 1998; Sutton 1991; Ha and Schmidhuber 2018; Hafner et al. 2019b,a, 2020, 2025; Moerland et al. 2023). This paradigm, often referred to as learning in imagination, enables agents to reduce reliance on costly real-world interactions by leveraging internally generated rollouts. It has led to strong performance across a range of challenging tasks with high-dimensional observations (Ma et al. 2023; Schrittwieser et al. 2020).

Despite the remarkable successes of MBRL, its effectiveness critically depends on the quality of the learned world model, which approximates the underlying environment and drives the agent's decision-making (Richens, Everitt, and Abel 2025). However, in practice, insufficient data coverage and model capacity may lead to approximation errors, rendering the learned world model a flawed simulator of the real environment (Talvitie 2017; Janner et al. 2019). We conceptualize this issue by introducing the model's *knowledge boundary*, within which its predictions remain reliable. Beyond this boundary, the model tends to produce inaccurate or even implausible outcomes—referred to as hallucinated rollouts—that deviate from real-world dynamics. In Figure 1, this hallucination is illustrated both conceptually, as rollouts stray far beyond the knowledge boundary, and empirically, as the model's predictions of rewards and future states severely diverge from the ground truth. This boundary thus serves as an effective representation of the model's capability, marking where its predictions can be trusted or not.

Neglecting the knowledge boundary poses a fundamental challenge to the learning process in MBRL, as it risks triggering a self-reinforcing cycle of degradation between exploration and policy learning. When the agent is unaware of the capability of its model, it lacks a principled mechanism to identify reliable and informative regions of the state space. As a result, exploration is misdirected either toward uninformative regions, where states reside completely inside the boundary and contribute little to model improvement (Rollout A in Figure 1), or toward implausible trajectories that lie far outside the boundary (Rollout C in Figure 1). In either case, the agent fails to access rollouts necessary to refine the world model and expand its reliable coverage (e.g., Rollout B in Figure 1). This flawed exploration contaminates policy optimization, as accurate and erroneous predictions become entangled. The agent may then overfit to model artifacts, resulting in degraded policies (Talvitie 2017). Conversely, the degraded policy fails to drive effective exploration, further limiting the diversity and usefulness of collected data. This vicious cycle highlights the need for agents to be aware of their world model's boundary and leverage this awareness in both exploration and policy optimization.

A principled and well-established approach in machine learning for identifying a model's knowledge boundary is to quantify its predictive uncertainty (Gal and Ghahramani 2016; Kendall and Gal 2017). As inputs move away from the training distribution, model uncertainty increases (Hendrycks and Gimpel 2016), and this increase can be quantified using proxies like predictive variance (Ovadia

*Corresponding author.
Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

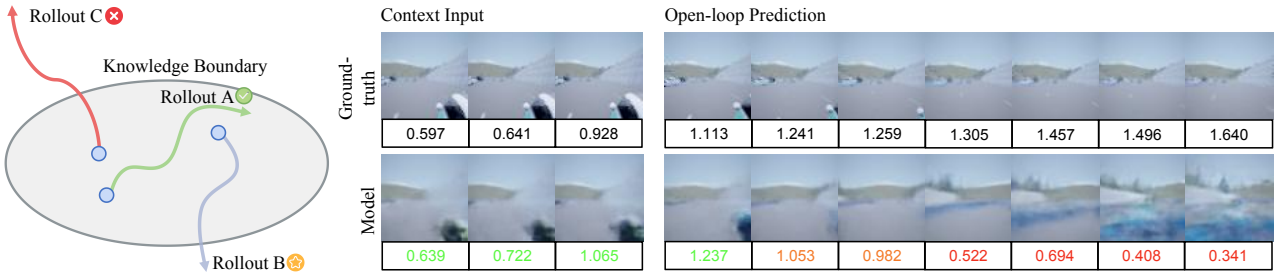

Figure 1: *Left:* A conceptual diagram of the knowledge boundary. Imagined rollouts can be reliable (A), exploratory (B), or unreliable hallucinations (C). Our framework aims to encourage rollouts like (B) to efficiently expand the knowledge boundary, while filtering out rollouts like (C) to prevent the contamination of policy learning. *Right:* A concrete example of an open-loop prediction, with the model's predicted rewards labeled below each frame. Inside the knowledge boundary, the world model's predictions are accurate (indicated by green). At the edge of the boundary, its predictions begin to slightly lose fidelity (indicated by orange), and as the rollout moves far beyond it, the model's predictions diverge into hallucinations (indicated by red).

et al. 2019). Ensemble disagreement—the variation in outputs from independently trained models—has proven to be a robust and scalable measure of epistemic uncertainty, particularly effective in MBRL settings (Lakshminarayanan, Pritzel, and Blundell 2017; Sekar et al. 2020). Therefore, we leverage ensemble disagreement as the core mechanism for identifying the knowledge boundary of the world model. Specifically, we train a lightweight ensemble of one-step predictors alongside the world model, and given the same state-action pair, the variance in their predictions serves as a reliable measure of epistemic uncertainty.

Building on the uncertainty signal provided by ensemble disagreement, we integrate it into the agent's learning process through a dual-objective optimization framework. First, the uncertainty signal serves as an intrinsic reward, guiding the agent to perform valuable exploratory rollouts that push the world model's knowledge boundary outward. Second, it functions as a dynamic filter during policy optimization, allowing the agent to discard unreliable, hallucinated rollouts. By combining intrinsic reward-based exploration with selective rollout filtering, the agent learns from both the information gained through exploration and the purified rollouts retained after filtering. This synergy leads to reliable model learning and ultimately boosts decision robustness.

To summarize, this study makes the following three key contributions:

- We formalize the concept of a *knowledge boundary* for world-model-based agents. We also introduce a unified framework that leverages this awareness to address both inefficient exploration and unreliable imagination.
- We propose a novel framework that uses ensemble disagreement as a unified signal. It is employed both as intrinsic rewards to guide exploration towards the knowledge boundary, and as a dynamic filter to remove hallucinated rollouts during policy optimization.
- We conduct extensive experiments on various challenging benchmarks, including CARLA, DeepMind Control Suite, Atari, and MemoryMaze. The results demonstrate that our approach significantly outperforms previous leading model-based and model-free methods.

2 Related Work

2.1 World-Model-Based Reinforcement Learning

Model-Based Reinforcement Learning (MBRL) improves sample efficiency by learning a dynamic model of the environment within a compact latent space (Ha and Schmidhuber 2018; Hafner et al. 2019b; Moerland et al. 2023). One prominent line of work trains a policy entirely on imagined rollouts generated by this model (Hafner et al. 2019a, 2020, 2025). Another major approach uses the latent model for decision-time planning, often integrating it with powerful search algorithms such as Monte Carlo Tree Search (Schrittwieser et al. 2020; Ye et al. 2021). This latent-space paradigm has driven state-of-the-art results across numerous challenging domains, from robotic control to complex games. However, a typical drawback of these methods is their implicit trust in the learned model (Janner et al. 2019; Talvitie 2017). This risks the agent learning from a flawed model, leading to suboptimal performance.

2.2 Exploration in Reinforcement Learning

Effective exploration remains a central challenge in reinforcement learning. A prominent line of work tackles this via *intrinsic motivation* (Chentanez, Barto, and Singh 2004), where agents generate their own rewards to encourage information-seeking behavior (Houthooft et al. 2016). Several methods design rewards based on prediction-based curiosity, which rewards surprising state transitions (Pathak et al. 2017; Burda et al. 2018), or state novelty, which incentivizes visiting unfamiliar states (Bellemare et al. 2016; Tang et al. 2017). A closely related paradigm uses epistemic uncertainty as an exploration bonus, typically estimated from deep-ensemble disagreement, to drive the agent toward data that will most improve the model (Pathak, Gandhi, and Gupta 2019; Sekar et al. 2020). Our work builds on this uncertainty-driven principle, but innovates by integrating it into a unified framework where the same signal is also used to ensure the reliability of the agent's imagination.

2.3 Uncertainty Quantification for Robust Model-Based Control

Quantifying model uncertainty is crucial for building robust and reliable agents. Common approaches in deep learning include Bayesian Neural Networks (BNNs) and Deep Ensembles, which estimate uncertainty through weight distributions (Blundell et al. 2015; Gal and Ghahramani 2016) or predictive variance (Lakshminarayanan, Pritzel, and Blundell 2017). In model-based RL, uncertainty has been widely used for robust control, for example, to enable risk-averse planning over a distribution of possible futures (Chua et al. 2018; Kurutach et al. 2018; Yu et al. 2020, 2021) or to learn safe policies that avoid high-uncertainty regions of the state space (Lütjens, Everett, and How 2019; Kidambi et al. 2020; Seo, Nakamura, and Bajcsy 2025). Differently, our approach leaves the generative imagination process untouched and uses uncertainty in a post-hoc manner.

3 Preliminary

Our work is centered on vision–based RL tasks, modelled as a *partially observable Markov decision process* (POMDP) (Kaelbling, Littman, and Cassandra 1998). A POMDP is denoted by the tuple $(\mathcal{O}, \mathcal{A}, p, r, \gamma)$. At time step t, the agent selects an action $a_t \sim \pi(a_t \mid o_{\leq t},\, a_{<t})$ and then receives an observation $o_t \sim p(o_t \mid o_{<t},\, a_{<t})$ together with a scalar reward $r_t = r(o_{\leq t},\, a_{<t})$, where both the transition kernel p and the reward function r are unknown. The objective of model-based RL is to learn a policy that maximizes the γ-discounted return $\mathbb{E}_{p,\pi}\left[\sum_{t=1}^{T} \gamma^{t-1} r_t\right]$ by exploiting a learned world model that approximates the true environment dynamics and reward functions (p, r).

World Model Learning. Our method and main analyses are built upon DreamerV3 (Hafner et al. 2025), a widely used baseline in MBRL. We instantiate the world model using a Recurrent State–Space Model (RSSM). First, an encoder transforms each sensory input x_t into a stochastic latent vector z_t. A recurrent sequence model then maintains a hidden state h_t that predicts the evolution of these latents, conditioned on the previous action a_{t-1}. Concatenating h_t and z_t yields the full model state, which is used to (i) predict the reward r_t, (ii) output an episode–continuation flag $c_t \in \{0, 1\}$, and (iii) reconstruct the original input, thereby encouraging informative representations:

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

Policy Learning. The world model trained above is used to generate massive imagined rollouts in the learned latent space. An actor-critic agent, operating on the latent state $s_t = (h_t, z_t)$, then learns a policy entirely from this imagined data. With a discount factor γ, the actor aims to maximize the expected return, defined as $R_t \doteq \sum_{\tau=0}^{\infty} \gamma^\tau r_{t+\tau}$. The critic's objective is to approximate the value of each state under the policy defined by the actor. The actor π_θ and critic V_ψ are formulated as follows:

$$\begin{aligned} &\text{Actor:} \quad a_t \sim \pi_\theta(a_t \mid s_t), \\ &\text{Critic:} \quad V_\psi(s_t) \approx \mathbb{E}_{\pi_\theta, p_\phi}[R_t] \end{aligned} \tag{2}$$

As analyzed in Section 1, a learned world model is only an approximation of the real environment and has its own knowledge boundary. Overlooking this boundary gives rise to a self-reinforcing cycle of degradation, where inefficient exploration fails to expand the model's knowledge boundary effectively, and policy learning is in turn corrupted by hallucinated rollouts generated from this flawed model.

4 Methodology

To break this vicious cycle, we equip the agent with a mechanism to perceive its world model's knowledge boundary. We realize this by quantifying the model's epistemic uncertainty via an ensemble of lightweight predictive models. This uncertainty signal serves a dual role within our framework: it is formulated as an intrinsic reward to guide boundary-aware exploration, and simultaneously functions as a reliability metric for a rollout filtering module that safeguards policy learning.

4.1 Perceiving the Knowledge Boundary via Ensemble Disagreement

Our method perceives the knowledge boundary by quantifying the world model's predictive uncertainty, a technique well-established in machine learning (Gal and Ghahramani 2016; Kendall and Gal 2017). This uncertainty is expected to increase as the agent encounters states further from its trusted regions (Ovadia et al. 2019). To this end, we implement a practical uncertainty estimator using an ensemble of K lightweight one-step models, realized as MLPs, trained concurrently with the world model.

In more detail, let e_t denote the latent embedding of an observation x_t from the world model's encoder. Each model in the ensemble, represented by a function f_{ω_k} parameterized by ω_k, is trained to predict the next embedding e_{t+1} given state-action pair (s_t, a_t). The ensemble is optimized by minimizing the mean squared error across the predictions, with the expectation taken over a minibatch $\mathcal{B}$ of samples:

$$\mathcal{L}_{\text{ensemble}}(\omega) = \mathbb{E}_{(s_t, a_t, e_{t+1}) \sim \mathcal{B}} \left[\sum_{k=1}^{K} \| f_{\omega_k}(s_t, a_t) - e_{t+1} \|^2 \right] \tag{3}$$

We also explored alternative targets, such as predicting the next stochastic or deterministic state. However, we found that predicting the next embedding yields the best performance. Once the ensemble has been trained, we use it to compute the uncertainty signal u_t for a given state-action pair (s_t, a_t). This uncertainty is defined as the variance across the predictions of the K models. Let $\bar{f}(s_t, a_t)$ denote the mean of these predictions, the uncertainty at time t is then calculated as follows:

$$u(s_t, a_t) = \frac{1}{K-1} \sum_{k=1}^{K} \left\| f_{\omega_k}(s_t, a_t) - \bar{f}(s_t, a_t) \right\|^2 \tag{4}$$

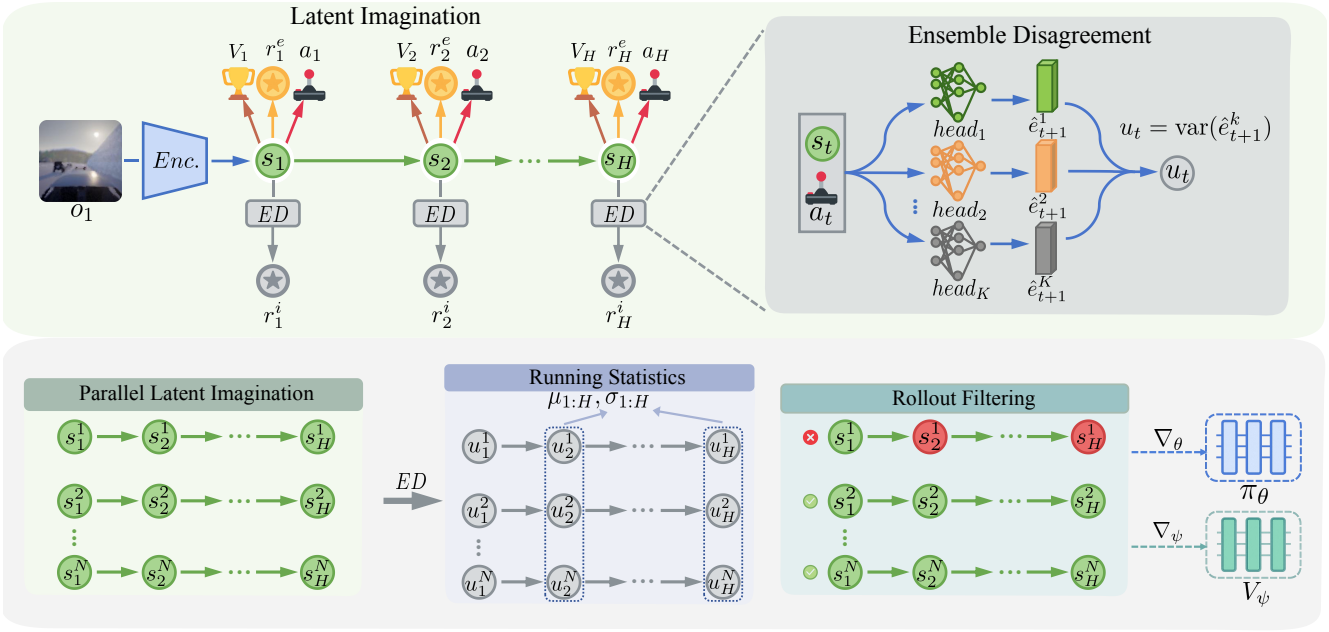

Figure 2: An overview of our proposed framework. Ensemble Disagreement serves as a unified uncertainty signal to perceive the world model's knowledge boundary. This signal serves a dual purpose: it is formulated as an intrinsic reward to drive efficient exploration, and it is used to identify and purge unreliable imagined rollouts from the policy's massive training data.

4.2 Uncertainty-Guided Exploration

Building on the notion of uncertainty quantified through ensemble disagreement, we now use this uncertainty to guide exploration. Specifically, we incentivize the agent to seek out states where its knowledge is limited, thereby effectively expanding its knowledge boundary. The epistemic uncertainty, as measured by ensemble disagreement, serves as an ideal proxy for the expected information gain in a given state (Houthooft et al. 2016), providing a principled signal for exploration.

As illustrated in Figure 2 (top), our framework imagines an H-step latent trajectory from a single starting state. In addition to the extrinsic reward r^e_t and value V_t predicted within the standard DreamerV3 paradigm, the uncertainty score derived from Ensemble Disagreement (ED) is used as an intrinsic reward r^i_t at each step. This intrinsic reward is then scaled by a hyperparameter w_i and added to the extrinsic reward to form the total reward used for policy learning. Specifically, the total reward is computed as:

$$r_t = r^e_t + w_i r^i_t \tag{5}$$

where w_i is a scalar hyperparameter that controls the balance between exploration and exploitation.

Using ensemble disagreement as an intrinsic reward offers several key advantages. First, the reward signal is naturally non-stationary. As the world model becomes more confident in a given region, the ensemble's disagreement decreases, and thus the exploration bonus gradually anneals toward zero once an area has been sufficiently explored. Second, this approach is robust to stochasticity in the environment. Unlike prediction error-based curiosity, which can be perpetually distracted by irreducible noise (the "noisy TV" problem) (Pathak et al. 2017; Burda et al. 2018), an ensemble of models learns to agree on the distribution of a stochastic outcome. As a result, its disagreement decreases even in noisy environments, allowing the agent to focus on exploring areas with reducible epistemic uncertainty (Pathak, Gandhi, and Gupta 2019).

4.3 Uncertainty-Guided Imagination Filtering

Given the uncertainty of each state in the imagined rollouts, the imagination filtering module aims to distinguish the hallucinated rollouts. We introduce a two-stage filtering mechanism that first identifies high-uncertainty states and then decides whether to discard the corresponding rollout, as depicted in Figure 2 (bottom).

State-Level Hallucination Detection. The primary goal of our filtering module is to differentiate between persistently hallucinatory rollouts, which should be discarded, and valuable exploratory rollouts that probe the knowledge boundary and should be retained. The key insight is that a valuable exploratory rollout will naturally exhibit a predictable increase in predictive uncertainty as it approaches the knowledge boundary. To account for this expected trend and avoid penalizing reasonable exploration, we maintain stepwise running statistics of uncertainty across the imagination horizon H. Specifically, we track the mean vector $\mu_{1:H}$ and variance vector $\sigma^2_{1:H}$, which are updated across training batches via an Exponential Moving Average (EMA):

$$\begin{aligned} \mu_t &\leftarrow \alpha\mu_t + (1-\alpha)\bar{u}_t, \\ \sigma^2_t &\leftarrow \alpha\sigma^2_t + (1-\alpha)\frac{1}{N}\sum_{n=1}^{N}(u^n_t - \bar{u}_t)^2 \end{aligned} \tag{6}$$

Method	Episode reward ↑	Distance (m) ↑	Crash intensity ↓	Average steer ↓	Average brake ↓
SAC (arXiv'18)	121 ± 26.1	74 ± 17.4	3930 ± 80.3	$17.52\% \pm 0.021\%$	$1.81\% \pm 0.013\%$
DrQ (ICLR'21)	154 ± 21.5	95 ± 27.2	$\mathbf{2419 \pm 72.3}$	$15.79\% \pm 0.018\%$	$1.70\% \pm 0.039\%$
TACO (NeurIPS'23)	208 ± 23.4	197 ± 17.6	2997 ± 104.8	$16.78\% \pm 0.022\%$	$1.58\% \pm 0.030\%$
MaDi (AAMAS'24)	177 ± 18.6	143 ± 28.9	2557 ± 86.3	$14.46\% \pm 0.035\%$	$2.47\% \pm 0.024\%$
DreamerV3 (Nature'25)	375 ± 24.3	391 ± 21.0	2449 ± 104.5	$11.39\% \pm 0.052\%$	$1.29\% \pm 0.011\%$
MuDreamer (arXiv'24)	382 ± 27.9	399 ± 24.6	2996 ± 114.3	$13.30\% \pm 0.041\%$	$1.07\% \pm 0.015\%$
HarmonyDream (ICML'24)	413 ± 38.5	426 ± 44.1	3128 ± 117.9	$12.23\% \pm 0.037\%$	$0.87\% \pm 0.016\%$
Ours	$\mathbf{476 \pm 33.8}$	$\mathbf{502 \pm 39.6}$	3064 ± 154.1	$\mathbf{10.70\% \pm 0.043\%}$	$\mathbf{0.42\% \pm 0.018\%}$

Table 1: Driving metrics at 100k training steps, the central line separates model-free (top) from model-based methods (bottom). Arrows denote whether larger or smaller values are preferred, and best results are highlighted in **bold**.

where $\bar{u}_t$ is the mean uncertainty of the current batch at imagination step t. A state s_t^n from the n-th rollout at step t is flagged as risky if its uncertainty exceeds this stepwise threshold. Here, β is a sensitivity hyperparameter, and the flagging function is defined as:

$$\text{flag}(s_t^n) = \mathbb{I}[u_t^n > \mu_t + \beta\sigma_t] \tag{7}$$

Rollout-Level Filtering. Once we identify risky states, we perform Rollout-Level Filtering to separate rollouts that devolve into persistent hallucinations from others. To do so, we calculate the proportion of states in each rollout that have been flagged as risky. A rollout is discarded if the proportion of flagged states meets a predefined threshold τ:

$$\text{discard}(\text{rollout}_n) = \mathbb{I}\left[\frac{1}{H}\sum_{t=1}^{H}\text{flag}(s_t^n) \geq \tau\right] \tag{8}$$

This proportion-based criterion is essential for the robustness of our method, as it effectively identifies rollouts that exhibit consistent patterns of hallucination, while still tolerating the temporary increases in uncertainty that result from legitimate exploration of uncertain regions.

Finally, the actor and critic are updated using only the purified batch of imagined rollouts, as depicted in Figure 2 (bottom). It is important to note that this entire filtering process is applied post-hoc, which is crucial for both the integrity of long-horizon return calculations and for maintaining computational efficiency.

5 Experiment

5.1 Environment Settings

Benchmark. We validate our method on an extensive collection of challenging visual RL benchmarks. We use the highway driving scenario in CARLA, a high-fidelity simulator that rigorously tests the agent's ability to deal with complex visual inputs (Dosovitskiy et al. 2017). To evaluate performance on robotic control, we use six challenging tasks from the popular DeepMind Control Suite (DMC) (Tassa et al. 2018), specifically choosing those where the baseline DreamerV3 agent (Hafner et al. 2025) shows a significant performance gap between learning from vision and proprioceptive states. We assess sample efficiency on the Atari100k benchmark (Kaiser et al. 2019), focusing on five games where the baseline DreamerV3 struggles. Finally, we use MemoryMaze, a 3D environment with procedurally generated mazes from a first-person perspective that assesses the agent's long-term memory and self-localization (Pasukonis, Lillicrap, and Hafner 2022). Collectively, these benchmarks form a comprehensive testbed for evaluating our method's capabilities in complex perception, sample efficiency, and long-term memorization.

Experiment Details. Our method is built upon DreamerV3, and its default network architectures and training hyperparameters are kept unless otherwise specified. Our ensemble disagreement module consists of $K = 5$ one-step predictive models, with each realized as a 2-hidden-layer MLP. For the imagination filtering module, we set the key hyperparameters as follows: the EMA decay rate for the running statistics is $\alpha = 0.995$, the sensitivity for the uncertainty threshold is $\beta = 2.0$, and the proportion threshold for filtering a rollout is $\tau = 0.25$. All experiments are conducted with 5 random seeds, with the mean performance reported.

5.2 Results

CARLA. We first evaluate our method on a challenging highway driving task in the high-fidelity CARLA simulator, which requires robust policy learning from complex visual inputs. For comparison, we evaluate against a suite of representative prior methods, including both model-free and model-based agents. The model-free baselines include SAC (Haarnoja et al. 2018), DrQ (Kostrikov, Yarats, and Fergus 2020), TACO (Zheng et al. 2024), and MaDi (Grooten et al. 2023). The model-based baselines include the original DreamerV3 (Hafner et al. 2025), MuDreamer (Burchi and Timofte 2024), and HarmonyDream (Ma et al. 2023).

Table 1 demonstrates that our method significantly outperforms all baselines on the primary metrics of episode reward and distance traveled. This superior performance is also reflected in the agent's behavior, achieving the lowest average steer and average brake suggests a more stable and efficient control policy. We hypothesize that this success is particularly pronounced in a visually complex, open-world environment like CARLA, where the world model is highly susceptible to producing hallucinated rollouts. Our imagination filtering module is designed to counteract this failure mode by identifying and purging these unreliable rollouts, leading to a more robust and effective driving policy. Furthermore, Table 1 reveals a substantial performance gap be-

Task	SAC	DrQ-v2	TACO	MaDi	DreamerV3	MuDreamer	HarmonyDream	Ours
Acrobot Swingup	5.1	128.4	242.2	147.2	231.4	191.8	315.4	**361.0**
Cartpole Swingup Sparse	154.6	706.9	684.9	514.3	756.2	554.8	739.4	**789.6**
Finger Spin	312.2	**846.7**	833.4	623.5	401.3	507.5	533.7	756.9
Hopper Hop	3.1	189.9	241.1	117.0	269.5	247.4	298.4	**314.6**
Quadruped Run	50.5	407.0	508.5	512.9	477.0	**723.4**	684.0	689.1
Walker Run	26.9	517.1	617.8	443.7	603.8	542.7	683.1	**736.1**
Average	92.1	466.0	521.3	393.1	456.5	461.3	542.3	**607.9**

Table 2: Comparison against state-of-the-art methods on the six DMC tasks at 1M environment steps. The best results are highlighted in **bold**.

tween model-based and model-free agents, indicating that in sample-constrained, visually complex tasks like CARLA, access to a learned model provides an advantage.

DMC. As summarized in Table 2, our approach achieves the highest performance on most of the six visually complex DMC tasks. An interesting phenomenon revealed in the results is that while model-based methods show a clear advantage on most tasks, they are outperformed by top model-free agents on Finger Spin. This may suggest that the rapid, high-frequency, and precise dynamics of this particular task are especially challenging for world models to capture perfectly. In such specific cases, the direct, reactive policies learned by model-free methods can be more effective than policies trained on an imperfect imagined environment.

Method	Amidar	Crazy	Gopher	Road	Up
EfficientZero	102	80125	**3518**	18512	16096
DreamerV3	90	76093	2977	9972	11894
MuDreamer	114	80017	1347	9003	4766
HarmonyDream	107	83560	4479	15502	13361
Ours	**124**	**89313**	3122	**19442**	**20197**

Table 3: Performance comparison on five selected Atari100k tasks. Crazy, Road, and Up stand for Crazy Climber, Road Runner, and Up N Down, respectively.

Atari. To evaluate sample efficiency, we test our method on five challenging games from the Atari100k. We additionally compare against EfficientZero, a highly competitive method in this low-data regime. As shown in Table 3, our method outperforms others on four of the five selected games. We attribute this high sample efficiency to the synergy between our two core contributions. The uncertainty-guided exploration ensures the limited interaction budget is spent gathering the most informative data, while the imagination filtering provides a clean and stable signal for the policy update, making each learning step more effective.

MemoryMaze. Finally, we evaluate our agent's long-term memory and self-localization capabilities in MemoryMaze, specifically on the 9×9 and 11×11 procedurally generated mazes. As shown in Figure 3, our method consistently outperforms its opponents in both tasks. By encouraging systematic exploration and ensuring the agent learns from a reliable stream of imagined data, our framework enables the construction of more coherent mental maps, leading to more effective long-horizon navigation.

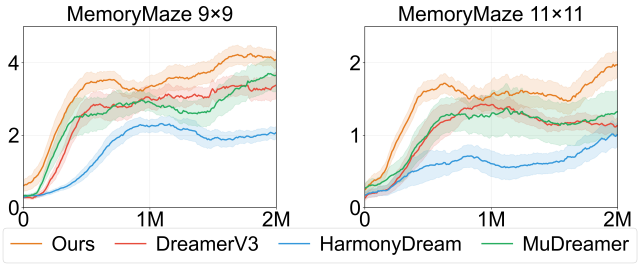

Figure 3: Performance comparison on MemoryMaze.

5.3 Ablation Study

We present several ablation studies on the challenging CARLA highway task to understand the contribution of each component and validate our design choices.

Effectiveness of Each Component. First, we analyze the contribution of our two main components: uncertainty-guided exploration and imagination filtering. As shown in Figure 4(a), both the exploration-only and filtering-only variants individually outperform the DreamerV3 baseline. Our full method achieves the highest performance, indicating a synergistic relationship between actively seeking the knowledge boundary and robustly learning from within it.

Bootstrap Sampling Ablation. We investigate the necessity of bootstrap sampling (Osband et al. 2016) for training the ensemble disagreement module. As shown in Figure 4(b), we find that both approaches yield comparable performance. This aligns with the observation of Lakshminarayanan, Pritzel, and Blundell (2017) that the inherent stochasticity from random initializations and optimizer dynamics is sufficient to produce a diverse ensemble. For algorithmic simplicity, we therefore opt not to use bootstrap sampling in our final setting.

Ensemble Size Ablation. We study the effect of the ensemble size K on performance. As shown in Figure 4(c), performance increases from $K = 3$ to $K = 5$, but we observe diminishing marginal returns for $K > 5$, with the performance of $K = 7$ and $K = 10$ being similar to $K = 5$. Given the trade-off between performance and computational cost, we select $K = 5$ as the default for all our experiments.

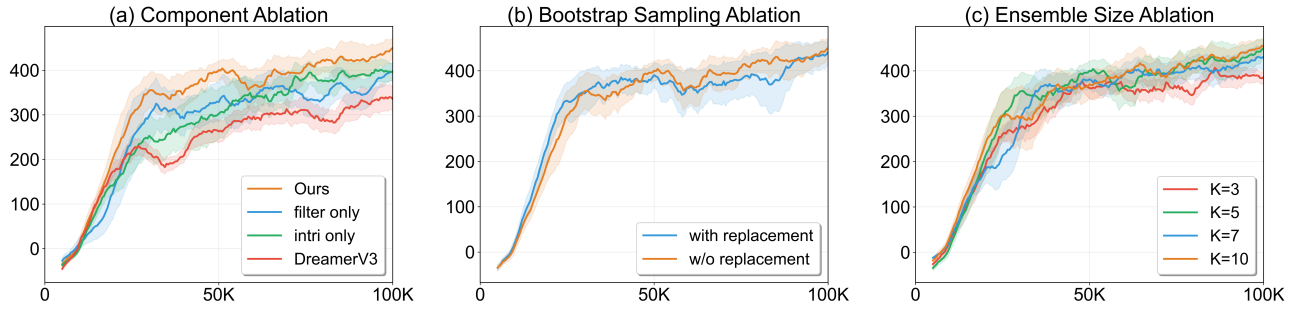

Figure 4: Ablation studies on CARLA. **(a)** Ablation of the effectiveness of each component. **(b)** Ablation of the bootstrap sampling used for training the ensemble. **(c)** Ablation of the ensemble size used for uncertainty estimation. All curves show the average performance over 5 random seeds, and the shaded regions indicate one standard deviation.

5.4 Discussion and Analysis

Stepwise versus Aggregated Statistics. A successful filtering mechanism must be able to distinguish between valuable exploratory behavior and persistent hallucination. Effective exploration requires the agent to probe its knowledge boundary, transitioning from known to unknown regions of the state space. This process is expected to cause a natural increase in predictive uncertainty as an imagined rollout extends further from its reliable starting point. Figure 5 (left) provides strong empirical evidence for this phenomenon, showing a consistent upward trend in uncertainty across all four benchmarks. This demonstrates that a robust filtering mechanism must account for this temporal trend, rather than using a single threshold for all timesteps.

To validate our hypothesis that maintaining stepwise statistics is crucial, we compare our method against a variant using aggregated statistics. This variant pools the uncertainty scores from all timesteps within each rollout to compute a global mean and variance. Figure 5 (right) shows that our stepwise method outperforms the aggregated variant. It confirms that assessing uncertainty relative to each imagination depth is crucial, as it allows us to identify anomalous hallucinations without unfairly penalizing the natural increase in uncertainty inherent to long-horizon exploration.

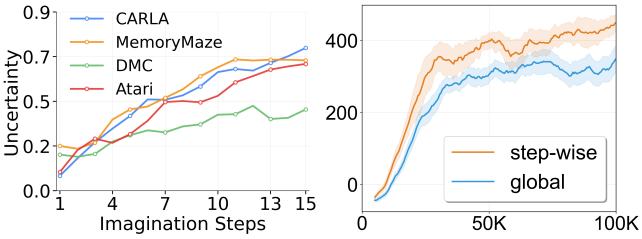

Figure 5: Analysis of uncertainty trends and the benefit of using stepwise statistics.

Exploration Behavior. To evaluate our uncertainty-guided exploration module, we visualize the latent state visitation of our agent against the baseline. We collected latent states from both agents at the same training stage and projected them onto a two-dimensional plane using Principal Component Analysis (PCA) (Abdi and Williams 2010).

The resulting state space coverage is shown in Figure 6. The baseline agent's exploration is confined to a smaller region of the state space and converges to fewer distinct modes. In contrast, our agent covers a substantially larger area and discovers a more diverse set of high-density regions. This comparison demonstrates that by incentivizing the agent to probe its knowledge boundary, our method effectively improves the breadth and efficiency of exploration.

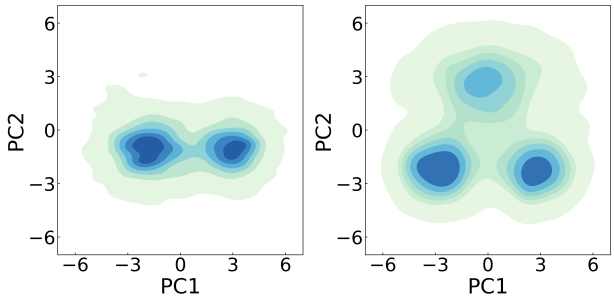

Figure 6: We visualize the state space coverage of our method against baseline in CARLA. Latent states from 500 environment steps are sampled from each agent at the same training stage and visualized using Principal Component Analysis (PCA). To ensure a fair comparison, the axes for both plots are set to identical ranges. Thus, a larger state visitation area directly indicates more effective exploration.

6 Conclusion

We proposed a novel framework that perceives the world model's knowledge boundary to break the vicious cycle between inefficient exploration and policy corruption. Our method uses a single ensemble disagreement signal for a dual purpose: as an intrinsic reward to guide boundary-aware exploration, and as a dynamic filter to purge hallucinated rollouts from the policy learning process. Extensive experiments across a diverse suite of challenging benchmarks demonstrated that our approach significantly outperforms prior state-of-the-art methods, highlighting the importance of equipping agents with an awareness of their own model's capability.

Acknowledgments

The study was funded by the Shenzhen Science and Technology Program (KQTD20240729102051063), the National Natural Science Foundation of China under contracts No. 62425101, No. 62332002, No. 62027804, No. 62088102, No. 62422602, No. 62372010, Key Laboratory Grants 241-HF-D05-01, and the major key project of the Peng Cheng Laboratory (PCL2024AS204). Computing support was provided by Pengcheng Cloudbrain.

References

Abdi, H.; and Williams, L. J. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4): 433–459.

Bellemare, M.; Srinivasan, S.; Ostrovski, G.; Schaul, T.; Saxton, D.; and Munos, R. 2016. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29.

Blundell, C.; Cornebise, J.; Kavukcuoglu, K.; and Wierstra, D. 2015. Weight uncertainty in neural network. In *International conference on machine learning*, 1613–1622. PMLR.

Burchi, M.; and Timofte, R. 2024. Mudreamer: Learning predictive world models without reconstruction. *arXiv preprint arXiv:2405.15083*.

Burda, Y.; Edwards, H.; Pathak, D.; Storkey, A.; Darrell, T.; and Efros, A. A. 2018. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*.

Chentanez, N.; Barto, A.; and Singh, S. 2004. Intrinsically motivated reinforcement learning. *Advances in neural information processing systems*, 17.

Chua, K.; Calandra, R.; McAllister, R.; and Levine, S. 2018. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31.

Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; and Koltun, V. 2017. CARLA: An open urban driving simulator. In *Conference on robot learning*, 1–16. PMLR.

Gal, Y.; and Ghahramani, Z. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, 1050–1059. PMLR.

Grooten, B.; Tomilin, T.; Vasan, G.; Taylor, M. E.; Mahmood, A. R.; Fang, M.; Pechenizkiy, M.; and Mocanu, D. C. 2023. Madi: Learning to mask distractions for generalization in visual deep reinforcement learning. *arXiv preprint arXiv:2312.15339*.

Ha, D.; and Schmidhuber, J. 2018. World models. *arXiv preprint arXiv:1803.10122*, 2(3).

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, 1861–1870. Pmlr.

Hafner, D.; Lillicrap, T.; Ba, J.; and Norouzi, M. 2019a. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*.

Hafner, D.; Lillicrap, T.; Fischer, I.; Villegas, R.; Ha, D.; Lee, H.; and Davidson, J. 2019b. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, 2555–2565. PMLR.

Hafner, D.; Lillicrap, T.; Norouzi, M.; and Ba, J. 2020. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*.

Hafner, D.; Pasukonis, J.; Ba, J.; and Lillicrap, T. 2025. Mastering diverse control tasks through world models. *Nature*, 1–7.

Hendrycks, D.; and Gimpel, K. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Houthooft, R.; Chen, X.; Duan, Y.; Schulman, J.; De Turck, F.; and Abbeel, P. 2016. Vime: Variational information maximizing exploration. *Advances in neural information processing systems*, 29.

Janner, M.; Fu, J.; Zhang, M.; and Levine, S. 2019. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2): 99–134.

Kaiser, L.; Babaeizadeh, M.; Milos, P.; Osinski, B.; Campbell, R. H.; Czechowski, K.; Erhan, D.; Finn, C.; Kozakowski, P.; Levine, S.; et al. 2019. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*.

Kendall, A.; and Gal, Y. 2017. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30.

Kidambi, R.; Rajeswaran, A.; Netrapalli, P.; and Joachims, T. 2020. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33: 21810–21823.

Kostrikov, I.; Yarats, D.; and Fergus, R. 2020. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*.

Kurutach, T.; Clavera, I.; Duan, Y.; Tamar, A.; and Abbeel, P. 2018. Model-ensemble trust-region policy optimization. *arXiv preprint arXiv:1802.10592*.

Lakshminarayanan, B.; Pritzel, A.; and Blundell, C. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

Lütjens, B.; Everett, M.; and How, J. P. 2019. Safe reinforcement learning with model uncertainty estimates. In *2019 International Conference on Robotics and Automation (ICRA)*, 8662–8668. IEEE.

Ma, H.; Wu, J.; Feng, N.; Xiao, C.; Li, D.; Hao, J.; Wang, J.; and Long, M. 2023. Harmonydream: Task harmonization inside world models. *arXiv preprint arXiv:2310.00344*.

Moerland, T. M.; Broekens, J.; Plaat, A.; Jonker, C. M.; et al. 2023. Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 16(1): 1–118.

Osband, I.; Blundell, C.; Pritzel, A.; and Van Roy, B. 2016. Deep exploration via bootstrapped DQN. *Advances in neural information processing systems*, 29.

Ovadia, Y.; Fertig, E.; Ren, J.; Nado, Z.; Sculley, D.; Nowozin, S.; Dillon, J.; Lakshminarayanan, B.; and Snoek, J. 2019. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32.

Pasukonis, J.; Lillicrap, T.; and Hafner, D. 2022. Evaluating long-term memory in 3d mazes. *arXiv preprint arXiv:2210.13383*.

Pathak, D.; Agrawal, P.; Efros, A. A.; and Darrell, T. 2017. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, 2778–2787. PMLR.

Pathak, D.; Gandhi, D.; and Gupta, A. 2019. Self-supervised exploration via disagreement. In *International conference on machine learning*, 5062–5071. PMLR.

Richens, J.; Everitt, T.; and Abel, D. 2025. General agents need world models. In *Forty-second International Conference on Machine Learning*.

Schrittwieser, J.; Antonoglou, I.; Hubert, T.; Simonyan, K.; Sifre, L.; Schmitt, S.; Guez, A.; Lockhart, E.; Hassabis, D.; Graepel, T.; et al. 2020. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839): 604–609.

Sekar, R.; Rybkin, O.; Daniilidis, K.; Abbeel, P.; Hafner, D.; and Pathak, D. 2020. Planning to explore via self-supervised world models. In *International conference on machine learning*, 8583–8592. PMLR.

Seo, J.; Nakamura, K.; and Bajcsy, A. 2025. UNISafe: Uncertainty-aware Latent Safety Filters for Avoiding Out-of-Distribution Failures. In *Second Workshop on Out-of-Distribution Generalization in Robotics at RSS 2025*.

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

Sutton, R. S.; Barto, A. G.; et al. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Talvitie, E. 2017. Self-correcting models for model-based reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Tang, H.; Houthooft, R.; Foote, D.; Stooke, A.; Xi Chen, O.; Duan, Y.; Schulman, J.; DeTurck, F.; and Abbeel, P. 2017. # exploration: A study of count-based exploration for deep reinforcement learning. *Advances in neural information processing systems*, 30.

Tassa, Y.; Doron, Y.; Muldal, A.; Erez, T.; Li, Y.; Casas, D. d. L.; Budden, D.; Abdolmaleki, A.; Merel, J.; Lefrancq, A.; et al. 2018. Deepmind control suite. *arXiv preprint arXiv:1801.00690*.

Ye, W.; Liu, S.; Kurutach, T.; Abbeel, P.; and Gao, Y. 2021. Mastering atari games with limited data. *Advances in neural information processing systems*, 34: 25476–25488.

Yu, T.; Kumar, A.; Rafailov, R.; Rajeswaran, A.; Levine, S.; and Finn, C. 2021. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34: 28954–28967.

Yu, T.; Thomas, G.; Yu, L.; Ermon, S.; Zou, J. Y.; Levine, S.; Finn, C.; and Ma, T. 2020. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33: 14129–14142.

Zheng, R.; Wang, X.; Sun, Y.; Ma, S.; Zhao, J.; Xu, H.; Daumé III, H.; and Huang, F. 2024. TACO: Temporal Latent Action-Driven Contrastive Loss for Visual Reinforcement Learning. *Advances in Neural Information Processing Systems*, 36.