Uncertainty-aware Latent Safety Filters
for Avoiding Out-of-Distribution Failures

Junwon Seo, Kensuke Nakamura, Andrea Bajcsy
Carnegie Mellon University
{junwonse, kensuken, abajcsy}@andrew.cmu.edu

Abstract: Recent advances in generative world models have enabled classical safe control methods, such as Hamilton-Jacobi (HJ) reachability, to generalize to complex robotic systems operating directly from high-dimensional sensor observations. However, obtaining comprehensive coverage of all safety-critical scenarios during world model training is extremely challenging. As a result, latent safety filters built on top of these models may miss novel hazards and even fail to prevent known ones, overconfidently misclassifying risky out-of-distribution (OOD) situations as safe. To address this, we introduce an uncertainty-aware latent safety filter that proactively steers robots away from both known and unseen failures. Our key idea is to use the world model's epistemic uncertainty as a proxy for identifying unseen potential hazards. We propose a principled method to detect OOD world model predictions by calibrating an uncertainty threshold via conformal prediction. By performing reachability analysis in an augmented state space–spanning both the latent representation and the epistemic uncertainty– we synthesize a latent safety filter that can reliably safeguard arbitrary policies from both known and unseen safety hazards. In simulation and hardware experiments on vision-based control tasks with a Franka manipulator, we show that our uncertainty-aware safety filter preemptively detects potential unsafe scenarios and reliably proposes safe, in-distribution actions. Video results can be found on the project website: https://cmu-intentlab.github.io/UNISafe

Keywords: safe control, uncertainty quantification, world models

1 Introduction

Robots operating in complex open-world environments must interact safely with the world based on high-dimensional sensor observations. A promising approach to scale safe control to such settings is to learn a world model (WM) [1] that jointly compresses observations into compact latent representations and predicts their dynamics, allowing the robot to anticipate the consequences of candidate actions to prevent unsafe ones [2]. However, without unlimited unsafe exploration, the WM's training data can fail to capture the full range of possible safety hazards. For example, in the *Jenga* game (right, Fig. 1), most of the ways in which the tower can fall are not seen during training. During interaction, if the robot fails to reliably predict how its actions can lead to such *out-of-distribution (OOD)* scenarios, it may inadvertently execute actions that lead to unsafe outcomes [3, 4].

One way to address this model uncertainty is through OOD detection, which identifies when the robot encounters anomalous observations or generates uncertain predictions [5, 6, 7]. However, on its own, OOD detection lacks actionable mitigation strategies, leaving robots *aware* of their uncertainty yet unable to *act* appropriately. Here, safe control methods such as Hamilton-Jacobi (HJ) reachability analysis [8, 9] offer a complementary approach by synthesizing fallback policies that proactively enforce safety constraints, keeping the system within control-invariant sets. Yet, they typically assume a perfect state representation and a faithful dynamics model, assumptions that may not hold in OOD scenarios when relying on a world model for safe control. To bridge this gap,
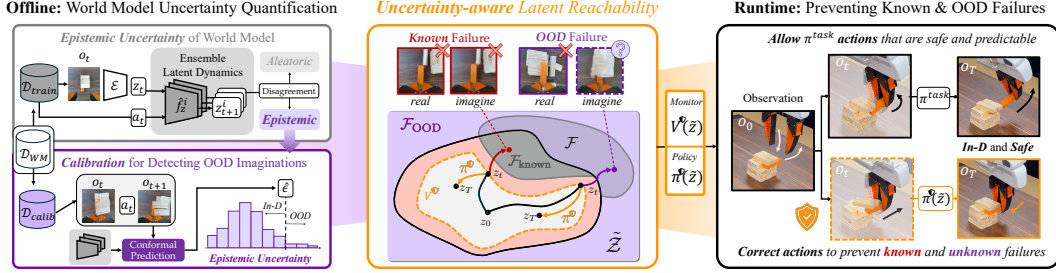
Figure 1: *Left*: We quantify the world model's epistemic uncertainty for detecting unseen failures in latent space and calibrate an uncertainty threshold via conformal prediction, resulting in an *OOD failure set*, $\mathcal{F}_{OOD}$. *Center*: Uncertainty-aware latent reachability analysis synthesizes a safety monitor $V^{\bullet}$ and fallback policy $\pi^{\bullet}$ that steers the system away from both known and OOD failures. *Right*: Our safety filter reliably safeguards arbitrary task policies during hard-to-model vision-based tasks, like a teleoperator playing the game of Jenga.

we argue that safety constraints for latent-space control should be augmented to identify unreliable model predictions, enabling the synthesis of a *safety filter* that prevents the system from entering both *known failures* and potentially unsafe *OOD failures*.

In this work, we propose *UNcertainty-aware Imagination for Safety filtering* (*UNISafe*): a policy-agnostic safety mechanism that reliably steers robots away from known and unseen safety hazards using a latent world model [1, 10]. Our key idea is to use the world model's epistemic uncertainty as a proxy for identifying unseen potential hazards. We propose a principled method to quantify the epistemic uncertainty of the world model and detect unreliable world model predictions by calibrating an uncertainty threshold via conformal prediction. By performing reachability analysis in an augmented state space spanning both the latent states and the uncertainty, we synthesize a safety filter that can reliably prevent a system from entering both predictable and unforeseen failure modes.

We evaluate our framework in simulation and hardware on three vision-based safe-control tasks. We find that *UNISafe* effectively prevents failures with world models trained on an offline dataset with limited coverage. Importantly, by penalizing overly optimistic safety evaluations of OOD scenarios during reachability analysis, our safety filter preemptively detects potential safety risks and proposes reliable backup actions, consistently guiding the system toward safe, in-distribution behaviors.

2 Related Works

Out-of-distribution Detection for Robotics. Data-driven control often exhibits unreliable behavior when encountering data that deviates from its training distribution [6, 5, 3, 11, 4]. To detect such out-of-distribution (OOD) conditions, uncertainty is estimated via pre-trained feature spaces [12, 13], reconstruction [14, 15, 16, 17], density estimation [18, 19, 20, 21, 22], or ensembles [23, 24, 25, 26]. While these methods can detect OOD and serve as runtime monitors [27, 28, 29, 3], they often lack control invariance, limiting them to passive detection rather than proactive failure prevention. Moreover, they typically do not distinguish between epistemic uncertainty (i.e., lack of knowledge) and aleatoric uncertainty (i.e., inherent noise) [23, 24, 25], while capturing epistemic uncertainty is critical for reliable OOD detection [30, 31]. To bridge this gap, we quantify the epistemic uncertainty of a world model [30, 32, 31] to formulate a constraint, enabling reachability analysis to synthesize control strategies that prevent the system from OOD scenarios.

Safety Filtering. Safety filtering is a control-theoretic approach for safeguarding robotic systems from unsafe conditions [8, 33, 34, 9, 35, 36]. While they can provide robust safety assurances under model uncertainty [34, 37, 38, 39, 40], they focus on worst-case disturbances, addressing aleatoric uncertainty rather than epistemic uncertainty of the model. Self-supervised [41] and reinforcement learning methods [42, 43] have been used to scale safety filtering to high-dimensional systems, but these approaches typically rely on known system dynamics with simple safety specifications [38, 44] or online rollouts in simulators [45, 46, 47]. To generalize safety filters with complex dynamics and constraints, latent world models [1] have been used [22, 48, 2], but the epistemic uncertainty of the

learned model can compromise reliability [25]. Recent works prevent the system from entering OOD states [20, 22], but they restrict in-distribution to safe trajectories or do not construct constraints with calibrated OOD detection [49, 22, 50], limiting their scalability to complex settings. Our method leverages calibrated OOD detection, enabling reliable prevention of both known and unseen failures.

3 Setup: Latent Safety Filters via Reachability Analysis in a World Model

In this section, we briefly introduce the computation and use of latent safety filters [2] for systems with hard-to-model dynamics and safety specifications inferred from high-dimensional observations.

Latent World Model. To model complex systems, we train a world model [1] using a fixed offline dataset of robot–environment interactions, $\mathcal{D}_{\text{train}} := \{\{(o_t, a_t, l_t)\}_{t=1}^{T}\}_{i=1}^{N_{\text{train}}} \subset \mathcal{D}_{\text{WM}}$, consisting of trajectories with high-dimensional observations $o \in \mathcal{O}$, robot actions $a \in \mathcal{A}$, and failure labels $l \in \{-1, 1\}$ indicating visible safety hazards. The latent world model consists of an encoder $\mathcal{E}$ that maps an observation into the latent representation $z \in \mathcal{Z}$ and a latent dynamics model:

$$\text{Encoder: } z_t \sim \mathcal{E}(z_t \mid \hat{z}_t, o_t) \quad \text{Dynamics: } \hat{z}_t \sim f_z(\hat{z}_t \mid z_{t-1}, a_{t-1}) \quad \text{Failure: } l_t = \ell_z(z_t). \quad (1)$$

Safety Specification ($\mathcal{F}$). Hard-to-model safety constraints (e.g., spilling, block toppling) are specified in the latent space via a failure set $\mathcal{F} := \{z : \ell_z(z) \leq 0\} \subset \mathcal{Z}$ encoded via the zero-sublevel set of a margin function ℓ_z in (1). In practice, ℓ_z is a binary classifier learned with $\mathcal{D}_{\text{train}}$.

Computing Latent Safety Filters ($\pi^{\mathbf{O}}, V^{\mathbf{O}}$). Following [2], we conduct HJ reachability analysis [8, 35] in the latent space to synthesize both a safety value function $V^{\mathbf{O}} : \mathcal{Z} \to \mathbb{R}$ and a safety-preserving policy $\pi^{\mathbf{O}} : \mathcal{Z} \to \mathcal{A}$, entirely within the imagination of the world model. Specifically, we solve the fixed-point safety Bellman equation with a time discounting factor $\gamma \in [0, 1)$ [42]:

$$V^{\mathbf{O}}(z_t) = (1 - \gamma)\ell_z(z_t) + \gamma \min\left\{\ell_z(z_t), \max_{a_t \in \mathcal{A}} V^{\mathbf{O}}(\hat{z}_{t+1})\right\}, \quad \pi^{\mathbf{O}}(z) = \arg\max_{a \in \mathcal{A}} V^{\mathbf{O}}(\hat{z}_{t+1}), \quad (2)$$

where $\hat{z}_{t+1}$ is sampled from f_z. Intuitively, $V^{\mathbf{O}}$ represents how close the robot comes to failure starting from z_t despite its best efforts, and $\pi^{\mathbf{O}}$ is a maximally safety-preserving policy. Note that, in contrast to typical RL for reward maximization, this optimization performs a *min-over-time* to *remember* safety-critical events. Therefore, $V^{\mathbf{O}} < 0$ indicates that the robot is doomed to fail, while $V^{\mathbf{O}} \geq 0$ means that there exists a safety-preserving action to prevent failures (e.g., returned by $\pi^{\mathbf{O}}$).

Runtime Safety Filtering. At runtime, the latent safety filter safeguards an arbitrary task policy π^{task} based on the current observations and proposed action. By checking $V^{\mathbf{O}}$ as a monitor with a small margin $\delta \approx 0$, the safety filter either allows π^{task} or overrides it with the fallback policy $\pi^{\mathbf{O}}$:

$$a^{\text{exec}} := \mathbb{1}\{\underbrace{V^{\mathbf{O}}(z') > \delta}_{\pi^{\text{task}} \text{ is safe, proceed}}\} \pi^{\text{task}} + \mathbb{1}\{\underbrace{V^{\mathbf{O}}(z') \leq \delta}_{\pi^{\text{task}} \text{ is unsafe, fallback to } \pi^{\mathbf{O}}}\} \pi^{\mathbf{O}}(z), \quad z' \sim f_z(z, \pi^{\text{task}}). \quad (3)$$

Challenge: Unreliable WM Can Result in OOD Failures. While latent safety filters can compute control strategies that prevent hard-to-model failures, their training (2) and runtime filtering (3) rely on imagined futures generated by the latent dynamics model. However, a pretrained world model can hallucinate in uncertain scenarios where it lacks knowledge, leading to *OOD failures*.
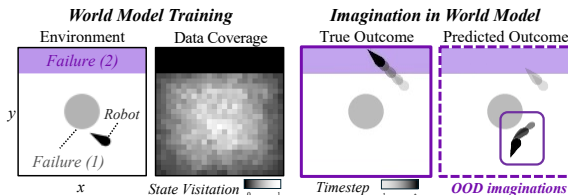

Figure 2: WM imaginations can lead to *OOD Failures*.

Consider the simple example in Fig. 2 where a Dubins car must avoid two failure sets: a circular grey and a rectangular purple region. The world model is trained with RGB images of the environment and angular velocity actions, but the model training data is limited, lacking knowledge of the robot entering the purple failure set. When the world model imagines an action sequence in which the robot enters this region (*third* image of Fig. 2), the world model hallucinates as soon as the scenario goes out-of-distribution: the robot teleports away from the failure region and to a safe state (*rightmost* image of Fig. 2). This phenomenon leads to latent safety filters that cannot prevent unseen failures, and even known failures, due to optimistic safety estimates of uncertain out-of-distribution scenarios.

4 Formalizing Uncertainty-aware Latent Safety Filters

To formalize reliable safe control in latent space, our key idea is to use the epistemic uncertainty of the world model as a proxy for detecting safety hazards not represented in the training dataset. Specifically, we augment the safety specification that accounts for **known failures**–scenarios the world model can anticipate with confidence–with **OOD failures**: potentially unsafe, out-of-distribution scenarios where the model's imaginations are highly uncertain and lose their reliability.

Uncertainty-aware Latent Space & Dynamics. We quantify the epistemic uncertainty of the world model, $u \in \mathbb{R}$, to identify OOD imaginations of the world model. To assess the reliability of latent dynamics predictions, the uncertainty should capture the *dynamics uncertainty* induced by latent–action transitions (z, a). This is crucial because generative world models are prone to hallucination, often producing in-distribution predictions when exposed to OOD inputs. Therefore, OOD detection methods that rely solely on the predicted latent state z are overconfident, as predicted latents from OOD scenarios are projected into in-distribution representations, as depicted in Fig. 2.

We then augment the latent space to incorporate this epistemic uncertainty, $\tilde{z}_t = (z_t, u_t)^\top \in \mathcal{Z} \times \mathbb{R}$. This formulation enables modeling both *known failures* $\mathcal{F}_{\text{known}} := \{\tilde{z} \mid \ell_z(z) < 0\}$, which are predictable with the learned model, and *OOD failures* $\mathcal{F}_{\text{OOD}} := \{\tilde{z} \mid u > \epsilon\}$ which are OOD imaginations with quantified uncertainty exceeding a predefined threshold ϵ. The latent dynamics and safety margin function are extended to operate in the augmented latent space:

$$f_{\tilde{z}}(\tilde{z}_{t+1} \mid \tilde{z}_t, a_t) = \left[f_z(z_{t+1} \mid z_t, a_t), \, \mathrm{D}(z_t, a_t) \right]^\top, \quad \ell_{\tilde{z}}(\tilde{z}_t) = \min\left\{ \ell_z(z_t), \kappa\left(\epsilon - u_t\right) \right\}, \quad (4)$$

with $\kappa \in \mathbb{R}^+$. The uncertainty $u_{t+1} = \mathrm{D}(z_t, a_t)$ is obtained via measuring reliability of a transition (described in Sec 5.1) and the uncertainty-aware failure set $\tilde{\mathcal{F}}$ is represented via the zero sub-level set of the augmented margin function: $\tilde{\mathcal{F}} := \mathcal{F}_{\text{known}} \cup \mathcal{F}_{\text{OOD}} = \{\tilde{z} \mid \ell_{\tilde{z}}(\tilde{z}) < 0\}$.

Uncertainty-aware Latent Reachability Analysis. We compute a latent safety filter via Eq. 2 and perform safety filtering as in Eq. 3, but use the uncertainty-aware latent dynamics from Eq. 4 throughout. This formulation ensures that the value function assigns negative values to OOD scenarios where the uncertainty exceeds a predefined threshold. By explicitly penalizing such transitions, the resulting safety filter discourages the system from entering OOD regions while also avoiding known, predictable failures. This mitigates overly optimistic imaginations and enables the filter to reliably learn both a safety monitor and a fallback policy that proposes safe, in-distribution actions.

5 Computing Uncertainty-aware Latent Safety Filters

While the prior section formalize the uncertainty-aware latent safety filter by augmenting the latent space with the world model's epistemic uncertainty, we face two key challenges when instantiating our framework in practice: (i) How can we quantify the epistemic uncertainty of the world model? (Sec. 5.1) and (ii) How can we ensure the OOD threshold ϵ is appropriately *calibrated* to reliably detect OOD failures based on the estimated measures of epistemic uncertainty? (Sec. 5.2).

5.1 Quantifying Epistemic Uncertainty in World Models

Training a Probabilistic Ensemble Latent Predictor. To capture the epistemic uncertainty of the world model, we employ an ensemble of next-latent predictors, $E := \{\hat{f}_z^k\}_{k=1}^K$, which is a separate module regressing the pretrained latent dynamics f_z. Each ensemble member is initialized with distinct parameters ψ_k and trained to predict the next latent z_{t+1} given the current latent z_t and action a_t with Gaussian negative log-likelihood loss (see A.2 & A.3 for more details.):

$$\text{Latent Predictor:} \quad z_{t+1}^k \sim \hat{f}_z^k(z_t, a_t; \psi_k), \quad \hat{f}_z^k(z_t, a_t; \psi_k) := \mathcal{N}(\mu_{\psi_k}(z_t, a_t), \Sigma_{\psi_k}(z_t, a_t)), \quad (5)$$

where μ_{ψ_k} and Σ_{ψ_k} denote the predicted mean and diagonal covariance, respectively. Note that the covariance models the *aleatoric uncertainty* inherent in the latent dynamics due to partial observability and stochasticity. The ensemble latent predictor is trained on latent transitions $\{\{(z_t, a_t, z_{t+1})\}_{t=1}^{T-1}\}_{i=1}^{N_{\text{train}}}$, encoded from a pretrained latent world model with the same offline dataset $\mathcal{D}_{\text{train}}$ used for world model training.

4

Epistemic Uncertainty Quantification. While empirical variance over ensemble predictions is widely used as an uncertainty measure [23, 24, 28, 39], this conflates aleatoric uncertainty (i.e., inherent uncertainty of latent dynamics) with epistemic uncertainty (i.e., uncertainty arising from a lack of knowledge). Since our goal is to control away from *OOD failures*, which the world model has never encountered and thus cannot reliably predict, it is essential to focus explicitly on the model's *epistemic uncertainty* to form our constraint on the latent space. Otherwise, the safety filter may fail to reject unsafe OOD imaginations or become overly conservative in response to intrinsic stochasticity. Following [30, 31], we quantify the epistemic uncertainty of the latent dynamics $D(z_t, a_t)$ via the Jensen-Rényi Divergence (JRD) [51] of the ensemble predictions with Rényi entropy H_α:

$$
\underbrace{u_{t+1} = D(z_t, a_t)}_{\text{epistemic uncertainty}} := \underbrace{H_\alpha \left(\sum_{k=1}^{K} \frac{1}{K} \hat{f}_z^k \right)}_{\text{total uncertainty}} - \underbrace{\sum_{k=1}^{K} \frac{1}{K} H_\alpha \left(\hat{f}_z^k \right)}_{\text{aleatoric uncertainty}}, \quad H_\alpha(Z) = \frac{1}{1-\alpha} \log \int p(z)^\alpha \mathrm{d}z,
\tag{6}
$$

where Z is a random variable. In general, computing the disagreement between an ensemble of Gaussian distributions lacks a closed-form solution [30, 31], and Monte Carlo sampling approximations are computationally expensive for high-dimensional latent spaces. Therefore, we adopt JRD with $\alpha = 2$, which has a closed-form expression [52] for GMM (see A.2 for further details).

5.2 Detecting Out-of-Distribution Imaginations via Conformal Prediction

Recall that during reachability analysis, *OOD failures* are detected when the uncertainty of an imagined transition exceeds a threshold, $\mathcal{F}_{\text{OOD}} = \{\tilde{z} \mid u > \epsilon\}$. However, setting this threshold is nontrivial: too strict a threshold can result in high false-positive rates (misclassifying in-distribution transitions as OOD), leading to overly conservative filters; too loose a threshold may fail to detect true OOD transitions. We employ conformal prediction (CP) [53, 54] to automatically calibrate the threshold $\epsilon \in \mathbb{R}$ in a principled way, using a held-out calibration dataset $\mathcal{D}_{\text{calib}} = \mathcal{D}_{\text{WM}} \setminus \mathcal{D}_{\text{train}}$.

In-distribution Recall Guarantee via Class-Conditioned Conformal Prediction. CP typically requires the calibration set $\mathcal{D}_{\text{calib}}$ to contain both inputs to the prediction model (e.g., (z_t, a_t)) and their corresponding ground-truth labels (e.g., ID or OOD). Unfortunately, in our setting, true OOD labels are, by definition, not accessible. As such, we assume the calibration dataset consists only of in-distribution transitions. Formally, we adopt class-conditioned conformal prediction [55, 44] to calibrate the uncertainty threshold ϵ, providing conditional recall guarantees for detecting in-distribution transitions with user-defined confidence level $\alpha_{\text{cal}} \in [0, 1]$:

$$
\mathbb{P}\left(D(z_t, a_t) < \hat{\epsilon} \mid (z_t, a_t) \in \mathcal{D}_{\text{WM}}\right) \geq 1 - \alpha_{\text{cal}},
\tag{7}
$$

Intuitively, conformal prediction can help us select an uncertainty threshold $\hat{\epsilon}$ such that in-distribution latent transitions can be detected with probability at least $1 - \alpha_{\text{cal}}$. Conversely, latent transitions with uncertainty greater than this threshold can be interpreted as OOD.

Trajectory-Level Calibration. While standard class-conditioned conformal prediction assumes exchangeability of the data, this assumption does not hold in our setting, as each transition depends on the full history of latent states and actions. To address this, we adopt a trajectory-level calibration approach [56], assuming that the calibration trajectories $\tau_i = \{(z_t, a_t)\}_{t=1}^{T} \in \mathcal{D}_{\text{calib}}$ are drawn i.i.d. from the same distribution as the world model training data, $\{\tau_i\}_{i=1}^{N} \overset{\text{iid}}{\sim} \mathcal{D}_{\text{WM}}$. For each trajectory, we define the trajectory-level nonconformity score $Q_{\tau_i}^{\alpha_{\text{trans}}}$ as the $(1 - \alpha_{\text{trans}})$-quantile of the set of quantified epistemic uncertainties $\{u_t\}_{t=1}^{T}$. This ensures that at most an α_{trans} fraction of a trajectory's uncertainty values exceed $Q_{\tau_i}^{\alpha_{\text{trans}}}$, making the estimate more robust to noise in uncertainty predictions. We then determine the calibration threshold $\hat{\epsilon}$ as the $(1 - \alpha_{\text{cal}})$-quantile of the set $\{Q_{\tau_i}^{\alpha_{\text{trans}}}\}_{i=1}^{N}$ by selecting the $\lceil (1 - \alpha_{\text{cal}})(N + 1) \rceil$-th smallest value over trajectories. With the exchangeability assumption between calibration and test trajectories, conformal prediction guarantees that for a new test trajectory $\tau_{\text{test}} = \{(z_t^{\text{test}}, a_t^{\text{test}})\}_{t=1}^{T}$, the following probabilistic guarantee holds:

$$
\mathbb{P}_{\tau_{\text{test}} \sim \mathcal{D}_{\text{WM}}}\left(Q_{\tau_{\text{test}}}^{\alpha_{\text{trans}}} \leq \hat{\epsilon}\right) = \mathbb{P}_{\tau_{\text{test}} \sim \mathcal{D}_{\text{WM}}}\left(\mathbb{P}_t\left\{D(z_t^{\text{test}}, a_t^{\text{test}}) \leq \hat{\epsilon}\right\} \geq 1 - \alpha_{\text{trans}}\right) \geq 1 - \alpha_{\text{cal}}.
\tag{8}
$$

Although this guarantee applies only to in-distribution data, it ensures a low false positive rate by bounding the probability of misclassifying in-distribution transitions as OOD. Specifically, the probability that the trajectory-level nonconformity score exceeds the threshold for in-distribution data is bounded by $\mathbb{P}_{\tau_{\text{test}} \sim \mathcal{D}_{\text{WM}}} \left(Q_{\tau_{\text{test}}}^{\alpha_{\text{trans}}} \geq \hat{\epsilon} \right) \leq \alpha_{\text{cal}}$. As a result, any transition with a quantified epistemic uncertainty above $\hat{\epsilon}$ can be reliably classified as OOD, since such events are guaranteed to be rare under the in-distribution distribution (see Appendix C for details).

6   Simulation & Hardware Experiments

6.1   Simulation: A Benchmark Safe Control Task with a 3D Dubins Car

We first conduct experiments with a low-dimensional, benchmark safe navigation task where privileged information about the state, dynamics, safe set, and safety controller is available.

Privileged Dynamics: Dubins Car.   Let the privileged Dubins car state be $s = [p_x, p_y, \theta]$, with discrete-time dynamics $s_{t+1} = s_t + \Delta t \, [v \cos(\theta_t), \, v \sin(\theta_t), \, a_t]$. We assume a fixed velocity $v = 1 \, \text{m/s}$, time step $\Delta t = 0.05 \, \text{s}$, and discrete action space $a_t \in \mathcal{A} = \{-1.25, 0, 1.25\}$ rad/s.

Evaluation & Metrics.   Given access to ground-truth dynamics, we compute the ground-truth safety value function using grid-based methods [57], enabling direct evaluation of the safety monitor $V^{\mathbf{0}}$)'s classification accuracy across all three state dimensions. To assess $\pi^{\mathbf{0}}$, we roll out the learned policies from safe initial states with positive ground-truth safety values and measure the safety rate by checking whether the resulting trajectories remain safe without violating constraints.

Baselines.   We evaluate *UNISafe*, which learns the uncertainty-aware unsafe set $\tilde{\mathcal{U}}$ from the failure set $\tilde{\mathcal{F}} = \mathcal{F}_{\text{known}} \cup \mathcal{F}_{\text{OOD}}$, against *LatentSafe* [2], which considers only known failures, $\mathcal{F}_{\text{known}}$. Also, we compare *JRD* with other OOD detection baselines to assess uncertainty quantification. *TotalUncertainty* compute variance of mean predictions across the ensemble without isolating aleatoric components [28, 25, 26].

	Method	TPR↑	TNR↑	B.Acc.↑
UQ	*TotalUncertainty*	0.88	0.97	0.93
	MaxAleatoric	0.78	0.88	0.83
	DensityEst (z, a)	0.98	0.87	0.92
	DensityEst (z)	0.99	0.56	0.77
Calib	*JRD* $(\epsilon = \hat{\epsilon})$	0.93	0.95	0.94
	JRD $(\epsilon = \hat{\epsilon} + 0.3)$	0.98	0.43	0.71
	JRD $(\epsilon = \hat{\epsilon} - 0.3)$	0.85	0.96	0.90

Table 1: Safety value function quality with different OOD detection methods.

MaxAleatoric uses the maximum predicted ensemble variance, $\max_k \|\Sigma_{\psi_k}(z_t, a_t)\|_F$, representing aleatoric uncertainty [24, 58]. *DensityEst* employs neural spline flows [18, 20] to compute likelihoods of (z) or (z, a) for OOD detection. For every method, thresholds are calibrated with the same held-out calibration dataset and Double DQN (DDQN) [59] is used to train all the safety value function (See B.2 and D.1 for more details).

UNISafe **reliably identifies the OOD failure** $\mathcal{F}_{\mathbf{OOD}}$**.**   To evaluate OOD detection, we first consider a setting where failure states are never observed by $\mathcal{D}_{\text{train}}$. The ground-truth failure set is defined as $|p_y| > 0.6$, while the offline dataset contains only 1000 safe trajectories that never enter this region, making the failure set entirely OOD. As shown in Fig. 3, our method reliably infers the OOD failure set from the quantified uncertainty, and the resulting safety value function accurately identifies the unsafe region. Table 1 shows that *JRD* achieves the highest balanced accu-

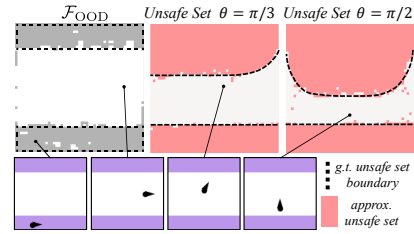
Figure 3:   *Dubins Car with* $\mathcal{F}_{\text{OOD}}$ *only.* OOD detection successfully identifies $\mathcal{F}_{\text{OOD}}$ and unsafe set.

racy (B.Acc.) compared to other OOD detection methods, whereas methods not targeting epistemic uncertainty exhibit higher FPRs and lower balanced accuracies. Additionally, *DensityEst* based only on z shows low TNR, highlighting the necessity of latent-action transition-based OOD detection.

A calibrated OOD threshold yields a higher quality value function.   We perturb our calibrated threshold $\hat{\epsilon}$ to obtain $\epsilon = \hat{\epsilon} \pm 0.3$ and study the sensitivity of the value function to threshold selection. Table 1 shows that our automatic calibration process selects thresholds that lead to value functions with both high TPR and TNR, unlike the uncalibrated thresholds that degrade accuracy.

UNISafe **robustly learns safety filters despite high uncertainties in the world models.** We evaluate whether our method can synthesize a robust safety filter with uncertain world due to limited data coverage. In this setting, the vehicle must avoid a circular obstacle of radius $0.5\,\mathrm{m}$ at center, with the failure set defined as $p_x^2 + p_y^2 < 0.5^2$, and $\mathcal{D}_{\text{train}}$ consists of both safe and unsafe trajectories. We construct a dataset of 1000 expert trajectories that never enter the ground-truth unsafe sets and 50 random trajectories that may include failure states. Expert trajectories are generated using the ground-truth safety value, applying fallback actions near the unsafe boundary and random actions elsewhere,

inducing high uncertainty around the unsafe boundary. Fig. 4 shows that *UNISafe* robustly learns the safety monitor with higher balanced accuracy, whereas *LatentSafe* overconfidently misclassifies unsafe states as safe. In rollouts from 181 challenging safe initial states, where the vehicle is oriented toward failure, *UNISafe* also achieves higher safety rates. (See D.1 & E.1 for more details and analysis.)
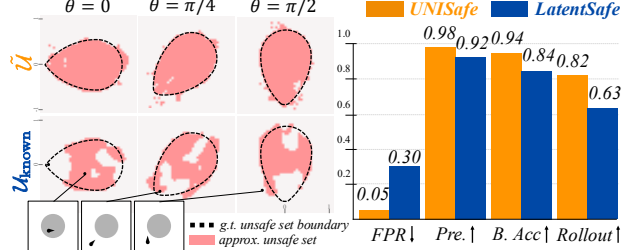

Figure 4: *UNISafe* **vs** *LatentSafe*. Without *OOD failures*, the safety value learned from the unreliable world model leads to higher FPR, overconfidently classifying unsafe states as safe.

6.2 Simulation: Vision-Based Block Plucking

Setup. We scale our method to a visual manipulation task using IsaacLab [60], where a Franka manipulator must pluck the middle block from a stack of three while ensuring the top one remains on the bottom one. Observations consist of images from a wrist-mount and a tabletop camera, with 7-D proprioceptive inputs. Actions are a 6-DoF end-effector delta pose with a discrete gripper command.

Evaluations. We adopt DreamerV3 [10] as our task policy π^{task}, trained with a dense reward signal to achieve the task with a soft penalty for failures. The training dataset $\mathcal{D}_{\text{train}}$ consists of 3000 trajectories comprising both safe and unsafe behavior rolled out from π^{task}. We adopt Soft Actor-Critic (SAC) [61] as our solver for latent reachability. For evaluation, task policy rollouts are filtered using the safety filter with $\delta = 0.1$, evaluated over 1000 randomly sampled initial conditions. (See B.2 and D.2 for details.)

Figure 5: *Block Plucking*. *UNISafe* prevents failure with ID actions, while *LatentSafe* fails to preempt it by overestimating unsafe OOD actions.

Baselines. As in Sec. 6.1, we compare *UNISafe* with *LatentSafe* [2] trained on the same dataset with and without $\mathcal{F}_{\text{OOD}}$, as well as different OOD detection baselines. *SafeOnly* learns a WM and latent safety filter only on successful demonstrations without $\mathcal{F}_{\text{known}}$, implicitly treating all failures as $\mathcal{F}_{\text{OOD}}$, as in [20, 22, 4]. Also, we adapt *CQL* [62] and *COMBO* [63] to optimize Eq. 2 with conservative losses, but without uncertainty quantification.

UNISafe **minimizes failure by preventing safety overestimation.** Table 2 shows that *UNISafe*, which incorporates both known and OOD failures, achieves the lowest failure rates and model errors. In contrast, *LatentSafe* overestimates the safety of OOD actions, leading to unsafe action proposals, as shown in Fig. 5. *SafeOnly* shows limited effectiveness, showing OOD detection from success-only data is insufficient in complex settings. Offline RL with conservative losses performs even worse than *LatentSafe*, indicating that conservatism alone cannot replace failure set identification.

Quantifying epistemic uncertainty leads to safe but non-conservative behaviors. While all OOD detection methods improve filtering performance over *LatentSafe*, targeting aleatoric uncertainty (*TotalUncertainty* and *MaxAleatoric*) tends to be overly conservative, resulting in higher incompletion rates and more frequent interventions. In contrast, *UNISafe* with *JRD* explicitly targets epistemic uncertainty and achieves the most reliable performance. *DensityEst* shows limited performance, highlighting the challenge of modeling likelihood in high-dimensional latent spaces.
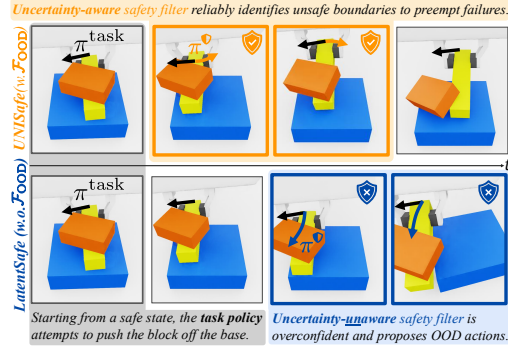
7

Method	f_z	$\mathcal{F}_{\text{known}}$	$\mathcal{F}_{\text{OOD}}$	Safe Success (↑)	Failure (↓)	Incompletion	Filtered (%)	Model Error (↓)
No Filter (π^{task})	-	-	-	0.58	0.41	0.01	0.0 ± 0.0	59.3 ± 3.3
CQL [62]	✗	✓	✗	0.63	0.33	0.04	2.3 ± 0.9	50.9 ± 11.5
COMBO [63]	✓	✓	✗	0.47	0.41	0.12	54.8 ± 6.8	51.6 ± 12.8
SafeOnly	✓	✗	✓	0.71	0.28	0.01	13.5 ± 3.5	46.9 ± 2.6
LatentSafe [2]	✓	✓	✗	0.68	0.30	0.01	7.2 ± 2.6	60.2 ± 4.7
UNISafe (*TotalUncertainty*)	✓	✓	✓	0.54	0.18	0.28	50.9 ± 7.1	39.1 ± 4.1
UNISafe (*MaxAleatoric*)	✓	✓	✓	0.64	0.25	0.11	38.7 ± 7.0	41.4 ± 9.1
UNISafe (*DensityEst*)	✓	✓	✓	0.66	0.24	0.10	27.9 ± 5.1	41.4 ± 5.2
UNISafe (*JRD*)	✓	✓	✓	0.72	0.20	0.08	37.7 ± 6.7	43.1 ± 1.2

Table 2: *Rollout Results on Block Plucking.* Safe success is plucking a block without failure, and incompletion is a timeout without success or failure. The average world model training loss per trajectory is reported as a proxy for uncertainty. Safety filter is most effective when f_z imaginations incorporate both $\mathcal{F}_{\text{known}}$ and $\mathcal{F}_{\text{OOD}}$.

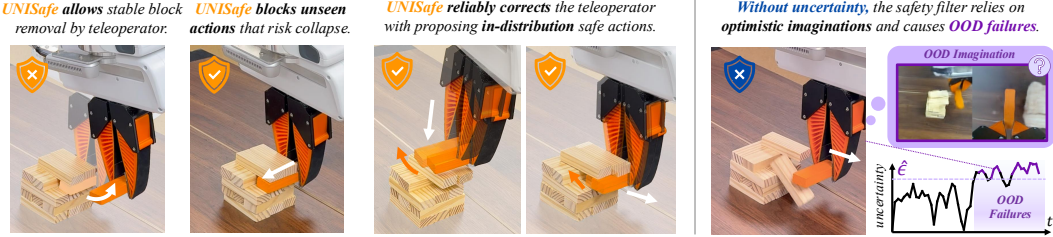## 6.3 Hardware: Vision-based *Jenga* with a Robotic Manipulator

Figure 6: *Teleoperator Playing Jenga with Safety Filters.* *UNISafe* enables non-conservative yet effective filtering of the teleoperator's actions, ensuring the system remains within the in-distribution regions. In contrast, the uncertainty-unaware safety filter *LatentSafe* optimistically treats uncertain actions as safe, leading to failure.

Setup. We evaluate our method on a real-world robotic manipulation task using a fixed-base Franka Research 3 arm, equipped with a third-person camera and a wrist-mounted camera. The robot must extract a target block from a tower without collapsing, then place it on top. For $\mathcal{D}_{\text{train}}$, we collect 720 trajectories: 150 random (no contact), 480 successful, and 90 failure cases. (See D.3 for details.)

UNISafe **reliably filters both known and unseen failures.** First, a teleoperator is π^{task}, controlling the end-effector pose and gripper while assisted by *UNISafe*. As shown in Fig. 6, the teleoperator can freely execute safe behaviors, which require careful tilting and precise block manipulation that are non-trivial to perform. When erratic or OOD actions are attempted, posing a risk of tower collapse, *UNISafe* reliably intervenes to correct the behavior and maintain stability within the in-distribution region. In contrast, *LatentSafe* fails to preemptively detect such boundaries due to optimistic OOD imagination, ultimately allowing high-uncertainty actions. Next, we quantitatively evaluate filtering by replaying 50 failure trajectories as π^{task} that result in tower collapse. The corresponding action sequences are replayed as a task policy with either *UNISafe* or *LatentSafe* as the safety filter. Fig. 7 shows that *UNISafe* leads to lower failure rates and maintains low model uncertainty.
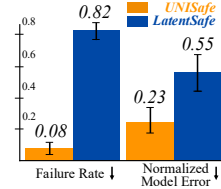

Figure 7: Filtering π^{task} on hardware.

7 Conclusion

In this work, we propose *UNISafe*, a framework for reliable latent-space safe control that *unifies* reachability analysis in a latent world model with OOD detection of the world model predictions. To detect unreliable out-of-distribution imaginations of the world model, we introduce a principled method to quantify the world model's epistemic uncertainty and calibrate a threshold. We then augment the latent space with epistemic uncertainty and perform an uncertainty-aware latent reachability analysis to synthesize a safety filter that reliably safeguards arbitrary policies from both known failures and unseen safety hazards. We demonstrate that our approach reliably identifies OOD imaginations and synthesizes an uncertainty-aware latent safety filter from an offline dataset with limited coverage, enabling safe control in complex vision-based tasks by preemptively detecting safety risks and proposing safe, in-distribution backup actions.

Limitations

Component vs. System-level Safety Assurances. While our uncertainty-aware safety filter empirically can prevent both seen and unseen failures by incorporating OOD failures, it does not formally guarantee zero failure rates. In this work, we only provide a *component-level* statistical assurance on detecting OOD transitions within the world model via conformal prediction. Future work should study *system-level* assurances on the overall safety filter that is also influenced by our reinforcement-learning approximations in high-dimensional learned latent spaces. Moreover, our framework assumes that the system starts from an in-distribution safe initial state and that no unknown disturbances or visual distractions appear during operation. Therefore, for robust deployment, a system-level failure monitoring mechanism is necessary, which can reliably detect when the system loses its confidence. While our supplementary experiments indicate that our uncertainty measure can be leveraged for such system-level failure detection (see Sec. B.4), further exploration on system-level failure detection and mitigation remains as an important future work [5, 3].

Limited Generalizability and Reliability. Our latent safety filter relies on the capabilities of the learned world model. While recent generative world models have demonstrated promising results [64, 65], the world model's predictions can be imprecise even within in-distribution regions or fail to generalize to unseen scenarios. Although our safety filter adopts a minimally conservative approach to uncertain scenarios, its performance can be further improved with additional data. Future work should explore safe exploration strategies or active learning methods, using quantified epistemic uncertainty as intrinsic rewards to enhance world model generalization.

Challenges in Uncertainty Quantification. While our method adopts epistemic uncertainty quantification as a proxy for detecting unreliable world model imaginations, there are several limitations to this approach. Even within regions that are nominally in-distribution, world model predictions can still be imprecise or biased, particularly in complex or stochastic systems. In other words, while a transition may be classified as in-distribution, this does not guarantee the correctness of the model's prediction, potentially leading to an imprecise safety filter. Moreover, our uncertainty quantification assumes a Gaussian distribution over the next latent prediction, which may not hold in systems with complex, multimodal dynamics. It also adopts an ensemble as a separate module from the world model, which may not faithfully capture the model's true uncertainty (see Sec A.3 for further discussions). Exploring methods for faithfully detecting OOD scenarios under complex, multimodal data distributions presents an important direction for future work. Additionally, our framework and the safety Bellman equation (2) does not account for aleatoric uncertainty, and thus optimizes for the expected safety violation. Extending the framework to explicitly model aleatoric uncertainty in the latent dynamics could improve robustness, enabling latent-space safe control that better anticipates worst-case outcomes under the world model's predictions [66, 67, 68, 69, 47].

References

[1] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. In *International Conference on machine learning (ICML)*, pages 2555–2565, 2019.

[2] K. Nakamura, L. Peters, and A. Bajcsy. Generalizing safety beyond collision-avoidance via latent-space reachability analysis. *Robotics: Science and Systems (RSS)*, 2025.

[3] C. Agia, R. Sinha, J. Yang, Z.-a. Cao, R. Antonova, M. Pavone, and J. Bohg. Unpacking failure modes of generative policies: Runtime monitoring of consistency and progress. In *Conference on Robot Learning (CoRL)*, 2024.

[4] C. Xu, T. K. Nguyen, E. Dixon, C. Rodriguez, P. Miller, R. Lee, P. Shah, R. Ambrus, H. Nishimura, and M. Itkina. Can we detect failures without failure data? uncertainty-aware runtime failure detection for imitation learning policies. *Robotics: Science and Systems (RSS)*, 2025.

[5] R. Sinha, A. Sharma, S. Banerjee, T. Lew, R. Luo, S. M. Richards, Y. Sun, E. Schmerling, and M. Pavone. A system-level view on out-of-distribution data in robotics. *arXiv preprint arXiv:2212.14020*, 2022.

[6] M. Salehi, H. Mirzaei, D. Hendrycks, Y. Li, M. H. Rohban, and M. Sabokrou. A unified survey on anomaly, novelty, open-set, and out of-distribution detection: Solutions and future challenges. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

[7] J. Yang, K. Zhou, Y. Li, and Z. Liu. Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, 132(12):5635–5662, 2024.

[8] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin. A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50 (7):947–957, 2005.

[9] K. P. Wabersich, A. J. Taylor, J. J. Choi, K. Sreenath, C. J. Tomlin, A. D. Ames, and M. N. Zeilinger. Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems. *IEEE Control Systems Magazine*, 43(5):137–177, 2023.

[10] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, Apr. 2025. ISSN 1476-4687.

[11] R. Sinha, A. Elhafsi, C. Agia, M. Foutter, E. Schmerling, and M. Pavone. Real-time anomaly detection and reactive planning with large language models. In *Robotics: Science and Systems (RSS)*, 2024.

[12] J. Wong, A. Tung, A. Kurenkov, A. Mandlekar, L. Fei-Fei, S. Savarese, and R. Martín-Martín. Error-aware imitation learning from teleoperation data for mobile manipulation. In *Conference on Robot Learning (CoRL)*, pages 1367–1378, 2022.

[13] H. Liu, S. Dass, R. Martín-Martín, and Y. Zhu. Model-based runtime monitoring with interactive imitation learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4154–4161, 2024.

[14] C. Richter and N. Roy. Safe visual navigation via deep learning and novelty detection. In *Robotics: Science and Systems (RSS)*, 2017.

[15] L. Wellhausen, R. Ranftl, and M. Hutter. Safe robot navigation via multi-modal anomaly detection. *IEEE Robotics and Automation Letters*, 5(2):1326–1333, 2020.

[16] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K.-R. Müller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 109(5):756–795, 2021.

[17] R. Schmid, D. Atha, F. Schöller, S. Dey, S. Fakoorian, K. Otsu, B. Ridge, M. Bjelonic, L. Wellhausen, M. Hutter, et al. Self-supervised traversability prediction by learning to reconstruct safe terrain. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12419–12425, 2022.

[18] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios. Neural spline flows. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

[19] W. Liu, X. Wang, J. Owens, and Y. Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:21464–21475, 2020.

[20] K. Kang, P. Gradu, J. J. Choi, M. Janner, C. Tomlin, and S. Levine. Lyapunov density models: Constraining distribution shift in learning-based control. In *International Conference on Machine Learning (ICML)*, pages 10708–10733, 2022.

[21] A. Reichlin, G. L. Marchetti, H. Yin, A. Ghadirzadeh, and D. Kragic. Back to the manifold: Recovering from out-of-distribution states. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8660–8666. IEEE, 2022.

[22] F. Castaneda, H. Nishimura, R. T. McAllister, K. Sreenath, and A. Gaidon. In-distribution barrier functions: Self-supervised policy filters that avoid out-of-distribution states. In *Learning for Dynamics and Control Conference (L4DC)*, pages 286–299, 2023.

[23] R. Kidambi, A. Rajeswaran, P. Netrapalli, and T. Joachims. Morel: Model-based offline reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 33: 21810–21823, 2020.

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

[25] R. Rafailov, T. Yu, A. Rajeswaran, and C. Finn. Offline reinforcement learning from images with latent space models. In *Learning for Dynamics and Control Conference (L4DC)*, pages 1154–1168, 2021.

[26] T. Seyde, W. Schwarting, S. Karaman, and D. Rus. Learning to plan optimistically: Uncertainty-guided deep exploration via latent model ensembles. In *Conference on Robot Learning (CoRL)*, pages 1156–1167, 2022.

[27] K. Chua, R. Calandra, R. McAllister, and S. Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018.

[28] R. Sekar, O. Rybkin, K. Daniilidis, P. Abbeel, D. Hafner, and D. Pathak. Planning to explore via self-supervised world models. In *International Conference on machine learning (ICML)*, pages 8583–8592, 2020.

[29] R. Mendonca, O. Rybkin, K. Daniilidis, D. Hafner, and D. Pathak. Discovering and achieving goals via world models. *Advances in Neural Information Processing Systems (NeurIPS)*, 34: 24379–24391, 2021.

[30] P. Shyam, W. Jaśkowski, and F. Gomez. Model-based active exploration. In *International Conference on machine learning (ICML)*, pages 5779–5788, 2019.

[31] T. Kim, J. Mun, J. Seo, B. Kim, and S. Hong. Bridging active exploration and uncertainty-aware deployment using probabilistic ensemble neural network dynamics. In *Robotics: Science and Systems (RSS)*, 2023.

[32] M. Vlastelica, S. Blaes, C. Pinneri, and G. Martius. Risk-averse zero-order trajectory optimization. In *Conference on Robot Learning (CoRL)*, 2021.

[33] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2016.

[34] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Transactions on Automatic Control*, 64(7):2737–2752, 2018.

[35] K.-C. Hsu, H. Hu, and J. F. Fisac. The safety filter: A unified view of safety-critical control in autonomous systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 7, 2023.

[36] M. Ganai, S. Gao, and S. Herbert. Hamilton-jacobi reachability in reinforcement learning: A survey. *IEEE Open Journal of Control Systems*, 2024.

[37] S. Herbert, J. J. Choi, S. Sanjeev, M. Gibson, K. Sreenath, and C. J. Tomlin. Scalable learning of safety guarantees for autonomous systems using hamilton-jacobi reachability. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5914–5920, 2021.

[38] K.-C. Hsu, D. P. Nguyen, and J. F. Fisac. Isaacs: Iterative soft adversarial actor-critic for safety. In *Learning for Dynamics and Control Conference (L4DC)*, pages 90–103, 2023.

[39] H. Wang, J. Borquez, and S. Bansal. Providing safety assurances for systems with unknown dynamics. *IEEE Control Systems Letters*, 2024.

[40] Y. U. Ciftci, D. Chiu, Z. Feng, G. S. Sukhatme, and S. Bansal. Safe-gil: Safety guided imitation learning for robotic systems. *arXiv preprint arXiv:2404.05249*, 2024.

[41] S. Bansal and C. J. Tomlin. Deepreach: A deep learning approach to high-dimensional reachability. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1817–1824, 2021.

[42] J. F. Fisac, N. F. Lugovoy, V. Rubies-Royo, S. Ghosh, and C. J. Tomlin. Bridging hamilton-jacobi safety analysis and reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 8550–8556, 2019.

[43] K.-C. Hsu, V. Rubies-Royo, C. J. Tomlin, and J. F. Fisac. Safety and liveness guarantees through reach-avoid reinforcement learning. In *Robotics: Science and Systems (RSS)*, 2021.

[44] K. Chakraborty, A. Gupta, and S. Bansal. Enhancing safety and robustness of vision-based controllers via reachability analysis. *arXiv preprint arXiv:2410.21736*, 2024.

[45] K.-C. Hsu, A. Z. Ren, D. P. Nguyen, A. Majumdar, and J. F. Fisac. Sim-to-lab-to-real: Safe reinforcement learning with shielding and generalization guarantees. *Artificial Intelligence*, 314:103811, 2023.

[46] T. He, C. Zhang, W. Xiao, G. He, C. Liu, and G. Shi. Agile but safe: Learning collision-free high-speed legged locomotion. In *Robotics: Science and Systems (RSS)*, 2024.

[47] D. P. Nguyen, K.-C. Hsu, W. Yu, J. Tan, and J. F. Fisac. Gameplay filters: Robust zero-shot safety through adversarial imagination. In *Conference on Robot Learning (CoRL)*, 2024.

[48] A. Wilcox, A. Balakrishna, B. Thananjeyan, J. E. Gonzalez, and K. Goldberg. Ls3: Latent space safe sets for long-horizon visuomotor control of sparse reward iterative tasks. In *Conference on Robot Learning (CoRL)*, pages 959–969, 2022.

[49] A. Robey, H. Hu, L. Lindemann, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni. Learning control barrier functions from expert demonstrations. In *IEEE Conference on Decision and Control (CDC)*, pages 3717–3724, 2020.

[50] L. Lindemann, A. Robey, L. Jiang, S. Das, S. Tu, and N. Matni. Learning robust output control barrier functions from safe expert demonstrations. *IEEE Open Journal of Control Systems*, 2024.

[51] A. Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, volume 4, pages 547–562. University of California Press, 1961.

[52] F. Wang, T. Syeda-Mahmood, B. C. Vemuri, D. Beymer, and A. Rangarajan. Closed-form jensen-renyi divergence for mixture of gaussians and applications to group-wise shape registration. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2009: 12th International Conference, London, UK, September 20-24, 2009, Proceedings, Part I 12*, pages 648–655. Springer, 2009.

[53] V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic learning in a random world*, volume 29. Springer, 2005.

[54] A. N. Angelopoulos, S. Bates, et al. Conformal prediction: A gentle introduction. *Foundations and Trends® in Machine Learning*, 16(4):494–591, 2023.

[55] T. Ding, A. Angelopoulos, S. Bates, M. Jordan, and R. J. Tibshirani. Class-conditional conformal prediction with many classes. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:64555–64576, 2023.

[56] A. Z. Ren, A. Dixit, A. Bodrova, S. Singh, S. Tu, N. Brown, P. Xu, L. Takayama, F. Xia, J. Varley, Z. Xu, D. Sadigh, A. Zeng, and A. Majumdar. Robots that ask for help: Uncertainty alignment for large language model planners. In *Conference on Robot Learning (CoRL)*, 2023.

[57] I. M. Mitchell et al. A toolbox of level set methods. *UBC Department of Computer Science Technical Report TR-2007-11*, 1:6, 2007.

[58] Y. Sun, J. Zhang, C. Jia, H. Lin, J. Ye, and Y. Yu. Model-bellman inconsistency for model-based offline reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 33177–33194, 2023.

[59] H. Van Hasselt, A. Guez, and D. Silver. Deep reinforcement learning with double q-learning. In *AAAI Conference on Artificial Intelligence*, 2016.

[60] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State. Isaac gym: High performance gpu-based physics simulation for robot learning, 2021.

[61] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning (ICML)*, pages 1861–1870, 2018.

[62] A. Kumar, A. Zhou, G. Tucker, and S. Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:1179–1191, 2020.

[63] T. Yu, A. Kumar, R. Rafailov, A. Rajeswaran, S. Levine, and C. Finn. Combo: Conservative offline model-based policy optimization. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:28954–28967, 2021.

[64] G. Zhou, H. Pan, Y. LeCun, and L. Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *International Conference on machine learning (ICML)*, 2025.

[65] N. Agarwal, A. Ali, M. Bala, Y. Balaji, E. Barker, T. Cai, P. Chattopadhyay, Y. Chen, Y. Cui, Y. Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

[66] M. P. Chapman, R. Bonalli, K. M. Smith, I. Yang, M. Pavone, and C. J. Tomlin. Risk-sensitive safety analysis using conditional value-at-risk. *IEEE Transactions on Automatic Control*, 67 (12):6521–6536, 2021.

[67] D. Yu, W. Zou, Y. Yang, H. Ma, S. E. Li, Y. Yin, J. Chen, and J. Duan. Safe model-based reinforcement learning with an uncertainty-aware reachability certificate. *IEEE Transactions on Automation Science and Engineering*, 21(3):4129–4142, 2023.

[68] M. Ganai, Z. Gong, C. Yu, S. Herbert, and S. Gao. Iterative reachability estimation for safe reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 36: 69764–69797, 2023.

[69] M. Rigter, B. Lacerda, and N. Hawes. RAMBO-RL: Robust adversarial model-based offline reinforcement learning. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[70] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations (ICLR)*, 2020.

[71] D. Hafner, T. P. Lillicrap, M. Norouzi, and J. Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations (ICLR)*, 2021.

[72] T. Kim, R. I. Kee, and D. Panagou. Learning to refine input constrained control barrier functions via uncertainty-aware online parameter adaptation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[73] Y. As, B. Sukhija, L. Treven, C. Sferrazza, S. Coros, and A. Krause. Actsafe: Active exploration with safety constraints for reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2025.

[74] V. Micheli, E. Alonso, and F. Fleuret. Transformers are sample-efficient world models. In *International Conference on Learning Representations (ICLR)*, 2023.

[75] S. Levine, A. Kumar, G. Tucker, and J. Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[76] B. Thananjeyan, A. Balakrishna, S. Nair, M. Luo, K. Srinivasan, M. Hwang, J. E. Gonzalez, J. Ibarz, C. Finn, and K. Goldberg. Recovery rl: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 6(3):4915–4922, 2021.

[77] N. A. Urpí, S. Curi, and A. Krause. Risk-averse offline reinforcement learning. *International Conference on Learning Representations (ICLR)*, 2021.

[78] V. Kolev, R. Rafailov, K. Hatch, J. Wu, and C. Finn. Efficient imitation learning with conservative world models. In *Learning for Dynamics and Control Conference (L4DC)*, pages 1777–1790, 2024.

[79] Y. Jin, Z. Yang, and Z. Wang. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning (ICML)*, pages 5084–5096, 2021.

[80] C. Bai, L. Wang, Z. Yang, Z.-H. Deng, A. Garg, P. Liu, and Z. Wang. Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2022.

[81] J. Buckman, D. Hafner, G. Tucker, E. Brevdo, and H. Lee. Sample-efficient reinforcement learning with stochastic ensemble value expansion. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018.

[82] G. Shafer and V. Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

[83] V. Vovk. Conditional validity of inductive conformal predictors. In *Asian Conference on Machine Learning (ACML)*, pages 475–490, 2012.

[84] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Robotics: Science and Systems (RSS)*, 2024.

[85] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 2024.

Appendix

A Model Training

In this section, we present a brief overview of the latent world model used in our work. For a more comprehensive understanding, we refer readers to the original papers [1, 70, 71, 10].

A.1 Latent World Model.

Model Architecture We adopt the Dreamer [10] framework as our latent world model. Given sequences $\{o_t, a_t, l_t\}_{t=1}^T$, which consist of sensor observations o_t, action vectors a_t, and scalar failure margins $l_t \in \{-1, 1\}$, the model defines a generative process over observations and failure margins via a latent sequence $\{z_t\}_{t=1}^T$, under the partially observable Markov decision process (POMDP) formulation. Following the Dreamer architecture, the transition model is implemented as a Recurrent State-Space Model (RSSM) [1], which predicts future latent states using either Gaussian or categorical distributions parameterized by feed-forward neural networks. The transition model outputs the prior latent state $\hat{z}_t$ conditioned on the previous latent and action. The encoder then combines $\hat{z}_t$ and the current observation o_t to produce the posterior latent z_t:

$$\text{Encoder: } z_t \sim \mathcal{E}(z_t \mid \hat{z}_t, o_t) \quad \text{Transition: } \hat{z}_t \sim f_z(\hat{z}_t \mid z_{t-1}, a_{t-1}) \quad \text{Failure: } l_t \sim \ell_z(l_t \mid z_t).$$

Note that during imagination rollouts, only the prior latents are used, as observations are unavailable. The RSSM uses a Gated Recurrent Unit (GRU) to compute deterministic recurrent features, which are concatenated with samples from the stochastic state to form the full latent z_t. Observations are decoded from the latent state using either a deconvolutional network or a multilayer perceptron (MLP), and modeled with a Gaussian likelihood. The failure classifier is trained to predict a Bernoulli likelihood over failure margins.

Failure Margin Function We further assume that this offline dataset is annotated with failure labels at each timestep, indicating whether the ground-truth, but unknown, state has violated safety. These labels are used to train a margin function $\ell_z(z_t)$ that implicitly defines a failure set in the latent space $\mathcal{F} = \{z \mid \ell_z(z) < 0\}$.

Loss Function Due to the model's nonlinearity, the true posterior over latent states required for learning cannot be computed analytically. Instead, RSSM adopts a mean-field approximation that extends the framework to partially observable Markov decision processes (POMDPs). Specifically, it factorizes the variational distribution $q(z_{1:T} \mid o_{1:T}, a_{1:T})$ as a product of encoder and latent dynamics terms:

$$q(z_{1:T} \mid o_{1:T}, a_{1:T}) = \prod_{t=1}^T q(z_t \mid z_{t-1}, o_t, a_t),$$

which infers an approximate posterior using past observations and actions. A variational lower bound on the data log-likelihood can then be derived using Jensen's inequality:

$$\ln p(o_{1:T}, l_{1:T} \mid a_{1:T}) \triangleq \ln \mathbb{E}_{p(z_{1:T}\mid a_{1:T})} \left[\prod_{t=1}^T p(o_t, l_t \mid z_t) \right] \tag{9}$$

$$\geq \mathbb{E}_{q(z_{1:T}\mid o_{1:T}, a_{1:T})} \left[\sum_{t=1}^T \ln p(o_t \mid z_t) + \ln p(l_t \mid z_t) + \ln p(z_t \mid z_{t-1}, a_{t-1}) - \ln q(z_t \mid o_{\leq t}, a_{<t}) \right] \tag{10}$$

$$= \mathbb{E}_q \left[\sum_{t=1}^T \underbrace{\ln p(o_t \mid z_t)}_{\text{reconstruction loss}} + \underbrace{\ln p(l_t \mid z_t)}_{\text{failure margin loss}} - \underbrace{\text{KL}\left[q(z_t \mid o_{\leq t}, a_{<t}) \,\|\, p(z_t \mid z_{t-1}, a_{t-1})\right]}_{\text{KL loss}} \right] . \tag{11}$$

All components of the world model are optimized jointly. The encoder and failure margin function are trained to maximize the log-likelihood of their respective targets, while the dynamics model is optimized to produce latent states that facilitate these prediction tasks.

Implementation Details   We build on the open-source implementation of DreamerV3[1]. For the Dubin's Car experiments, we use a continuous stochastic latent space modeled as a 32-dimensional Gaussian. In contrast, for high-dimensional visual manipulation tasks, we adopt a discrete latent representation composed of 32 categorical variables, each with 32 classes, resulting in a 1024-dimensional stochastic latent space. Actions are represented using delta pose control—a 6-dimensional vector corresponding to normalized changes in the end-effector pose—along with an additional dimension for the gripper action. The relevant hyperparameters for Dubin's Car and visual manipulation experiments are listed in Table 3 and Table 4, respectively.

HYPERPARAMETER	VALUE
IMAGE DIMENSION	[128, 128, 3]
ACTION DIMENSION	3 (Discrete)
STOCHASTIC LATENT	Gaussian
LATENT DIM (DETERMINISTIC)	512
LATENT DIM (STOCHASTIC)	32
ACTIVATION FUNCTION	SiLU
ENCODER CNN DEPTH	32
ENCODER MLP LAYERS	5
FAILURE CLASSIFIER LAYERS	2
BATCH SIZE	16
BATCH LENGTH	32
OPTIMIZER	Adam
LEARNING RATE	1e-4
ITERATIONS	100000

Table 3: Dubin's Car Hyperparameters

HYPERPARAMETER	VALUE
IMAGE DIMENSION (SIMULATION)	[2, 128, 128, 3]
IMAGE DIMENSION (REAL-WORLD)	[2, 256, 256, 3]
PROPRIOCEPTION DIMENSION	7
ACTION DIMENSION	7 (Continuous)
STOCHASTIC LATENT	Categorical
LATENT DIM (DETERMINISTIC)	512
LATENT DIM (STOCHASTIC)	32×32
ACTIVATION FUNCTION	SiLU
ENCODER CNN DEPTH	32
ENCODER MLP LAYERS	5
FAILURE CLASSIFIER LAYERS	2
BATCH SIZE	16
BATCH LENGTH	64
OPTIMIZER	Adam
LEARNING RATE	1e-4
ITERATIONS	200000

Table 4: Visual Manipulation Hyperparameters

A.2   Probabilistic Ensemble Latent Predictor.

LAYER	INPUT DIM	OUTPUT DIM	NORMALIZATION
Linear	d_{in}	d_{in}	LayerNorm
Linear	d_{in}	$2d_{\text{in}}$	LayerNorm
Linear	$2d_{\text{in}}$	$3d_{\text{in}}$	LayerNorm
Linear	$3d_{\text{in}}$	d_{in}	LayerNorm
Linear	d_{in}	$2d_{\text{out}}$	None

Table 5: Latent Predictor Architecture

Model Architecture   We build upon the open-source implementation provided by [72][2]. Each ensemble member is initialized independently with random weights, resulting in diverse initializations across the ensemble. The architecture of each ensemble member in the latent predictor is summarized in Table 5. Each member consists of five fully connected layers, initialized independently at random, and outputs a $2d_{\text{out}}$-dimensional vector corresponding to the mean and variance of a diagonal Gaussian. The input dimension d_{in} corresponds to the latent state, which is $512 + 32 = 544$ for continuous Gaussian latents and $512 + 32 \times 32 = 1536$ for discrete categorical latents. All ensemble members are trained independently and implemented using `torch.baddbmm` for efficient batch matrix multiplication during inference. The size of the ensemble is presented in Table 6.

Training   The ensemble latent predictor is trained with a frozen, pre-trained latent dynamics model, with the same dataset $\mathcal{D}_{\text{train}}$. Given a set of latent transitions $\left\{\{(z_t, a_t, z_{t+1})\}_{t=1}^{T-1}\right\}$ obtained from the offline dataset using the learned latent model, each ensemble member is trained by

---

[1] https://github.com/NM512/dreamerv3-torch
[2] https://github.com/tkkim-robot/online_adaptive_cbf

	DUBIN'S CAR	BLOCK PLUCKING	JENGA
ENSEMBLE SIZE K	10	5	5

Table 6: Number of ensemble members in experiments.

minimizing the Gaussian negative log-likelihood (NLL):

$$\mathcal{L}_{\text{train}}(\psi_k) = \sum_{t=1}^{T-1} \left[\mu_{\psi_k} - z_{t+1} \right]^\top \Sigma_{\psi_k}^{-1} \left[\mu_{\psi_k} - z_{t+1} \right] + \log \det \Sigma_{\psi_k}. \tag{12}$$

A Brief Background on Jensen-Rényi Divergence To quantify epistemic uncertainty from ensemble disagreement, it is essential to distinguish it from aleatoric uncertainty. Without this separation, it becomes unclear whether the uncertainty is from a lack of model knowledge or from inherent, irreducible system stochasticity, such as model ambiguity or sensor noise. A common approach for measuring disagreement between predictive distributions is the Kullback–Leibler (KL) divergence. However, KL divergence is asymmetric and limited to pairwise comparisons, making it unsuitable for capturing ensemble-wide disagreement.

The Jensen-Rényi divergence (JRD) extends the well-known Jensen-Shannon divergence with Rényi entropy $H_\alpha(Z)$ of a random variable Z:

$$\text{JRD}\left(\hat{f}_{1:K}\right) \triangleq H_\alpha\left(\sum_{k=1}^{K} \frac{1}{K} \hat{f}_k\right) - \sum_{k=1}^{K} \frac{1}{K} H_\alpha\left(\hat{f}_k\right), \quad H_\alpha(Z) = \frac{1}{1-\alpha} \log \int p(z)^\alpha \mathrm{d}x. \tag{13}$$

However, computing JRD is intractable as it involves estimating the entropy of a mixture of Gaussians, which has no analytical solution. Although JRD can be estimated via Monte Carlo sampling, such approximations are computationally expensive and impractical for real-time applications. To address this, Wang et al.[52] introduced a closed-form JRD formulation based on quadratic Rényi entropy ($\alpha = 2$) [52], enabling efficient and analytic computation of the divergence among Gaussian mixture models (GMMs):

$$\text{JRD}\left(\hat{f}_{1:K}\right) = -\log\left[\frac{1}{K^2} \sum_{i,j}^{K} \mathfrak{D}\left(\hat{f}_i, \hat{f}_j\right)\right] + \frac{1}{K} \sum_{i}^{K} \log\left[\mathfrak{D}\left(\hat{f}_i, \hat{f}_i\right)\right], \quad \text{where}$$

$$\mathfrak{D}\left(\hat{f}_i, \hat{f}_j\right) = \frac{1}{|\Phi|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\Delta^\top \Phi^{-1} \Delta\right) \quad \text{with} \quad \Phi = \Sigma_{\phi_i} + \Sigma_{\phi_j} \quad \text{and} \quad \Delta = \mu_{\phi_i} - \mu_{\phi_j}. \tag{14}$$

We refer readers to [52] for a detailed explanation of closed-form JRD, and its practical applications in learning-based settings [30, 31].

A.3 Why not ensemble the latent dynamics model itself?

While prior works often ensemble the dynamics model directly to estimate epistemic uncertainty [27, 30, 31], we instead introduce a separate ensemble of latent predictors as a proxy for uncertainty estimation of the learned latent world model. Although a detailed analysis of this design choice is beyond the scope of our contributions, we briefly address several challenges of ensembling the latent dynamics model and give justification for our design choice.

Practical Challenges of Ensembling RSSM Since the latent world model jointly optimizes both the latent representation and the transition dynamics, ensembling this model would require optimizing over a non-stationary and noisy target—the next latent state z_{t+1}—which can lead to training instability. Furthermore, because the latent dynamics are trained via distribution matching rather than a direct regression objective, it becomes intractable to apply standard ensemble training techniques. Although recent works [25, 26, 73] attempt to circumvent this issue by randomly sampling

a single dynamics member during training, we empirically find that this approach results in weaker representation learning and less reliable uncertainty estimation compared to our method (See D.1). Additionally, the stochastic latent variables in the world model are optimized for sampling within the latent space to enable next-state prediction, rather than for explicitly modeling aleatoric uncertainty. We empirically find that the predicted variance or distribution produced by RSSM does not faithfully capture the calibrated aleatoric uncertainty inherent in the dynamics, which is critical for accurate uncertainty quantification and for distinguishing aleatoric from epistemic uncertainty.

Separate Ensemble Module Enables Efficient Uncertainty Quantification Lastly, our design choice enables efficient epistemic uncertainty quantification for the latent world model. As the latent world model tends to be significantly larger than typical state-based dynamics models, employing a lightweight, separate module offers a practical and scalable way to capture uncertainty. This is especially important as recent generative world models, such as [74, 64], continue to grow in size, making it infeasible to ensemble the model itself. Note that our method is capable of forwarding both the latent representation and its associated uncertainty in under 0.1 seconds on a standard desktop setup, using approximately $2 - 4$GB of VRAM for the ensemble.

B Latent-Space Reachability Anaylsis

B.1 A Brief Background on HJ Reachability

Hamilton-Jacobi (HJ) reachability is a control-theoretic framework for safety analysis that identifies when current actions may lead to future failures and computes best-effort policies to mitigate such outcomes [8, 35]. Given a dynamical system with state $s \in \mathcal{S}$, action $a \in \mathcal{A}$, and dynamics $s_{t+1} = f(s_t, a_t)$, HJ reachability seeks to determine the safe set that can prevent the system from entering a designated *failure set* $\mathcal{F} = \{s \mid \ell(s) < 0\}$, which is represented by a margin function $\ell : \mathcal{S} \to \mathbb{R}$. The framework aims to find the *unsafe set*, denoted $\mathcal{U} \subset \mathcal{S}$, which includes all states from which the system is inevitably driven into $\mathcal{F}$ despite the best effort, and the best effort safety-preserving policy to avoid entering the unsafe set.

The framework jointly computes (i) a safety value function $V^{\mathbf{0}} : \mathcal{S} \to \mathbb{R}$, which quantifies the minimal safety margin the system can achieve from a given state s under optimal behavior, and (ii) a best-effort safety-preserving policy $\pi^{\mathbf{0}} : \mathcal{S} \to \mathcal{A}$. These are obtained by solving an optimal control problem governed by the following fixed-point safety Bellman equation:

$$V(s) = \min \left\{ \ell(s), \max_{a \in \mathcal{A}} V(f(s, a)) \right\}, \quad \pi^{\mathbf{0}}(s) := \arg\max_{a \in \mathcal{A}} V(f(s, a)). \quad (15)$$

To tractably approximate solutions to high-dimensional reachability problems, Fisac et al. [42] propose using reinforcement learning by replacing the standard Bellman equation for cumulative reward with a time-discounted counterpart of Eq. 15:

$$V(s_t) = (1 - \gamma)\ell(s) + \gamma \min \left\{ \ell_\theta(s), \max_{a \in \mathcal{A}} V\big(f(s, a)\big) \right\}, \quad (16)$$

where γ is the discount factor that ensures contraction of the Bellman operator. The resulting *unsafe set*, denoted $\mathcal{U} \subset \mathcal{S}$, captures all states from which the system can no longer avoid entering $\mathcal{F}$, and is defined as the zero sublevel set of the value function: $\mathcal{U} := \{s \mid V(s) < 0\}$. At deployment time, the safety value function and safety policy enable *safety filtering*: detecting unsafe actions proposed by any task policy π^{task} and minimally adjusting them only when necessary to ensure the system remains within the safe set. We refer readers to survey papers for further details [35, 9].

B.2 Implementation Details.

To approximate the safety filter via reinforcement learning, we adopt training strategies from model-based reinforcement learning, enabling reachability analysis entirely through latent imagination. Using the learned dynamics model, we initialize rollouts by encoding randomly sampled data from

the offline training dataset into the latent space. Starting from these initial latent states, imagined rollouts are used as a simulated environment for off-policy reinforcement learning. Note that during imagination rollouts, only the prior latents are used, as observations are unavailable

Discrete Action Space   For the Dubins Car experiments, which operate in a discrete action space, we use Double DQN (DDQN) [59] to train the safety value function. The Q-function is implemented as a 3-layer multilayer perceptron (MLP) with a hidden dimension of 100, producing Q-values for each of the 3 discrete actions. The associated hyperparameters are summarized in Table. 7.

HYPERPARAMETER	VALUE
ARCHITECTURE	[100, 100]
LEARNING RATE	1e-3
OPTIMIZER	AdamW
DISCOUNT FACTOR γ	0.9999
NUM ITERATIONS	50000
MEMORY BUFFER SIZE	20000
BATCH SIZE	256
MAX IMAGINATION STEPS	20

Table 7: DDQN Hyperparameters

HYPERPARAMETER	VALUE
ACTOR ARCHITECTURE	[512, 512, 512, 512]
CRITIC ARCHITECTURE	[512, 512, 512, 512]
NORMALIZATION	LayerNorm
ACTIVATION	ReLU
DISCOUNT FACTOR γ	$0.85 \rightarrow 0.9999$
LEARNING RATE (CRITIC)	1e-4
LEARNING RATE (ACTOR)	1e-4
OPTIMIZER	AdamW
NUMBER OF ITERATIONS	200000
REPLAY BUFFER SIZE	500000
BATCH SIZE	512
MAX IMAGINATION STEPS	30

Table 8: SAC hyperparameters.

Importantly, in the Dubins car task, the vehicle is tasked with avoiding a target located at the center, and its trajectory may extend beyond the bounding box, which is safe, while regions outside the box are highly uncertain and out-of-distribution. To avoid overly conservative behavior that penalizes successful avoidance, we track the true state of the vehicle and omit the OOD penalty for states outside the bounding box, defined as those with $|p_x| > 1$ or $|p_y| > 1$.

Continuous Action Space   For continuous control in both simulated and real-world visual manipulation tasks, we adopt Soft Actor-Critic (SAC) [61] within an off-policy, model-based reinforcement learning framework. We model the safety value function as a latent-action value function $Q(z, a)$ that is conditioned on the action. The safety policy is parameterized by an actor-network $a \sim \pi^{\Phi}(\cdot \mid z)$. The safety value can be evaluated by $V^{\Phi}(z) = \max_a Q(z, a) = Q(z, \pi^{\Phi}(z))$.

At each time step, we store imagined transitions $(\tilde{z}, a, l, \tilde{z}')$ in the replay buffer $\mathcal{B}$, where l is given by the safety margin function $\ell_{\tilde{z}}(\tilde{z}, u)$ as defined in Eq. 4. We then optimize the critic using the following objectives:

$$\mathcal{L}_{\text{critic}} := \mathbb{E}_{(\tilde{z}, a, r, \tilde{z}',) \sim \mathcal{B}} \left[(Q(z, a) - y)^2 \right], \quad y = (1 - \gamma) r + \gamma \min\{r, \max_{a'} Q(z', a')\}. \quad (17)$$

Note that we do not parametrize the uncertainty variable u directly in the value function but intrinsically leverage it with the safety margin function. To stabilize training, we maintain two Q-functions and use target networks for the temporal difference updates. The policy is optimized following the policy gradient induced by the critic and entropy loss term:

$$\mathcal{L}_{\text{actor}} := \mathbb{E}_{z \sim \mathcal{B}} \left[-Q(z, a) + \beta \log \pi^{\Phi}(a \mid z) \right], \quad a \sim \pi^{\Phi}(\cdot \mid z), \quad (18)$$

where γ is scheduled from 0.85 to 0.9999 and β is the hyperparameter for exploration. These updates enable the actor to select actions that maximize expected safety margins while the critic estimates the corresponding safety values under uncertainty. The hyperparameters for training SAC are summarized in Table. 8.

B.3 Uncertainty-aware Safety Filter.

To implement the safety filter in the uncertainty-aware latent space described in Eq. 4, we leverage the trained action-conditioned safety value function Q. At each time step, we evaluate whether executing the action proposed by the task policy π^{task} leads to an unsafe outcome, as determined by reachability analysis, and, if so, override it using the safety policy $\pi^{\mathbf{0}}$.

In practice, the filtering condition $V^{\mathbf{0}}(\tilde{z}') \leq \delta$, which evaluates on the predicted next latent $\tilde{z}' \sim f_{\tilde{z}}(\tilde{z}, \pi^{\text{task}})$, is assessed by monitoring two criteria: (i) if the epistemic uncertainty exceeds the threshold (i.e., $\mathrm{D}(z, \pi^{\text{task}}) > \epsilon$); or (ii) the safety value is low (i.e. $Q(z', \pi^{\mathbf{0}}(z')) \leq \delta$). The union of these conditions indicates that the safety value of the next state is below the safe margin. Recall the definition of the uncertainty-aware safety margin function (4) and the Bellman update (2):

$$\ell_{\tilde{z}}(\tilde{z}_t) = \min\left\{\ell_z(z_t), \kappa(\epsilon - u_t)\right\}, \quad V^{\mathbf{0}}(z, u) = \min\left\{\ell_{\tilde{z}}(z, u), \max_a V^{\mathbf{0}}(z', u')\right\},$$

where z', u' is sampled from $f_{\tilde{z}}(z, a)$. When the disagreement is above the threshold, the uncertainty triggers the OOD penalty, which yields $\ell_{\tilde{z}}(\tilde{z}) = -\kappa < 0$, ensuring that the value function is below the safety filter margin $\delta > 0$. Alternatively, if the value at the next latent state is small, i.e., $Q(z', \pi(z')) \leq \delta$, then the second term in the Bellman update becomes small, also driving $V^{\mathbf{0}}(\tilde{z}') \leq \delta$. Hence, either condition implies that the next transition is unsafe under the safety filter:

$$\mathrm{D}(z, \pi^{\text{task}}) > \epsilon \quad \text{or} \quad Q(z', \pi(z')) \leq \delta \quad \Longleftrightarrow \quad V^{\mathbf{0}}(\tilde{z}') \leq \delta.$$

B.4 Failure Detection of the Uncertainty-aware Safety Filter

Does our safety filter always guarantee safety? Our framework assumes that the robot starts from an in-distribution initial state and maintains approximate control invariance with respect to an estimated safe set in the latent space. However, since the safety filter is trained via reinforcement learning and relies on an imperfect latent dynamics model, safety cannot be guaranteed in all cases. The reliability of the learned filter can degrade in several situations—for instance, when the system begins in an out-of-distribution state (e.g., due to an OOD visual input at test time) or when the filter fails to prevent transitions into unsafe regions. In such cases, the safety filter may behave unpredictably, executing random or overconfident actions or even exacerbating unsafe situations. To ensure safe deployment, it is essential to detect when the safety filter becomes unreliable. In such cases, the system should halt and request human intervention. Without this safeguard, the robot may continue operating despite its internal safety mechanism failing.

System-level Failure Detection Failures of learned safety filters can arise from a range of sources, including OOD sensory inputs, misspecified dynamics models, or inaccurately learned safety value functions. A reliable safety filter should exhibit consistent behavior under bounded epistemic uncertainty. To detect violations of this principle, we monitor whether the backup action $\pi^{\mathbf{0}}(z)$ leads to a transition with sufficiently low predictive uncertainty. If it does not, we assume the system has entered the OOD failure set and must stop operation.

Based on the safety filtering rule in Eq. 3, the selected action is expected to avoid transitions that induce high predictive uncertainty. Formally, the safety filter should satisfy: $\mathrm{D}(z, a^{\text{exec}})) \leq \epsilon$. Conversely, if the filtered action itself leads to excessive epistemic uncertainty, we consider the system to have entered the unsafe set, which the robot cannot automatically recover from. In this case, the safety guarantees provided by the filter no longer hold, and the system should halt operation. In particular, if even the fallback action $\pi^{\mathbf{0}}(z)$ results in high disagreement, the system is deemed unrecoverable under the current safety filter: $\mathrm{D}(z, \pi^{\mathbf{0}}(z)) > \epsilon$. This motivates a modification to the filtering rule, introducing an explicit halting condition when the filter is unable to guarantee a safe and confident action. With predicted next latent state $\tilde{z}' \sim f_{\tilde{z}}(\tilde{z}, \pi^{\text{task}})$ the filter is constructed as:

$$\phi\left(\tilde{z}, \pi^{\text{task}}\right) := \begin{cases} \pi^{\text{task}}, & \text{if } V^{\mathbf{0}}(\tilde{z}') > \delta, \\ \pi^{\mathbf{0}}(\tilde{z}), & \text{if } V^{\mathbf{0}}(\tilde{z}') \leq \delta \text{ and } \mathrm{D}(z, \pi^{\mathbf{0}}(z)) \leq \epsilon, \\ \text{HALT}, & \text{otherwise.} \end{cases} \tag{19}$$
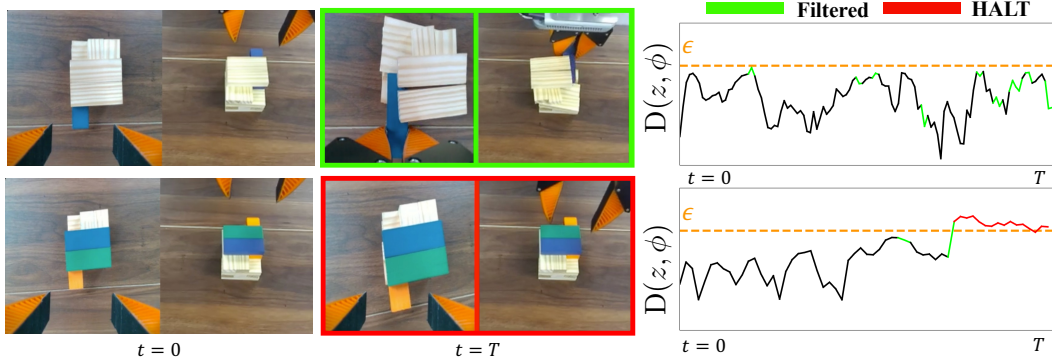
Figure 8: *Top row*: despite a color change in the target block, the latent dynamics model remains reliable, maintaining predictive uncertainty below the threshold. *Bottom row*: in contrast, when the visual input deviates significantly from the training distribution, the model becomes unreliable. The safety filter fails to maintain predictive uncertainty below the threshold, prompting the system to halt in order to avoid actions that could compromise or aggravate safety.

Results: OOD Visual inputs Fig. 8 illustrates the outcome of failure detection by the safety filter in the Jenga task. In this scenario, a teleoperator attempts to grasp a block and executes an unsafe action—pushing the block to the right. The learned safety filter intervenes to suppress this unsafe behavior. Although the block colors differ from those encountered during training, such visual changes do not inherently indicate out-of-distribution inputs. Instead, the decision to halt is governed by the reliability of the filtering system. When the color of the target block changes but remains within the model's generalization capacity, the latent dynamics model remains reliable, maintaining predictive uncertainty below the threshold. In contrast, when the visual input deviates substantially from the training distribution, the model becomes unreliable. The safety filter then fails to keep uncertainty within acceptable bounds, prompting the system to halt in order to prevent potentially dangerous actions.
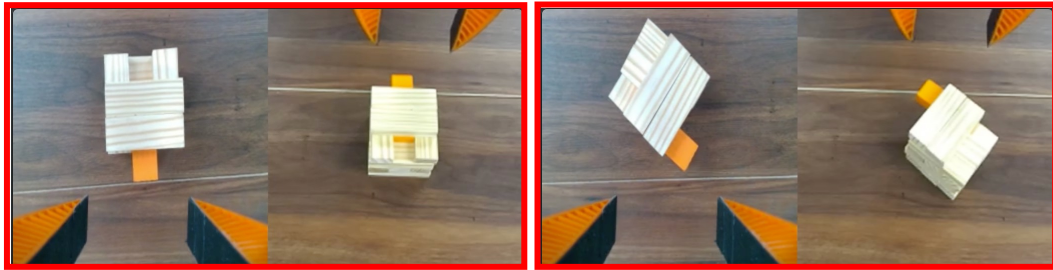

Figure 9: OOD settings that lead the system to halt.

Fig. 9 shows additional scenarios where the system safely halts upon detecting unrecoverable conditions due to OOD inputs that differ significantly from the training data.

B.5 Brief Backgrounds on Offline Reinforcement Learning

Offline reinforcement learning (RL) learns policies from a static dataset of past interactions, making it well-suited for applications where online exploration poses safety risks [75, 76, 77, 78]. A major challenge in offline RL is the distribution shift between the learned policy and the behavior policy that collected the data [79, 58], which often leads to overestimation of policy evaluations on OOD scenarios [80]. To address this, conservatism is introduced by penalizing value functions, preventing over-optimism on OOD actions [62, 63, 69]. In offline model-based RL (MBRL), a dynamics model is learned from the static dataset and used to generate synthetic data for policy learning [27, 81, 24, 23, 63, 58]. By quantifying the uncertainty of the learned dynamics model, these methods

mitigate model exploitation and discourage the system from entering OOD scenarios. Inspired by this, our work quantifies uncertainty in a learned latent dynamics model and ensures a safety filter to proactively prevent the system from entering OOD regions.

C Conformal Prediction for Calibrating OOD Threshold

In this section, we briefly introduce conformal prediction and outline the details of our calibration procedure. For a comprehensive overview, we refer readers to [53, 54]. Conformal prediction is a statistically principled framework for constructing prediction sets or regions with guaranteed coverage, relying only on mild assumptions such as data exchangeability or i.i.d. sampling. Given a user-specified significance level α, it guarantees that the true target lies within the constructed prediction region with probability at least $1 - \alpha$.

Let $\{P_1, P_2, \ldots, P_N\}$ be a set of N i.i.d. nonconformity scores. The goal is to compute a threshold C such that a new test sample is included in the prediction region with high probability. Conformal prediction provides the following guarantee:

$$\mathbb{P}(P_{\text{test}} \leq C) \geq 1 - \alpha.$$

The threshold C is typically chosen as the empirical $(1 - \alpha)$-quantile of the calibration scores. This is computed by sorting the set $\{P_1, \ldots, P_N\}$ and selecting the $\lceil (1-\alpha)(N+1) \rceil$-th smallest value, where $\lceil \cdot \rceil$ denotes the ceiling function.

C.1 Trajectory-level Conformal Prediction

Eq. 8 provides a probabilistic guarantee on the recall of in-distribution transitions—or, equivalently, a bound on the false positive rate. Rewriting the equation:

$$\mathbb{P}\left(Q_{\tau_{\text{test}}}^{\alpha_{\text{trans}}} \leq \epsilon \mid \tau_{\text{test}} \in \mathcal{D}_{\text{WM}} \right) \geq 1 - \alpha_{\text{cal}}, \tag{20}$$

this implies that the quantile of ensemble disagreement within a test trajectory is less than or equal to the threshold ϵ with probability at least $1 - \alpha_{\text{cal}}$. In other words, for an in-distribution test trajectory, at least a fraction $1 - \alpha_{\text{trans}}$ of its transitions are expected to have disagreement scores below ϵ with high confidence. Note that the maximum is a special case of the quantile function with $\alpha_{\text{trans}} = 0$:

$$\mathbb{P}\left(\max_t \mathrm{D}(z_t^{\text{test}}, a_t^{\text{test}}) \leq \epsilon \mid (z_t^{\text{test}}, a_t^{\text{test}}) \in \mathcal{D}_{\text{WM}} \right) \geq 1 - \alpha_{\text{cal}}, \tag{21}$$

where all transitions within a trajectory must fall below the threshold. Similar to [56], this trajectory-level in-distribution prediction set $\mathcal{C}(\tau_{\text{test}}) = \left\{ \tau_i : Q_{\tau_i}^{\alpha_{\text{trans}}} \leq \epsilon \right\}$, enables causal reconstruction of a transition-level in-distribution set $\mathcal{C}(z_{\text{test}}, a_{\text{test}}) = \left\{ (z_t, a_t) : \mathrm{D}(z_t, a_t) \leq \epsilon \right\}$, since:

$$\max_t \mathrm{D}(z_t^{\text{test}}, a_t^{\text{test}}) \leq \epsilon \iff \mathrm{D}(z_t^{\text{test}}, a_t^{\text{test}}) \leq \epsilon \quad \forall t \in [T]. \tag{22}$$

However, this strict formulation tends to produce overly optimistic thresholds in practice, resulting in a high false negative rate—that is, misclassifying OOD transitions as in-distribution. This is largely due to noise and imperfection in the ensemble-based disagreement estimates. To mitigate this, we adopt a quantile-based nonconformity score, which allows for a small, controlled level of transition-level misclassification $(1 - \alpha_{\text{trans}})$ within each trajectory.

C.2 Dataset-conditional Guarantee.

Equations 8 and 20 hold marginally, with the probability taken over both the sampling of the test data and the calibration data [44]. However, by fixing the calibration dataset, which is drawn i.i.d. from $\mathcal{D}_{\text{WM}}$, we obtain a dataset-conditional guarantee [82]. Specifically, conditioned on a calibration dataset $\mathcal{D}_{\text{cal}} \subset \mathcal{D}_{\text{WM}}$, the coverage achieved by conformal prediction follows a Beta distribution [83]:

$$\mathbb{P}\left(Q_{\tau_{\text{test}}}^{\alpha_{\text{trans}}} \leq \epsilon \mid \tau_{\text{test}} \in \mathcal{D}_{\text{WM}} \right) \sim \text{Beta}(N + 1 - C, C), \quad \text{where} \quad C := \lfloor (N+1)\alpha_{\text{trans}} \rfloor. \tag{23}$$

C.3   Implementation Details.

For each task, we collect a calibration dataset to determine the OOD threshold based on ensemble disagreement. This calibration dataset is a held-out subset collected alongside the training data, but it is not used during model training. Table 9 summarizes the calibration dataset sizes and the conformal prediction hyperparameters used for each task.

TASK	CALIBRATION SET SIZE (N)	α_{cal}	α_{trans}
DUBIN'S CAR	500	0.05	0.05
BLOCK PLUCKING	100	0.05	0.05
JENGA	30	0.10	0.10

Table 9: Conformal Prediction Parameters for Each Task
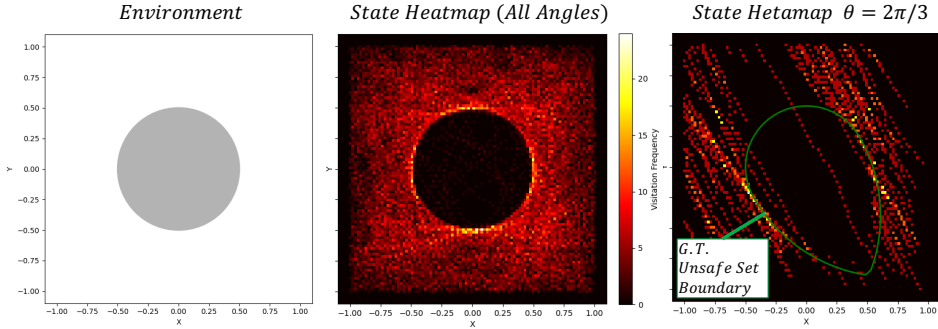
D   Experiment Details.

D.1   Dubins Car

Figure 10: Visualization of dataset consisting of expert trajectories and a few random trajectories. *Left*: Environment. The gray circle at the center denotes the failure set, which expert trajectories consistently avoid. *Middle*: State visitation heatmap based on (x, y) positions, showing that most data is concentrated in safe regions. *Right*: Heatmap of states with a specific heading angle. Most of the data lies outside the ground-truth unsafe set, resulting in high model uncertainty around the true failure region.

Expert Trajectories   The expert trajectories never enter the failure set. Importantly, this does not mean they simply stop at the boundary; rather, the vehicle also avoids entering the ground-truth unsafe set—states from which failure is inevitable despite being currently safe. These trajectories are generated using the ground-truth safety value function, computed via a grid-based solver [57]. At the boundary of the unsafe set, the vehicle executes only safe actions, whereas outside this region, it performs random actions. As shown in Fig. 10, the dataset exhibits low density within the unsafe region, resulting in high epistemic uncertainty in those areas and leading to overly optimistic world model imagination near critical decision boundaries.

Evaluation   Since the ground-truth dynamics are known, we can compute the exact safety value function using traditional grid-based methods [57]. This enables us to evaluate the accuracy of the safety filter's monitor by comparing its safe/unsafe classifications against the ground truth. We compute the value functions over all three dimensions for both safety monitoring and policy evaluation. To assess the effectiveness of the learned safety policies, we evaluate whether they can successfully steer the system away from failure. Feasible initial conditions are identified using the ground-truth state-based value function, yielding approximately 10,000 candidates. To highlight challenging scenarios, we additionally report results on a curated subset of 181 cases where the system starts in a

safe region but is oriented toward the failure set. Table 10 presents a quantitative comparison between two settings: one using an uncertainty-aware latent space that incorporates OOD failures, and one using latent dynamics without explicit uncertainty modeling.

	FPR↓	Pre.↑	B.Acc.↑	Safe (Total) ↑	Safe (Challenging) ↑
LatentSafe (w.o. $\mathcal{F}_{\text{OOD}}$)	0.30	0.92	0.84	0.98	0.63
UNISafe (w. $\mathcal{F}_{\text{OOD}}$)	0.05	0.98	0.94	0.97	0.82

Table 10: Safety filter performance on the Dubins Car experiment with expert trajectories.

Dubins Car without Failure Trajectories To further validate the reliability of uncertainty quantification, we quantitatively compare several UQ methods applicable to the latent dynamics model in the Dubins Car setting with only OOD failures (see Sec. 6.1). Using the same offline dataset, we train latent dynamics models with different UQ methods and synthesize corresponding safety filters. For each method, the threshold is calibrated using the same held-out calibration dataset. We then evaluate the learned safety value functions against the ground-truth safety value function exactly computed with grid-based methods. Additionally, we perform closed-loop rollouts from random initial states sampled across all three dimensions of the Dubins Car state space, measuring the resulting safety rates of full trajectories. Results are summarized in Table 11.

Method	FPR↓	Recall ↑	Pre.↑	F_1 ↑	B.Acc.↑	Safe Rate↑
TotalUncertainty	0.028	0.888	0.965	0.926	0.930	0.938
EnsembleRSSM	0.024	0.846	0.969	0.904	0.911	0.925
MaxAleatoric	0.119	0.785	0.854	0.819	0.834	0.790
DensityEst (z, a)	0.123	0.981	0.876	0.925	0.929	0.769
DensityEst (z)	0.438	0.993	0.799	0.667	0.778	0.640
JRD $(\epsilon + 0.3)$	0.561	0.987	0.609	0.753	0.712	-
JRD $(\epsilon - 0.3)$	0.033	0.852	0.957	0.902	0.909	-
JRD	0.049	0.931	0.944	0.937	0.941	0.967

Table 11: Performance comparison of different uncertainty quantification methods.

The *TotalUncertainty* method assumes a fixed unit variance and predicts only the mean [28], without distinguishing between aleatoric and epistemic uncertainty. *EnsembleRSSM* [25, 26] trains an ensemble of transition models f_θ, using random sampling of ensemble indices at each step and estimating uncertainty via the variance of mean predictions. The *MaxAleatoric* approach, defined as $\max_k \|\Sigma_{\psi_k}(z_t, a_t)\|_F$, uses the maximum variance across the ensemble as a proxy for max aleatoric uncertainty [24, 58]. Additionally, we evaluate a density-estimation method based on neural spline flows [18] trained on the learned latent space, which estimates the likelihood of either latent-action pairs (z, a) or latent states alone (z).

Overall, the JRD formulation achieves the best performance in both the balanced accuracy of the safety value function and the closed-loop evaluation. While the gap is less pronounced in the Dubins car setting, which is relatively simple and contains limited aleatoric uncertainty, methods that fail to distinguish between aleatoric and epistemic uncertainty consistently underperform. These approaches struggle to isolate epistemic uncertainty, which is critical for detecting OOD transitions that stem from limited training coverage. Density-based methods show higher false positive rates, showing limited effectiveness in modeling likelihood in high-dimensional latent spaces. In particular, latent-only density models exhibit the worst performance, frequently misclassifying in-distribution safe states as OOD. This is likely due to the latent dynamics model hallucinating overconfident predictions on OOD actions during imagination, highlighting the importance of transition-based OOD detection for the uncertainty-aware reachability analysis in imagination.

Reward Term	Condition	Weight
success	stacked_3on1 $\wedge$ ¬stacked_2on1 $\wedge$ ¬stacked_3on2 $\wedge$ ¬c2_drop	+10
failure	c2_drop $\vee$ c3_drop	-10
check_3on1	Block c_3 is stacked on c_1	+1
not_stacked_2on1	Block c_2 is not stacked on c_1	+1
not_stacked_3on2	Block c_3 is not stacked on c_2	+2
dist_12	Relative distance between c_1 and c_2	+5

Table 12: Reward terms used for training the task policy. The goal is to extract the middle block (c_2) and place it on the base block (c_1) without collapsing the tower.

D.2 Block Plucking

Task Policy Training. The task policy is trained using DreamerV3 [10] with dense reward signals. The environment consists of three blocks: c_1 (base block), c_2 (middle target block), and c_3 (top block). The goal is to extract c_2 from the tower and place it on top of c_1 without causing the tower to collapse. Table 12 shows the reward design for training the task policy. The action space is normalized to [-1, 1], and the task policy is modeled as a Gaussian distribution. During execution, only the mean is used, with a small additive noise sampled from Uniform [-0.02, 0.02]. Note that for training the world model and latent safety filter, only binary failure labels are used.

Experimental Setup In Table 2, *No Filter* refers to the base policy executed without any safety intervention. *LatentSafe* uses only the learned failure margin function without explicitly modeling OOD failures. *SafeOnly* represents a setting where both the latent dynamics and safety filter are trained exclusively on successful demonstrations with 1500 trajectories, implicitly treating all failures as OOD. This setup aligns with prior approaches in failure detection and safety analysis that define the safe set based solely on successful demonstrations [22, 4].
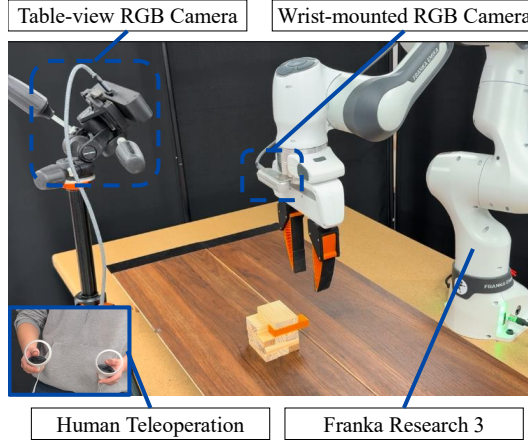
D.3 Jenga Experiments

Figure 11: Setup for the Jenga experiments.

Hardware Setup Fig. 11 shows the setup. The fixed-base Franka Research 3 manipulator is equipped with a 3D-printed gripper [84]. Two RGB cameras (third-person and wrist-mounted) capture 256×256 images at 15 Hz. It also takes 7D proprioceptive inputs (6D end-effector pose and gripper state) as input. A teleoperator uses a Meta Quest Pro to control the end-effector pose and gripper state. The robot must extract a target orange block from a tower and place it on top without causing collapse—a task characterized by high uncertainty due to complex contact dynamics and limited coverage in the offline dataset.

We keep the block configurations fixed, although minor variations in block positions naturally occur across trajectories. While the model cannot reliably filter block towers with entirely different configurations, it can still detect such settings as out-of-distribution (see Sec. B.4). We believe that collecting more diverse demonstrations with varying configurations would enable the world model to build confidence across different setups and generalize more effectively to broader scenarios.

Method	Failure ↓	Filtered(%)	Model Loss ↓
UNISafe	0.08 ± 0.08	0.19	28.49 ± 7.72
LatentSafe	0.82 ± 0.11	0.08	33.25 ± 13.22

Table 13: Open Loop Evaluation.

Open-loop Rollout Experiments For open-loop experiments, *UNISafe* uses a filtering threshold of $\delta = 0.05$. In contrast, *LatentSafe* exhibits inflated safety values due to overestimation of OOD actions. To enable effective filtering and ensure a fair comparison, we increase the threshold to $\delta = 0.2$ for *LatentSafe*, allowing it to identify the safe set more reasonably. However, even with the higher threshold, it still selects uncertain actions, as the elevated value estimates are attributed to the overestimation of these OOD actions. The detailed results are summarized in Table 13. The detailed results are summarized in Table 13. *LatentSafe* still fails to intervene at the appropriate moment and exhibits higher model loss.

E Additional Results.

E.1 How do dataset size and failure classifier performance affect the safety filter?

	Number of Random Trajectories					
	0	**10**	**50**	**100**	**500**	**1000**
Safe (%)	100.0	99.4	98.5	97.5	91.1	85.0
Unsafe (%)	0.0	0.6	1.5	2.5	8.9	15.0
Failure (%)	0.0	0.4	1.1	1.9	6.7	11.3
Failure Classifier (Acc.)	0.82	0.90	0.91	0.89	0.96	0.97
B.Acc (*LatentSafe*)	0.50	0.79	0.84	0.75	0.97	0.97
B.Acc (*UNISafe*)	0.84	0.93	0.93	0.92	0.97	0.97

Table 14: Dataset and failure classifier configurations for ablations on the Dubins Car.

Our uncertainty-aware safety filter relies on two types of failure sets: (*i*) the known failure set, derived from labeled failure data, and (*ii*) the OOD failure set, which accounts for distributional shift and epistemic uncertainty. As the dataset size increases, the latent world model and failure classifier become more accurate, reducing the size and impact of the OOD failure set. In contrast, with smaller datasets, large regions of the state space remain uncovered, making the OOD failure set crucial for robust safety filtering.

To investigate how dataset size impacts the learned safety filter, we perform an ablation study varying the number of random trajectories. Following the Dubins Car setup in Sec. 6.1, we construct a dataset consisting of 1000 expert trajectories that never enter the ground-truth unsafe set, along with a varying number of random trajectories, some of which do enter unsafe or failure regions. Using privileged state information, we train a failure margin function to approximate the known failure set. The number of random trajectories is varied across $\{0, 10, 50, 100, 500, 1000\}$. When fewer random trajectories are included, the dataset is biased toward safe states, increasing epistemic uncertainty near the unsafe set and reducing the accuracy of the failure classifier, thereby necessitating the inclusion of OOD failure modeling to capture risk in unexplored regions.

Table 14 summarizes the dataset statistics and failure classifier performance across different dataset sizes. As the number of random trajectories increases, a larger portion of the failure region is covered, improving the failure classifier's accuracy.
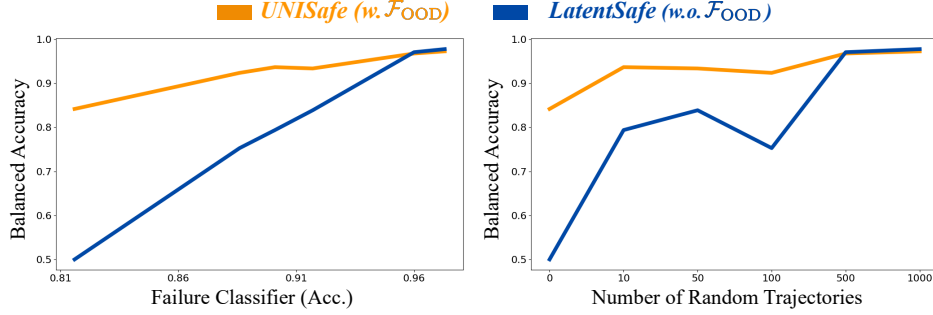
Figure 12: *UNISafe* can synthesize a much more robust safety filter even under an unreliable world model. When trained on smaller datasets with an ineffective failure classifier, *LatentSafe* results in an unreliable safety filter, whereas *UNISafe* remains robust by explicitly incorporating OOD failures.

We then evaluate the performance of the safety value function trained under each dataset configuration. The results indicate that incorporating OOD failures via uncertainty quantification is particularly beneficial when the dataset does not adequately cover the true failure set. When fewer random trajectories are available, safety filters trained without OOD modeling exhibit significantly degraded performance due to unmodeled epistemic uncertainty in the dynamics model. Figure 12 visualizes the accuracy of the learned safety value function with respect to the ground-truth failure set. The results demonstrate that integrating OOD detection yields substantially more robust performance, especially when the failure classifier is weak and the dataset is small. In contrast, the baseline approach, trained without OOD detection, fails to maintain safety in such challenging conditions.

E.2 The safety filter reliably safeguards diverse base policies.

	π^{task}	Method	Safe Success ($\uparrow$)	Failure ($\downarrow$)	Incompletion	Filtered (%)	Seq. Length	Model Error ($\downarrow$)
Normal	Dreamer	No Filter	0.58	0.41	0.01	0.0 ± 0.0	61.2 ± 45.7	59.3 ± 3.3
		SafeOnly	0.71	0.28	0.01	13.5 ± 3.5	70.25 ± 54.5	46.9 ± 2.6
		LatentSafe	0.68	0.30	0.01	7.2 ± 2.6	70.01 ± 55.8	60.2 ± 4.7
		UNISafe	0.72	0.20	0.08	37.7 ± 6.7	95.7 ± 90.2	43.1 ± 1.2
	Diffusion Policy	No Filter	0.52	0.44	0.04	0.0 ± 0.0	87.3 ± 74.6	89.8 ± 20.9
		SafeOnly	0.50	0.38	0.12	26.6 ± 3.8	126.9 ± 101.0	85.1 ± 20.8
		LatentSafe	0.42	0.51	0.07	9.3 ± 2.6	93.5 ± 82.1	70.7 ± 18.1
		UNISafe	0.57	0.15	0.28	37.6 ± 5.5	150.4 ± 120.0	48.3 ± 5.5
Hard	Dreamer	No Filter	0.48	0.52	0.00	0.0 ± 0.0	53.03 ± 43.04	44.5 ± 4.8
		SafeOnly	0.47	0.46	0.07	50.7 ± 8.5	68.9 ± 77.3	45.5 ± 9.2
		LatentSafe	0.51	0.49	0.00	15.9 ± 3.9	60.3 ± 52.7	44.9 ± 10.9
		UNISafe	0.64	0.22	0.14	40.0 ± 6.1	103.5 ± 95.0	37.3 ± 2.5
	Diffusion Policy	No Filter	0.17	0.58	0.25	0.0 ± 0.0	179.4 ± 91.5	79.6 ± 20.0
		SafeOnly	0.29	0.53	0.18	22.9 ± 2.5	163.9 ± 88.0	82.7 ± 19.7
		LatentSafe	0.18	0.62	0.20	6.4 ± 0.9	183.5 ± 88.7	75.2 ± 22.5
		UNISafe	0.38	0.31	0.31	21.9 ± 2.0	179.8 ± 91.3	43.9 ± 13.5

Table 15: Additional result on block plucking in simulation environments.

For a more thorough evaluation, we additionally consider a *Hard* setting (see Fig. 13), which varies the block size, weight, and friction to allow for a more comprehensive assessment. As nominal task policies, we evaluate (1) DreamerV3[10], trained online with a dense reward signal, and (2) Diffusion Policy [85], an imitation learning trained on 200 safe trajectories. Table 15 shows that *UNISafe* consistently minimizes failure rates and model errors compared to the baselines.

E.3 Is model-based imagination or explicit uncertainty quantification essential?

We perform ablations to evaluate the necessity of the two core components in our uncertainty-aware latent-space reachability framework: (1) a *latent dynamics model* for reachability analysis in imagination and (2) explicit *epistemic uncertainty quantification* for preventing distributional shift.
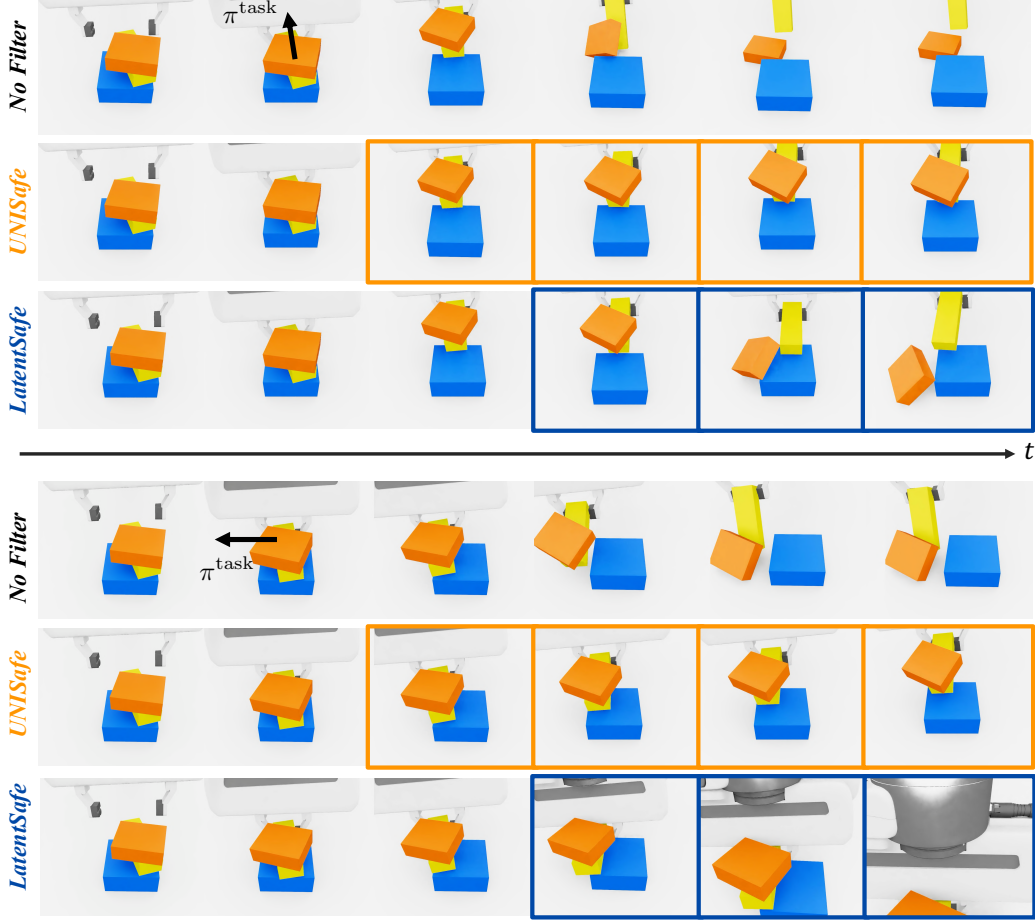
Figure 13: Qualitative results on the *Hard* setting. The task policy attempts to pick the block but naively pulls it in one direction, causing failures where the orange block falls. *UNISafe* accurately detects the boundary of the unsafe set and prevents failures caused by blocks falling with momentum. The safety policy reliably corrects the task policy's actions by proposing safe, in-distribution alternatives, keeping the block stable. In contrast, *LatentSafe* detects the unsafe set too late and eventually proposes abrupt, unsafe actions that lead to failure.

Conservative Q-Learning In the model-free offline RL setting, policies can be learned solely from offline datasets, without learning a learned dynamics model. To address value overestimation on out-of-distribution actions, Conservative Q-Learning (CQL) [62] introduces a conservative training objective that penalizes high Q-values for unseen or randomly sampled actions. This is achieved by augmenting the standard Bellman error with a behavioral-cloning-style regularization term, which constrains the Q-values for out-of-distribution actions while preserving those associated with in-distribution actions:

$$\min_{Q} \; \alpha \, \mathbb{E}_{z \sim \mathcal{D}} \left[\log \sum_{a} \exp(Q(z,a)) - \mathbb{E}_{a \sim \hat{\pi}_{\beta}(a|z)} \left[Q(z,a) \right] \right], \tag{24}$$

where $\hat{\pi}_{\beta}$ denotes the behavior policy from the offline dataset. We apply this conservative loss to train the safety filter. The safety value function and policy are learned on top of the latent representation space using offline transitions without relying on model-based imagination.

This conservatism principle can also be extended to the model-based setting without requiring explicit uncertainty estimation. COMBO [63] combines offline transitions with model-generated rollouts to train a value function and regularizes Q-values on out-of-support state-action pairs generated by the model. In our case, we adopt the same conservative objective from Eq. 24, applying it to

30

imagined latent transitions produced by the learned latent dynamics model, thus eliminating the need for explicit epistemic uncertainty quantification.

Analysis One limitation of the conservative objective is that, by directly penalizing the value function, it introduces bias into the learned safety values. As a result, the value function can no longer reliably serve as a level-set representation for identifying the unsafe set via its zero sublevel set. To address this, we use a calibration dataset to select a value threshold δ that best separates safe and unsafe states in practice.

Quantitative results of this ablation study are provided in Table 2. While learning a safety monitor and policy without model-based imagination is feasible, its effectiveness is limited in high-dimensional visual manipulation settings where the offline dataset may not adequately cover the state space, restricting the quality of reachability approximation. Furthermore, not using uncertainty quantification and relying solely on the conservative loss results in overly conservative filtering behavior. These results suggest that although conservative objectives are effective in standard offline RL tasks that aim to maximize expected returns, they are less suitable for safety analysis. Safety-critical applications require accurate specification of the safe and unsafe sets, which conservative regularization alone fails to guarantee.

E.4 Can uncertainty-penalized offline RL ensure the safety of the task policy?

Offline reinforcement learning (RL) methods frequently incorporate uncertainty estimation to enhance safety and robustness when learning task policies from static datasets. To assess the effectiveness of uncertainty-penalized policy learning in this context, we train a task policy using an offline model-based reinforcement learning (MBRL) framework (LOMPO [25]) that employs a latent dynamics model for image-based control. It formulates an uncertainty-penalized POMDP, where an ensemble of RSSMs is used to estimate epistemic uncertainty and penalize transitions softly accordingly during policy optimization. We investigate this with the visual manipulation tasks in simulation.

In contrast to the safety filter, which relies learned failure margin function, we learn dense task-relevant rewards $\bar{r}_\theta(z_t, a_t)$ for the task policy training. To ensure the penalty term accurately reflects epistemic uncertainty, we employ JRD-based uncertainty quantified by ensembles (Sec. 5.1). The resulting reward function used for training is defined as: $r_t(z_t, a_t) = \bar{r}_\theta(z_t, a_t) - \lambda \, \mathrm{D}(z_t, a_t)$, where $\lambda = 0.5$ controls the strength of the uncertainty penalty.

	π^{task}	Safety Filter	Safe Success	Failure	Incompletion	Filtered (%)	Seq. Length	Model Error
Normal	LOMPO [25]	No Filter	0.36	0.63	0.01	0.0 ± 0.0	66.3 ± 54.2	79.1 ± 4.1
		UNISafe	0.41	0.26	0.33	60.3 ± 6.8	154.1 ± 122.9	54.3 ± 8.8
Hard	LOMPO [25]	No Filter	0.05	0.95	0.00	0.0 ± 0.0	83.7 ± 50.9	71.2 ± 16.3
		UNISafe	0.41	0.36	0.23	41.7 ± 4.9	143.5 ± 106.5	32.7 ± 10.3

Table 16: Rollout results with offline learned task policy following LOMPO with the pessimistic MDP.

Table 16 presents the experimental results. The uncertainty-penalized policy trained via LOMPO exhibits high failure rates and accumulates large model errors during rollouts. It exhibits limited performance, especially in the Hard setting, suggesting that a soft uncertainty penalty in offline MBRL is insufficient for safety. Using the same offline dataset, we instead train a safety filter that explicitly incorporates uncertainty and use it to filter the task policy learned by LOMPO. While this approach does not guarantee zero failure, it significantly reduces the failure rate and also lowers the model error. These results suggest that an uncertainty-aware safety filter learned from offline data is more effective at ensuring safety than directly penalizing uncertainty during task policy optimization. Additionally, when uncertainty penalties are omitted entirely during offline policy learning, the resulting policy fails to perform meaningful behavior, primarily due to value overestimation, a well-documented challenge in offline RL.