VIII. Appendix

A. Implementation Details

Dubins Car. For the Dubins Car experiments, we use a Recurrent State-Space Model (RSSM) [1] as the world model. RSSM decomposes the latent state into deterministic and stochastic components, $z_t := [h_t | x_t]$, where the stochastic latent is modeled as a distribution and optimized via the KL divergence between its prior and posterior. We build on the open-source implementation of DreamerV3[4]. We use a continuous stochastic latent space modeled as a 32-dimensional Gaussian. The action space is normalized to [-1, 1]. The hyperparameters for the Dubins Car experiments are provided in Table IV.

Hyperparameter	Value
Image Dimension	[128, 128, 3]
Action Dimension	1
Stochastic Latent	Gaussian
Latent Dim (Deterministic)	512
Latent Dim (Stochastic)	32
Latent Dim (Failure Projector)	512
Activation Function	SiLU
Encoder CNN Depth	32
Encoder MLP Layers	5
Failure Projector Layers	2
Batch Size	16
Batch Length	64
Optimizer	Adam
Learning Rate	1e-4
Iterations	10000

TABLE IV: Dreamer Hyperparameters

Vision-Based Sweeping. For the real-world hardware task, we use DINO-WM [4], a transformer-based world model that represents latent states using the patch tokens of DINOv2 [37]. The DINOv2 encoder is kept frozen, and only the parameters of a vision transformer are trained. The latent $\hat{z}_{t+1}$ consisting of dense patch tokens is deterministically predicted by conditioning on a sequence of past normalized actions $a_{t-H:t}$ and latent tokens $z_{t-H:t}$ from the previous H timesteps. The model is trained with teacher forcing to ensure temporal consistency by regressing the latent patches, using the mean squared error (MSE) loss: $\mathcal{L}_{\text{DINO}} = \|\mathcal{E}(o_{t+1}) - f_z(z_{t-H:t}, a_{t-H:t})\|^2$. The hyperparameters for DINO-WM are provided in Table V. To measure the similarity of DINOv2 features in Sec. VI-C, we compute the norm of the patch tokens to obtain a global feature $z \in \mathbb{R}^{384}$ from the dense patches $z \in \mathbb{R}^{N_{\text{patches}} \times 384}$.

HJ Reachability Analysis. To solve the latent fixed-point safety Bellman equation, we adopt DDPG [35] within an off-policy, model-based reinforcement learning framework, using the implementation from [38][5]. We model the safety value function as a latent-action value function conditioned on the constraint representation $Q(z, a; z_c)$.

[4] https://github.com/NM512/dreamerv3-torch

[5] https://github.com/jamesjingqili/Lipschitz_Continuous_Reachability_Learning

Hyperparameter	Value
Image Dimension	[224, 224, 3]
Action Dimension	3
DINOv2 Patch Size	(16 × 16, 384)
ViT depth	6
ViT attention heads	16
ViT MLP dim	2048
Latent Dim (Failure Projector)	512
Activation Function	SiLU
Failure Projector Layers	2
Batch Size	16
Batch Length	4
Optimizer	Adam
Learning Rate	5e-5
Iterations	100000

TABLE V: DINO-WM Hyperparameters

The safety policy is parameterized by an actor-network $a = \pi^{\blacktriangledown}(\cdot \mid z, z_c) \in [-1, 1]^{d_{\text{action}}}$, and the safety value is evaluated by $V^{\blacktriangledown}(z) = \max_a Q(z, a; z_c) = Q(z, \pi^{\blacktriangledown}(z, z_c))$.

Each trajectory starts from random initial states with randomly sampled constraint representations. The replay buffer $\mathcal{B}$ then stores transitions of the form $(z, z_c, a, \tilde{l}, z')$. The safety filter is optimized using the following objectives:

$$\mathcal{L}_{\text{critic}} := \mathbb{E}_{\mathcal{B}} \left[(Q(z, a; z_c) - y)^2 \right] \tag{9}$$

$$y = (1 - \gamma)\, \tilde{l} + \gamma \min\{\tilde{l}, \max_{a'} Q(z', a'; z_c)\}. \tag{10}$$

$$\mathcal{L}_{\text{actor}} := \mathbb{E}_{z \sim \mathcal{B}} \left[-Q(z, a; z_c) \right], \quad a = \pi^{\blacktriangledown}(\cdot \mid z, z_c), \tag{11}$$

where γ is scheduled from 0.85 to 0.9999. The hyperparameters for training DDPG are summarized in Table. VI.

Hyperparameter	Value
Actor Architecture	[512, 512, 512, 512]
Critic Architecture	[512, 512, 512, 512]
Normalization	LayerNorm
Activation	ReLU
Discount Factor γ	0.9999
Learning Rate (Critic)	1e-3
Learning Rate (Actor)	1e-4
Optimizer	AdamW
Number of Iterations	640000
Replay Buffer Size	1000000
Batch Size	512
Max Imagination Steps	30

TABLE VI: DDPG hyperparameters.

B. Failure Projector

In Sec. IV and Eq. (3), we define a failure projector that maps the raw latent representation of the world model into a metric space where similarities are better aligned with the user's notion of failure. Since *AnySafe* constructs the failure set based on a latent-space similarity metric, the projected latent space retains only failure-relevant features. In Sec. VI-C, we qualitatively demonstrate that the raw DINOv2 latent space is insufficient to define a failure set, yielding a noisy and uninformative similarity metric that cannot capture fine-grained position-based differences, which in turn degrades the quality of the value function.

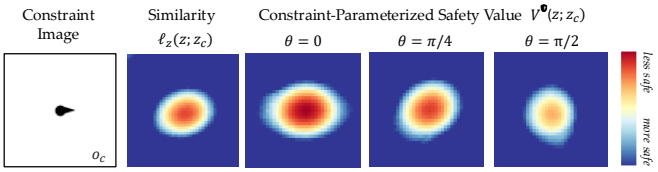

Fig. 9: **Dubins' Car Qualitative Result with Raw Unprojected Features.** We visualize the latent similarity and safety value function at heading slices $\theta \in \{0, \pi/4, \pi/2\}$ using the raw Dreamer features without applying the failure projector.

Architecture. We implement the failure projector as a 2-layer MLP, with the architecture summarized in Table VII. For RSSM, the input is the latent vector formed by concatenating the deterministic and stochastic components, $z_t = [h_t \,|\, x_t]$, resulting in an input dimension of $d_{\text{in}} = 512 + 32 = 544$. For DINO-WM, we compute the norm of the patch tokens from the dense latent features $z \in \mathbb{R}^{N_{\text{patches}} \times 384}$ and concatenate the proprioceptive state, yielding an input dimension of $d_{\text{in}} = 384 + 3 = 387$.

Layer	Input Dim	Output Dim	Normalization
`Linear`	d_{in}	d_{in}	`LayerNorm`
`Linear`	d_{in}	32	`LayerNorm`

TABLE VII: Failure Projector Architecture

Ablation. We provide a qualitative evaluation of the raw world model latent space of Dreamer in the Dubins Car. Fig. 9 shows both the feature similarities computed using unprojected world model latents and the corresponding safety value function learned with these latents. Compared to the results learned from projected features shown in Fig. 3 and Fig. 4, the unprojected similarities fail to represent position-based, failure-relevant distinctions, resulting in poorly learned value functions.

C. Calibration of Latent Similarity

In Sec. IV, we compute a threshold for defining a latent failure set and then apply this threshold at runtime for safety filtering in (8), operating on the safety value function. Formally, this calibration procedure guarantees only that the similarity measure—and the corresponding failure set—are calibrated. Approximate value function solvers (e.g, RL) can still induce errors in the downstream safety filter, and calibrating the value function directly requires stronger assumptions about access to the ground-truth unsafe set labels [33]. Nevertheless, we prove in Theorem 1 that under a perfect value function solver, the calibrated threshold δ can be applied directly to the value function learned from the similarity measure, yielding an unsafe set defined as the sub-threshold level set of the value function.

Specifically, we show that learning a threshold-dependent safety value function $V^{\blacktriangledown}_{\delta}$ with a safety margin $\ell^{\delta}_z := \ell_z - \delta$, is equivalent to learning the value function $V^{\blacktriangledown}$ from the raw latent similarity ℓ_z and applying the threshold δ post hoc. This equivalence implies that calibration can be performed after computing the value function using the latent similarities without adjusting them with thresholds, whereas threshold-dependent training would require recomputing the value function whenever the threshold changes.

Theorem 1. *Let $\ell_z : \mathcal{Z} \to \mathbb{R}$ be a safety margin function. Consider the discrete-time safety Bellman backup operator:*

$$(TV)(z_t) = (1-\gamma)\,\ell(z_t) + \gamma \min\Big\{\ell(z_t),\ \max_{a_t \in \mathcal{A}} \mathbb{E}_{\hat{z}_{t+1} \sim f_z(\cdot|z_t, a_t)}\big[V(\hat{z}_{t+1})\big]\Big\}. \tag{12}$$

Let $V^{\blacktriangledown}$ denote its unique fixed point. For a constant $\delta \in \mathbb{R}$, define $\ell^{\delta}_z := \ell_z - \delta$ and let T_δ be the safety Bellman backup operator T where the margin function ℓ_z is replaced by ℓ^{δ}_z:

$$(T_\delta V)(z_t) = (1-\gamma)\,(\ell(z_t) - \delta) + \gamma \min\Big\{\ell(z_t) - \delta,\ \max_{a_t \in \mathcal{A}} \mathbb{E}_{\hat{z}_{t+1} \sim f_z(\cdot|z_t, a_t)}\big[V(\hat{z}_{t+1})\big]\Big\}. \tag{13}$$

Let the fixed point of this Bellman backup be $V^{\blacktriangledown}_{\delta}$. Then:

$$V^{\blacktriangledown}_{\delta} = V^{\blacktriangledown} - \delta \quad \textit{and} \tag{14}$$

$$\{\, z :\ V^{\blacktriangledown}_{\delta}(z) < 0 \,\} = \{\, z :\ V^{\blacktriangledown}(z) < \delta \,\}. \tag{15}$$

Proof. Let $V : \mathcal{Z} \to \mathbb{R}$ be any value function. Recall the linearity of expectation and the shift-invariance identities:

$$\min\{a-\delta, b-\delta\} = \min\{a,b\} - \delta,$$
$$\max\{a-\delta, b-\delta\} = \max\{a,b\} - \delta.$$

Using $\ell^{\delta}_z = \ell_z - \delta$, we compute

$$\begin{aligned}
(T_\delta(V-\delta))(z_t) &= (1-\gamma)\,(\ell_z(z_t) - \delta) \\
&\quad + \gamma \min\Big\{\ell_z(z_t) - \delta,\ \max_{a_t \in \mathcal{A}} \mathbb{E}\big[\,V(\hat{z}_{t+1}) - \delta\,\big]\Big\} \\
&= (1-\gamma)\,\ell_z(z_t) - (1-\gamma)\delta \\
&\quad + \gamma\Big(\min\Big\{\ell_z(z_t),\ \max_{a_t \in \mathcal{A}} \mathbb{E}\big[V(\hat{z}_{t+1})\big]\Big\} - \delta\Big) \\
&= \Big((1-\gamma)\,\ell_z(z_t) \\
&\quad + \gamma \min\Big\{\ell_z(z_t),\ \max_{a_t \in \mathcal{A}} \mathbb{E}\big[V(\hat{z}_{t+1})\big]\Big\}\Big) - \delta \\
&= (TV)(z_t) - \delta.
\end{aligned}$$

In particular, if $V := V^{\blacktriangledown}$ is the fixed point of T, then,

$$T_\delta(V^{\blacktriangledown} - \delta) = TV^{\blacktriangledown} - \delta = V^{\blacktriangledown} - \delta, \tag{16}$$

where $V^{\blacktriangledown} - \delta$ is a fixed point of T_δ. Since the fixed point is unique [16], $V^{\blacktriangledown}_{\delta} = V^{\blacktriangledown} - \delta$.

Thus, the corresponding unsafe sets are equivalent:

$$\begin{aligned}
\{\, z :\ V^{\blacktriangledown}_{\delta}(z) < 0 \,\} &= \{\, z :\ V^{\blacktriangledown}(z) - \delta < 0 \,\} \\
&= \{\, z :\ V^{\blacktriangledown}(z) < \delta \,\}.
\end{aligned} \tag{17}$$

□