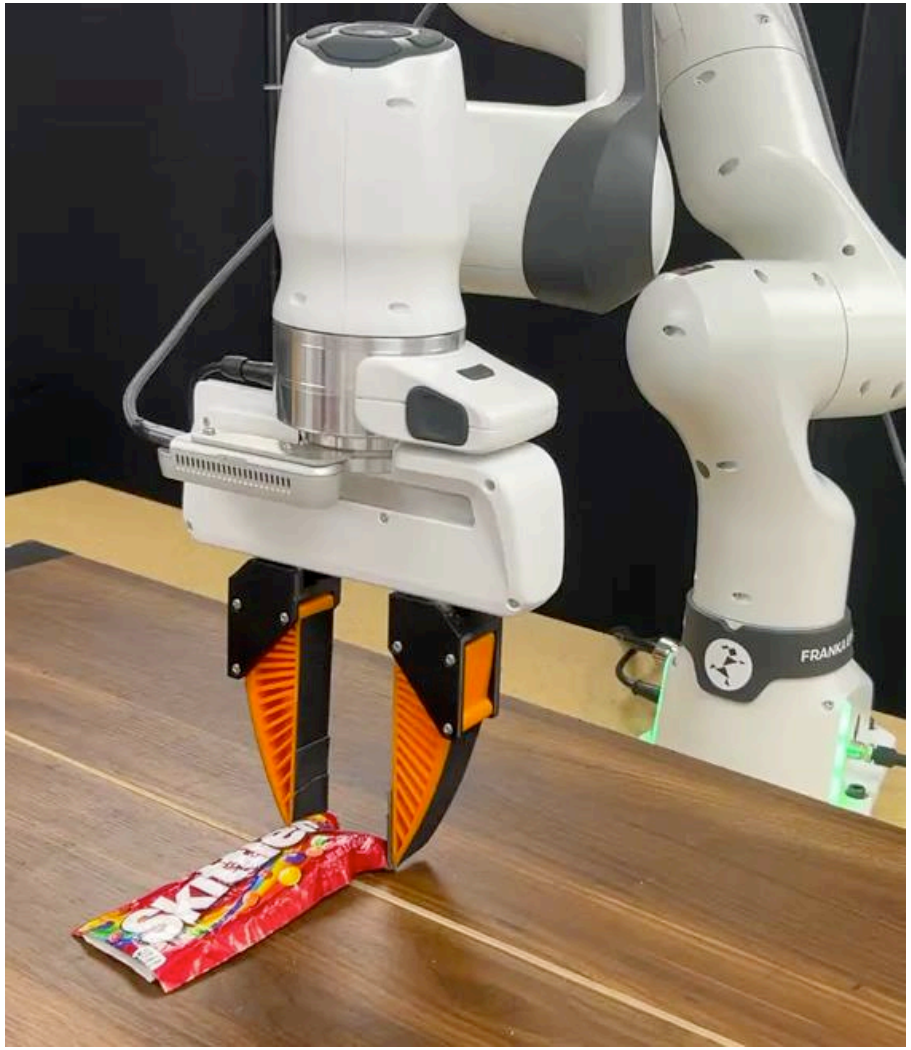
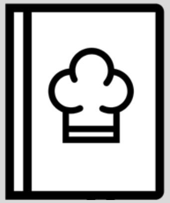
What is holding us back?



Four Ingredients for Safety



x
f
D
$\mathcal{F}$

What is *state space*, X?

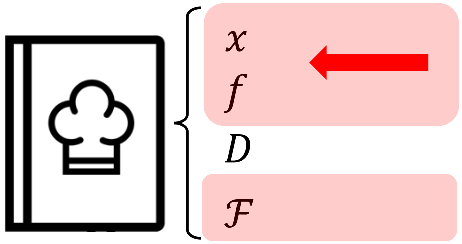
What are the *dynamics*, f?

What is *failure set*, $\mathcal{F} \subset X$?

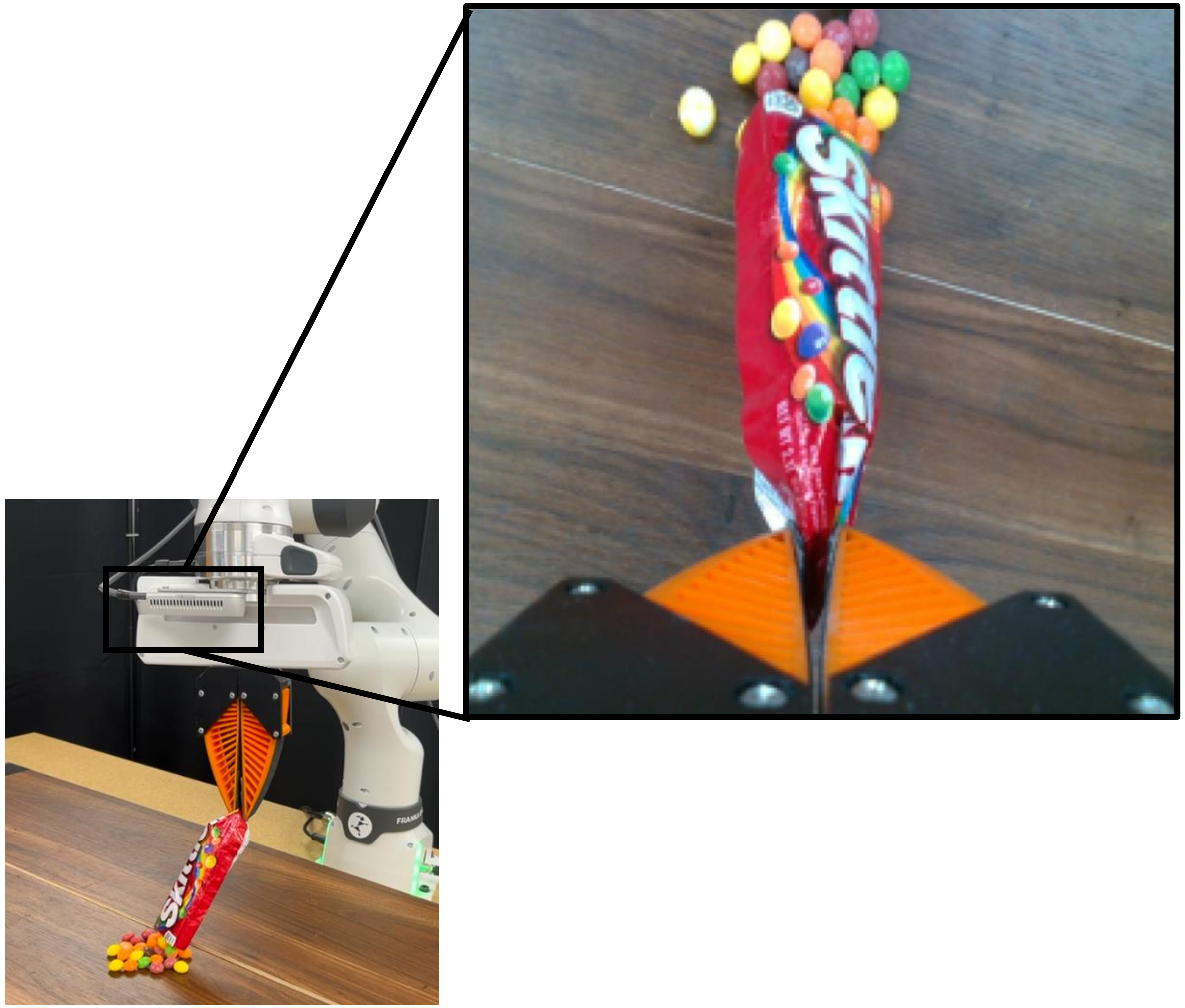
How can safety go beyond collision-avoidance?

We need to reason about safety

beyond hand-designed state and dynamics



Although we don’t know how to write $\underbrace{x}_{\textit{state}}$ or $\underbrace{\mathcal{F}}_{\textit{failure}}$ by hand…



Failure is observable from high-dimensional observations!

But how can we *predict* if the robot’s actions will result in these (failure) observations?

🔑 **Idea:** Compute a safety filter in the ***latent state space*** learned by generative world models

Before	*Ours*
$x' = f_x(x, a)$	$z' = f_z(z, a)$
$\mathcal{F} \subset X$	$\mathcal{F} \subset Z$

K. Nakamura, L. Peters, A. Bajcsy. "Generalizing Safety Beyond Collision-Avoidance via Latent-Space Reachability Analysis." RSS, 2025.

🔑 **Idea:** Compute a safety filter in the *latent state space* learned by generative world models

Before

$$x' = f_x(x, a)$$

$$\mathcal{F} \subset X$$

Ours

$$z' = f_z(z, a)$$

$$\mathcal{F} \subset Z$$

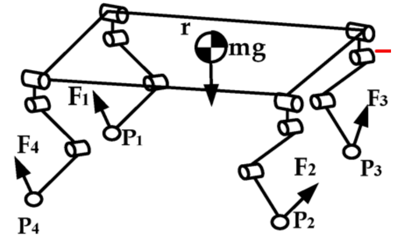
States *(pose, joint angles, velocities)*

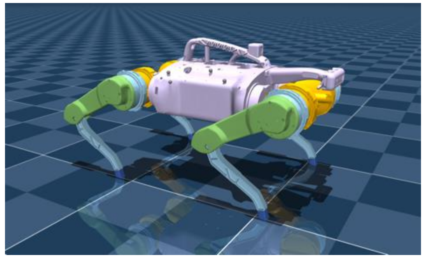
$$x = \left[\mathbf{p}, \dot{\mathbf{p}}, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \boldsymbol{\theta}_{\mathrm{J}}, \dot{\boldsymbol{\theta}}_{\mathrm{J}}\right]$$

Safety Spec *(robot falls)*

$$g(x) = \min_i \left\{p_g^i - \bar{p_g}^i\right\},$$

Dynamics *(first-principles model, physics simulator)*



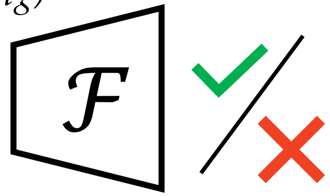


States *(embedding of image(s))*

z

Safety Spec *(classifier on embedding)*

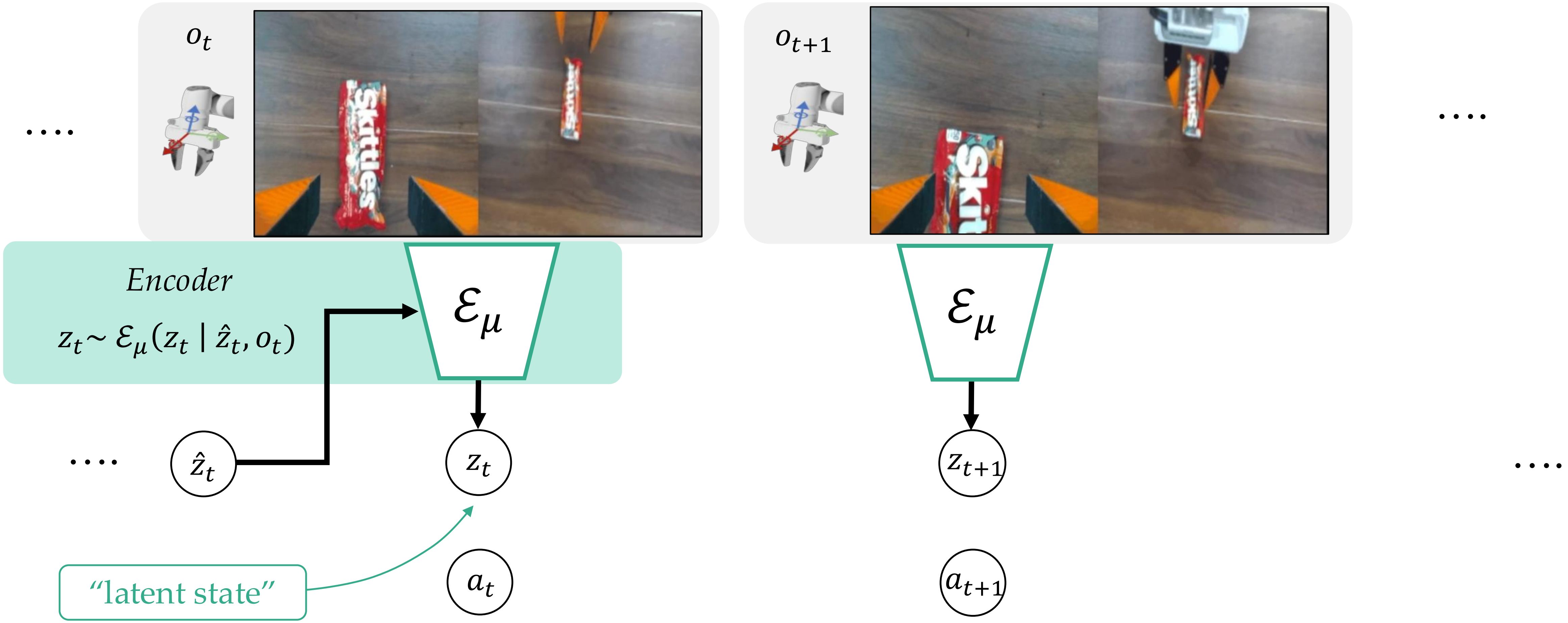
z



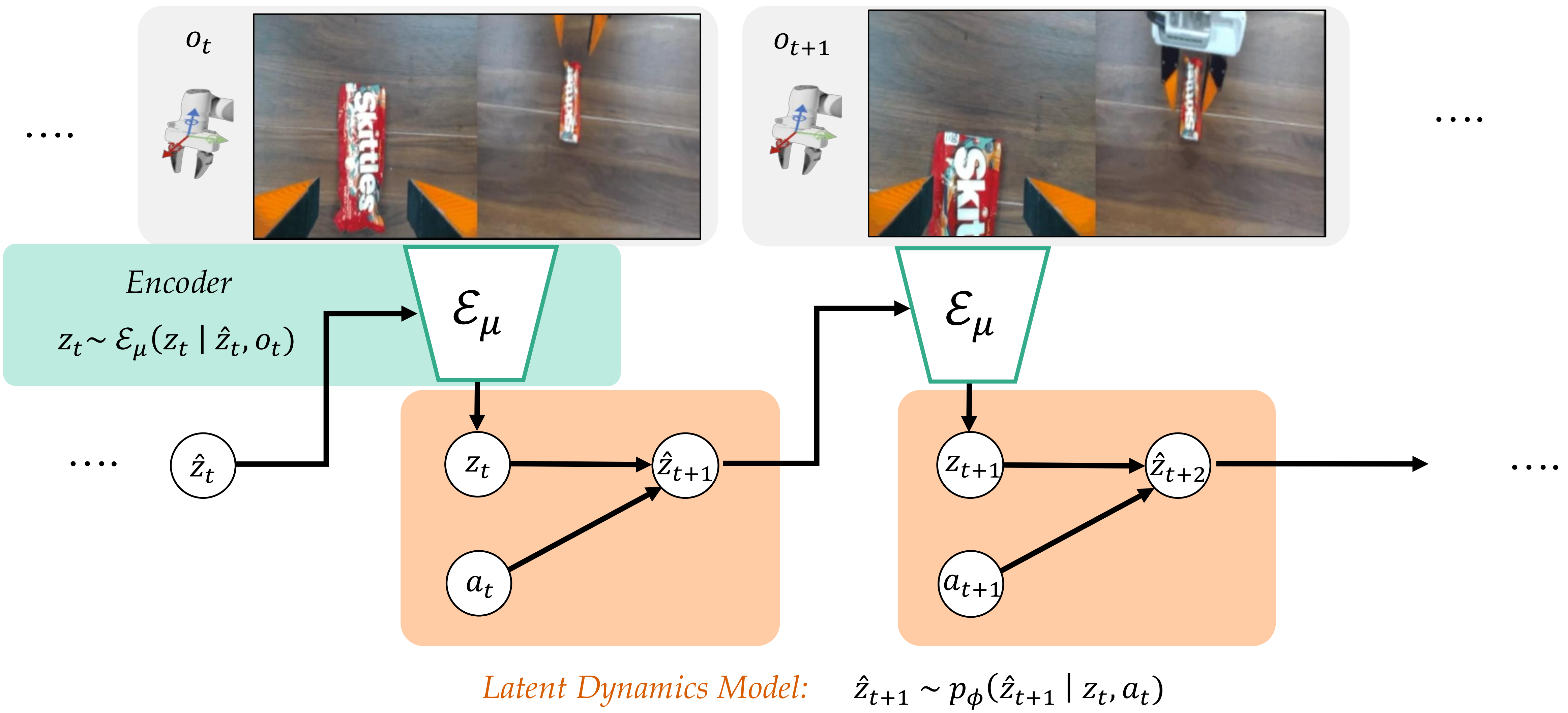
"World Model" Dynamics *(operates on embedding)*

$$z' = f_z(z, a)$$

World Model Training Time

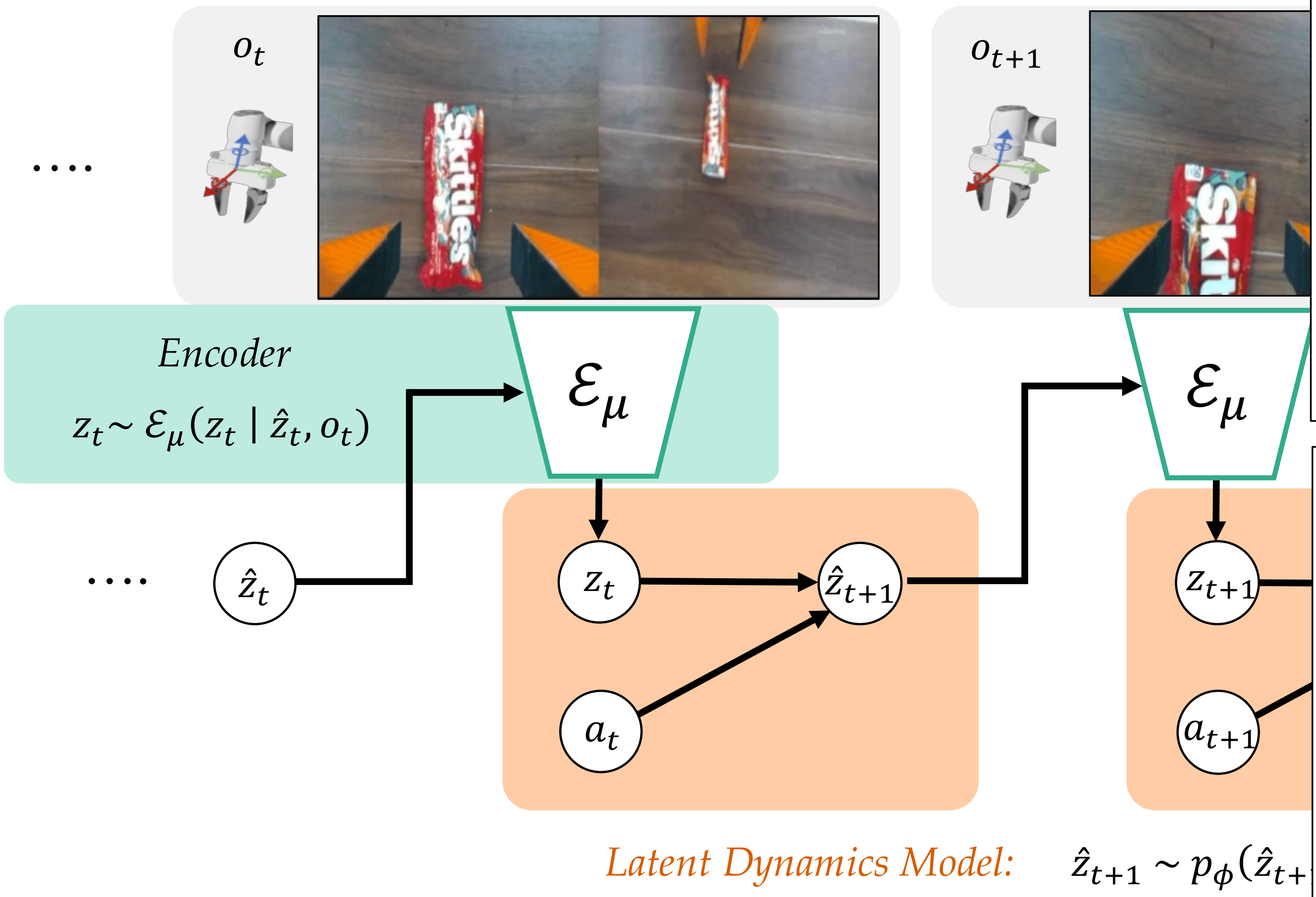


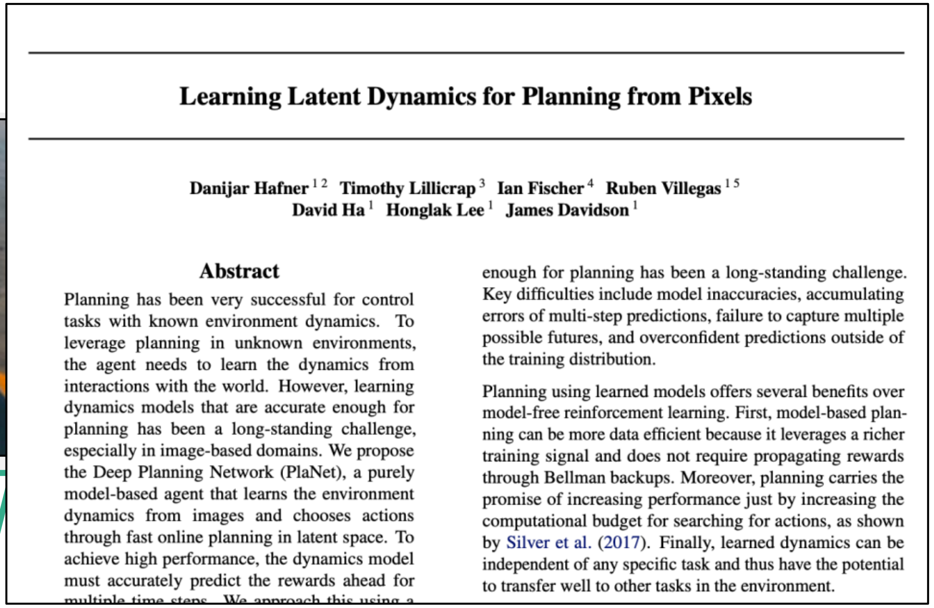
World Model Training Time



Training loss: reconstruction or teacher-forcing (min. diff. between $\hat{z}_t$ and z_t) + auxiliary losses

World Model Training Time





Learning Latent Dynamics for Planning from Pixels

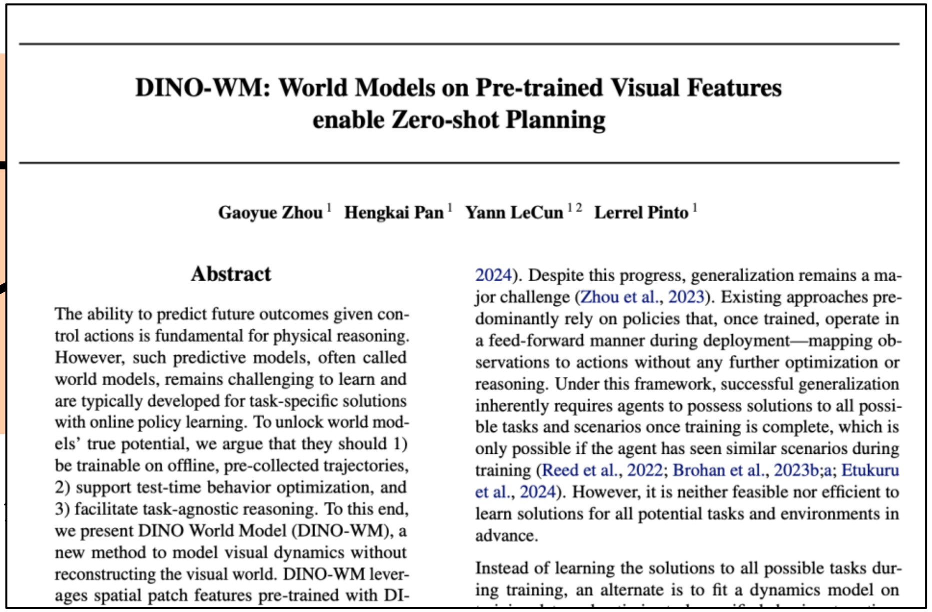
Danijar Hafner [1 2] **Timothy Lillicrap** [3] **Ian Fischer** [4] **Ruben Villegas** [1 5] **David Ha** [1] **Honglak Lee** [1] **James Davidson** [1]

Abstract

Planning has been very successful for control tasks with known environment dynamics. To leverage planning in unknown environments, the agent needs to learn the dynamics from interactions with the world. However, learning dynamics models that are accurate enough for planning has been a long-standing challenge, especially in image-based domains. We propose the Deep Planning Network (PlaNet), a purely model-based agent that learns the environment dynamics from images and chooses actions through fast online planning in latent space. To achieve high performance, the dynamics model must accurately predict the rewards ahead for

enough for planning has been a long-standing challenge. Key difficulties include model inaccuracies, accumulating errors of multi-step predictions, failure to capture multiple possible futures, and overconfident predictions outside of the training distribution.

Planning using learned models offers several benefits over model-free reinforcement learning. First, model-based planning can be more data efficient because it leverages a richer training signal and does not require propagating rewards through Bellman backups. Moreover, planning carries the promise of increasing performance just by increasing the computational budget for searching for actions, as shown by Silver et al. (2017). Finally, learned dynamics can be independent of any specific task and thus have the potential to transfer well to other tasks in the environment.



DINO-WM: World Models on Pre-trained Visual Features enable Zero-shot Planning

Gaoyue Zhou [1] **Hengkai Pan** [1] **Yann LeCun** [1 2] **Lerrel Pinto** [1]

Abstract

The ability to predict future outcomes given control actions is fundamental for physical reasoning. However, such predictive models, often called world models, remains challenging to learn and are typically developed for task-specific solutions with online policy learning. To unlock world models' true potential, we argue that they should 1) be trainable on offline, pre-collected trajectories, 2) support test-time behavior optimization, and 3) facilitate task-agnostic reasoning. To this end, we present DINO World Model (DINO-WM), a new method to model visual dynamics without reconstructing the visual world. DINO-WM leverages spatial patch features pre-trained with DI-

2024). Despite this progress, generalization remains a major challenge (Zhou et al., 2023). Existing approaches predominantly rely on policies that, once trained, operate in a feed-forward manner during deployment—mapping observations to actions without any further optimization or reasoning. Under this framework, successful generalization inherently requires agents to possess solutions to all possible tasks and scenarios once training is complete, which is only possible if the agent has seen similar scenarios during training (Reed et al., 2022; Brohan et al., 2023b;a; Etukuru et al., 2024). However, it is neither feasible nor efficient to learn solutions for all potential tasks and environments in advance.

Instead of learning the solutions to all possible tasks during training, an alternate is to fit a dynamics model on

Training loss: reconstruction or teacher-forcing (min. diff. between $\hat{z}_t$ and z_t) + auxiliary losses

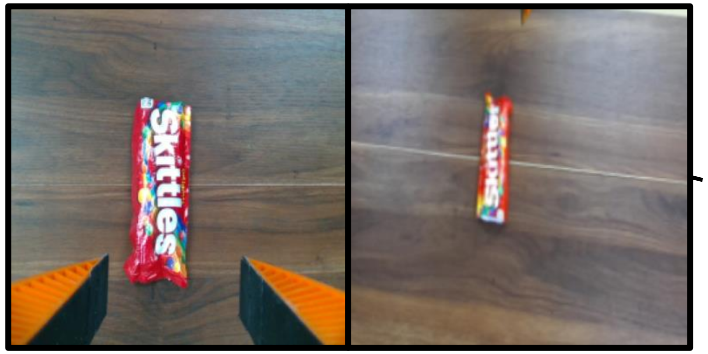
Safety Analysis Time

Given its best effort to avoid failure, how close does the robot come to failing?

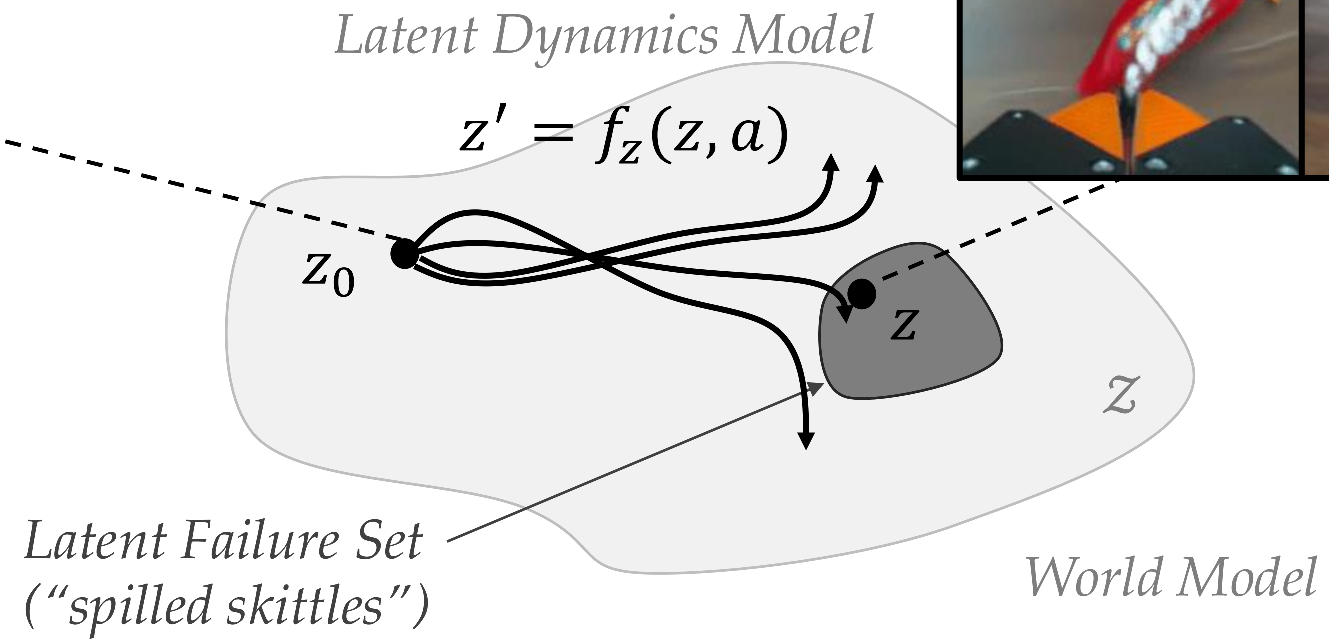
Latent Safety Problem

$$V(z_0) \coloneqq \max_{\pi}\left(\min_{t\geq 0}\ \ell_\theta(z_t^\pi)\right)$$

Initial Observation



Imagined Failure



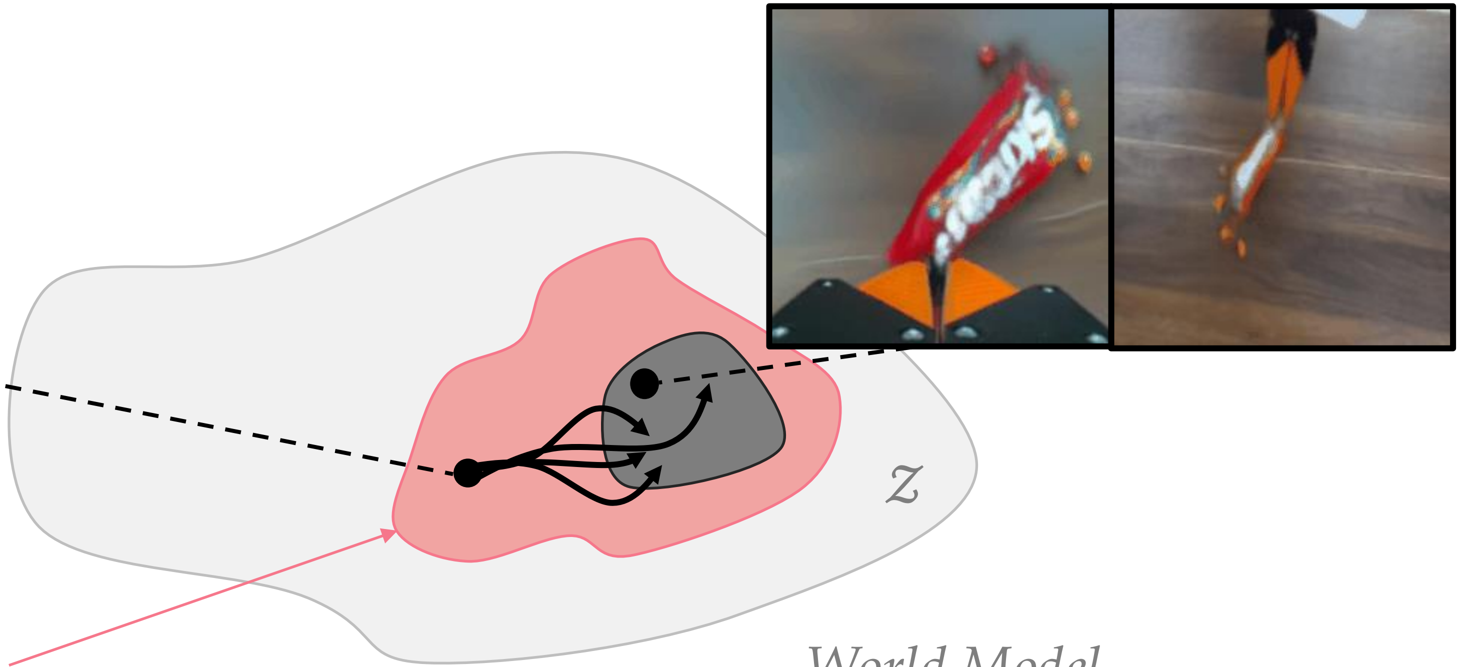
Safety Analysis Time

Latent Hamilton-Jacobi Safety Bellman Equation

$$V(z) = \min\{\ell_\theta(z), \max_{a \in \mathcal{A}} \mathbb{E}_{\hat{z}' \sim p_\phi(\cdot|z,a)}[V(\hat{z}')]\}$$

Robot is **doomed to** fail

Imagined Failure



World Model

Latent Unsafe Set

encoded as z's where $V(z) < 0$

Safety Analysis Time

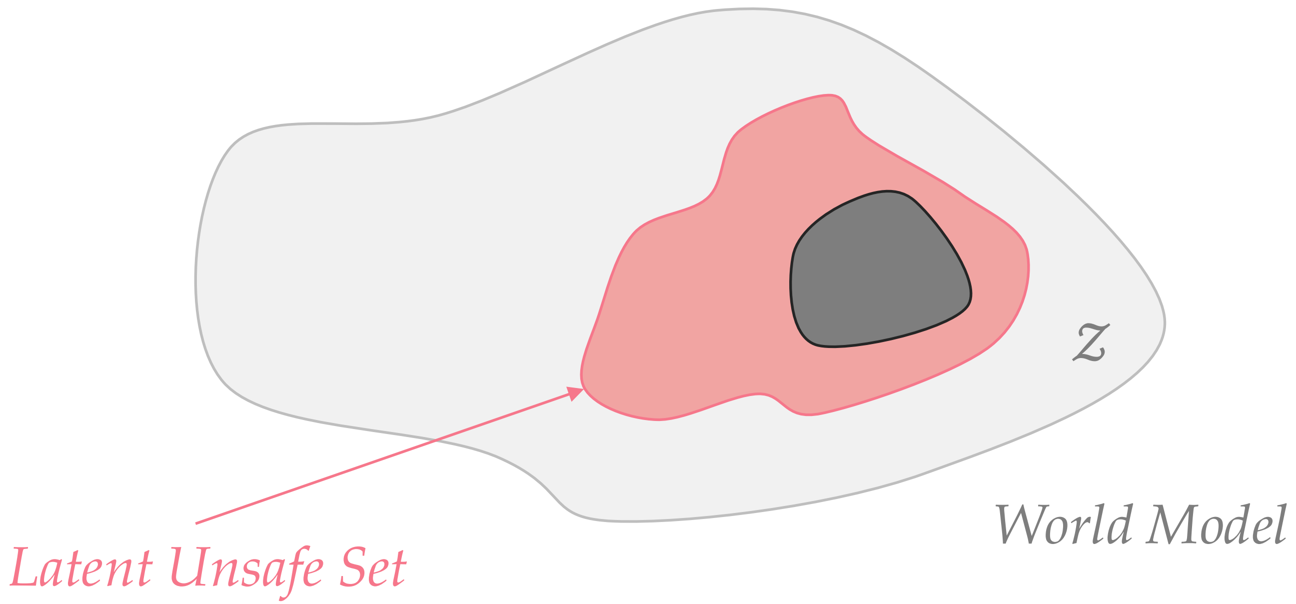
Challenge:
latent state (z) is
high-dimensional!

e.g., **512-D**!

Note: still better
than space of all
image observations:
2 x 3 × 128 × 128 =
98,304-D

Latent Hamilton-Jacobi Safety Bellman Equation

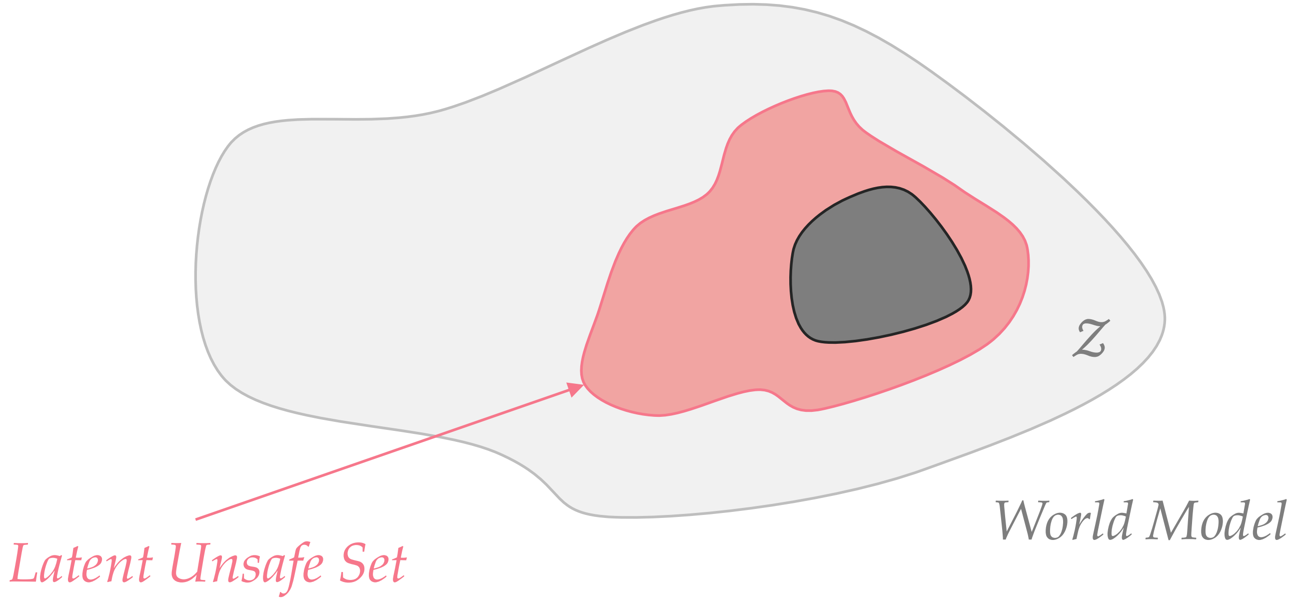
$$V(z) = \min\{\ell_\theta(z), \max_{a \in \mathcal{A}} \mathbb{E}_{\hat{z}' \sim p_\phi(\cdot|z,a)}[V(\hat{z}')]\}$$



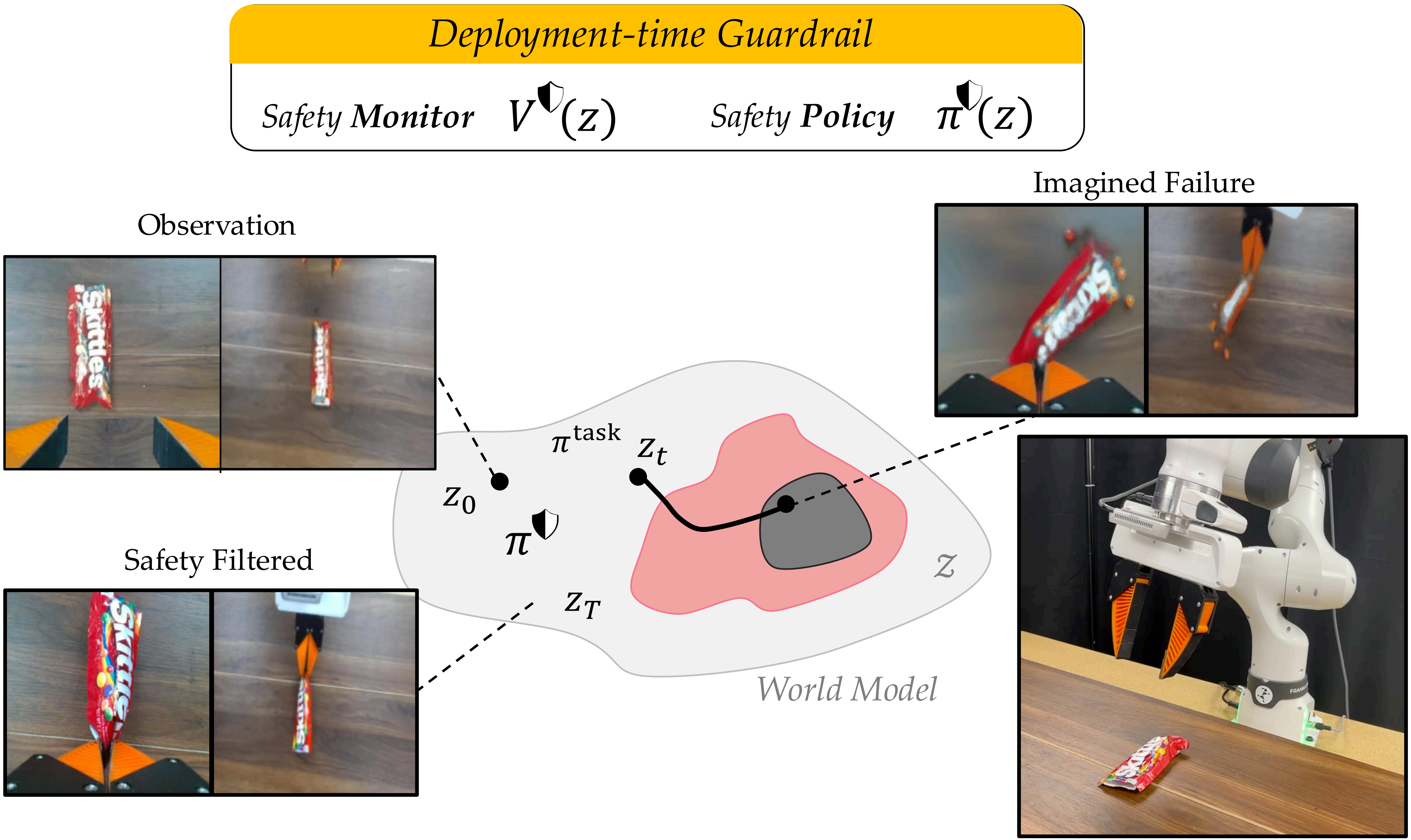
Safety Analysis Time

Approximation via Reinforcement Learning in World Model

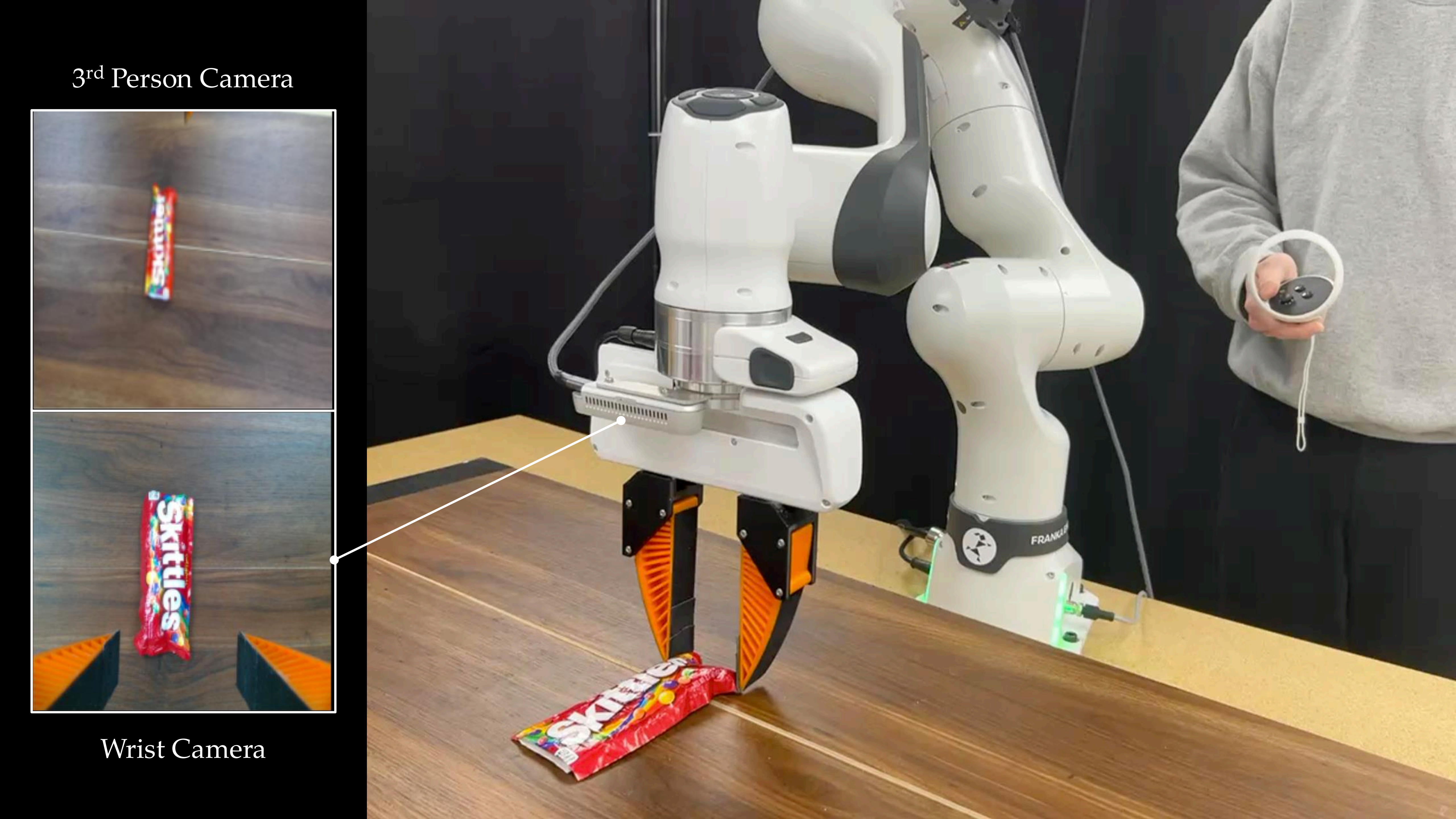
$$V(z) = (1-\gamma)\ell_\theta(z) + \gamma \min\{\ell_\theta(z), \max_{a\in\mathcal{A}} \mathbb{E}_{\hat{z}'\sim p_\phi(\cdot|z,a)}[V(\hat{z}')]\}$$



Related Work:
[Fisac*, Lugovoy* et al. "Bridging Safety Analysis and RL", ICRA 2019]



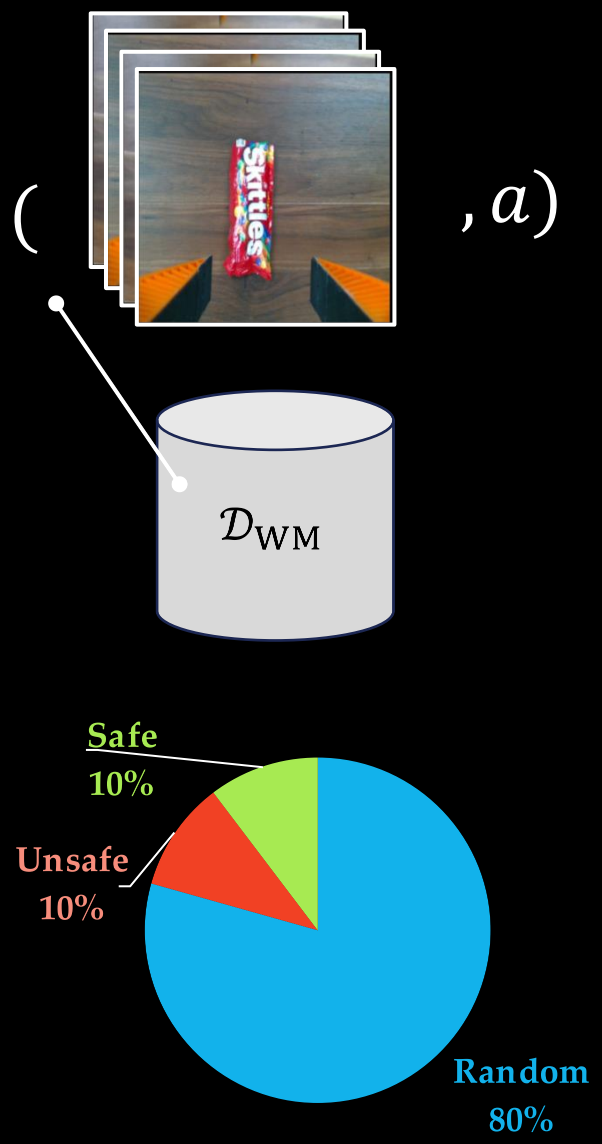
Deployment-time Guardrail
Safety **Monitor** $V^{\shield}(z)$
Safety **Policy** $\pi^{\shield}(z)$
Observation
Safety Filtered
Imagined Failure
π^{task}
z_t
z_0
$\pi^{\shield}$
z_T
$\mathcal{Z}$
World Model


3rd Person Camera
Wrist Camera
Skittles

No Safety Filter
(direct teleop.)

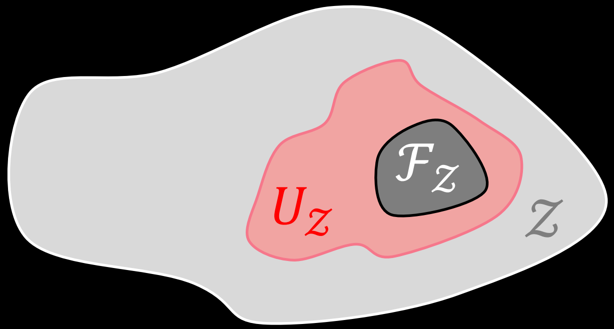
Failure happens (spill)!

World Model Training



$|\mathcal{D}_{\mathrm{WM}}|$ = 1,300

Safety Computation
Time *(Offline)*



[Nakamura, Peters, Bajcsy. *RSS* 2025]

Robot Safeguards a Human Teleoperator

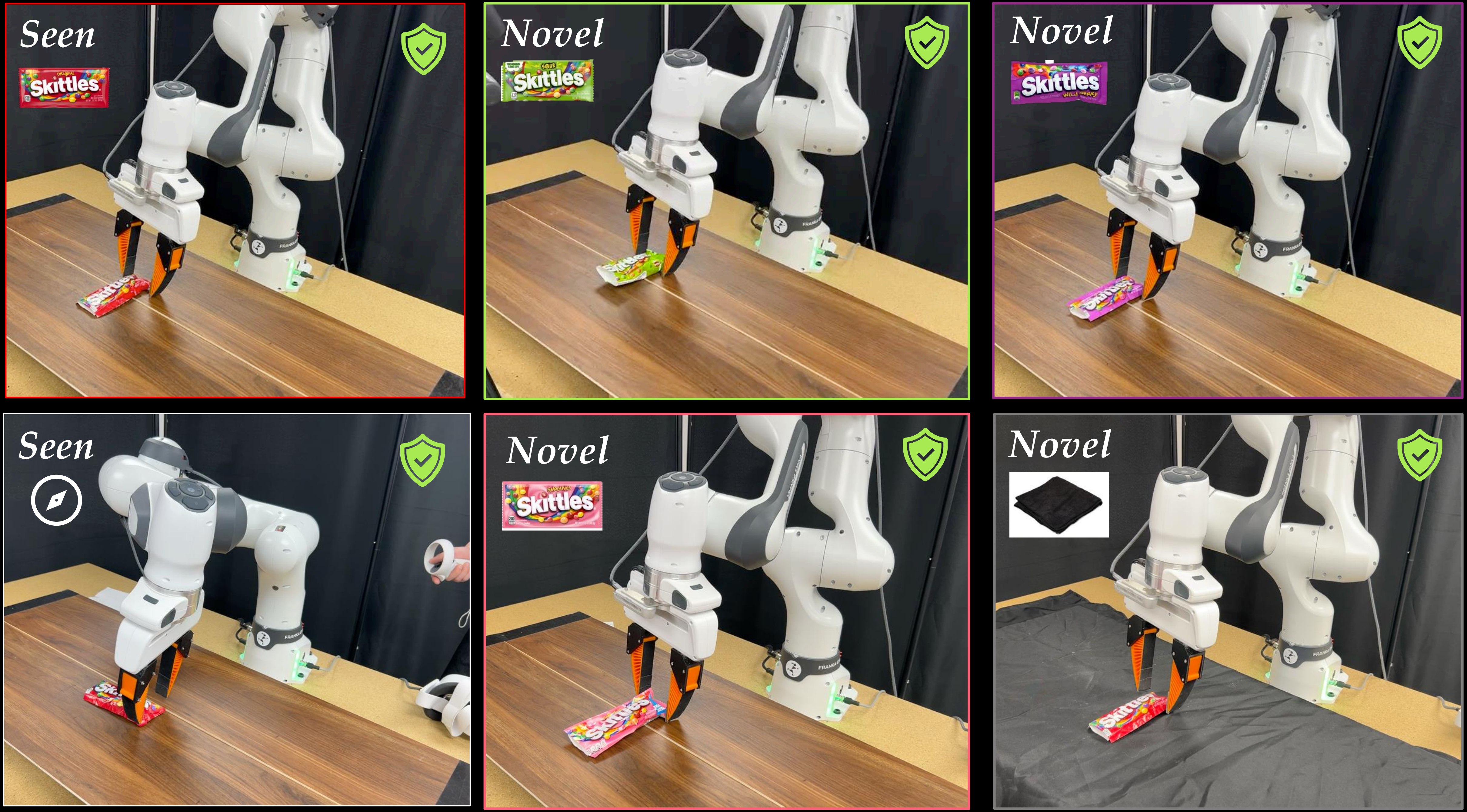
Robot allows a safe grasp… *Sliding motion is* ***filtered to slow*** *…* *But unsafe pickup is* ***filtered to stop*****!**

Green border = *Robot is in control!*

Closer Look: Visualization of Latent Safety Filter ($\pi^{🛡}$, $V^{🛡}$)

Robot POV

$V(z')$


Seen
Skittles
Novel
Skittles
Novel
Skittles
Seen
Novel
Skittles
Novel

Dirty Laundry

Failure – Observability (of bag opening)

Failure – OOD Dynamics (peanut M&Ms)

More sophisticated safety filtering can "remove" unsafe modes but maintain task performance

Base Diffusion Policy

Executes unsafe interaction mode

"Switching" Safety

<u>Stops</u> unsafe lifting motion

Optimization-based Safety

Guides base policy to <u>safer grasp</u>

[Nakamura, Bishop, Man, Johnson, Manchester, Bajcsy. *L4DC* 2026]