COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Google DeepMind

NEURAL INFORMATION PROCESSING SYSTEMS

Counterfactual Debugging the World Model Transfer Gap






Mingxuan Li[1]* **Kai-Zhan Lee[1]*** **Michael Dennis[2]** **Elias Bareinboim[1]**

Columbia University, Causal AI Lab[1], Google DeepMind[2], *Equal Contribution

Speaker: Mingxuan Li

October 2026

Motivations

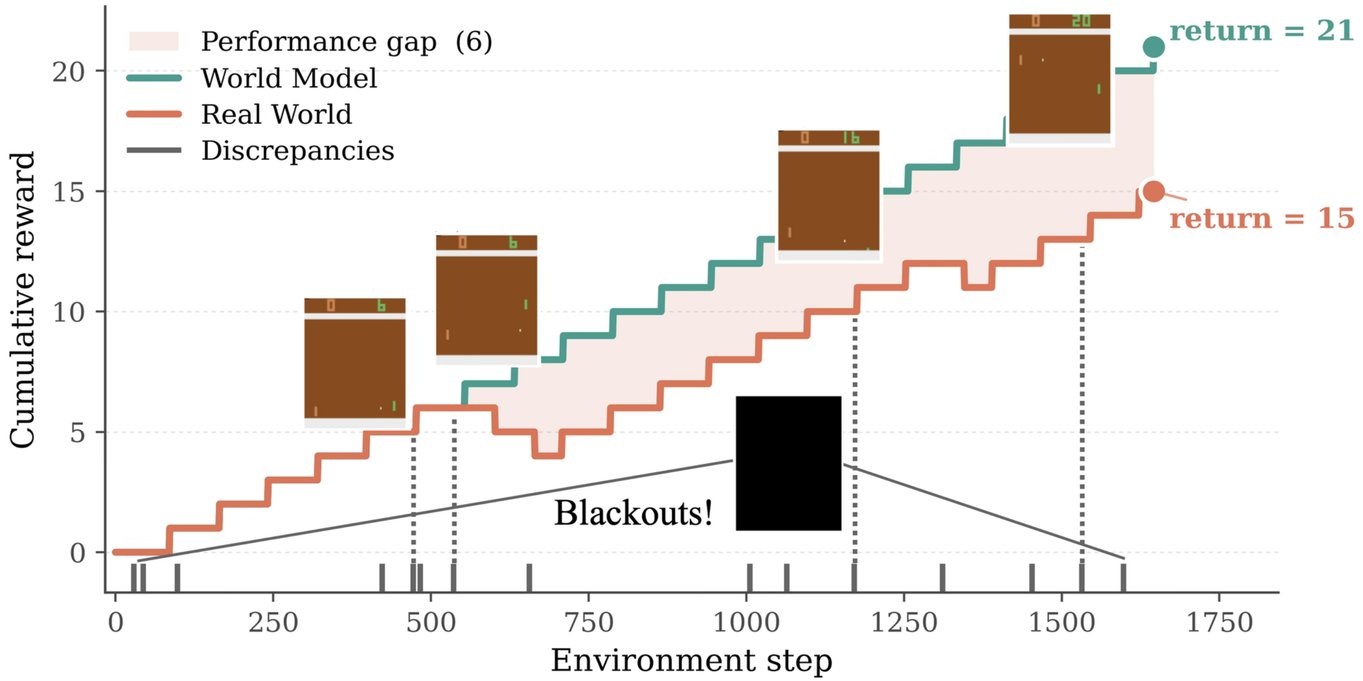
Motivations

The policy trained to optimal in world models may not transfer well to the real world due to sim-to-real gap.

Motivations

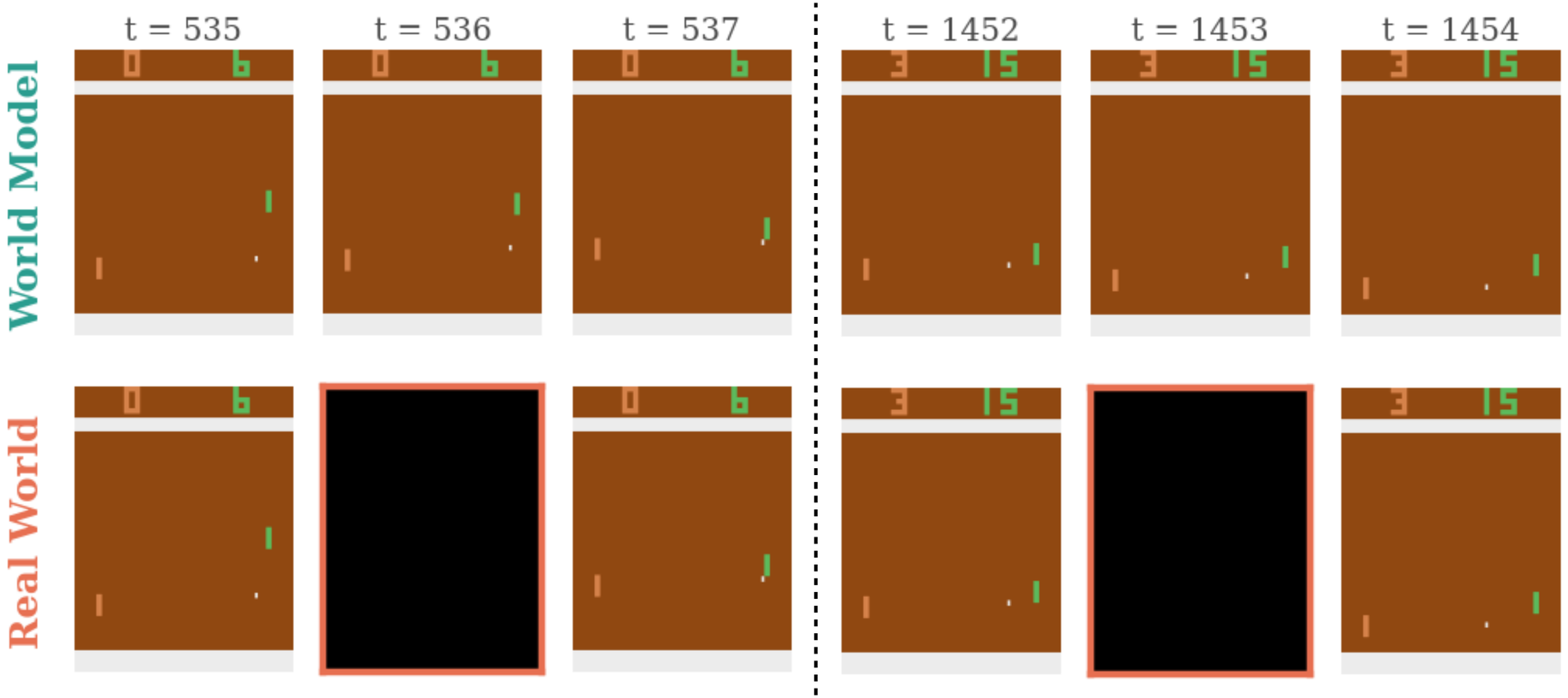
The policy trained to optimal in world models may not transfer well to the real world due to sim-to-real gap.

But do all world model prediction errors causally responsible for the transfer gap?



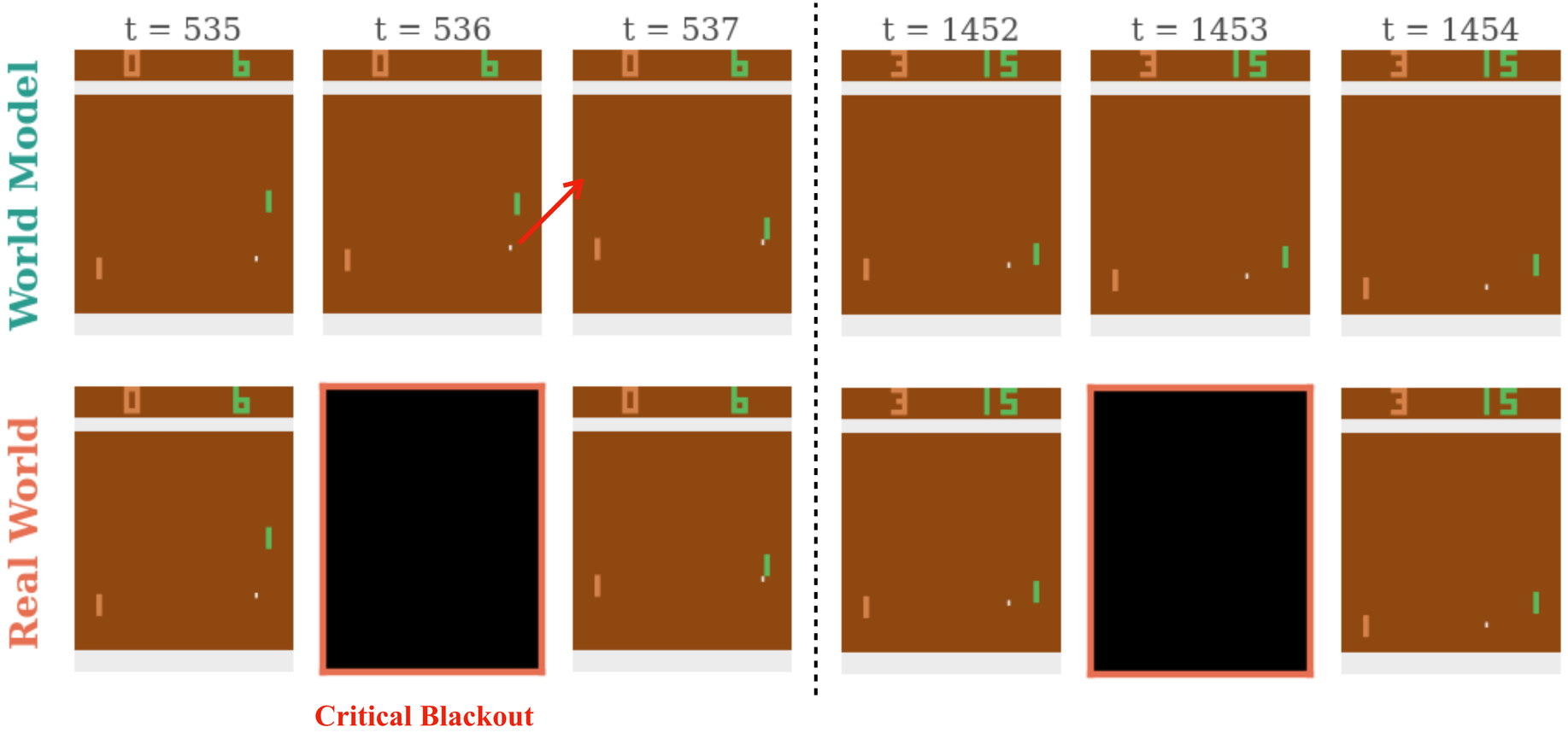
Which blackout should be blamed for the performance degradation? An optimal PPO policy scores 21 in a Pong game world model (green) but only gets 15 in a turbulent "real world Pong" where the screen randomly disconnects (red).

Not Every World-Model Error Matters



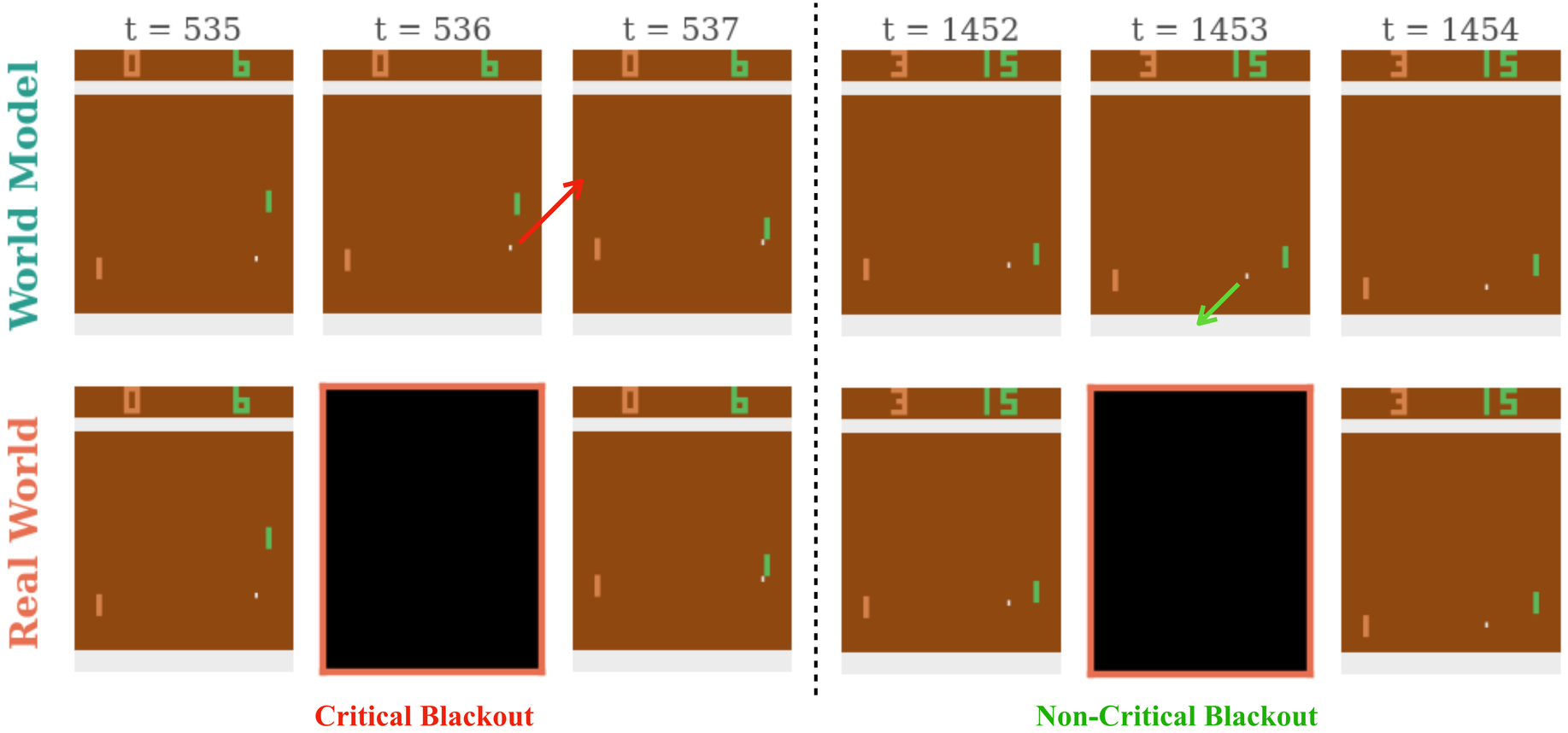
Only blackouts right before the ball hitting agent's paddle matter. In this example, only two of the blackout frames are causally responsible for the score loss that happens more than 50 steps later. **But world model prediction error spikes on all blackouts.**

Not Every World-Model Error Matters



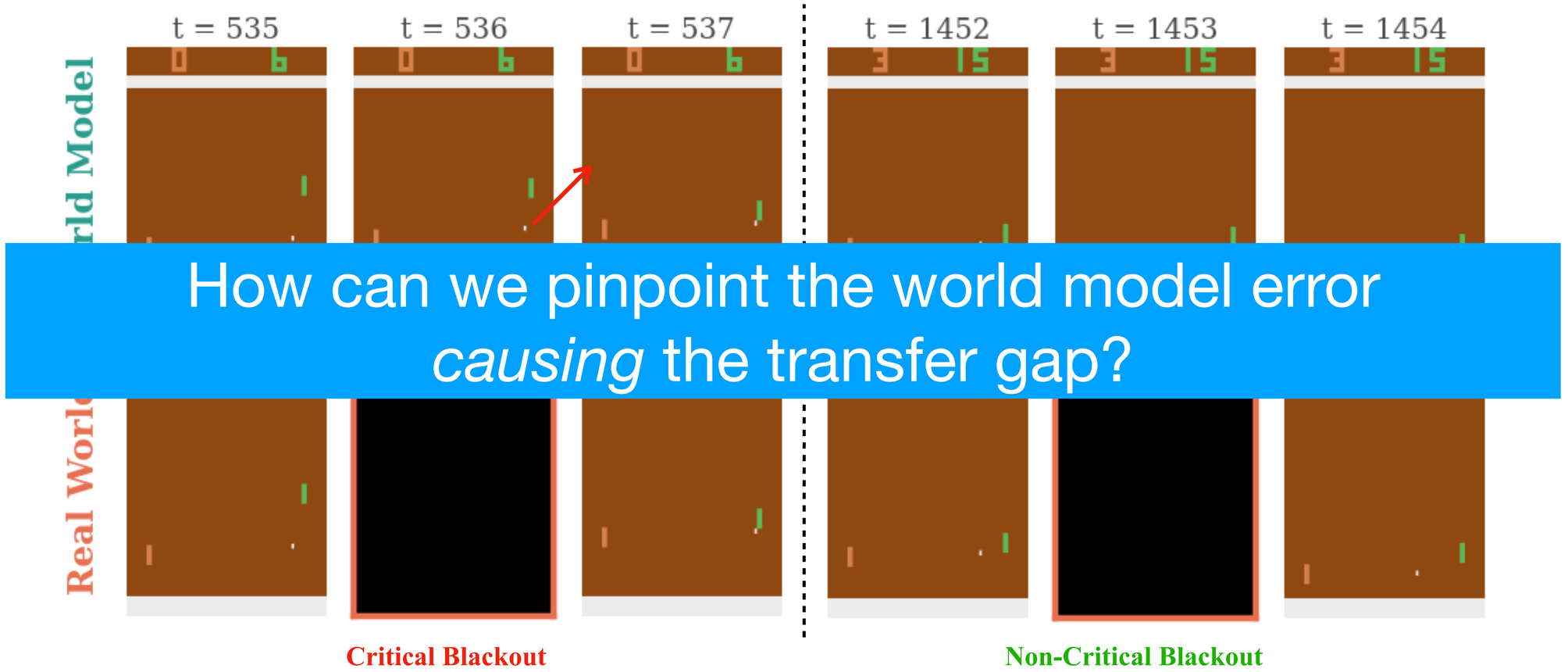
Only blackouts right before the ball hitting agent's paddle matter. In this example, only two of the blackout frames are causally responsible for the score loss that happens more than 50 steps later. **But world model prediction error spikes on all blackouts.**

Not Every World-Model Error Matters



Only blackouts right before the ball hitting agent's paddle matter. In this example, only two of the blackout frames are causally responsible for the score loss that happens more than 50 steps later. **But world model prediction error spikes on all blackouts.**

Not Every World-Model Error Matters



Only blackouts right before the ball hitting agent's paddle matter. In this example, only two of the blackout frames are causally responsible for the score loss that happens more than 50 steps later. **But world model prediction error spikes on all blackouts.**

The World Model Transfer Gap

The World Model Transfer Gap

Let a policy π trained in the world model $\mathscr{M}_{\text{wm}}$ act in the real environment $\mathscr{M}_{\text{env}}$, producing a trajectory τ.

The World Model Transfer Gap

Let a policy π trained in the world model $\mathscr{M}_{\text{wm}}$ act in the real environment $\mathscr{M}_{\text{env}}$, producing a trajectory τ.

The **world-model transfer gap** is the difference between the return it gets and the return the world model promised,

$$\Delta(\mathscr{M}_{\text{env}}, \mathscr{M}_{\text{wm}}) = Y^{\tau}_{\mathscr{M}_{\text{env}}} - \mathbb{E}^{\pi}_{\mathscr{M}_{\text{wm}}}[Y]$$

The World Model Transfer Gap

Let a policy π trained in the world model $\mathscr{M}_{\text{wm}}$ act in the real environment $\mathscr{M}_{\text{env}}$, producing a trajectory τ.

The **world-model transfer gap** is the difference between the return it gets and the return the world model promised,

$$\Delta(\mathscr{M}_{\text{env}}, \mathscr{M}_{\text{wm}}) = Y^{\tau}_{\mathscr{M}_{\text{env}}} - \mathbb{E}^{\pi}_{\mathscr{M}_{\text{wm}}}[Y]$$

$Y^{\tau}_{\mathscr{M}_{\text{env}}}$ is the return of an observed trajectory and $\mathbb{E}^{\pi}_{\mathscr{M}_{\text{wm}}}[Y]$ is the expected reward of π under the world model.

The World Model Transfer Gap

Let a policy π trained in the world model $\mathscr{M}_{\text{wm}}$ act in the real environment $\mathscr{M}_{\text{env}}$, producing a trajectory τ.

The **world-model transfer gap** is the difference between the return it gets and the return the world model promised,

$$\Delta(\mathscr{M}_{\text{env}}, \mathscr{M}_{\text{wm}}) = Y^{\tau}_{\mathscr{M}_{\text{env}}} - \mathbb{E}^{\pi}_{\mathscr{M}_{\text{wm}}}[Y]$$

$Y^{\tau}_{\mathscr{M}_{\text{env}}}$ is the return of an observed trajectory and $\mathbb{E}^{\pi}_{\mathscr{M}_{\text{wm}}}[Y]$ is the expected reward of π under the world model.

In the Pong example, the transfer gap is $15 - 21 = -6$.

Counterfactual Attribution for World Model Transfer Gap

Counterfactual Attribution for World Model Transfer Gap

Intuitively, a world model prediction error is causally responsible for the transfer gap only if the gap would be smaller had that step been from the real world.

Counterfactual Attribution for World Model Transfer Gap

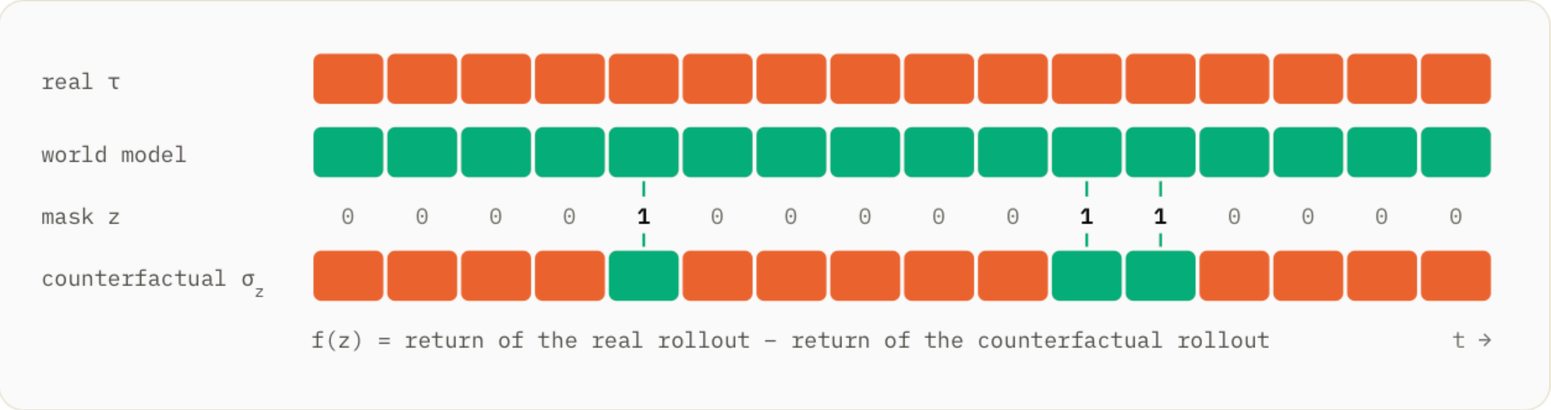
Intuitively, a world model prediction error is causally responsible for the transfer gap only if the gap would be smaller had that step been from the real world.

We use *Counterfactual Shapley Values* (ϕ-values; Lee et al., 2025) to attribute the transfer gap to each time step.

Counterfactual Attribution for World Model Transfer Gap

Intuitively, a world model prediction error is causally responsible for the transfer gap only if the gap would be smaller had that step been from the real world.

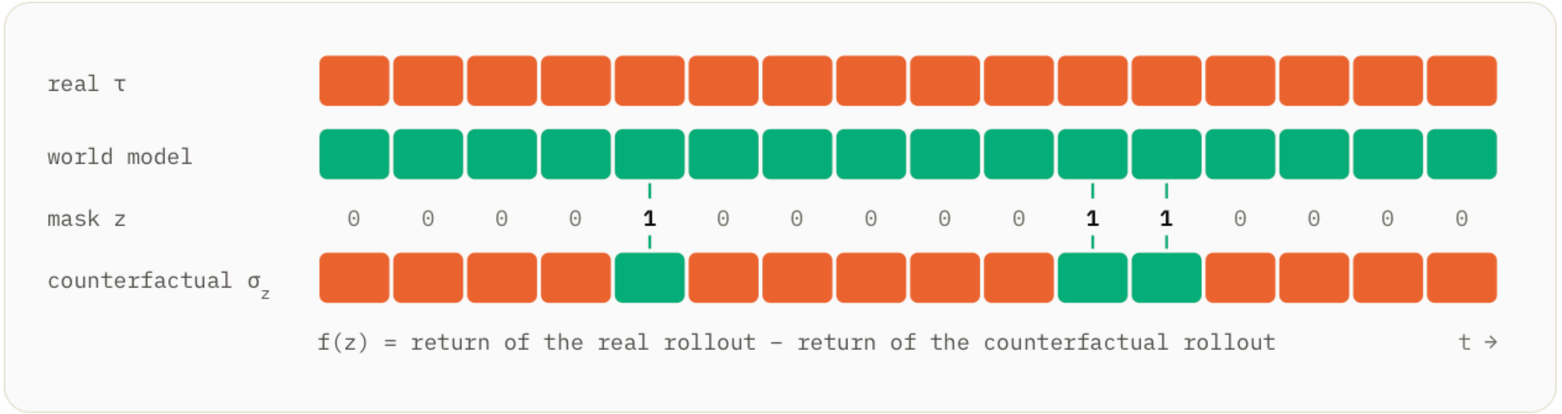
We use *Counterfactual Shapley Values* (ϕ-values; Lee et al., 2025) to attribute the transfer gap to each time step.



Counterfactual Attribution for World Model Transfer Gap

Intuitively, a world model prediction error is causally responsible for the transfer gap only if the gap would be smaller had that step been from the real world.

We use *Counterfactual Shapley Values* (ϕ-values; Lee et al., 2025) to attribute the transfer gap to each time step.

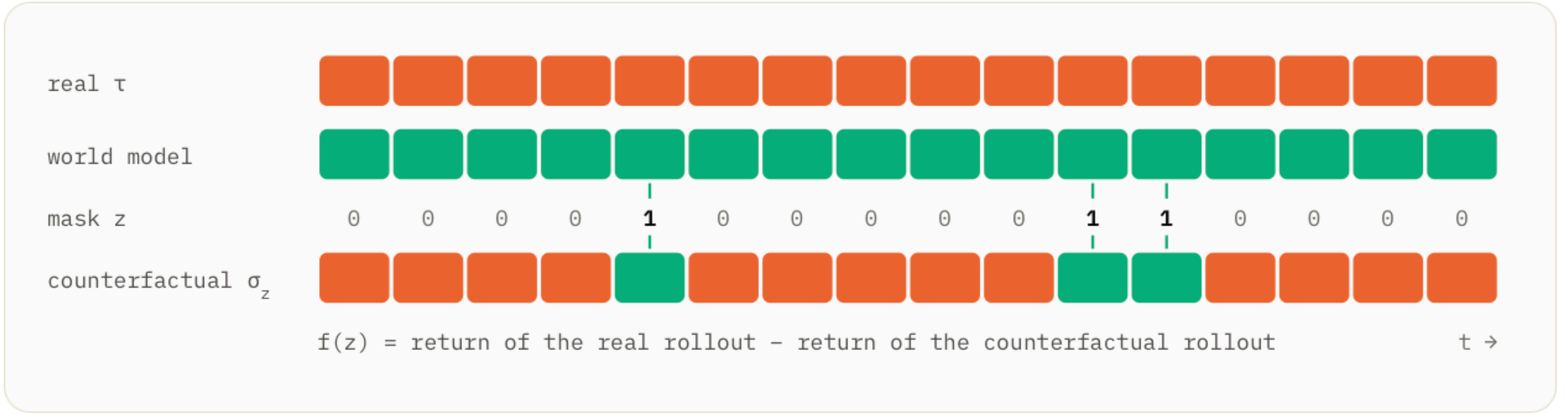


Counterfactual Simulation (Algo. 2)

Counterfactual Attribution for World Model Transfer Gap

Intuitively, a world model prediction error is causally responsible for the transfer gap only if the gap would be smaller had that step been from the real world.

We use *Counterfactual Shapley Values* (ϕ-values; Lee et al., 2025) to attribute the transfer gap to each time step.

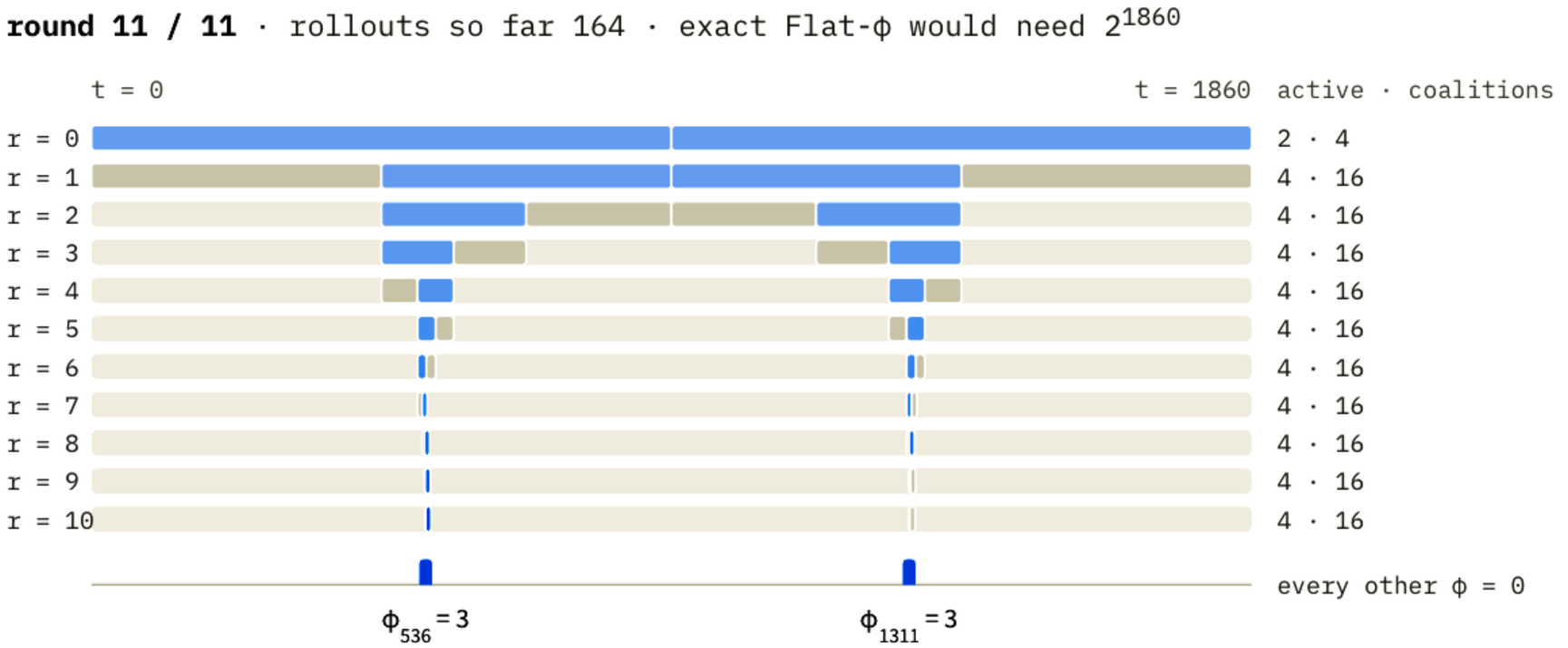


Counterfactual Simulation (Algo. 2)

$$\sum_t \phi_t = \sum_t \sum_{\mathbf{z}\in\{0,1\}^T} \kappa_t(\mathbf{z}) f(\mathbf{z}) = \Delta(\mathscr{M}_{\text{env}}, \mathscr{M}_{\text{wm}})$$

Scaling to 1M Steps with Recursive-ϕ

Directly calculating ϕ-values is expensive. We propose to use a coarse-to-fine recursive scheme, Recursive-ϕ, to speed up the computation. Below is from the Pong example.



Recursive-ϕ on the Pong example. Each row represents a round. Blue blocks are still active, with shade proportional to their per-step block ϕ-values. Grey blocks were pruned, and every step in them gets $\phi_t = 0$. The coalition count per round is merely 2^n rollouts for n active blocks when enumerated exactly.

Experiment Design

Experiment Design

We test our Recursive-ϕ on,

- Learned neural world models

Experiment Design

We test our Recursive-ϕ on,

- Learned neural world models

  (1) SuperMimic, 2026 (humanoid)

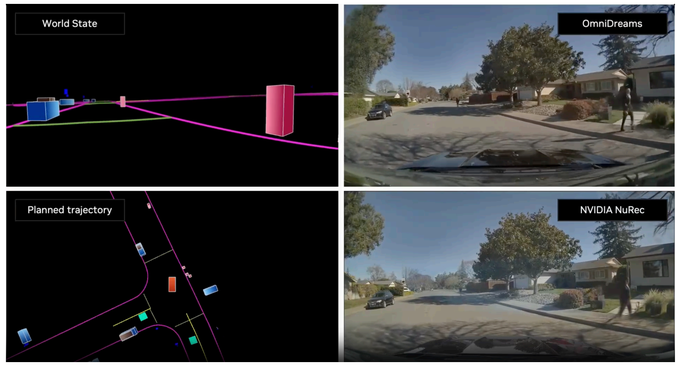


Experiment Design

We test our Recursive-ϕ on,

- Learned neural world models
  (1) SuperMimic, 2026 (humanoid)
  (2) OmniDreams, 2026 (autonomous driving)



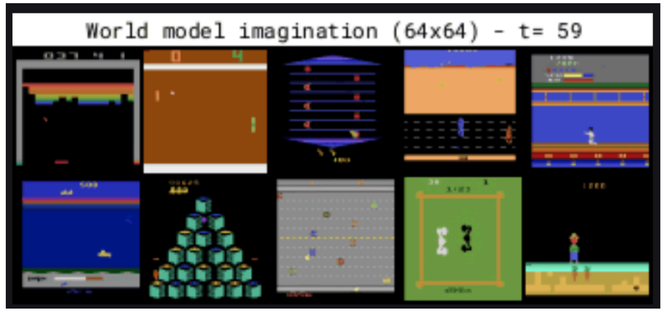


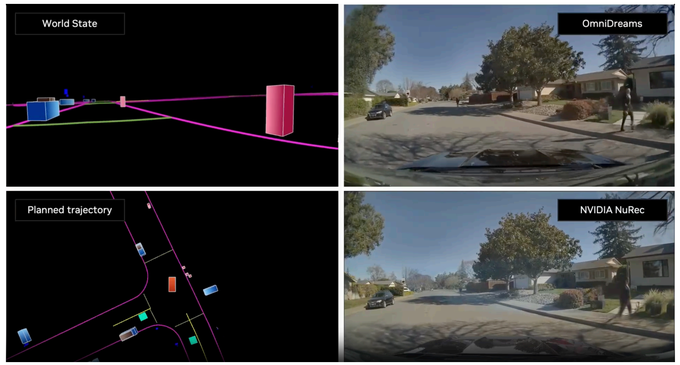
Experiment Design

We test our Recursive-ϕ on,

- Learned neural world models
  (1) SuperMimic, 2026 (humanoid)
  (2) OmniDreams, 2026 (autonomous driving)
  (3) DIAMOND, 2024 (atari games)



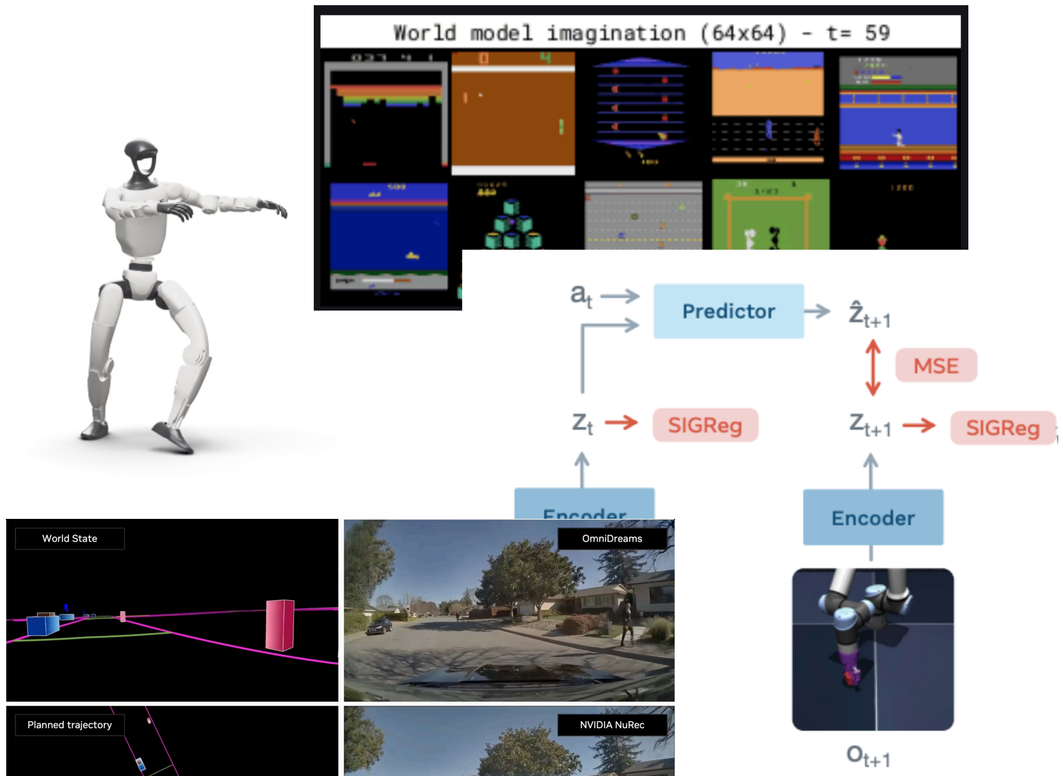




Experiment Design

We test our Recursive-ϕ on,

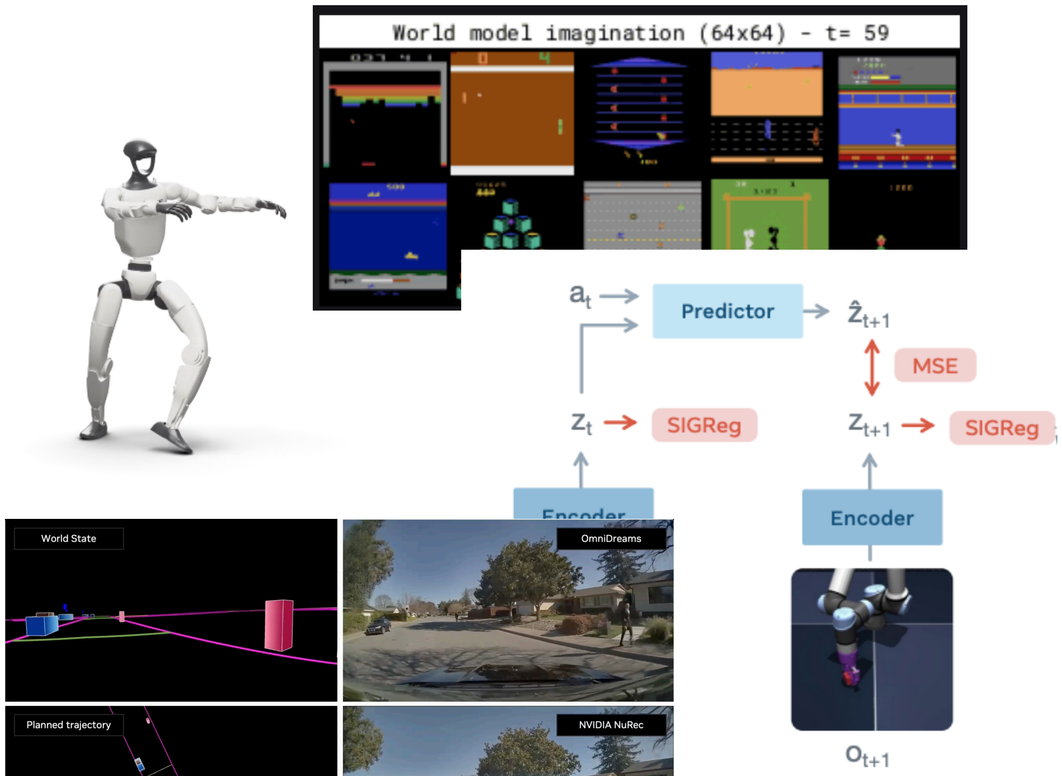
- Learned neural world models
  - (1) SuperMimic, 2026 (humanoid)
  - (2) OmniDreams, 2026 (autonomous driving)
  - (3) DIAMOND, 2024 (atari games)
  - (4) LeWorldModel, 2026 (manipulation)



Experiment Design

We test our Recursive-ϕ on,

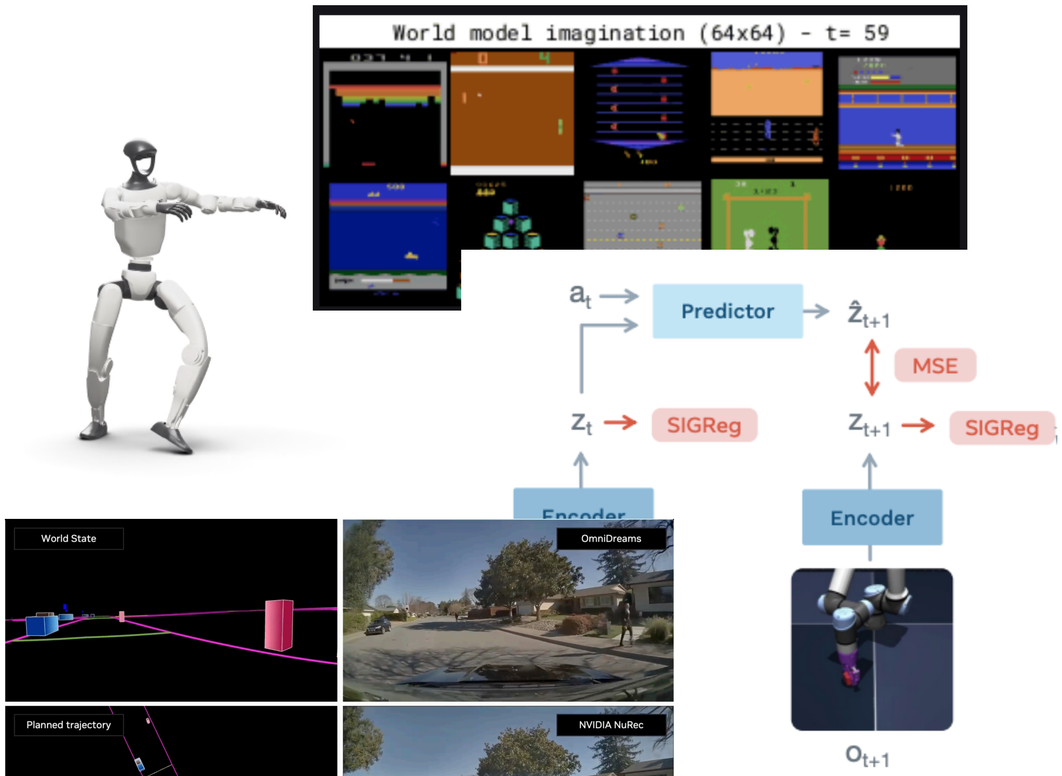
- Learned neural world models
  - (1) SuperMimic, 2026 (humanoid)
  - (2) OmniDreams, 2026 (autonomous driving)
  - (3) DIAMOND, 2024 (atari games)
  - (4) LeWorldModel, 2026 (manipulation)
- Simulator with controlled errors



Experiment Design

We test our Recursive-ϕ on,

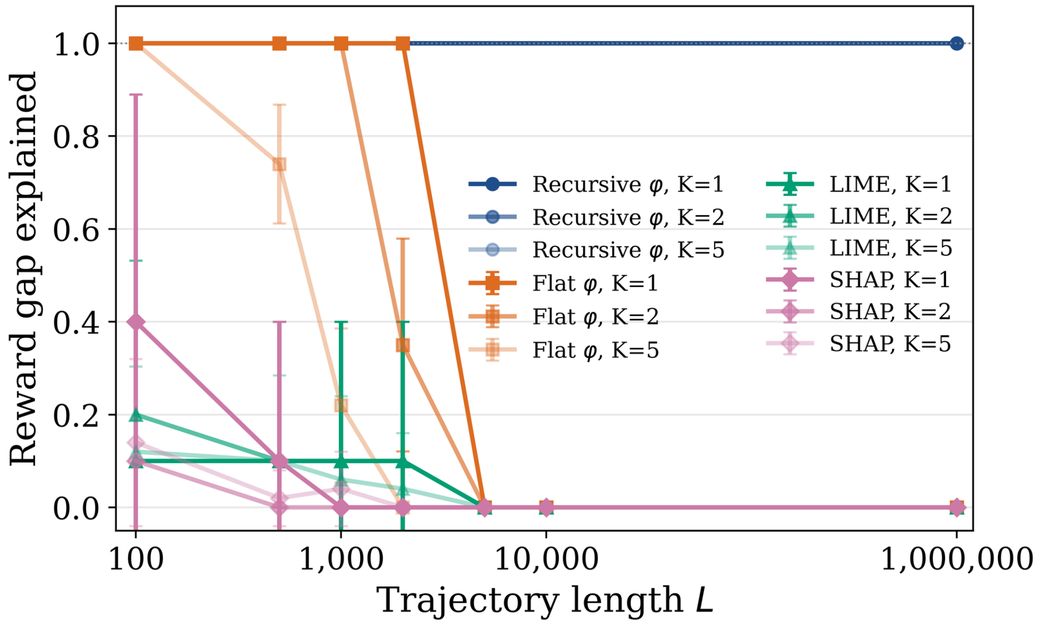
- Learned neural world models
  (1) SuperMimic, 2026 (humanoid)
  (2) OmniDreams, 2026 (autonomous driving)
  (3) DIAMOND, 2024 (atari games)
  (4) LeWorldModel, 2026 (manipulation)
- Simulator with controlled errors

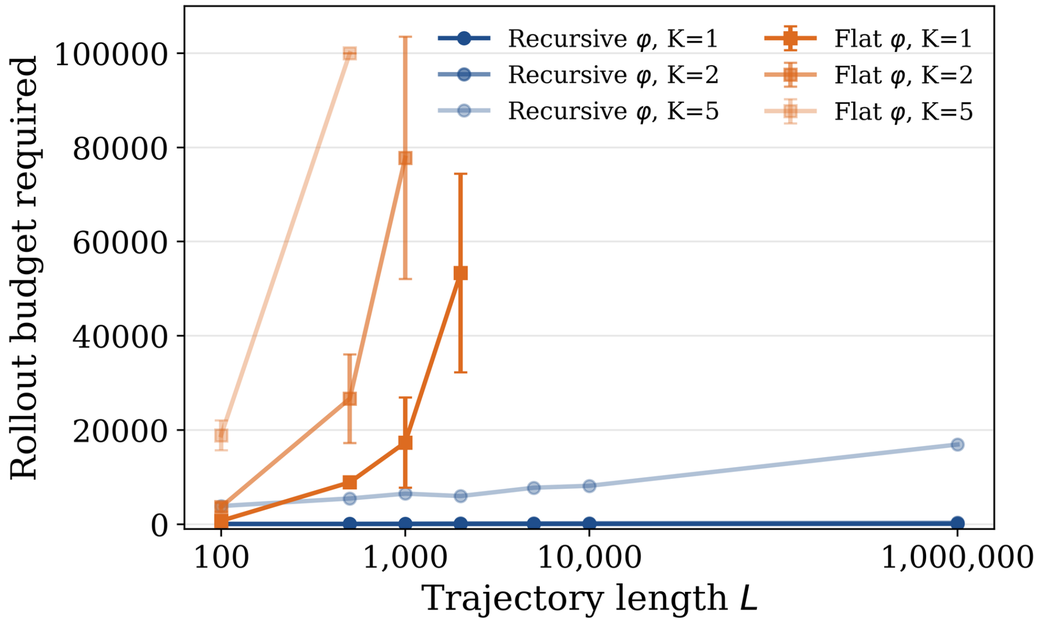


Our goal is to show that it can detect common world model failures including *observation corruption*, *reward misspecification*, and *dynamics shift* across a variety of applications.

Experiment Results

We test Recursive-ϕ in a grid world with injected step penalties with varying episode length. So each visit to a penalized state contributes a share to the transfer gap.

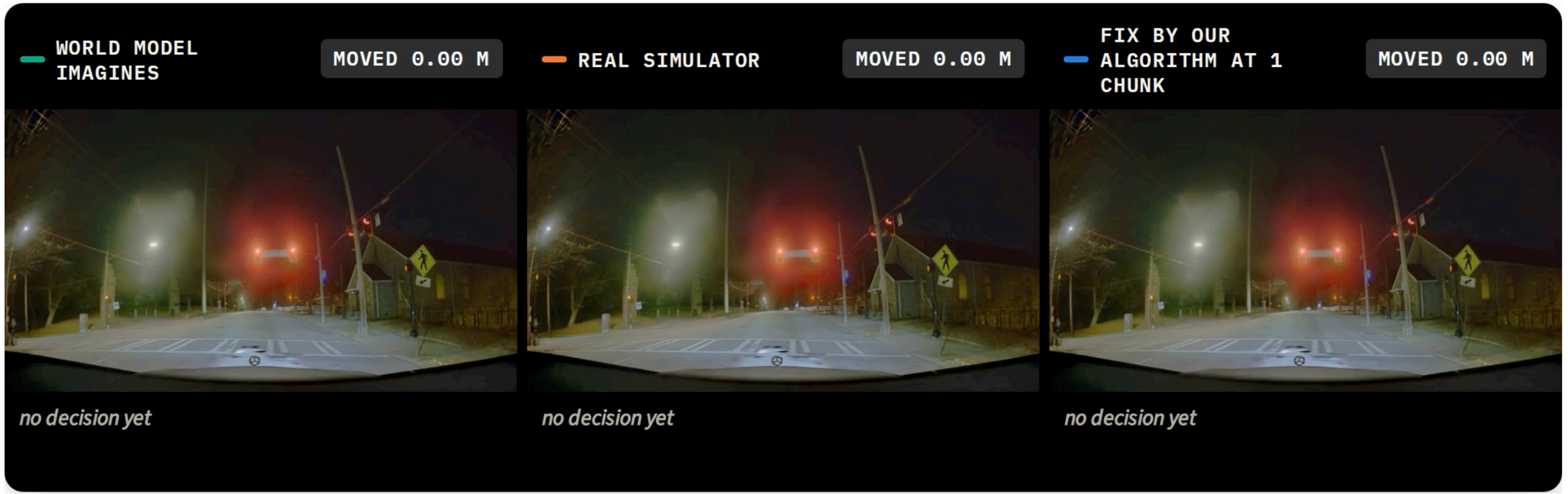




Scaling to 1M steps. Left: only Recursive-ϕ pinpoints all the errors. Right: the recursive design computes ϕ-values with sub-linear sample complexity.

Case Study - OmniDreams

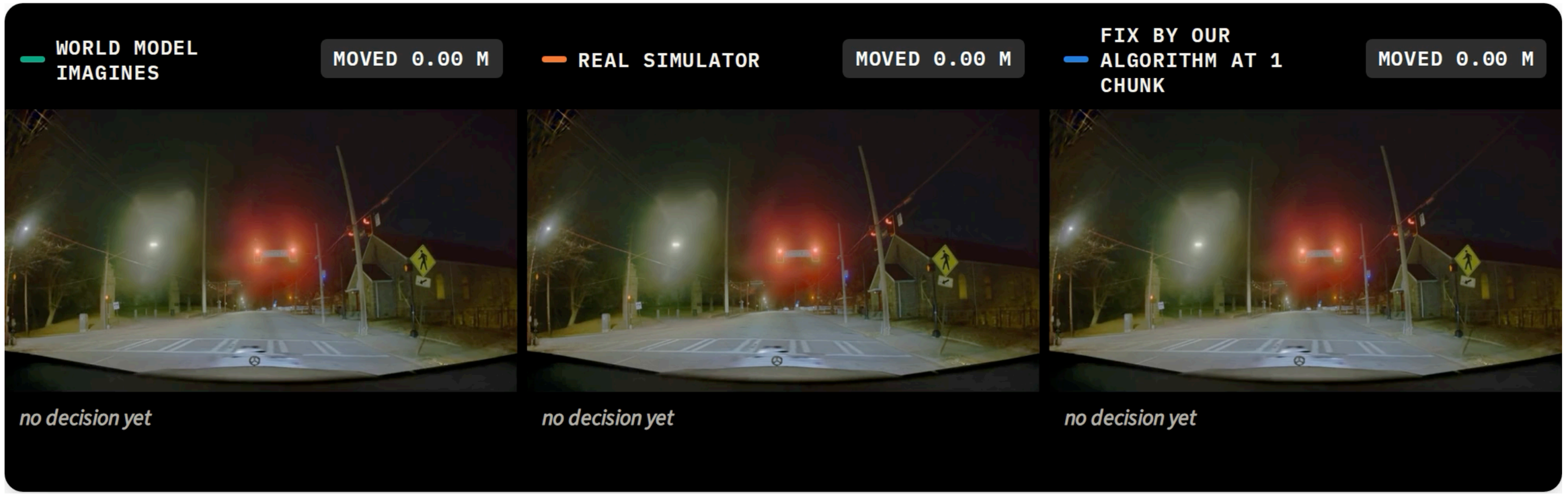
Below we show a clip where the world model thinks the light turns green early, leading to unsafe driving decisions.



OmniDreams hallucinates an early green light. Due to the wrong green light, Alpamayo 1.5, a 10B VLA, accelerates early towards the junction while in reality the driver waited another 2s before moving.

Case Study - OmniDreams

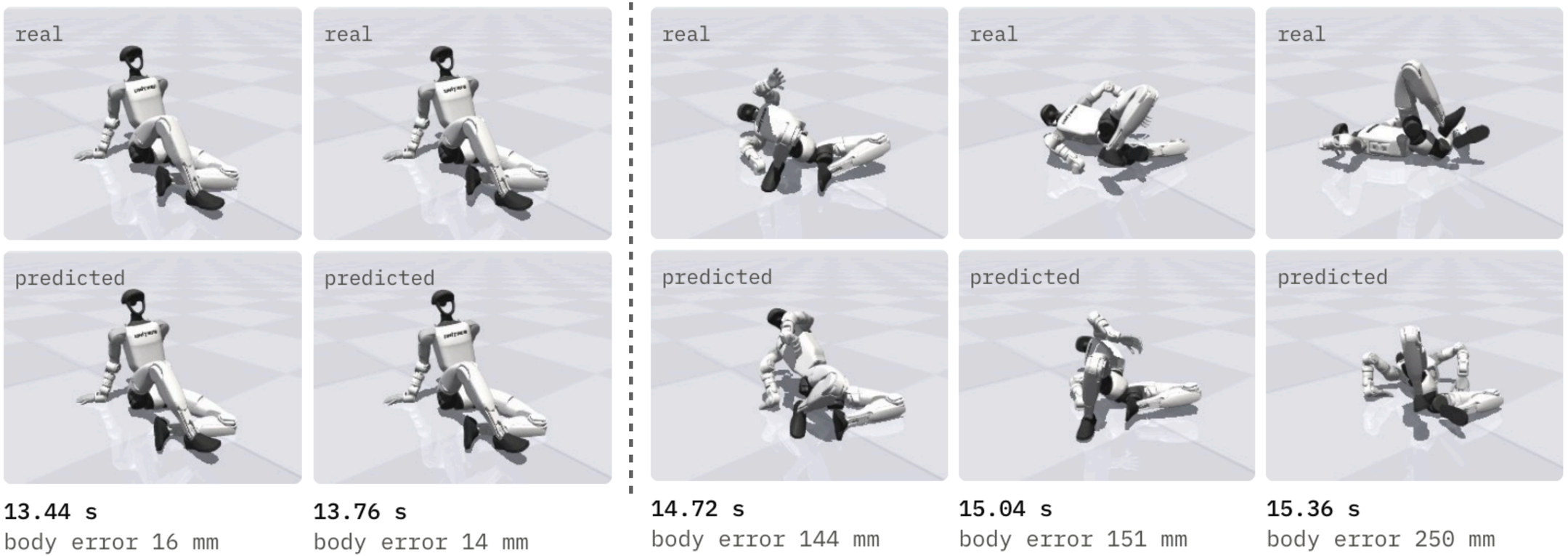
Below we show a clip where the world model thinks the light turns green early, leading to unsafe driving decisions.



OmniDreams hallucinates an early green light. Due to the wrong green light, Alpamayo 1.5, a 10B VLA, accelerates early towards the junction while in reality the driver waited another 2s before moving.

Case Study - SuperMimic

On the humanoid policy trained by SuperMimic, the world model imagines the robot stands up while it falls in real.



The frames that look most wrong are not the cause for transfer gap. The world model trained with SuperMimic thinks the humanoid stands up while it falls in reality. Left: the root cause frames detected by Recursive-ϕ; Right: frames picked by top prediction error.

Takeaway

Takeaway

1. Not all world model errors matter equally, and telling the critical few from the irrelevant many takes **causal reasoning** instead of pixel or latent errors.

Takeaway

1. Not all world model errors matter equally, and telling the critical few from the irrelevant many takes **causal reasoning** instead of pixel or latent errors.

2. **Counterfactual debugging** pinpoints those critical errors causing the transfer gap, providing principled, actionable insights for for model-based policy learning.

Takeaway

1. Not all world model errors matter equally, and telling the critical few from the irrelevant many takes **causal reasoning** instead of pixel or latent errors.

2. **Counterfactual debugging** pinpoints those critical errors causing the transfer gap, providing principled, actionable insights for for model-based policy learning.

Project Page

Takeaway

1. Not all world model errors matter equally, and telling the critical few from the irrelevant many takes **causal reasoning** instead of pixel or latent errors.

2. **Counterfactual debugging** pinpoints those critical errors causing the transfer gap, providing principled, actionable insights for for model-based policy learning.

Project Page

Thank you!