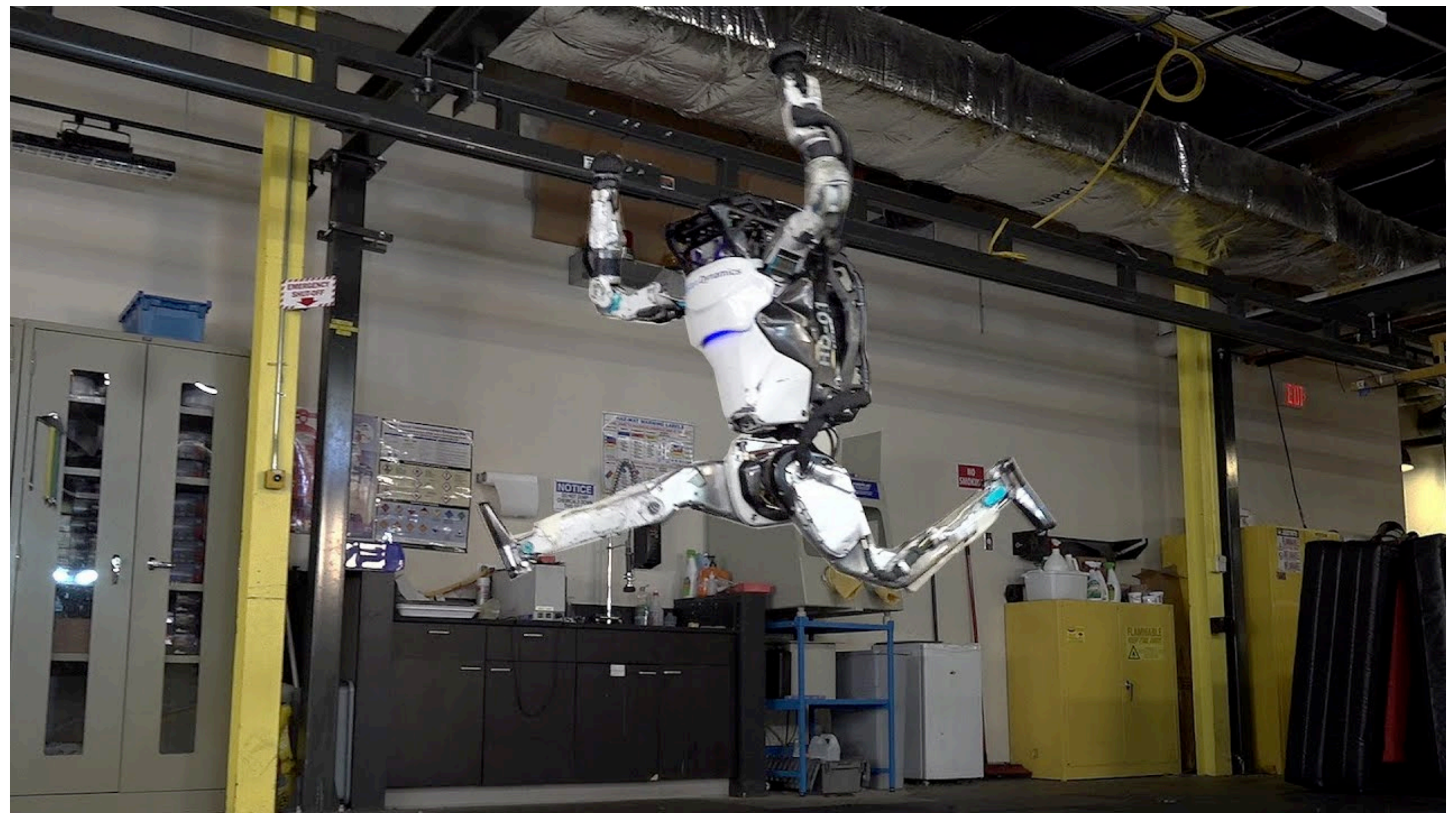
PWM: Policy Learning with Large World Models

Ignat Georgiev, Varun Giridhar, Nicklas Hansen, Animesh Garg

Motivation



Agile locomotion

Dexterous Manipulation

Common failure: contact-rich tasks!

Whole-Body Loco-Manipulation

The current state of robot learning

- **Reinforcement Learning (RL)**
- Formulated as MDP reward maximization
- Notoriously sample-inefficient
- Mostly simulation based
- Can do any single task given good simulation and data
- $\max_{\theta} \mathbb{E}_{\substack{s_1 \sim \rho \\ a_h \sim \pi(\cdot \mid s_h)}} \left[\sum_{h=1}^{H} r(s_h, a_h) \right]$

- **Behavior cloning (BC)**
- Formulated as supervised learning
- Very capable given optimal data
- Impressive multitasking
- Currently mostly simple tasks
- Lives and dies by its data
- $\min_{\theta} \mathbb{E}_{\hat{a}_{h:h+k} \sim \pi(\cdot|s_h)} \|\hat{a}_{h:h+k} - a_{h:h+k}\|_2^2$

Taking a page from deep learning

- Large models
- Large data
- Efficient optimization - SGD

The current state of robot learning

- Large models ✅
- Large data ✅
- Efficient optimization - SGD ✅

- **Behavior cloning (BC)**
- Formulated as supervised learning
- Very capable given optimal data
- Impressive multitasking
- Currently mostly simple tasks
- Lives and dies by its data
- $\min_{\theta} \mathbb{E}_{\hat{a}_{h:h+k} \sim \pi(\cdot|s_h)} ||\hat{a}_{h:h+k} - a_{h:h+k}||_2^2$

The current state of robot learning

- **Reinforcement Learning (RL)**
- Formulated as MDP reward maximization
- Notoriously sample-inefficient
- Mostly simulation based
- Can do any single task given good simulation and data

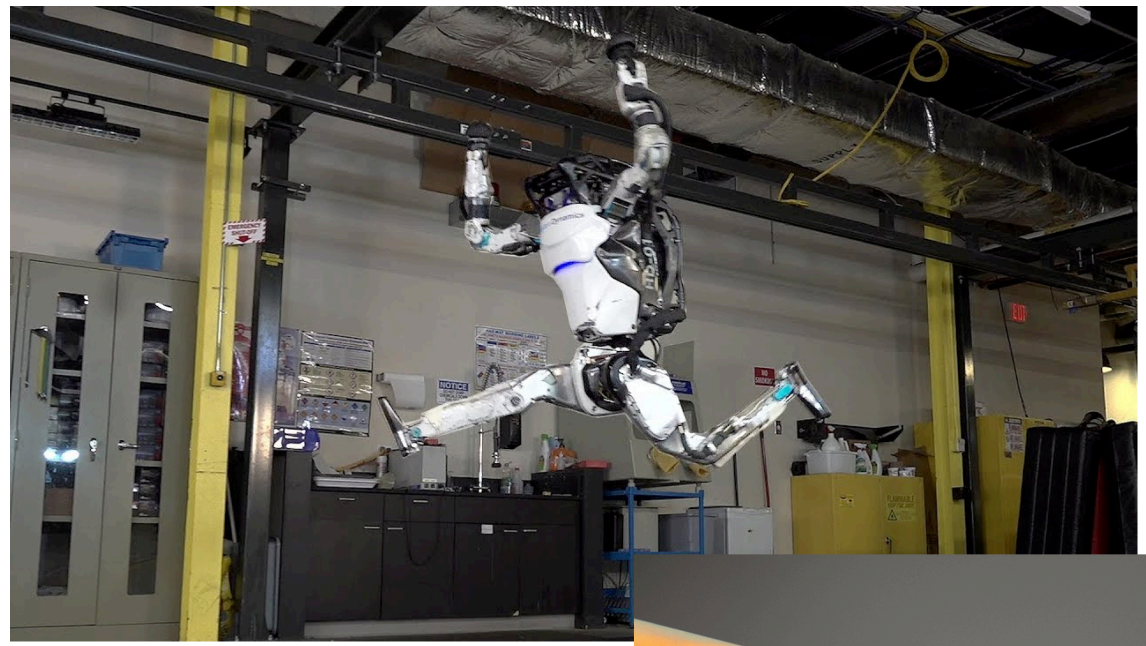
- Large models ✅
- Large data ✅
- Efficient optimization - SGD ❌

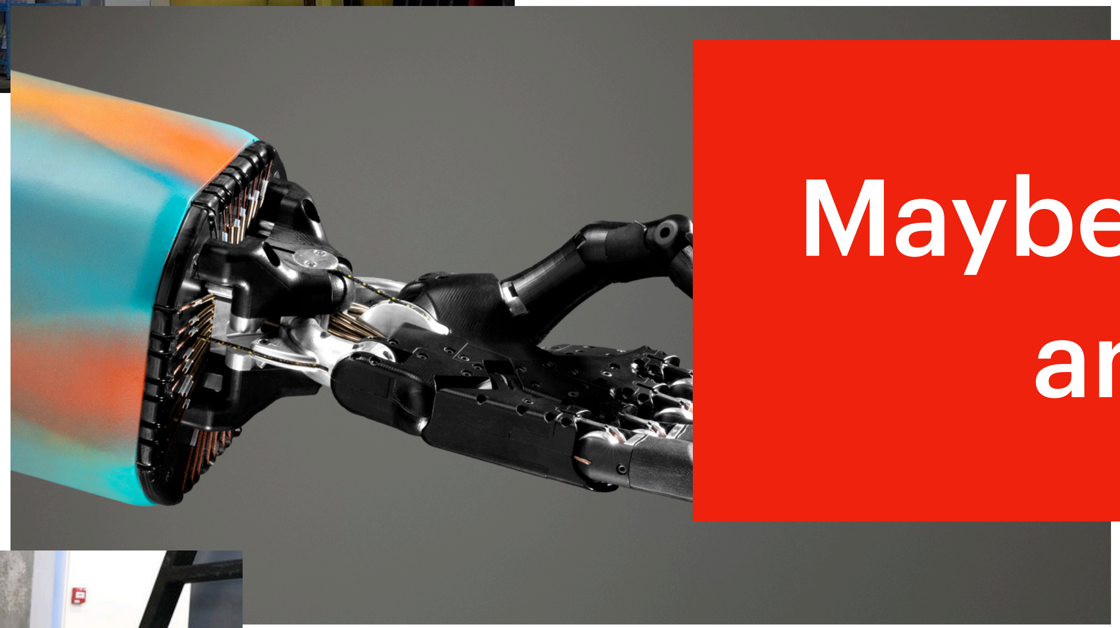
Most common: policy gradients (ZoG)

$$\nabla_\theta^{[0]} J(\theta) := \mathbb{E}_{a_h \sim \pi_\theta(\cdot|s_h)} \left[\left(\sum_{h=1}^{H} r(s_h, a_h) \right) \left(\sum_{h=1}^{H} \nabla_\theta \log \pi_\theta(a_h \mid s_h) \right) \right]$$

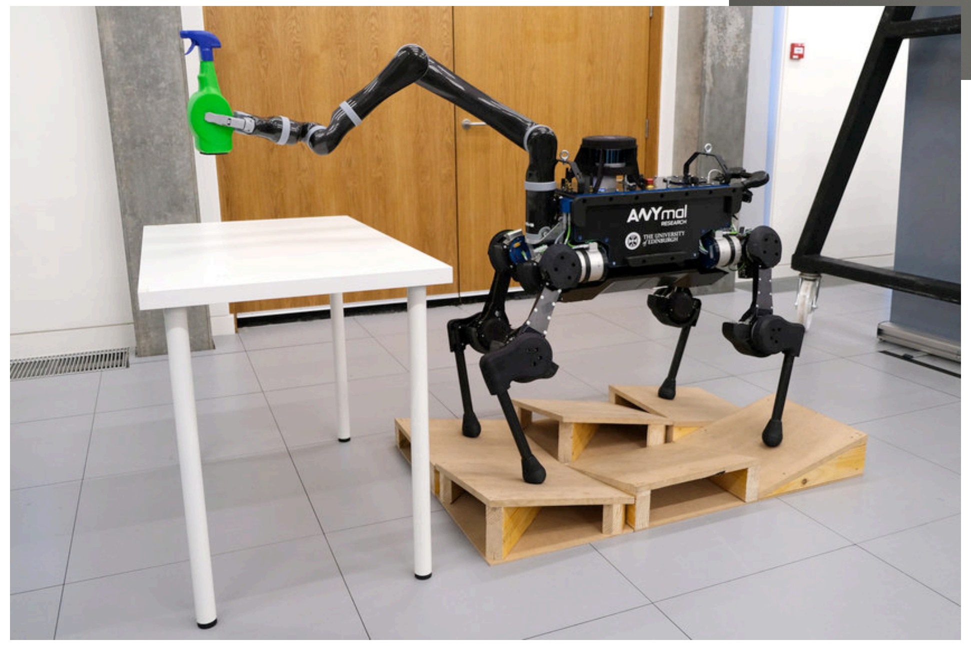
What are the current possibilities

- RL



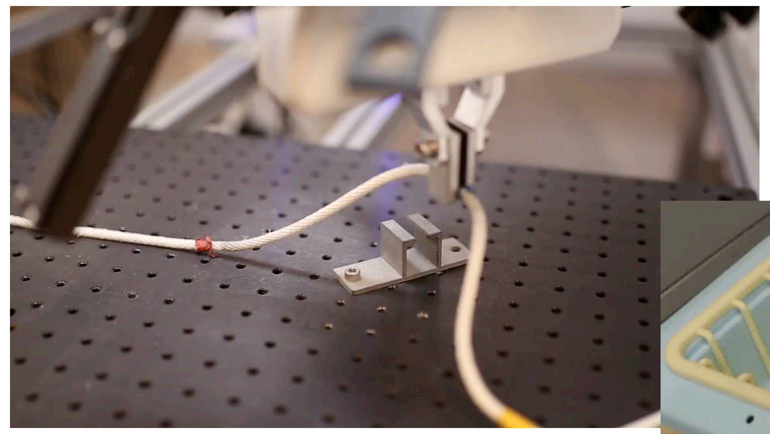


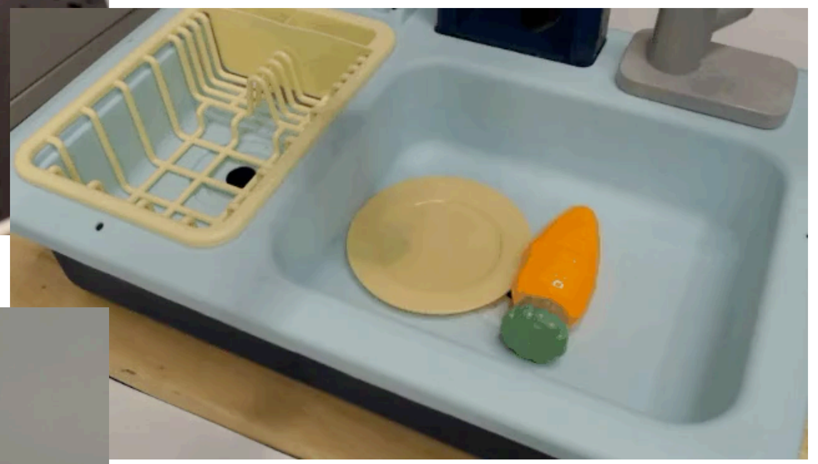
Maybe SGD is the answer?



- BC

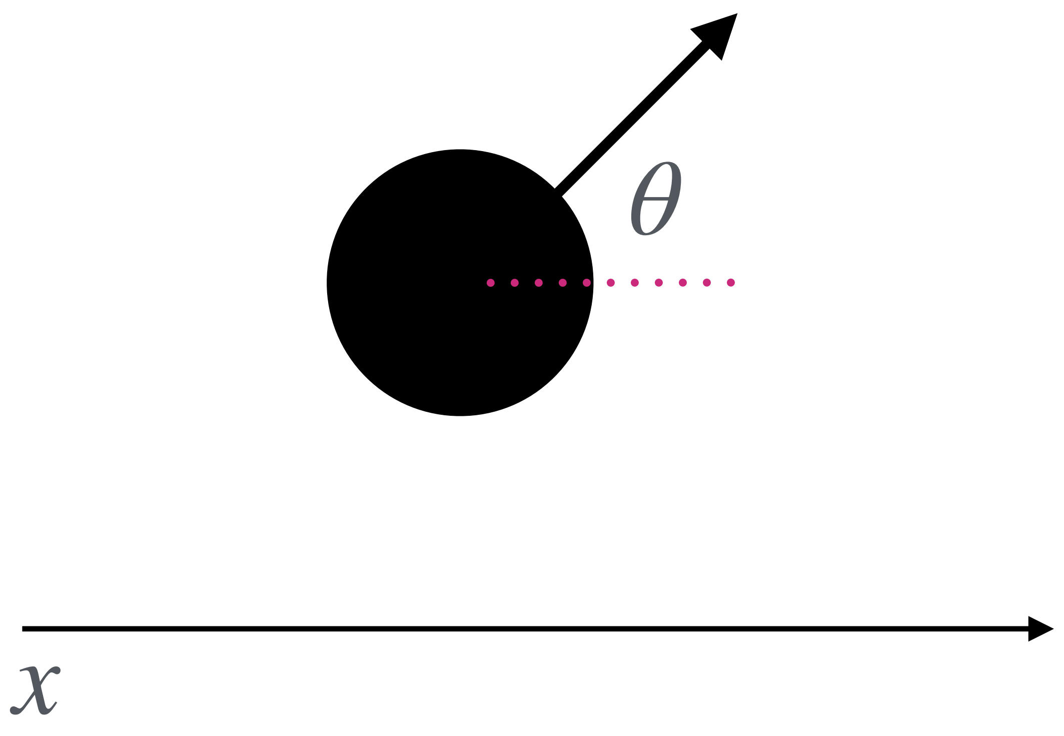
RT-X trained models are trained on 160k tasks





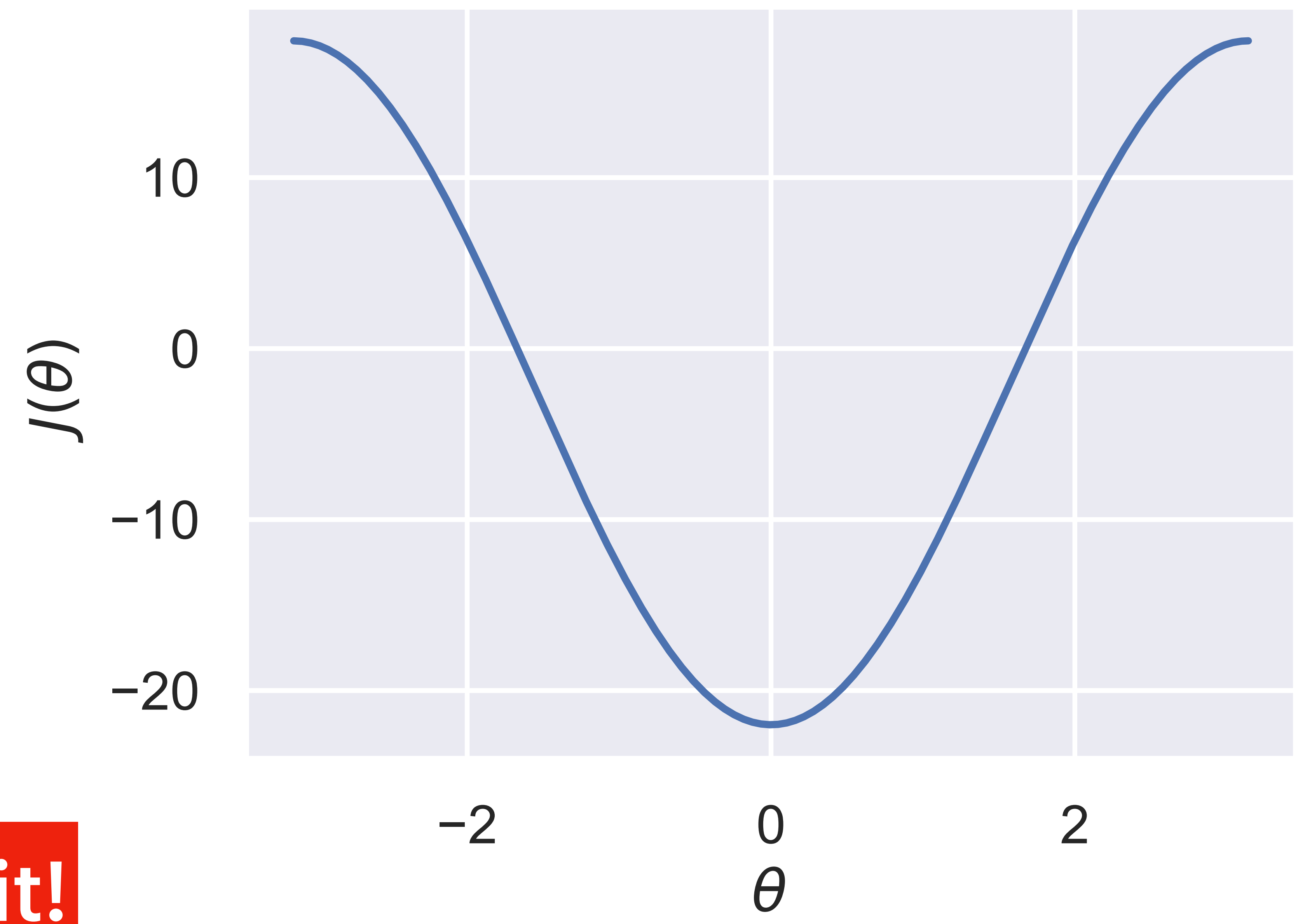
Why not SGD? Optimizing through contact

- What is so difficult?
- Let's optimize a simple task of a point mass thrown in free space for t=2s



$$J(\theta) = x_t = x_0 + v_x t + \frac{1}{2} a_x t^2$$

$$v_0 = 10\ m/s \quad g = 10\ m^2/s$$

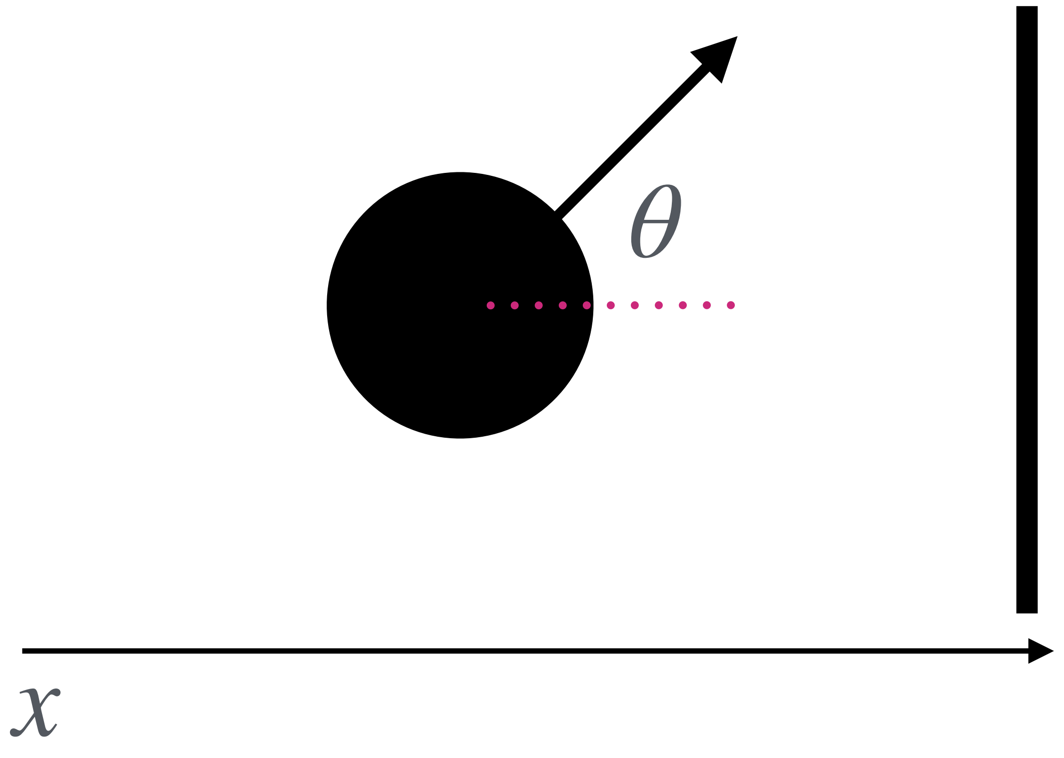


We can just throw GD at it!

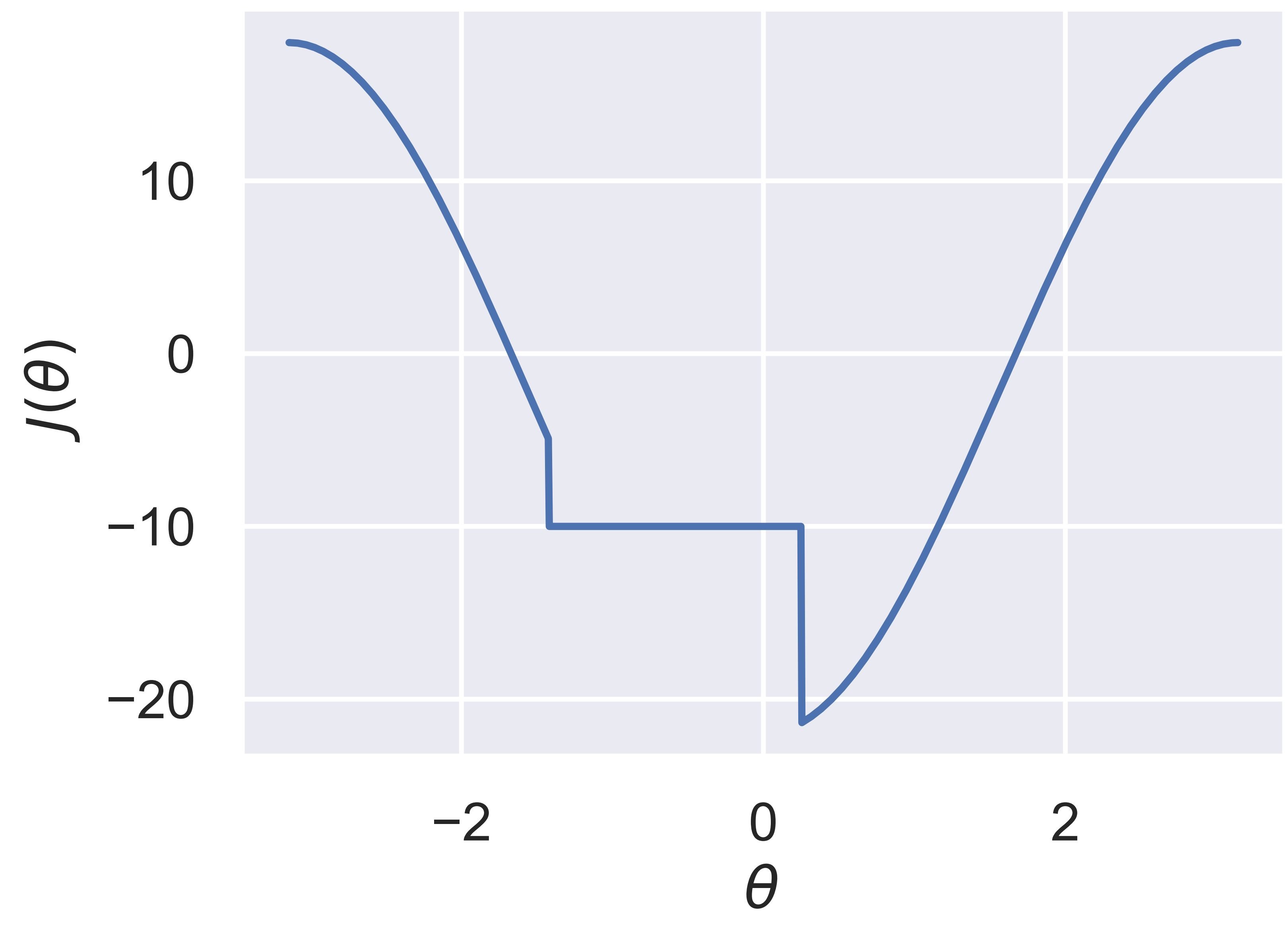
Optimizing through contact

- Now add a wall! Assume that the ball sticks to it.

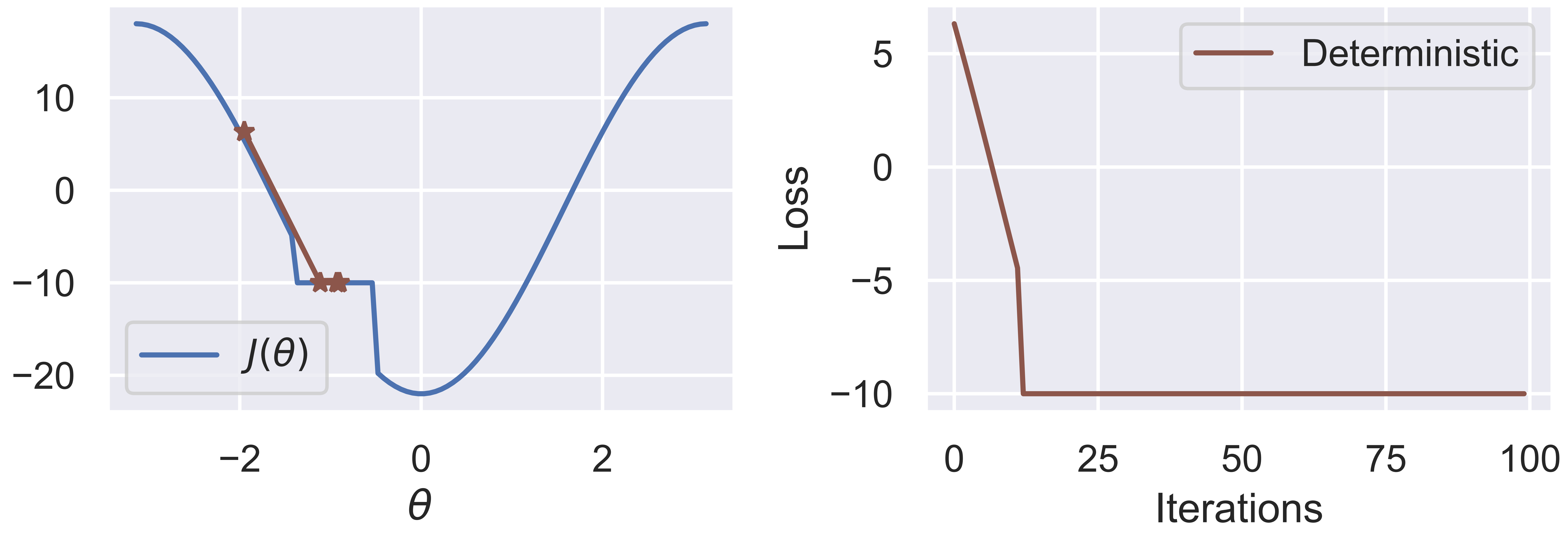
$$J(\theta) = \begin{cases} x_0 + v\cos(\theta)t + \frac{1}{2}at^2 & \text{if no contact} \\ w & \text{otherwise} \end{cases}$$



$$v_0 = 10\ m/s \quad g = 10\ m^2/s \quad w = 10$$



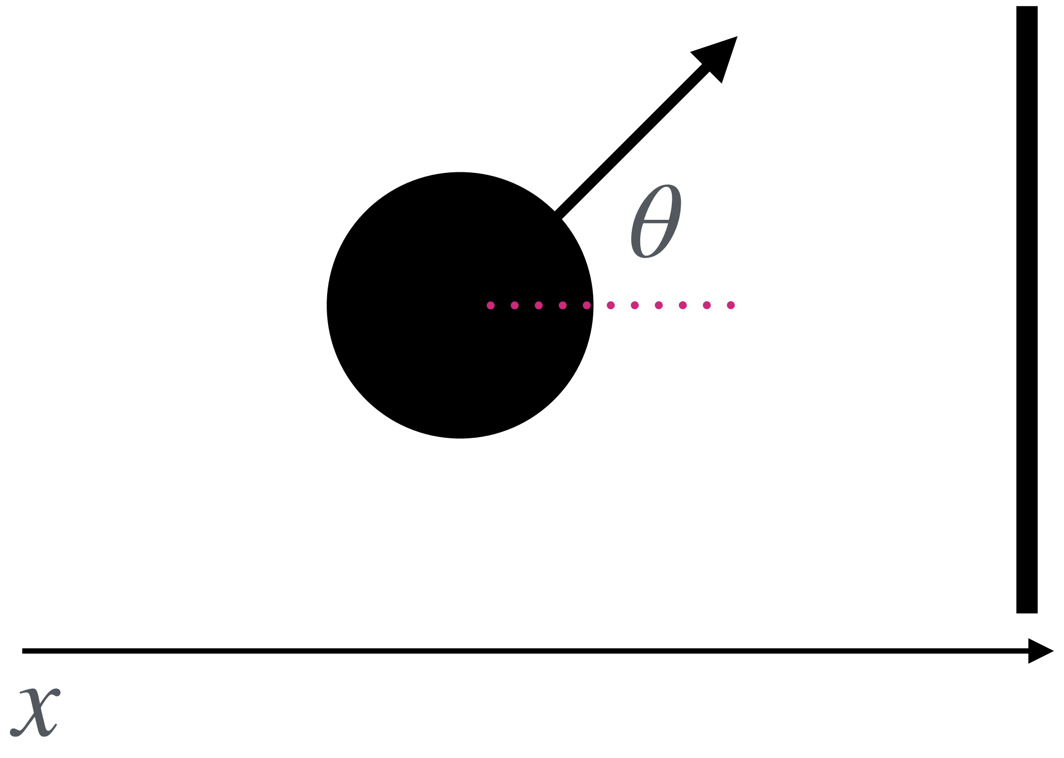
Optimizing through contact



Optimizing through contact

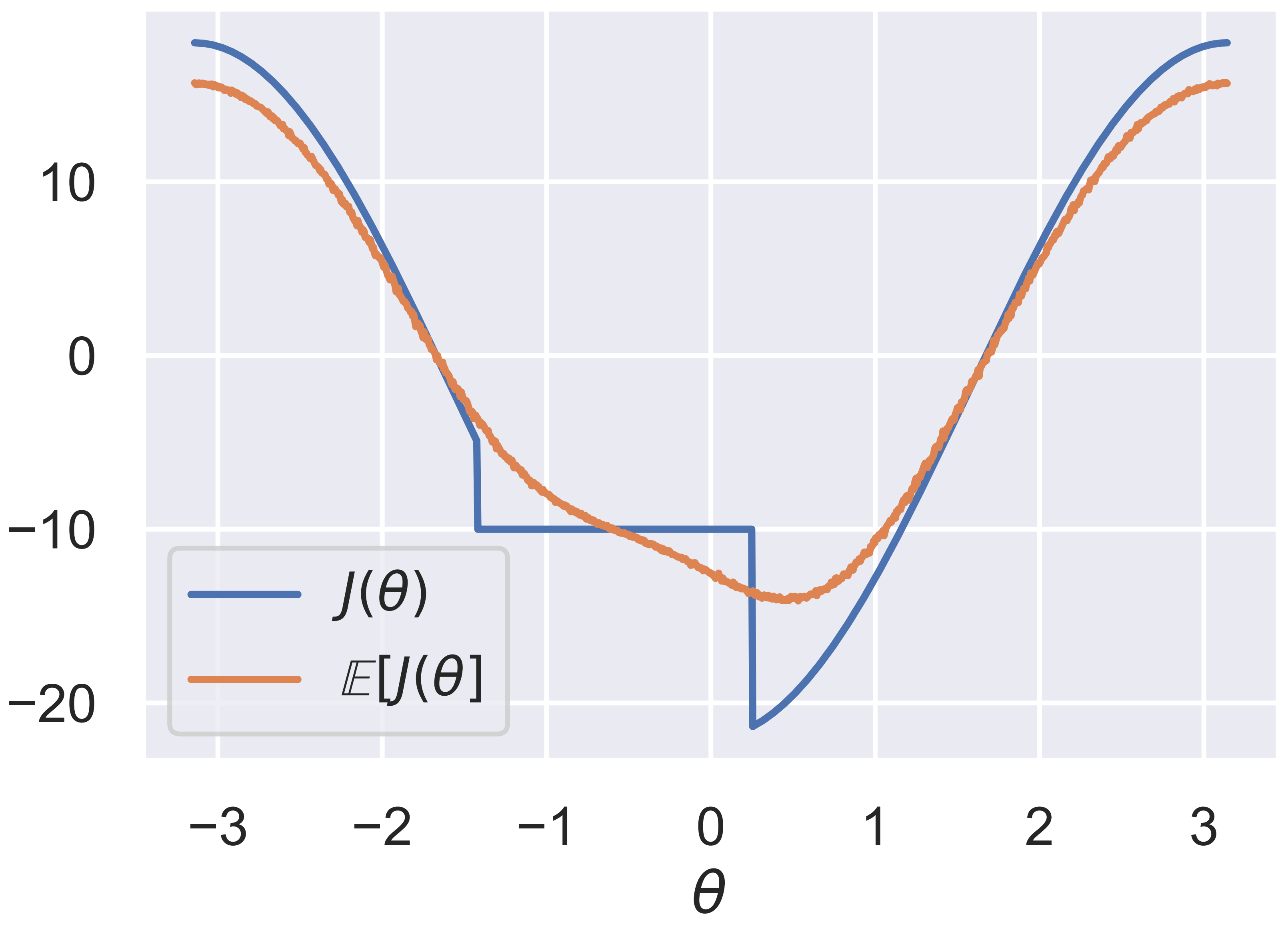
- Make input (policy) noisy

$$J(\theta) = \begin{cases} x_0 + v\cos(\theta)t + \frac{1}{2}at^2 & \text{if no contact} \\ w & \text{otherwise} \end{cases}$$

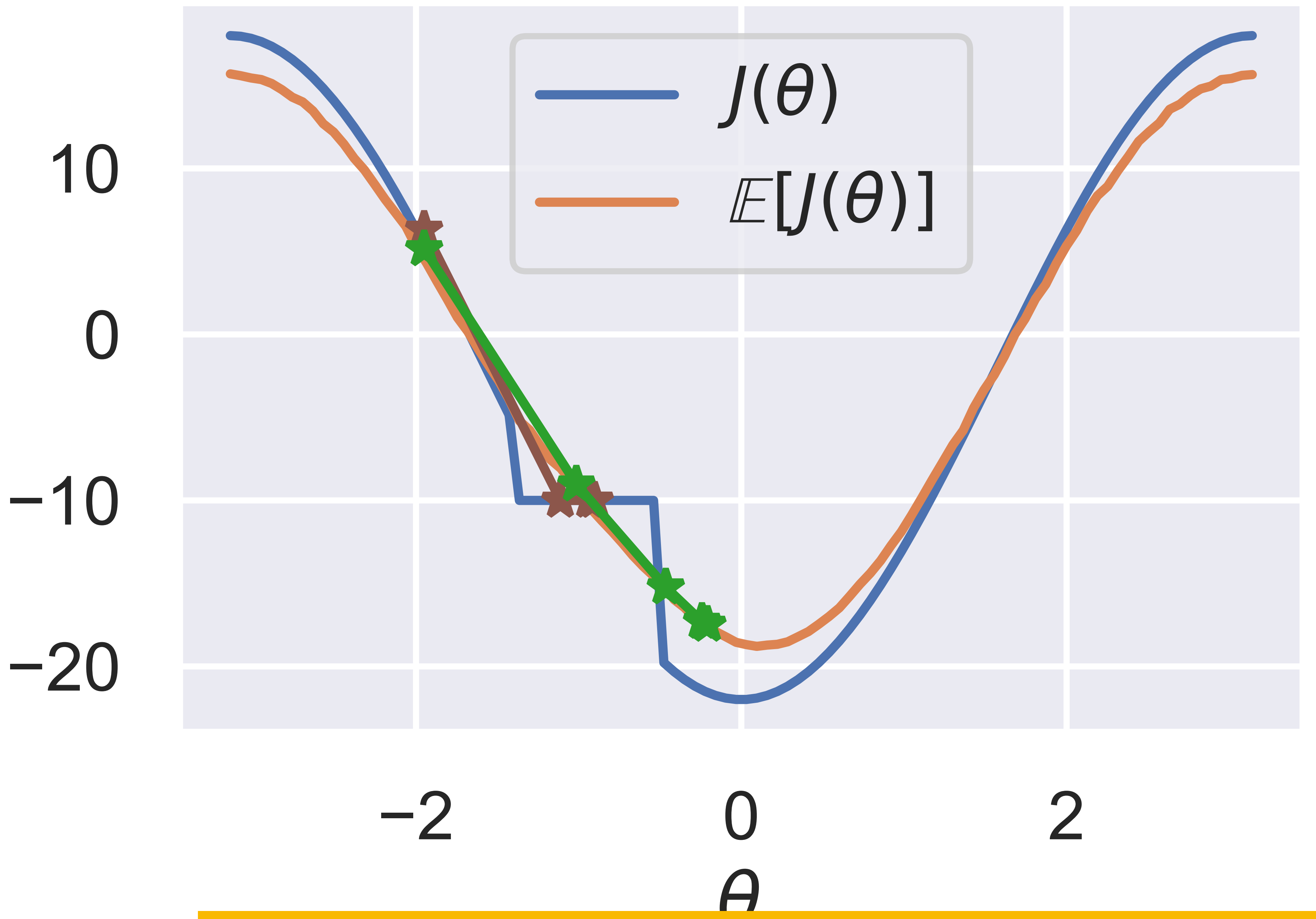


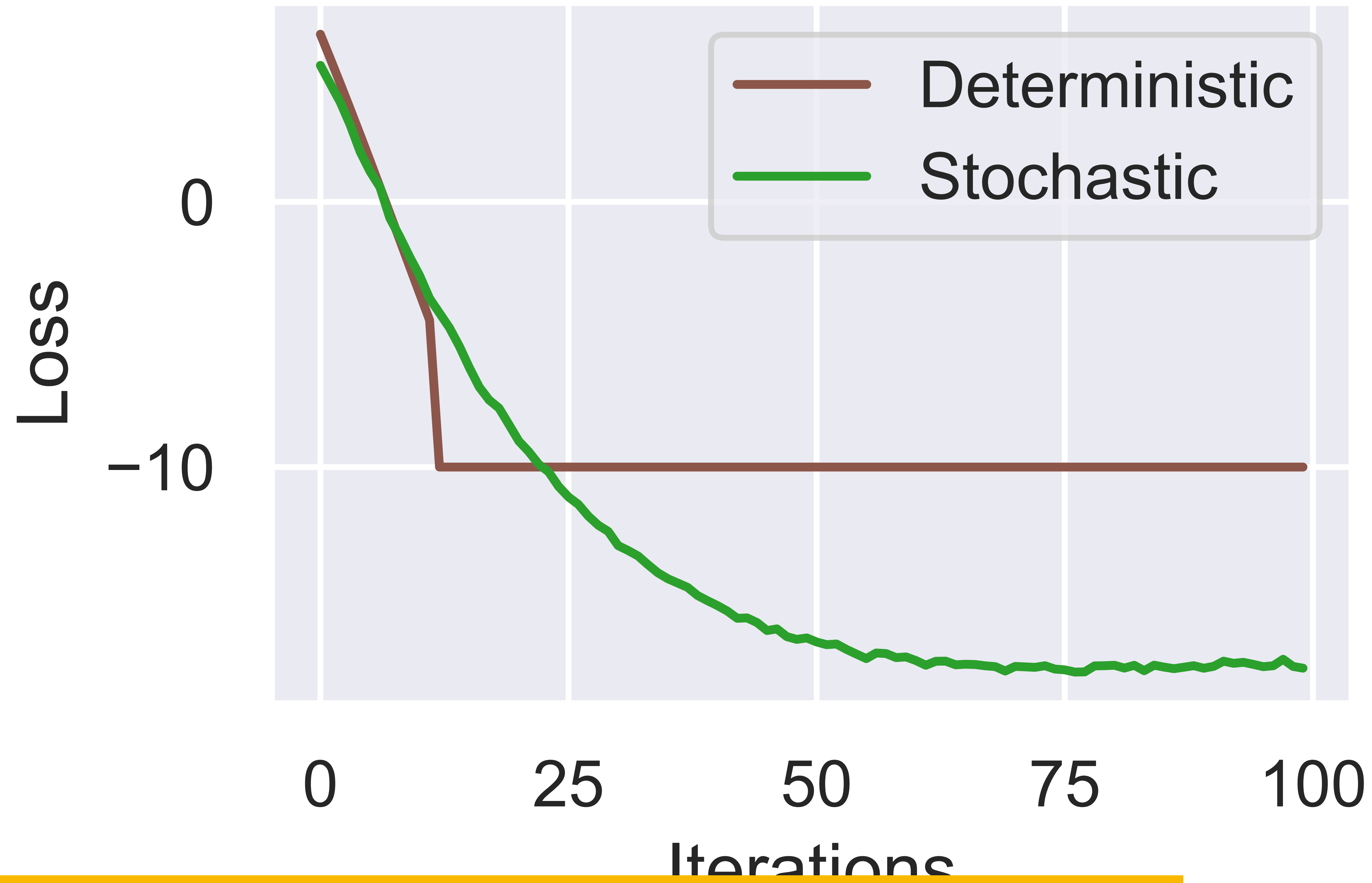
$$\boxed{\hat{\theta} \sim \mathcal{N}(\theta, \sigma^2)}$$

$$v_0 = 10\ m/s \quad g = 10\ m^2/s \quad w = 10$$



Optimizing through contact





Takeaway: Stochastic optimization helps us solve non-smooth tasks

Takeaway: ZoG have been successful in RL as they can learn through contact

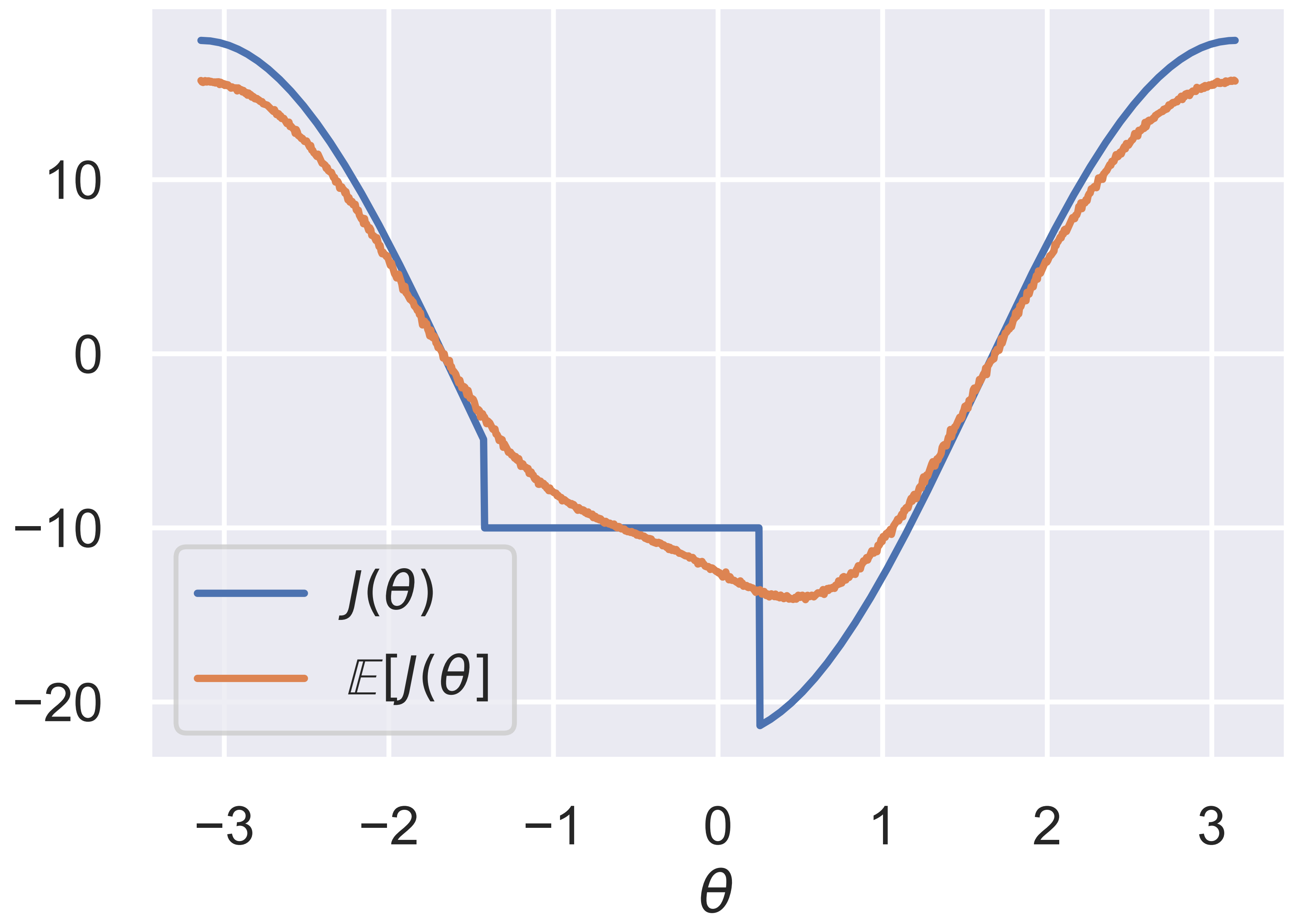
What can we do about it?

- ZoG RL by its formulation smooths out landscapes and enables optimization through contact.
- But it is SLOW and not scalable
- What else can we do?
  - Domain randomization (makes robust not good)
  - Initial state randomization
  - ....

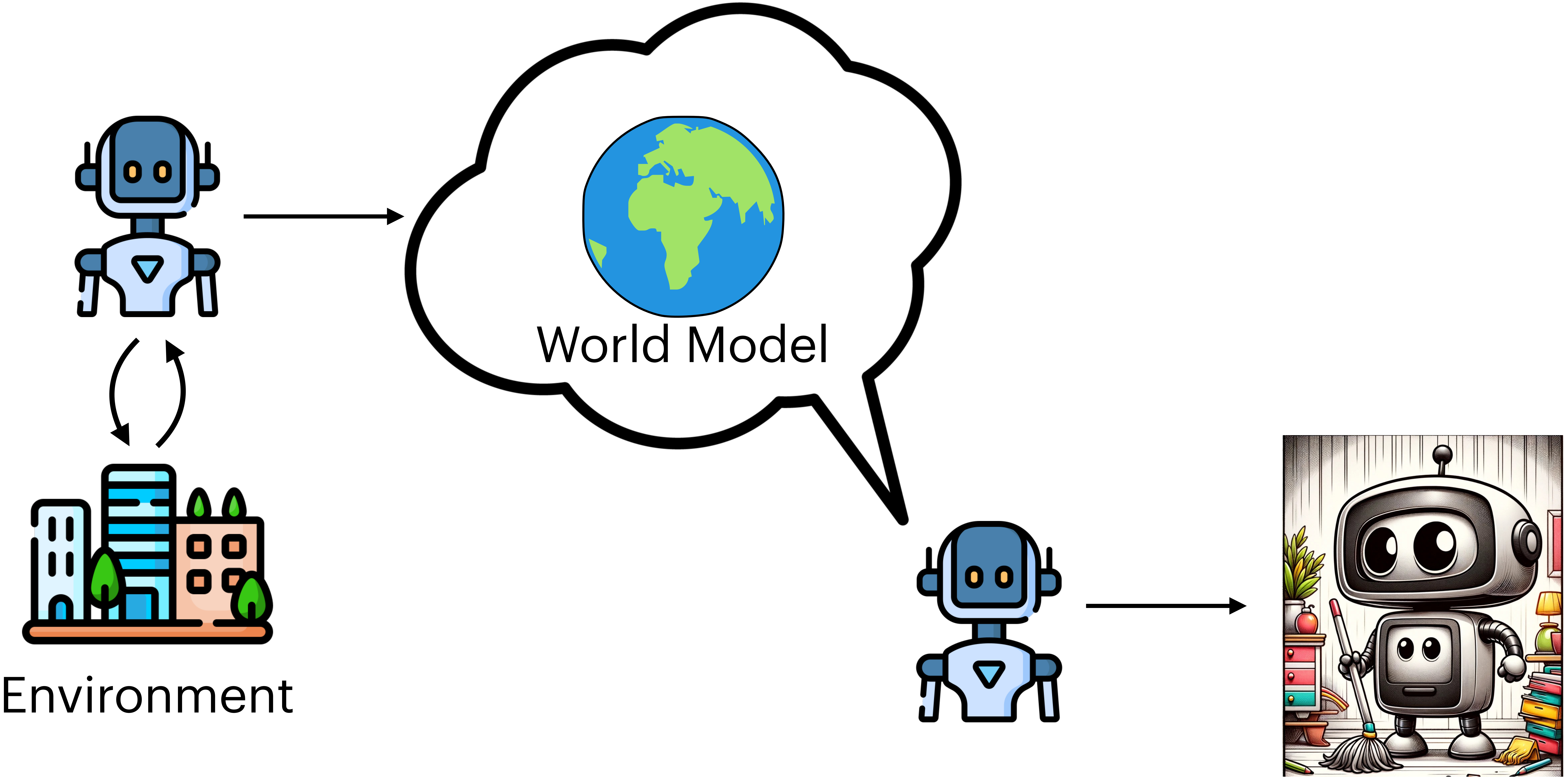
All of these work but they make data inefficient approaches even more data inefficient!

Surrogate problem formulation

- What if instead of solving the real problem, we optimize a surrogate problem directly?
- Instead of $f(\theta)$, solve for $F_\phi(\theta)$
- We need these models to
  - Have minimal optimality gap $||\hat{\theta} - \theta^*||$
  - Be smoother than the original $||\nabla F_\phi(\theta)|| \leq ||\nabla f(\theta)||$



World Model Framework

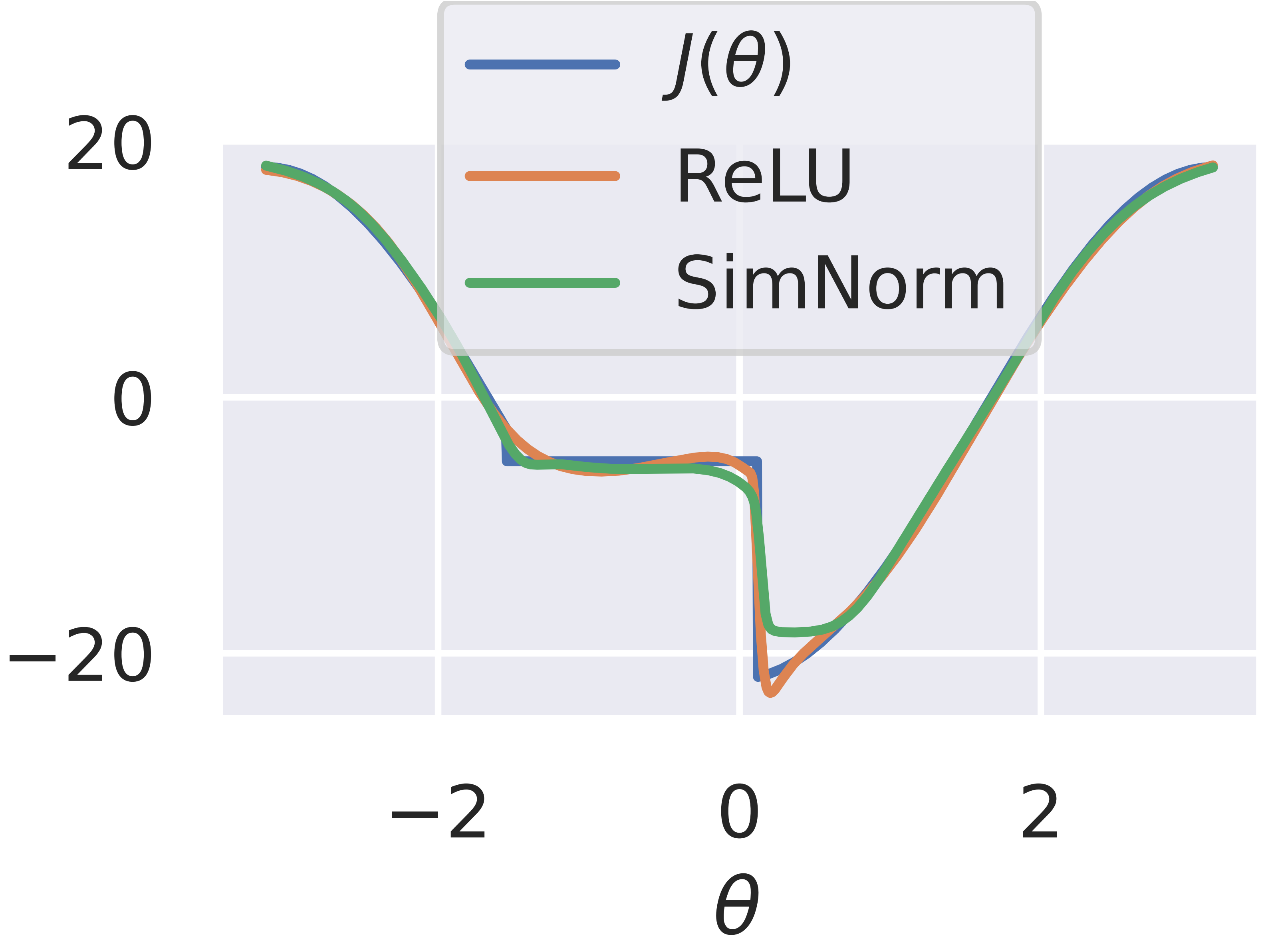


World models are smooth surrogates

- When regularized correctly, world models can act as smooth surrogates
  - No sampling required!
- Maps z into V L-dimensional simplices

$$\mathrm{SimNorm}(\boldsymbol{z}) := [\boldsymbol{g}_1, \dots, \boldsymbol{g}_L], \quad \boldsymbol{g}_i = \mathrm{Softmax}(\boldsymbol{z}_{i:i+V})$$

- The key is not to make models accurate
  - But to make them smooth
  - And have a low optimality gap



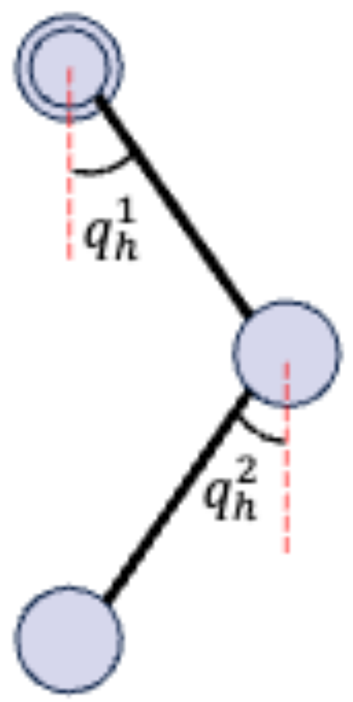
Model	Model error	Opt. gap
True	0.0	16.850
ReLU	0.707	16.046
SimNorm	1.131	3.473

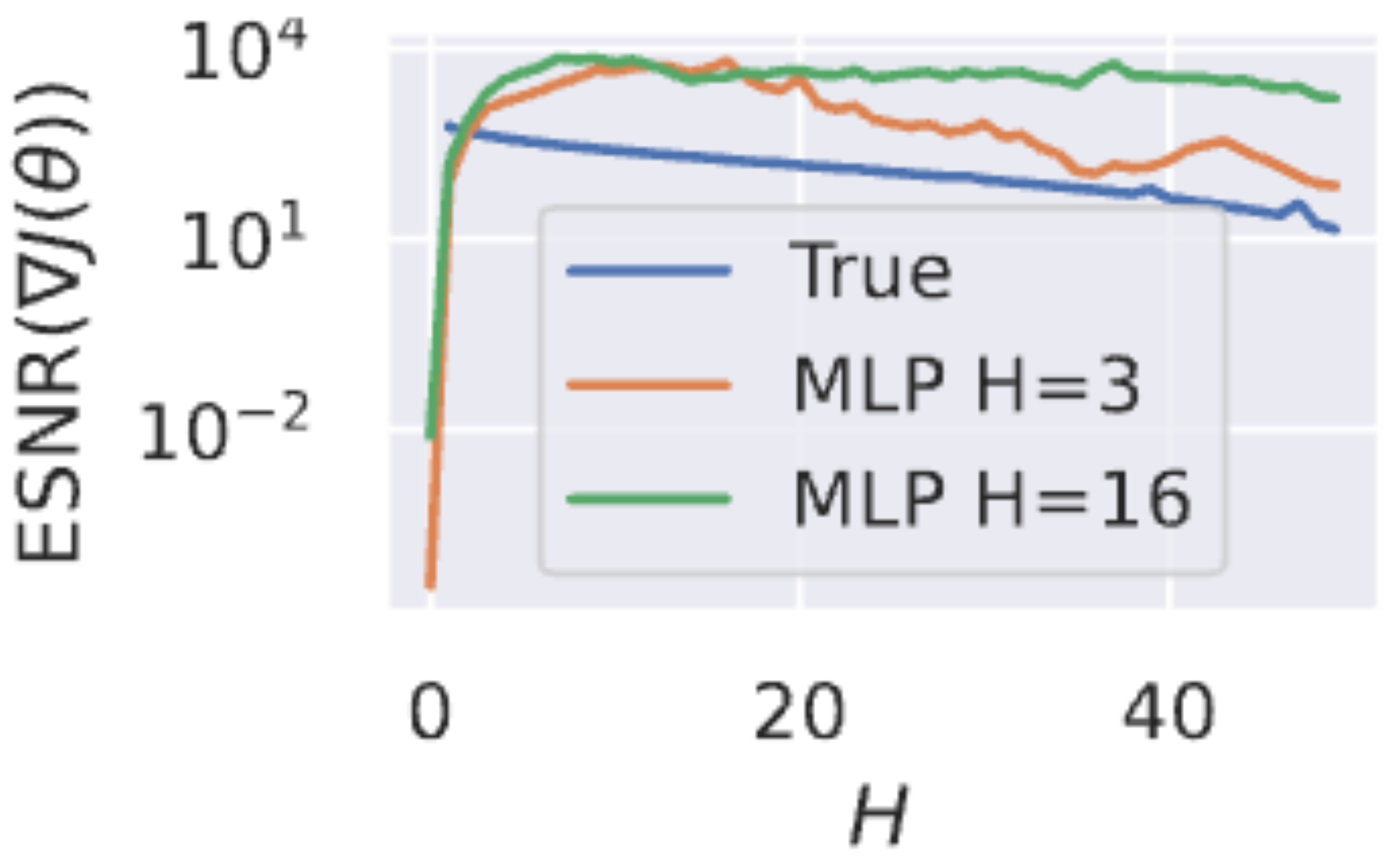
(c) Model error and optimality gap.

World models provide stability

- When dealing with chaotic systems (e.g. double pendulum) gradients become less useful as we increase horizon H
- Policy gradient usefulness can be measured by variance or Expected Signal-to-Noise Ratio (ESNR)

$$\mathrm{ESNR}(\nabla J(\boldsymbol{\theta})) = \mathbb{E}\left[\frac{\sum \mathbb{E}\left[\nabla^{[1]} J(\boldsymbol{\theta})\right]^2}{\sum \mathrm{Var}\left[\nabla^{[1]} J(\boldsymbol{\theta})\right]}\right]$$





World models are good surrogates

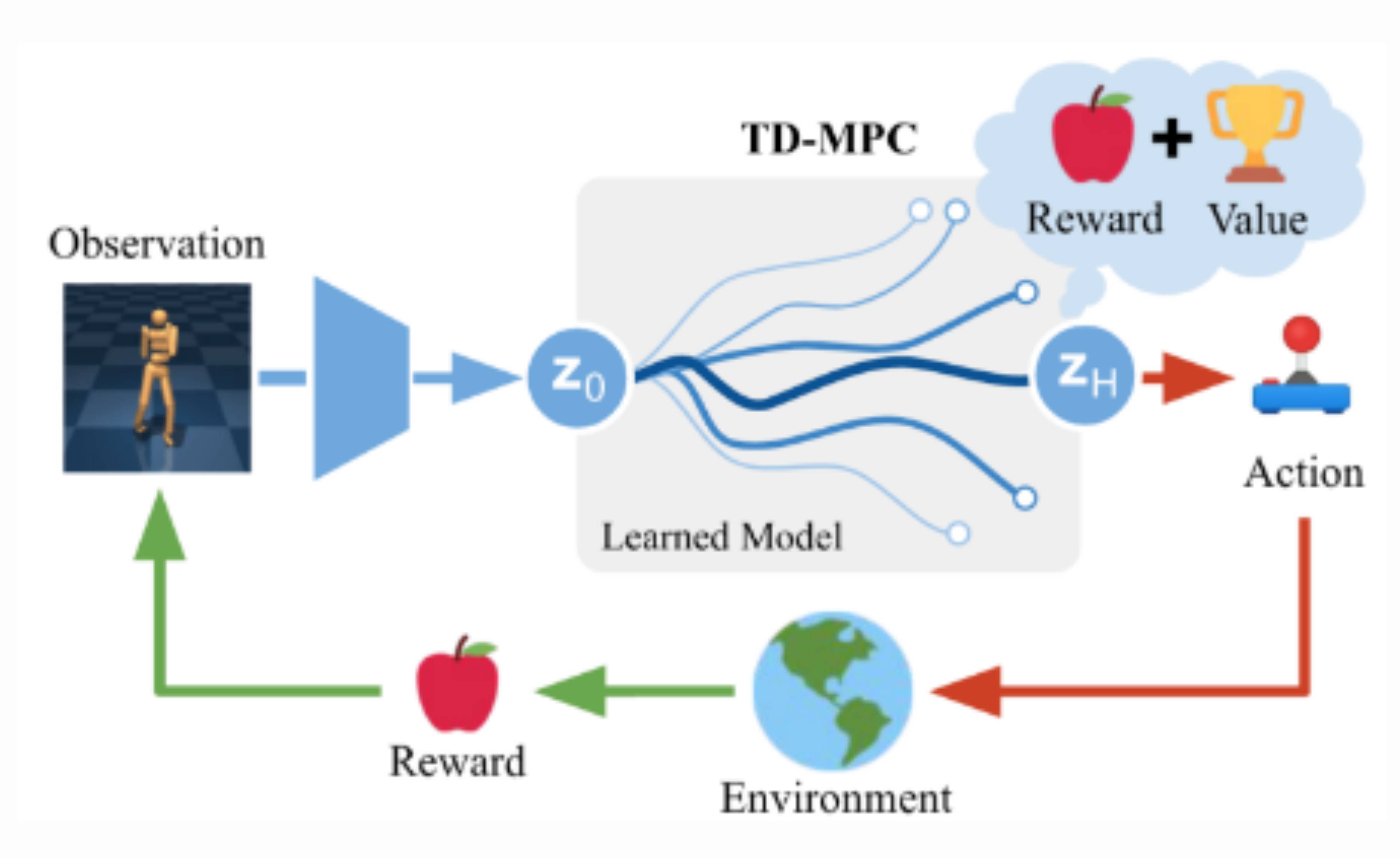
- Provide smooth problem landscapes
- Provide stable gradients through long trajectories

Takeaway: it is better to optimize over world models rather than true dynamics

TDMPC2

A scalable multi-task world model approach

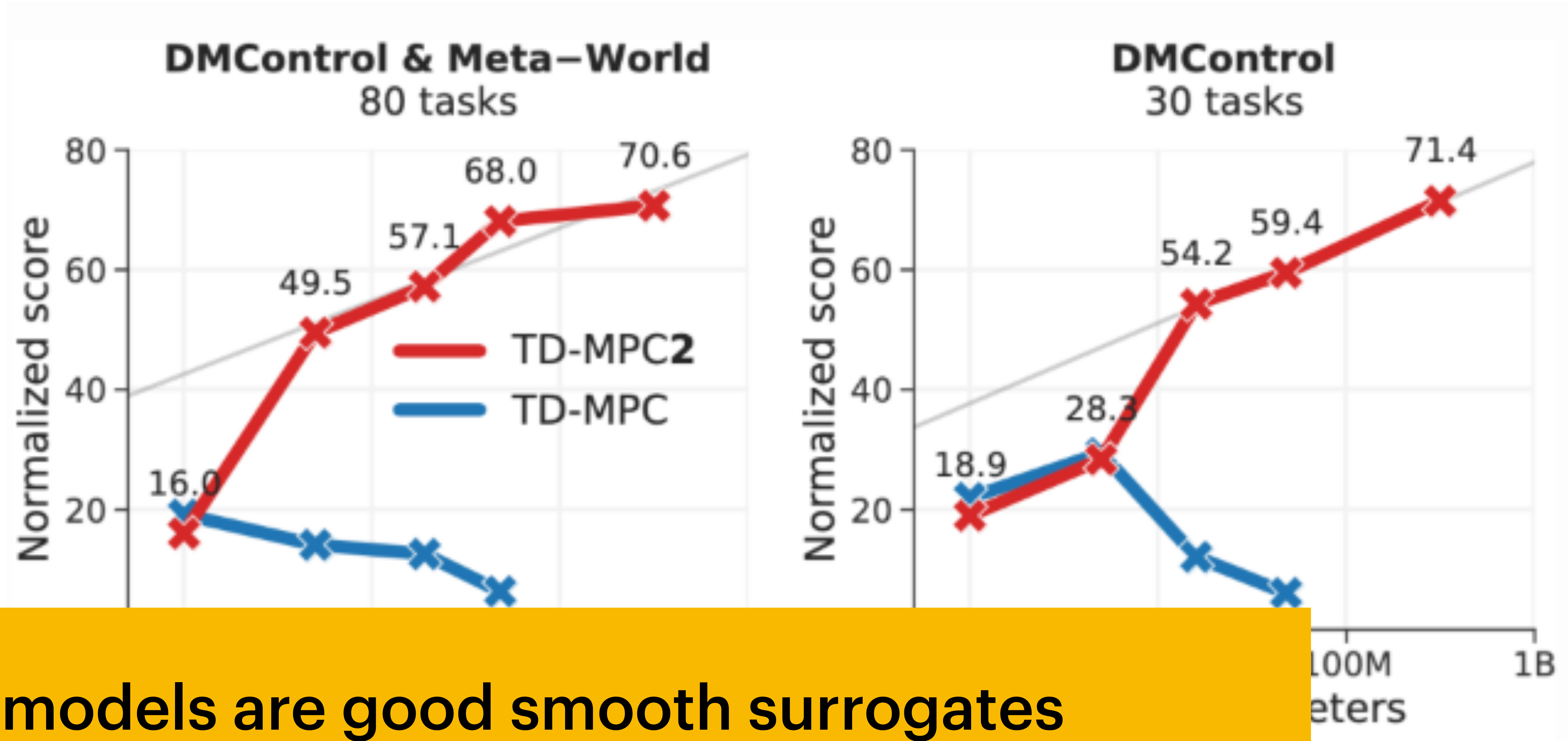
- Model-based RL approach
- Task-orientated latent dynamics model
- Learns by relevant reward, not by input reconstruction
- SAC actor-critic policy combined with online planning



TDMPC2

A scalable multi-task world model approach

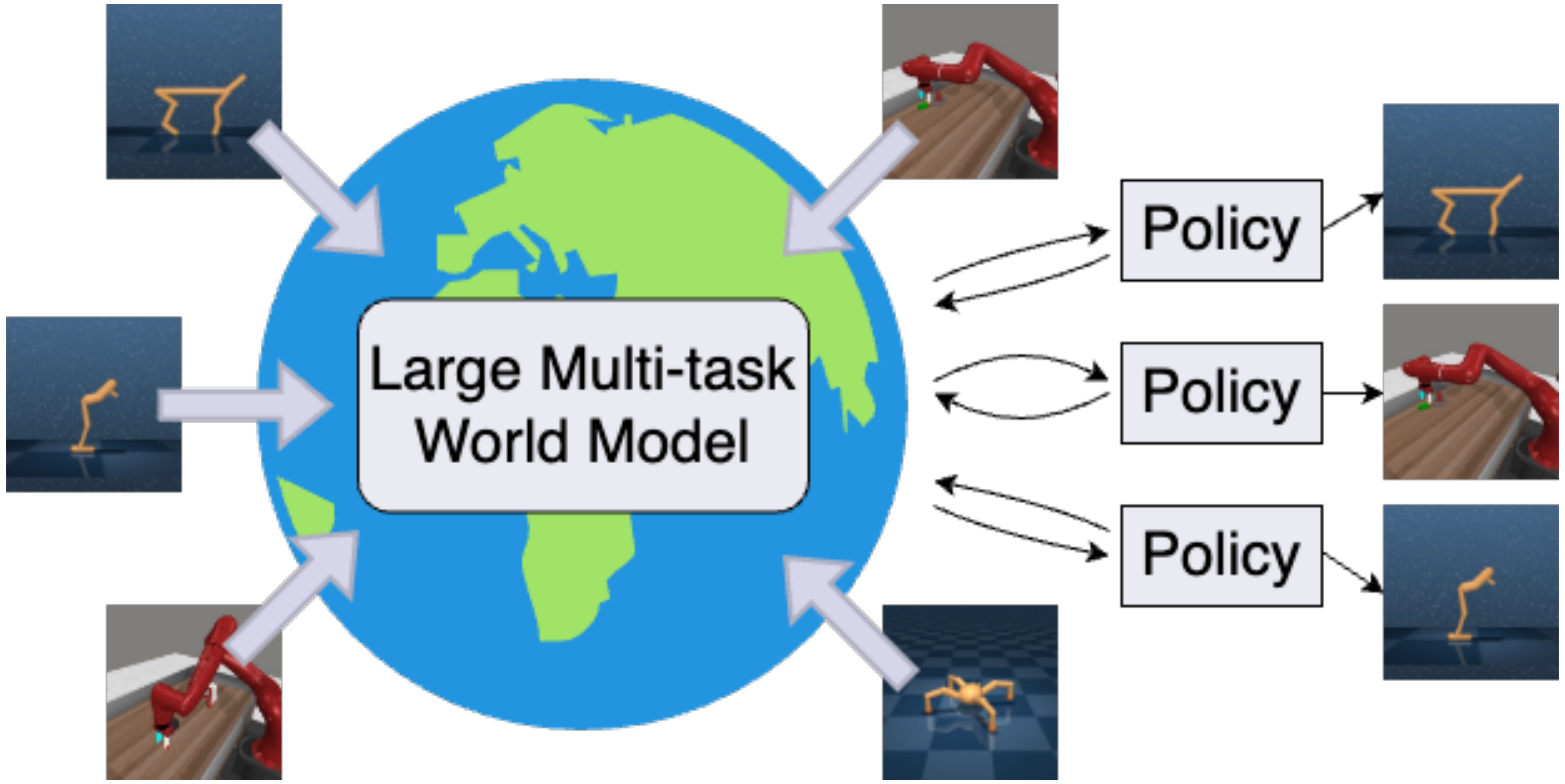
- First multi-task RL policy that scales to <u>80 different tasks</u>
- Across MetaWorld and DMC
- Relies mostly on online planning



Turns out that TDMPC2 world models are good smooth surrogates

But TDMPC2 choses to use ZoG, what if we use FoG?

PWM: Policy Learning with Large World Models



1. Regularized large models enable efficient policy learning
2. Use First-order optimization to train policies in <10m per task

PWM: Policy Learning with Large World Models

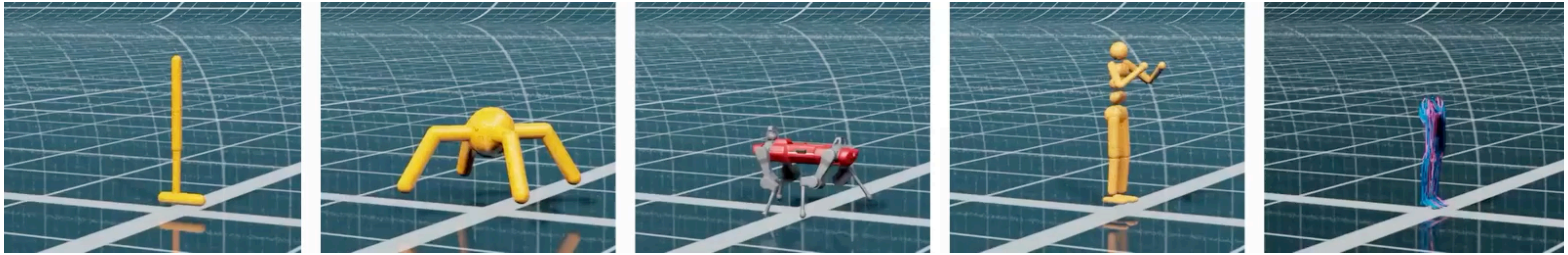
- TDMPC2 world models $(E_\phi(\boldsymbol{s},\boldsymbol{e}), F_\phi(\boldsymbol{s},\boldsymbol{a},\boldsymbol{e}), R_\phi(\boldsymbol{s},\boldsymbol{a},\boldsymbol{e}$
  - H=16 and γ=0.99
- Model-free critic trained with TD(λ)
- Actor trained with FoG

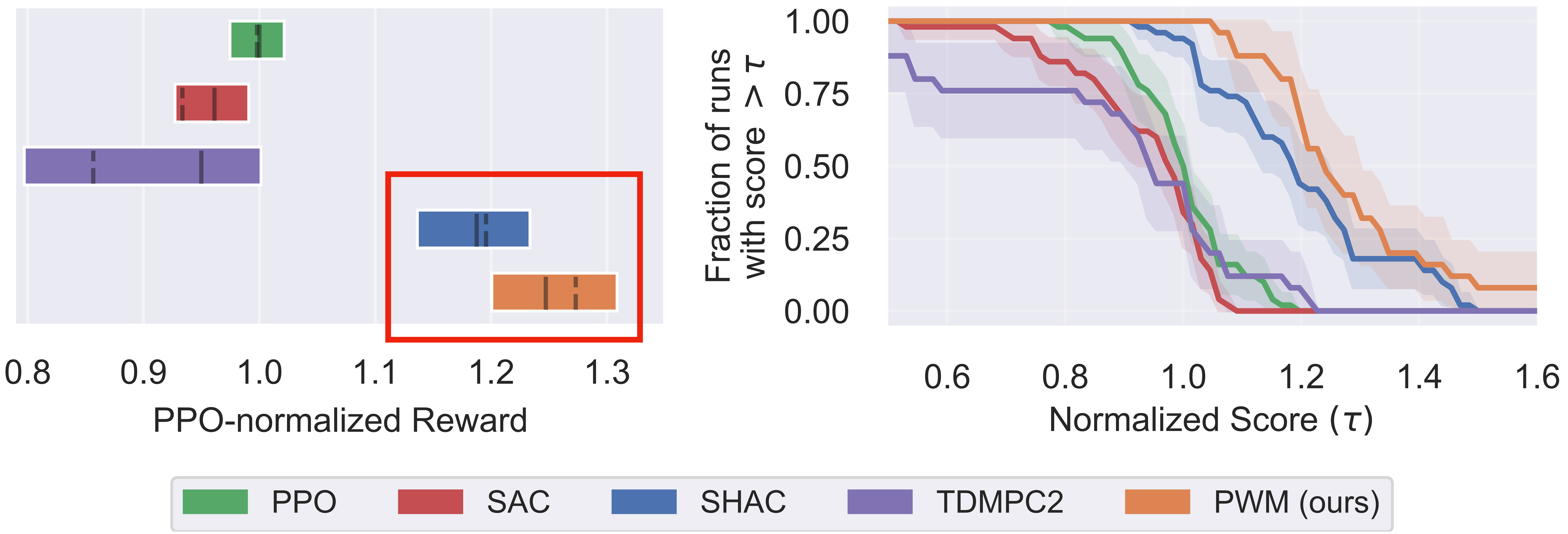
$$\mathcal{L}_\pi(\boldsymbol{\theta}) := \mathbb{E}_{\substack{\boldsymbol{s}_1 \sim \rho(\cdot) \\ \boldsymbol{a}_h \sim \pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{z}_h)}} \left[\sum_{h=1}^{H-1} \gamma^h R_\phi(\boldsymbol{z}_h, \boldsymbol{a}_h) + \gamma^H V_{\boldsymbol{\psi}}(\boldsymbol{z}_H) \right]$$

Algorithm 1: PWM: Policy optimization through World Model

Given: Multi-task dataset $\mathcal{B}$
Given: γ: discount rate
Given: $\alpha_{\boldsymbol{\theta}}, \alpha_{\boldsymbol{\psi}}, \alpha_{\boldsymbol{\phi}}$: learning rates
Initialize learnable parameters $\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\phi}$
▷ `Pre-train world model once`
for *N epochs* **do**
  $\boldsymbol{s}_{1:H}, \boldsymbol{a}_{1:H}, r_{1:H}, \boldsymbol{e} \sim \mathcal{B}$
  $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + \alpha_{\boldsymbol{\phi}} \mathcal{L}_{wm}(\boldsymbol{\phi})$ ▷ `Eq. 10`
end
▷ `Train policy on task embedding` $\boldsymbol{e}$
for *M epochs* **do**
  $\boldsymbol{s}_1 \sim \mathcal{B}$
  $\boldsymbol{z}_1 = E_{\boldsymbol{\phi}}(\boldsymbol{s}_1, \boldsymbol{e})$
  for *h=[1, ..., H]* **do** ▷ `Rollout`
    $\boldsymbol{a}_h \sim \pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{z}_h)$
    $r_h = R_\phi(\boldsymbol{z}_h, \boldsymbol{a}_h, \boldsymbol{e})$
    $\boldsymbol{z}_{h+1} = F_\phi(\boldsymbol{z}_h, \boldsymbol{a}_h, \boldsymbol{e})$
  end
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha_{\boldsymbol{\theta}} \mathcal{L}_\pi(\boldsymbol{\theta})$ ▷ `Eq. 7-9`
  $\boldsymbol{\psi} \leftarrow \boldsymbol{\psi} + \alpha_{\boldsymbol{\psi}} \mathcal{L}_V(\boldsymbol{\psi})$ ▷ `Eq. 6`
end

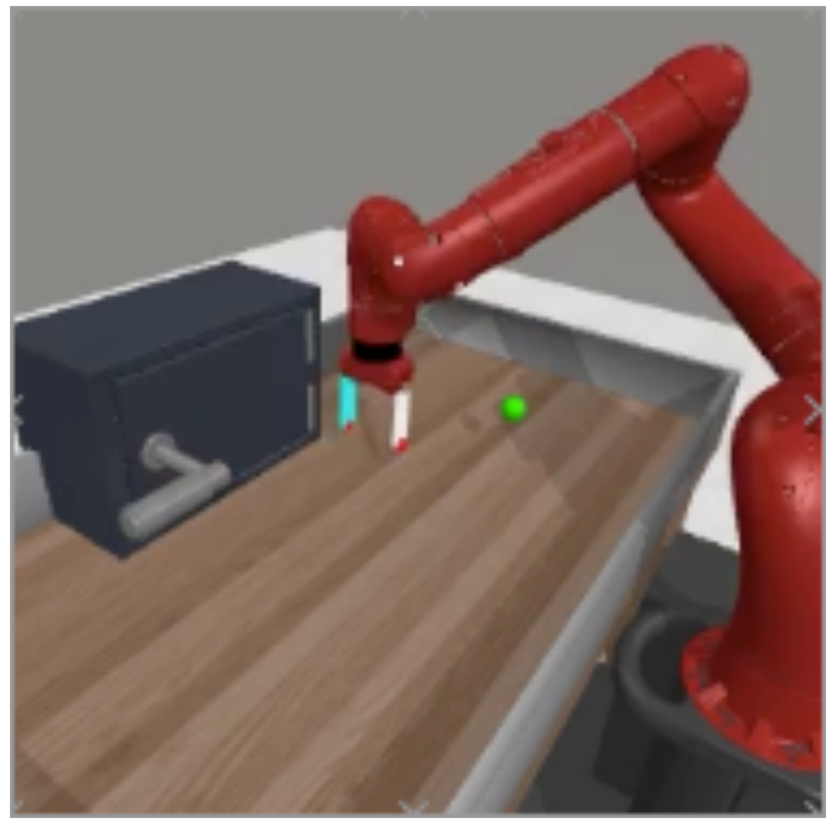
High-dimensional single-task

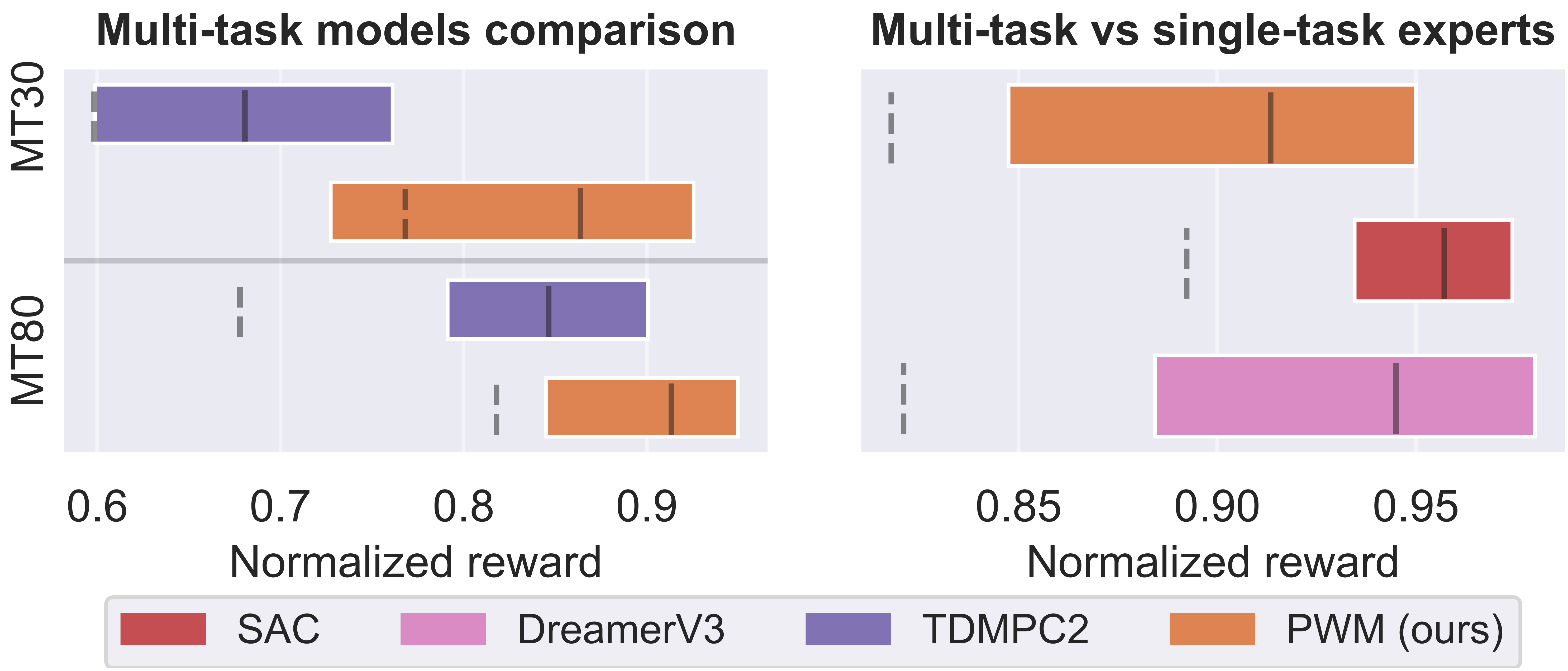




Takeaway: optimizing over surrogate models obtains better policies than ground truth!

Multi-task experiments



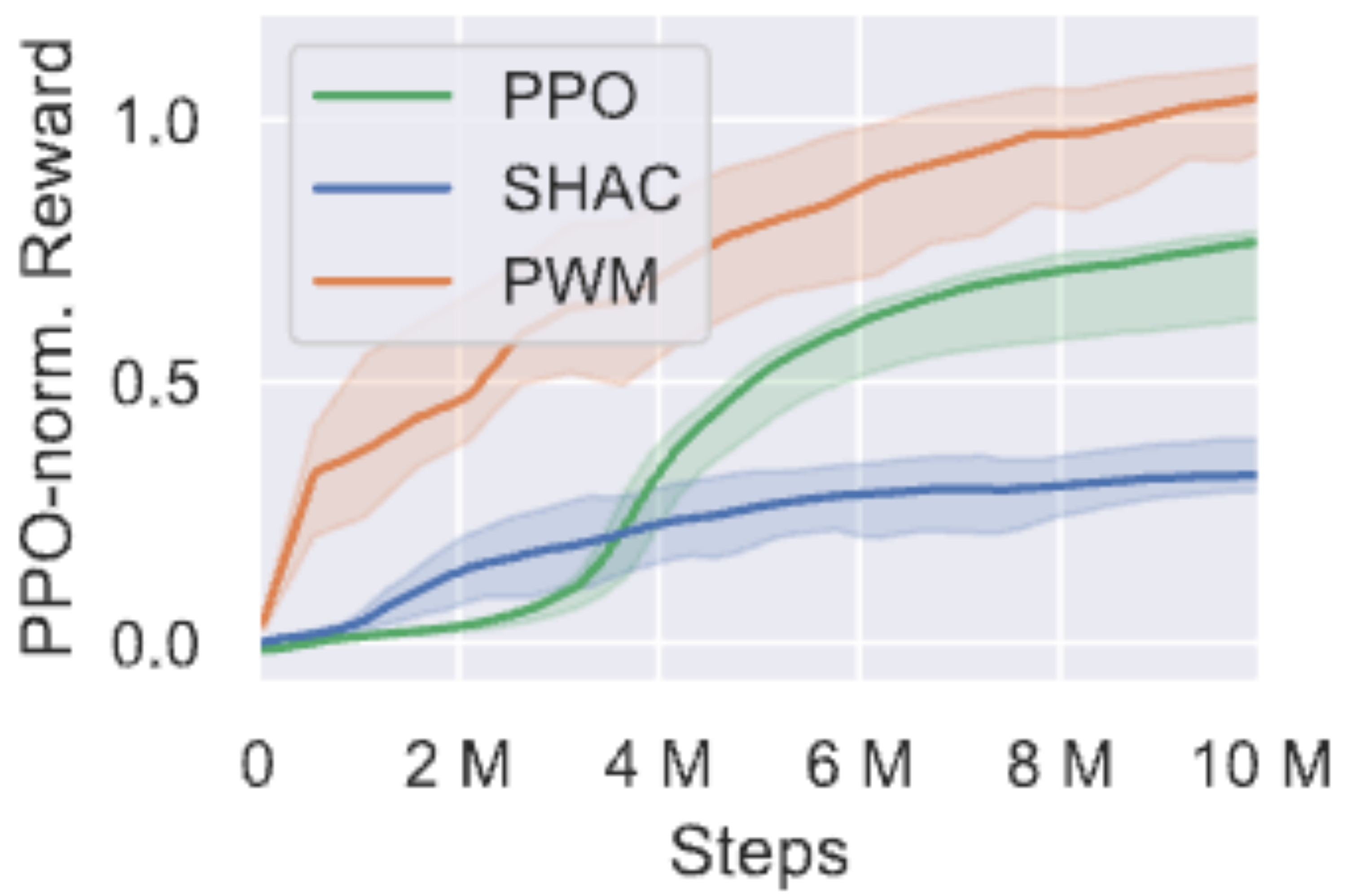


Beats TDMPC2 without the need for online planning -> more scalable

Matches single-task experts without any online interaction

Stiffness ablation

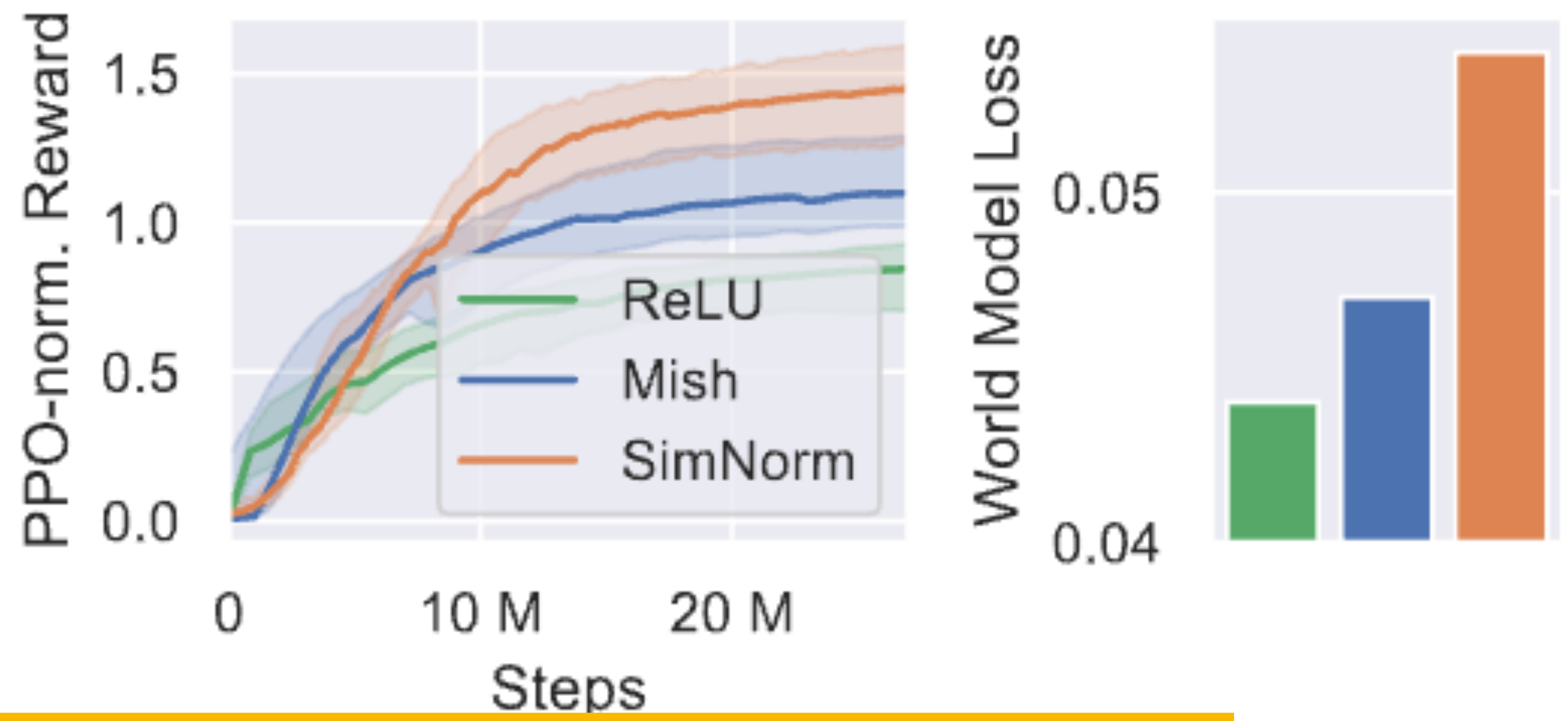
- If we increase contact stiffness, SHAC/FoG method performance decreases
- PWM sustains the same performance
- Hopper task



PWM is resilient to stiff dynamics

World Model Ablation

- More accurate models do not translate to better policy
- Actually the opposite



We should build world models for policy learning, not accuracy

Sample efficiency

1. Train world models for X time steps
2. Then train policy for 50k gradient steps

PWM is a more sample efficient policy learning technique but requires better trained world models.

(b) World model vs policy sample efficiency.

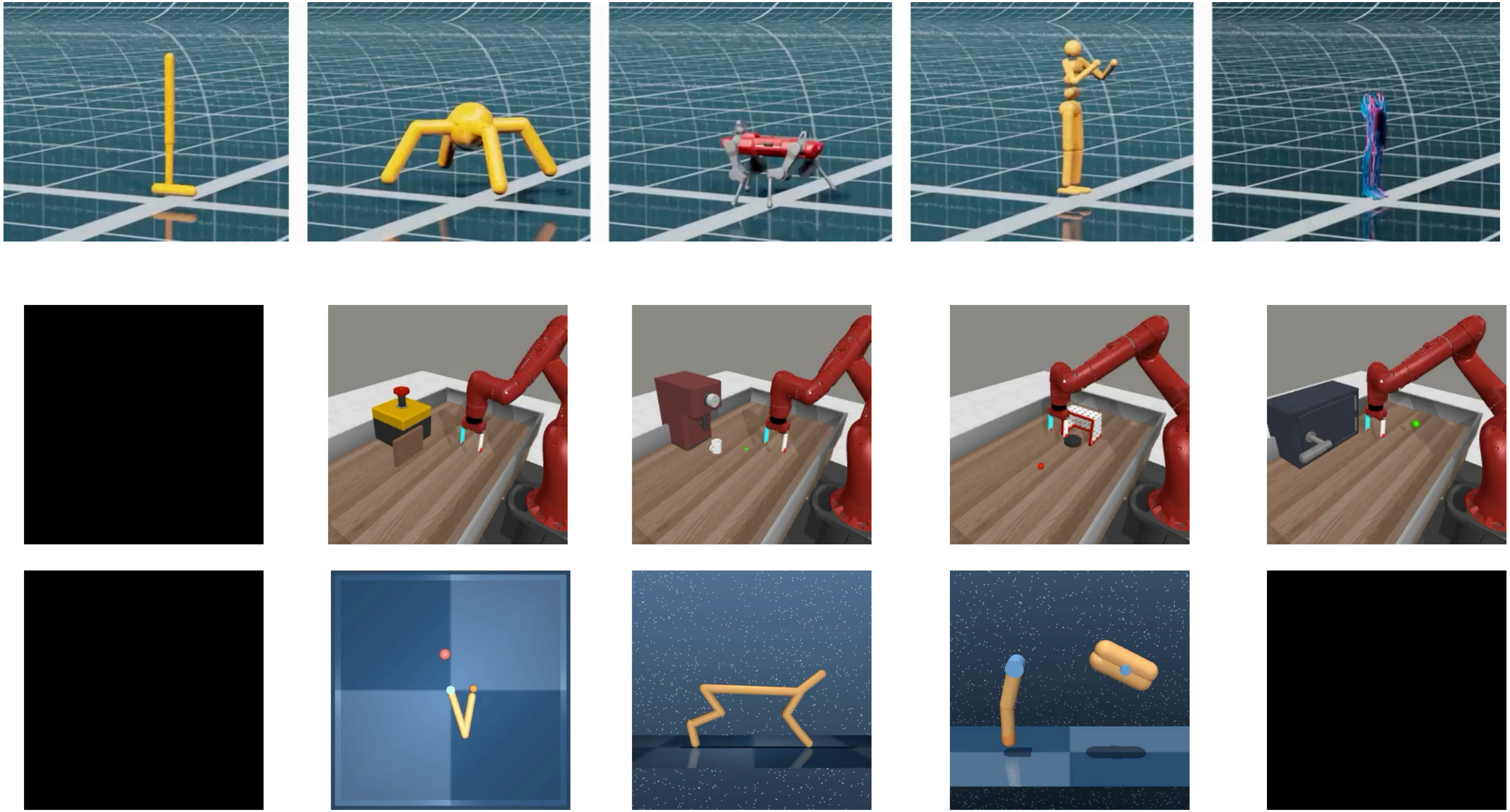
Conclusion

To get more efficient RL, we .. have to stop doing classic RL

PWM is a more scalable multi-task world model approach using FoG

When implemented correctly, world models can act as smooth surrogates of the true objective, resulting in better policies

Instead of training world models concurrently with policies, we should treat them as learned “offline simulations”



More at: https://policy-world-model.github.io/

Thank you

- Papers, code, data and more at: imgeorgiev.com/pwm/

Ignat
Georgiev

Varun
Giridhar

Nicklas
Hansen

Animesh
Garg