UNIVERSITY of WASHINGTON

A Data Diet for Robot Learning: Going Beyond On-Robot Data Collection

Abhishek Gupta

Everything discussed today is done by the **WEIRD Lab ++**

Washington **E**mbodied **I**ntelligence and **R**obotics **D**evelopment Lab

What are our research objectives?

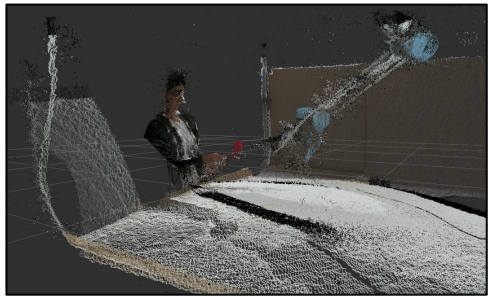
We want to deploy robots into messy, unstructured environments

Why – unstructured environments require diverse behaviors, and are difficult to model accurately

Requires moving beyond typical sense-plan-act paradigms

From Sense-Plan-Act to End-to-End Learning

Sense-plan-act



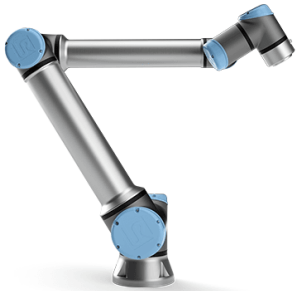
State Estimation

Modeling and Prediction

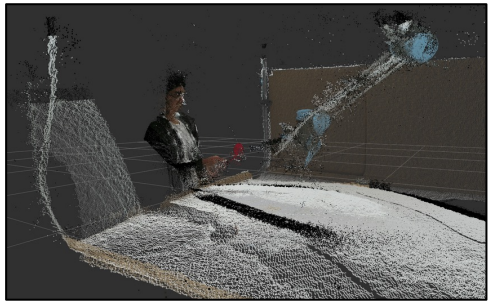
High-level planning

Low-level planning

Low-level control



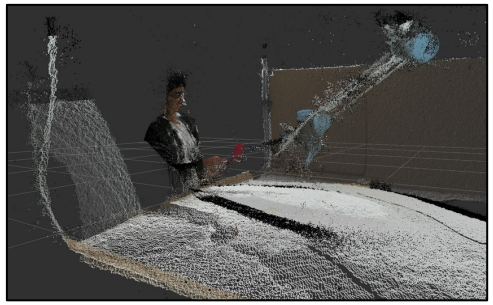
End-to-end learning



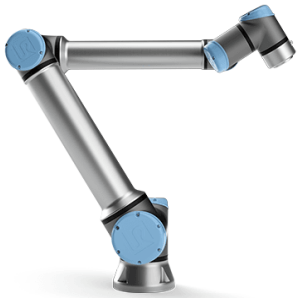
End to end policy – perception + control

Not quite so simple to actualize in practice!

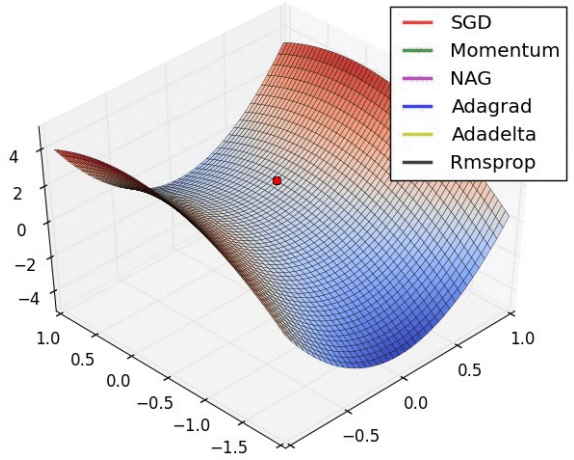
What are the elements of such a new recipe?



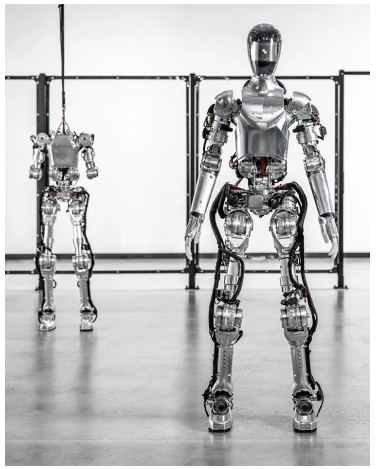
End to end policy –
perception + control



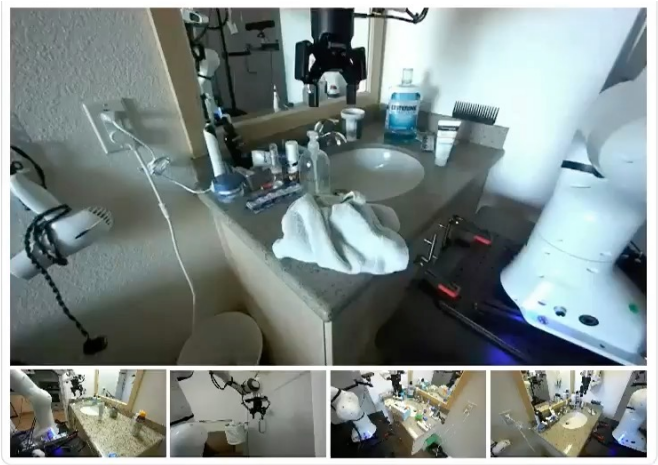
Algorithms



Hardware



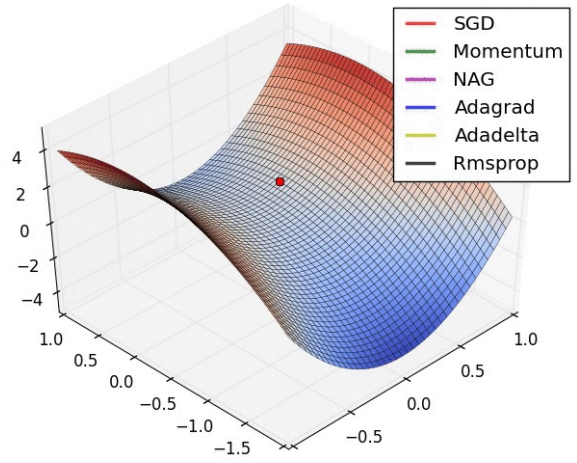
Data

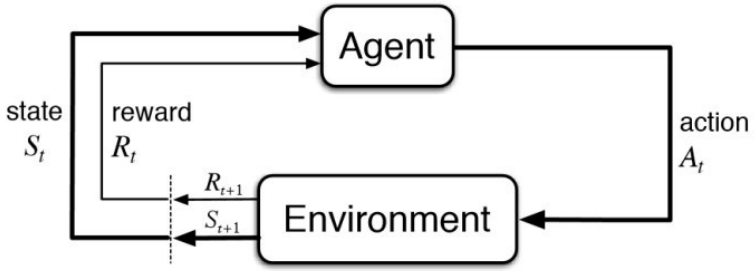


Where are we in instantiating this recipe?

Algorithms for Robotic Learning (kind of) exist and are stable

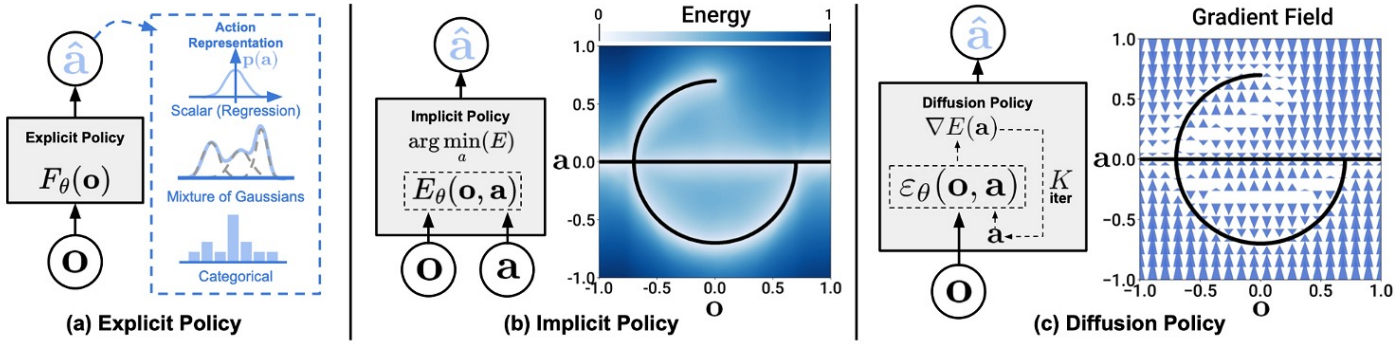
Algorithms





Proximal Policy Optimization

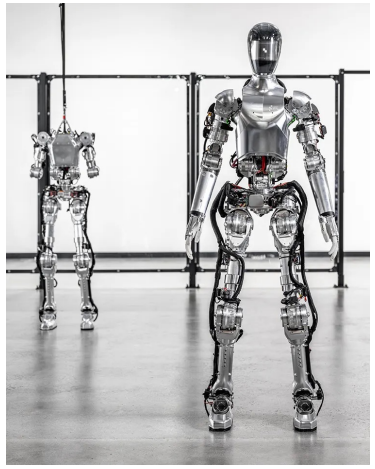
$$L^{CPI}(\theta) = \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t\right] = \hat{\mathbb{E}}_t\left[r_t(\theta)\hat{A}_t\right].$$



Where are we in instantiating this recipe?

Commodity Hardware for Robotic Learning is becoming cheaper

Hardware



(<$100K)

(<$10K)

(<$5K)

(<$30K)

Where are we in instantiating this recipe?

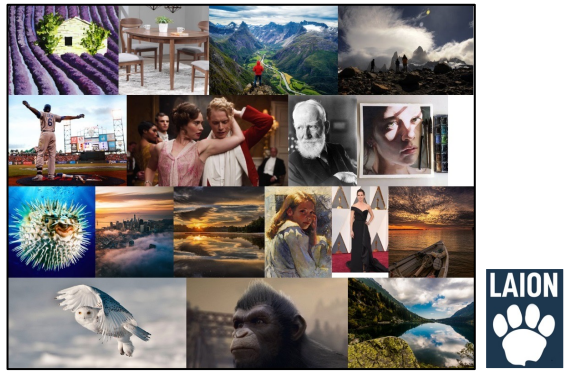
Data at Scale for Robotic Learning

Where is the data in robot learning???

Every other field has been transformed by the availability and quality of data!

What has it taken to scale data (a roboticists view)?

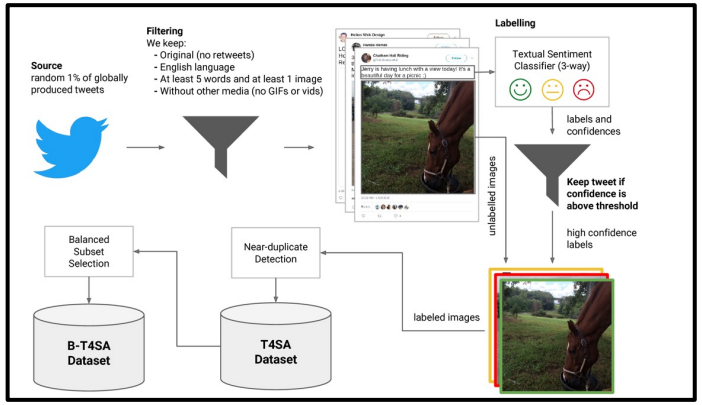
Vision

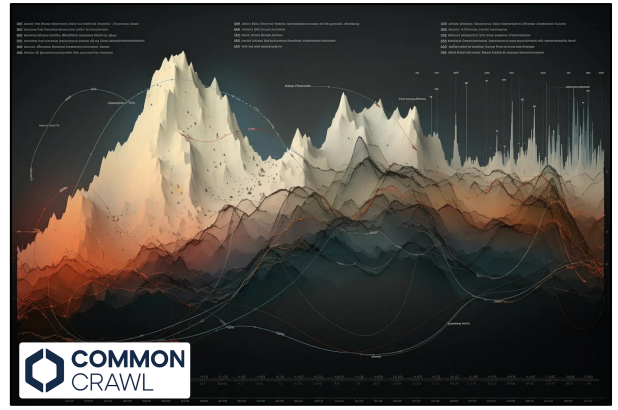


Order of magnitude growth

Language

```
If you like adult comedy cartoons, like South Park, then this is nearly a similar
format about the small adventures of three teenage girls at Bromwell High.
Keisha, Natella and Latrina have given exploding sweets and behaved like bitches,
I think Keisha is a good leader. There are also small stories going on with the
teachers of the school. There's the idiotic principal, Mr. Bip, the nervous
Maths teacher and many others. The cast is also fantastic, Lenny Henry's Gina
Yashere, EastEnders Chrissie Watts, Tracy-Ann Oberman, Smack The Pony's Doon
Mackichan, Dead Ringers' Mark Perry and Blunder's Nina Conti. I didn't know this
came from Canada, but it is very good. Very good!
```





Data grows passively, without requiring targeted collection

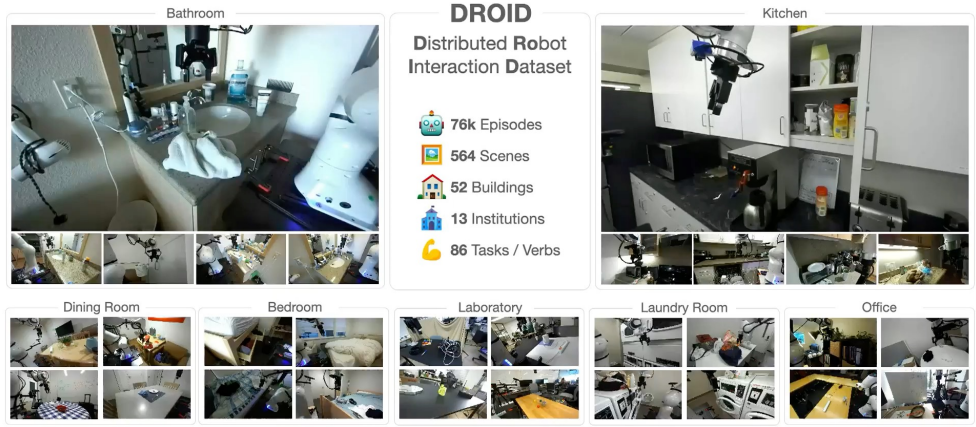
Will this work in robotics?

9.5 petabytes

April 2024 - 386 TiB of data/2.7 billion pages

5.85 billion image-text pairs

> 240TB



27 million frames

Limited diversity

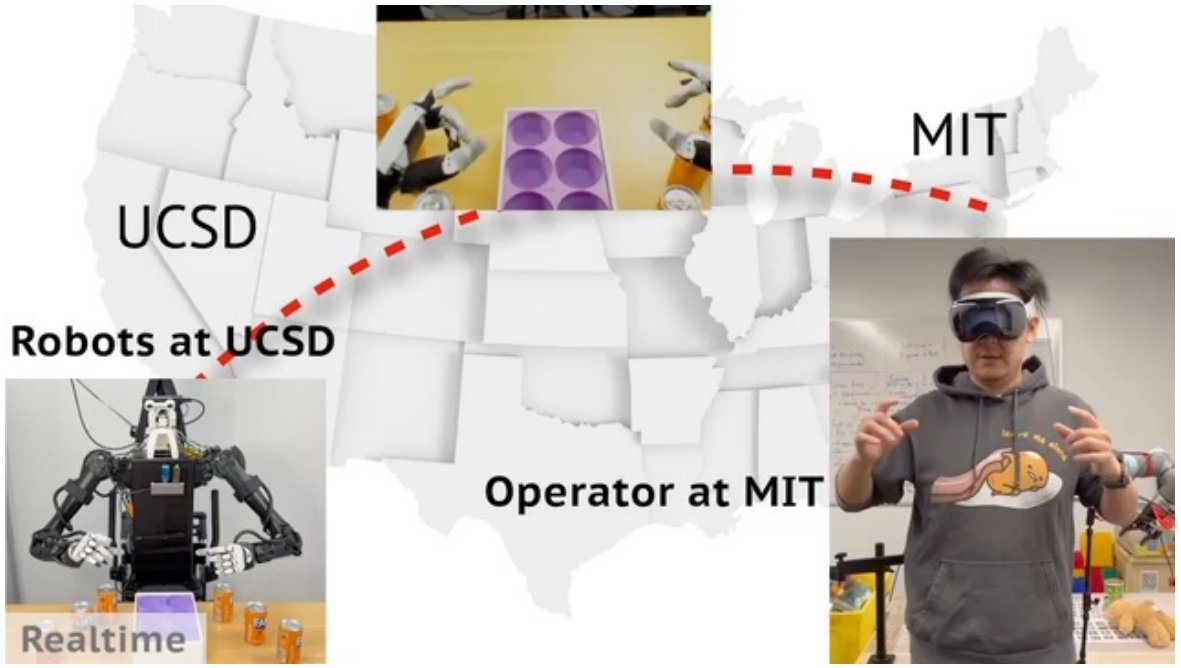
Manually collected by graduate students

<u>There are few robots deployed</u> – requires an untenable amount of manual effort!

Maybe we make teleoperation easier?

Teleoperation is getting better and easier – what if we just teleoperated billions of robot hours?

Chi et al

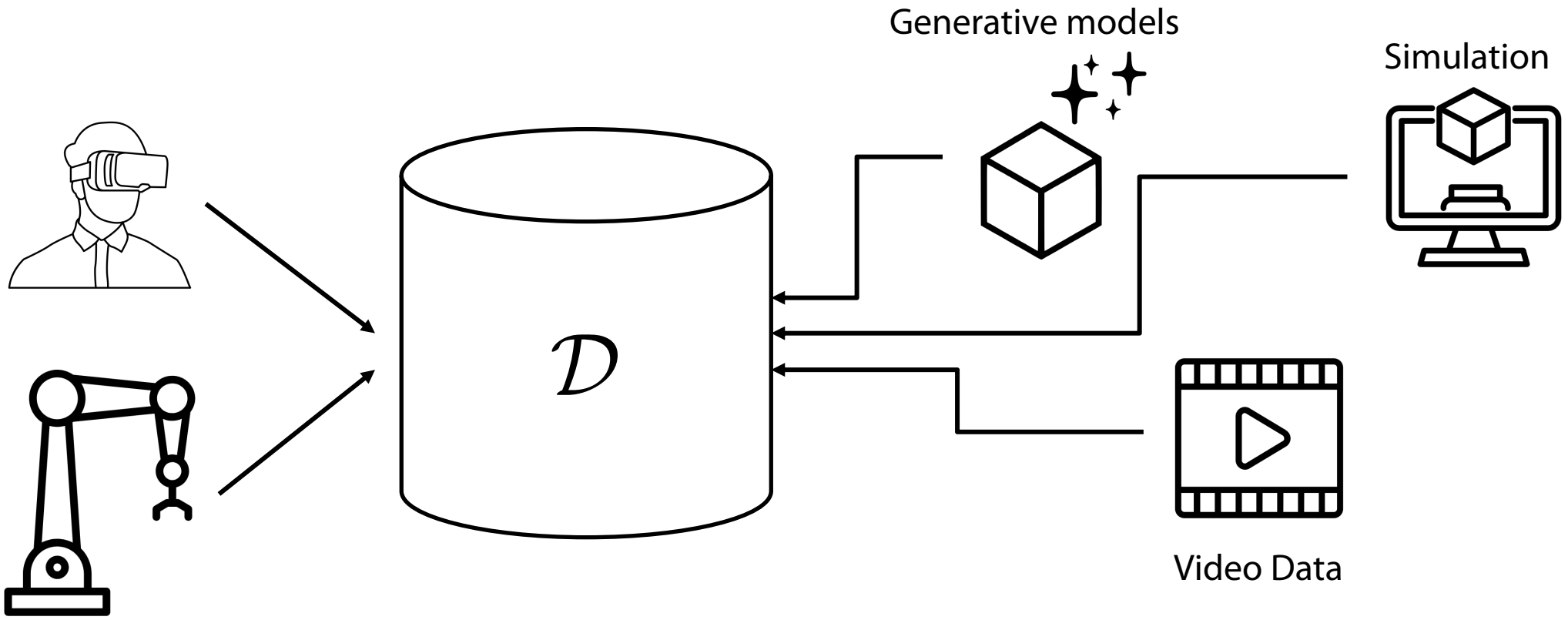


Cheng et al

Data must still be actively collected, requires coverage over diverse conditions

What might the solution to this data problem be?

Use cheaper, off-domain data!

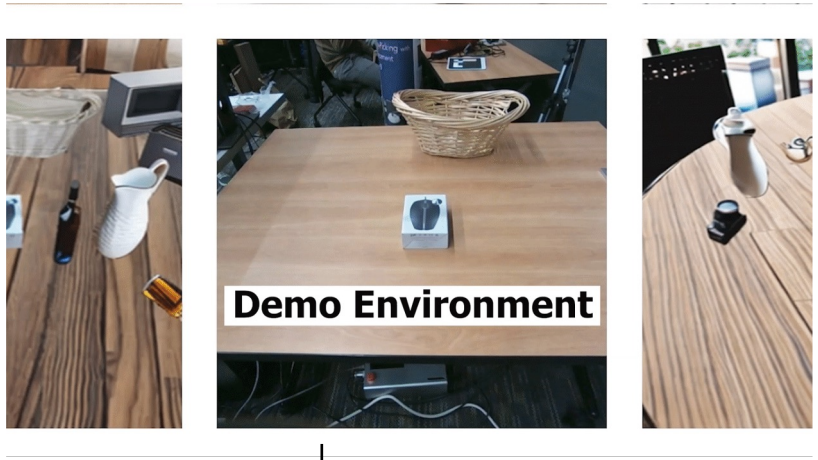


Data is lower “quality”, but still useful!

Can we learn from off-domain data?

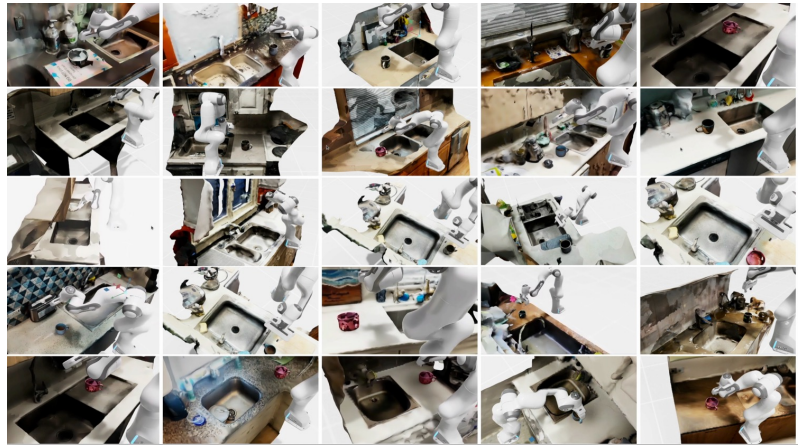
For robot learning data to scale economically, we must be able to leverage off-domain datasets

Generative Models



Video

Simulation



Requires in-domain expert data

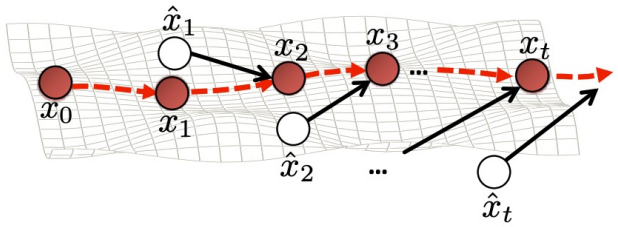
Action-free and physically unaware

Increasingly “off-domain”

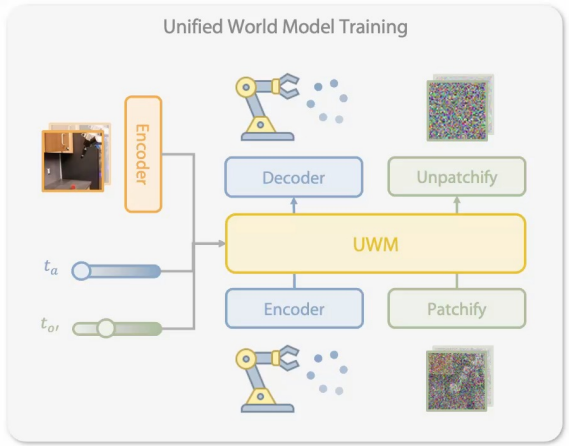
How can we learn from large-scale off-domain data in robotics?

Learning from Off-Domain Data

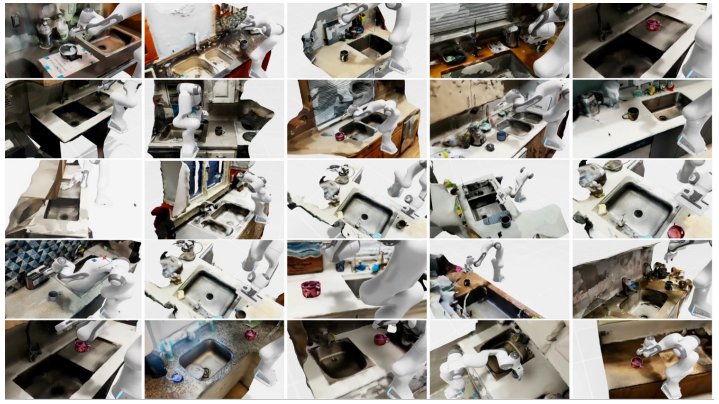
Learn from **Generative Models**



Learn from **Video** Datasets



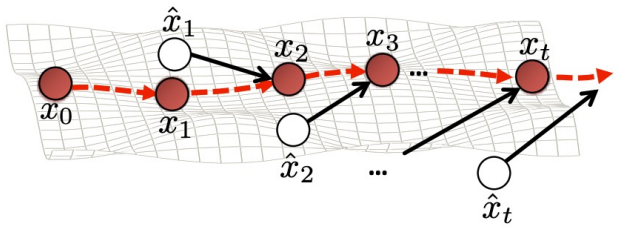
Learn from Scalable **Simulation**



Increasingly “off-domain”

Learning from Off-Domain Data

Learn from **Generative Models**



Learn from **Video** Datasets

Learn from Scalable **Simulation**

Increasingly “off-domain”

Let's start from imitation learning

Humans provide data to learn from through teleoperation

$$\mathcal{D} = \{(s, a, s')_i\}_{i=1}^{N}$$

Behavior Cloning

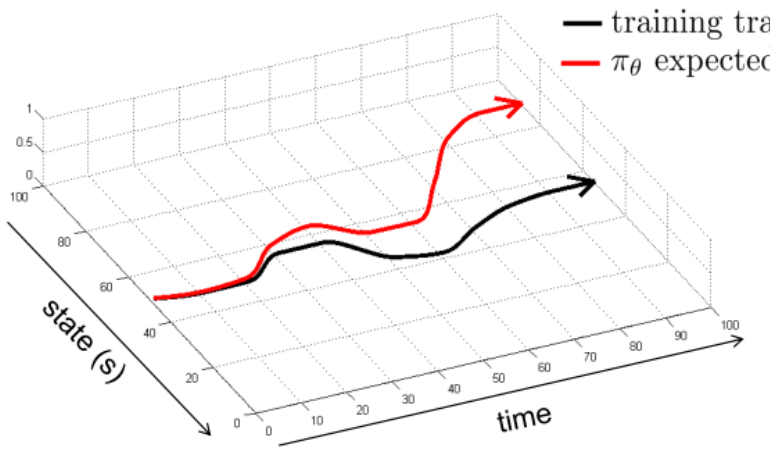
Treat imitation as a supervised learning problem, perform maximum likelihood training

$$\arg\max_{\theta} \mathbb{E}_{(s^*, a^*) \sim \mathcal{D}} \left[\log \pi_\theta(a^* | s^*)\right]$$

Why does imitation learning fail? - Sequential Reasoning

Learning sequential behaviors from data ≠ Supervised Learning

- We treated a sequential problem as a one-step problem



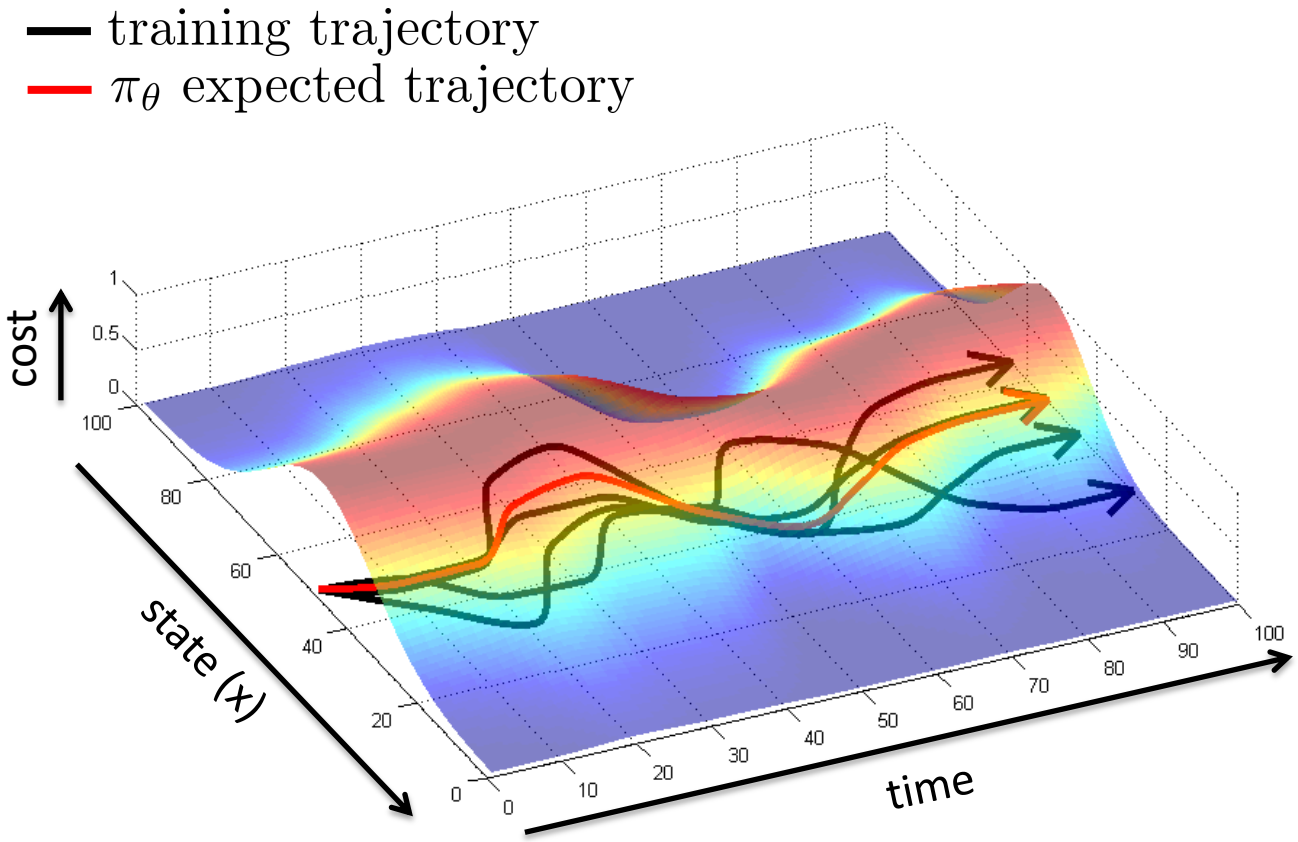
Compounding error!

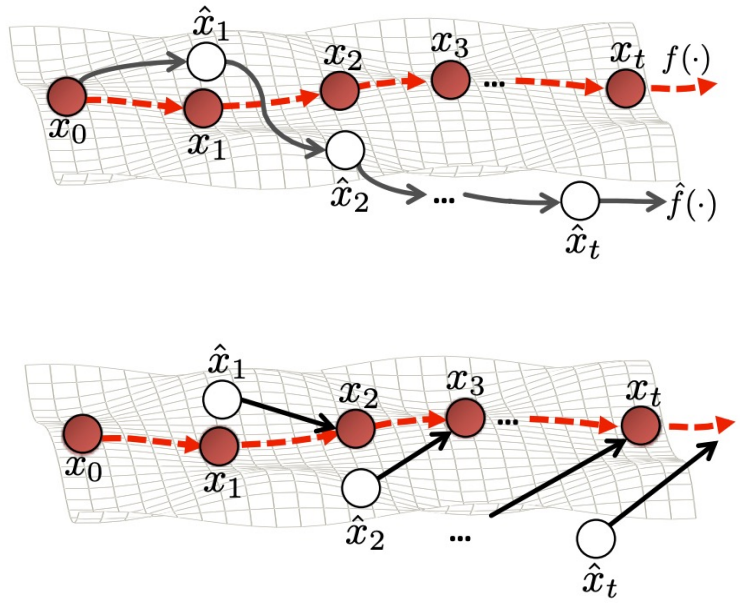
$$\arg\max_\theta \mathbb{E}_{(s^*,a^*)\sim\mathcal{D}}\left[\log \pi_\theta(a^*|s^*)\right] \qquad \mathbb{E}_{(s,a)\sim\rho(\pi)}\left[1(a=a^*)\right]$$

(1-step objective under data) (multi-step objective under policy)

Not the same!

How can we address compounding error?





Insight: Avoid compounding error through corrective data

But where does this data come from?

Corrective Labels from Humans via Dataset Aggregation

Often used idea: use human labelers to provide additional corrective data!

DAgger: **D**ataset **A**ggregation

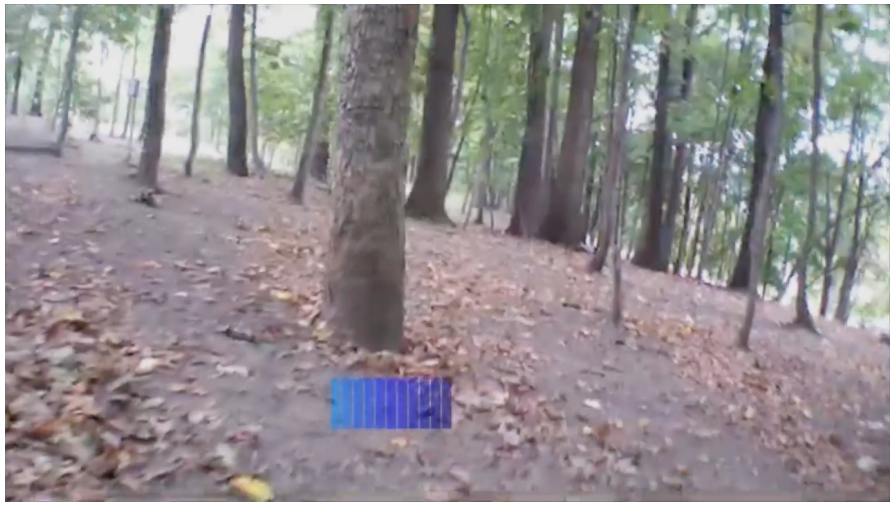
Train policy $\pi_\theta(a|s)$ on current expert dataset $\mathcal{D}$ via behavior cloning

Run $\pi_\theta(a|s)$ in the env to get dataset $\mathcal{D}_\pi = \{s_0^i, s_1^i, \ldots, s_H^i\}_{i=1}^N$

Ask human to label $\mathcal{D}_\pi$ with optimal corrective actions

Aggregate corrected dataset $\mathcal{D} = \mathcal{D} \cup \mathcal{D}_\pi$

Expensive and tedious for people to collect!



A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning, Ross et al '11

Can we avoid expensive online data collection/labeling?

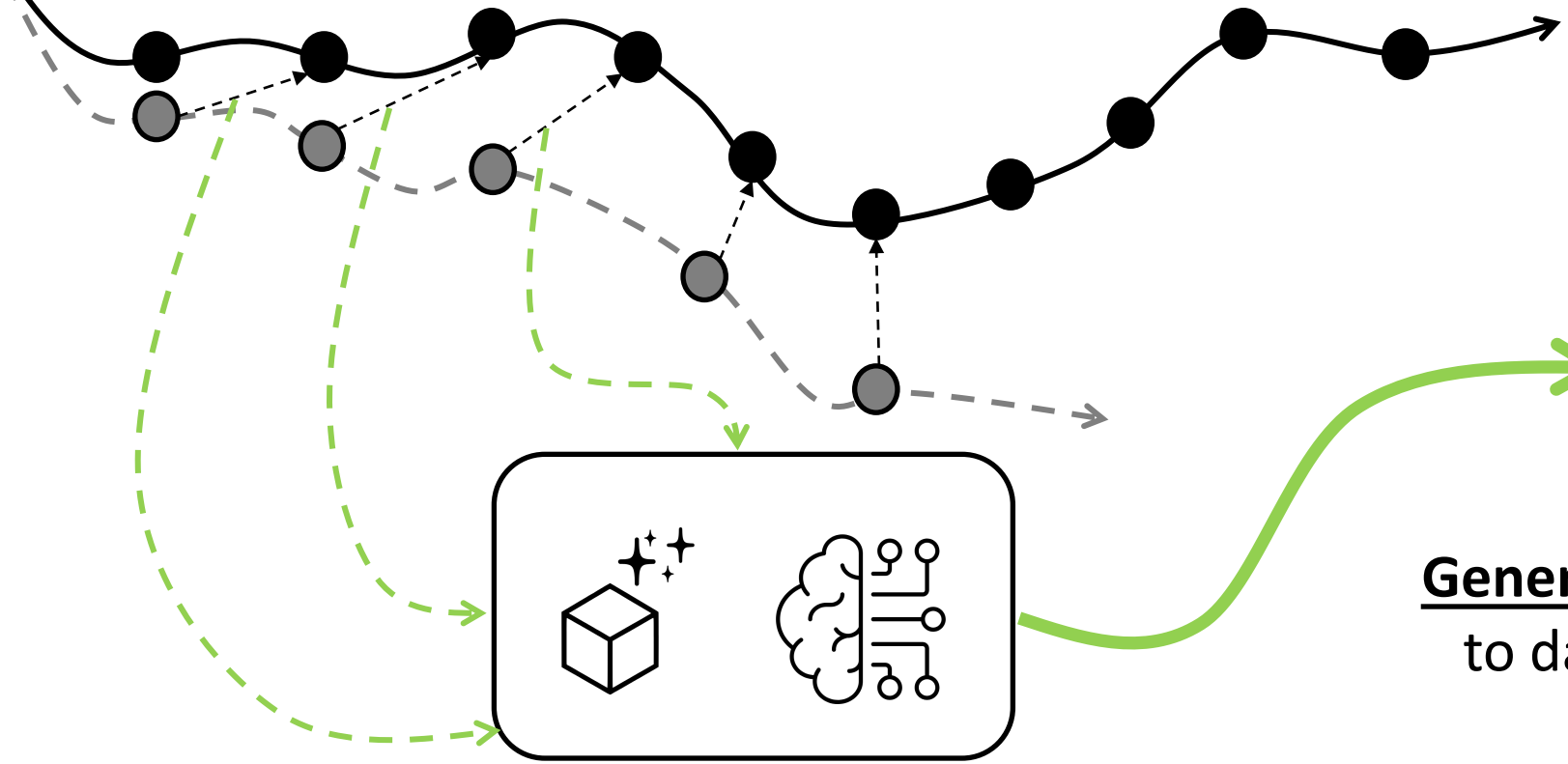
How can we find corrective labels without an expensive human in the loop and online data collection?

Liyiming Ke

Abhay Deshpande

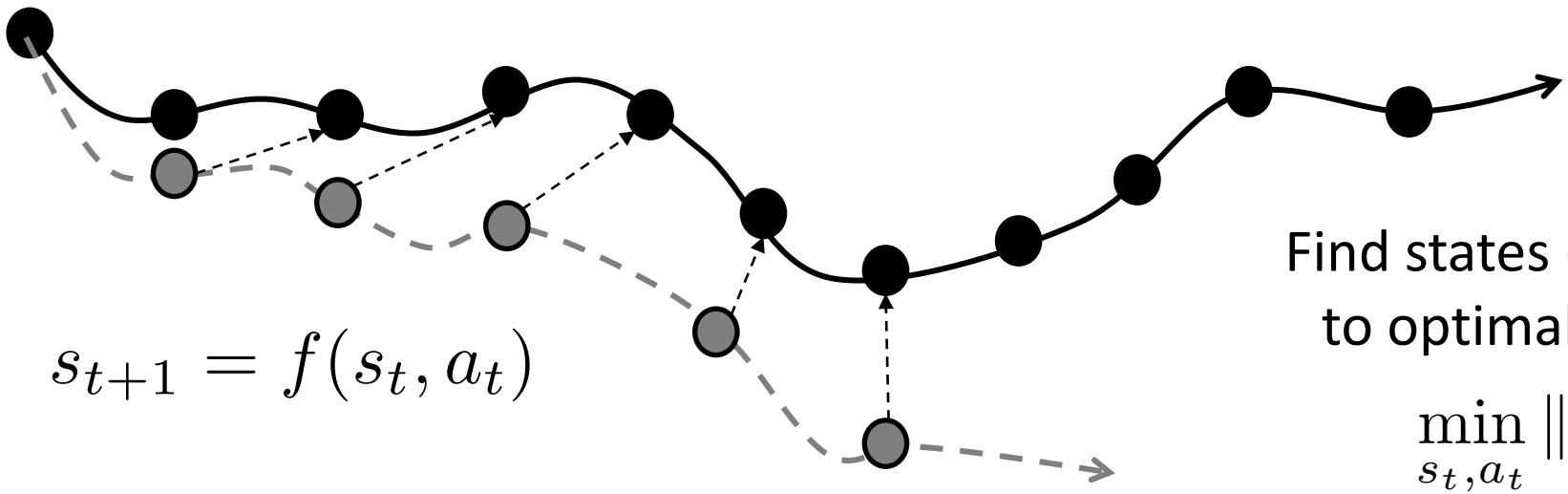
Yunchu Zhang

Generative models!



Generate corrective labels to dataset for imitation

What data **should** be generated?



$$s_{t+1} = f(s_t, a_t)$$

Find states (s_t), actions (a_t) that lead back to optimal states under true dynamics

$$\min_{s_t, a_t} \|s^*_{t+1} - f(s_t, a_t)\| \leq \epsilon$$

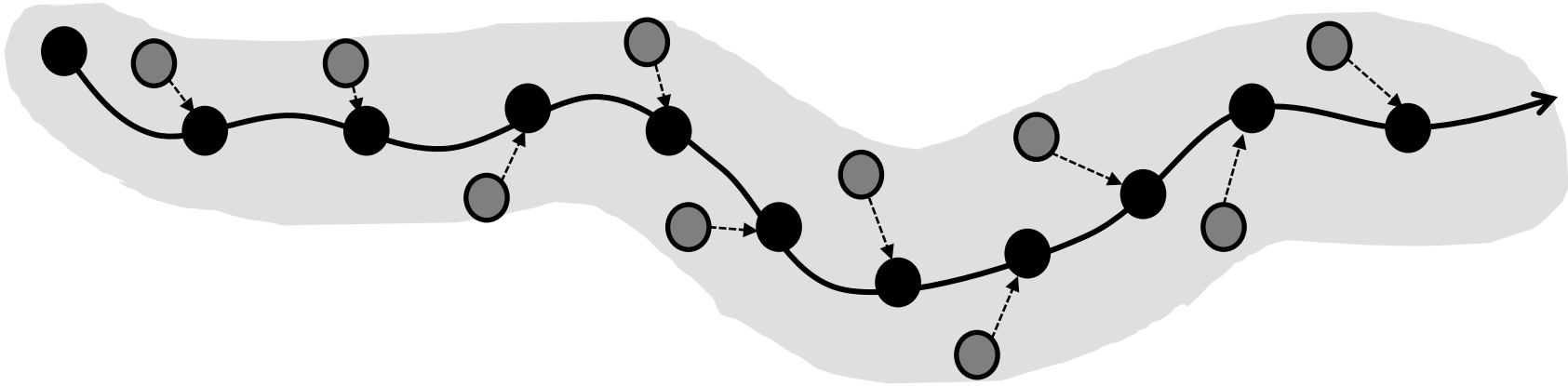
Easy with known dynamics

Intuition: find labels to bring OOD states back in distribution

But models are unknown! ☹

CCIL: Continuity-based Data Augmentation for Corrective Imitation Learning, Ke et al '24
Data Efficient Behavior Cloning for Fine Manipulation via Continuity-based Corrective Labels, Deshpande et al '24

Generating Corrective Labels with **Learned** Models



Ok models are unknown, let's learn them!

$$\min_{\hat{f}} \mathbb{E}_{(s_t, a_t, s_{t+1}) \sim \mathcal{D}} \left[\| \hat{f}(s_t, a_t) - s_{t+1} \|_2 \right]$$

But learned dynamics $\hat{f}_\phi$ are not globally accurate?

$$\| s^*_{t+1} - \hat{f}_\phi(s_t, a_t) \| \leq \epsilon$$

Under approximately Lipschitz smooth models, trust models around training data

Find states (s_t), actions (a_t) that lead back to optimal states under ~~true~~ learned dynamics, **where learned dynamics can be trusted**

$$\min_{s_t, a_t} \| s^*_{t+1} - \hat{f}_\phi(s_t, a_t) \| \leq \epsilon \longleftarrow \text{Corrective label}$$

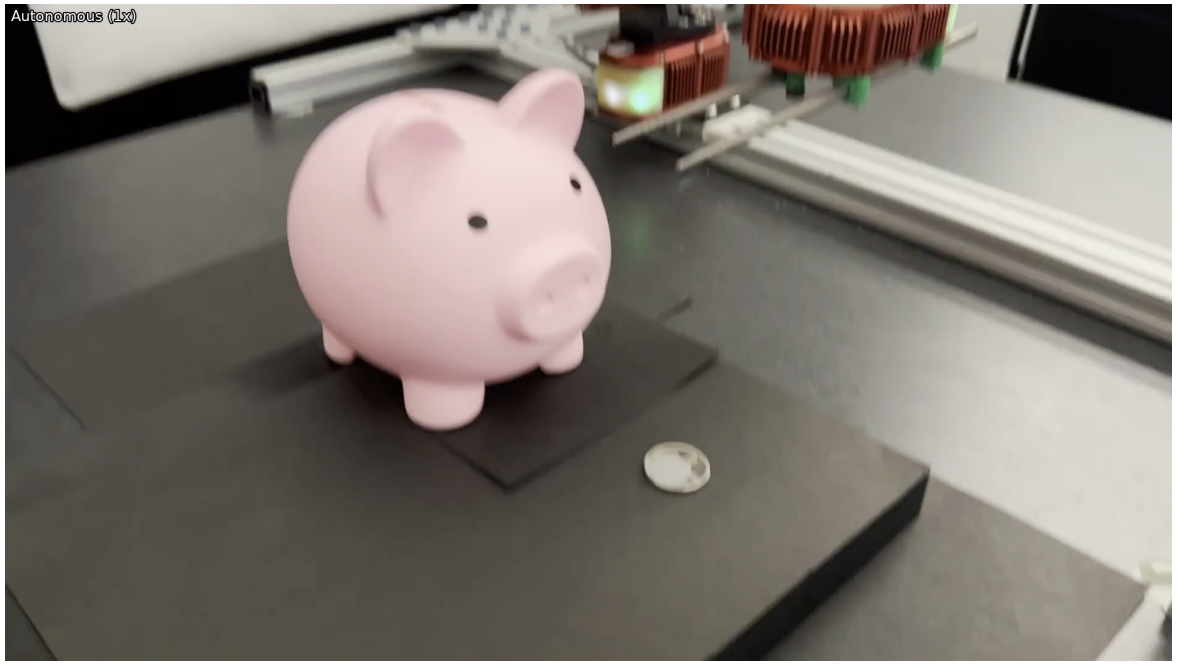
$$\text{s.t} \quad \| s^*_t - s_t \| \leq \epsilon_1, \| a^*_t - a_t \| \leq \epsilon_2 \longleftarrow \text{Close to data}$$

CCIL: Continuity-based Data Augmentation for Corrective Imitation Learning, Ke et al '24
Data Efficient Behavior Cloning for Fine Manipulation via Continuity-based Corrective Labels, Deshpande et al '24

How well does generating corrective labels work?

With corrective labels

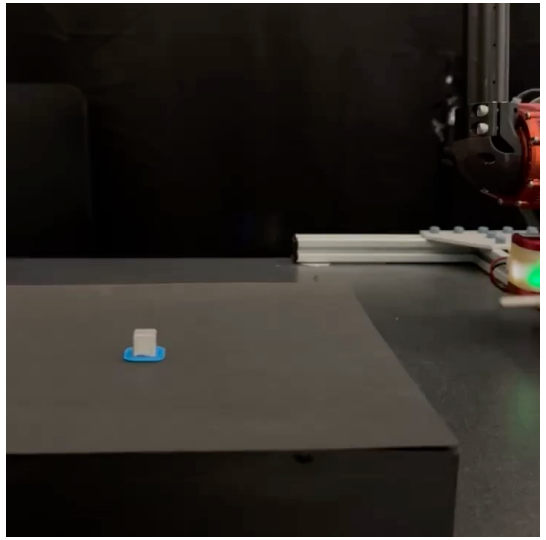
Without corrective labels

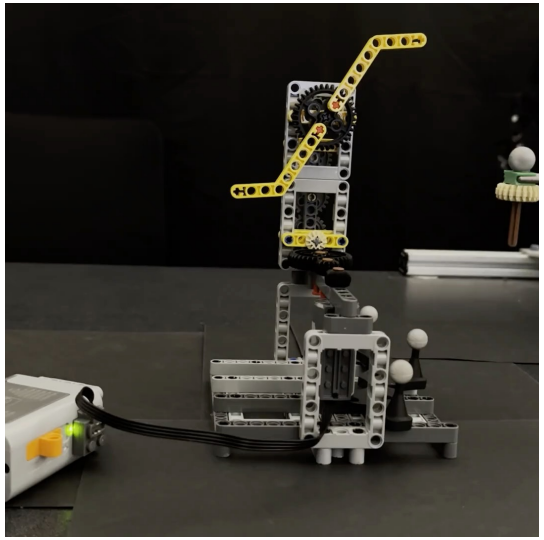


CCIL: Continuity-based Data Augmentation for Corrective Imitation Learning, Ke et al '24
Data Efficient Behavior Cloning for Fine Manipulation via Continuity-based Corrective Labels, Deshpande et al '24

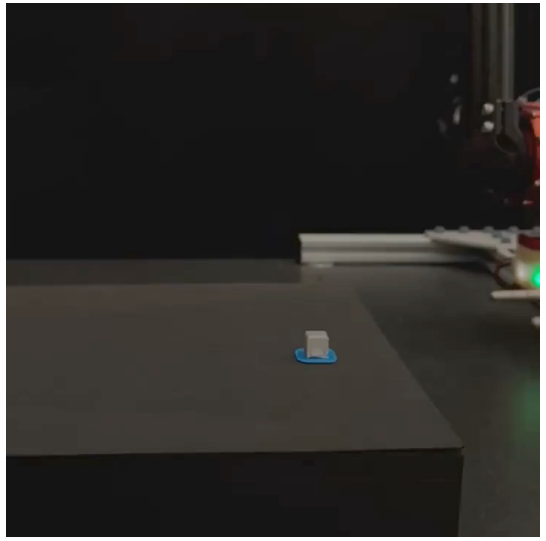
How well does generating corrective labels work?

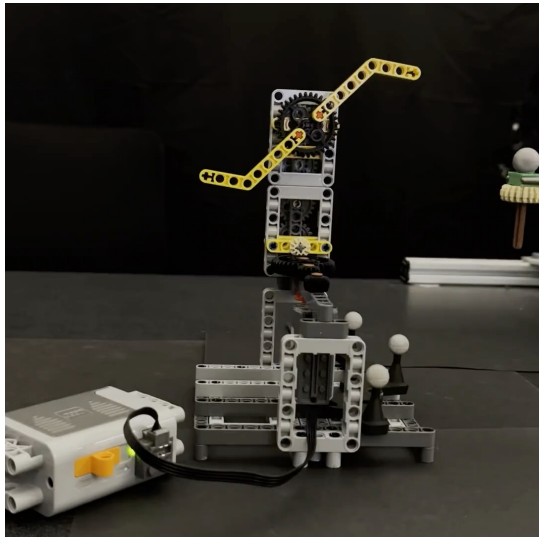
With corrective labels

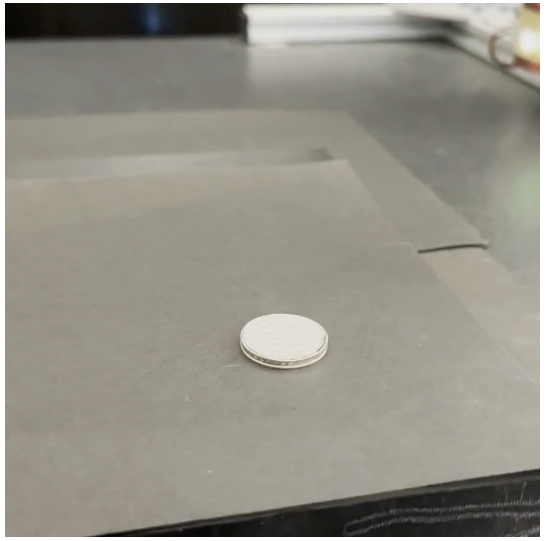




Without corrective labels

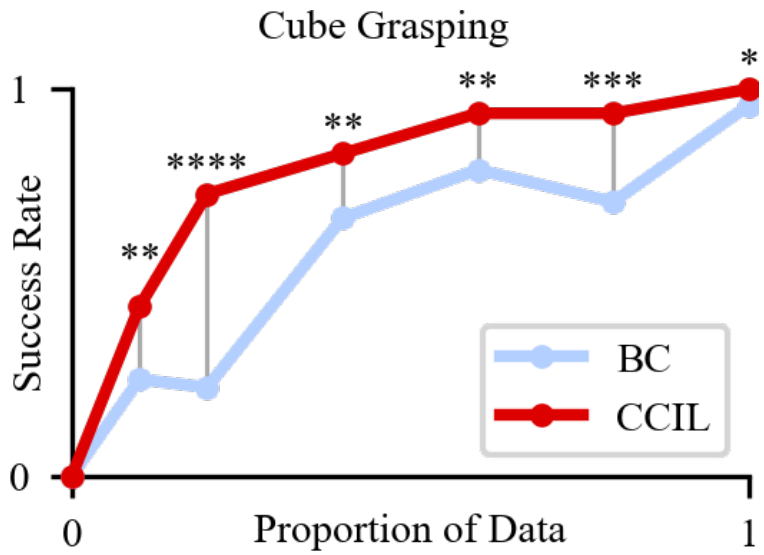


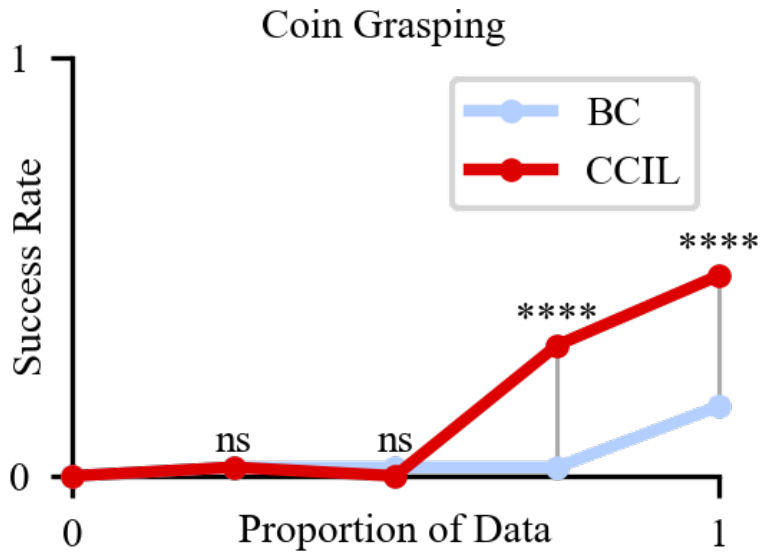




Let's get quantitative

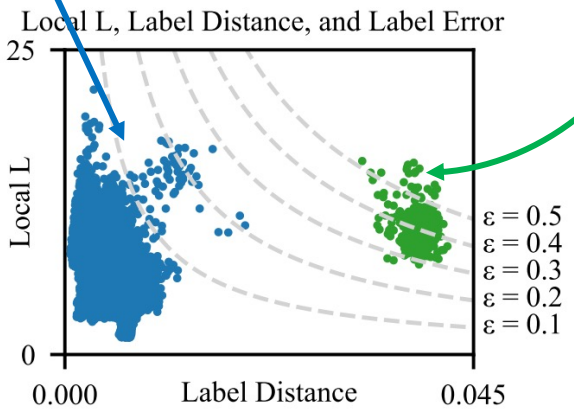
Bigger gains in low data regimes



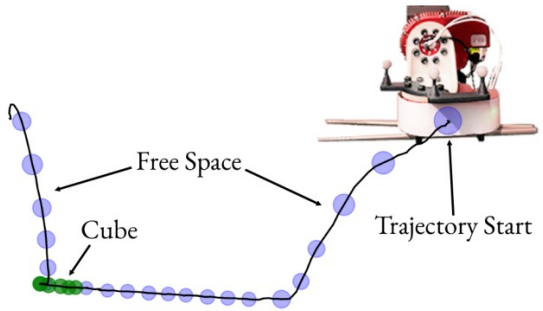


Low error

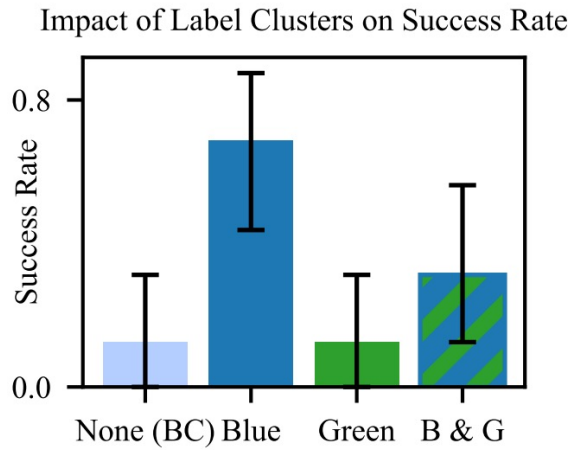
High error



(a) Clusters reveal dynamics regimes.



(b) High label rejection clusters around contact.



(c)

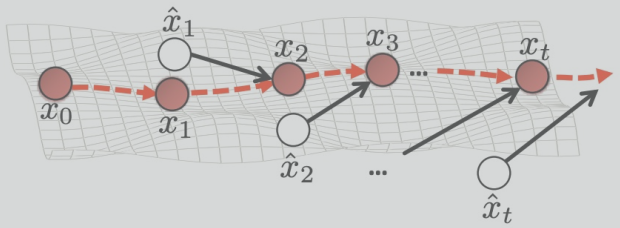
Skip training on areas with high Lipschitz

Insights: Learning from In-Domain Data

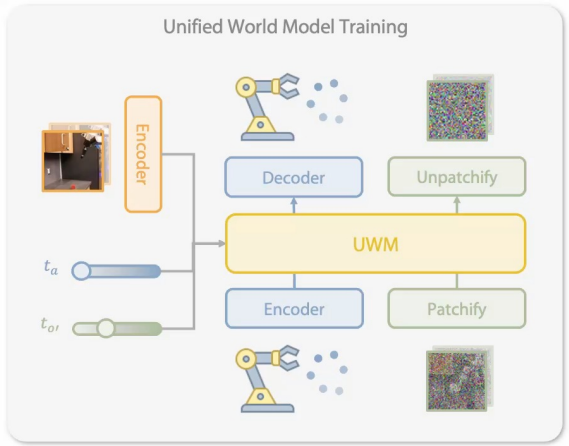
1. Generative models can produce targeted data, but only where they can be trusted

Learning from Off-Domain Data

Learn from **Generative Models**



Learn from **Video** Datasets

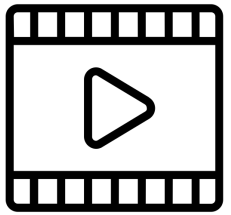


Learn from Scalable **Simulation**

Increasingly “off-domain”

Where else might we find abundant off-domain data?

Large video datasets contain information about dynamics and semantics



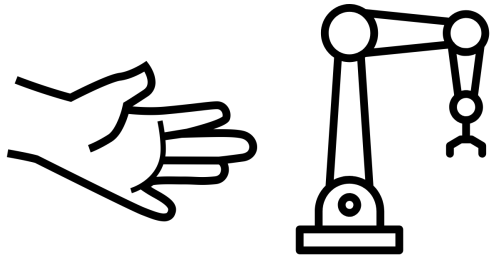
Cheaper and more scalable than on-robot data

Can we make this useful for robotics?

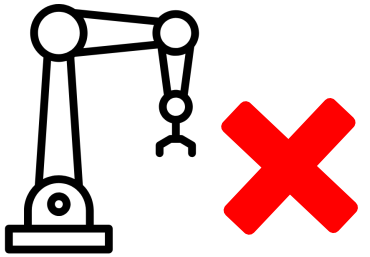
Can we directly use this data for policy learning?

Not directly clear how to learn policies from large-scale video data

Action-free



Potentially suboptimal

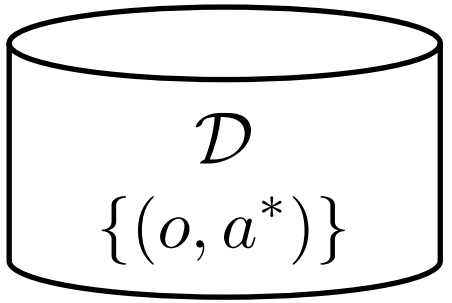


How do we directly learn $\pi^*(a|o)$?

From Policy Learning to World Modeling

Policy Learning

$$\pi^*(a|o)$$

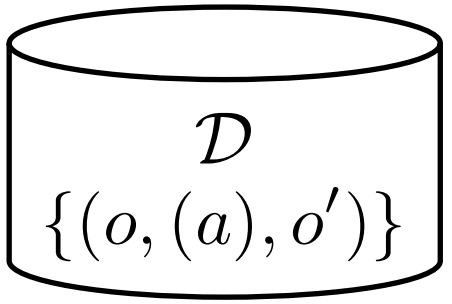


Requires optimal data with actions

What happens when this data is unavailable?

World Modeling

$$p(o'|o)$$
$$p(o'|o, a)$$



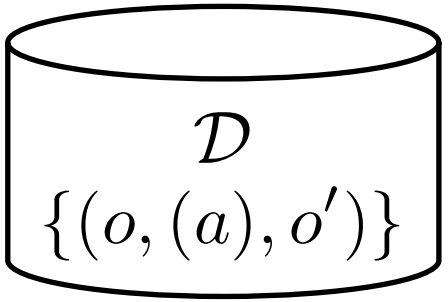
Data communicates temporal **dynamics**

("How the world works")

Trained via supervised learning on transition data!

World Models in Popular Media

$$p(o'|o)$$
$$p(o'|o,a)$$



Data communicates temporal **dynamics**

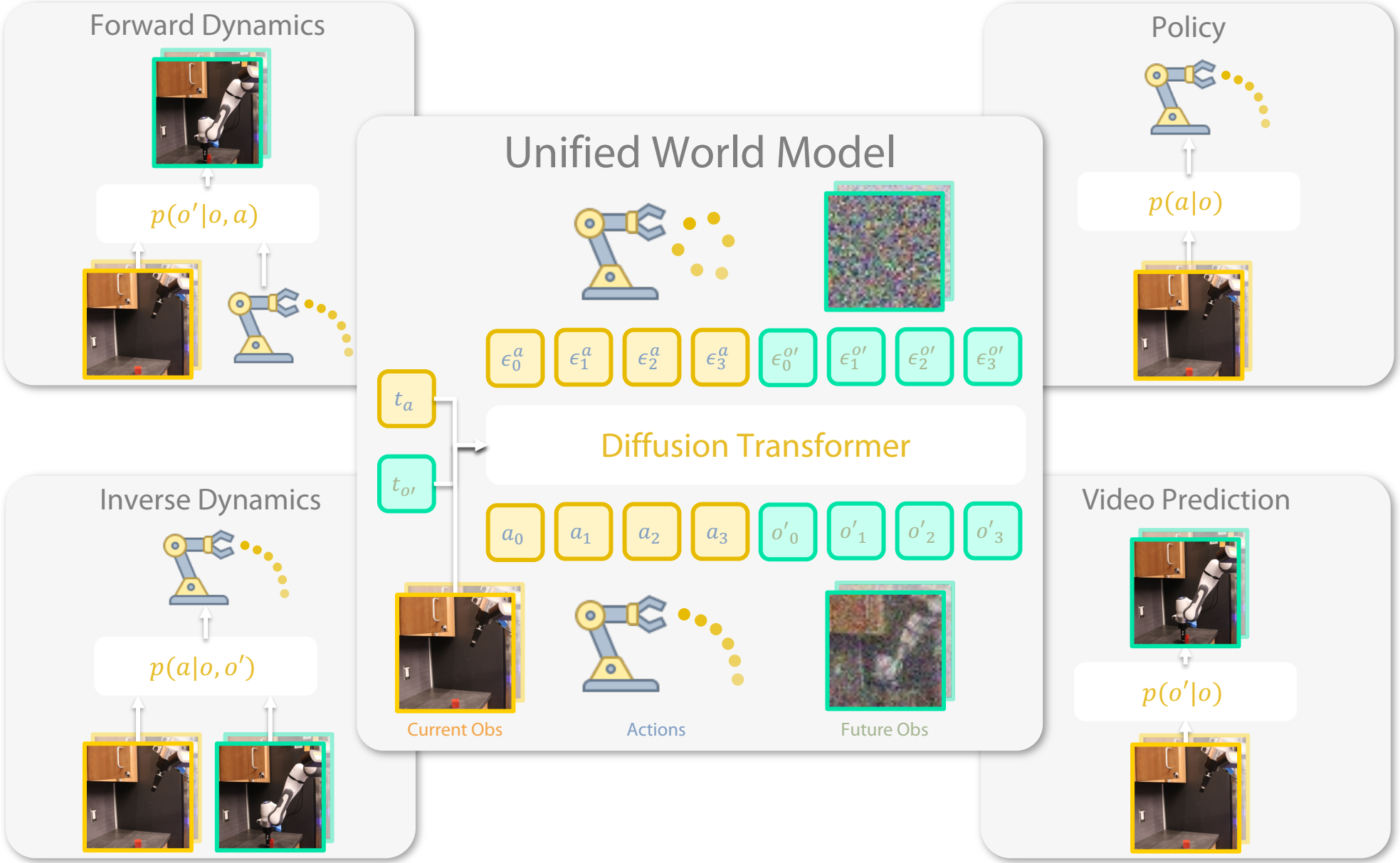
("How the world works")

Veo3

Genie

But how can these insights be **unified** with control?

Bridging the gap between world modeling and policy learning

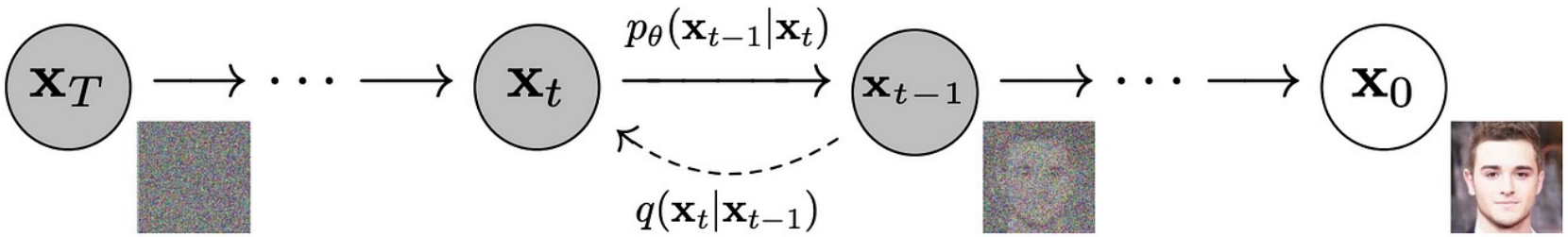


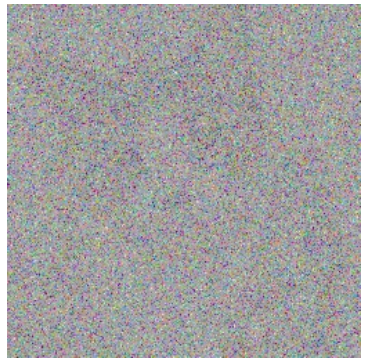
Chuning Zhu

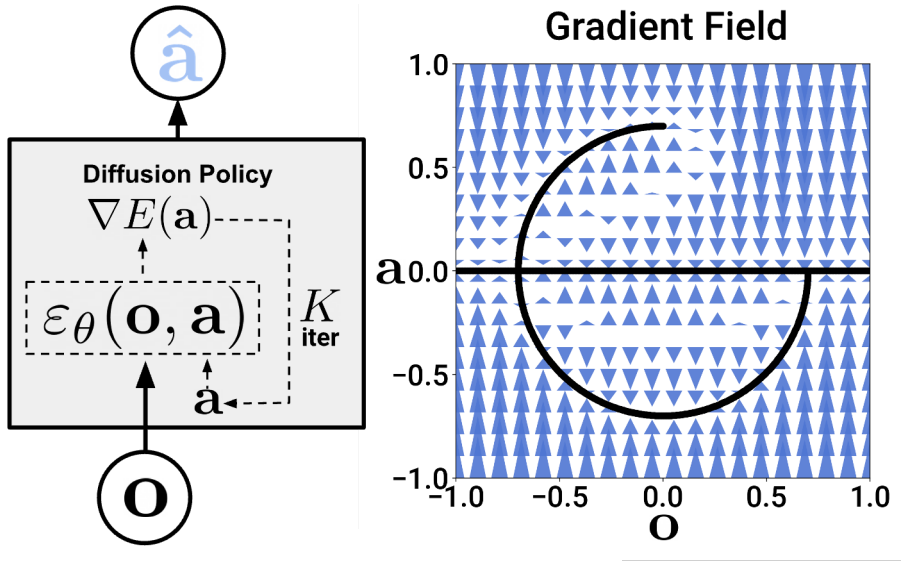
Background: Diffusion Models for Robotics

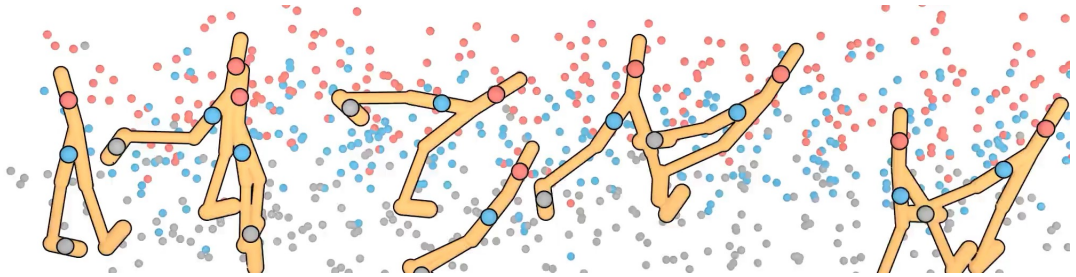
Forward: data to noise

Reverse: noise to data









Difference in variable being denoised

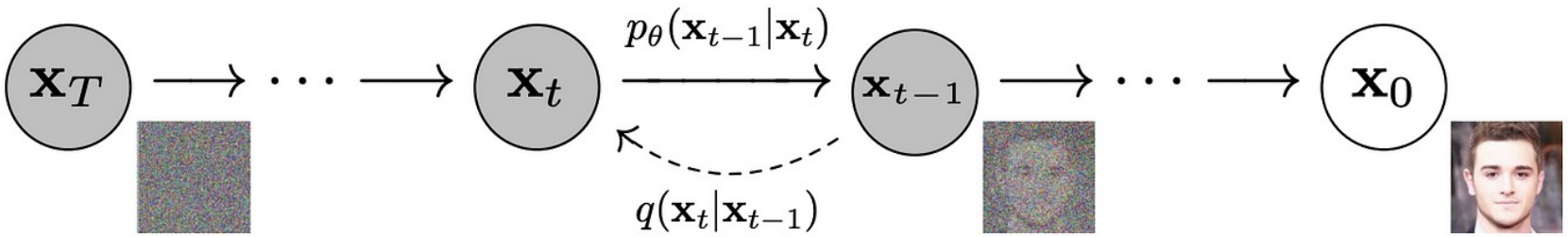
Diffusion-based policies – $p(a|o)$

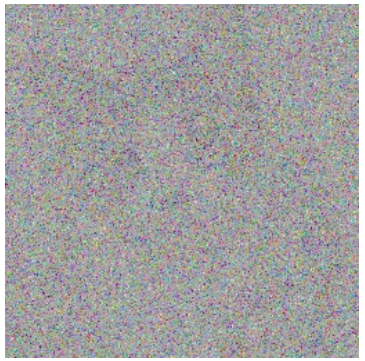
Diffusion-based dynamics – $p(o'|o)$

Joint Modeling of Future Images and Actions

Forward: data to noise

Reverse: noise to data





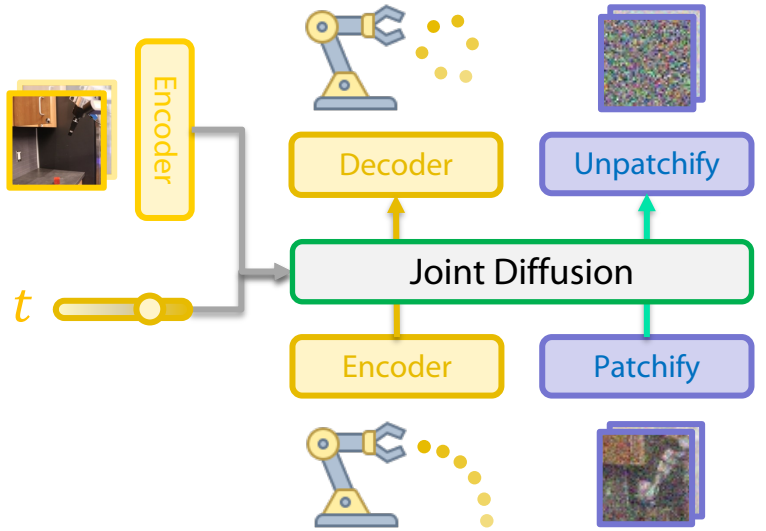
Conditional diffusion process over (o', a | o)

$$s_\theta(o, a_t, o'_t, t)$$

Forward: action, future obs to noise

Reverse: noise to action, future obs

Conditional on current obs

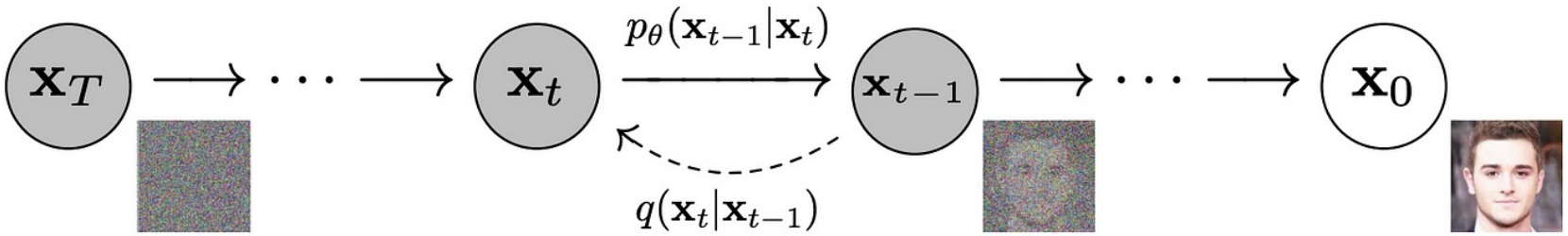


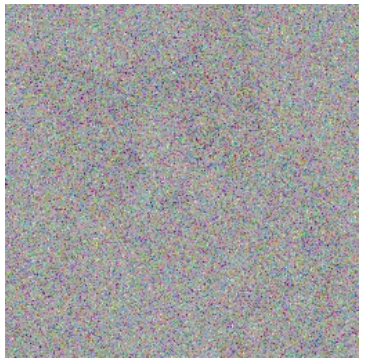
Not clear how to access policy $\pi(a|o)$ or world model $p(o'|o)$

Flexible Conditioning and Marginalization with Diffusion

Forward: data to noise

Reverse: noise to data





(Marginalization)

$$v_\theta(x_t, y_t, t) \longrightarrow v_\theta(x_t, y_t, t_x, t_y)$$

$t_x = T$ $t_y = T$

Denoising both x_t and y_t together

$$p(x) \qquad p(y)$$

$$p(x, y)$$

$$p(x|y) \qquad p(y|x)$$

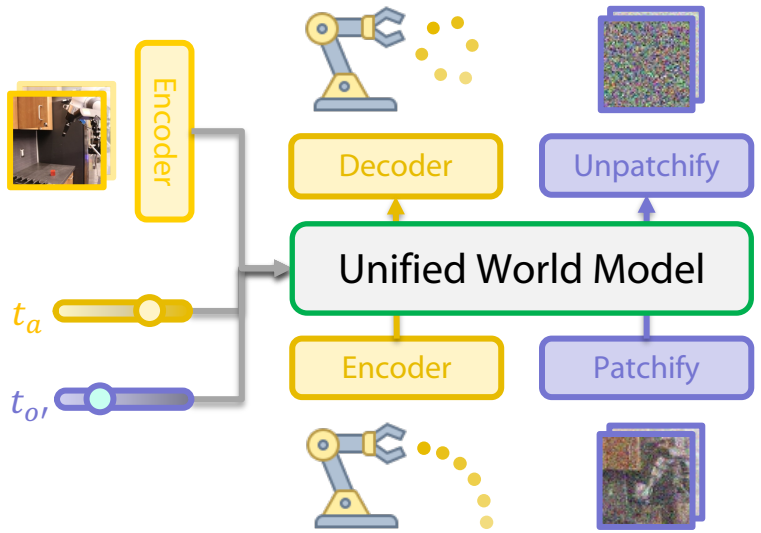
$t_x = 0$ $t_y = 0$

(Conditioning)

Mathematical Tool: Decoupled Timestep Diffusion Models

Key idea: noise as masking with decoupled time-steps between obs, actions

Decoupled time steps



Single model can be queried in different ways:

- Policy $\pi(a|o)$
- Dynamics model $p(o'|o,a)$
- World model $p(o'|o)$
- Inverse dynamics $p(a|o,o')$

t = T, marginalization

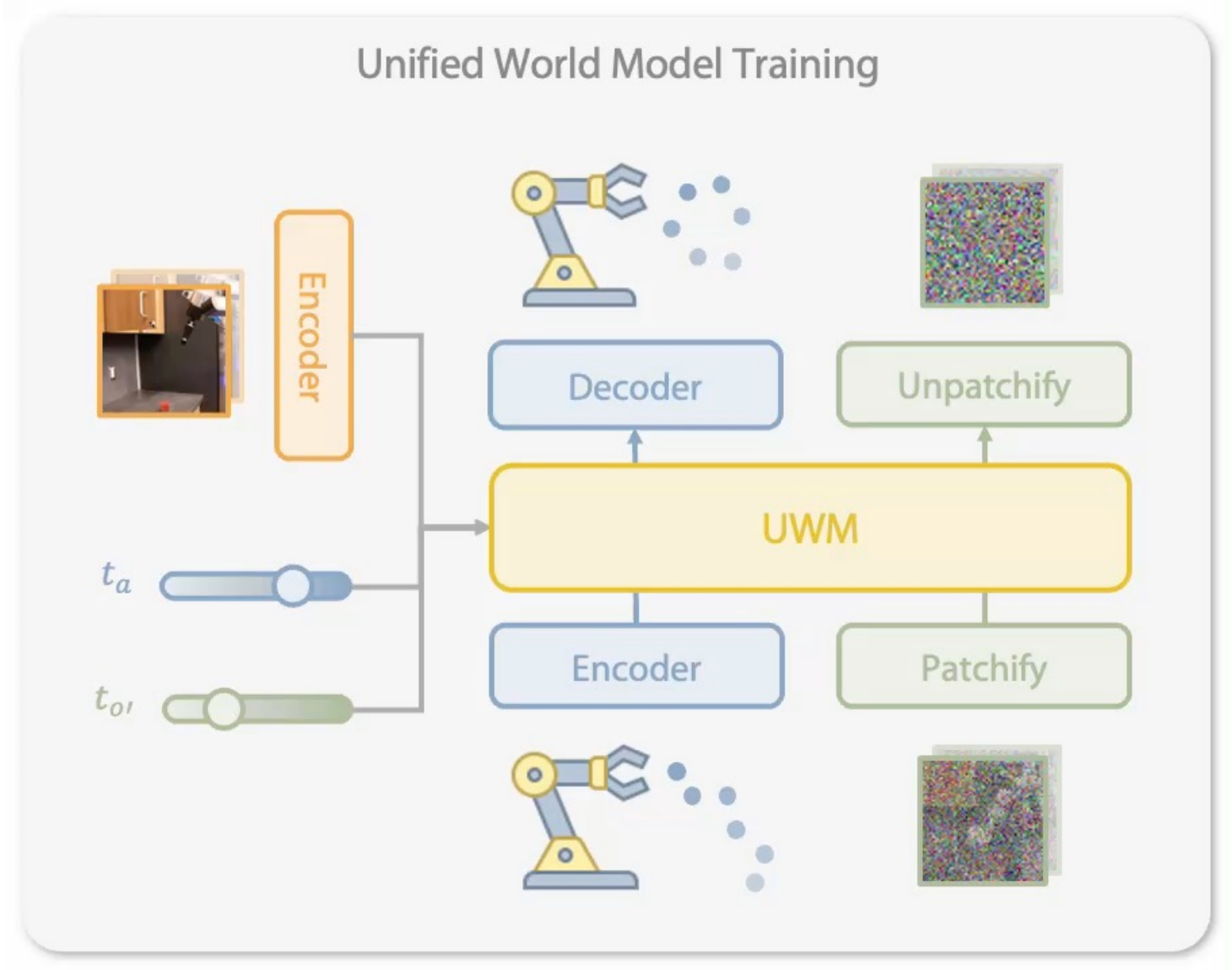
Intuition: variable **noised** out

t = 0, conditioning

Intuition: variable **fixed** to a value

Setting time-steps carefully gives access to different models from joint $p(o', a|o)$!

Unified World Models: Training



Unified World Models: Inference

DROID Experiments

Pretraining Dataset (DROID)





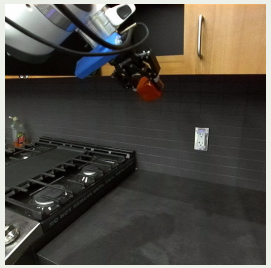
Cotraining Dataset (DROID)





Finetuning Datasets


Stack-Bowls

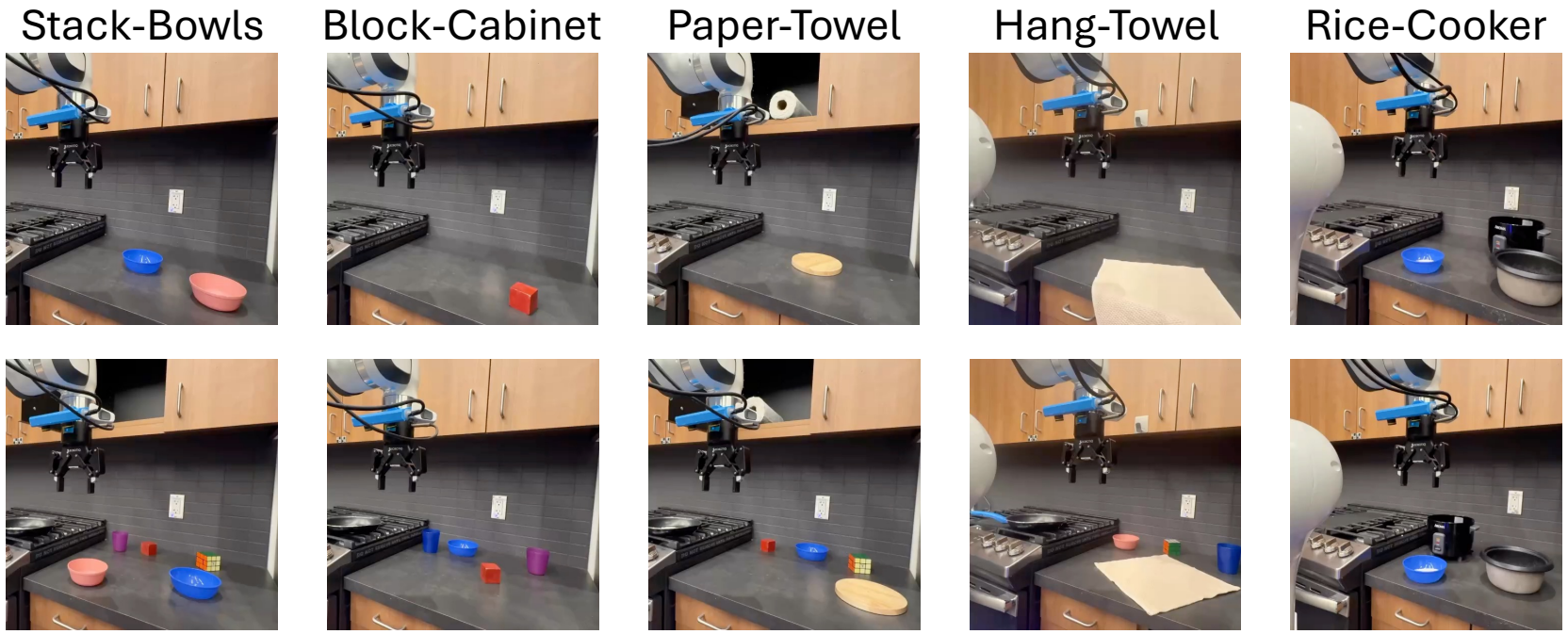

Block-Cabinet

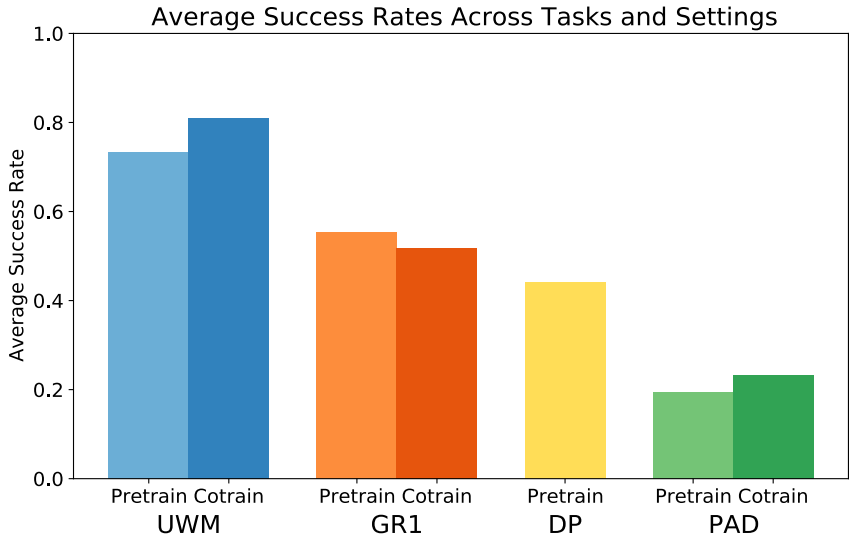

Paper-Towel


Hang-Towel


Rice-Cooker

DROID Experiments

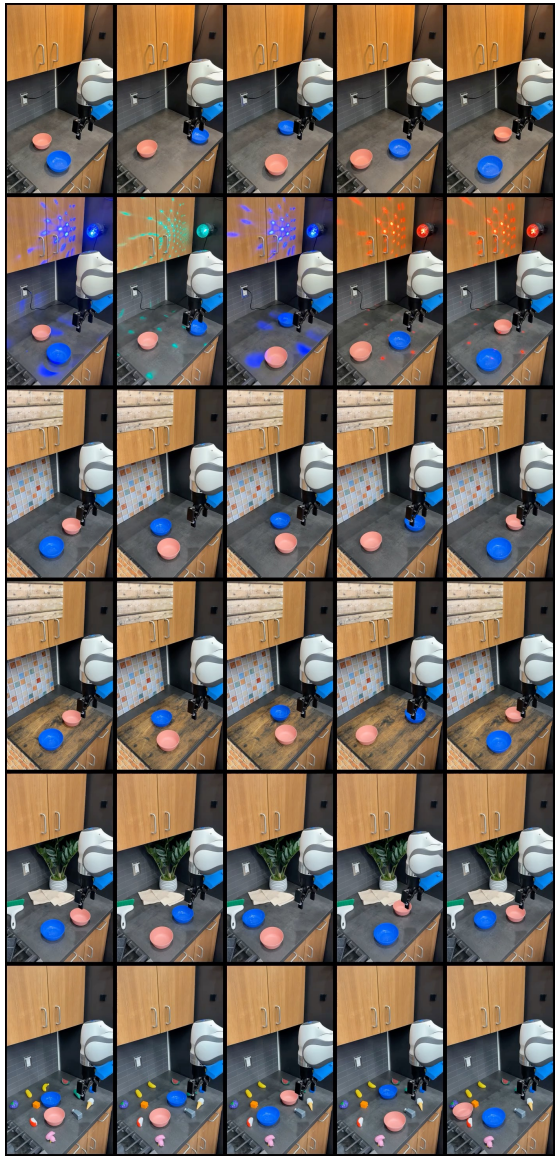




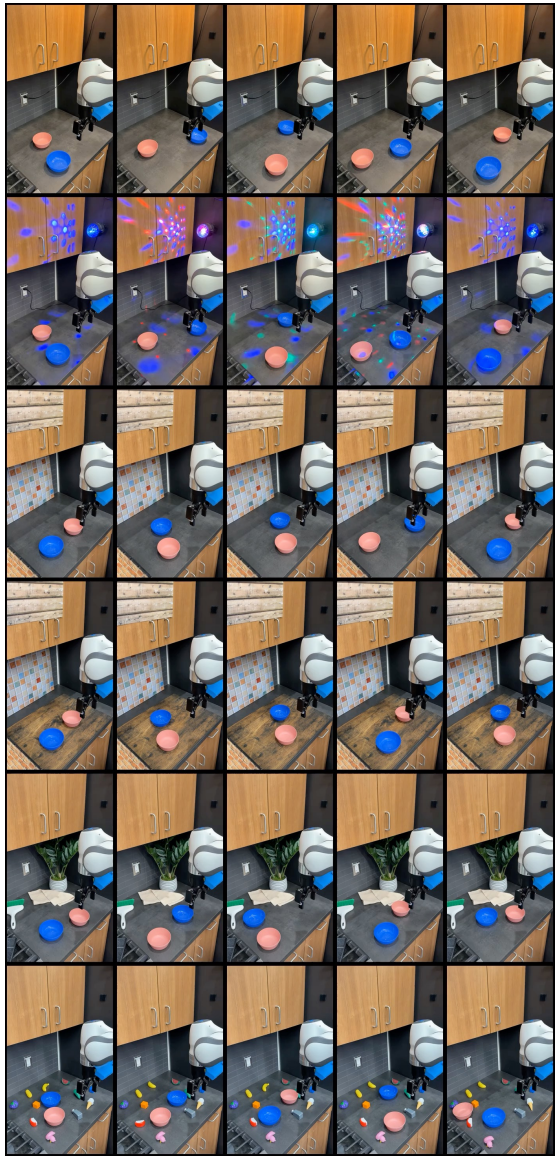
- UWM benefits from pretraining on robot data more than baselines
- Cotraining on videos improves performance

DROID OOD Experiments

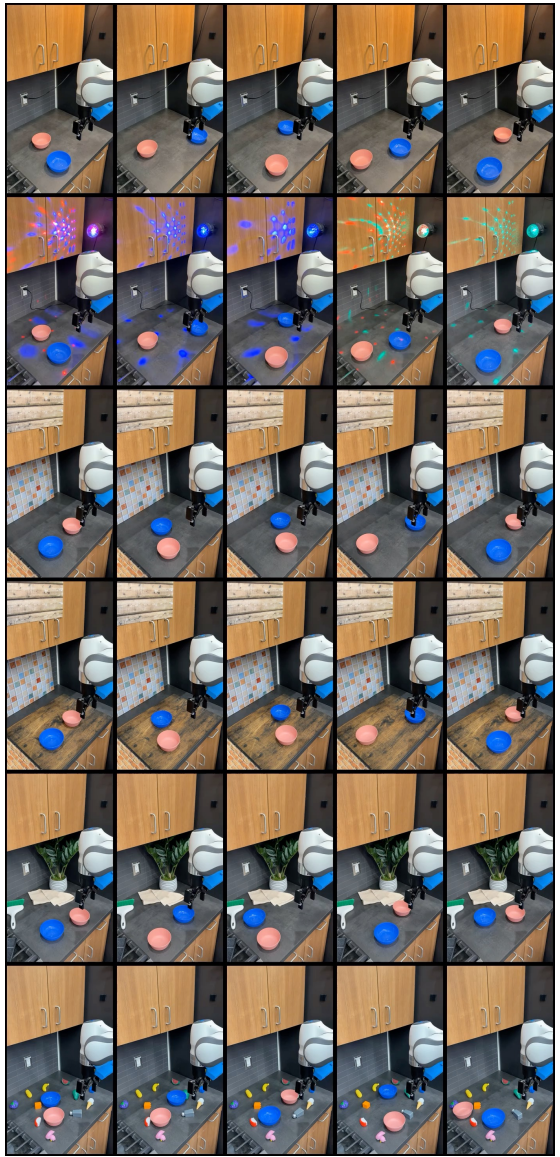
UWM Cotrained (21/30)



UWM Pretrained (15/30)

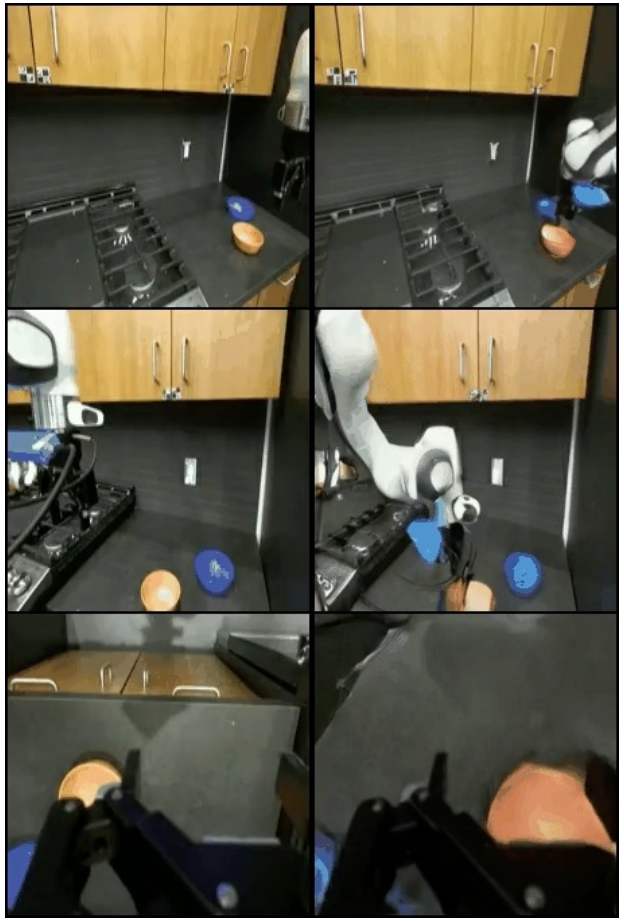


DP (12/30)



Forward Dynamics Visualization

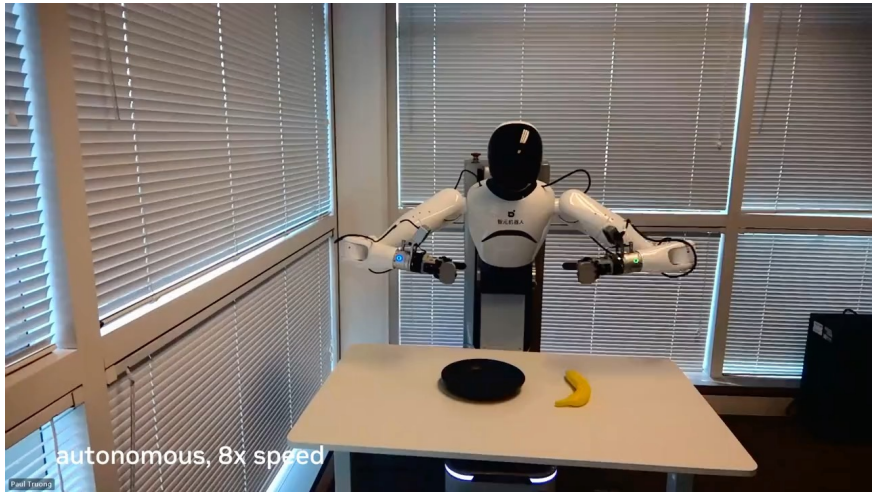
Ground Truth Prediction

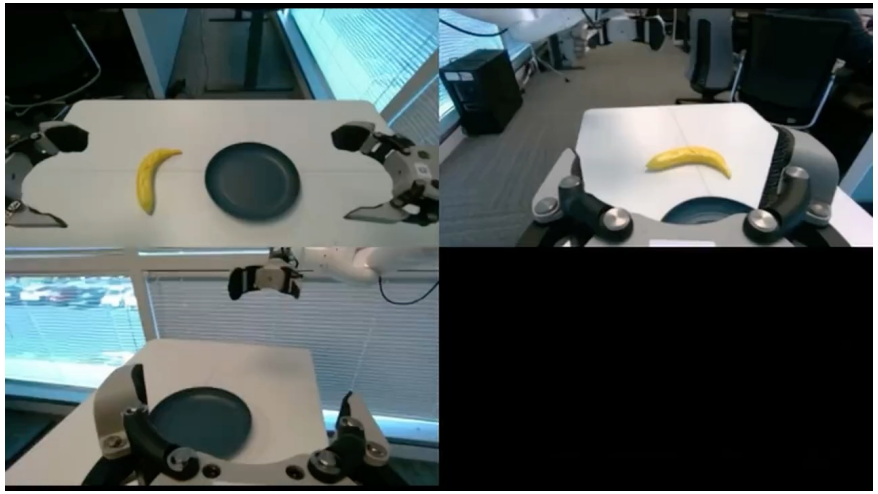


Ground Truth Prediction

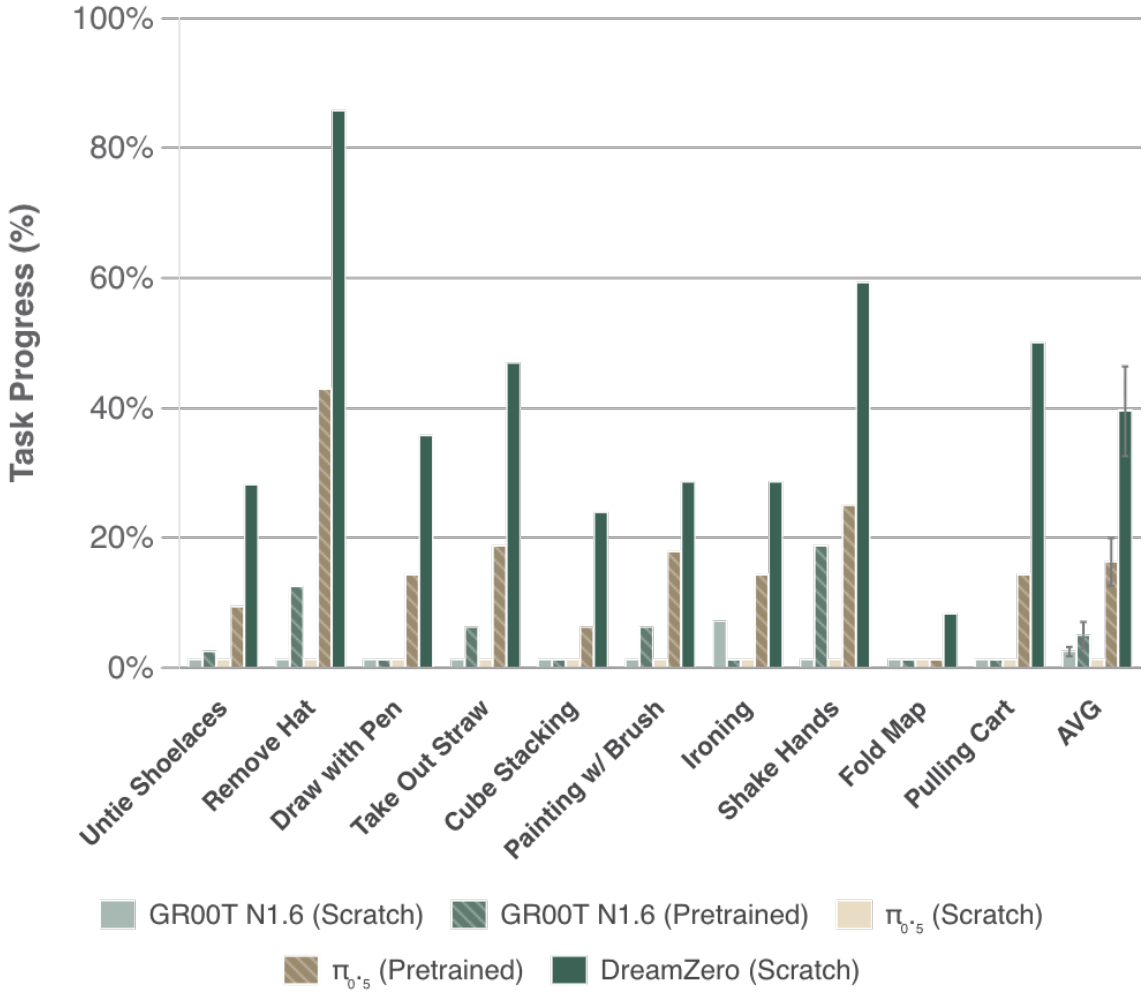
Ground Truth Prediction

What happens when this is scaled up?





Considerable gains for zero-shot performance on new tasks



DreamZero, Ye et al '26

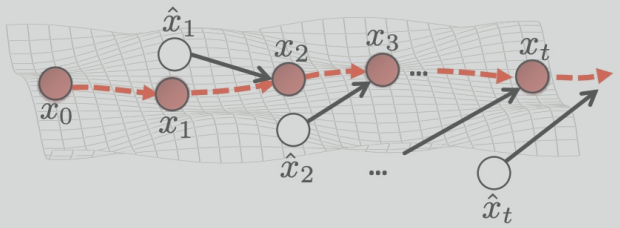
Insights: Learning from Video Data

1. Generative models can produce targeted data, but only where they can be trusted

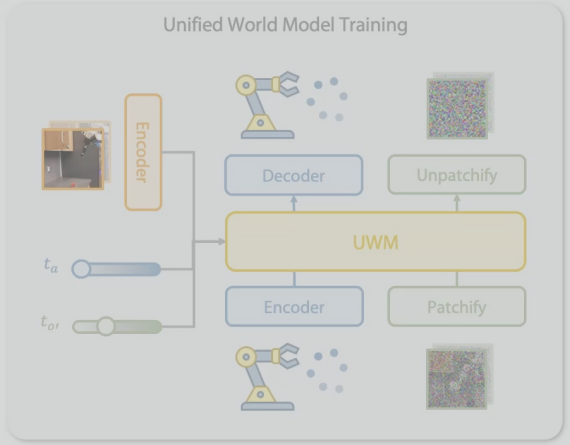
2. Video data can be absorbed by unified models with architectures to allow flexible inference

Learning from Off-Domain Data

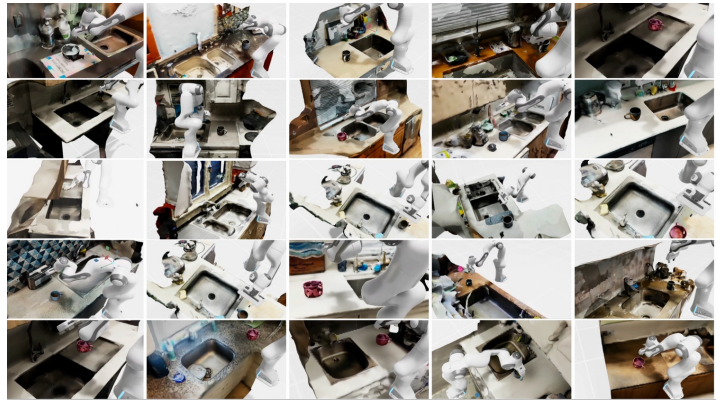
Learn from **Generative Models**



Learn from **Video** Datasets

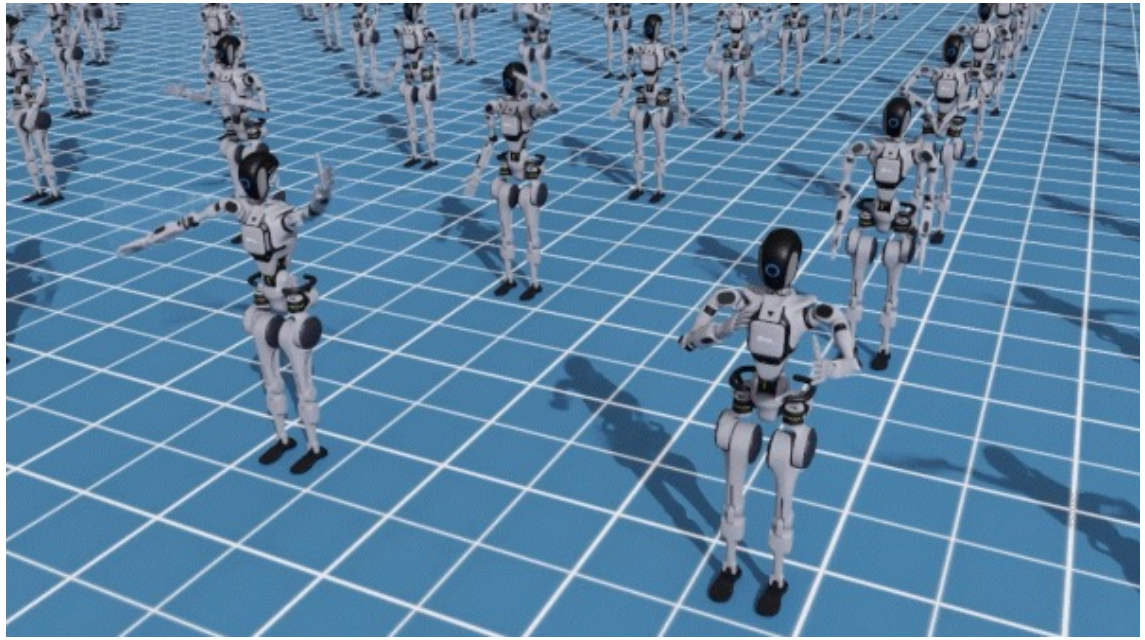


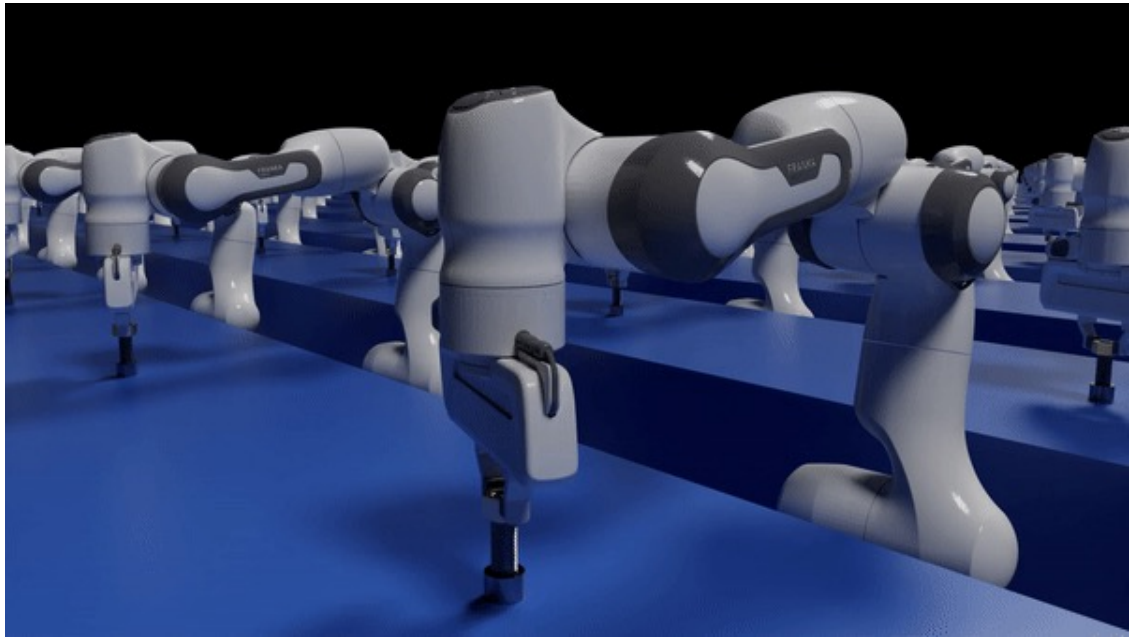
Learn from Scalable **Simulation**



Increasingly “off-domain”

Why use simulation for robot learning?





Pros:

1. Data should scale with compute
2. ~~Should accomplish~~ high coverage
3. Faster than real time data collection

Lots of hidden cost in practice

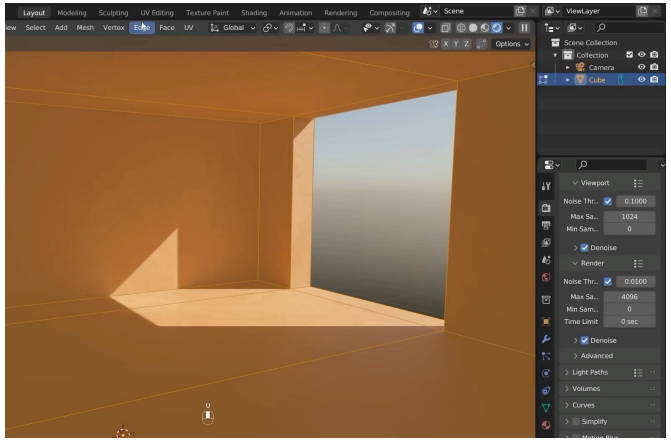
Counterfactuals great for pre-training

Why does sim not scale passively for data generation?

Data generation is not quite autonomous

Where are we spending effort?

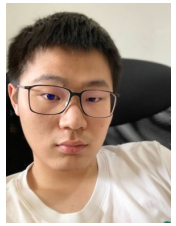
Environment Creation



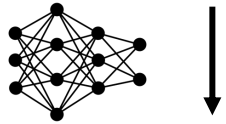
Behavior Generation

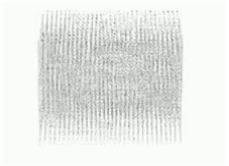
OpenVR: Teleoperation for Manipulation, George et al '23

Real-to-Sim Environment Generation



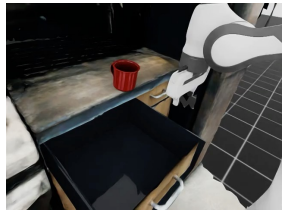
Synthetically generate fully-articulated, interactive scenes from web-scale images/videos



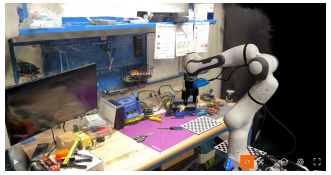


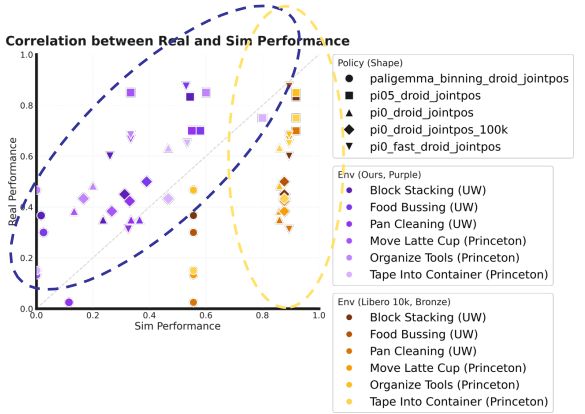
GPT - 4

Data generation for pre-training



Policy evaluation





DRAWER: Digital Reconstruction and Articulation With Environment Realism, Xia et al, CVPR '25

Polaris: Scalable Real-to-Sim Evaluations for Generalist Robot Policies, Jain et al, arXiv '26

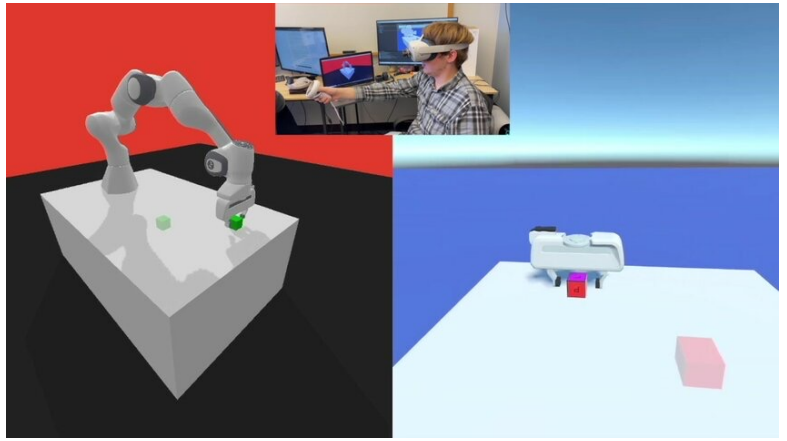
Why does sim not scale passively for data generation?

Data generation is not quite autonomous

Where are we spending effort?

Environment Creation

Behavior Generation



OpenVR: Teleoperation for Manipulation, George et al ‘23

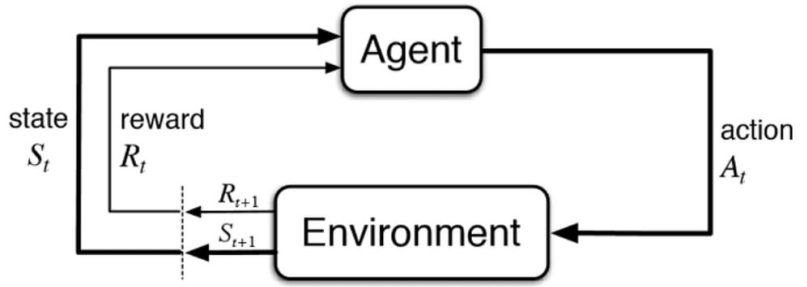
How do we generate behavioral data in simulation?

Human Data Collection

Hard to scale for high-coverage behavior!

Human reward design + RL

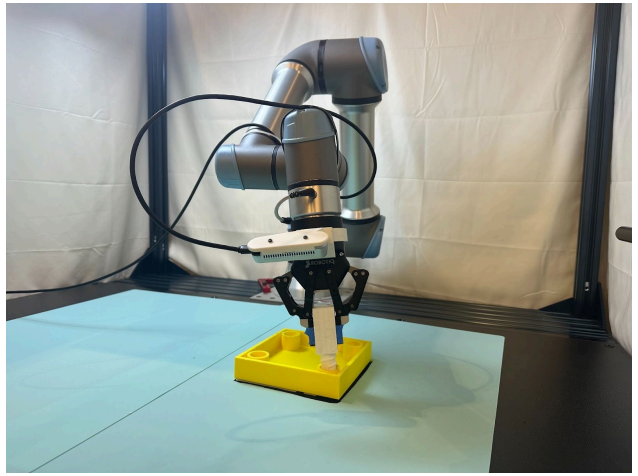
```
def compute_reward(object_rot, goal_rot, object_angvel, object_pos, fingertip_pos):
    # Rotation reward
    rot_diff = torch.abs(torch.sum(object_rot * goal_rot, dim=1) - 1) / 2
-   rotation_reward_temp = 20.0
+   rotation_reward_temp = 30.0                                          Changing hyperparameter
    rotation_reward = torch.exp(-rotation_reward_temp * rot_diff)

    # Distance reward
+   min_distance_temp = 10.0
    min_distance = torch.min(torch.norm(fingertip_pos - object_pos[:, None], dim=2), dim=1).values
-   distance_reward = min_distance
+   uncapped_distance_reward = torch.exp(-min_distance_temp * min_distance)
+   distance_reward = torch.clamp(uncapped_distance_reward, 0.0, 1.0)    Changing functional form

-   total_reward = rotation_reward + distance_reward
+   # Angular velocity penalty                                           Adding new component
+   angvel_norm = torch.norm(object_angvel, dim=1)
+   angvel_threshold = 0.5
+   angvel_penalty_temp = 5.0
+   angular_velocity_penalty = torch.where(angvel_norm > angvel_threshold,
+       torch.exp(-angvel_penalty_temp * (angvel_norm - angvel_threshold)), torch.zeros_like(angvel_norm))
+
+   total_reward = 0.5 * rotation_reward + 0.3 * distance_reward - 0.2 * angular_velocity_penalty

    reward_components = {
        "rotation_reward": rotation_reward,
        "distance_reward": distance_reward,
+       "angular_velocity_penalty": angular_velocity_penalty,
    }

    return total_reward, reward_components
```

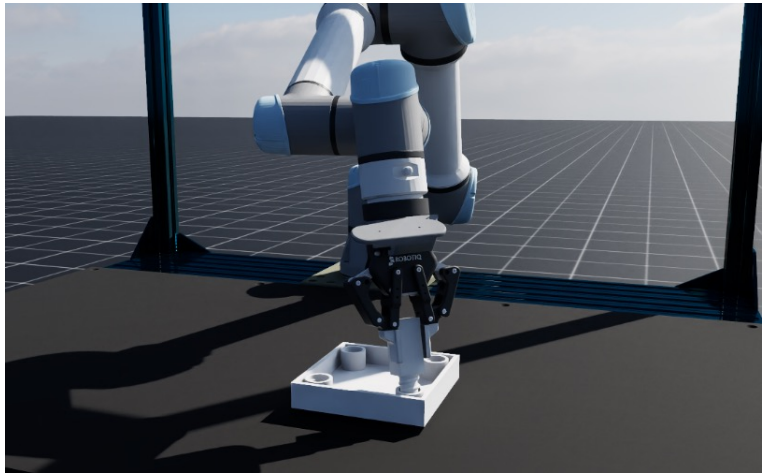


Can we bootstrap policy learning without demos?

Can we easily perform large-scale data generation **without** demos or reward design?

Key insight: Simulation != Reality





Simulation has privileged information that can be leveraged for policy search

- Resets
- Privileged information
- Cheap samples at scale

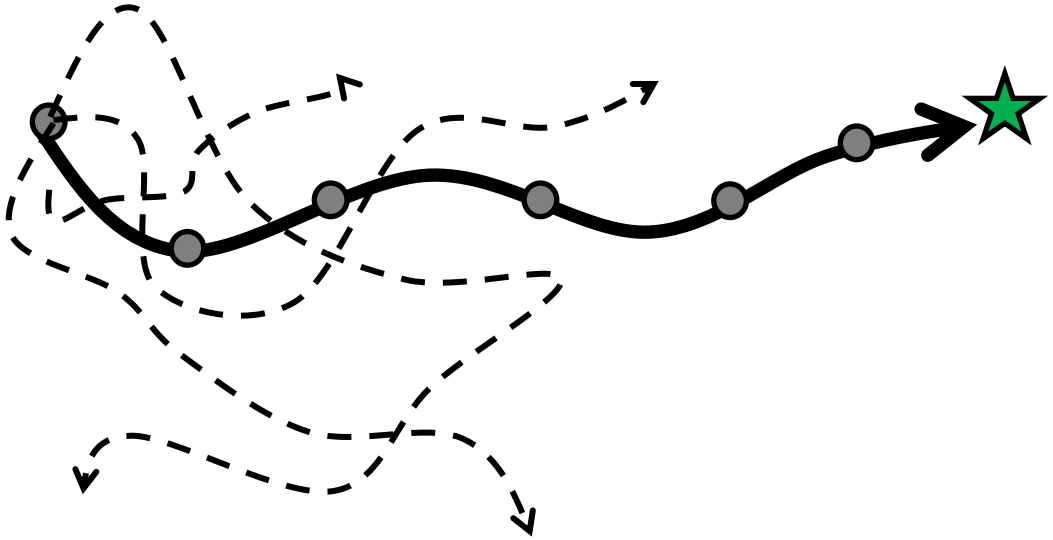
Patrick Yin

Tyler W

Emergent Dexterity via Diverse Resets and Large Scale Reinforcement Learning, Yin et al, ICLR '26

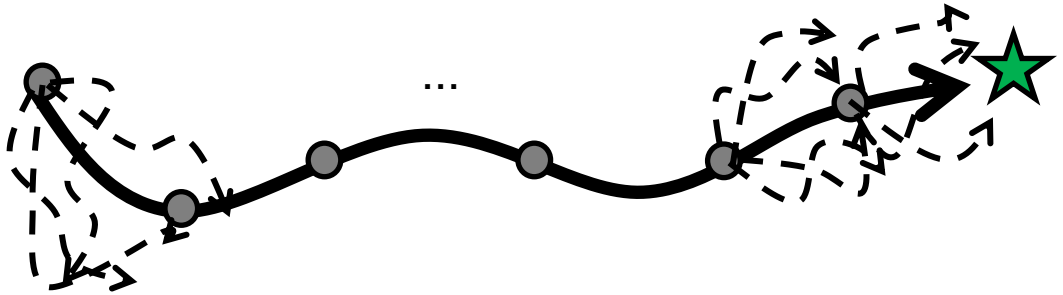
What can resets do for exploration?

Standard exploration



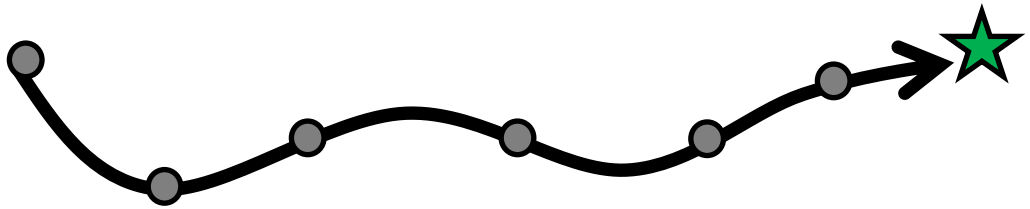
Requires **long-horizon** dithering to find rewards over horizon H

Reset-driven exploration



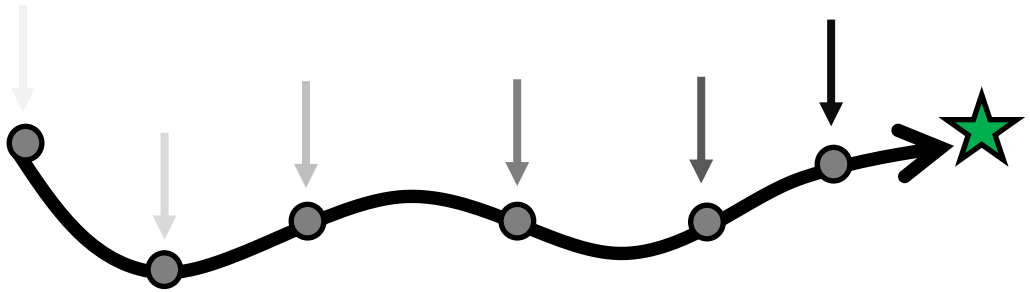
Resets allow for rewards to be found with **short-horizon** dithering

What should we reset to?

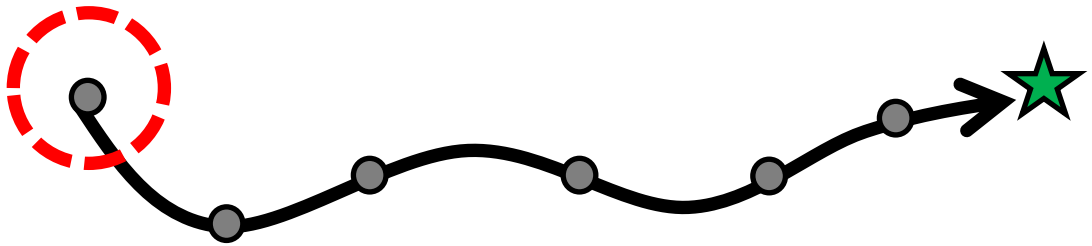


Conventional wisdom

Reset backwards from demonstrations

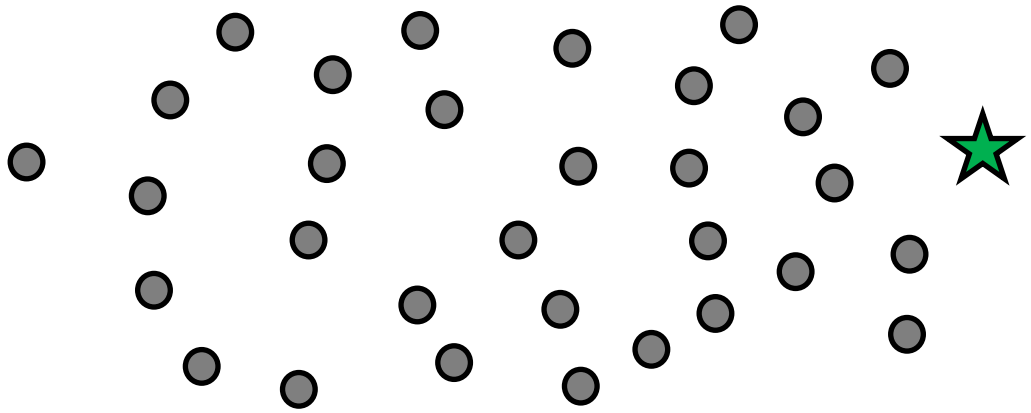


Learning is easier from a narrow initial dist



What should we reset to?

Insight:
Solve exploration by resetting very broadly rather than narrowly for RL



- ✓ No curricula, no ordering, just reset IID
- ✓ Use (largely) sparse rewards
- ✓ Same reset scheme across problems
- ✓ Easy to generate programmatically

Broad states for manipulation

Grasped

Random Poses

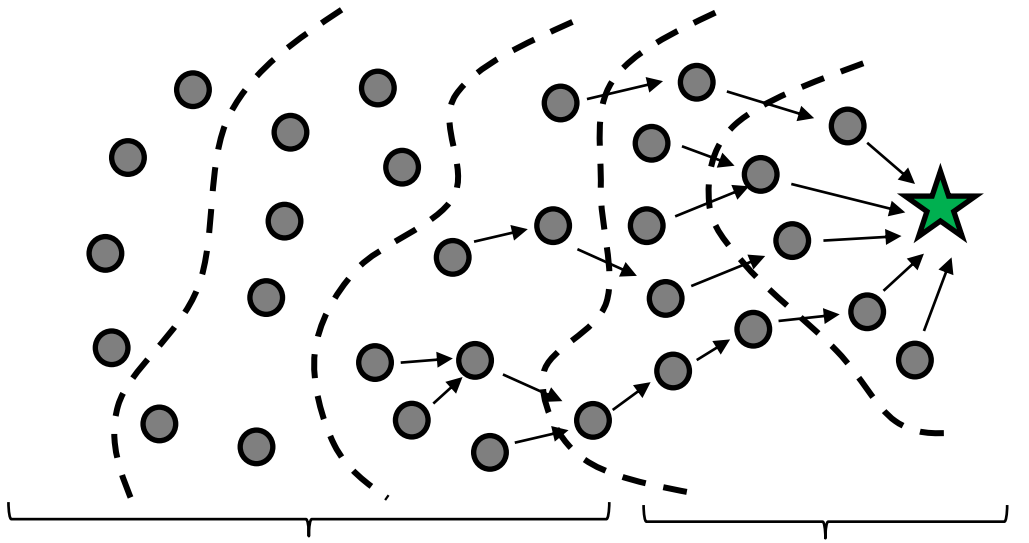
Near Objects

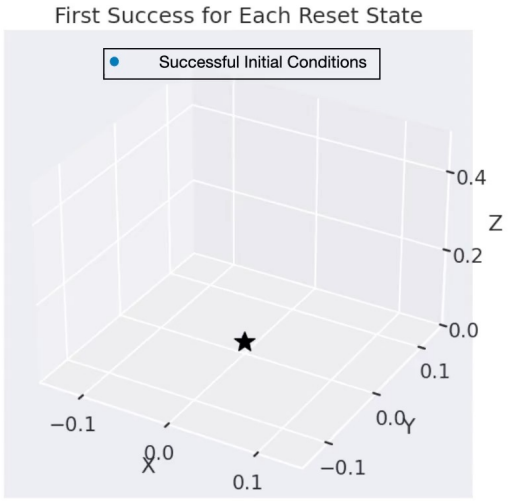
Partial Assemblies

Emergent Dexterity via Diverse Resets and Large Scale Reinforcement Learning, Yin et al, ICLR '26

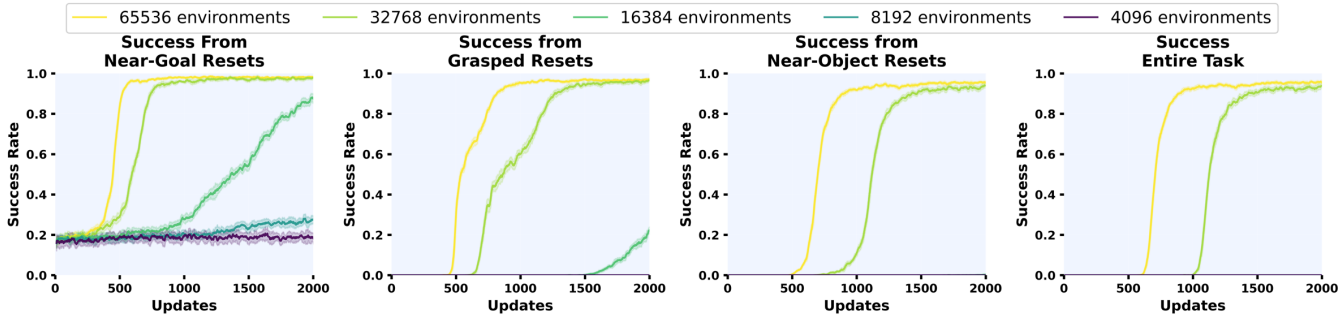
Broad Resets for Exploration Avoids Curricula

Emergent curriculum, no need to explicitly specify



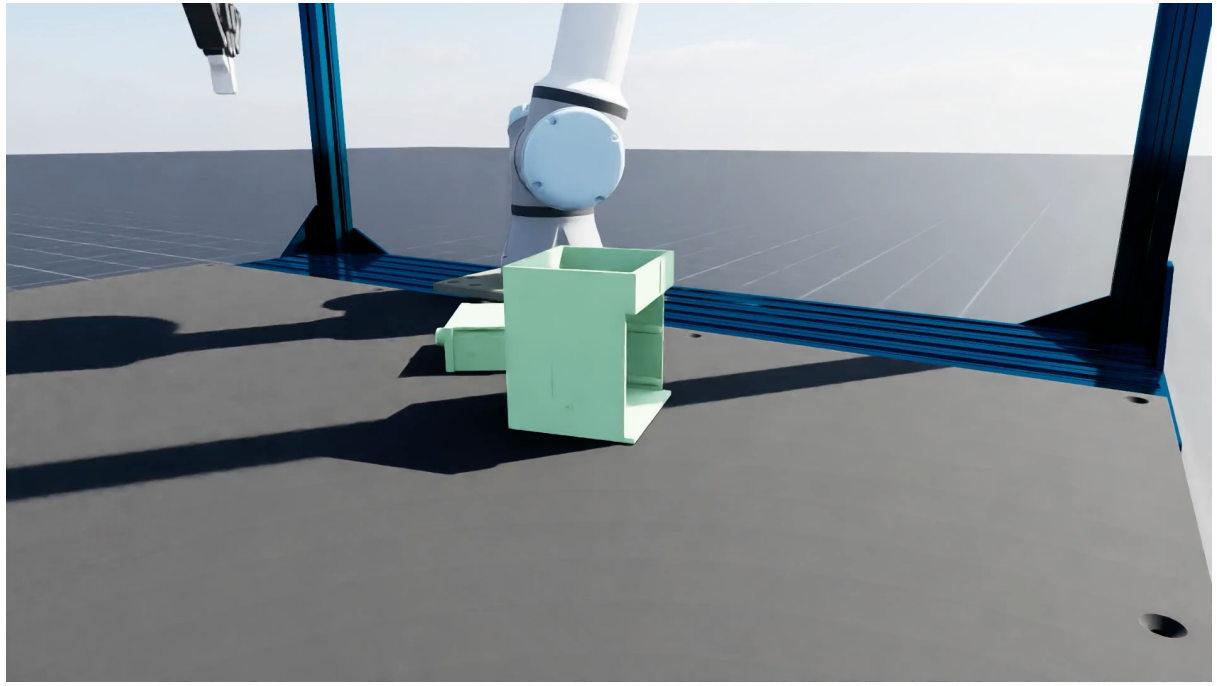


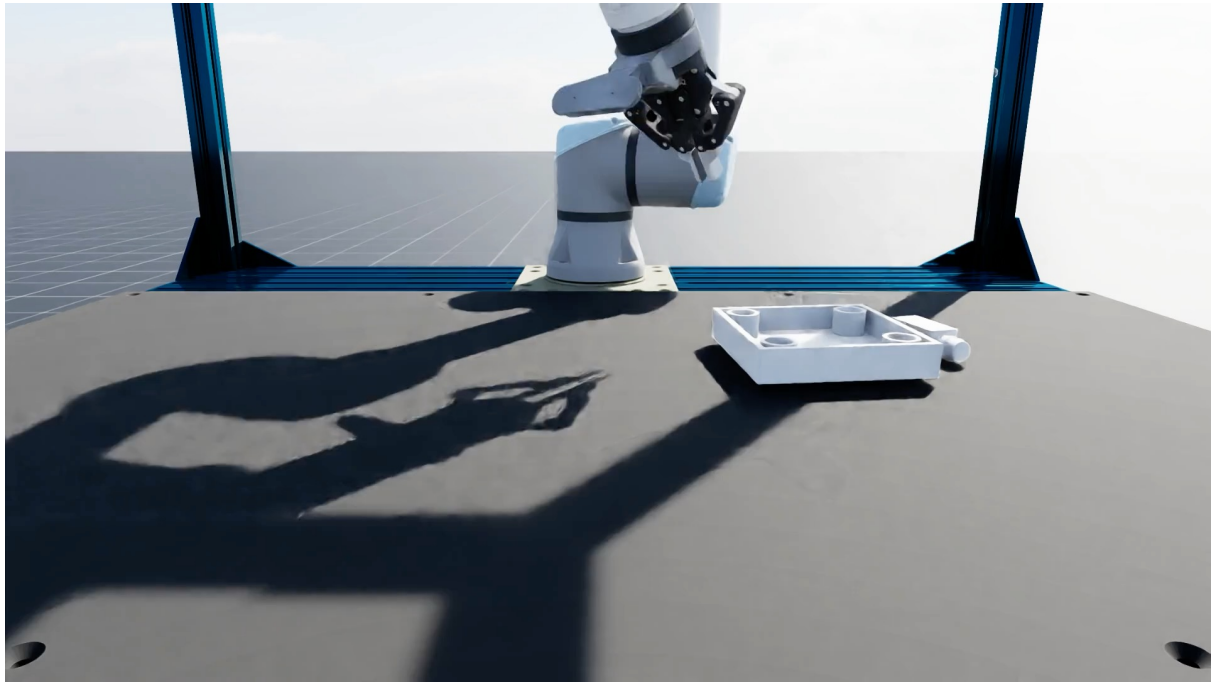
Requires batch size to be big enough to avoid forgetting



Emergent Dexterity via Diverse Resets and Large Scale Reinforcement Learning, Yin et al, ICLR '26

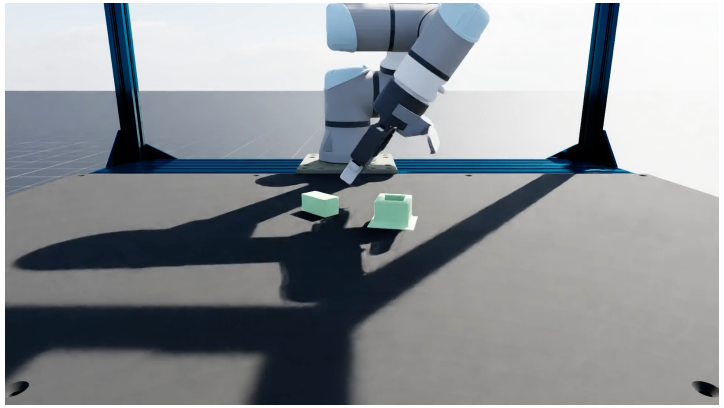
Ok so what does this get us?





No reward engineering

No demos!



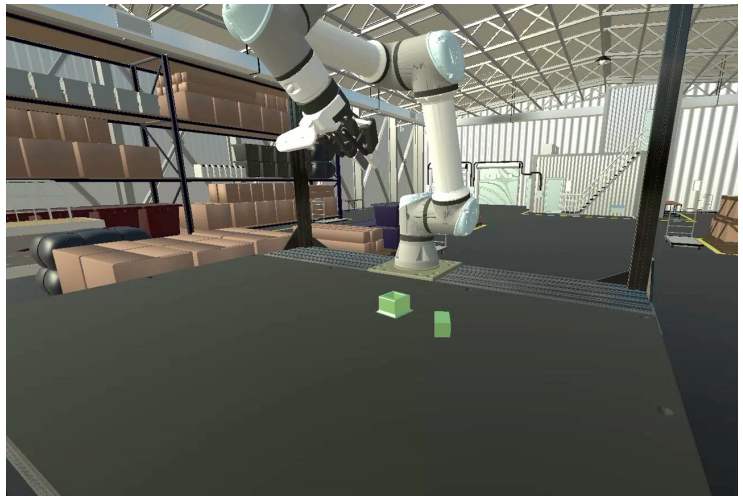
Same hyperparameters

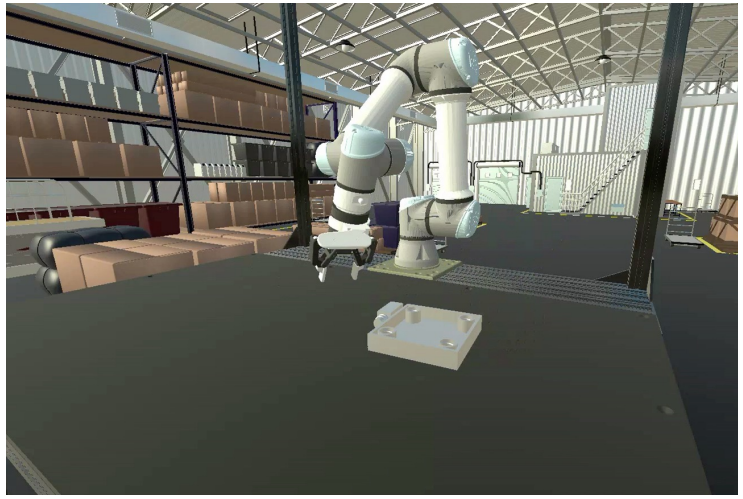
*Same algorithm across all tasks

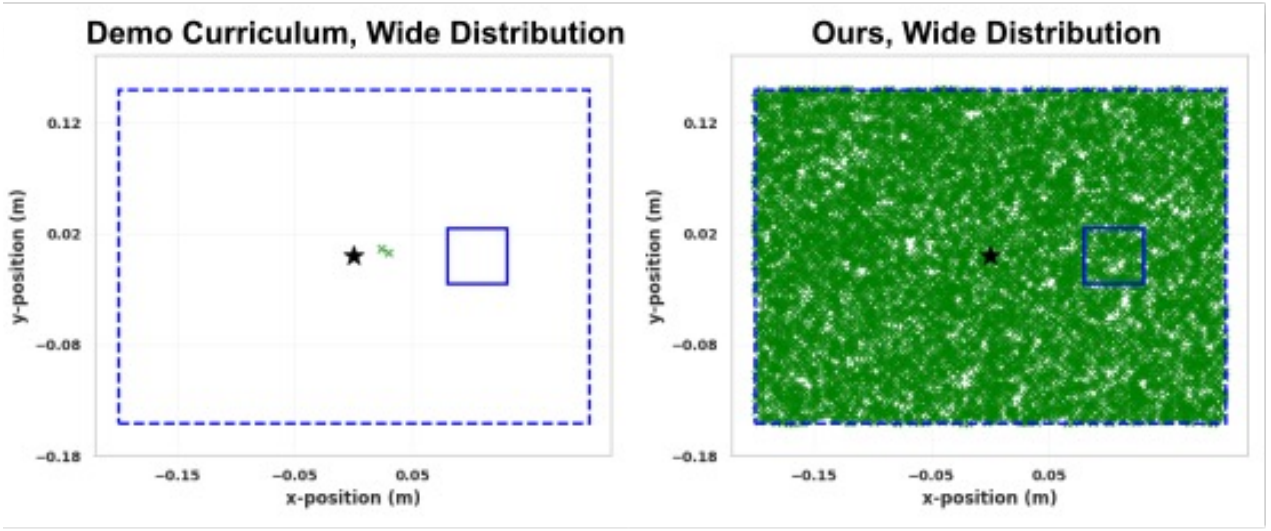
Emergent Dexterity via Diverse Resets and Large Scale Reinforcement Learning, Yin et al, ICLR '26

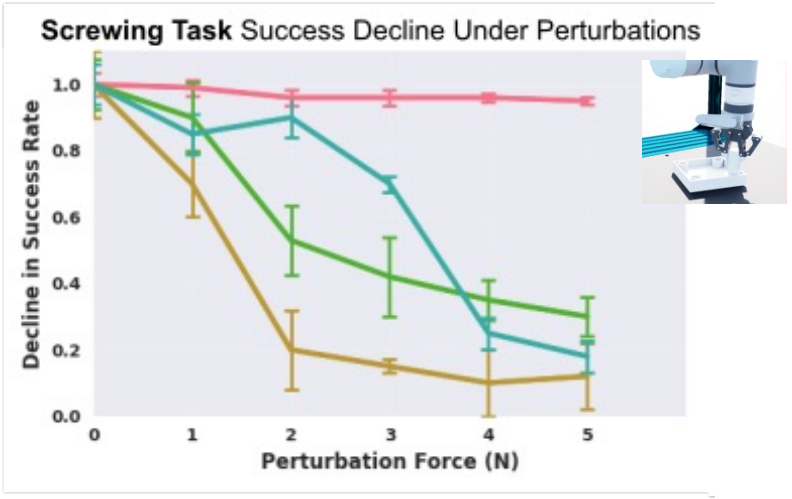
Extremely broad coverage and robustness

No-fuss tool for scalable, robust data generation





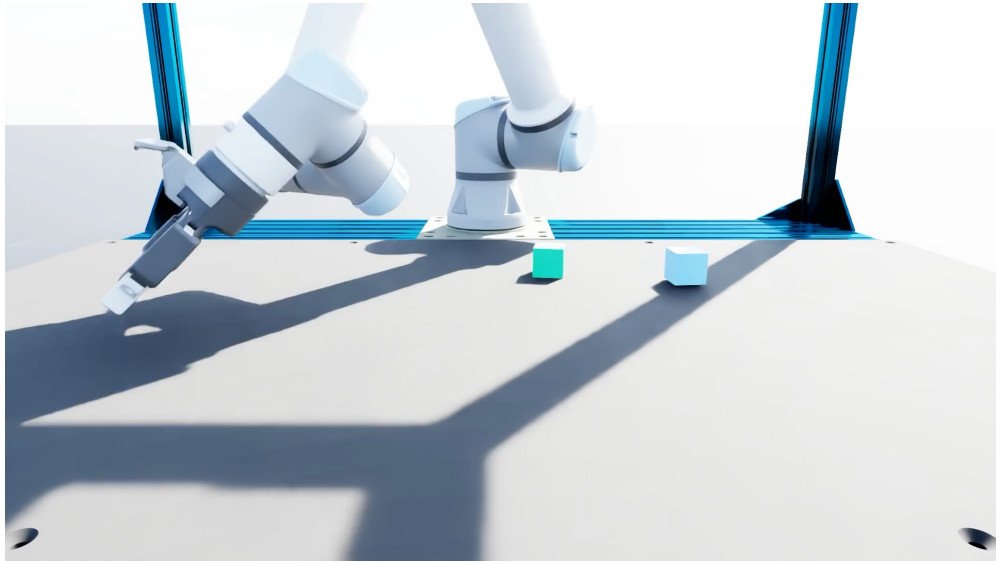




Trivially easy to apply to new tasks

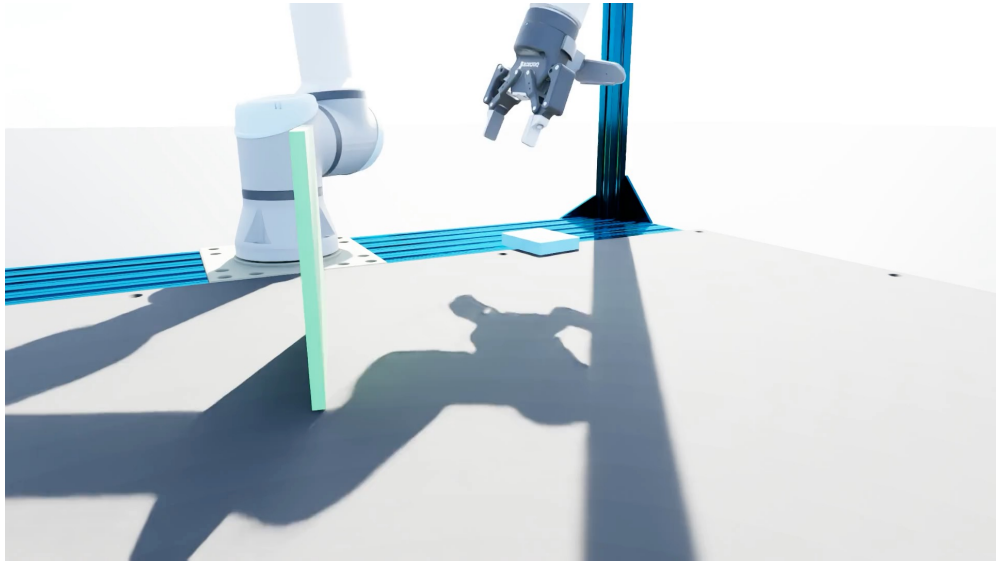
Assembly

Block Stacking



Non-regular shapes

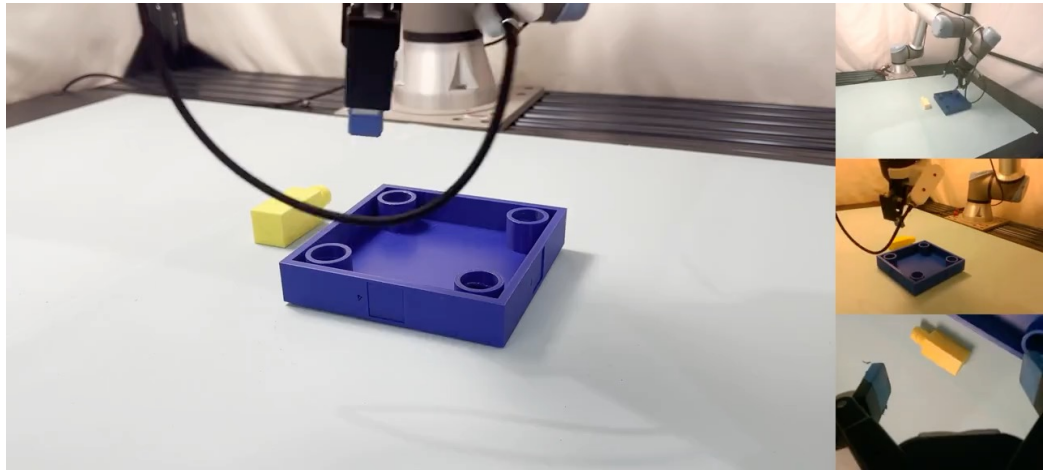
Non-prehensile motion

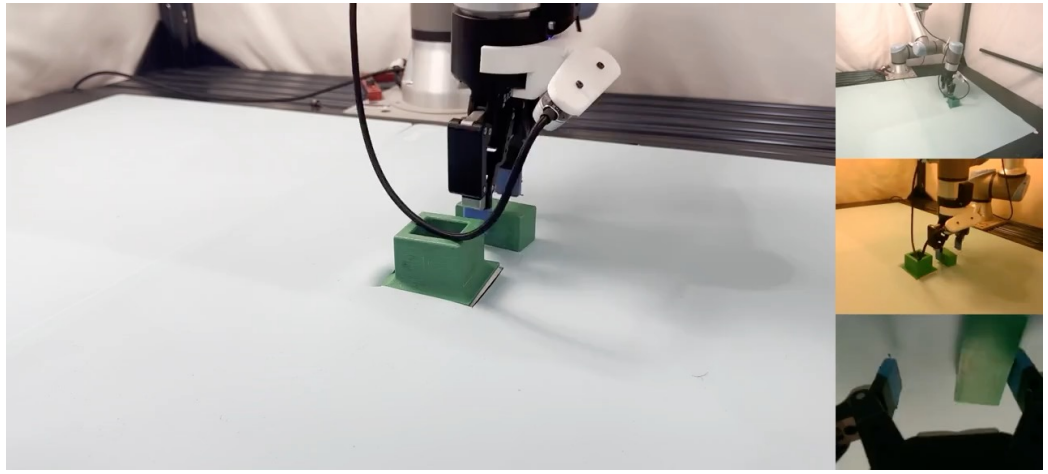


Emergent Dexterity via Diverse Resets and Large Scale Reinforcement Learning, Yin et al, ICLR '26

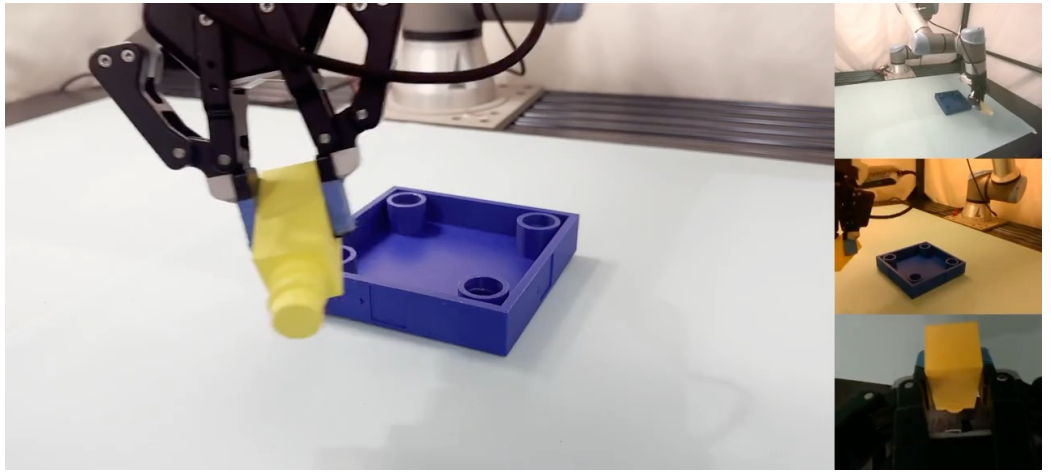
What happens when on transfer to the real world?

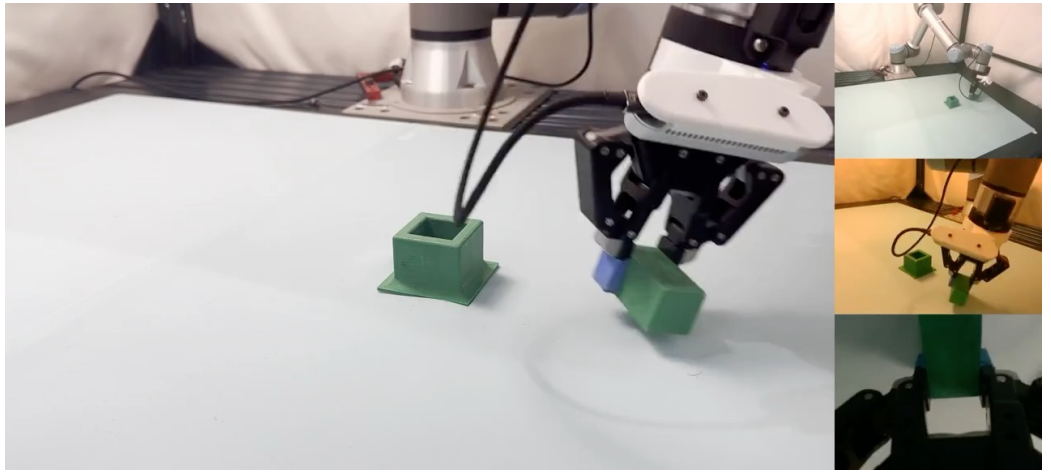
Some cool behavior





Many failures

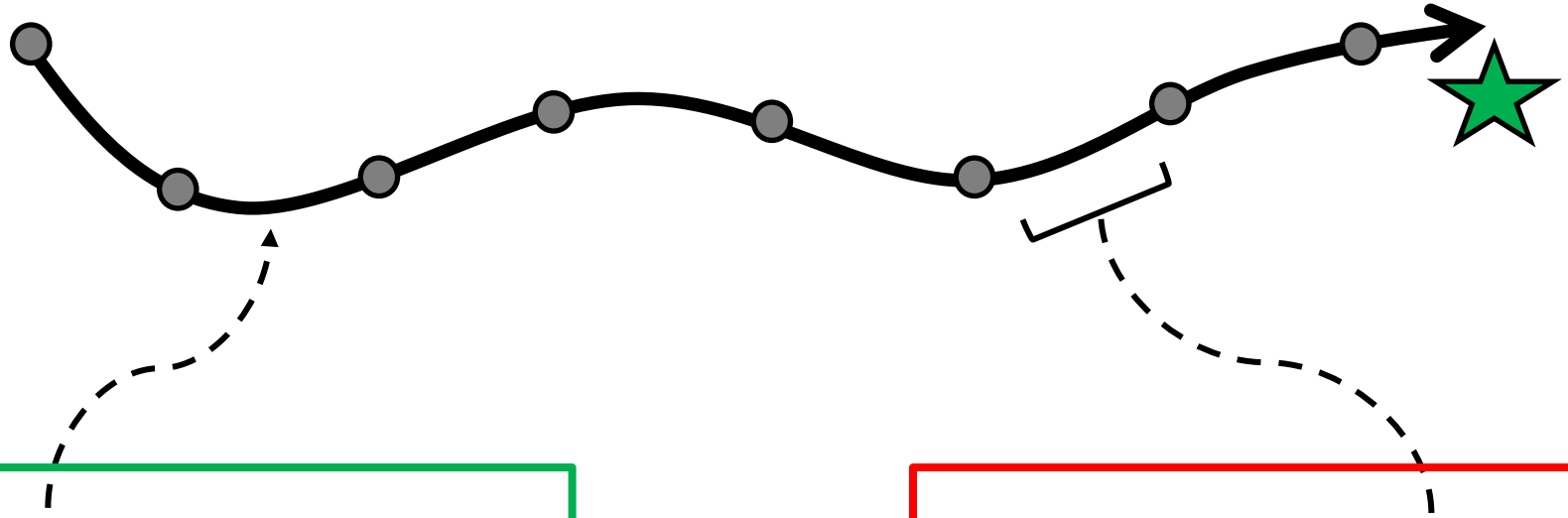




Still lots of room for improvement with real-world RL!

Even **Wrong** Simulation Can Make Real-World Adaptation Easier

Typically, end-to-end policies do both local acting and global planning



Global planning: Finding which states to visit to solve the problem

Transfers

(Represented by the value function)

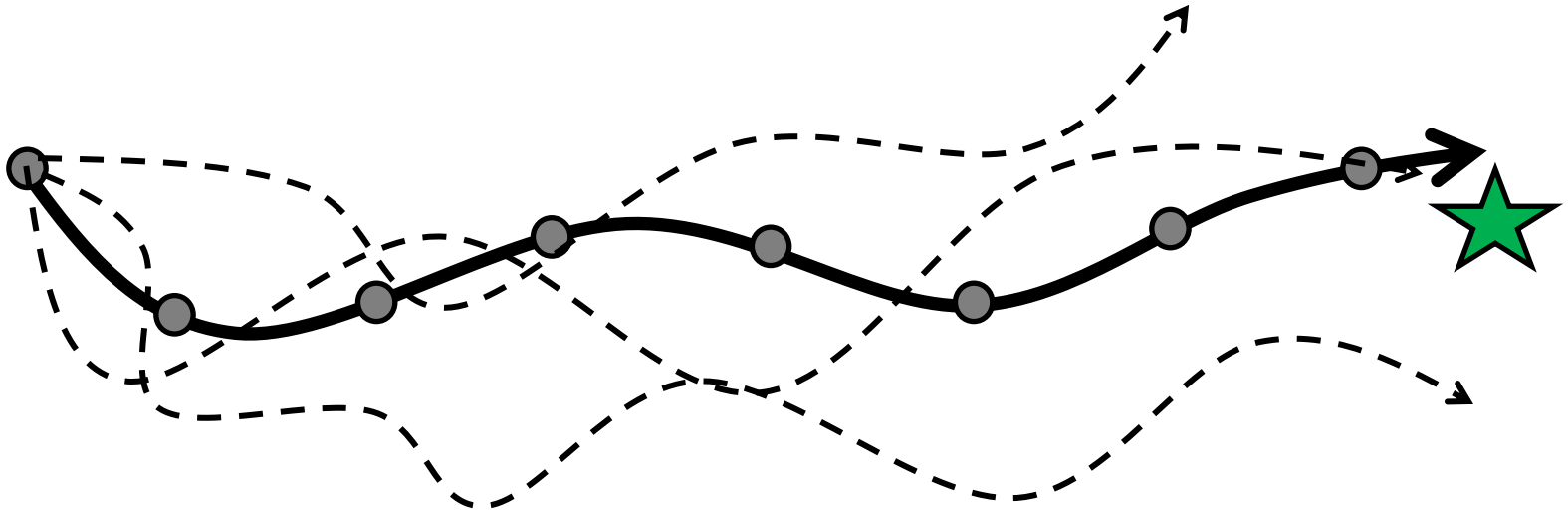
Local acting: Finding actions to accomplish global planned path

Doesn't fully transfer

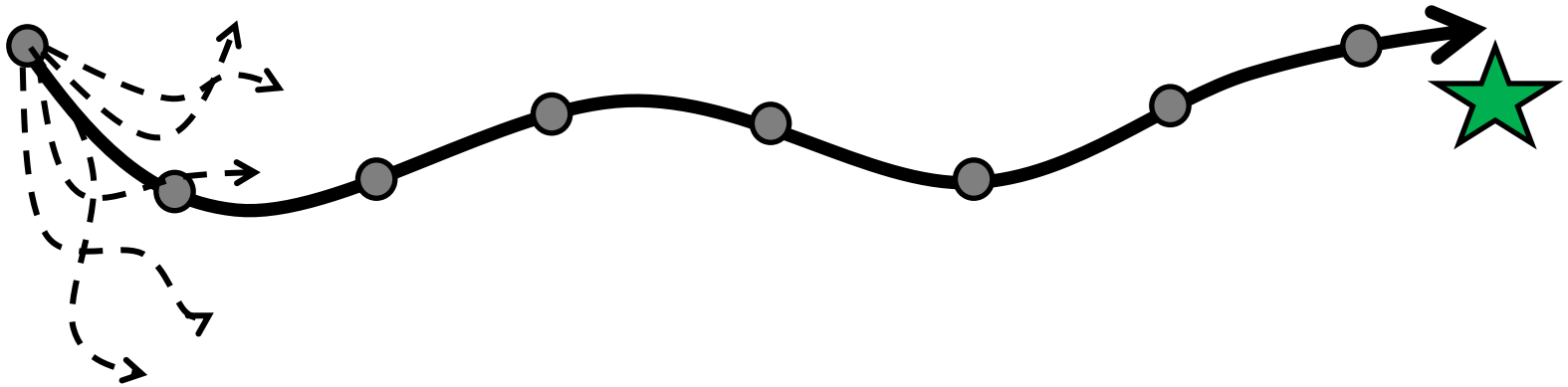
(Represented by the model)

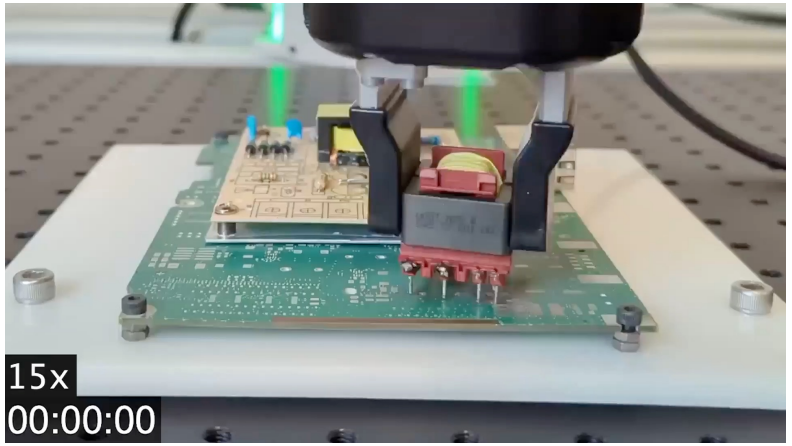
Simulation Distillation: Pretraining World Models in Simulation for Rapid Real-World Adaptation, Levy '26

What can simulation do for real-world RL?



Reduce to a tiny, short horizon problem

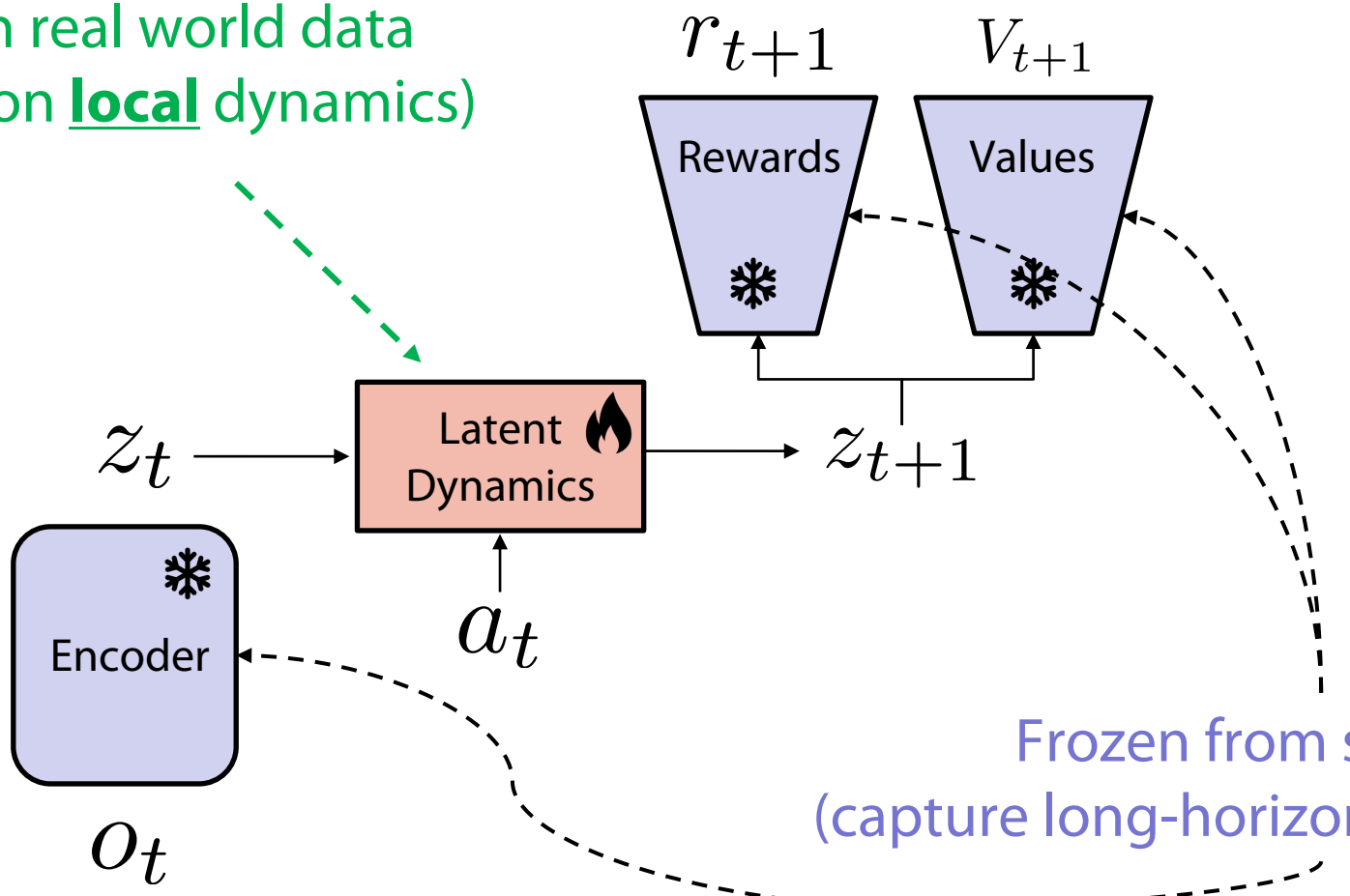




Rapidly Adapting Policies to the Real-World via Simulation-Guided Fine-Tuning, Yin et al , ICLR ‘25

Models Disentangle Local from Global

Adapted with real world data
(Fix short-horizon **local** dynamics)



Use for short-horizon planning at test time

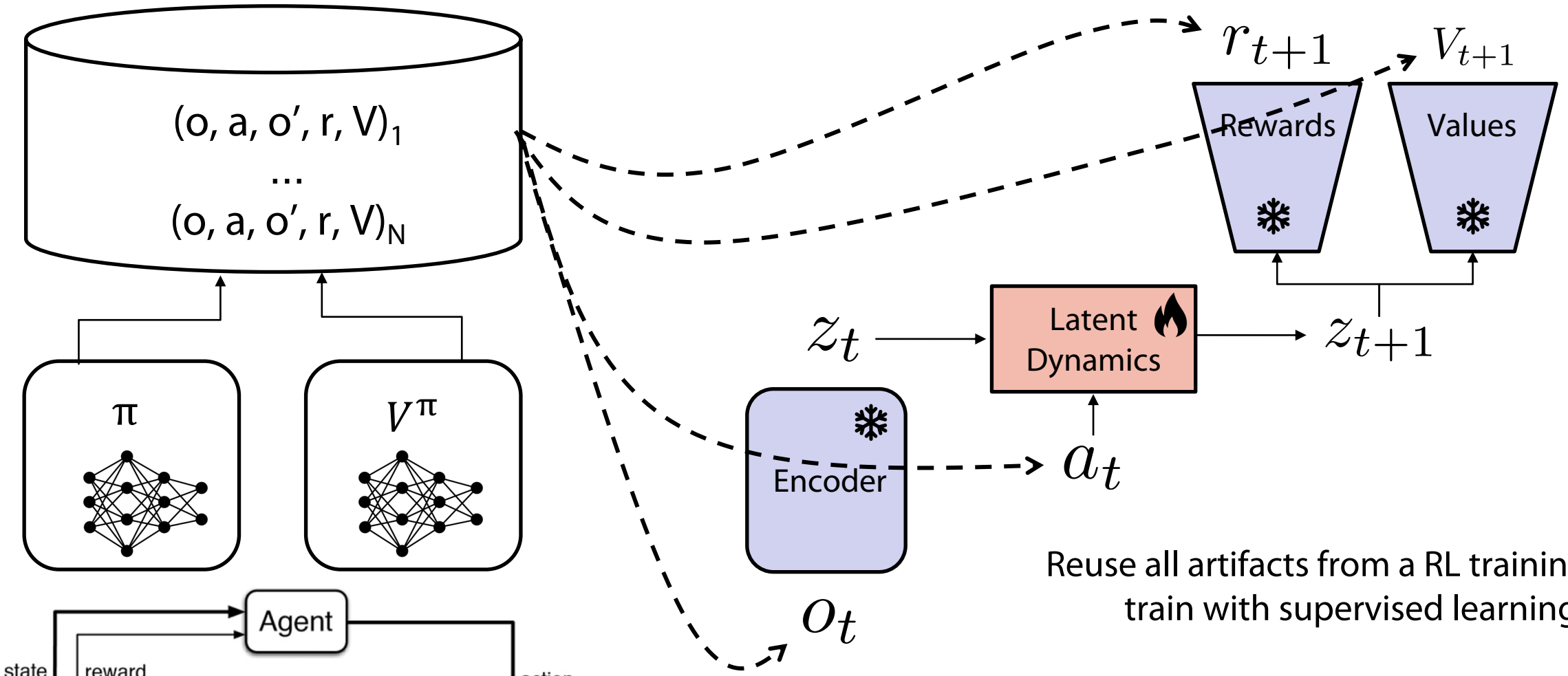
Frozen from simulation
(capture long-horizon **global** structure)


Tyler W


Jacob L

Simulation Distillation: Pretraining World Models in Simulation for Rapid Real-World Adaptation, ‘26

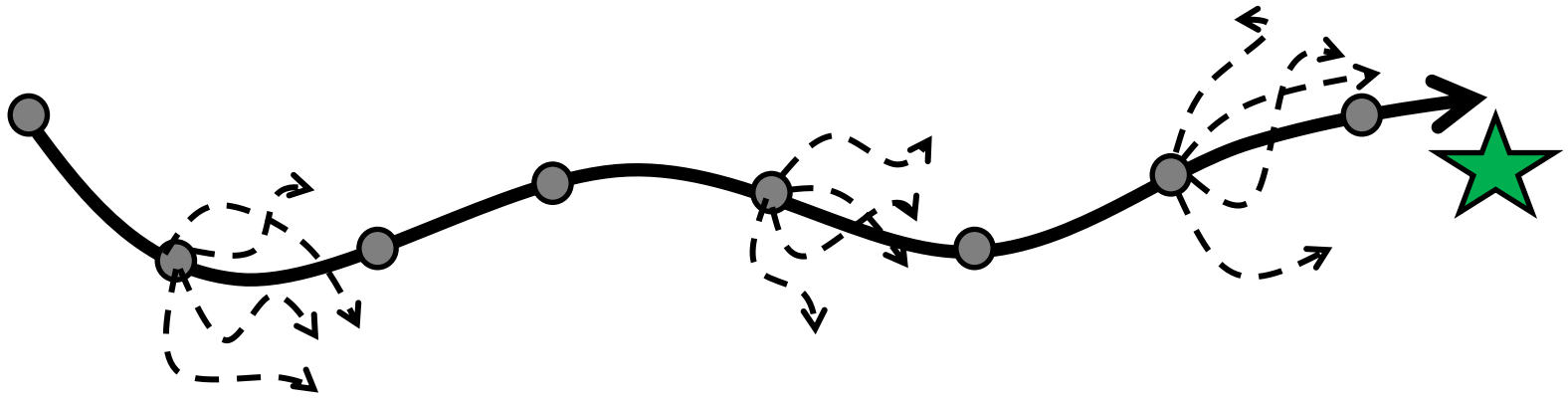
Models are easily trained from Simulation Pipelines



Reuse all artifacts from a RL training run, train with supervised learning

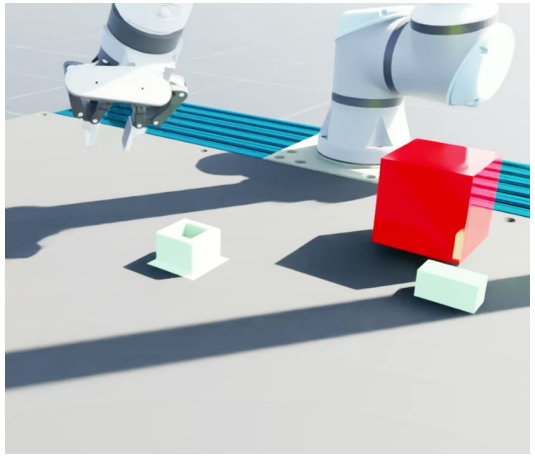
Simulation Distillation: Pretraining World Models in Simulation for Rapid Real-World Adaptation, '26

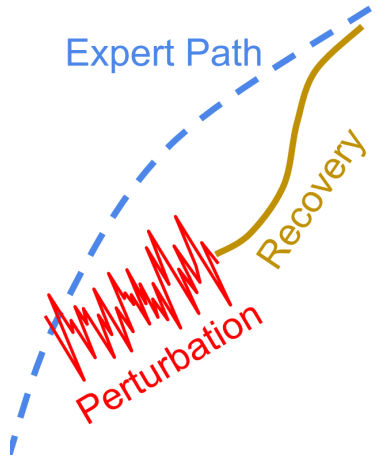
Models **must** be pre-trained on counterfactual data



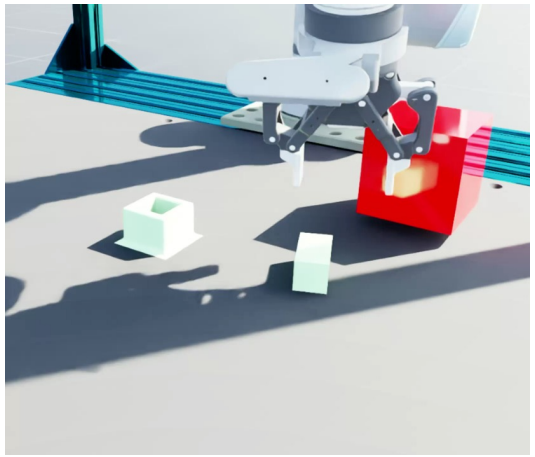
Need to leverage simulation pre-training to get coverage – avoid exploitation during planning

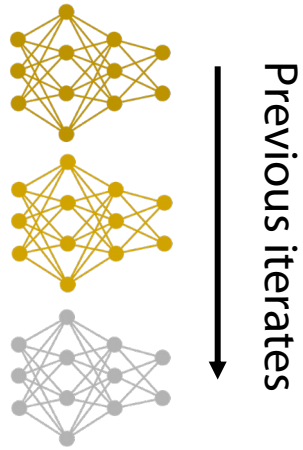
Action perturbation and recovery





Past checkpoints

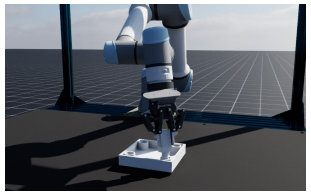




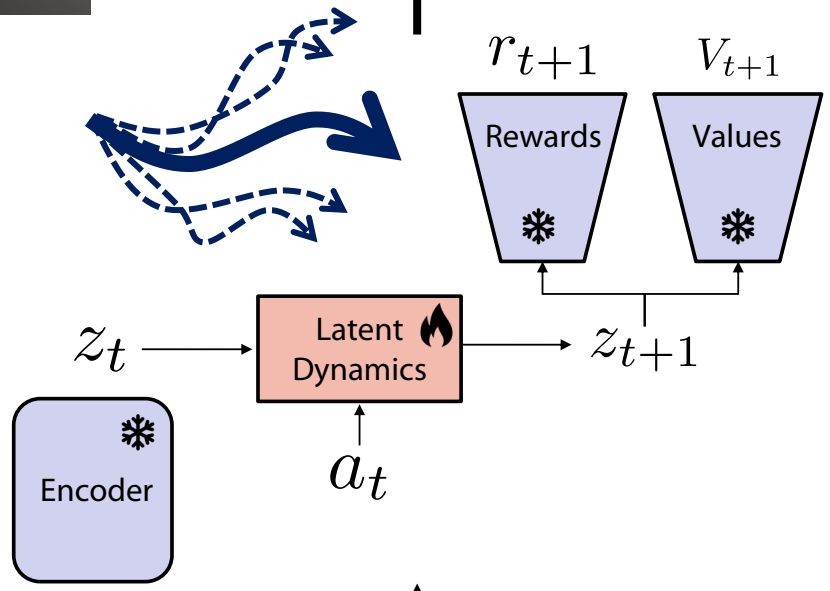
Simulation Distillation: Pretraining World Models in Simulation for Rapid Real-World Adaptation, ‘26

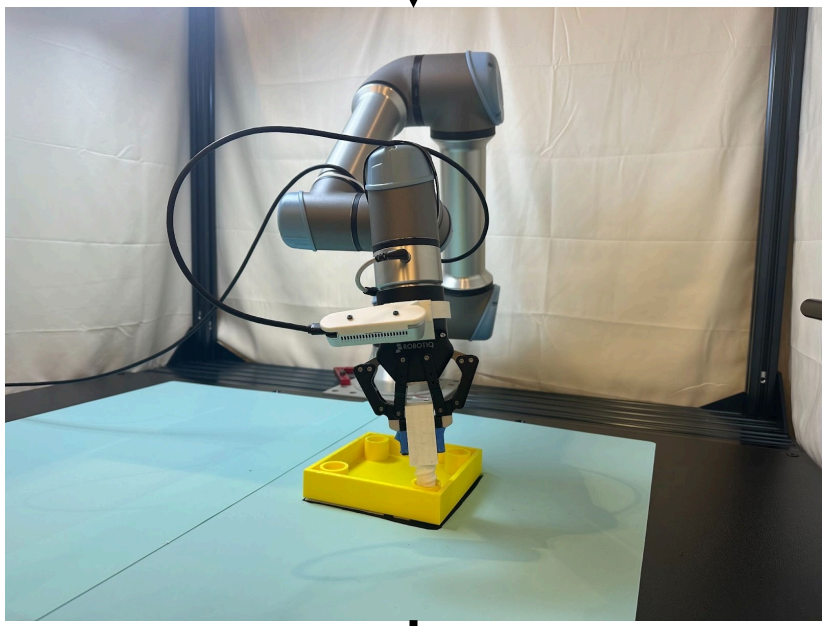
Finetuning only required on latent dynamics

Freeze pre-trained values/rewards/encoders, finetune latent dynamics



Short-horizon planning

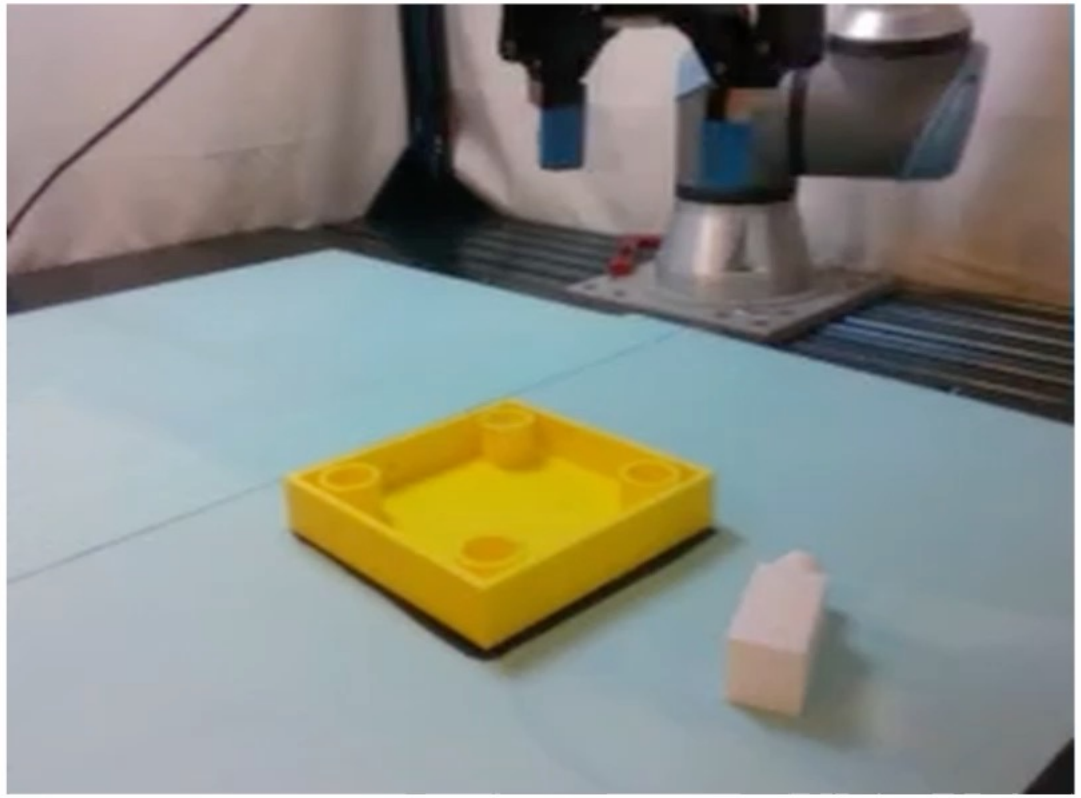




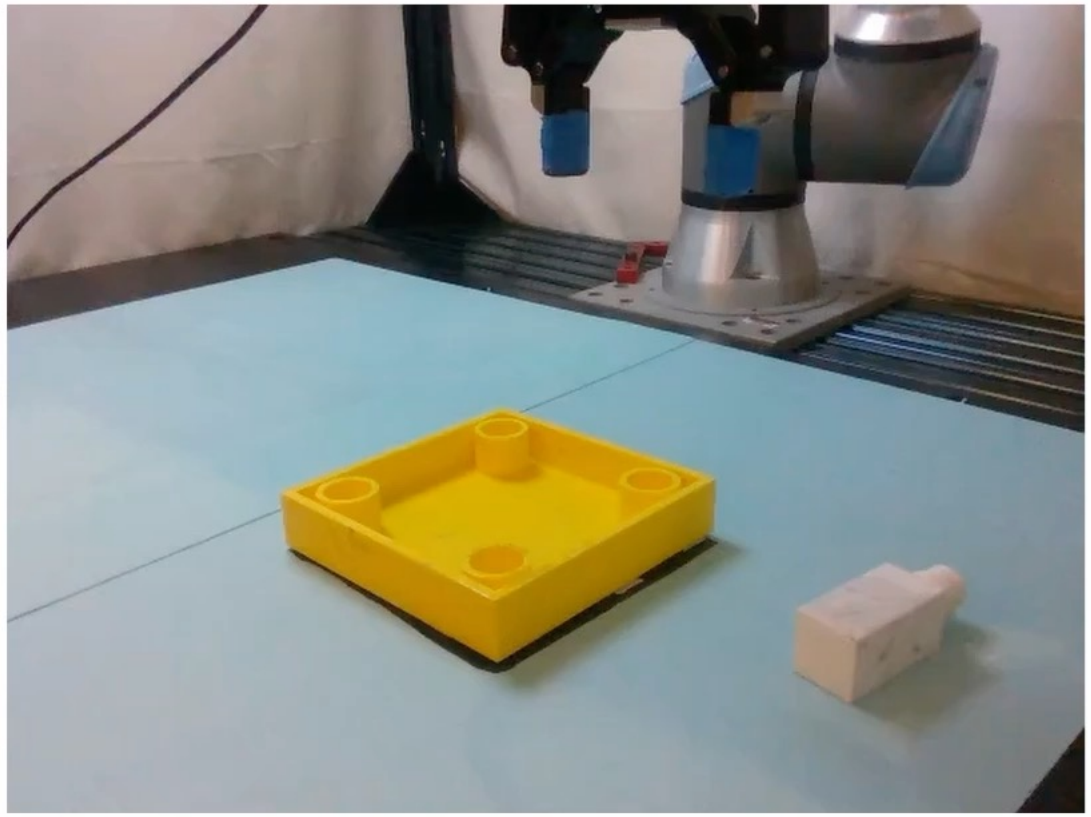
Supervised finetuning of latent dynamics

How well does finetuning work?

Zero-shot transfer



After finetuning



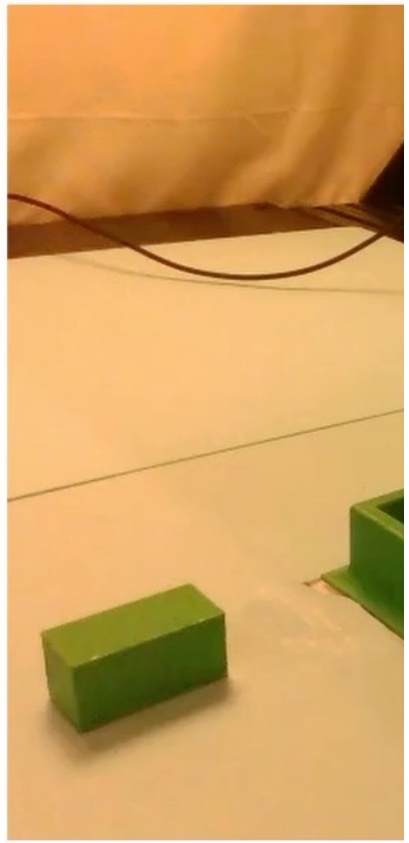
15 minutes of real-world data

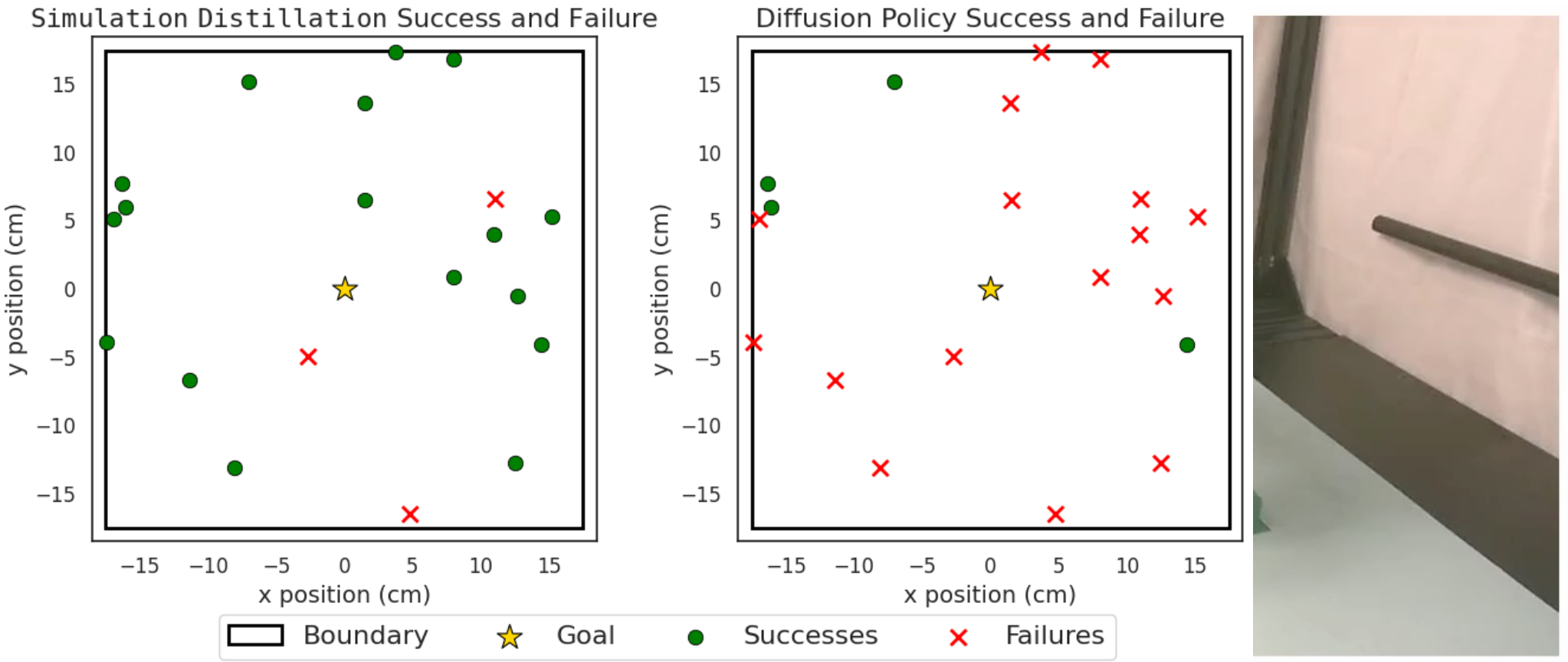
Simulation Distillation: Pretraining World Models in Simulation for Rapid Real-World Adaptation, ‘26

How well does finetuning work?

Zero-shot transfer

After finetuning



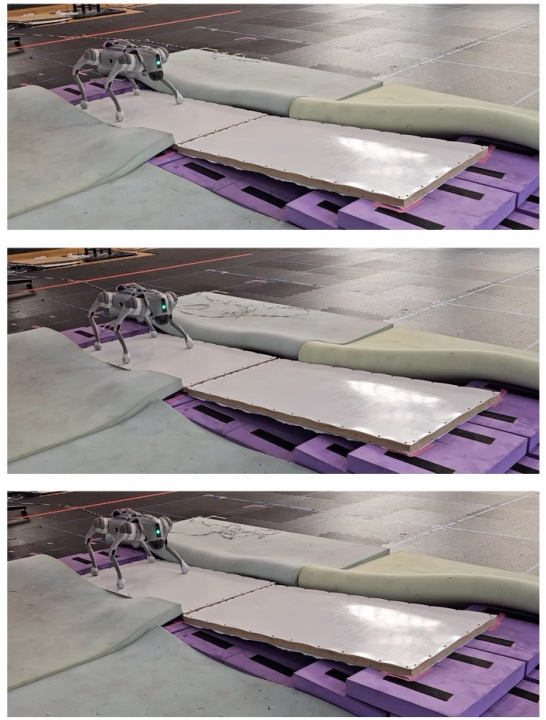


15 minutes of real-world data

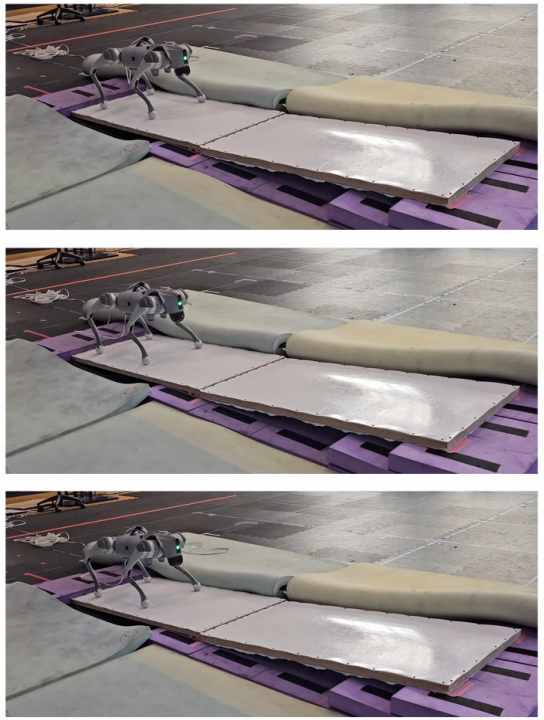
Simulation Distillation: Pretraining World Models in Simulation for Rapid Real-World Adaptation, ‘26

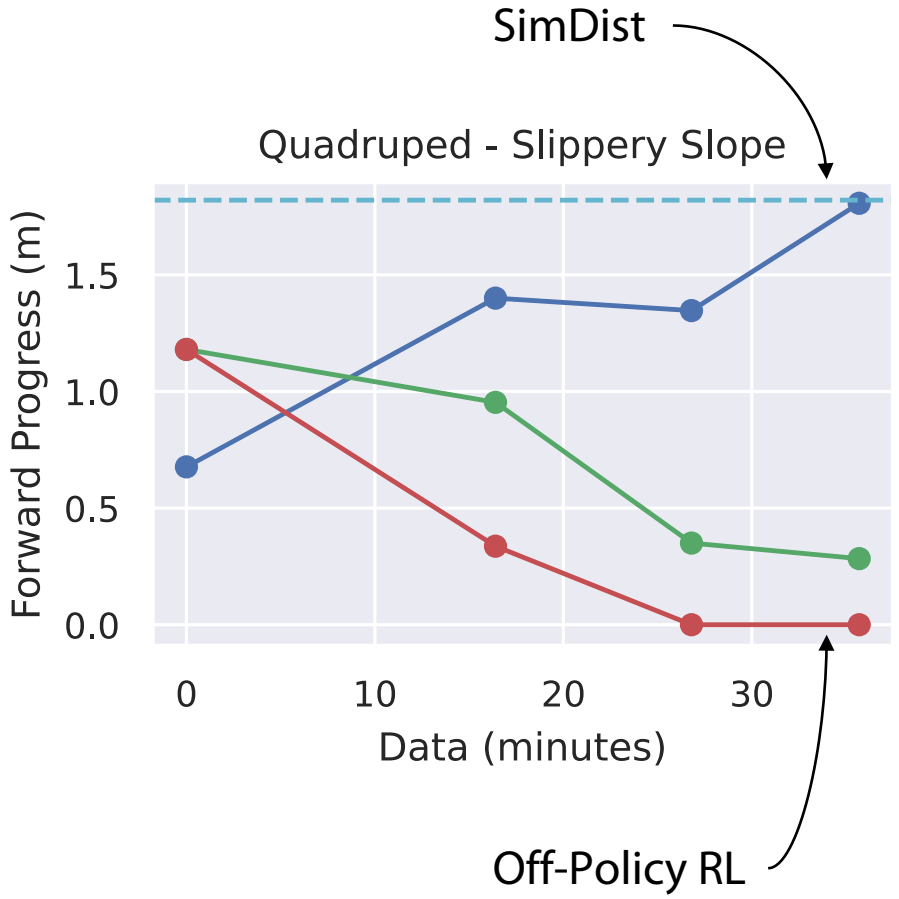
How well does finetuning work?

Zero-shot transfer



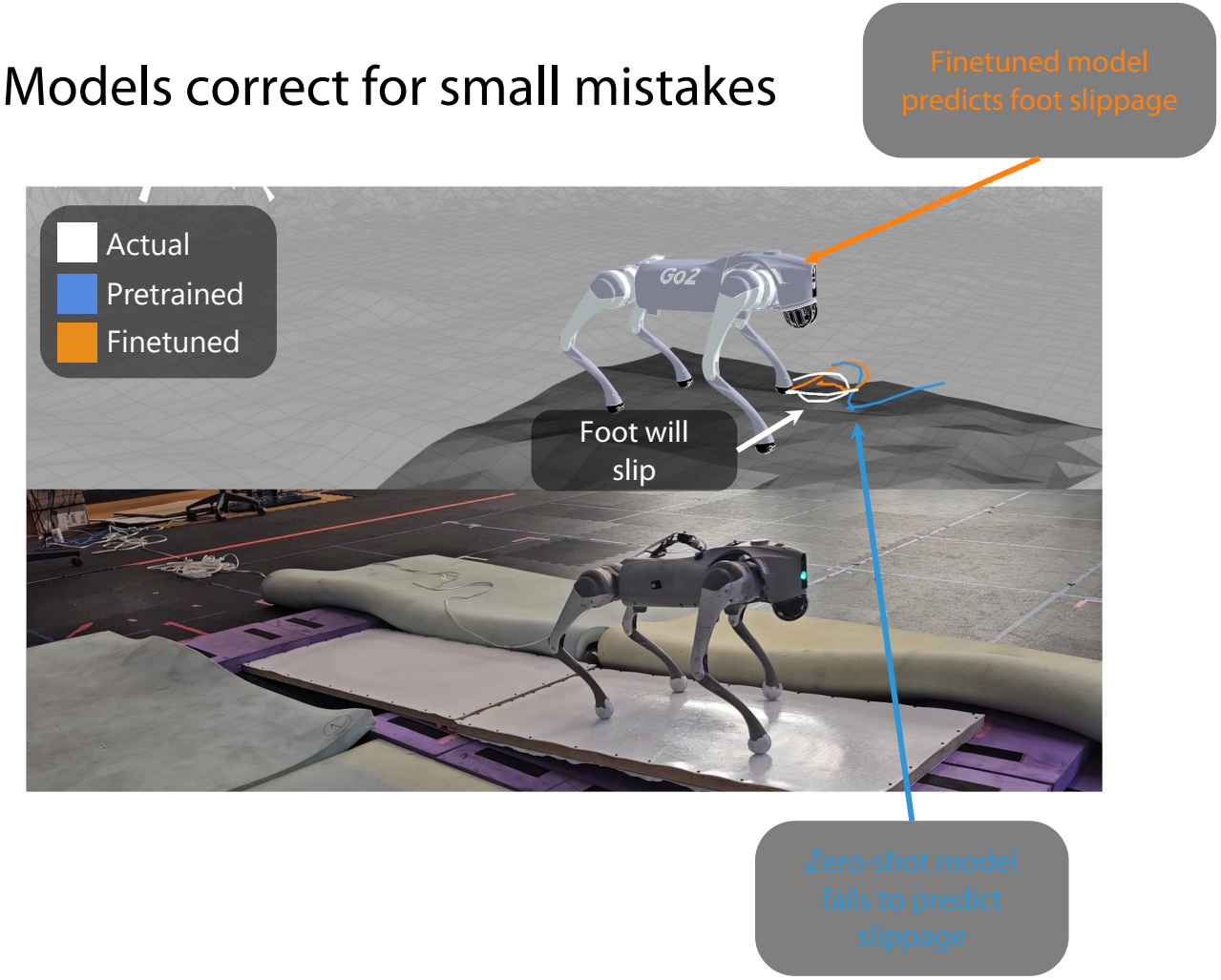
After finetuning



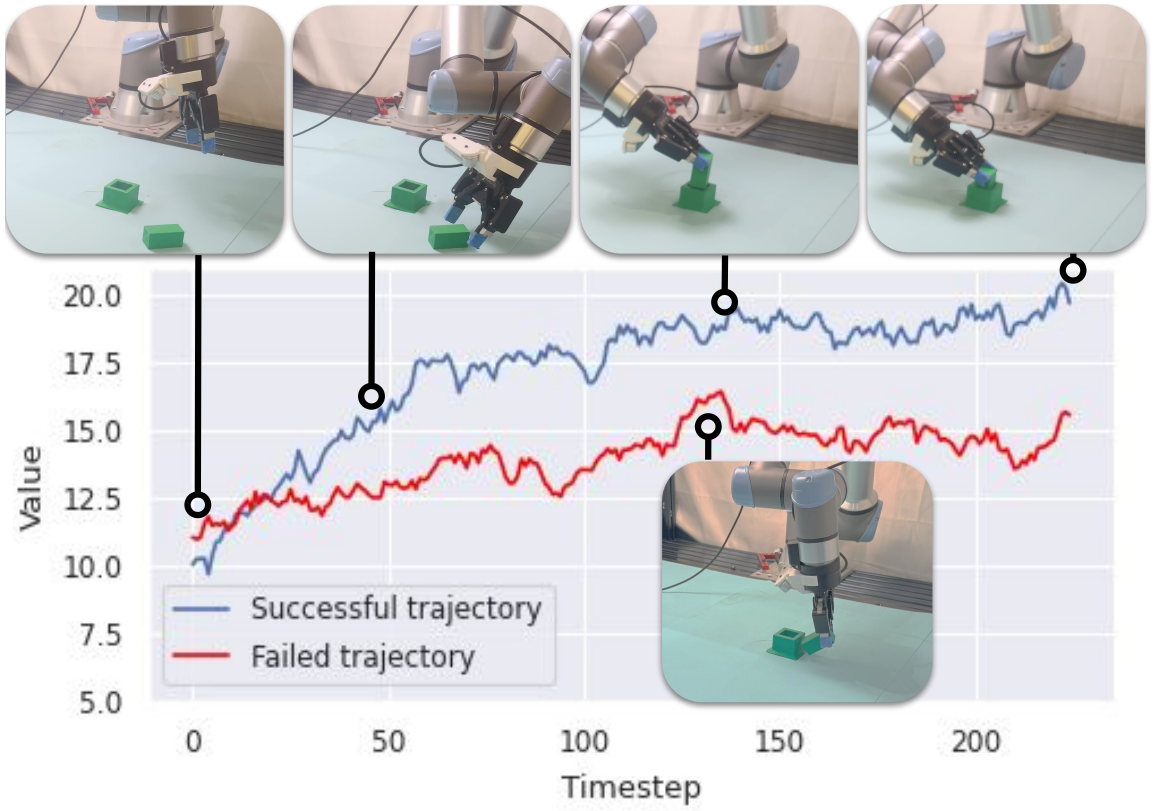


What does the model learn?

Models correct for small mistakes



Values capture global progress



Insights: Learning from Simulation

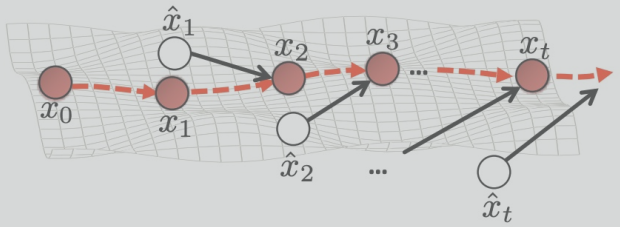
1. Generative models can produce targeted data, but only where they can be trusted

2. Video data can be absorbed by unified models with architectures to allow flexible inference

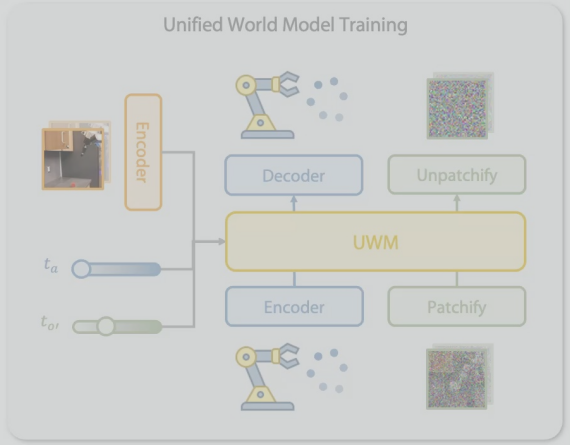
3. Simulation != Reality, use the privileges to learn reactive, contact-rich behavior and transfer

Learning from Off-Domain Data

Learn from **Generative Models**



Learn from **Video** Datasets



Learn from Scalable **Simulation**

Increasingly “off-domain”

The Bitter Lesson

The Bitter Lesson

Rich Sutton

March 13, 2019

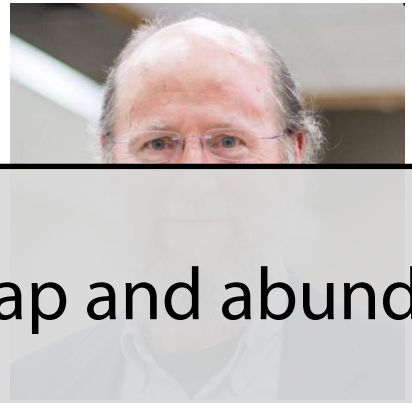
General methods that leverage computation are the most effective

Rich's reasoning:

1) AI researchers have often tried to build knowledge into their agents,
2) This always helps in the short term, and is personally satisfying to the researcher
3) In the long run it plateaus and even inhibits further progress, and
4) Breakthrough progress eventually arrives by scaling computation and data

Robotics Teaches us a More Bitter Lesson (Circa 2024)

The Bitter Lesson



Rich Sutton

March 13, 2019

General methods that leverage computation are the most effective

The bitter lesson assumes data is cheap and abundant

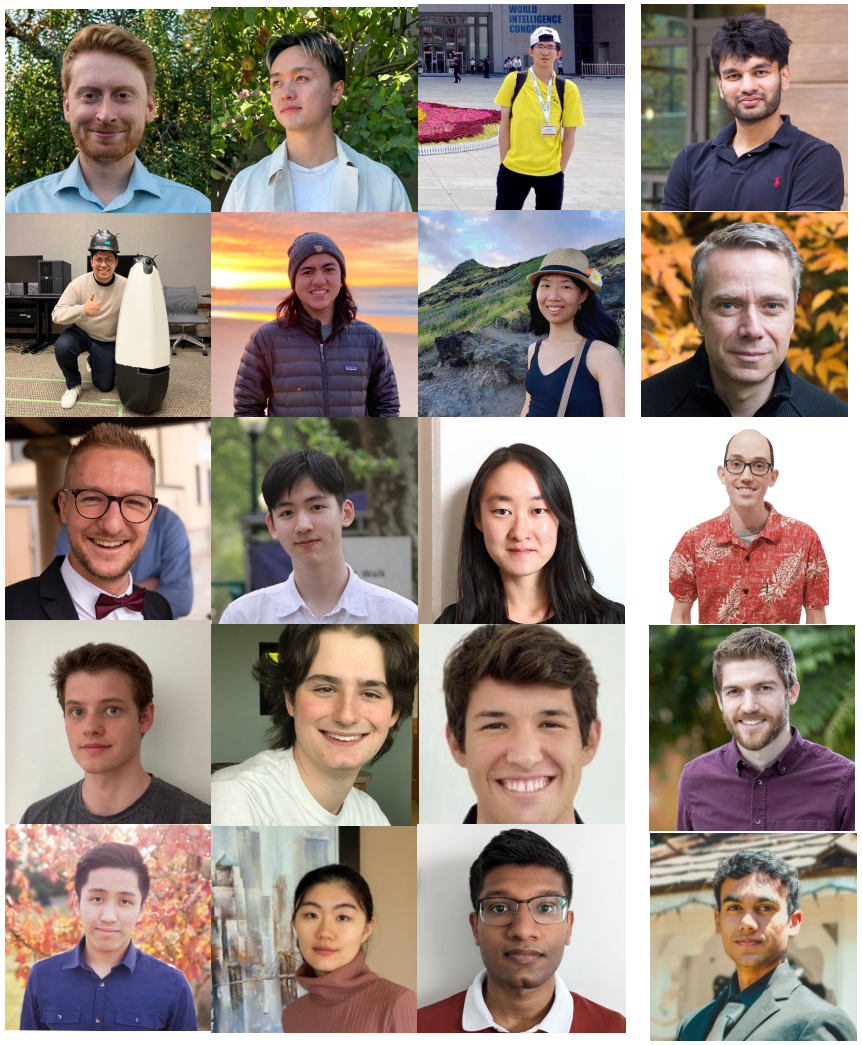
Along with compute scale, we need to find data sources that scale cheaply

Scale must be considered both **computationally** and **economically**, while capturing the "complexity" of the real world

Richard's message is:

1) AI researchers have often tried to build knowledge into their agents,
2) This always helps in the short term, and is personally satisfying to the researcher,
3) In the long run it plateaus and even inhibits further progress, and
4) Breakthrough progress eventually arrives by scaling computation and data

Thank You!



Thank You!

Papers Covered:

CCIL: Continuity-based Data Augmentation for Corrective Imitation Learning
Liyiming Ke*, Yunchu Zhang*, Abhay Deshpande, Siddhartha S. Srinivasa, Abhishek Gupta (ICLR '24)

Data Efficient Behavior Cloning for Fine Manipulation via Continuity-based Corrective Labels
Abhay Deshpande, Liyiming Ke, Quinn Pfeifer, Abhishek Gupta, Siddhartha S. Srinivasa (IROS '24)

Unified World Models: Coupling Video and Action Diffusion for Pretraining on Large Robotic Datasets
Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta (RSS '25)

Emergent Dexterity via Diverse Resets and Large-Scale Reinforcement Learning
Patrick Yin*, Tyler Westenbroek*, Zhengyu Zhang, Joshua Tran, Ignacio Dagnino, Eeshani Shilamkar, Numfor Mbiziwo-Tiapo, Simran Bagaria, Xinlei Liu, Galen Mullins, Andrey Kolobov, Abhishek Gupta (ICLR '26)

Simulation Distillation: Pretraining World Models in Simulation for Rapid Real-World Adaptation
Jacob Levy, Tyler Westenbroek, Kevin Huang, Fernando Palafox, Patrick Yin, Shayegan Omidshafiei, Dong-Ki Kim, Abhishek Gupta, David Fridovich-Keil (In Submission)