Robot Perception and Learning

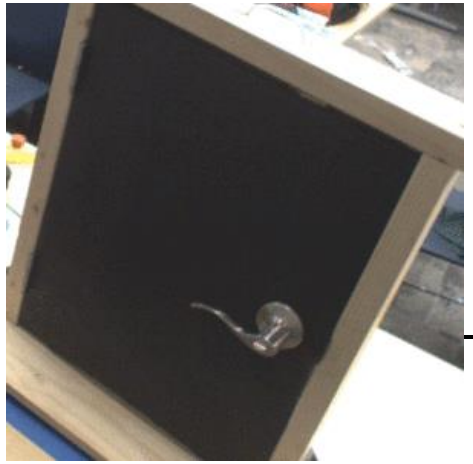
Real2Sim2Real Learning, Sim2real Transfer, Adaptation

Tsung-Wei Ke

Fall 2025

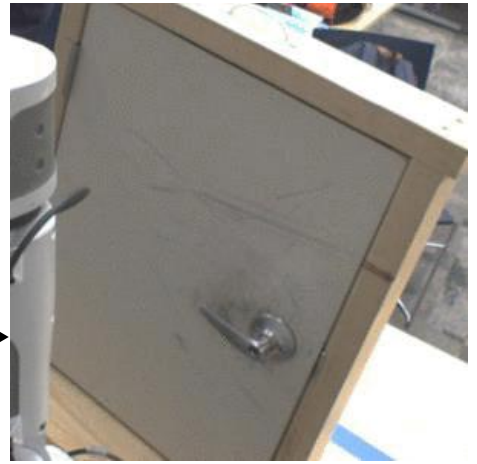
Let's Take a Little Break, Rethinking What We've Learned

- Types of visual imitation learning:
  1. Knowledge distillation via representation learning
  2. Knowledge distillation via video generation
  3. Knowledge distillation with correspondence



- From demonstration, we want to know:
  - Where to interact with the object
  - How to interact with the object
  - What is the goal state

- To know such information, we've considered:
  - Similarity of latent features
  - Video generation
  - Correspondence



Any other representations that generally encode such information?

What information should we extract from human demonstration videos?

- Types of visual imitation learning:
  1. Knowledge distillation via representation learning
  2. Knowledge distillation via video generation
  3. Knowledge distillation with correspondence
  4. Knowledge distillation with affordance
  5. Knowledge distillation with point trajectories
  6. Knowledge distillation with 3D hand modeling
  7. Knowledge distillation with digital twins

Knowledge Distillation with Affordance

Human video

Where to interact with the object

How to interact with the object

Frame without human

Frame without human

Contact points

Wrist trajectory

Affordances from Human Videos as a Versatile Representation for Robotics. Bahl et al.

Knowledge Distillation with Affordance

Human Videos

Robot execution

Affordances from Human Videos as a Versatile Representation for Robotics. Bahl et al.

How to Extract Contact Points from Video?

1. Detect bounding boxes of the hand and the object
2. Find all points on the hand intersect with the object, and keep track of:
   - A set of N contact points $\{c^i\}^N$
   - The first timestep where contact occurs in the human video $t_{contact}$
   - Find the post-contact trajectory of the 2D hand bounding box $\{h_t\}_{t_{contact}}^{t'}$

Affordances from Human Videos as a Versatile Representation for Robotics. Bahl et al.

How to Extract Contact Points from Video?

1. Detect bounding boxes of the hand and the object
2. Find all points on the hand intersect with the object, and keep track of:
   - A set of N contact points $\{c^i\}^N$
   - The first timestep where contact occurs in the human video $t_{contact}$
   - Find the post-contact trajectory of the 2D hand bounding box $\{h_t\}_{t_{contact}}^{t'}$
3. Fit a Gaussian Mixture Model to the points

$$p(c) = \underset{\mu_1,\ldots,\mu_K,\Sigma_1,\ldots,\Sigma_K}{\operatorname{argmax}} \sum_{i=1}^{N}\sum_{k=1}^{K} \alpha_k \mathcal{N}(c^i|\mu_k,\Sigma_k)$$

Affordances from Human Videos as a Versatile Representation for Robotics. Bahl et al.

How to Extract Contact Points from Video?

Post-contact trajectory:

$$\tau = \mathcal{H}_t \circ \{h_t\}_{t_{\text{contact}}}^{t'}$$

1. Detect bounding boxes of the hand and the object
2. Find all points on the hand intersect with the object, and keep track of:
   - A set of N contact points $\{c^i\}^N$
   - The first timestep where contact occurs in the human video $t_{contact}$
   - Find the post-contact trajectory of the 2D hand bounding box $\{h_t\}_{t_{contact}}^{t'}$
3. Fit a Gaussian Mixture Model to the points

$$p(c) = \underset{\mu_1,\ldots,\mu_K,\Sigma_1,\ldots,\Sigma_K}{\text{argmax}} \sum_{i=1}^{N}\sum_{k=1}^{K} \alpha_k \mathcal{N}(c^i|\mu_k,\Sigma_k)$$

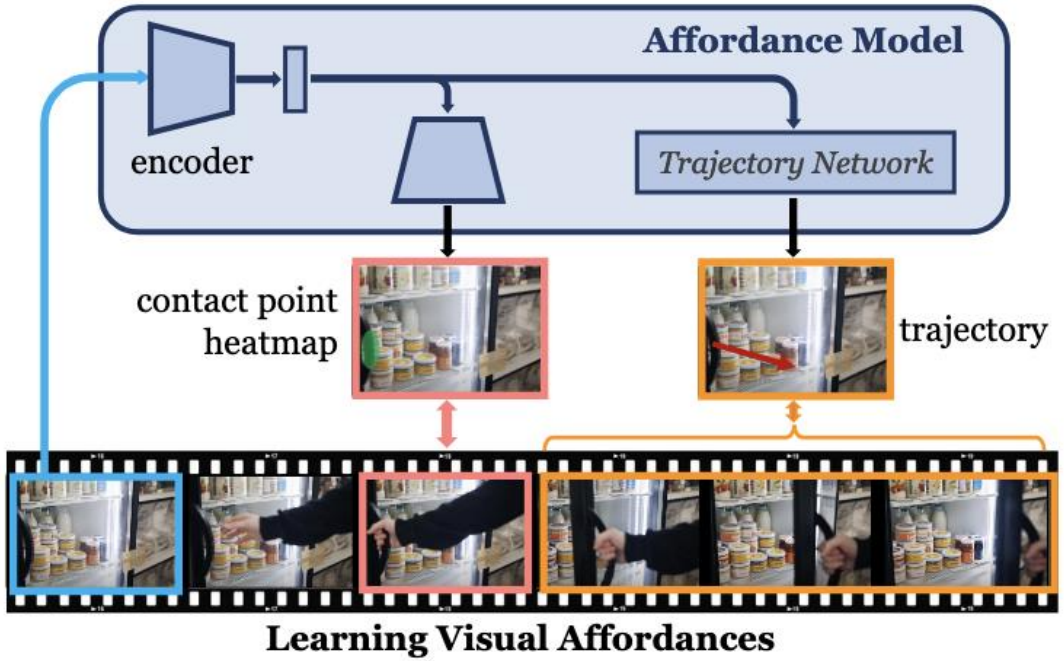
4. Accounting for camera motion over time. Compute the homography matrix H_t that aligns the image at time t to the starting frame $t_{contact}$

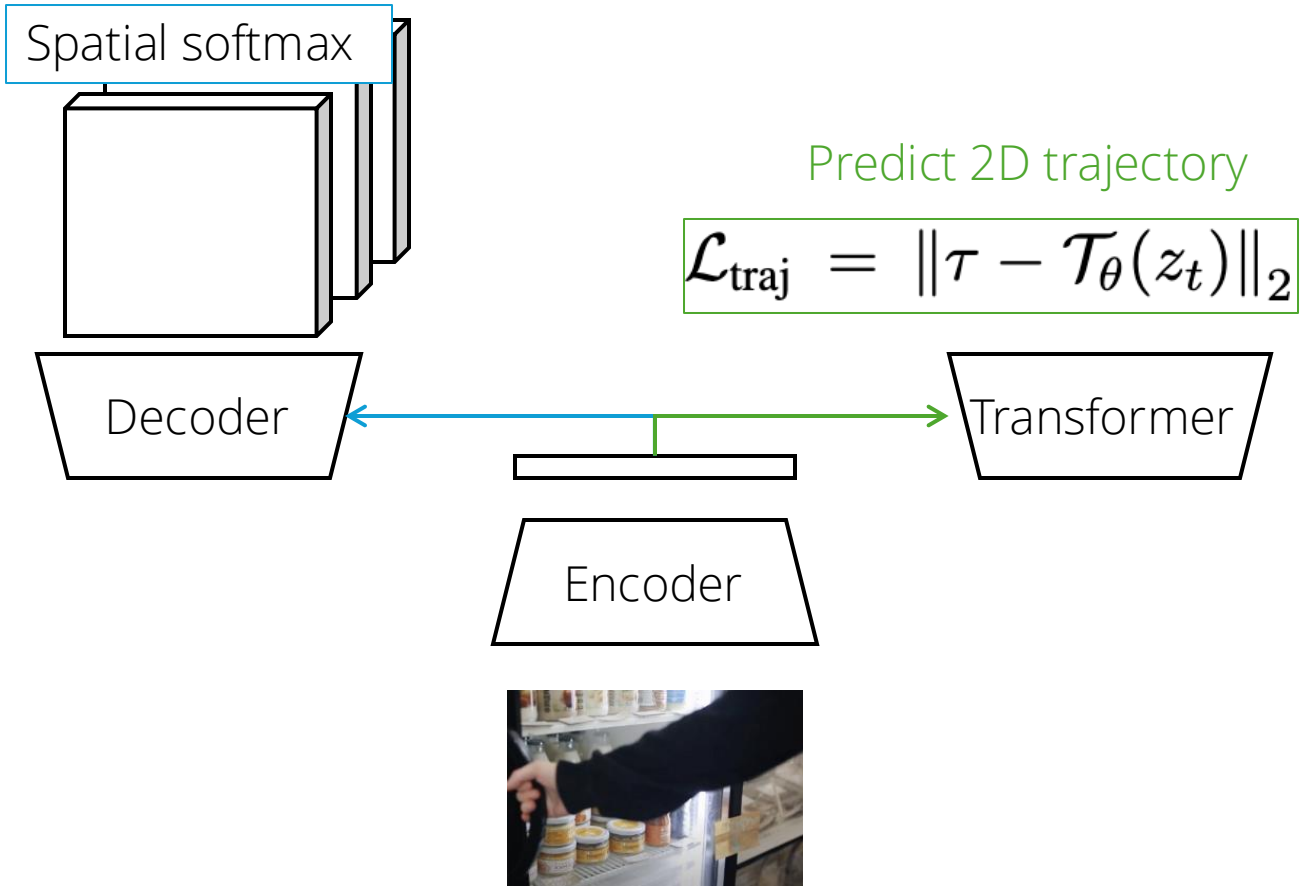
Affordances from Human Videos as a Versatile Representation for Robotics. Bahl et al.

Learn to Predict Contact Points and Post-contact Trajectory

Predict the means of the GMM

$$\mathcal{L}_{\text{contact}} = \left\| \mu_i - \sigma_{\text{2D}} \left(g_\theta^{\text{deconv}} \left(g_\theta^{\text{conv}} \left(I_t \right) \right) \right) \right\|_2$$



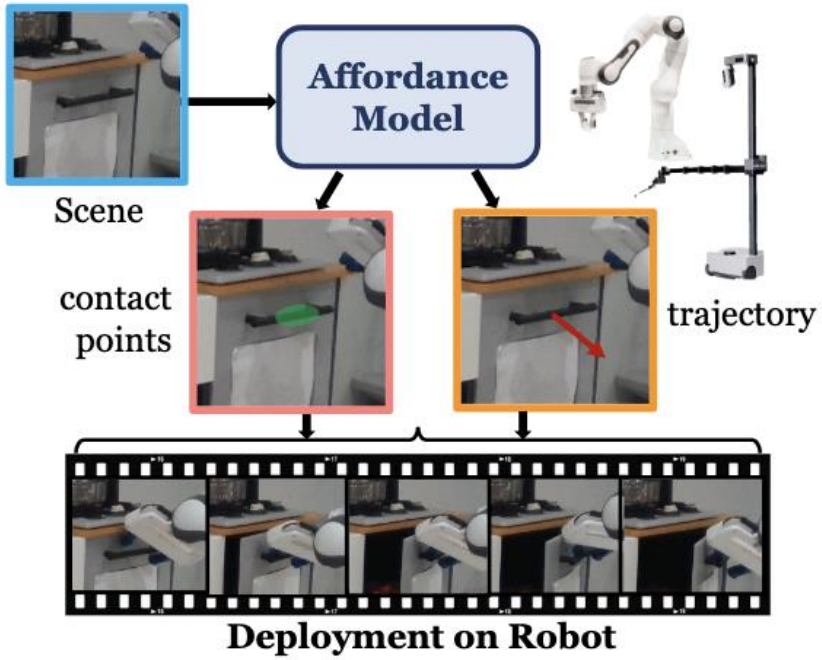


Predict 2D trajectory

$$\mathcal{L}_{\text{traj}} = \left\| \tau - \mathcal{T}_\theta(z_t) \right\|_2$$

Affordances from Human Videos as a Versatile Representation for Robotics. Bahl et al.

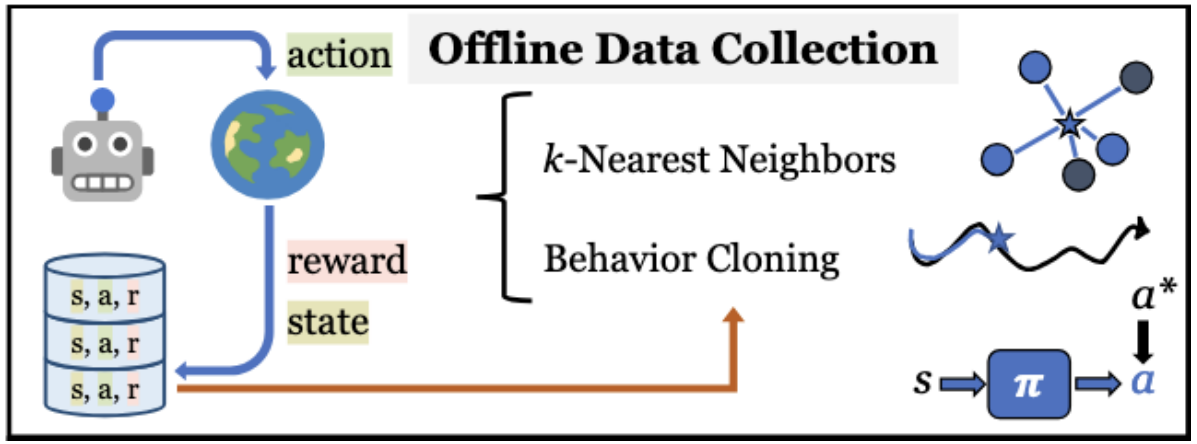
Deployment on Robots



1. Unproject 2D contact points to 3D, and use a motion planner to reach the point
2. Grasp the object
3. Unproject 2D post-contact trajectory to 3D, and move the robot's end-effector accordingly

Affordances from Human Videos as a Versatile Representation for Robotics. Bahl et al.

Application 1: Offline Data Collection

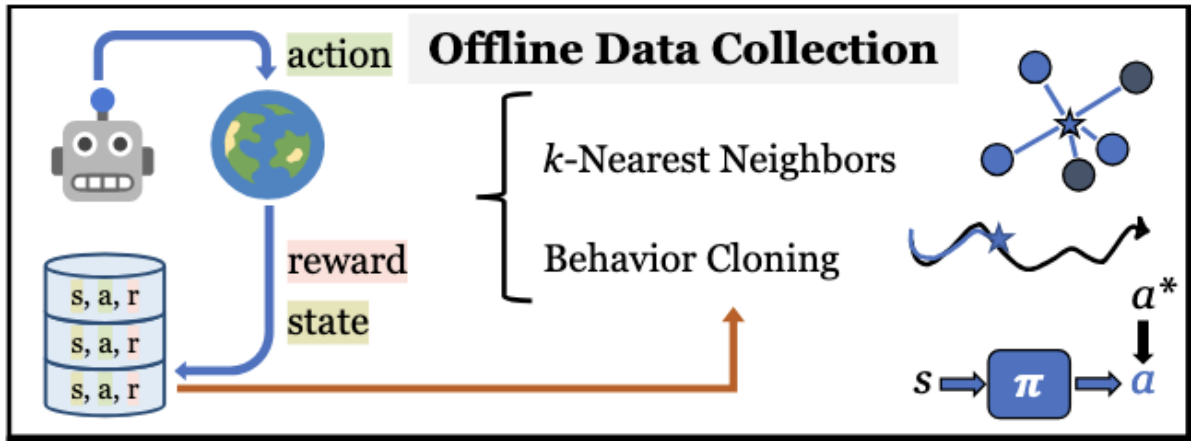


1. Collect a dataset of image, contact points and post-contact trajectories with the learned affordance model
2. KNN approach: given a goal image, retrieve the top-K trajectories.
3. Train a policy with behavior cloning on the retrieved trajectories

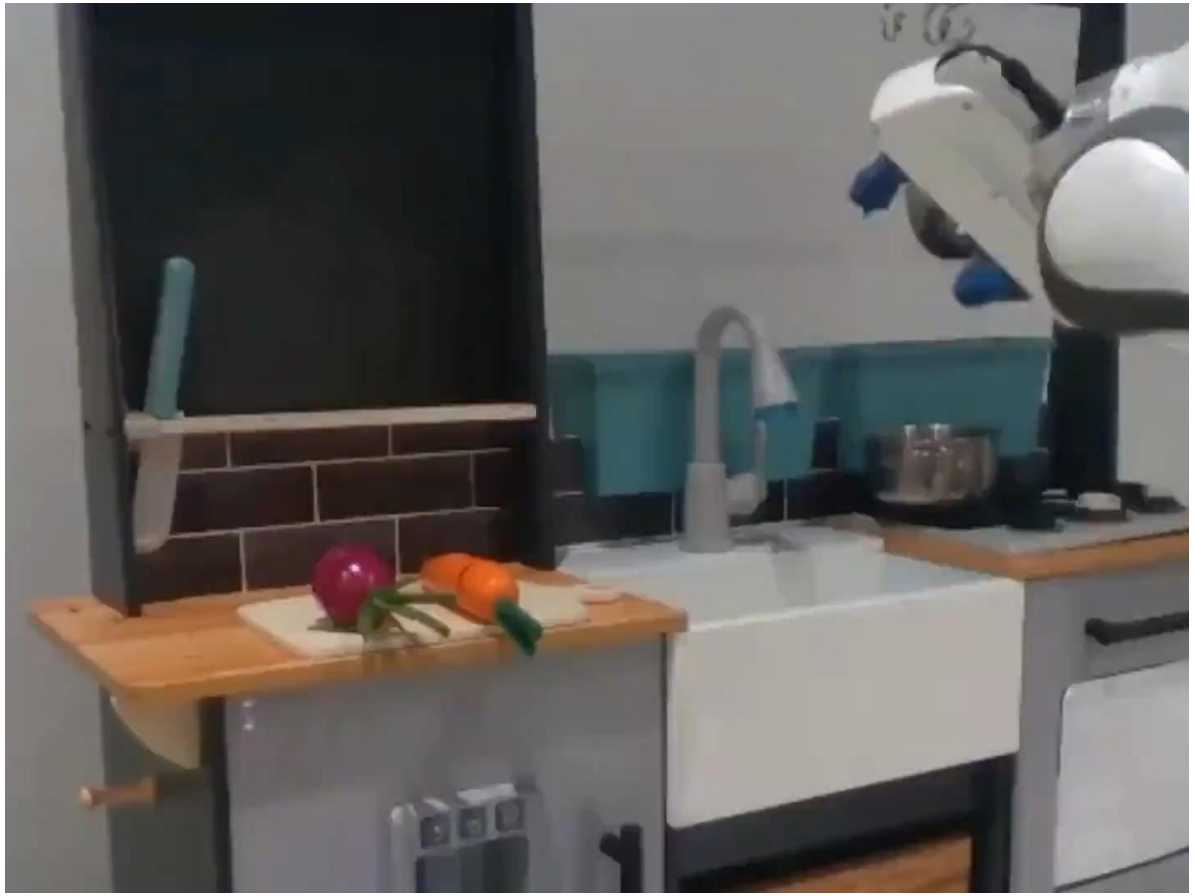
	Cabinet	Knife	Veg	Shelf	Pot	Door	Lid	Drawer
k-Nearest Neighbors:								
HOI	0.2	0.1	0.1	0.6	0.0	0.4	0.0	0.6
HAP	0.3	0.0	0.3	0.0	0.1	0.2	0.0	0.1
Hotspots	0.4	0.0	0.1	0.0	**0.5**	0.4	0.3	0.5
Random	0.3	0.0	0.1	0.3	0.4	0.2	0.1	0.2
VRB (ours)	**0.6**	**0.3**	**0.6**	**0.8**	0.4	**1.0**	**0.4**	**1.0**
Behavior Cloning:								
HOI	0.3	0.0	0.3	0.0	0.1	0.2	0.0	0.1
HAP	0.5	0.0	**0.4**	0.0	0.3	0.1	0.0	0.1
Hotspots	0.2	0.0	0.0	0.0	**0.8**	0.1	0.0	0.7
Random	0.1	**0.1**	0.1	0.0	0.2	0.1	0.0	0.0
VRB (ours)	**0.6**	**0.1**	0.3	**0.3**	**0.8**	**0.9**	**0.2**	**0.9**

Affordances from Human Videos as a Versatile Representation for Robotics. Bahl et al.

Application 1: Offline Data Collection

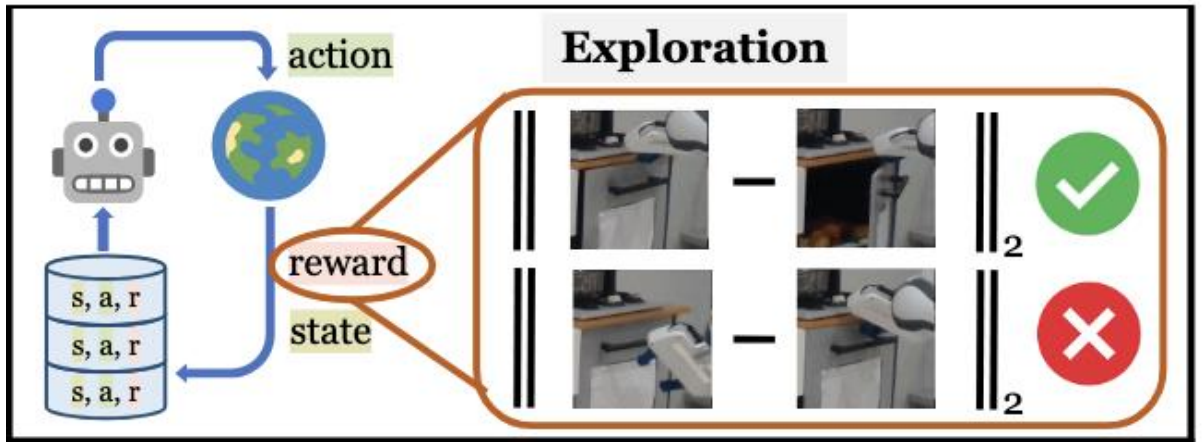


1. Collect a dataset of image, contact points and post-contact trajectories with the learned affordance model
2. KNN approach: given a goal image, retrieve the top-K trajectories.
3. Train a policy with behavior cloning on the retrieved trajectories



Affordances from Human Videos as a Versatile Representation for Robotics. Bahl et al.

Application 2: Reward-free Exploration



1. Collect a dataset with the affordance model
2. Rank all the trajectories by an exploration metric

$$\mathbf{EC}(I_i, I_j) = ||\phi(I_i) - \phi(I_j)||_2$$

3. Fit a distribution with the top-ranked trajectories.
4. Collect more data with the affordance model or the fitted distribution

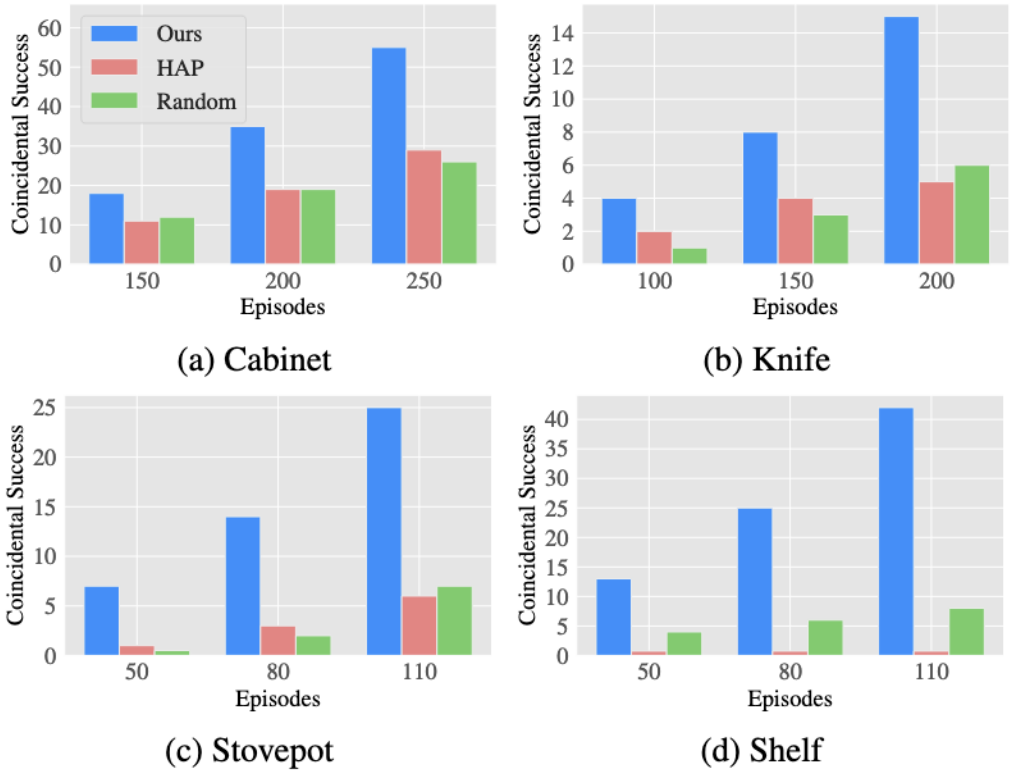
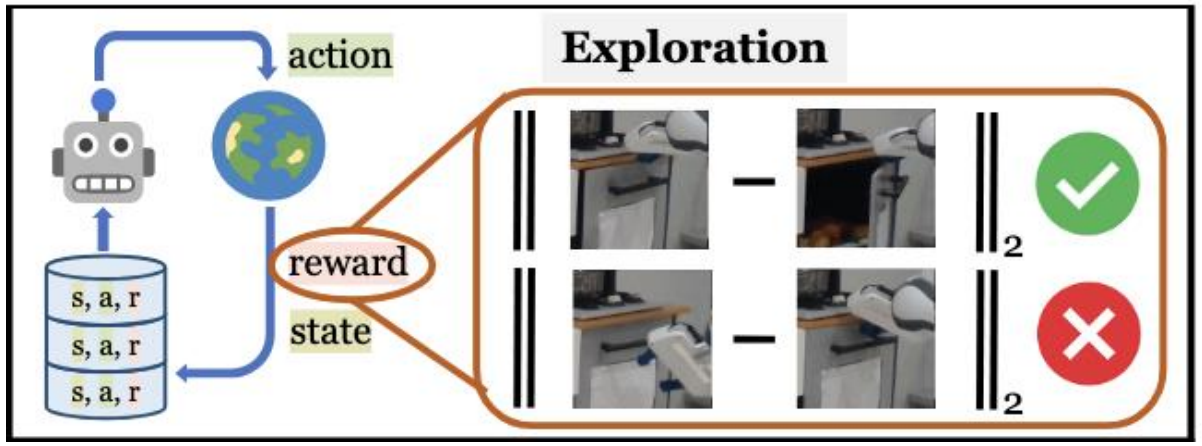


Figure 5. **Exploration**: Coincidental success of VRB in comparison to random exploration or the exploration based on HAP [39].

Affordances from Human Videos as a Versatile Representation for Robotics. Bahl et al.

Application 2: Reward-free Exploration



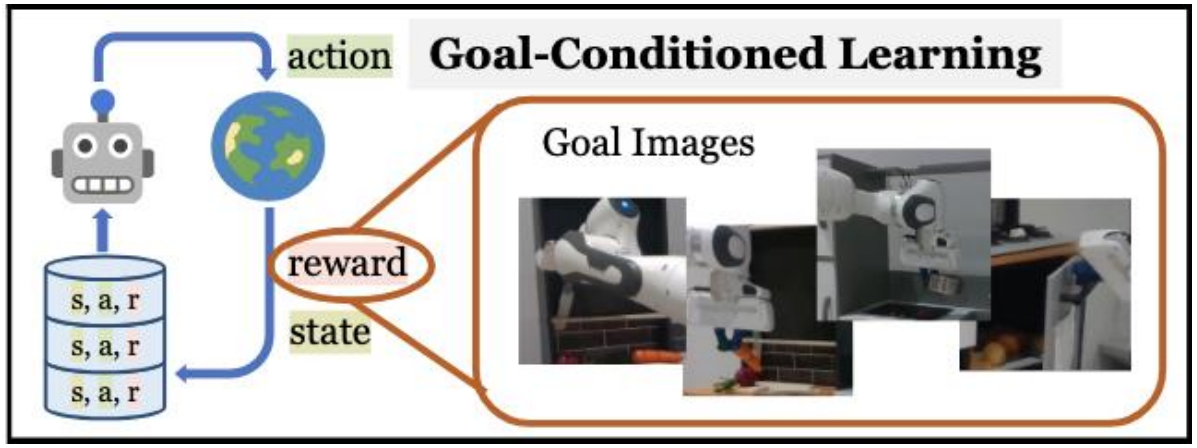
1. Collect a dataset with the affordance model
2. Rank all the trajectories by an exploration metric

$$\mathbf{EC}(I_i, I_j) = ||\phi(I_i) - \phi(I_j)||_2$$

3. Fit a distribution with the top-ranked trajectories.
4. Collect more data with the affordance model or the fitted distribution

Affordances from Human Videos as a Versatile Representation for Robotics. Bahl et al.

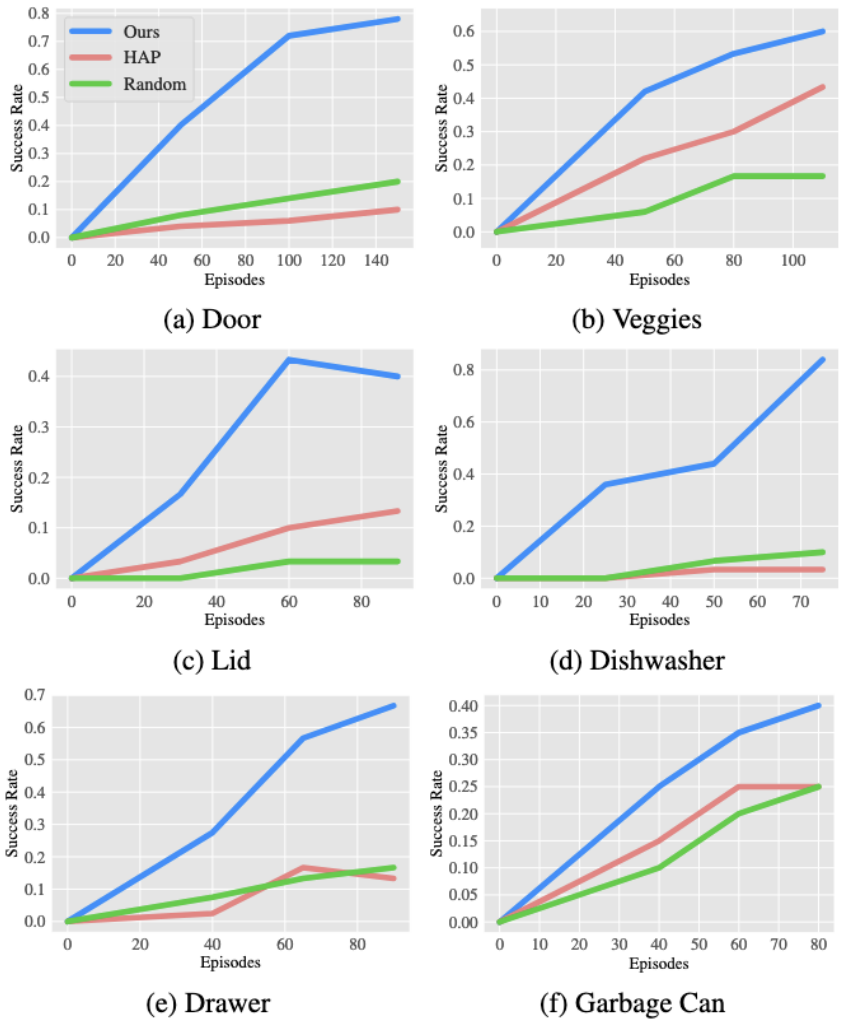
Application 3: Goal-conditioned Learning



1. Collect a dataset with the affordance model
2. Given a goal image I_g, rank all the trajectories with their last image I_T by the metric:

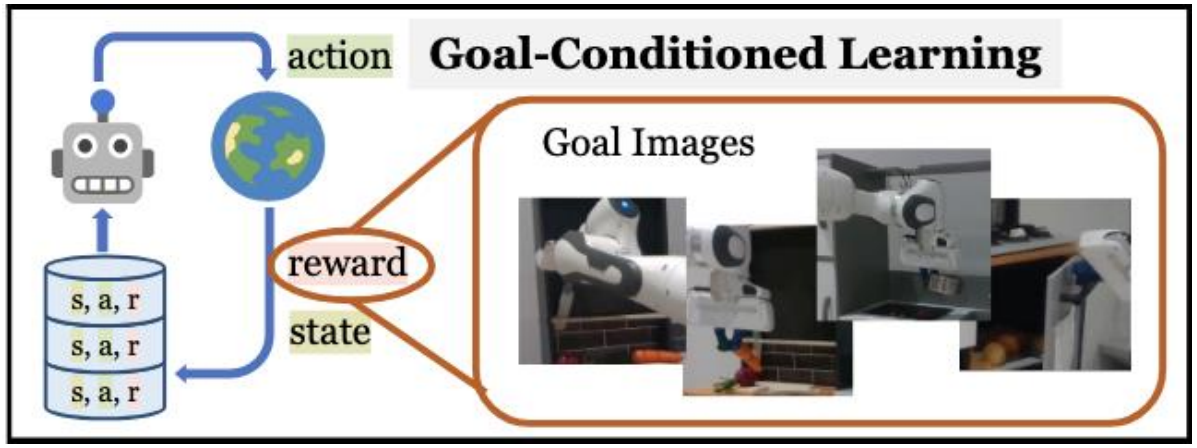
$$||\psi(I_g) - \psi(I_T)||_2^2$$

3. Fit a distribution with the top-ranked trajectories.
4. Collect more data with the affordance model or the fitted distribution
5. Train a policy with the collected data



Affordances from Human Videos as a Versatile Representation for Robotics. Bahl et al.

Application 3: Goal-conditioned Learning



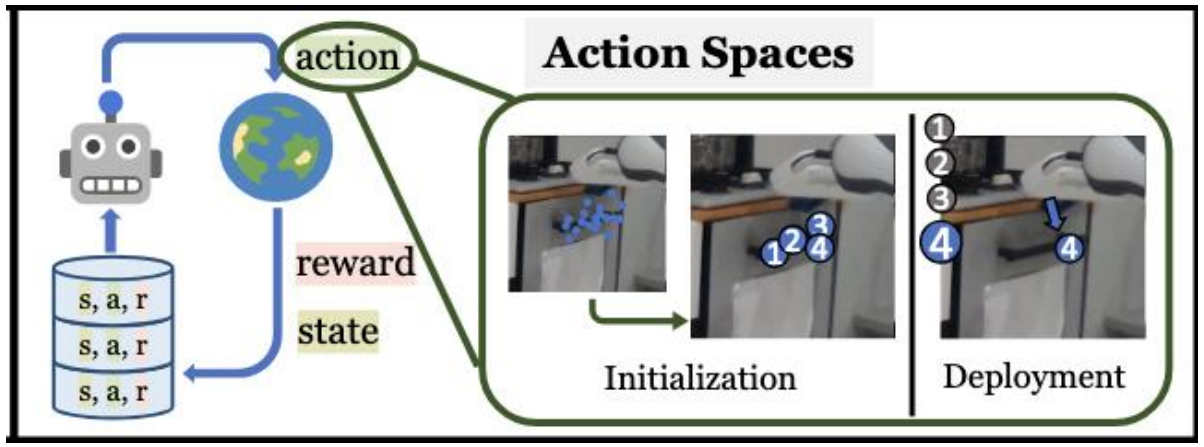
1. Collect a dataset with the affordance model
2. Given a goal image I_g, rank all the trajectories with their last image I_T by the metric:

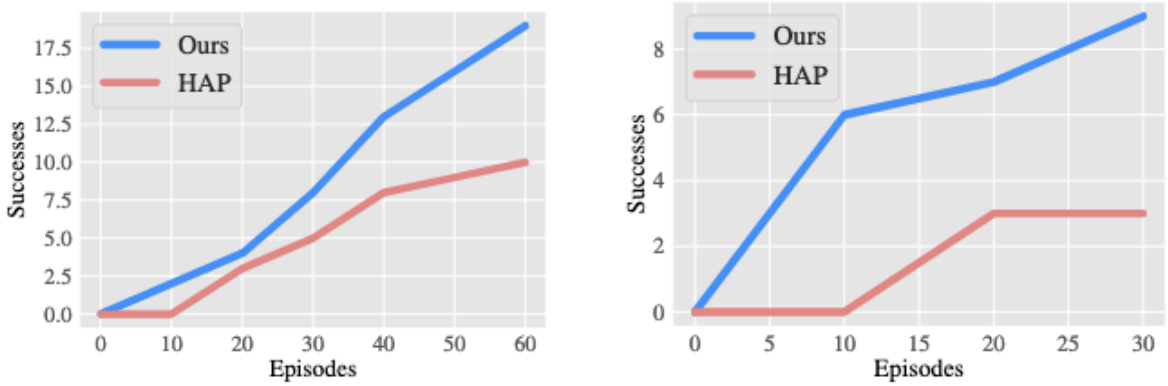
$$||\psi(I_g) - \psi(I_T)||_2^2$$

3. Fit a distribution with the top-ranked trajectories.
4. Collect more data with the affordance model or the fitted distribution
5. Train a policy with the collected data

Affordances from Human Videos as a Versatile Representation for Robotics. Bahl et al.

Application 4: More Efficient Action Space





(a) Cabinet (b) Veggies

Figure 7. **Action Space**: Success using DQN with the discretized action space, for reaching a specified goal image.

Affordances from Human Videos as a Versatile Representation for Robotics. Bahl et al.

Affordance Only Specifies the Pick-and-Move Motion of Gripper How about More Dexterous Tasks like Spinning Pen?

We need a finer-grained representation of robot-object interaction

Where **gripper should** interact with the object

Contact points

How **gripper should** interact with the object

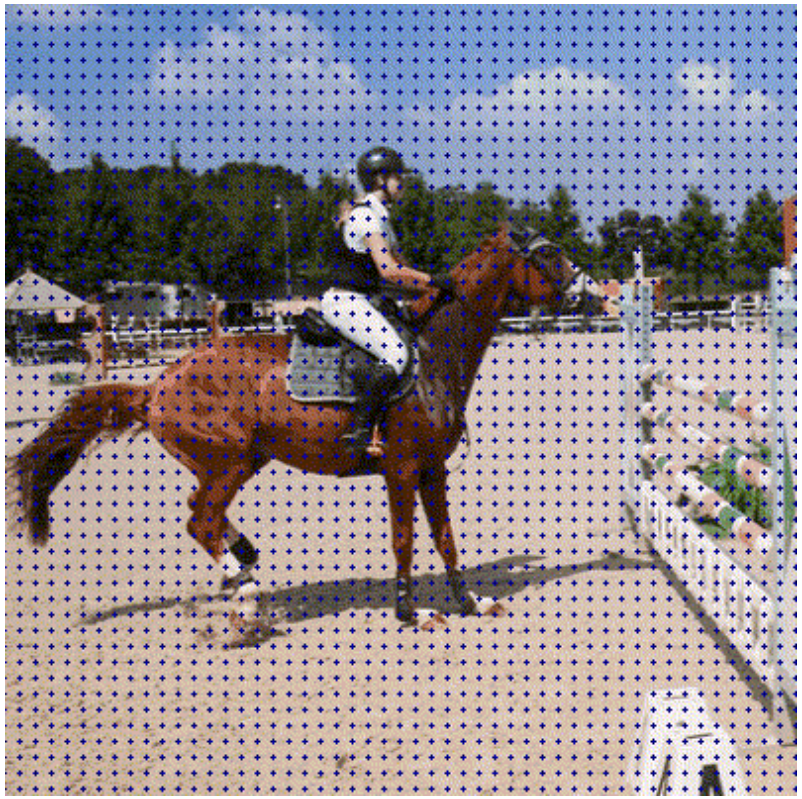
Wrist trajectory

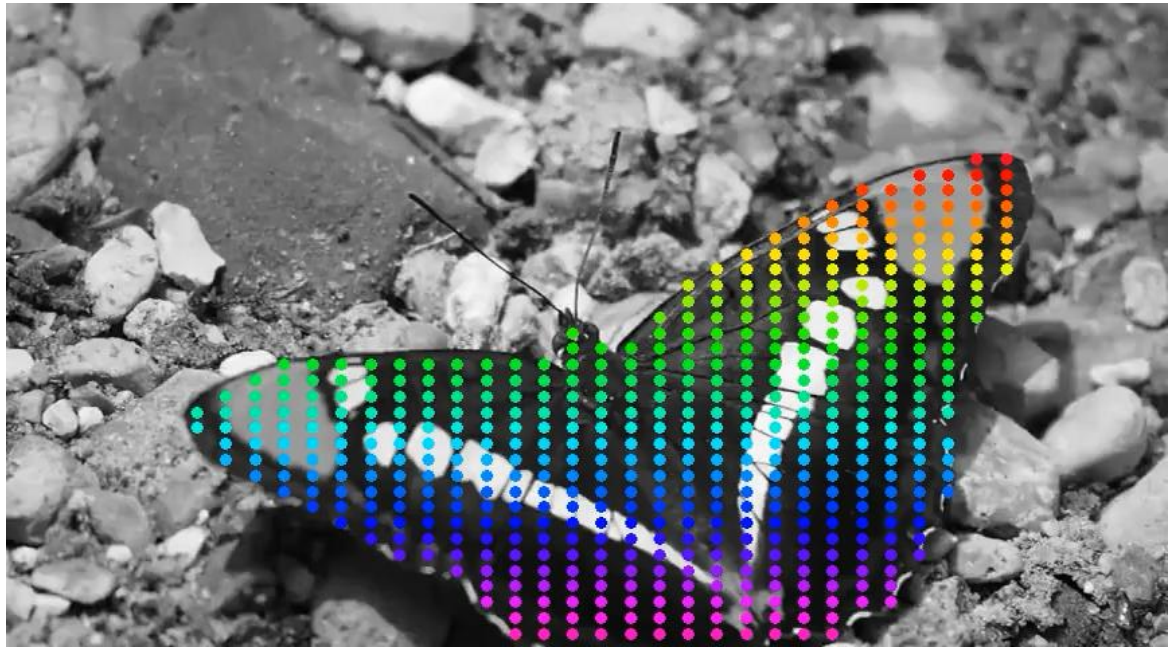
Affordances from Human Videos as a Versatile Representation for Robotics. Bahl et al.

What information should we extract from human demonstration videos?

- Types of visual imitation learning:
  1. Knowledge distillation via representation learning
  2. Knowledge distillation via video generation
  3. Knowledge distillation with correspondence
  4. Knowledge distillation with affordance
  5. Knowledge distillation with point trajectories
  6. Knowledge distillation with 3D hand modeling
  7. Knowledge distillation with digital twins

Point Trajectories: A Fine-grained Representation that Capture Dynamics

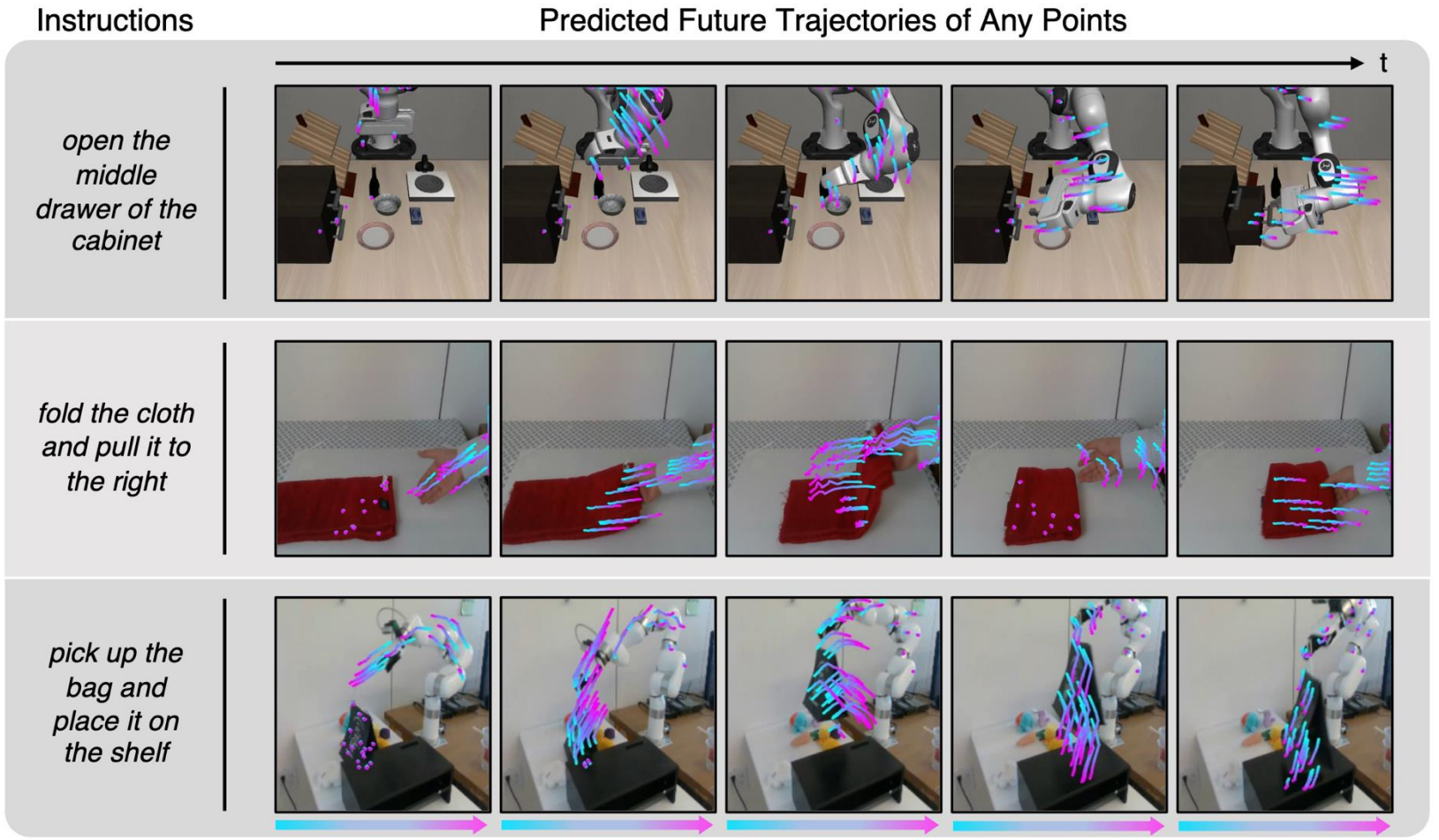




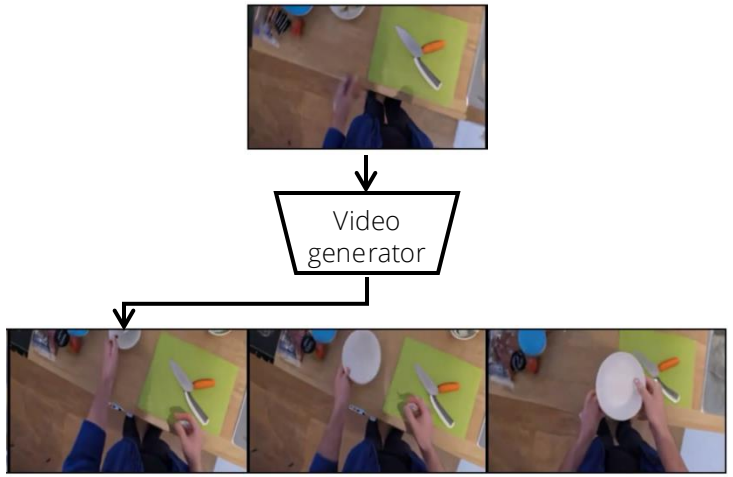
Particle Video Revisited: Tracking Through Occlusions Using Point Trajectories. Harley et al.

SpatialTracker: Tracking Any 2D Pixels in 3D Space. Xiao et al.

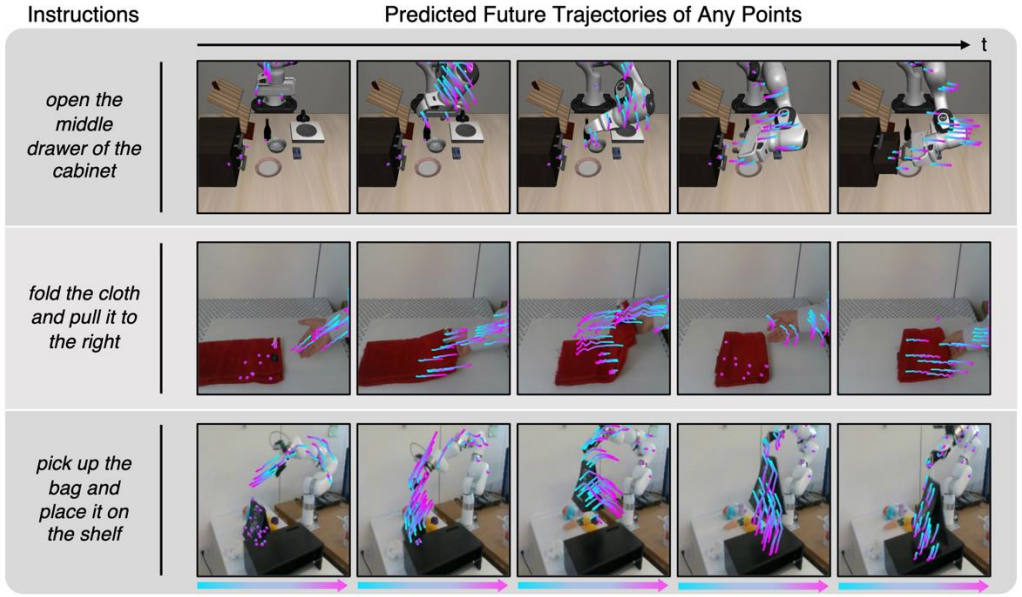
Knowledge distillation with Affordance



Video Generation vs. Point Trajectory Prediction

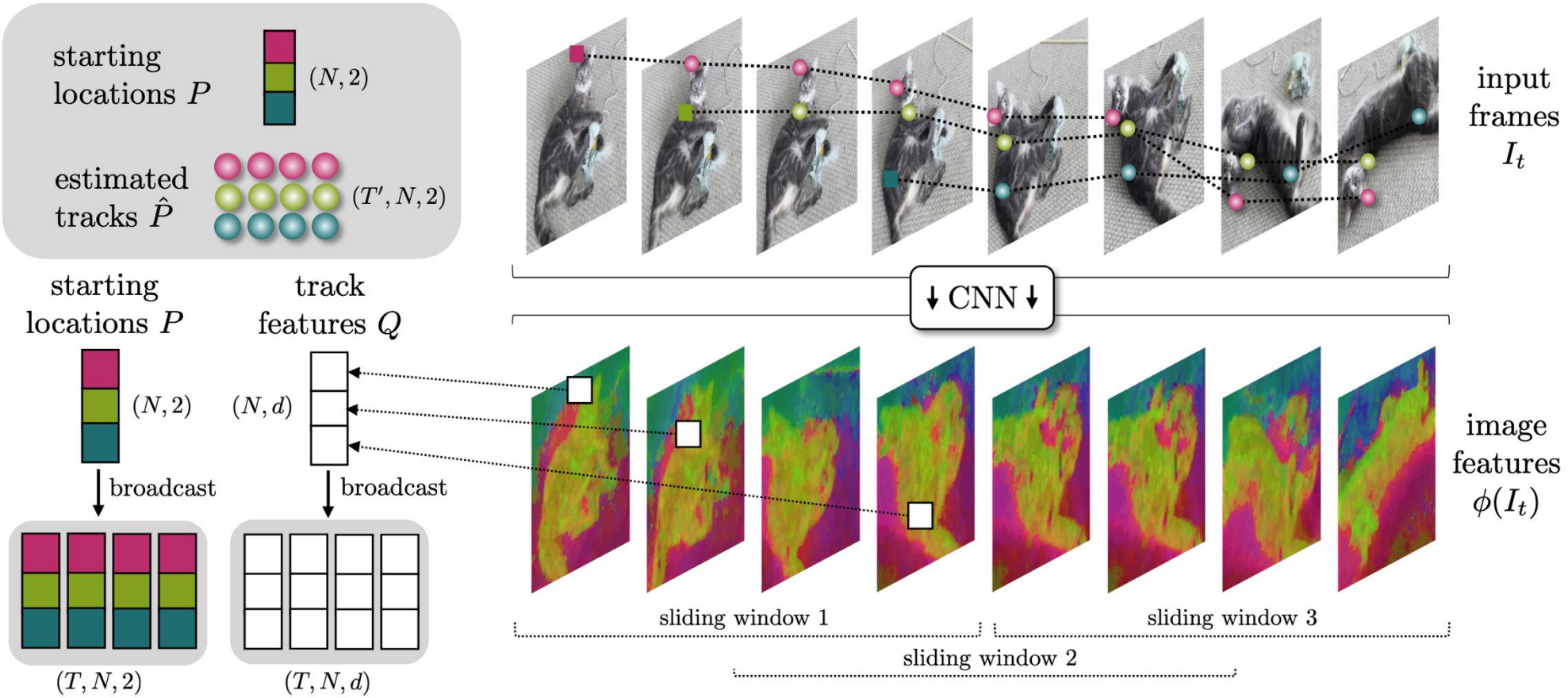


- Implicit dynamics: actions can be inferred by
  - ➢ Inverse dynamic model



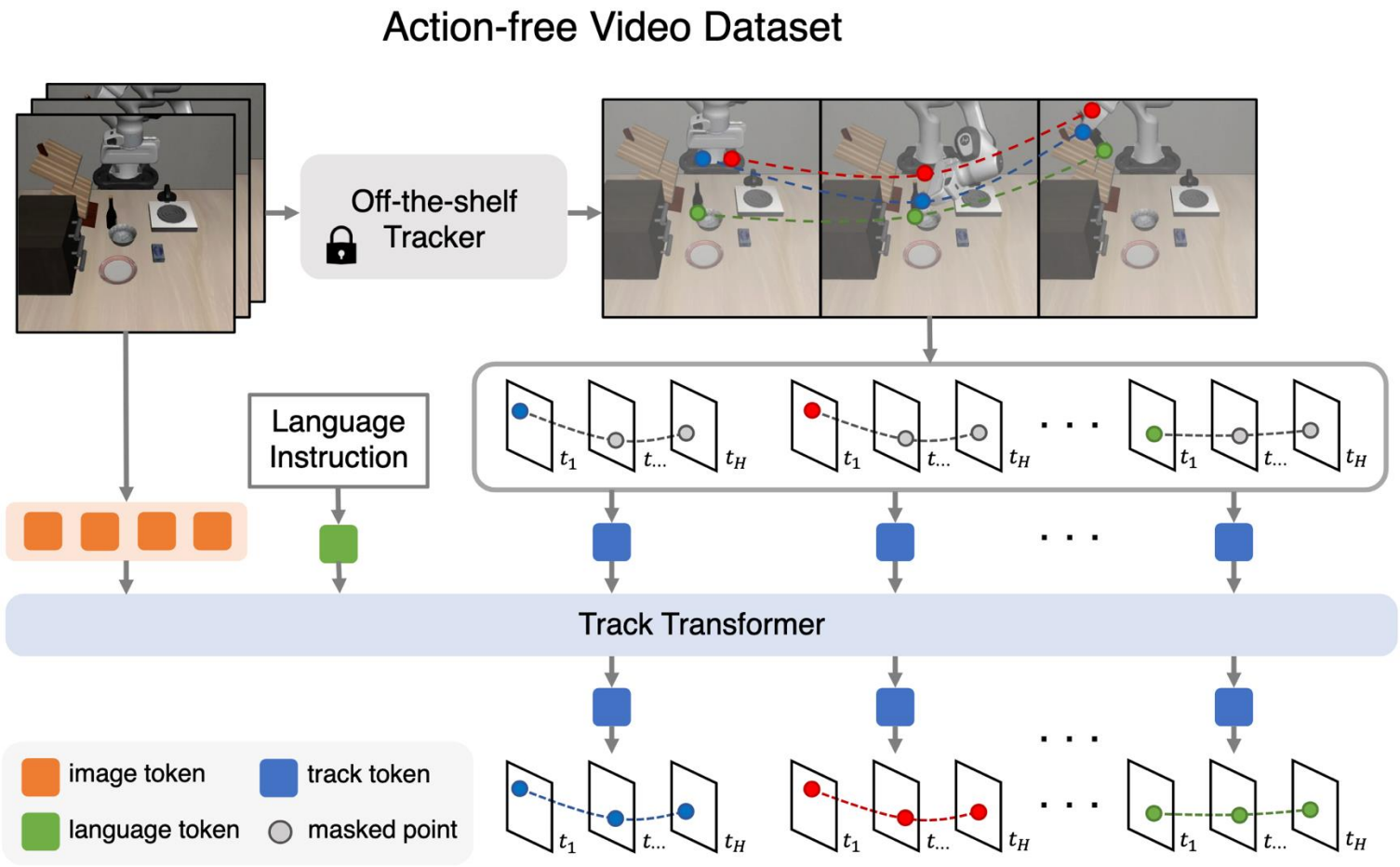
- Explicit dynamics: actions can be inferred by
  - ➢ Inverse dynamic model
  - ➢ Rigid transformation of 3D point trajectories

Annotate Action-less Videos with Off-the-shelf Point Trackers

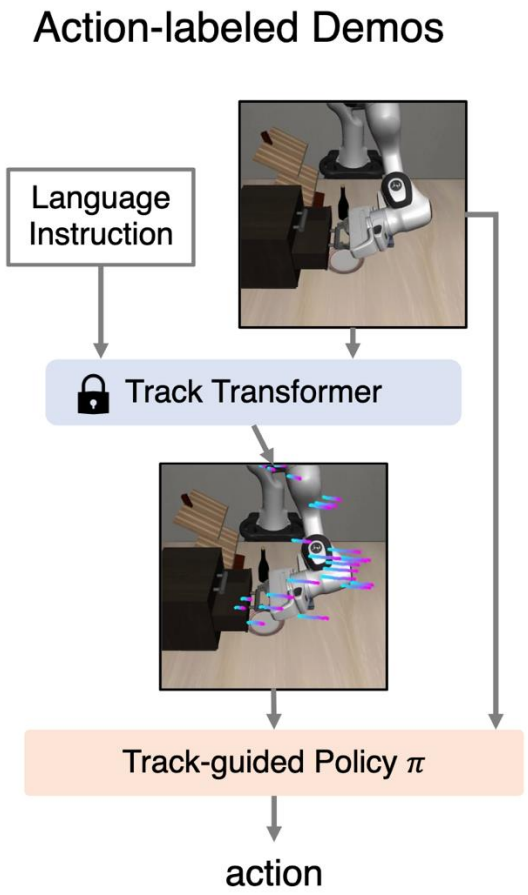


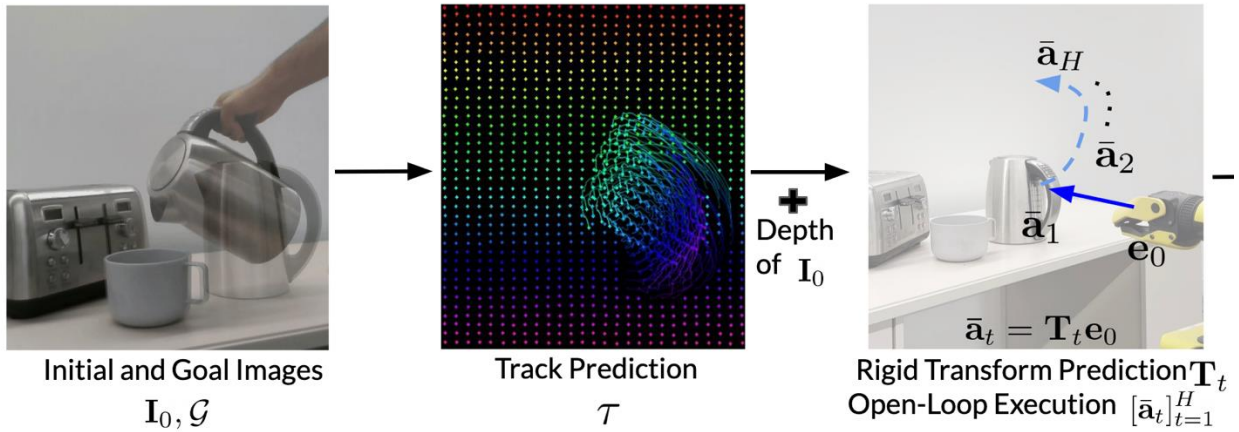
CoTracker: It is Better to Track Together. Karaev et al.

Train a Track Transformer Based on the Annotated Point Trajectories



Infer Actions from Predicted Point Trajectories

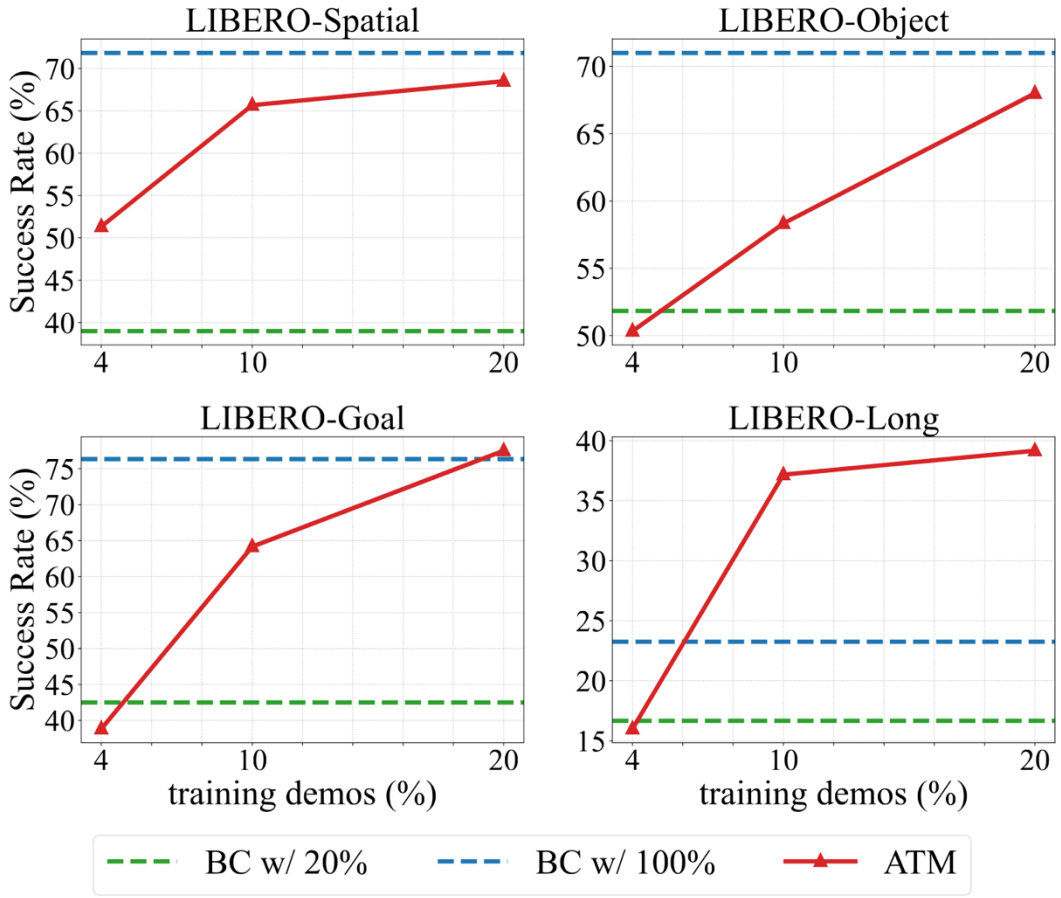


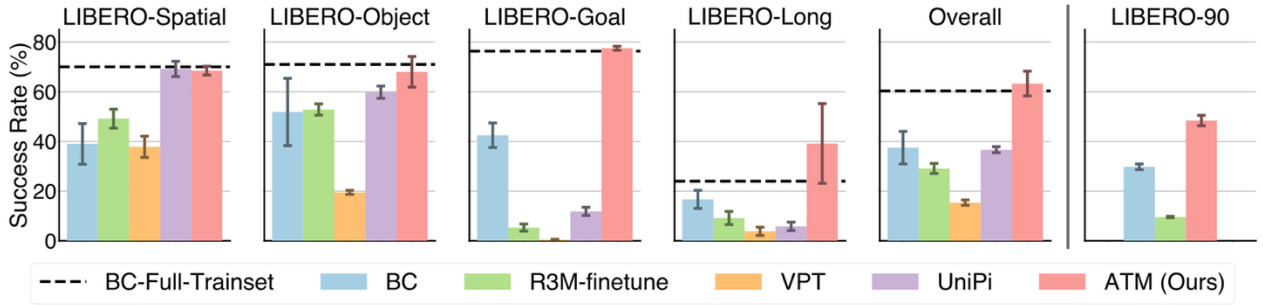


Any-point Trajectory Modeling for Policy Learning. Wen et al.

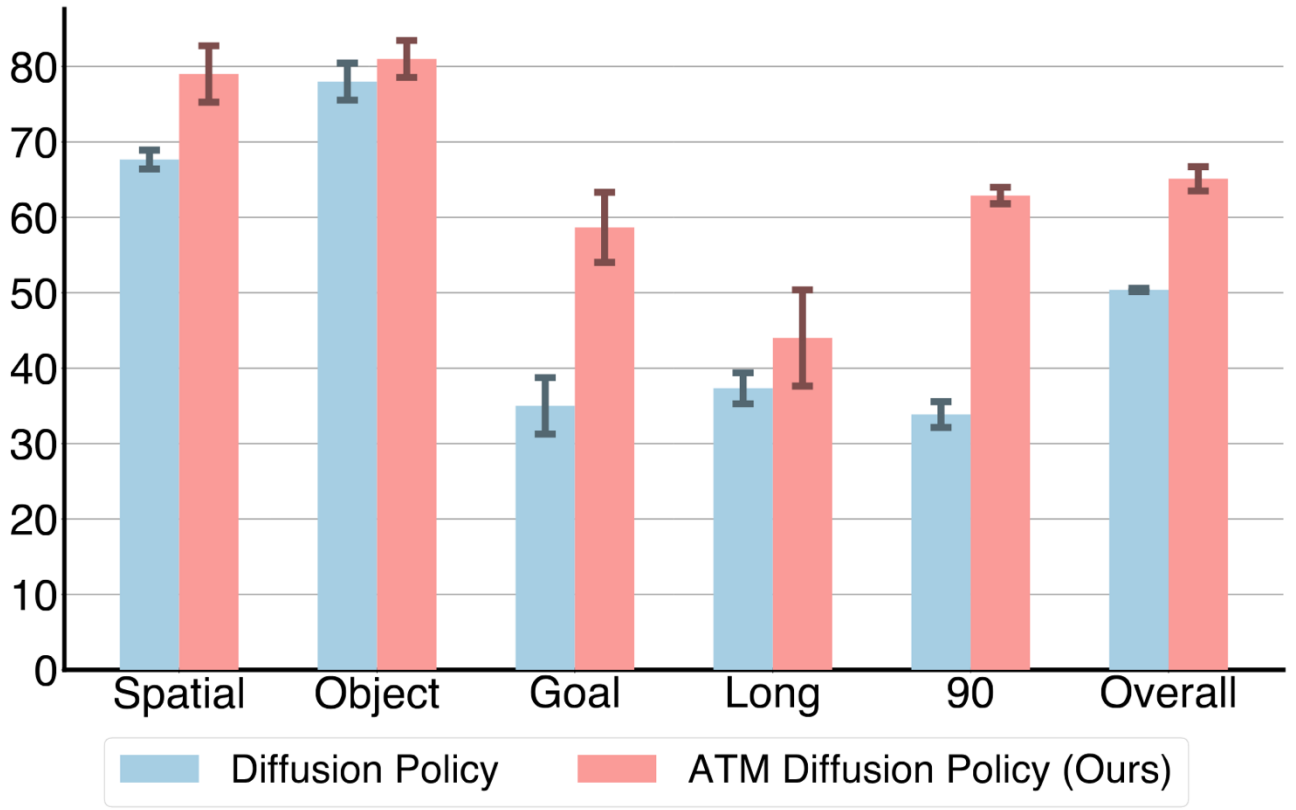
Track2Act: Predicting Point Tracks from Internet Videos enables Generalizable Robot Manipulation. Bharadhwaj et al.

Results: Point Trajectories Extract Knowledge from Actionless Videos and Improves Sample Efficiency





Results: Policies Conditioned on Point Trajectories Outperform Image-Conditioned Policies



Point Trajectories is a Cross-Embodiment Representation

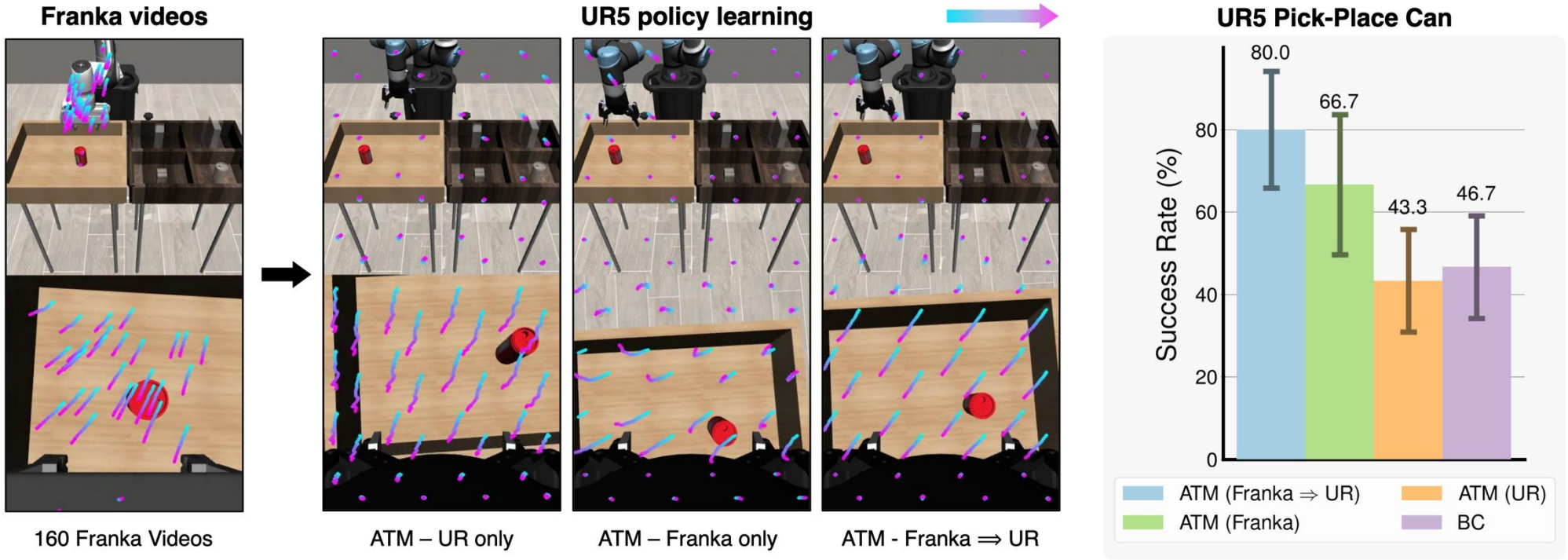
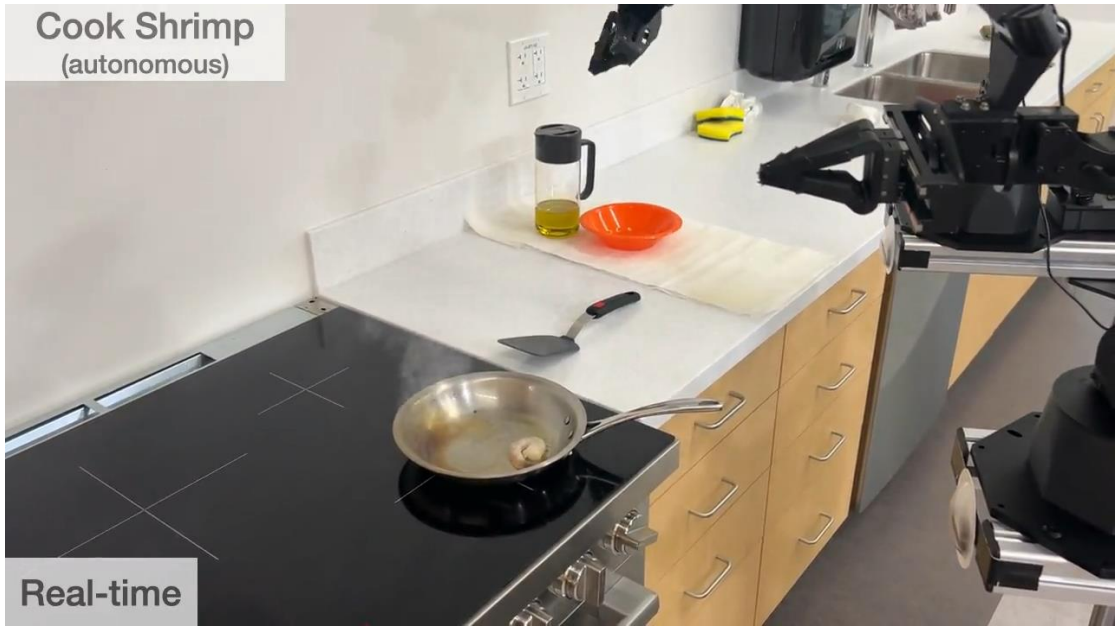


Fig. 8: Cross-morphology skill transfer for a pick-and-place task. Here, we collect 160 action-free videos of a Franka arm and 10 action-labeled demonstrations from a UR arm, with the final goal of learning a UR policy. We compare a vanilla BC baseline with ATM trained using types of data: using only the 10 UR videos, using only the 160 Franka videos, and using both Franka and UR videos (Franka ⇒ UR). In the right plot, we observe that the additional cross-embodiment data led to significantly better results. Surprisingly, even if the trajectory model is only trained using Franka videos, it exhibits much better performance than the BC without the Franka videos.

What information should we extract from human demonstration videos?

- Types of visual imitation learning:
  1. Knowledge distillation via representation learning
  2. Knowledge distillation via video generation
  3. Knowledge distillation with correspondence
  4. Knowledge distillation with affordance
  5. Knowledge distillation with point trajectories
  6. Knowledge distillation with 3D hand modeling
  7. Knowledge distillation with digital twins

Dexterous Manipulation Needs Multiple Fingers



https://youtu.be/fT8tQhovT0k?si=dVdkbkd1AJQJWEJi

Mobile ALOHA Learning Bimanual Mobile Manipulation with Low-Cost Whole-Body Teleoperation. Fu et al.

Human Hand Has 27 Degrees of Freedom
Learning with RL is Hard…

- Finger has 4 DoF
- Thumb has 5 DoF
- Wrist has 6 DoF

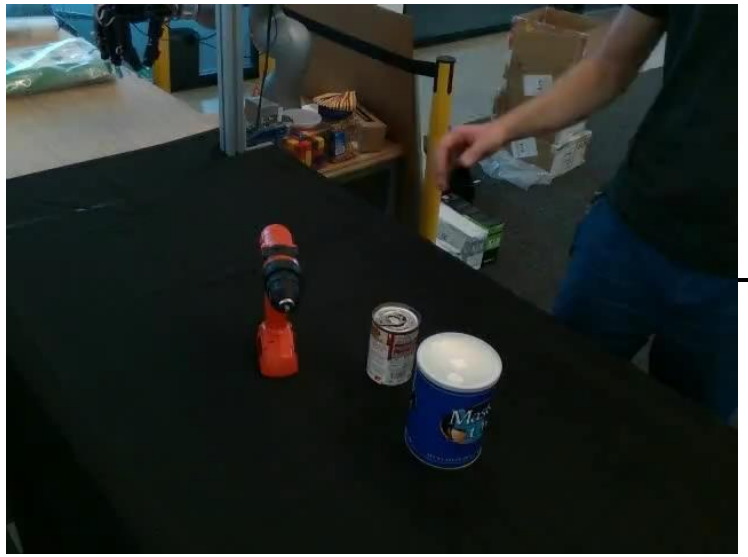
- We have 4 fingers, 1 thumb and 1 wrist, ending up with 27 DoF

https://biology.stackexchange.com/questions/30857/does-the-human-hand-have-27-degrees-of-freedom

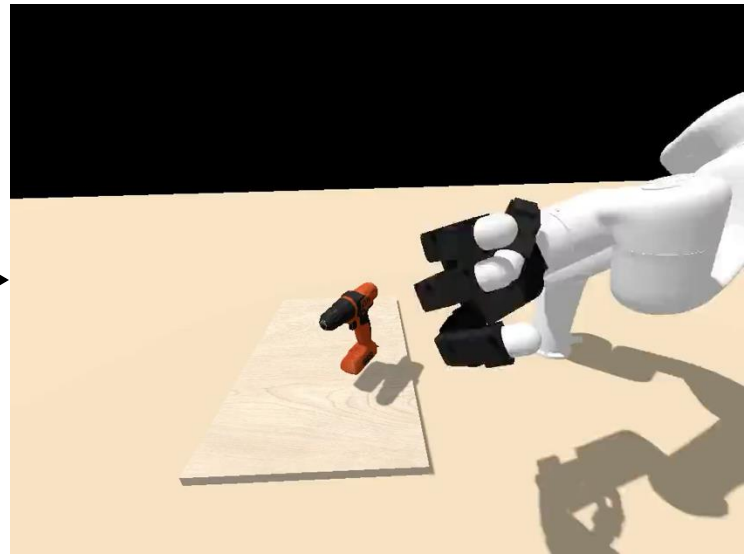
Computer Vision Works with Predicting 3D Hand Models from Actionless Videos

HaMeR: Hand Mesh Recovery. Pavlakos et al.

Knowledge distillation with 3D Hand Modeling

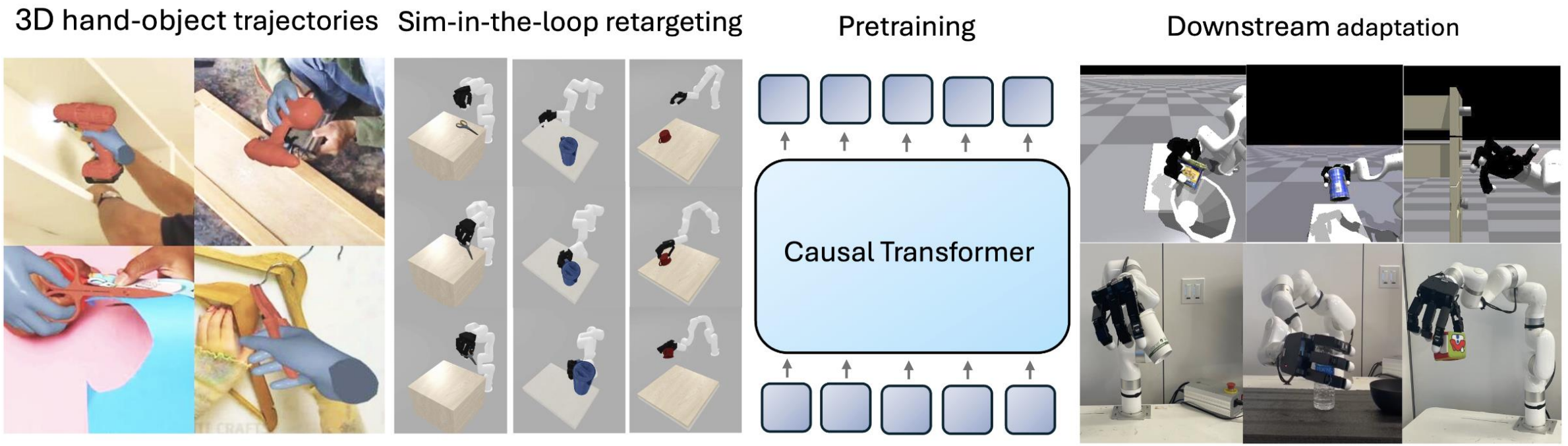


Retarget to 3D motion of human hand and the object to those in the simulator



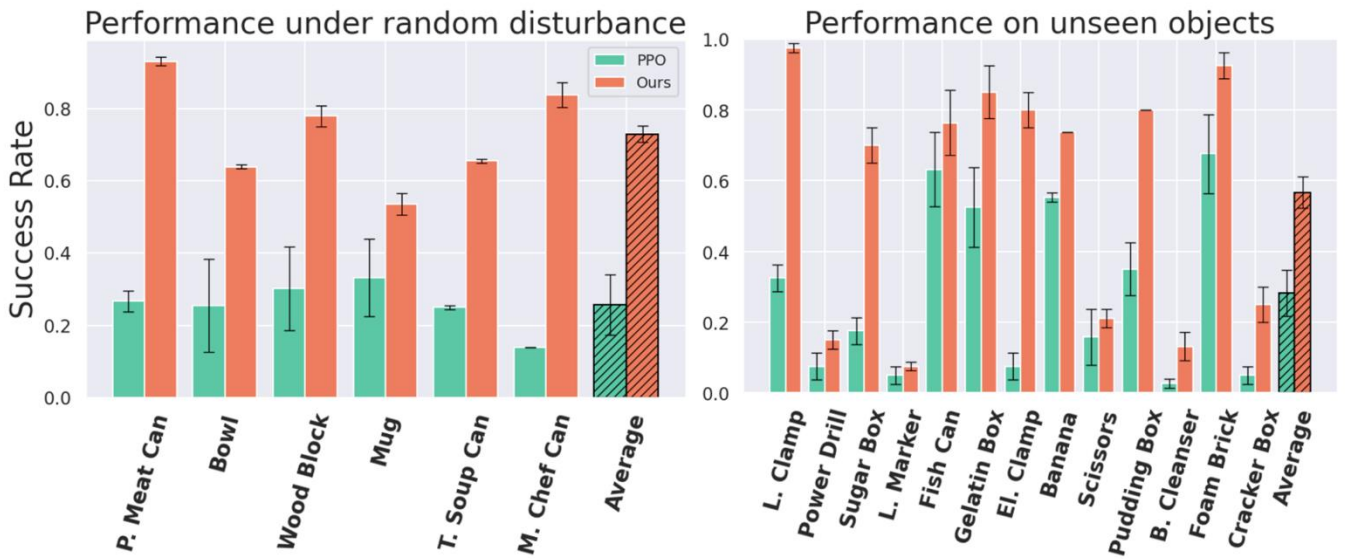
$$\min_{\boldsymbol{a}[k]} \frac{1}{2}\|\boldsymbol{x}_h[k] - f(\boldsymbol{a}[k])\|^2 + \lambda\|\boldsymbol{a}[k] - \boldsymbol{\phi}[k-1]\|^2 \quad \text{s.t.} \quad \boldsymbol{a}[k] \in \mathbb{A},$$

Pre-train Policies with 3D Hand Trajectories and Fine-tune on Downstream Tasks with RL / BC



$$\mathcal{L}(\tau;\theta) = \mathbb{E}_{t\sim[1\ldots T]}\left[\|\boldsymbol{a}[t-L:t] - \pi_b(\boldsymbol{o}[t-L:t])\|_1\right].$$

Results of RL: Better than Learning from Scratch and Other Demonstration-guided Baseline



Modeling the Hand-Object Motion is Better than only the Hand Motion

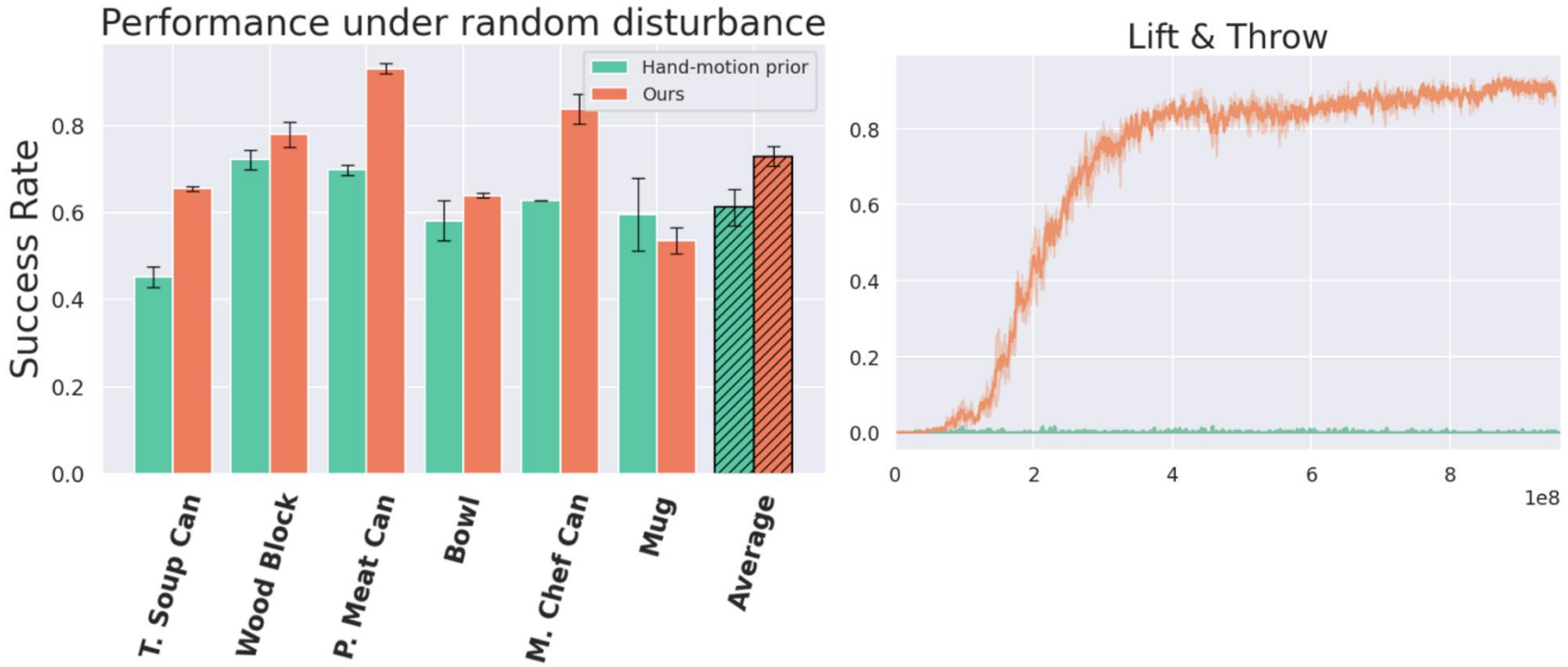
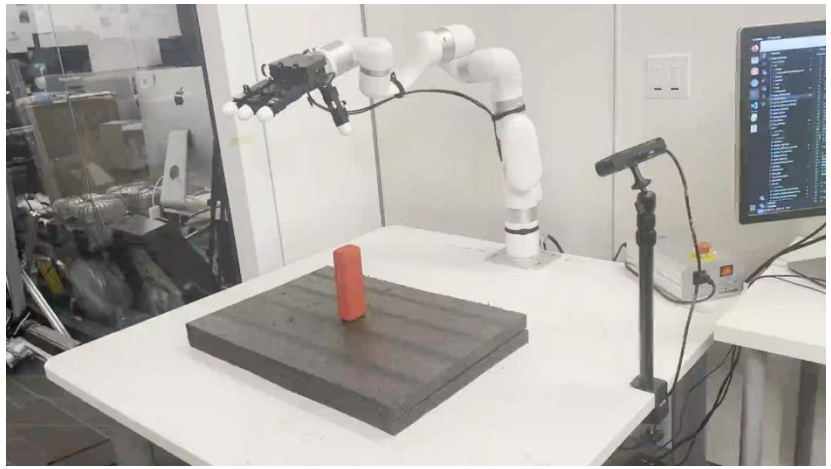
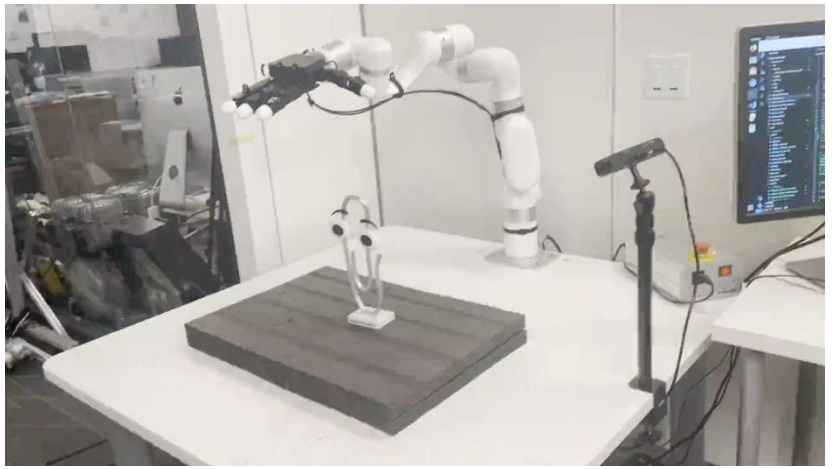
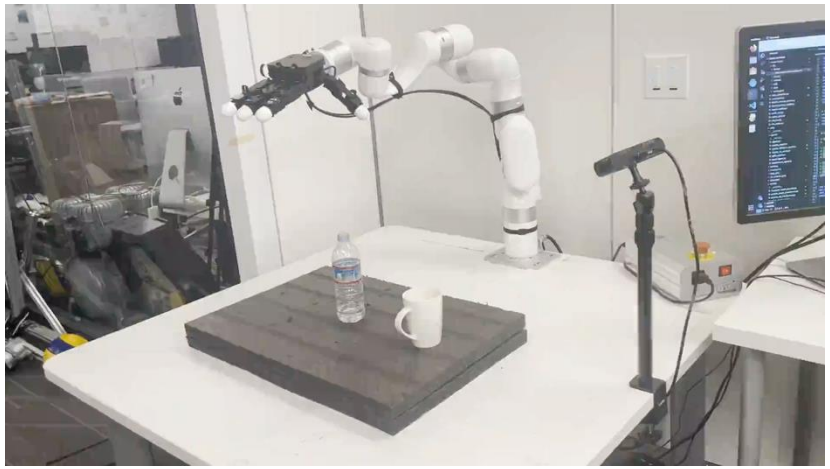


Figure 6: Pre-training only a hand-motion prior leads to decrease in robustness of grasps to force disturbances (left). With our approach, the pre-trained policy learns a prior on object affordances which leads to more robust grasps. In addition, pretraining with object poses leads to a more flexible prior and better finetuning to tasks less aligned with the pre-training data (right).

Results

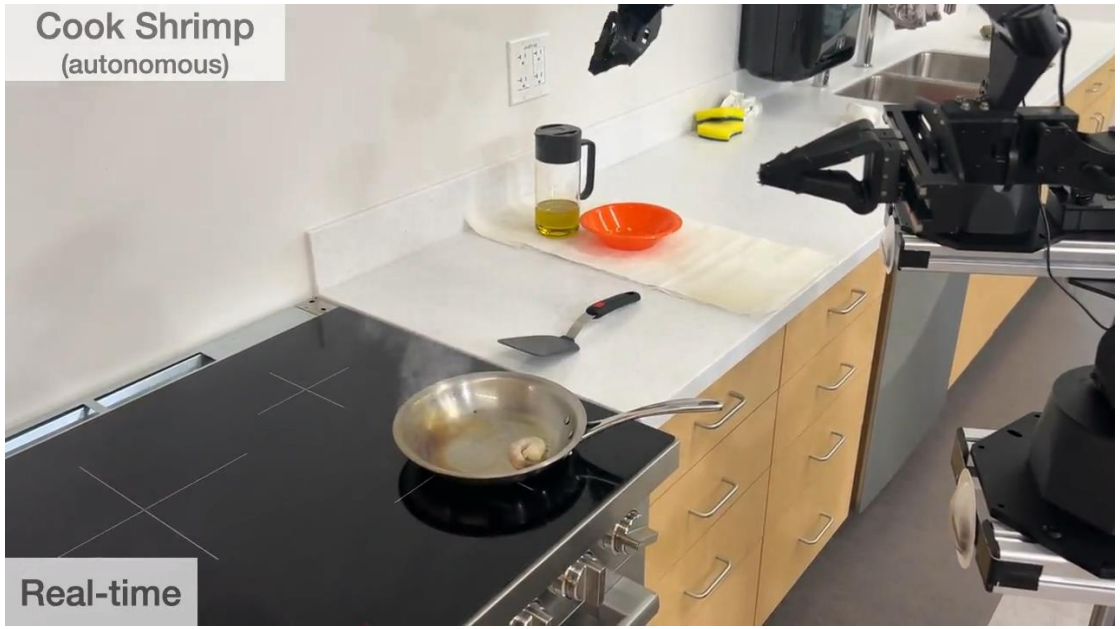
We'll talk more about bimanual dexterous manipulation in the next lecture!

What information should we extract from human demonstration videos?

- Types of visual imitation learning:
  1. Knowledge distillation via representation learning
  2. Knowledge distillation via video generation
  3. Knowledge distillation with correspondence
  4. Knowledge distillation with affordance
  5. Knowledge distillation with point trajectories
  6. Knowledge distillation with 3D hand modeling
  7. Knowledge distillation with digital twins

Mismatch of Training and Testing Distribution

Train



Test

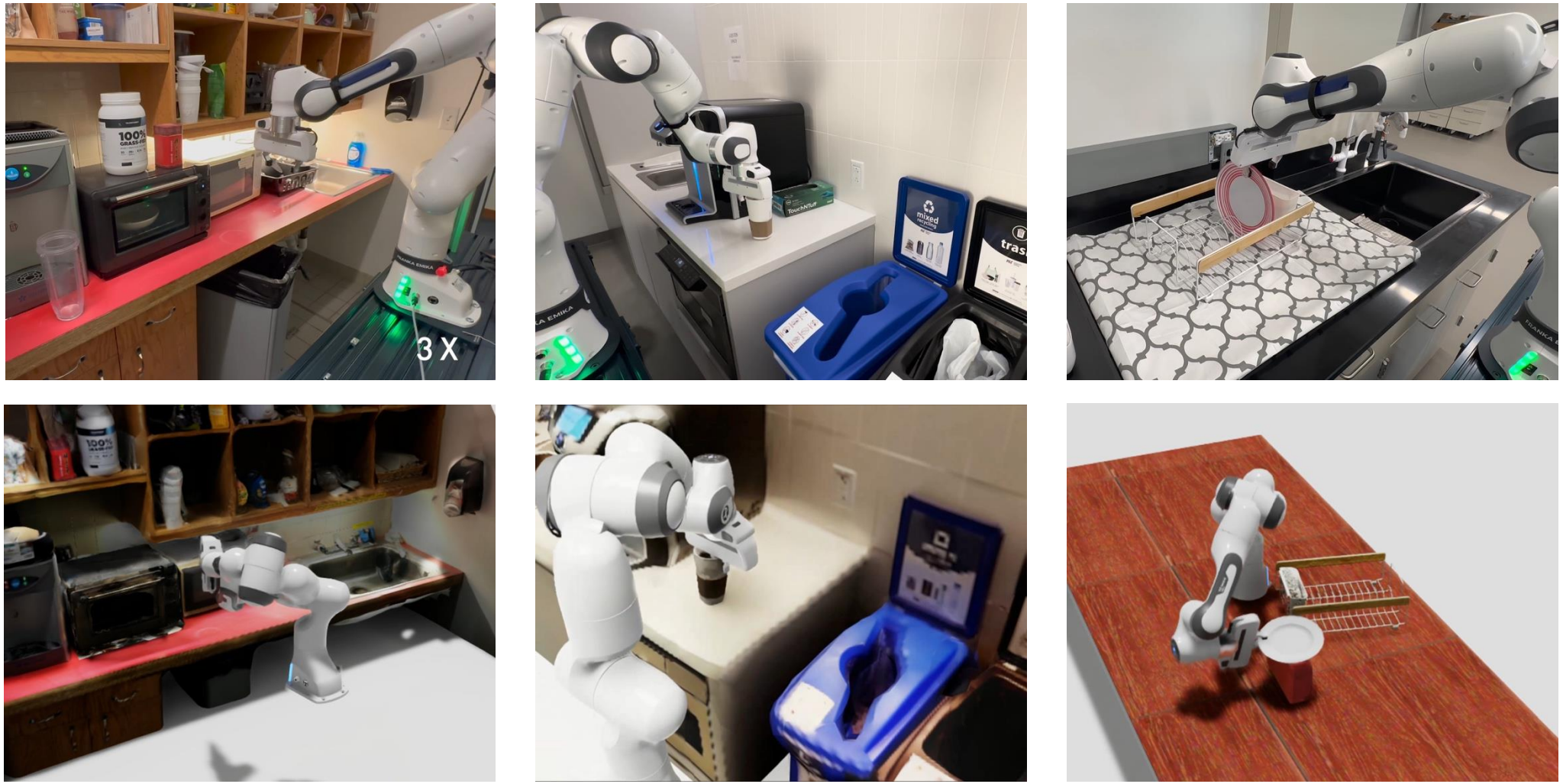
https://youtu.be/fT8tQhovT0k?si=dVdkbkd1AJQJWEJi

Idea: train on testing scenes. But How?

Mobile ALOHA Learning Bimanual Mobile Manipulation with Low-Cost Whole-Body Teleoperation. Fu et al.

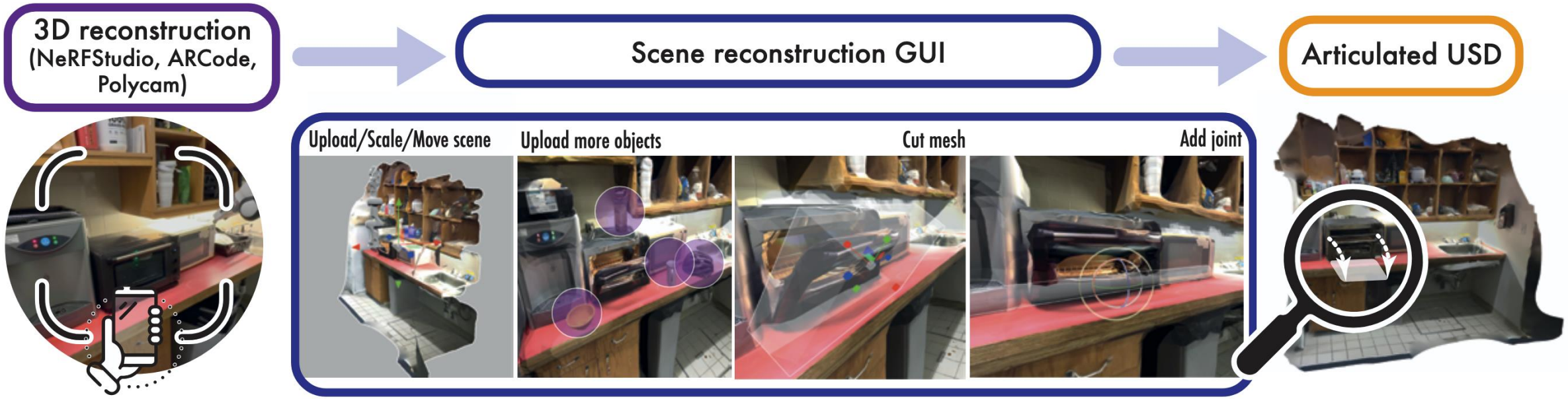
Step 1: Build the Same Scene in Simulation

3D reconstruction from multi-view images



Reconciling Reality Through Simulation: A Real-to-Sim-to-Real Approach for Robust Manipulation. Torne et al.

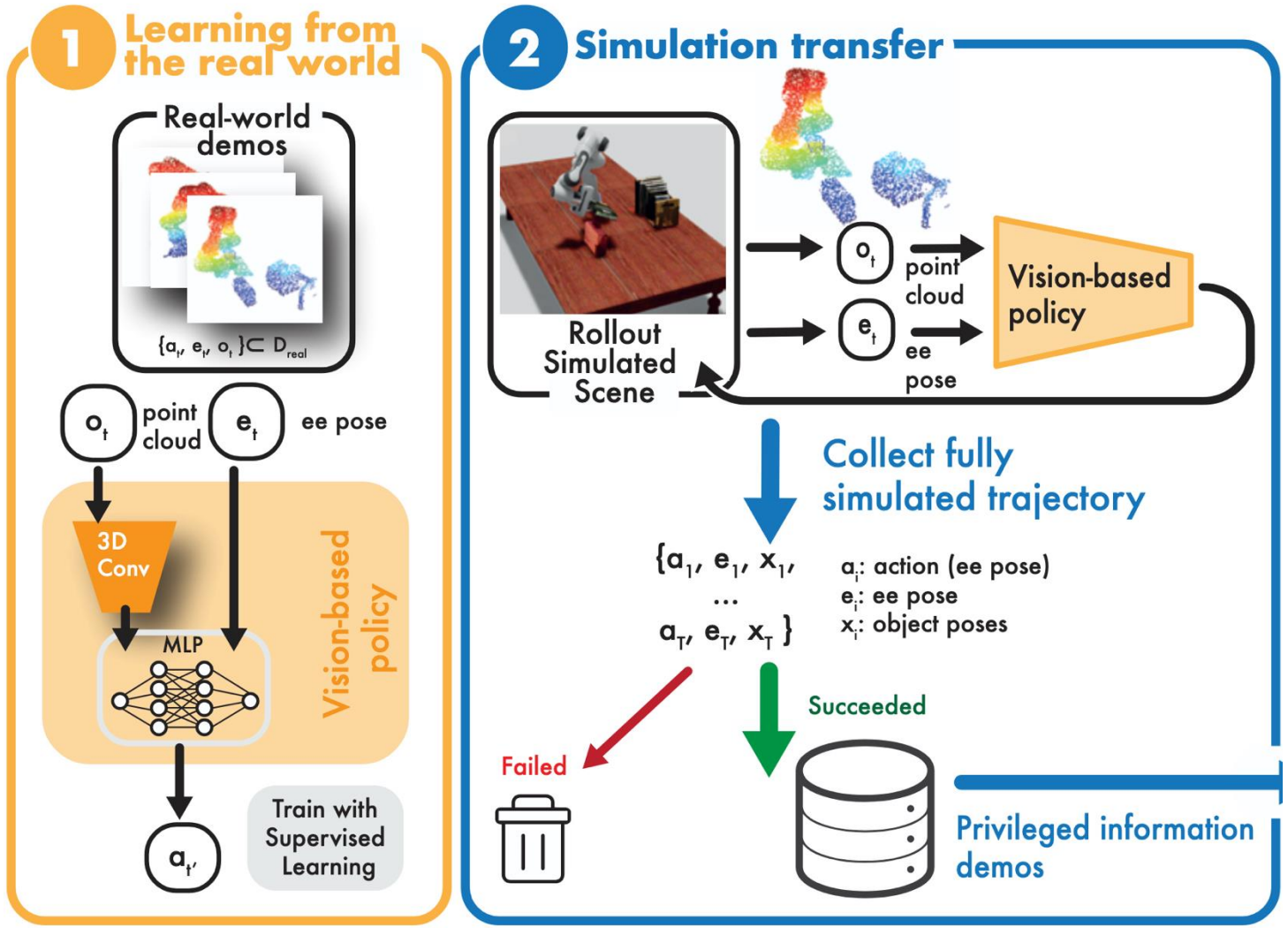
What Need to be Reconstructed?



Rigid-body objects, articulated objects

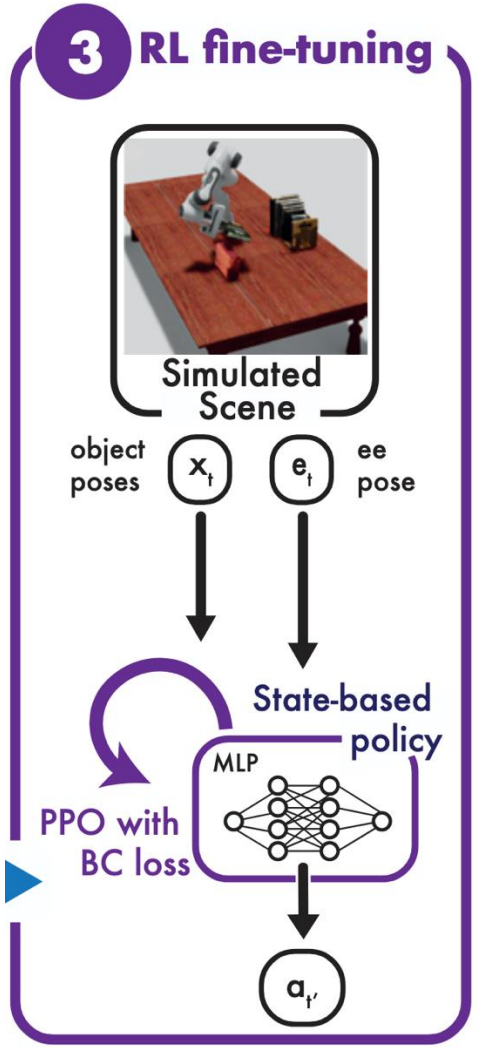
Reconciling Reality Through Simulation: A Real-to-Sim-to-Real Approach for Robust Manipulation. Torne et al.

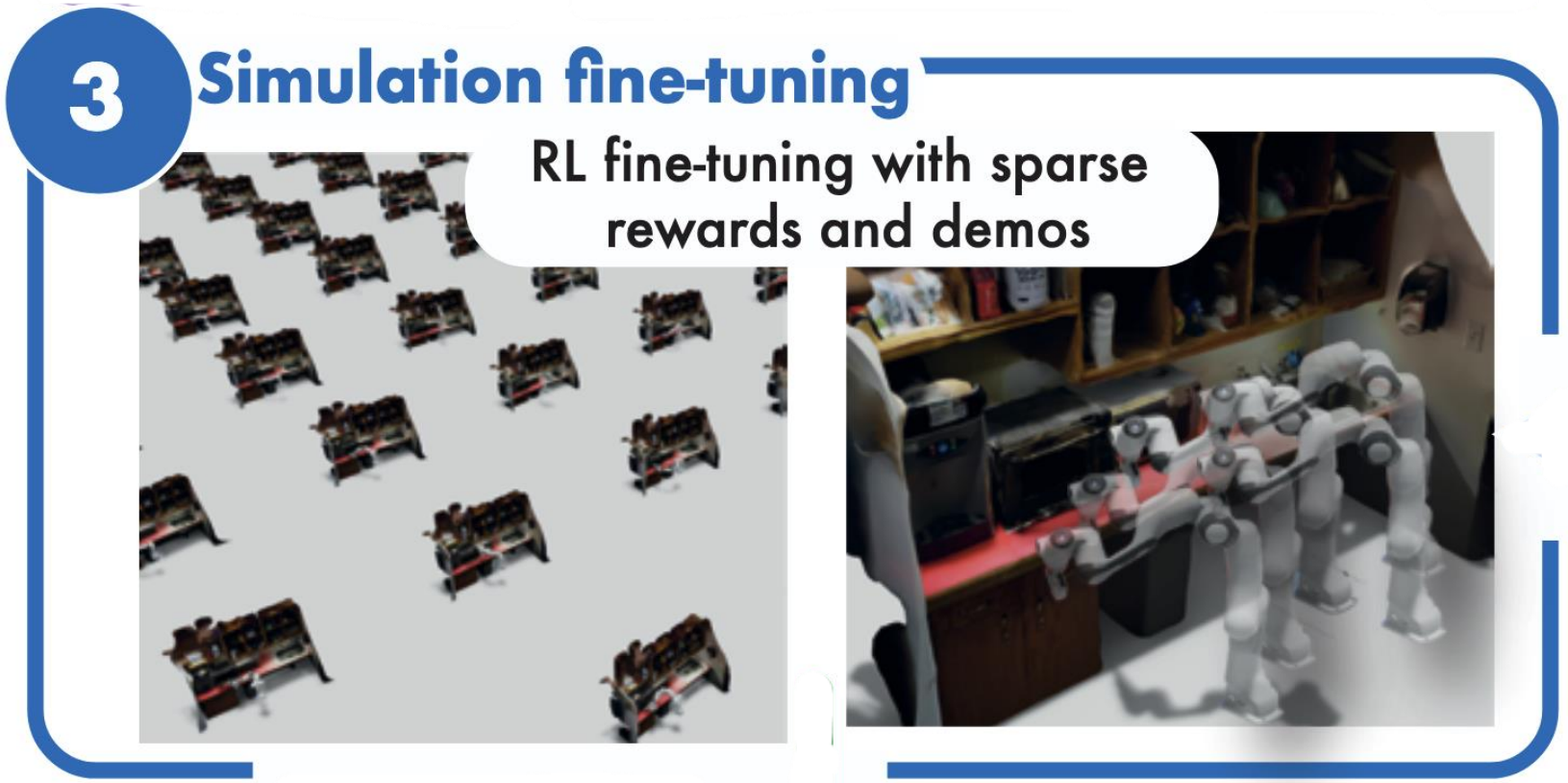
Step 2 (Optional): Imitation Learning



Reconciling Reality Through Simulation: A Real-to-Sim-to-Real Approach for Robust Manipulation. Torne et al.

Step 3: Reinforcement Learning in Digital Twins





Reconciling Reality Through Simulation: A Real-to-Sim-to-Real Approach for Robust Manipulation. Torne et al.

Putting Everything Together

3D scene reconstruction

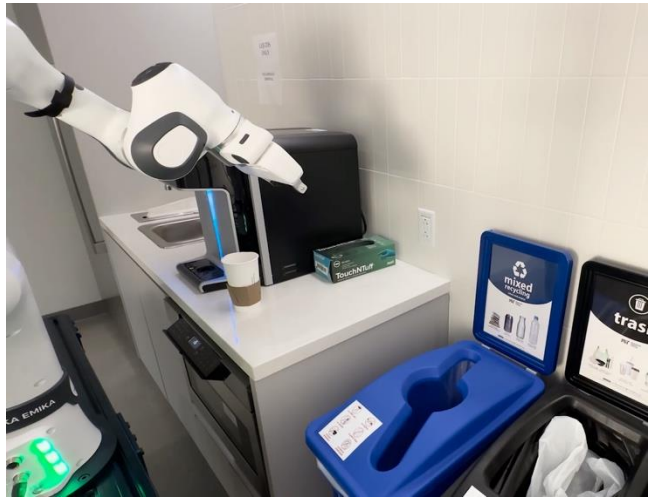
Real-to-sim transfer of policies

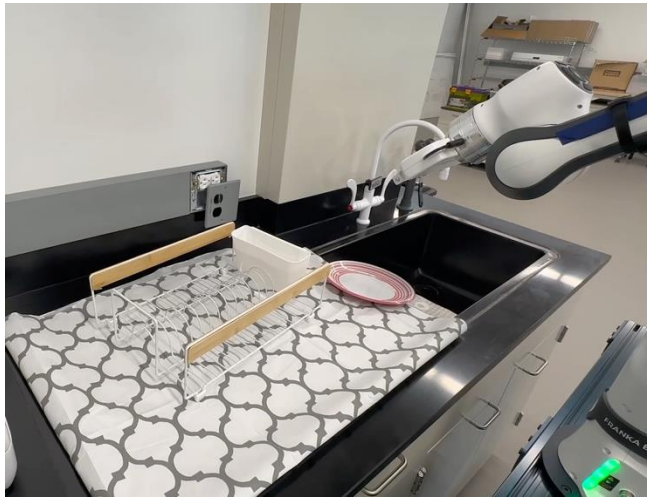
RL fine-tuning in sim

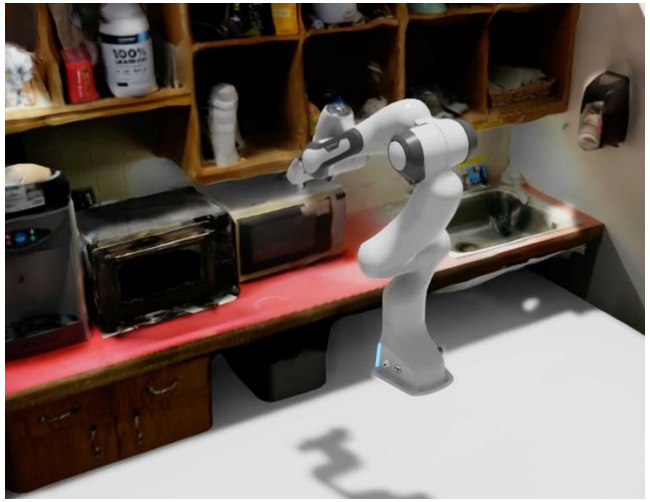
Robust policy in the real world

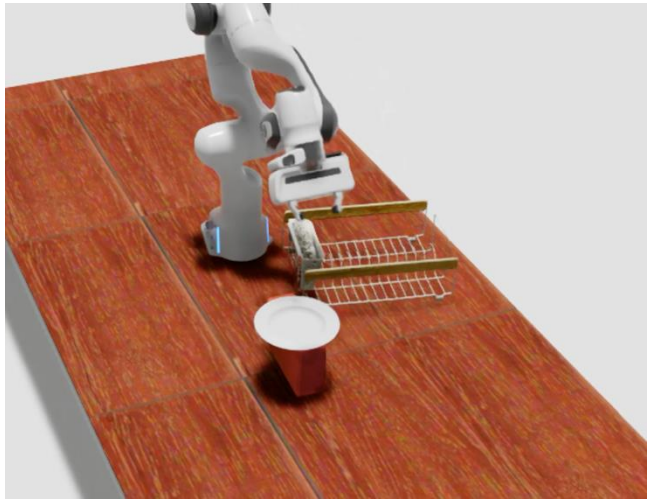
Reconciling Reality Through Simulation: A Real-to-Sim-to-Real Approach for Robust Manipulation. Torne et al.

Results: Real2Sim2Real Adaptation







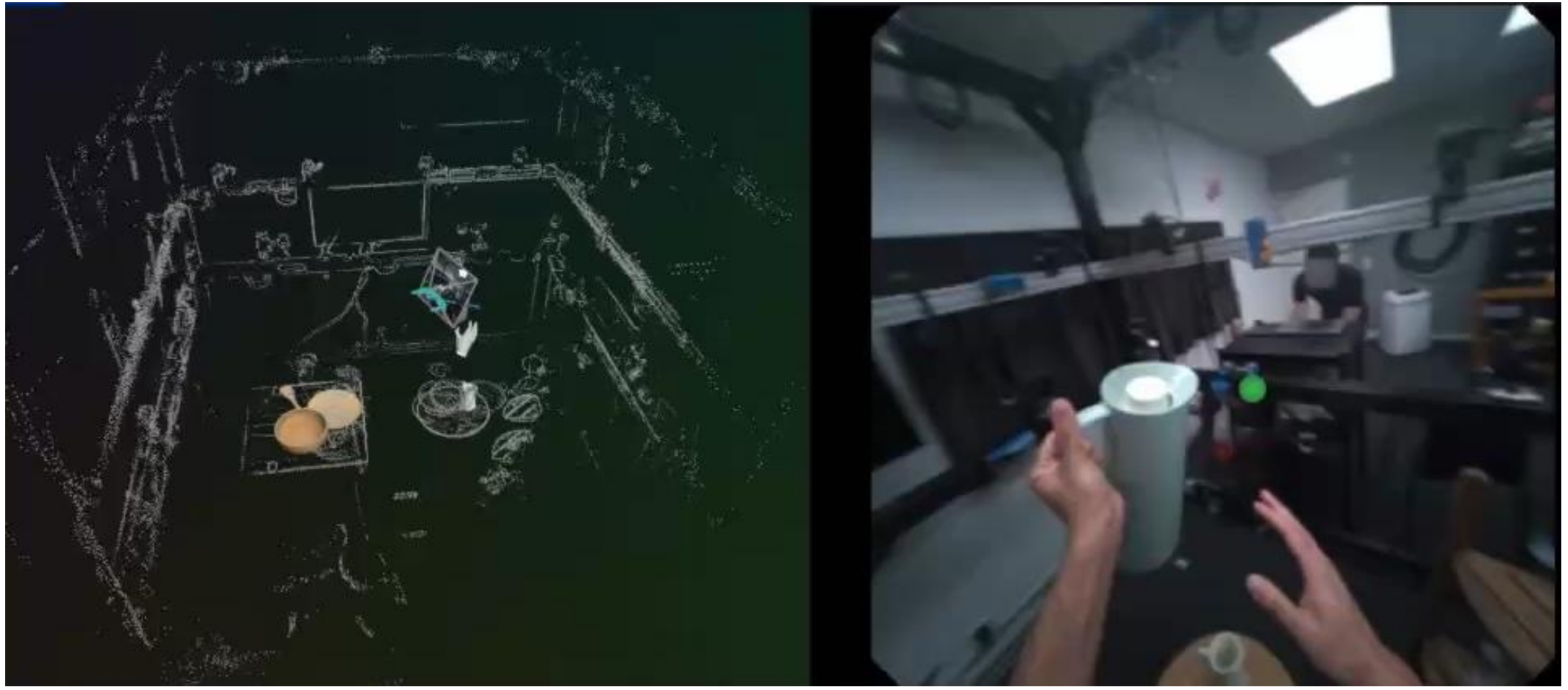


Reconciling Reality Through Simulation: A Real-to-Sim-to-Real Approach for Robust Manipulation. Torne et al.

Remember our goal is to learn from human videos…

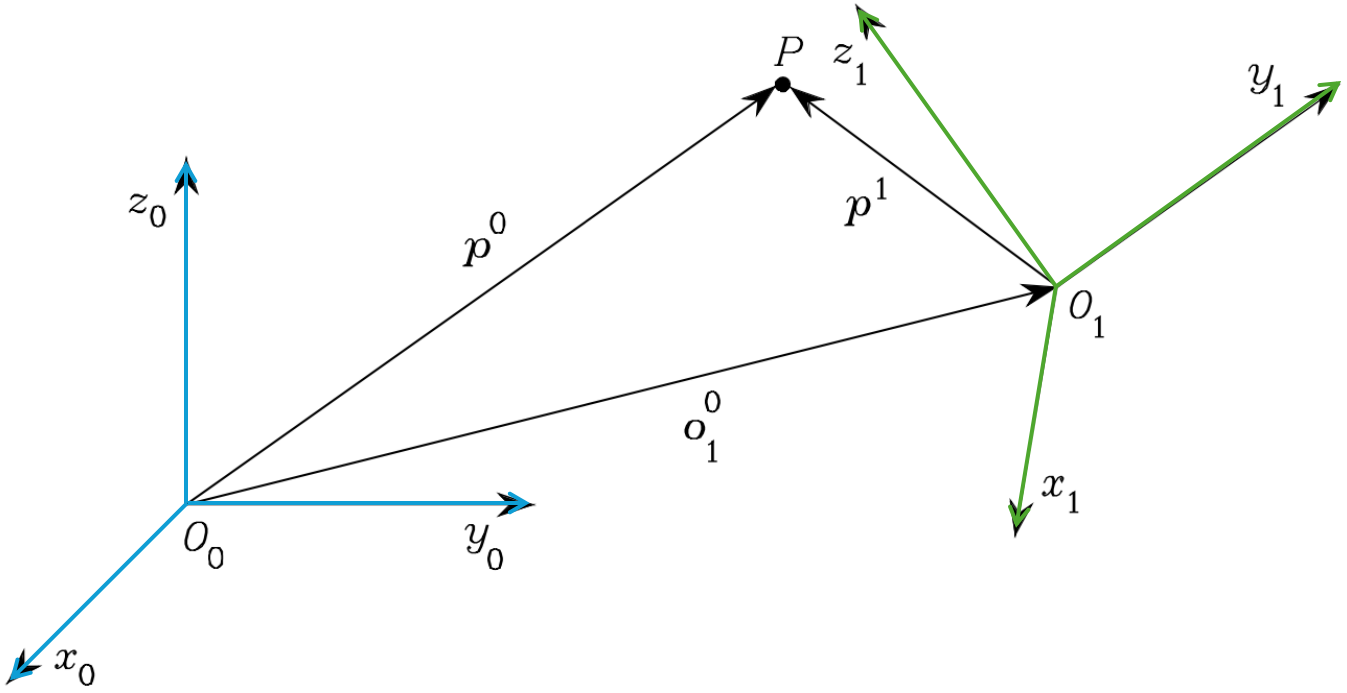
What is the Information Captured by Videos?

Contact points, hand motion and object motion



https://explorer.projectaria.com/hot3d-aria/P0001_95eabc0a?st=%220%22

Describing Rigid-body Object Motion



The coordinate of point P in frame O_0

The coordinate of point P in frame O_1

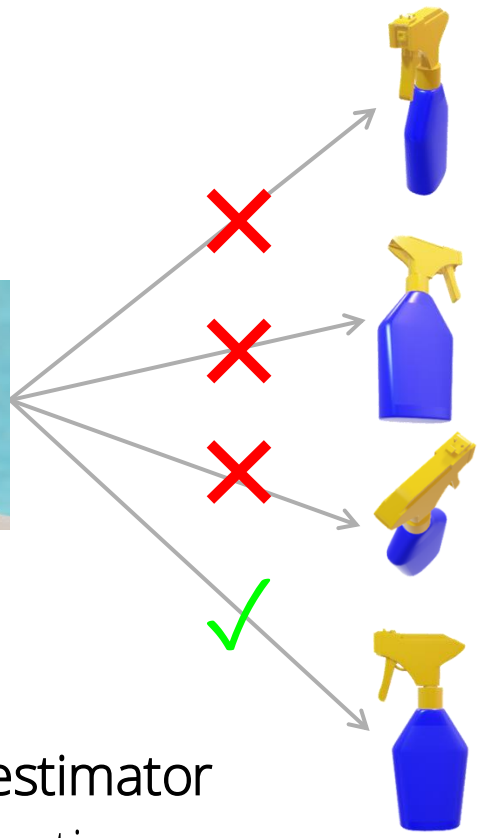
$$\begin{bmatrix} \boldsymbol{p}^0 \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_1^0 & \boldsymbol{o}_1^0 \\ \boldsymbol{0}^T & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{p}^1 \\ 1 \end{bmatrix}$$

Spatial Transformation between frame O_1 and O_0

Rigid-body Object Pose Estimation

Pose Proposals

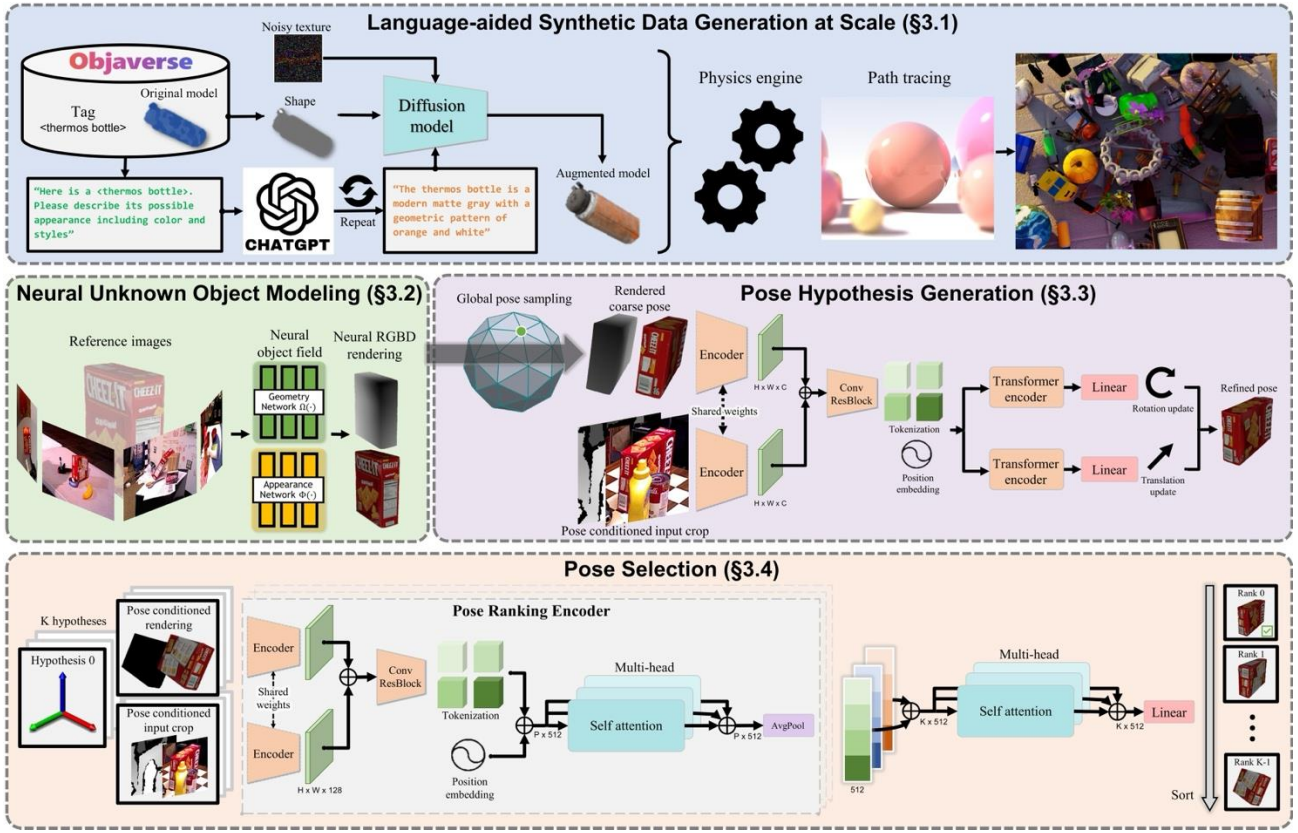
Conditioned Image



Analysis-by-synthesis estimator

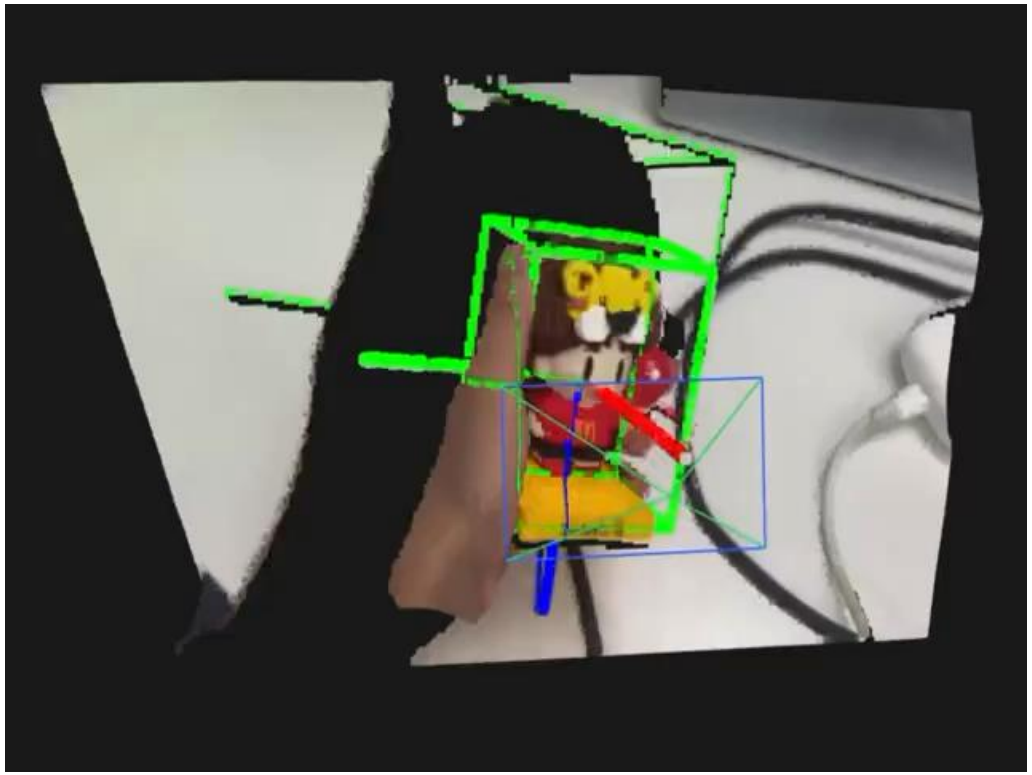
Pros: high-quality estimation

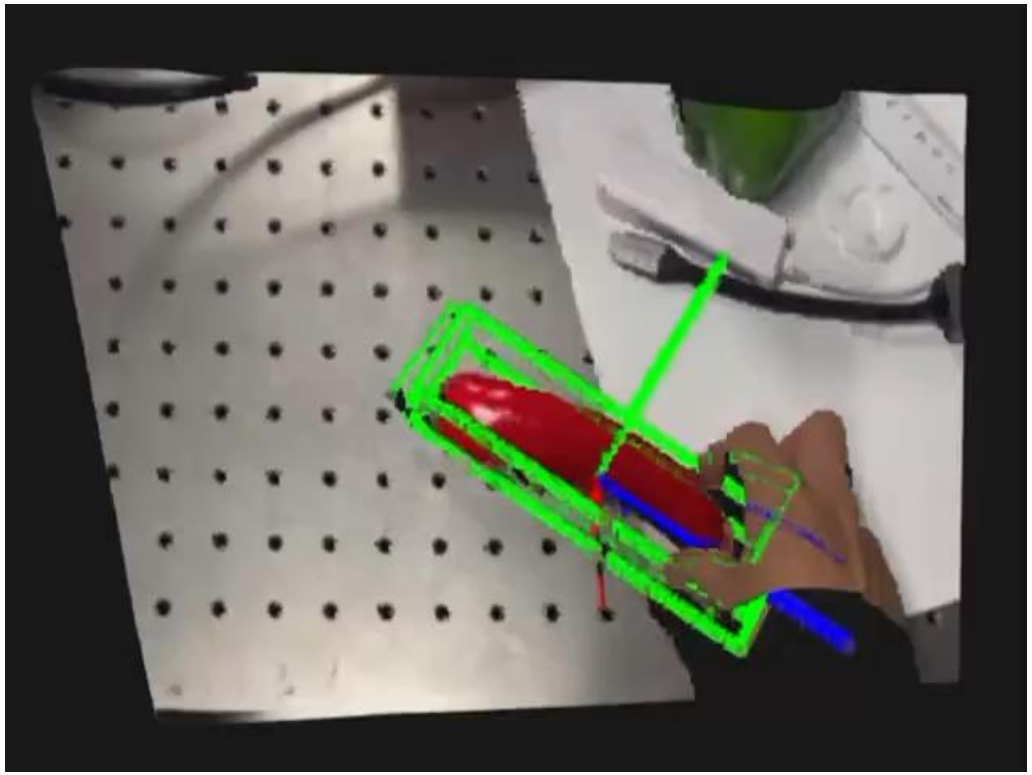
Cons: no temporal consistency



FoundationPose: Unified 6D Pose Estimation and Tracking of Novel Objects. Wen et al

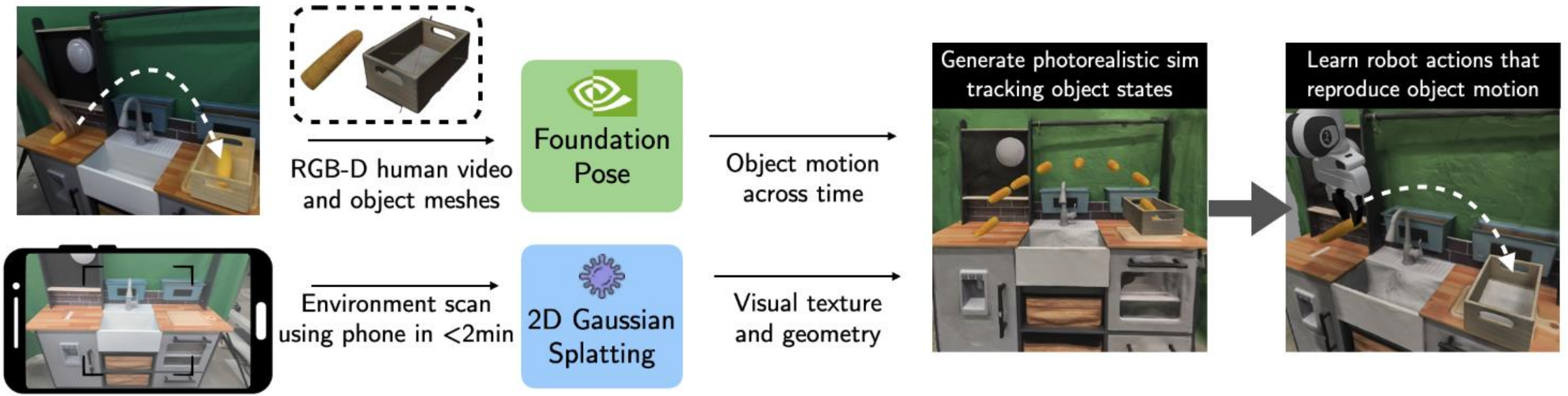
Rigid-body Object Motion Estimation





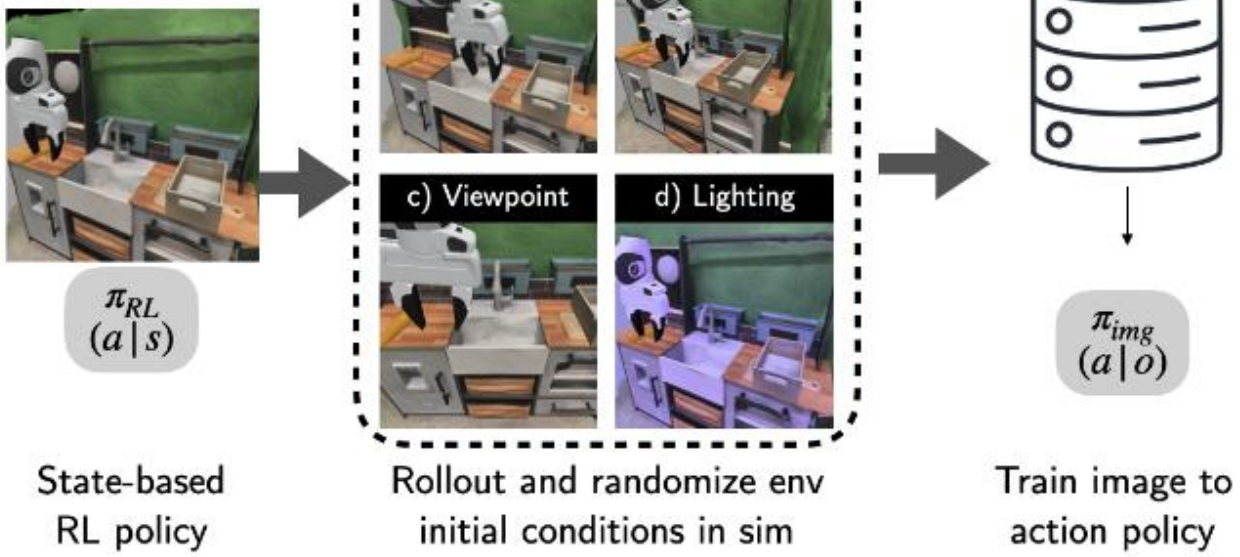
https://explorer.projectaria.com/hot3d-aria/P0001_95eabc0a?st=%220%22

Pose Estimation Needs High-quality Object 3D Model

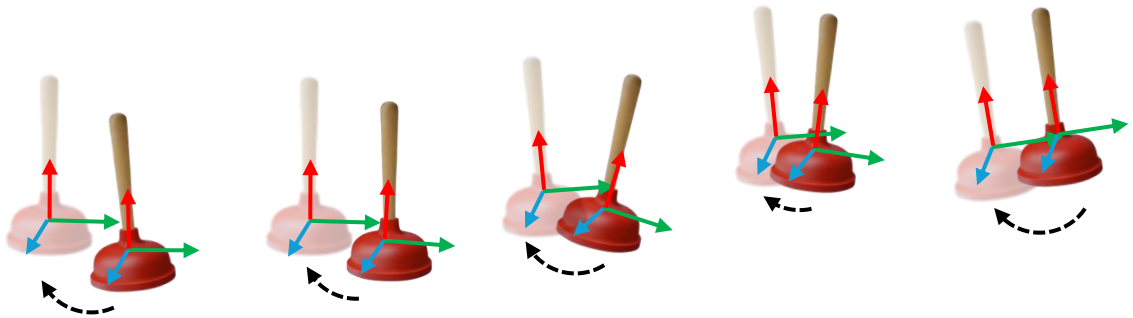


https://explorer.projectaria.com/hot3d-aria/P0001_95eabc0a?st=%220%22

RL in Sim



Object-following reward

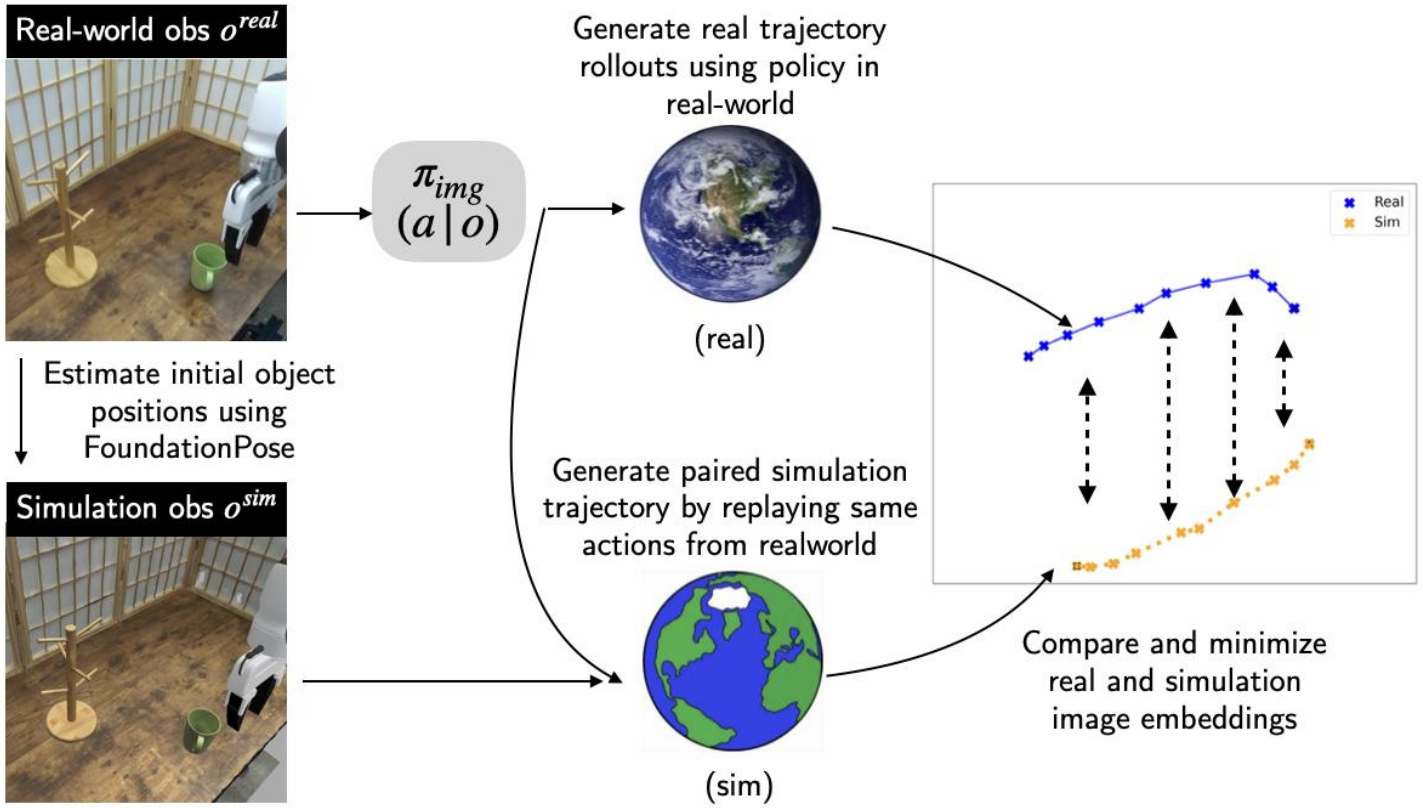


RL in Sim

X-SIM: Cross-Embodiment Learning via Real-to-Sim-to-Real. Dan et al.

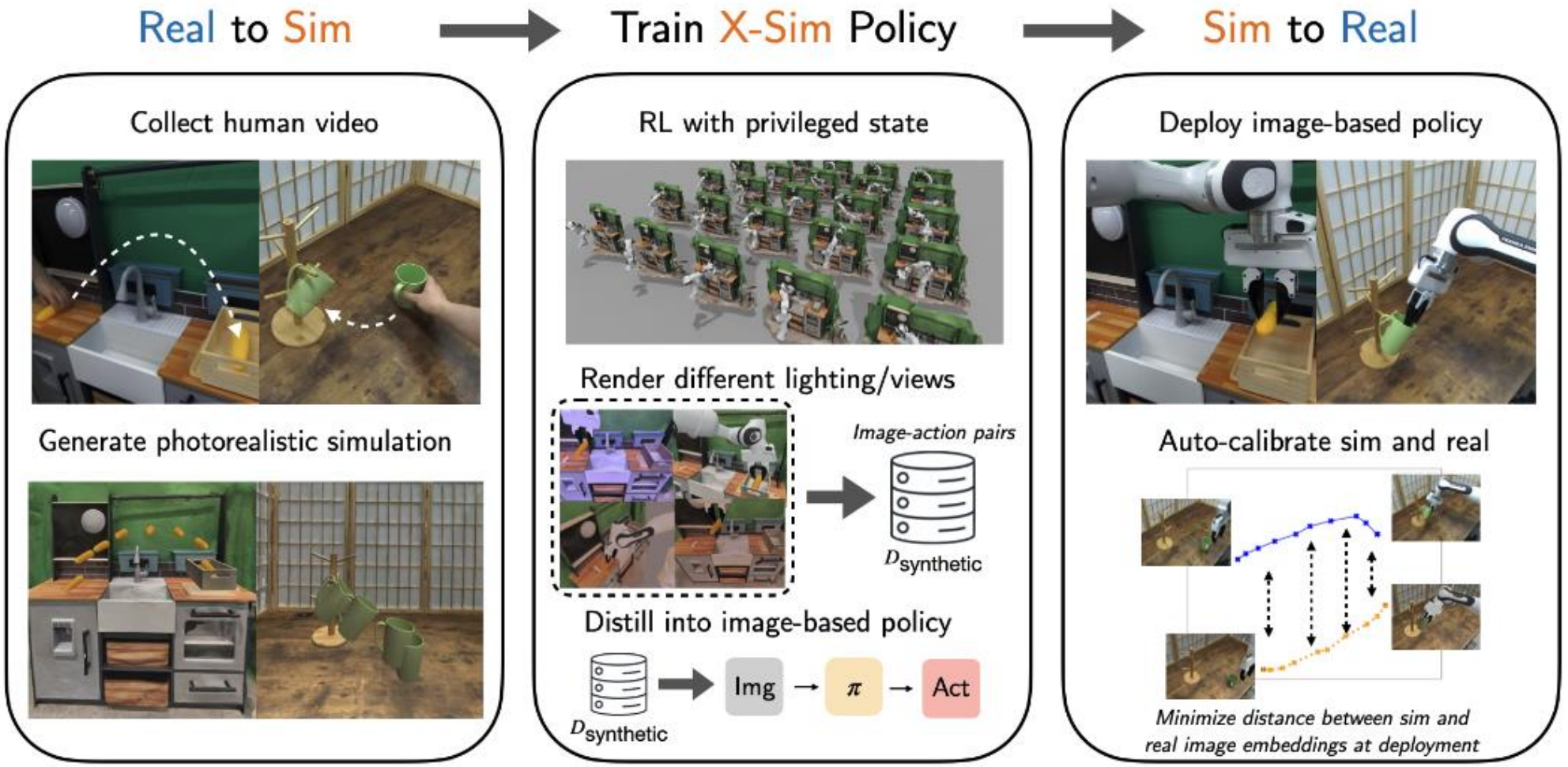
The Policy Learned in Sim Fails in the Real World due to Sim2Real Gap…

Idea: reduce the gap by aligning visual features across sim and real



X-SIM: Cross-Embodiment Learning via Real-to-Sim-to-Real. Dan et al.

Putting Everything Together



X-SIM: Cross-Embodiment Learning via Real-to-Sim-to-Real. Dan et al.

Results: Real2Sim2Real Robot Action Acquisition

Does Real2Sim2Real Pipeline Solve Robotics?

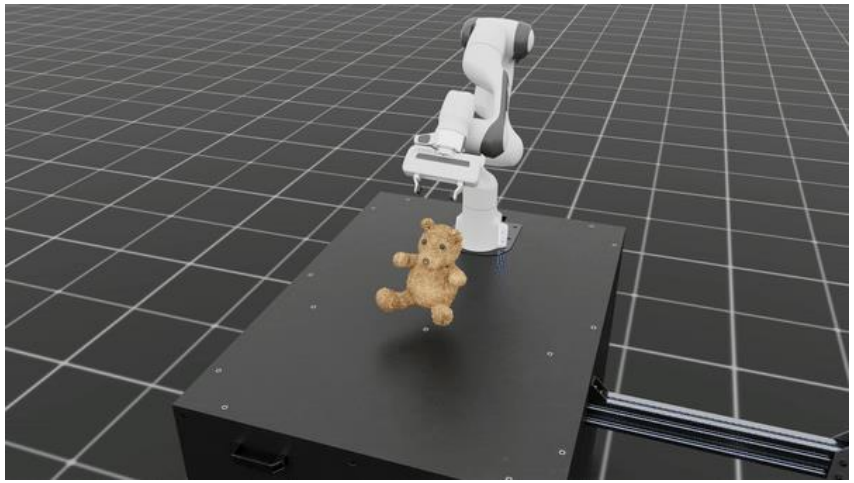
- Of course, NO!:
  1. 3D object scan is required. We can't use in-the-wild videos.
  2. Object pose estimation is still an unsolved vision tasks.
  3. Existing methods only tackle single-arm gripper-based manipulation. What about multi-arm, multi-finger manipulation?
  4. Sim2Real gap is more than visual difference. Physical gap is the hard problem...
  5. Static-camera videos are rare...
  6. What about articulated objects or deformable objects?

Does Real2Sim2Real Pipeline Solve Robotics?

- Of course, NO!:
  1. 3D object scan is required. We can't use in-the-wild videos.
  2. Object pose estimation is still an unsolved vision tasks.
  3. Existing methods only tackle single-arm gripper-based manipulation. What about multi-arm, multi-finger manipulation?
  4. Sim2Real gap is more than visual difference. Physical gap is the hard problem…
  5. Static-camera videos are rare…
  6. What about articulated objects or deformable objects?

We'll talk about this problem

Idea: Policy Learning in Simulator



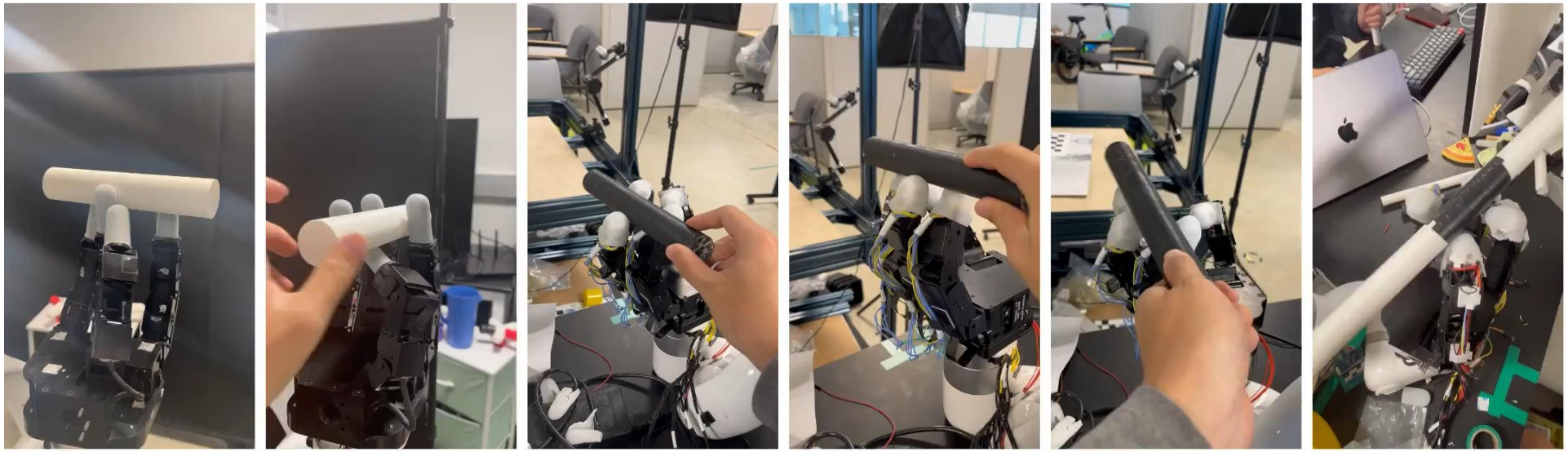
Issac Sim

- Pros:
  - Flexibility: Free to reset to any state
  - Safety: Dangers are not real
  - Low cost: robots and objects do not break
  - Generalizability: Easy to switch the robots
- Cons:
  - Under modeling: physics, geometry and textures are not realistic
  - Complexity: You need to engineers to set up the simulator well

Sim2Real Gap….

Learn in Sim

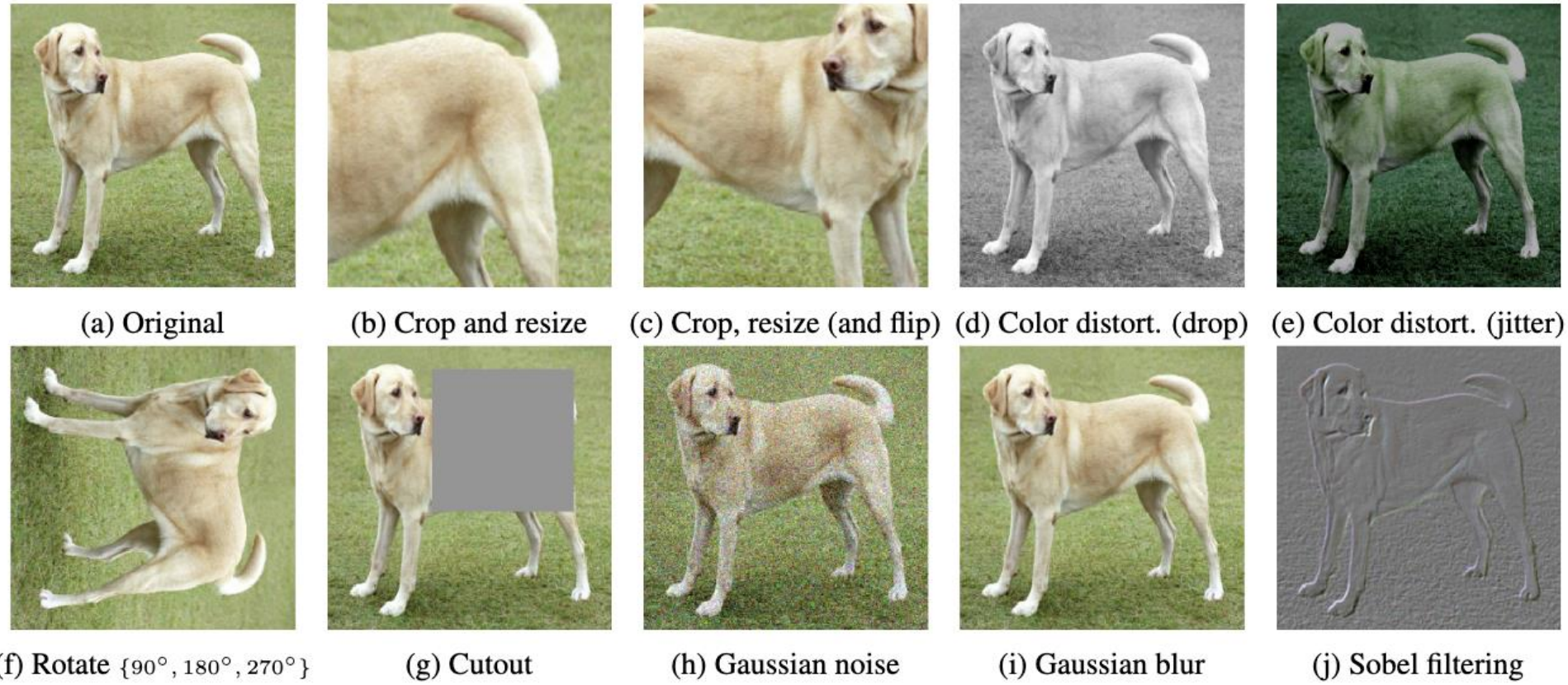
Deploy in Real



Lessons from Learning to Spin “Pens”. Wang et al.

What Cause Sim2Real Gap?

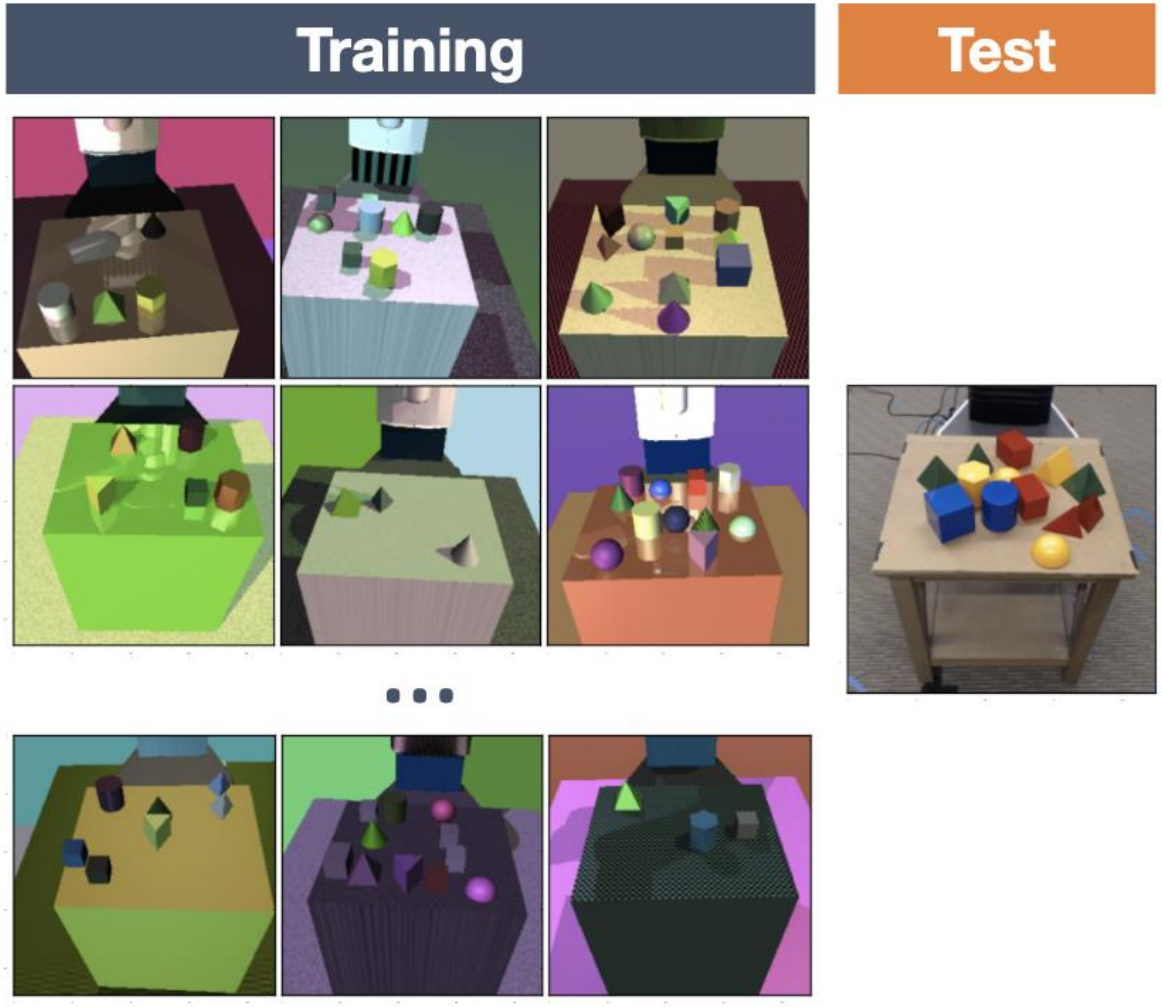
Intuition: Data Augmentation in Computer Vision Works Well..



(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)

(f) Rotate $\{90°, 180°, 270°\}$ (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering

Idea: Data Augmentation in Simulator to Reduce Sim2Real Gap



- Randomized domain: anything you believe that matters:
  1. Number and shape of distractor objects on the table
  2. Position and texture of all objects on the table
  3. Textures of the table, floor, skybox, and robot
  4. Position, orientation, and field of view of the camera
  5. Number of lights in the scene
  6. Position, orientation, and specular characteristics of the lights
  7. Type and amount of random noise added to images

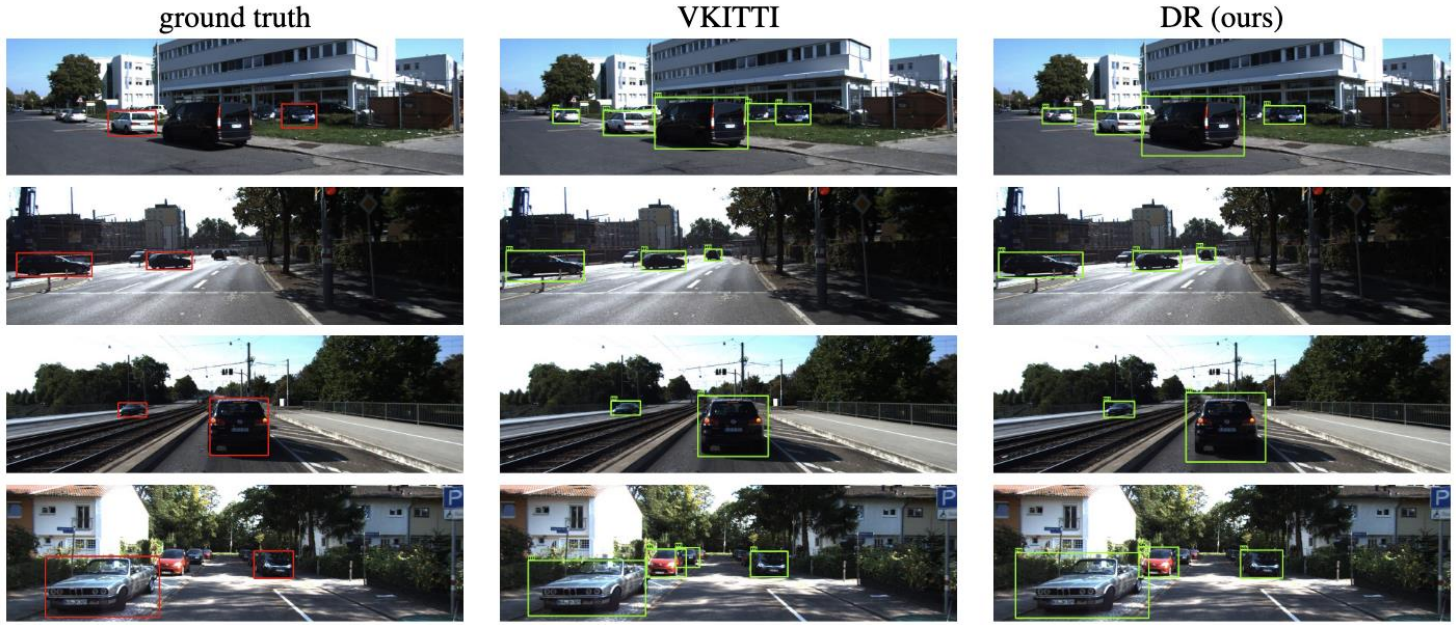
Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. Tobin et al.

An Example of Car Detection

Train on Virtual KITTI

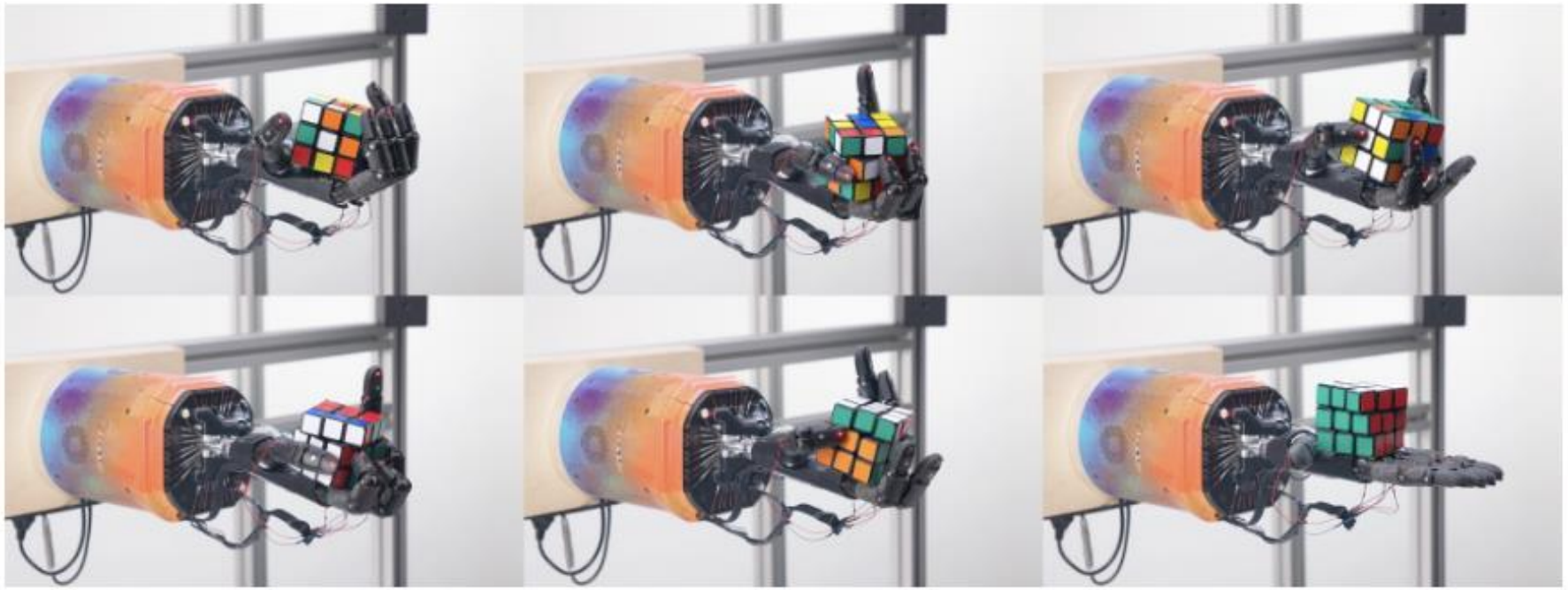
Train on Domain Randomization

Test on Real KITTI



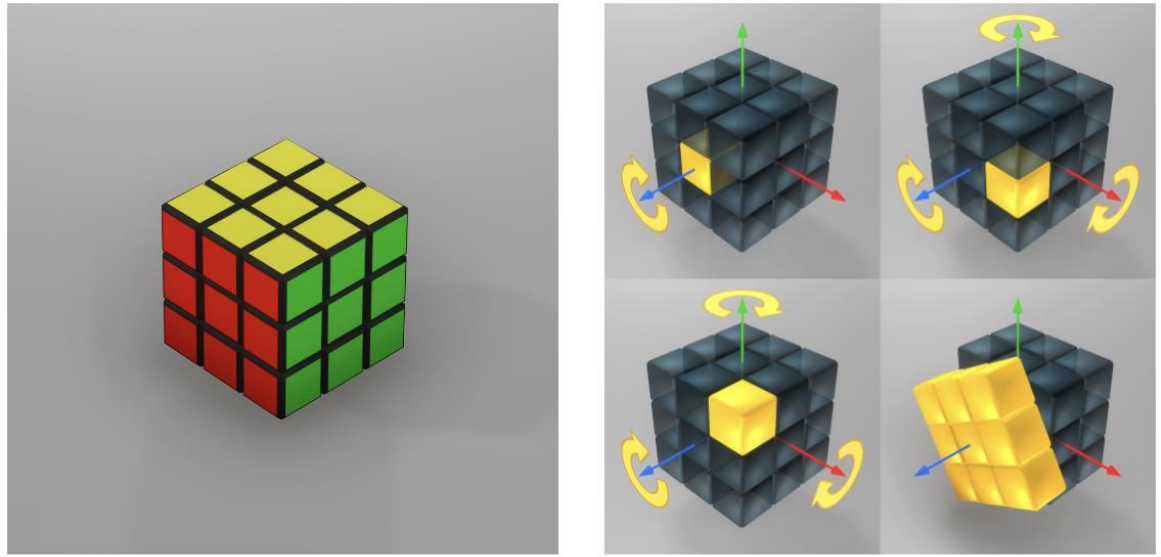
Training Deep Networks with Synthetic Data:
Bridging the Reality Gap by Domain Randomization. Tremblay et al.

An Example of Rubik‘s Cube



Solving Rubik’s Cube with a Robot Hand. OpenAI.

Building the Simulator



(a) Rendering of the cube model.

(b) Rendering of the different axis.

Figure 9: Our MuJoCo model of the Rubik's cube. On the left, we show a rendered version. On the right, we show the individual cublets that make up our model and visualize the different axis and degrees of freedom of our model.

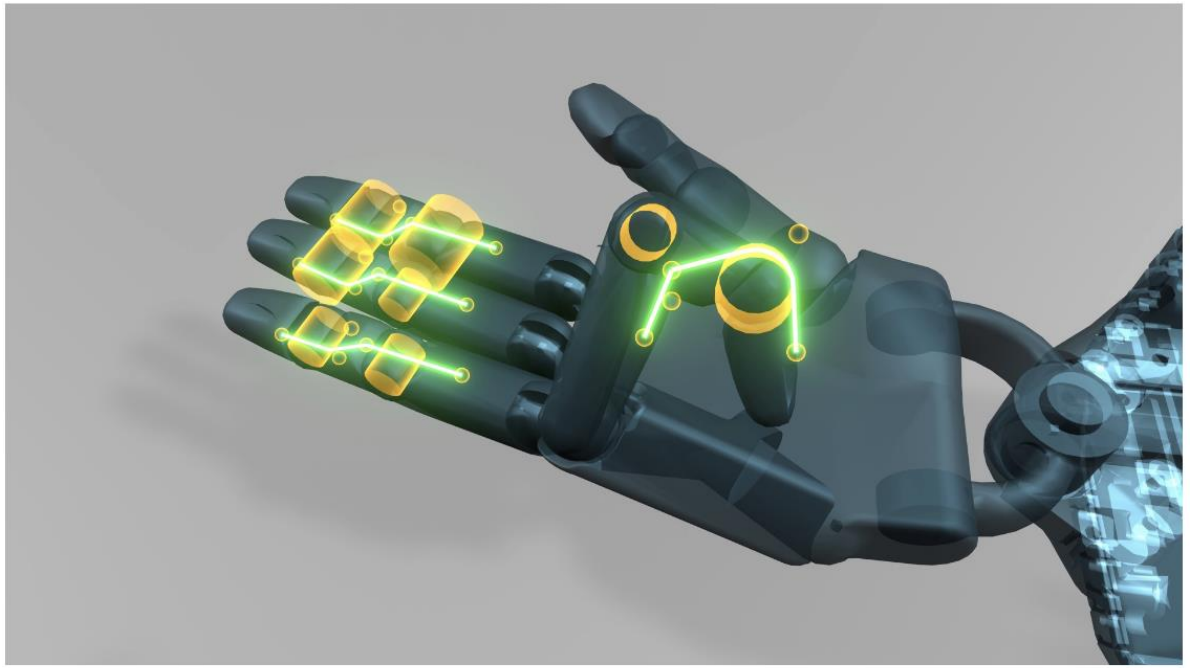
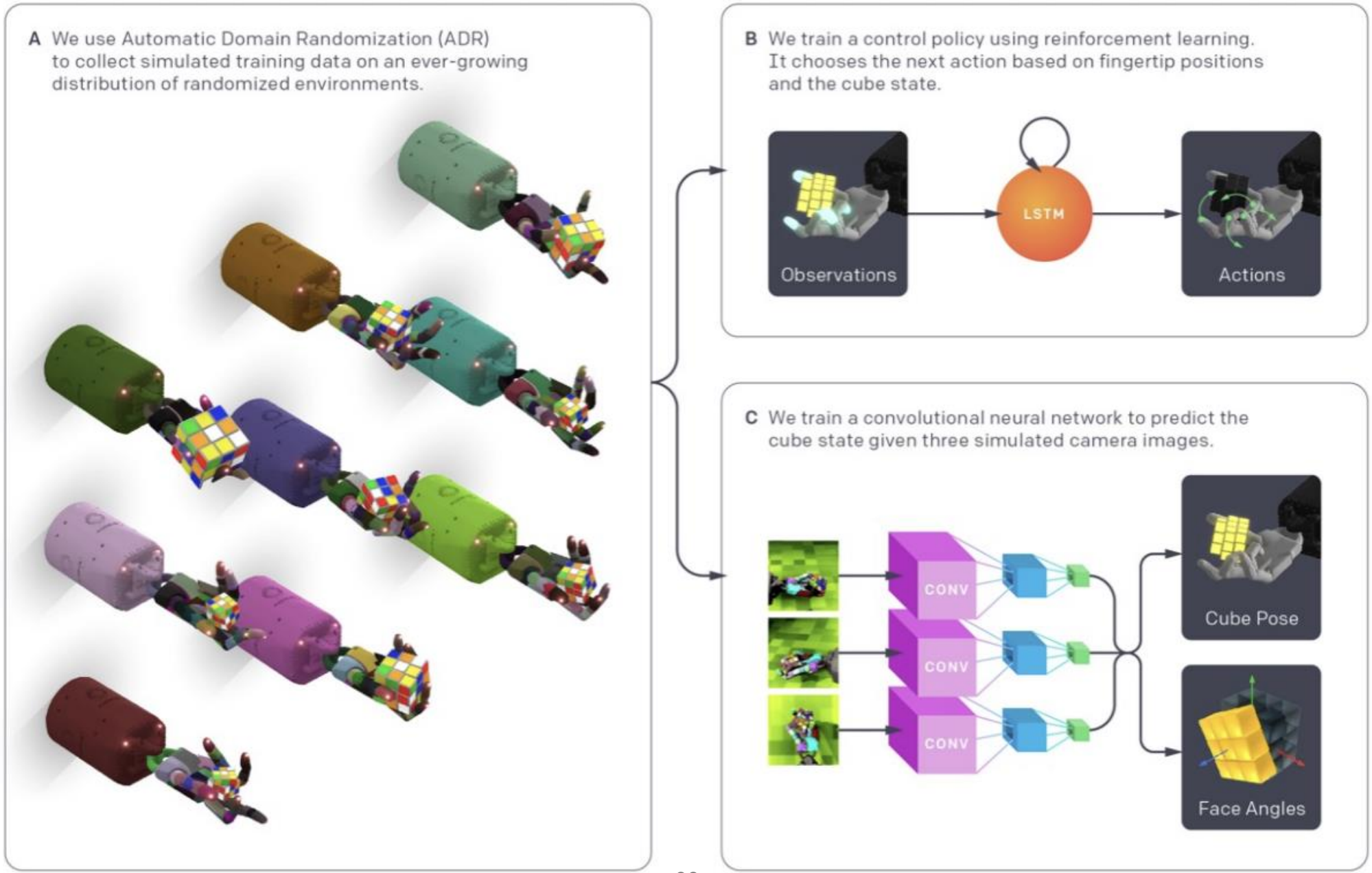


Figure 7: Transparent view of the hand in the new simulation. One spatial tendon (green lines) and two cylindrical geometries acting as pulleys (yellow cylinders) have been added for each non-thumb finger in order to achieve coupled joints dynamics similar to the physical robot.

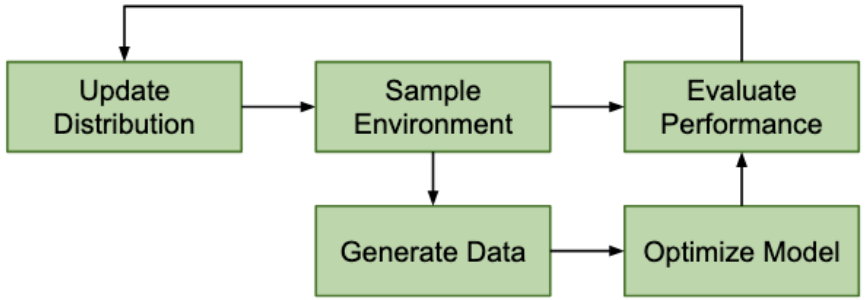
Train in Simulation



Solving Rubik's Cube with a Robot Hand. OpenAI.

Domain Randomization

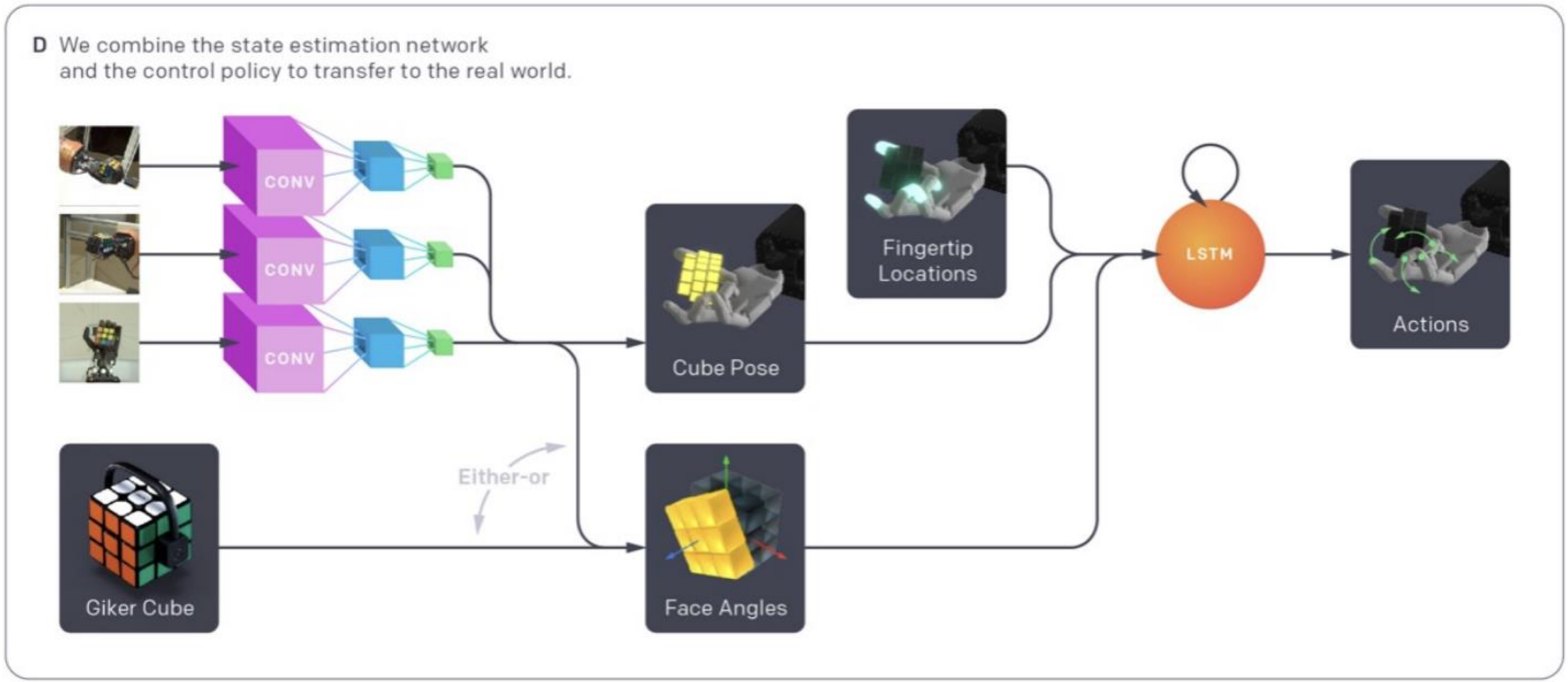
Adaptively select randomization parameters

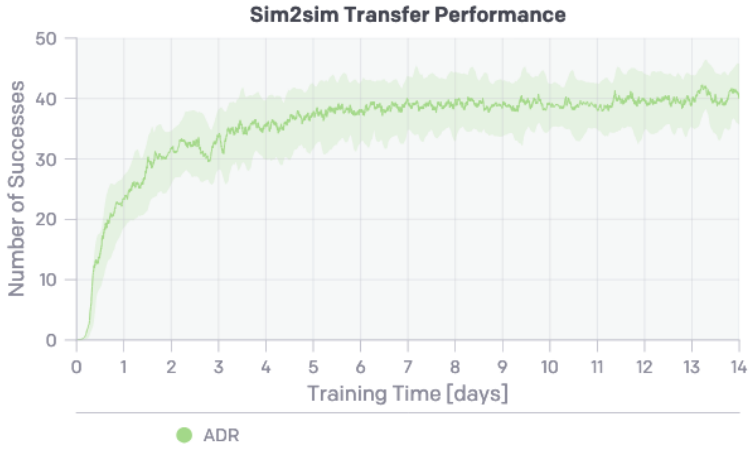


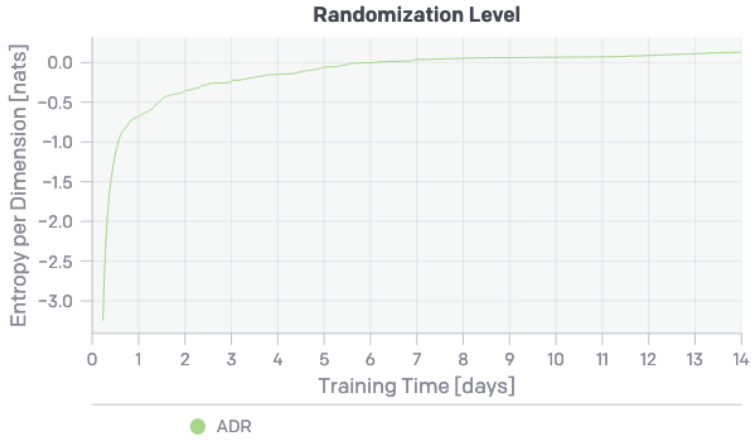
Category	All policies	Reorientation policy	Rubik's cube policy
Simulator physics (generic)	Actuator force range Actuator gain prm Body inertia Geom size robot spatial Tendon length spring (M, 0.75) Tendon stiffness (M, 0.75)	Dof armature Dof damping Dof friction loss Geom friction Geom gap (M, 0.03)	Body position (AG, 0.02) Dof armature cube Dof armature robot Dof damping cube Dof damping robot Dof friction loss cube Dof friction loss robot Geom gap cube (AU, 0.01) Geom gap robot (AU, 0.01) Geom pos cube (AG, 0.002) Geom pos robot (AG, 0.002) Geom margin cube (AG, 0.0005) Geom margin robot (AG, 0.0005) Geom solimp (M, 1.0) Geom solref (M, 1.0) Joint stiffness robot (UAG, 0.005)
Simulator physics (custom)	Body mass Cube size Tendon range		Friction robot Friction cube
Custom physics	Action latency Backlash Joint margin Joint range Time step		Action noise Time step variance
Adversary	Adversary		
Observation			Observation

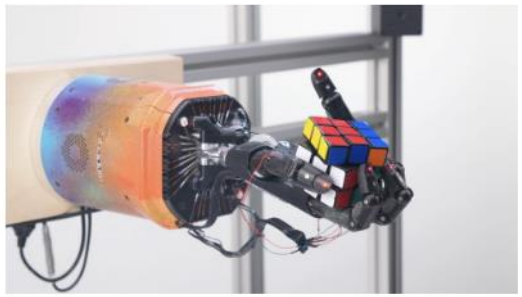
Solving Rubik's Cube with a Robot Hand. OpenAI.

Transfer to the Real World

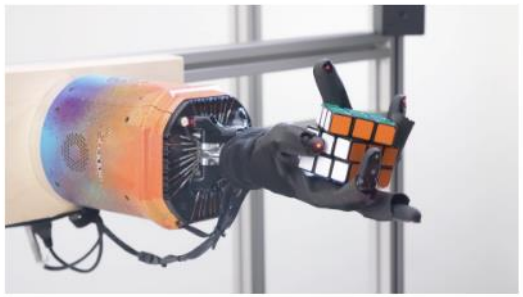




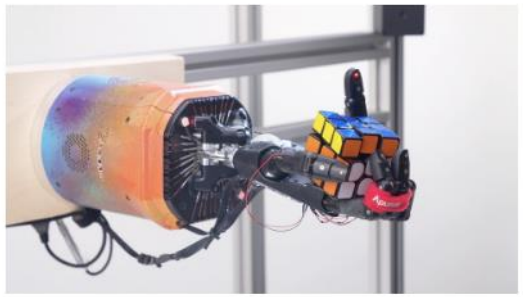




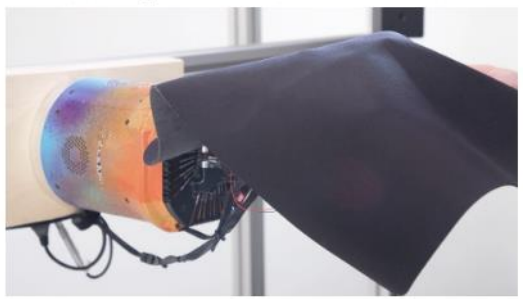
(a) Unperturbed (for reference).



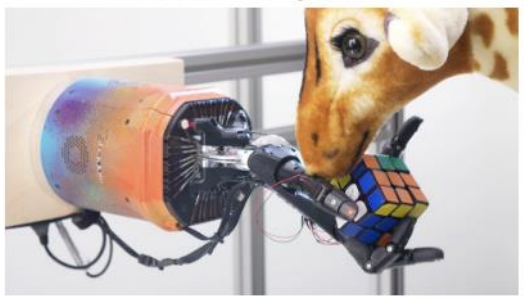
(b) Rubber glove.



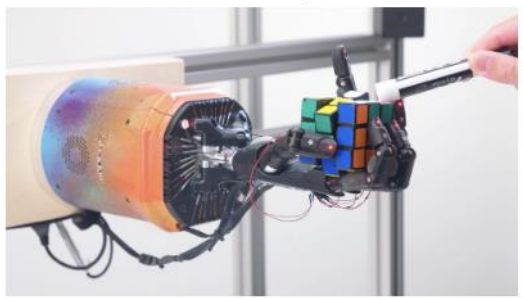
(c) Tied fingers.



(d) Blanket occlusion and perturbation.



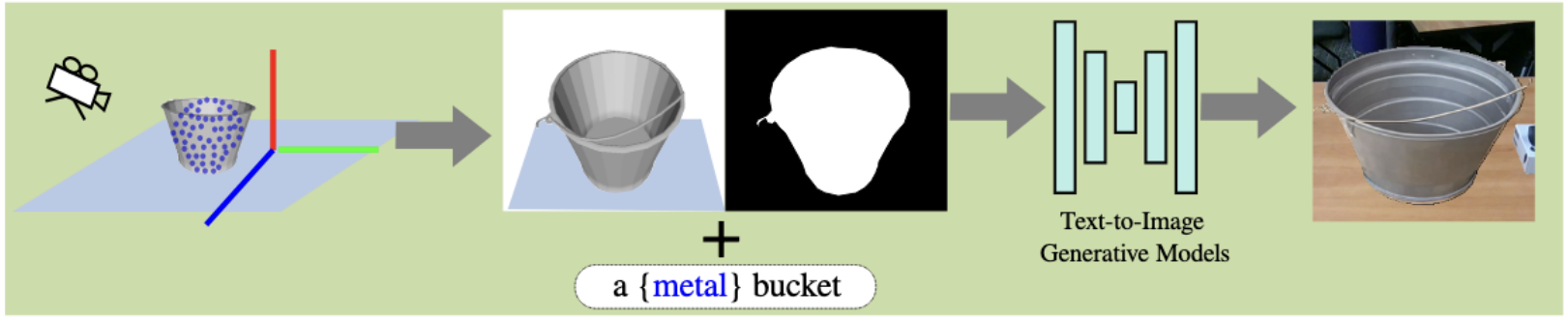
(e) Plush giraffe perturbation.[17]

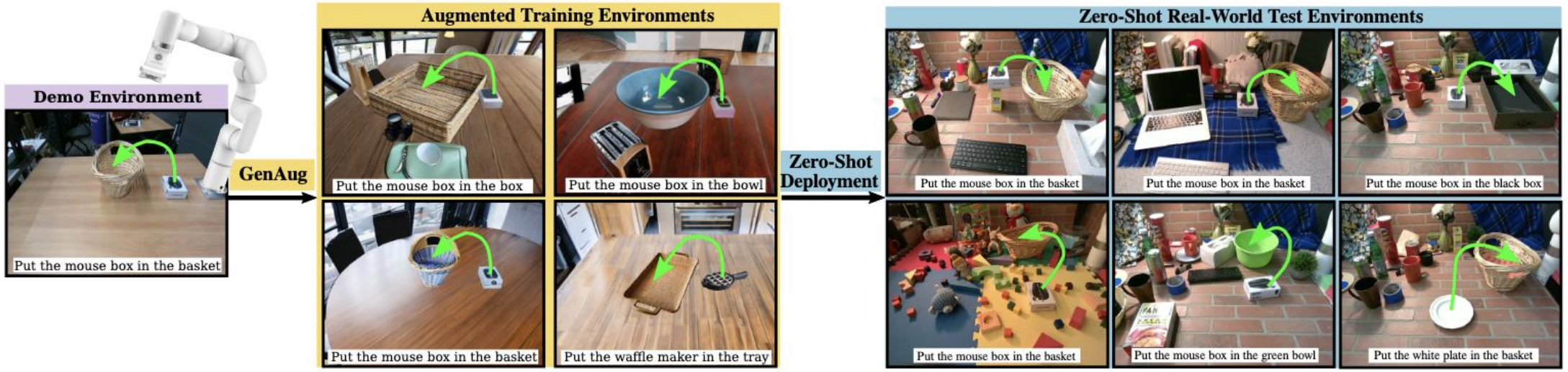


(f) Pen perturbation.

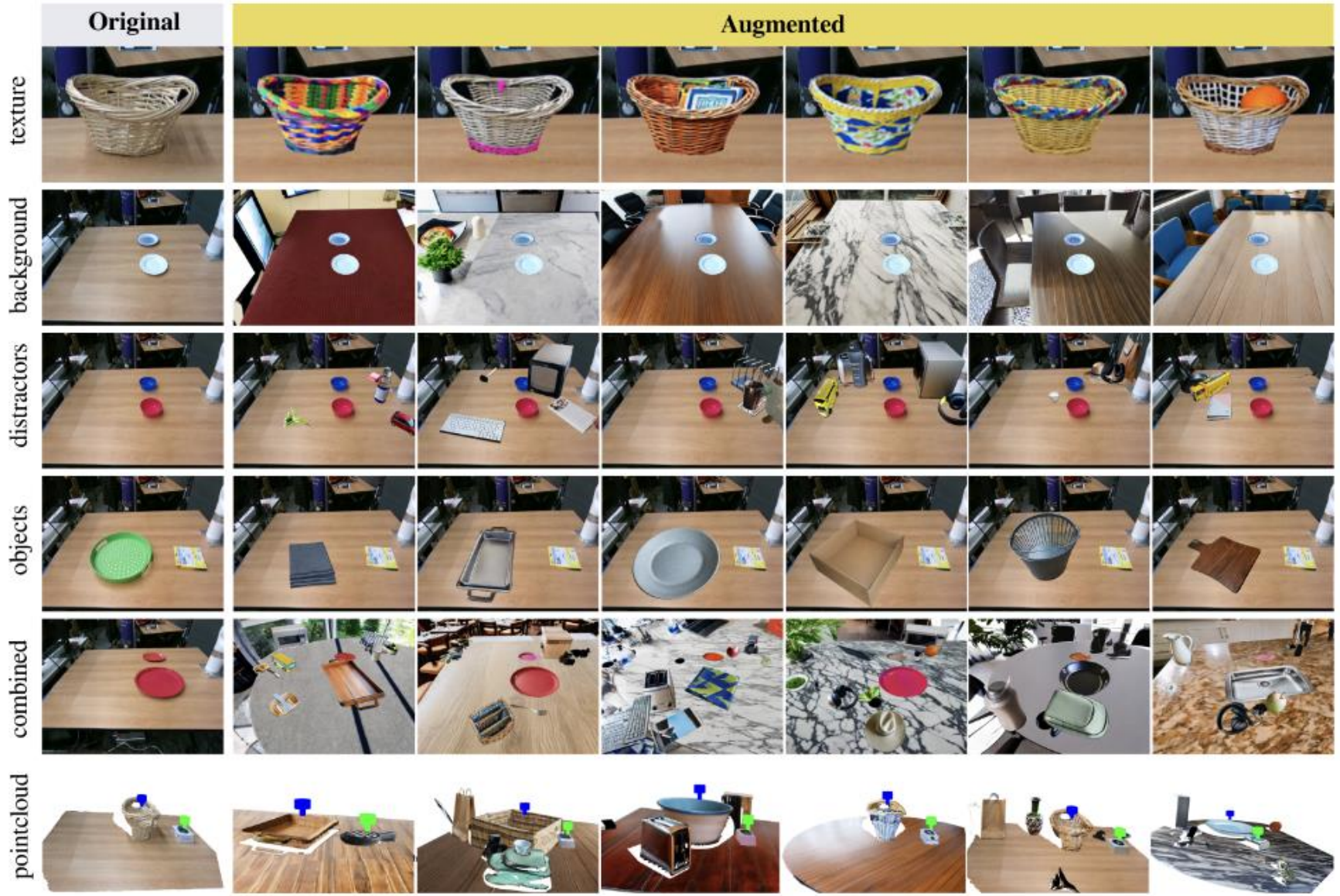
Any Better Domain Randomization?

- We have awesome image generative models, they are ready for visual augmentation



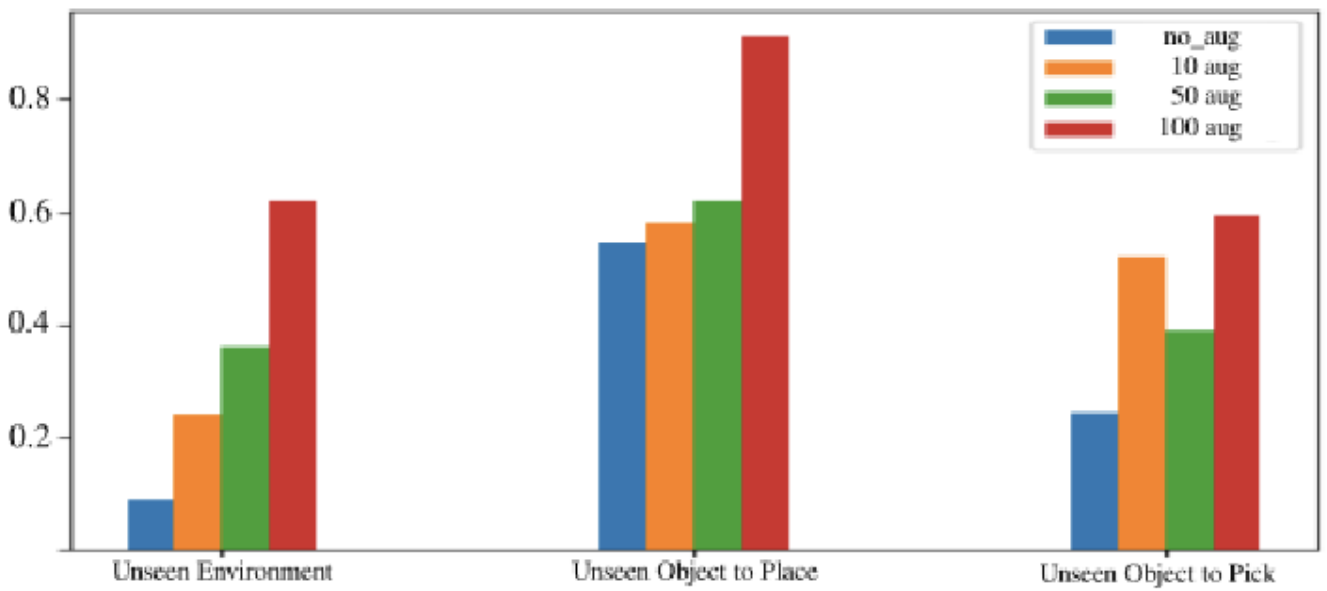


GenAug: Retargeting behaviors to unseen situations via Generative Augmentation. Chen et al.



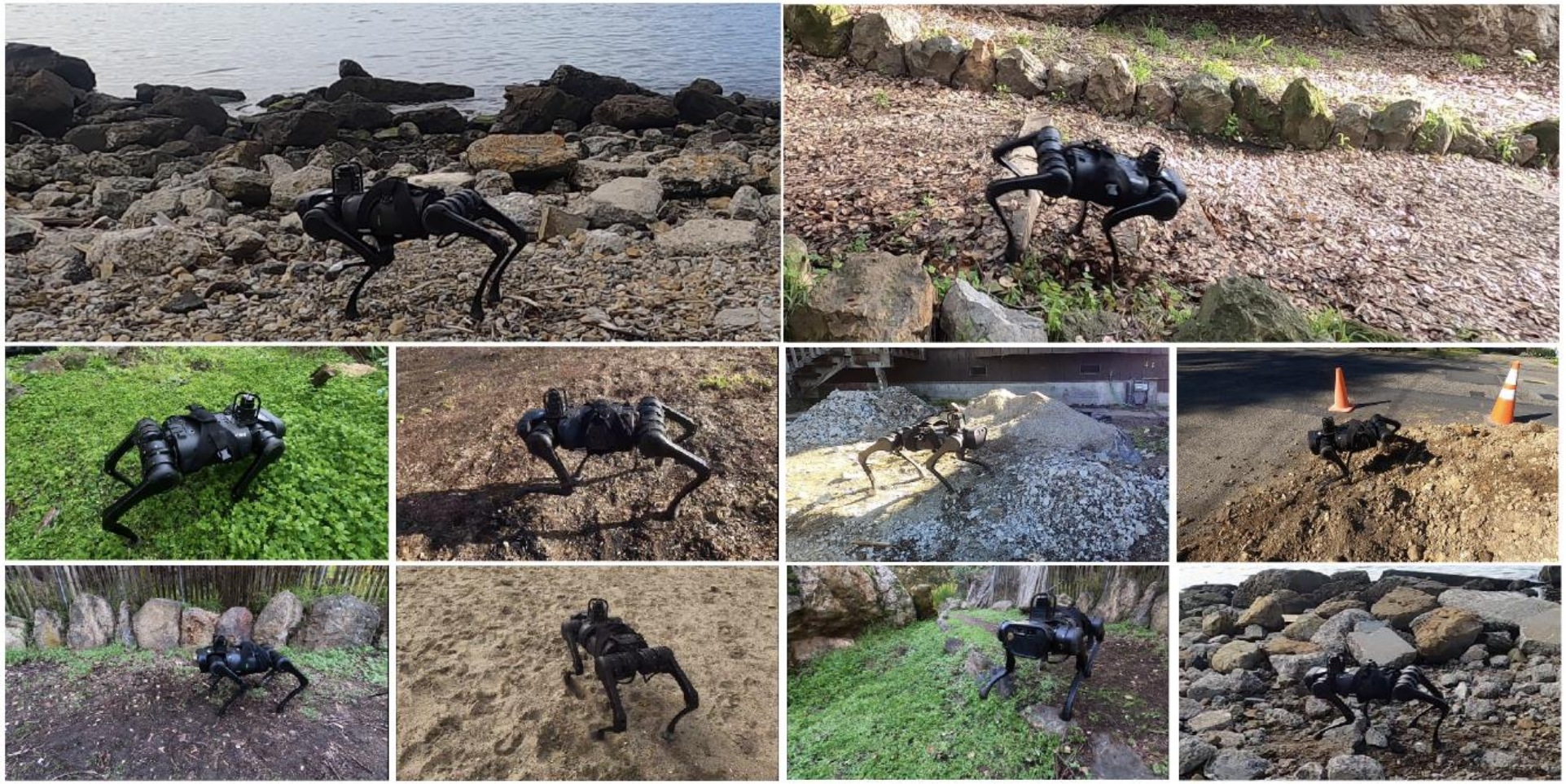
GenAug: Retargeting behaviors to unseen situations via Generative Augmentation. Chen et al.

	Unseen Environment						Unseen to place						Unseen to pick					
	TransporterNet			CLIPort			TransporterNet			CLIPort			TransporterNet			CLIPort		
Method	1	10	100	1	10	100	1	10	100	1	10	100	1	10	100	1	10	100
No Augmentation	4.8	8.1	9.8	11.7	14.3	14.4	15.1	30.4	52.6	39.4	40.8	44.6	8.5	34.6	54.9	46.0	67.0	64.1
Spatial Augmentation	11.0	12.2	8.3	23.3	16.1	26.7	44.3	50.5	65.3	26.1	36.9	50.7	53.6	57.2	66.4	38.2	56.9	80.3
Random Copy Paste	**53.1**	67.0	73.5	38.2	39.8	64.3	55.1	65.4	**84.9**	39.7	55.9	73.9	48.3	67.0	76.1	52.5	65.0	81.0
Random Background	53.0	75.3	79.1	33.6	62.2	55.4	24.5	22.1	35.5	7.6	9.9	17.9	44.4	40.7	35.9	19.2	52.7	72.3
Random Distractors	10.1	9.7	13.7	15.4	36.2	35.8	28.2	60.7	66.0	27.5	51.8	54.3	42.5	47.4	62.3	31.0	64.0	69.1
R3M Finetune	4.1	6.0	4.8	22.2	16.8	20.9	43.5	40.6	41.9	30.9	43.5	57.5	45.6	45.7	41.1	46.7	50.7	72.7
GenAug	43.9	58.5	77.6	46.6	57.0	71.9	**69.1**	**76.5**	83.6	62.6	**83.9**	**86.3**	**75.3**	75.6	**87.2**	**61.5**	**77.7**	**83.1**
GenAug **(w Depth)**	47.8	**83.8**	**91.2**	**47.2**	**60.9**	**73.4**	39.9	67.2	74.2	**64.8**	73.8	84.6	71.2	**83.4**	87.1	56.2	67.3	81.5



What If Testing Scenes Have a High Variety?

- Training locomotion robots in real world is extremely dangerous…



RMA: Rapid Motor Adaptation for Legged Robots. Kumar et al.

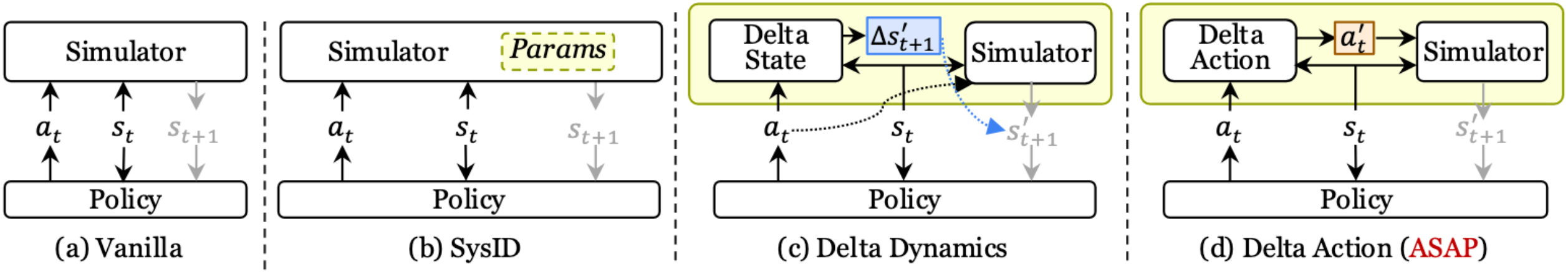

Rocky area next to river bed

RMA: Rapid Motor Adaptation for Legged Robots. Kumar et al.

RMA: Rapid Motor Adaptation for Legged Robots. Kumar et al.

RMA: Rapid Motor Adaptation for Legged Robots. Kumar et al.

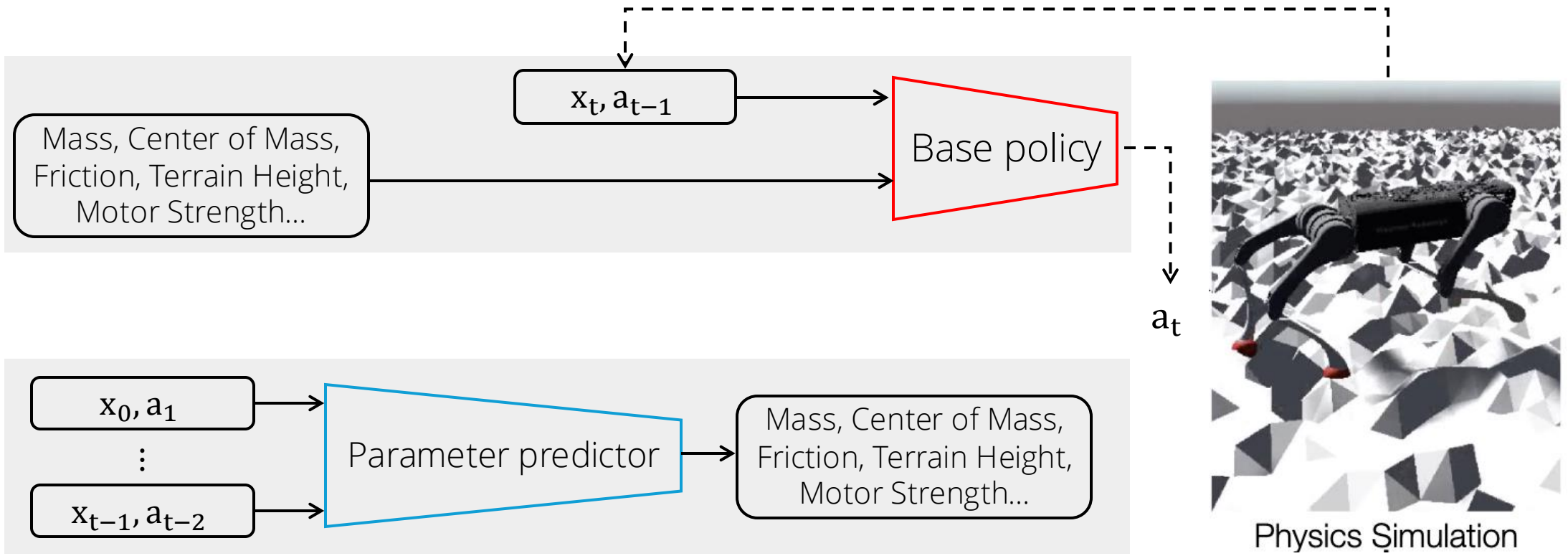
Solutions for Test-time Adaptation



- Ideas:
  1. System Identification: Guess the physical parameters
  2. Delta Dynamics: Learn a dynamic simulator which induces robot policies (e.g. Model Predictive Control). Adapt the dynamic simulator to each scene.
  3. Delta Action: Adapt the physics in simulation to those in the real world. Learn the policy based in the real-world-like physics in sim.

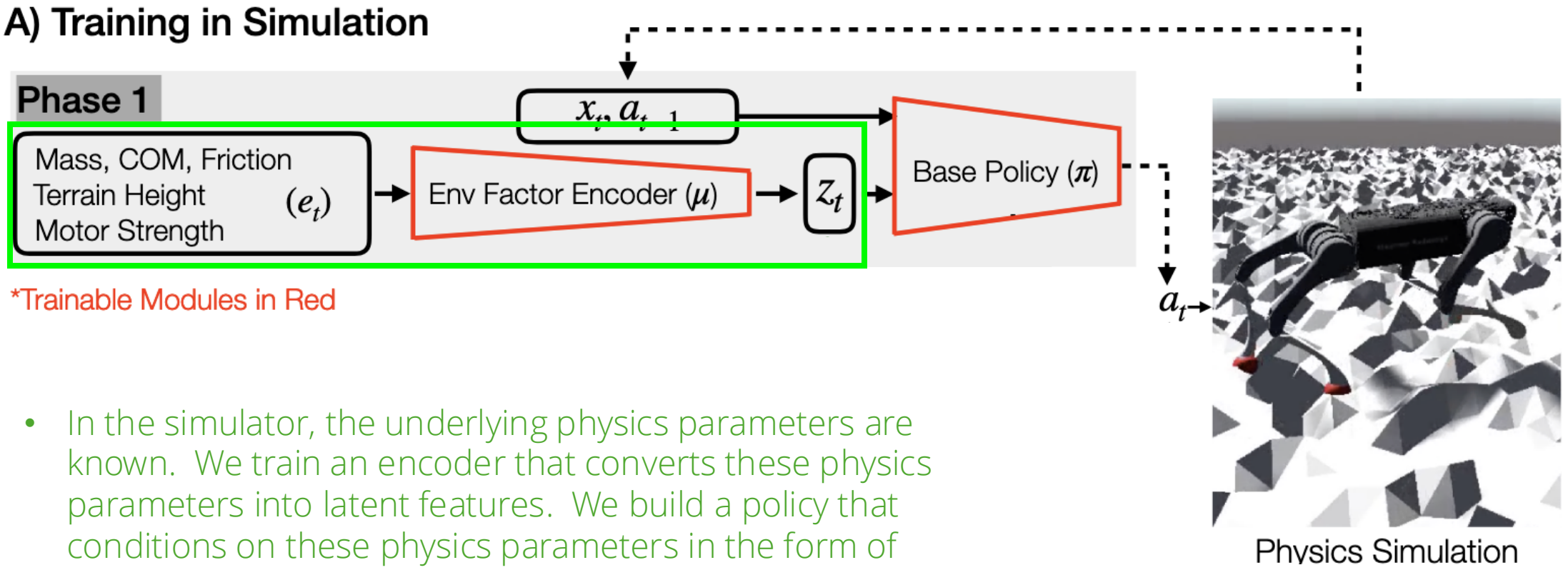
ASAP: Aligning Simulation and Real-World Physics for Learning Agile Humanoid Whole-Body Skills. He et al.

Explicit System Identification



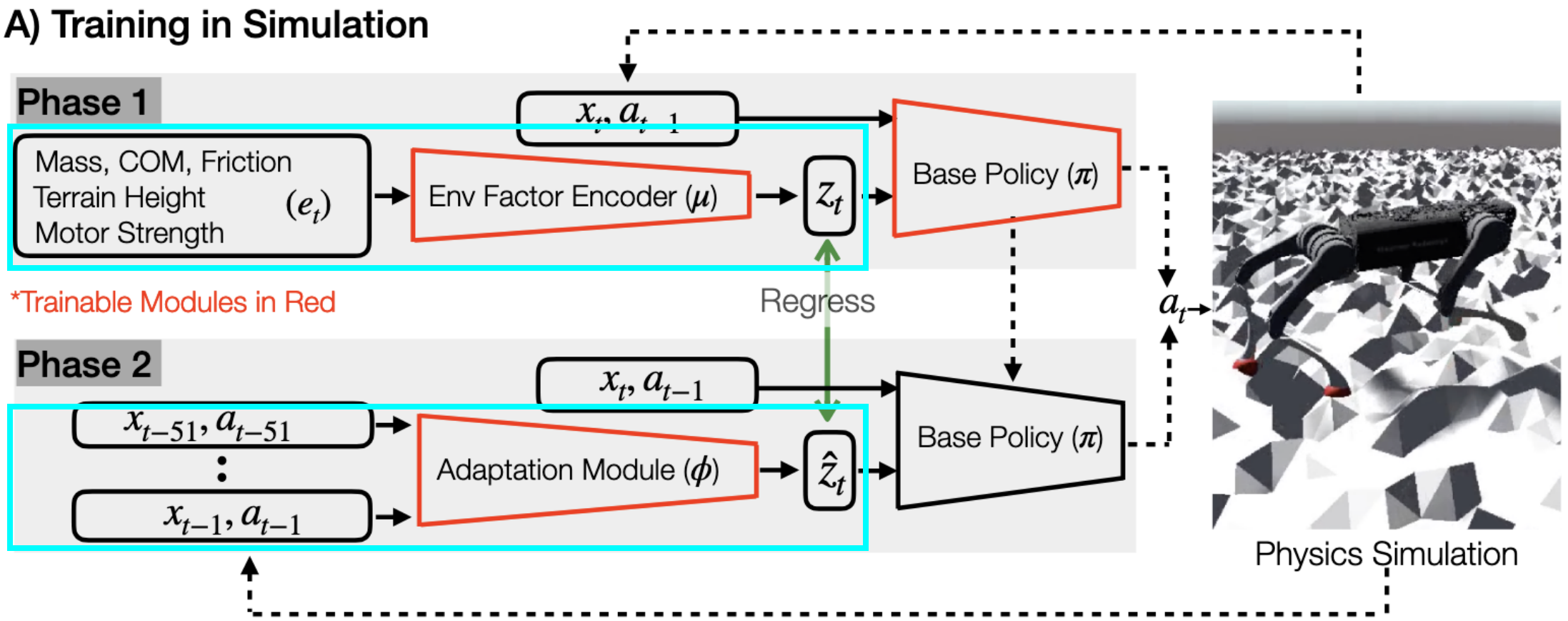
Does the policy need all the information?
Learning to predict them may sabotage the predictor's performance

Idea: Implicit System Identification

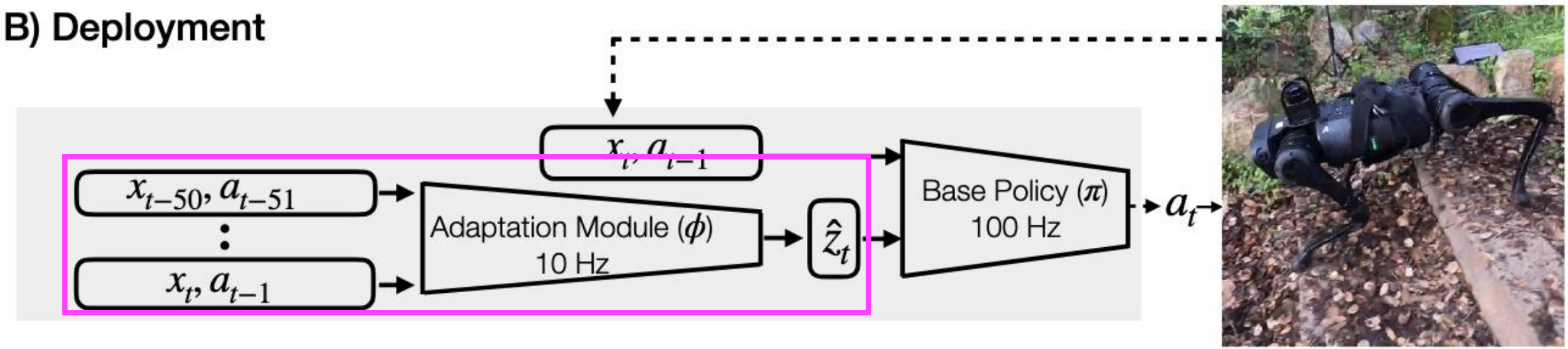


- In the simulator, the underlying physics parameters are known. We train an encoder that converts these physics parameters into latent features. We build a policy that conditions on these physics parameters in the form of latent features..

Rapid Motor Adaptation as Implicit System Identification



- In the real world, the physics parameters are unknown. We train an adaptation module that infers those physics parameters in the latent feature space



- Continuously estimate these extrinsics online, which is key to real-time adaptation

Domain Randomization

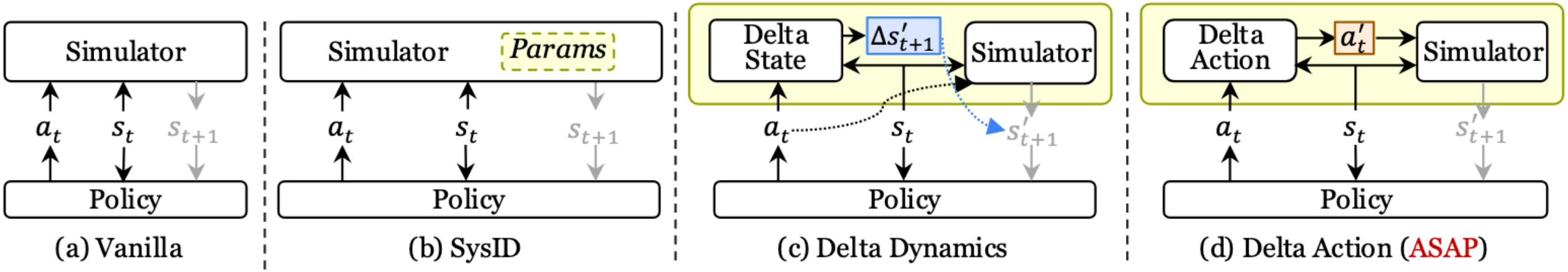
Parameters	Training Range	Testing Range
Friction	[0.05, 4.5]	[0.04, 6.0]
K_p	[50, 60]	[45, 65]
K_d	[0.4, 0.8]	[0.3, 0.9]
Payload (Kg)	[0, 6]	[0, 7]
Center of Mass (cm)	[-0.15, 0.15]	[-0.18, 0.18]
Motor Strength	[0.90, 1.10]	[0.88, 1.22]
Re-sample Probability	0.004	0.01

TABLE I: Ranges of the environmental parameters.

Predicting Physics in Terms of Latent Encodings is More Robust than Predicting them Explicitly

	Success (%)	TTF	Reward	Distance (m)	Samples	Torque	Smoothness	Ground Impact
Robust [52, 40]	62.4	0.80	4.62	1.13	0	527.59	122.50	4.20
SysID [57]	56.5	0.74	4.82	1.17	0	565.85	149.75	4.03
AWR [41]	41.7	0.65	4.17	0.95	40k	599.71	162.60	4.02
RMA w/o Adapt	52.1	0.75	4.72	1.15	0	524.18	106.25	4.55
RMA	73.5	0.85	5.22	1.34	0	500.00	92.85	4.27
Expert	76.2	0.86	5.23	1.35	0	485.07	85.56	3.90

Solutions for Test-time Adaptation



- Ideas:
  1. System Identification: Guess the physical parameters
  2. Delta Dynamics: Learn a dynamic simulator which induces robot policies (e.g. Model Predictive Control). Adapt the dynamic simulator to each scene.
  3. Delta Action: Adapt the physics in simulation to those in the real world. Learn the policy based in the real-world-like physics in sim.

ASAP: Aligning Simulation and Real-World Physics for Learning Agile Humanoid Whole-Body Skills. He et al.

Adaptive Model-Based RL

- The formulation of the dynamics model:

$$x_{t+1:t+H} \approx f_\theta^{\text{AnyCar}}\Big(\underbrace{\hat{x}_{t-K:t}, \hat{a}_{t-K:t-1}}_{\text{noisy state and action history}} , \underbrace{a_{t:t+H-1}}_{\text{future actions}} \Big),$$

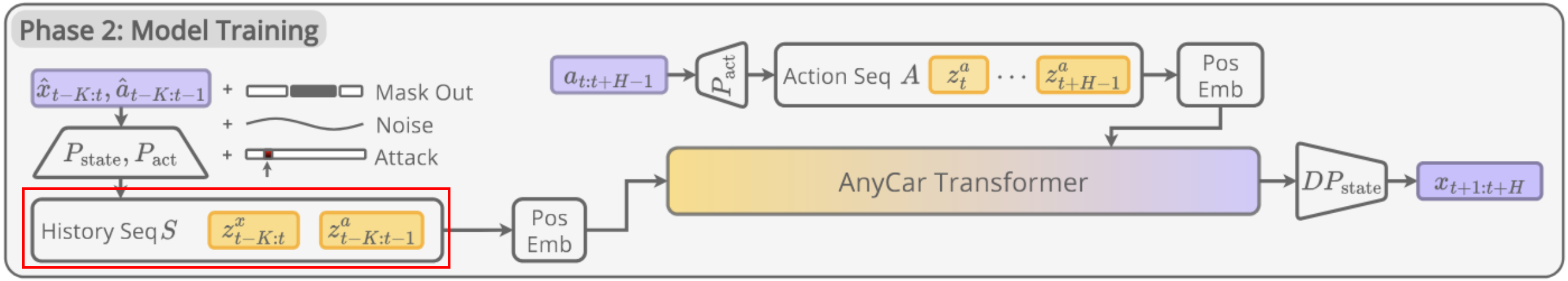
- The definition of states:

$$x_t \triangleq [p_t^x, p_t^y, \psi_t, \dot{p}_t^x, \dot{p}_t^y, \omega]$$

where (p_t^x, p_t^y) denotes the car position, ψ_t is the heading angle, $(\dot{p}_t^x, \dot{p}_t^y)$ denotes the velocity, and ω is the angular velocity

- The definition of actions a_t contain the throttle and steering angle

Adaptive Model-Based RL



Historical state-action pairs reveal the current dynamics of the environment

Learning the Dynamics Model from Existing Synthesized Demonstrations

- The simulation data generation has three sources of diversity:
  - Dynamics
  - Scenario
  - Controller
- Curriculum data generation:
  - Off-policy data: Pure pursuit controller for steering δ and a PD controller for throttle T , to track randomly synthesized reference track in simulation
  - On-policy data: NN-MPPI controller to track agile trajectories in simulation
  - Real-world data: NN-MPPI controller to track agile trajectories in the real world

Deployment

Phase 3: Deployment

1/16 Scale Car

20 cm

1/10 Scale Car

31 cm

w/ optional 3D-Printed Wheels

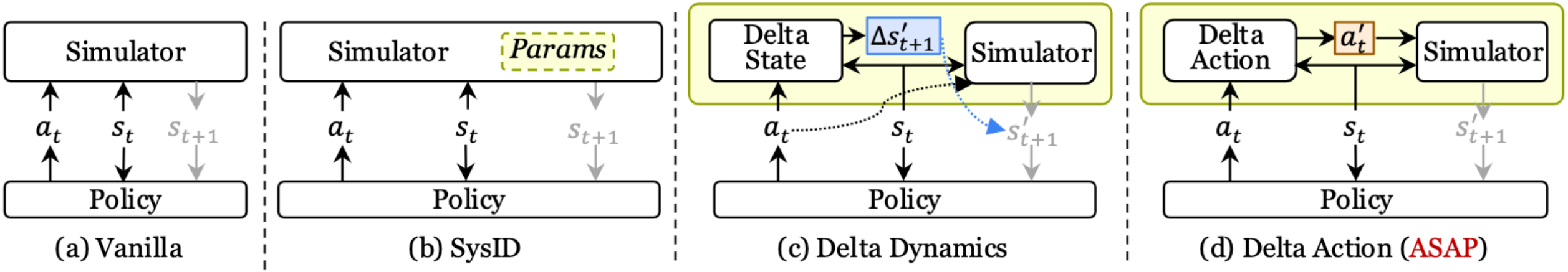
State Estimation Method

(a) MoCap

(b) ZED-VIO

(c) LiDAR-SLAM

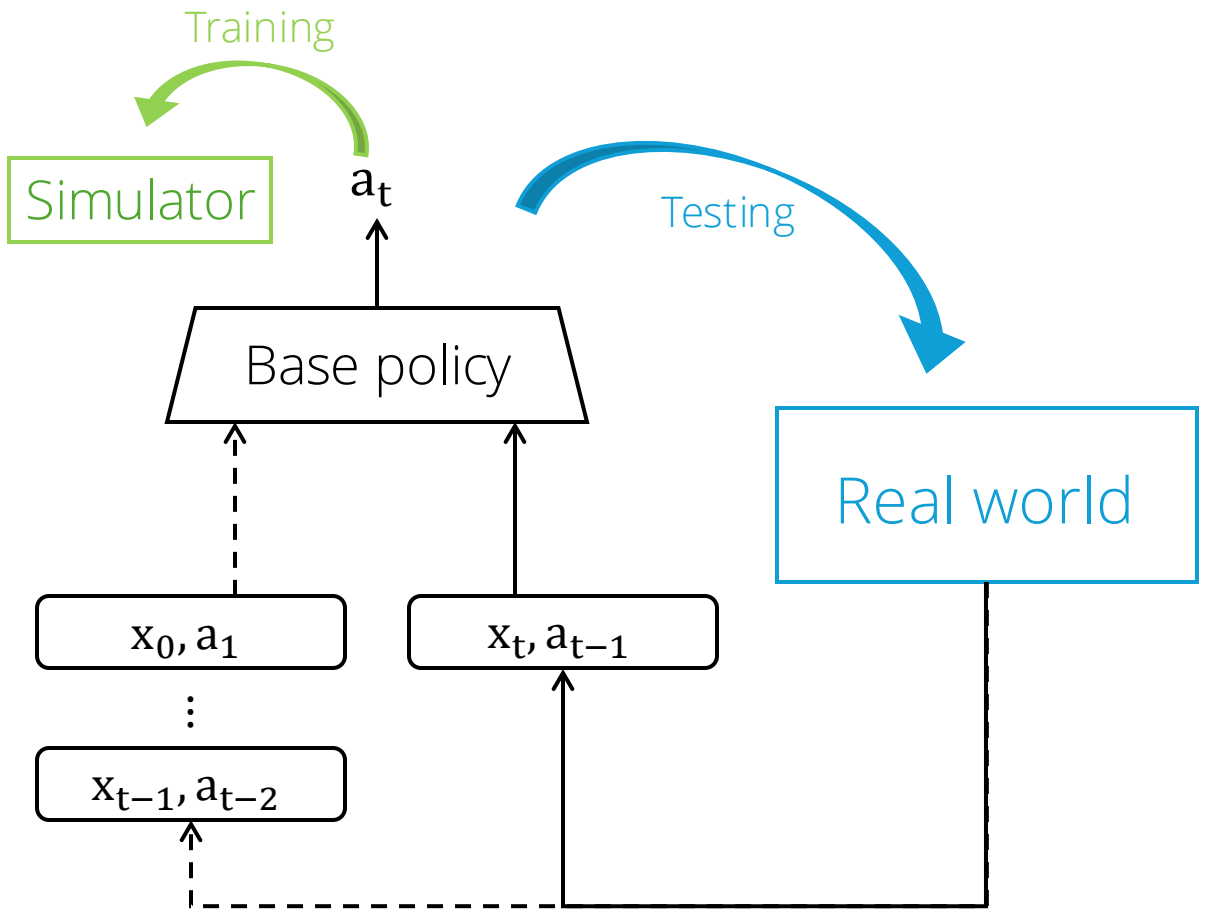
Solutions for Test-time Adaptation

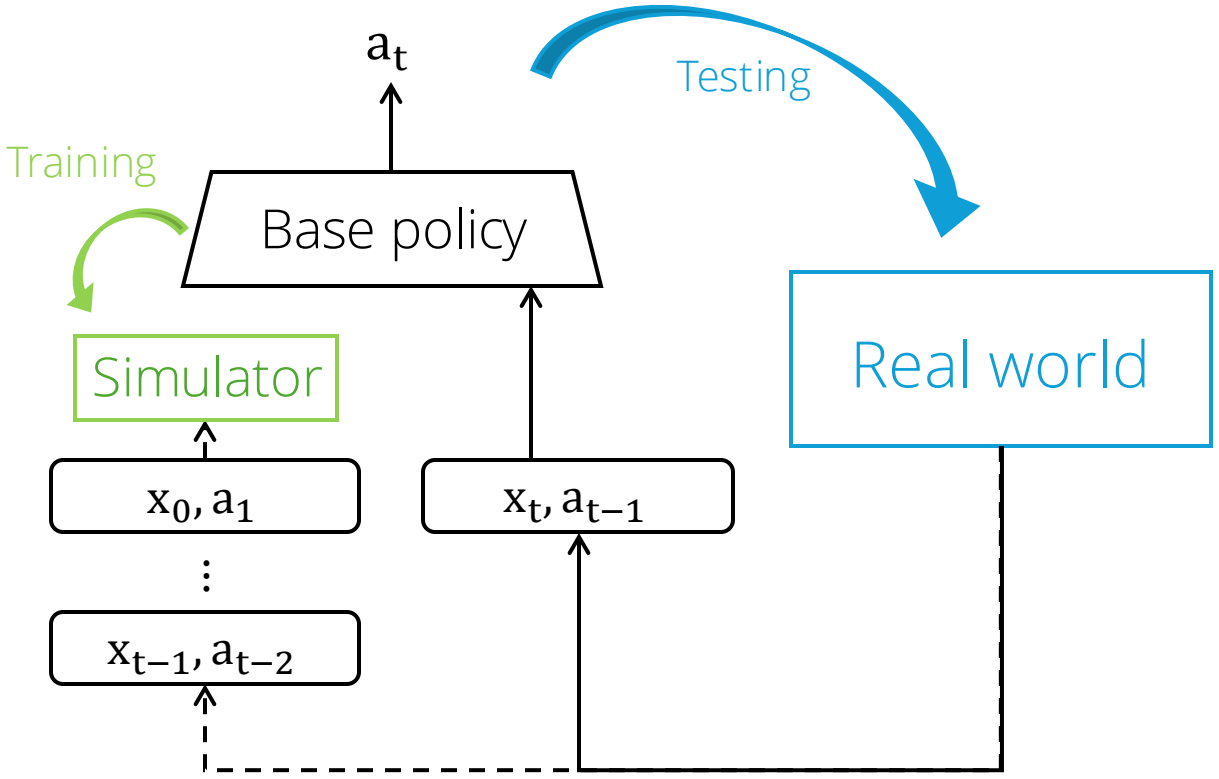


- Ideas:
  1. System Identification: Guess the physical parameters
  2. Delta Dynamics: Learn a dynamic simulator which induces robot policies (e.g. Model Predictive Control). Adapt the dynamic simulator to each scene.
  3. Delta Action: Adapt the physics in simulation to those in the real world. Learn the policy based in the real-world-like physics in sim.

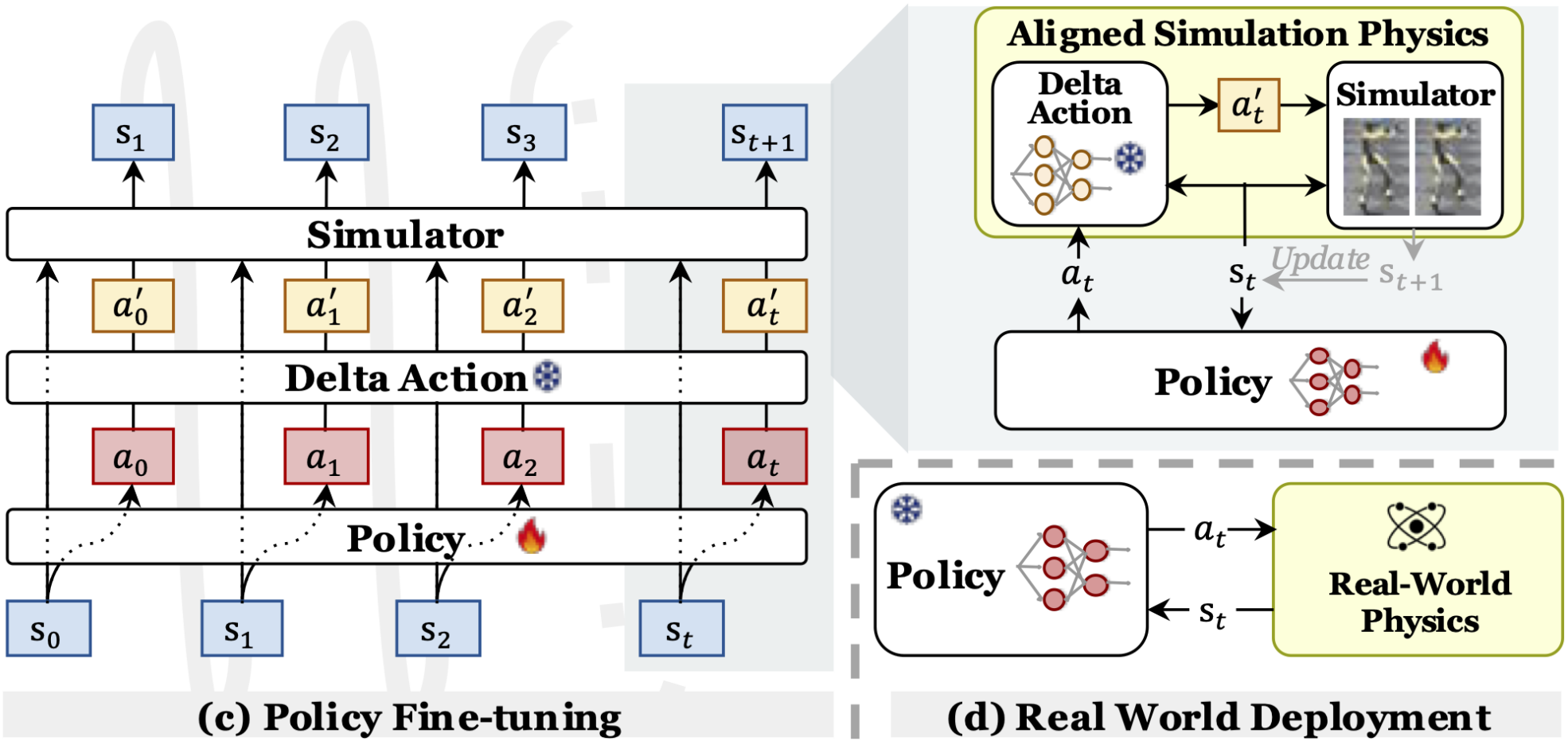
ASAP: Aligning Simulation and Real-World Physics for Learning Agile Humanoid Whole-Body Skills. He et al.

Adapt to Policy vs. Simulator to the Real World



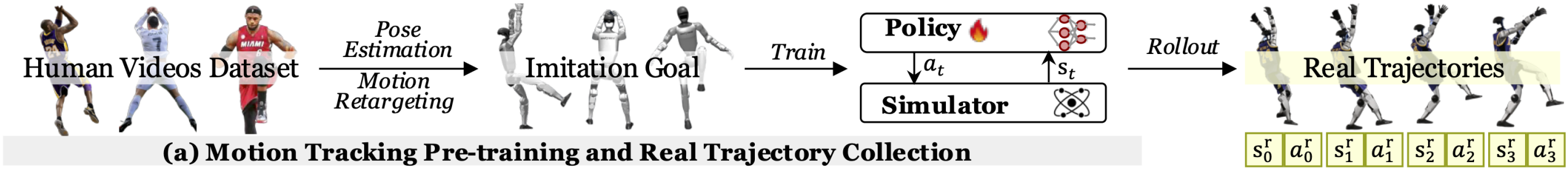


Idea: Predict Delta Actions to Simulate Real-world Physics

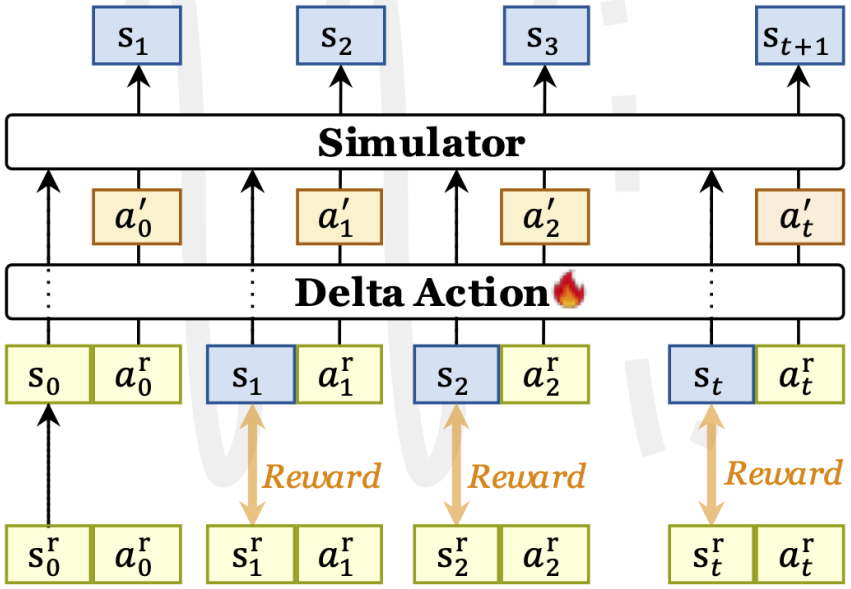


ASAP: Aligning Simulation and Real-World Physics for Learning Agile Humanoid Whole-Body Skills. He et al.

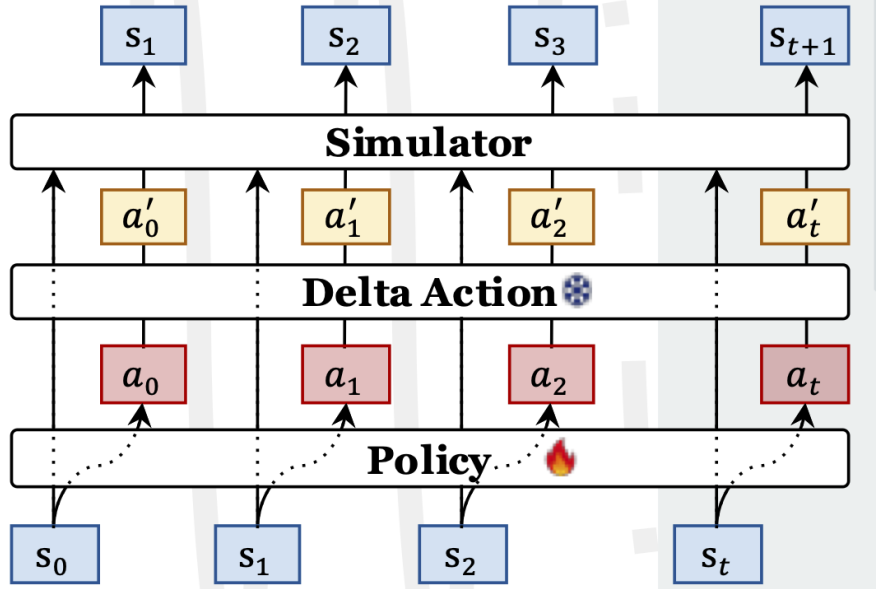
Idea: Predict Delta Actions to Simulate Real-world Physics



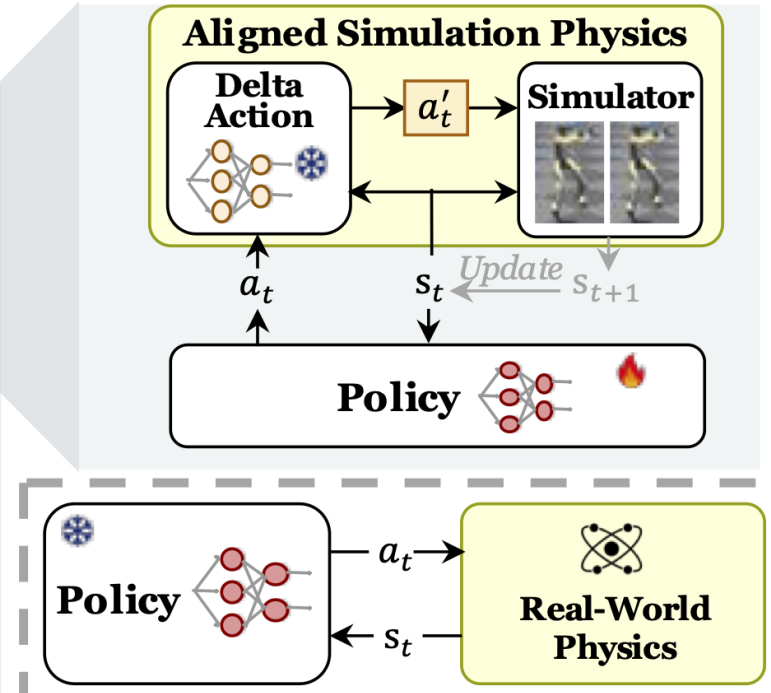
(a) Motion Tracking Pre-training and Real Trajectory Collection



(b) Delta Action Model Training



(c) Policy Fine-tuning

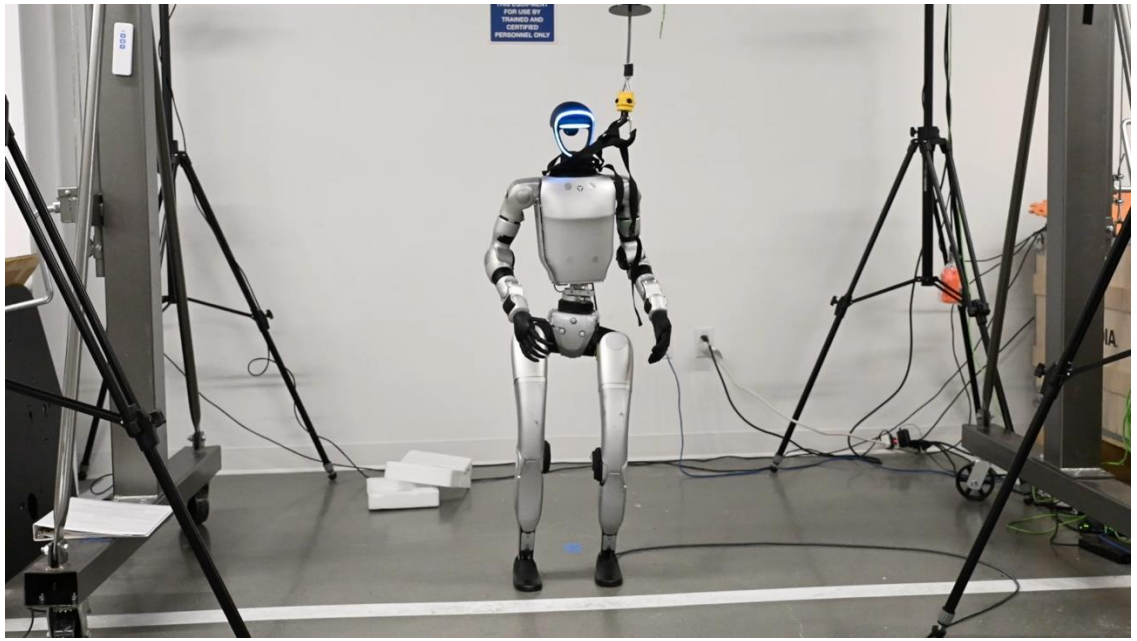


(d) Real World Deployment

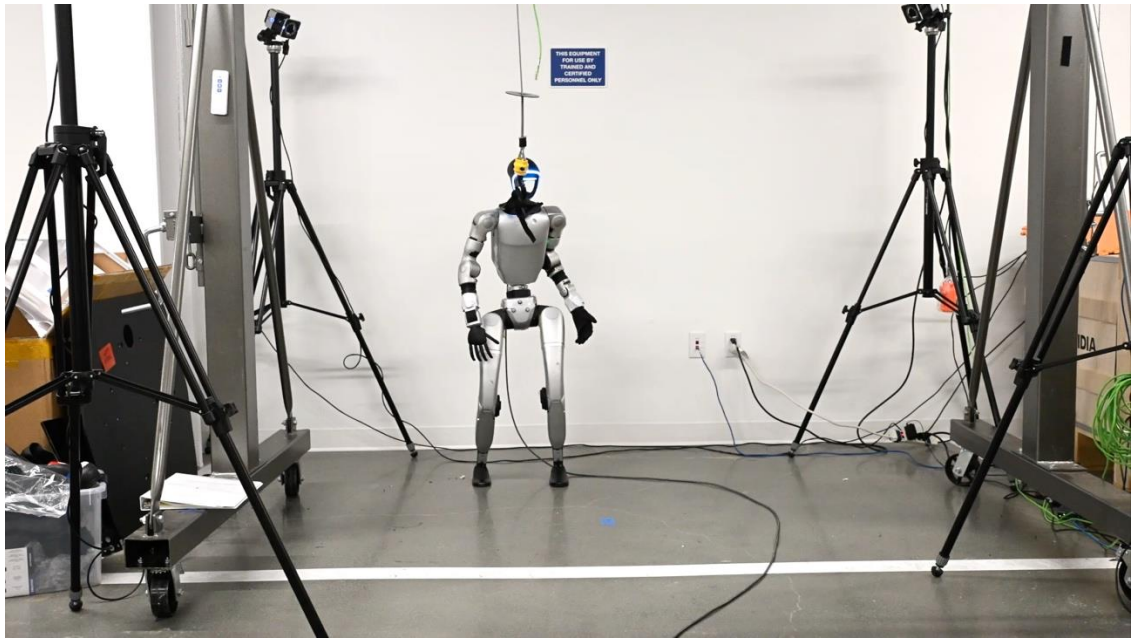
ASAP: Aligning Simulation and Real-World Physics for Learning Agile Humanoid Whole-Body Skills. He et al.

Results: Refined Sim-to-Real Robot Learning

w\o adaptation

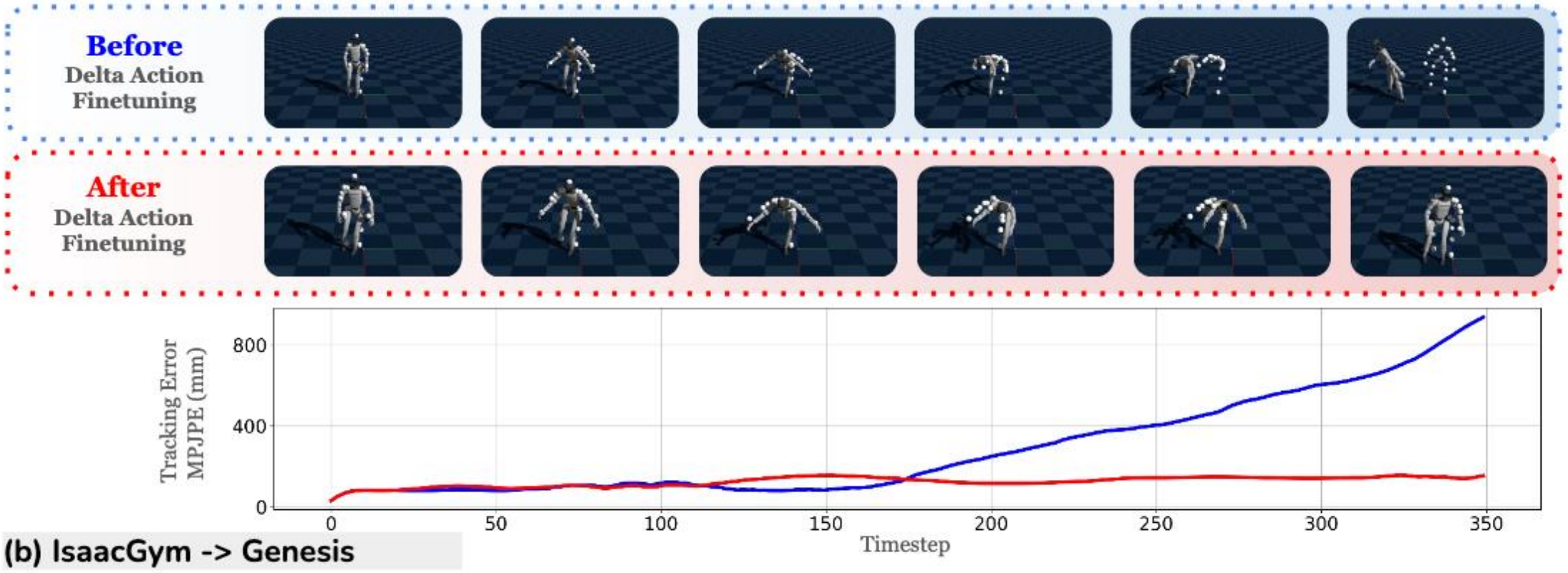


w\ adaptation



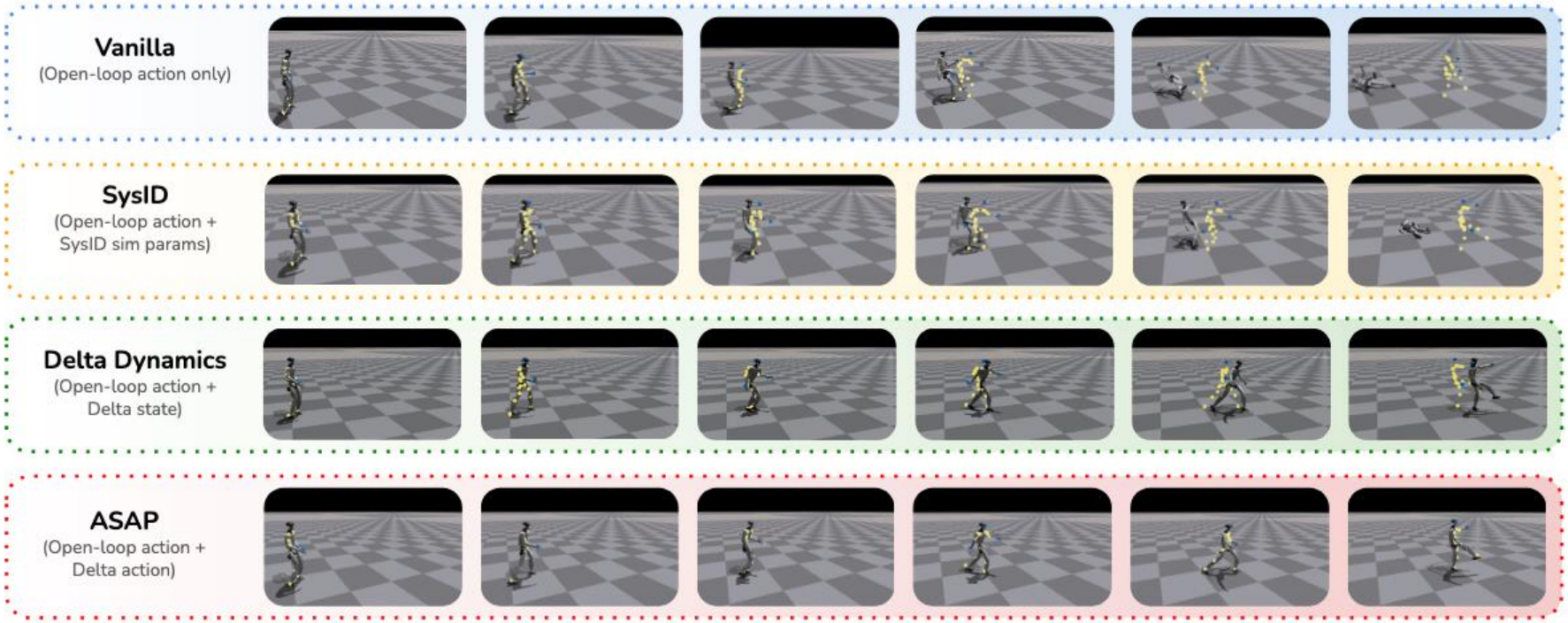
ASAP: Aligning Simulation and Real-World Physics for Learning Agile Humanoid Whole-Body Skills. He et al.

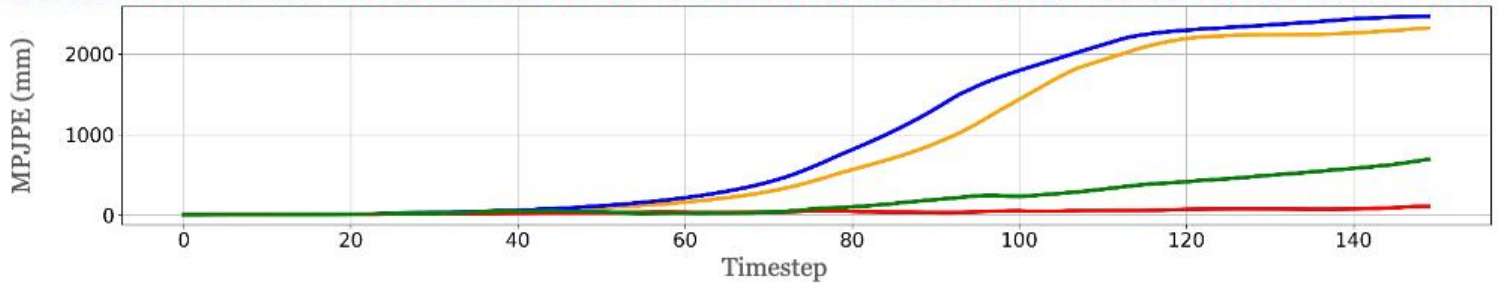
Results: Refined Sim-to-Real Robot Learning



ASAP: Aligning Simulation and Real-World Physics for Learning Agile Humanoid Whole-Body Skills. He et al.

Results 2: Better In-Domain Motion Tracking than SysID and Delta Dynamics



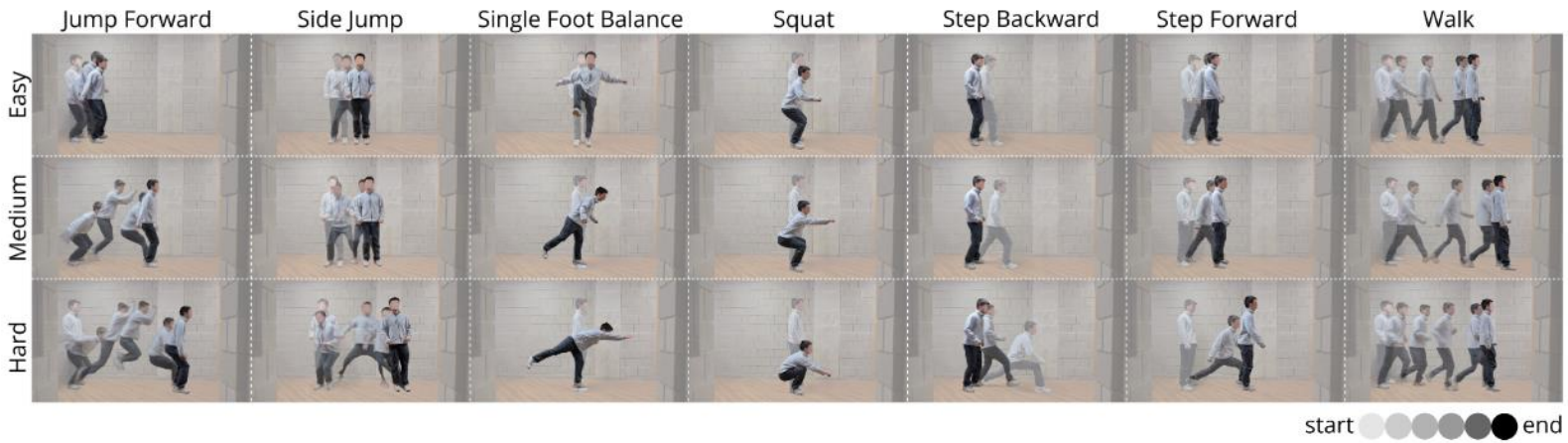


Results 3: Better Cross-Domain Motion Tracking than SysID and Delta Dynamics

TABLE IV

CLOSED-LOOP MOTION IMITATION EVALUATION ACROSS DIFFERENT SIMULATORS. ALL VARIANTS ARE TRAINED WITH IDENTICAL REWARDS.

	Test Environment	IsaacSim					Genesis				
Level	Method	Succ ↑	$E_{\text{g-mpjpe}}$ ↓	E_{mpjpe} ↓	E_{acc} ↓	E_{vel} ↓	Succ ↑	$E_{\text{g-mpjpe}}$ ↓	E_{mpjpe} ↓	E_{acc} ↓	E_{vel} ↓
Easy	Oracle (IsaacGym → IsaacGym)	100%±0.000%	97.5±0.605	43.2±0.112	2.56±0.024	4.48±0.023	100%±0.000%	97.5±0.605	43.2±0.112	2.56±0.024	4.48±0.023
	Vanilla (IsaacGym → TestEnv)	100%±0.000%	**107** ±0.578	45.4±0.169	2.83±0.012	4.59±0.021	100%±0.000%	140±1.85	**70.1**±0.626	2.68±0.042	4.65±0.046
	SysID	100%±0.000%	**105** ±1.35	47.8±0.970	3.09±0.011	4.98±0.020	100%±0.000%	**127**±0.233	79.9±0.330	2.99±0.035	4.95±0.012
	DeltaDynamics	100%±0.000%	127±2.97	56.7±0.390	3.50±0.028	5.56±0.031	83.3%±0.000%	168±7.62	87.0±1.51	3.08±0.18	5.39±0.34
	ASAP	100%±0.000%	**106** ±0.498	**44.3** ±0.103	**2.74** ±0.025	**4.46** ±0.020	100%±0.000%	**125**±4.75	73.5±0.570	**2.10**±0.083	**4.11**±0.133
Medium	Oracle (IsaacGym → IsaacGym)	100%±0.000%	111±0.635	48.8±0.133	2.63±0.017	4.82±0.019	100%±0.000%	111±0.635	48.8±0.133	2.63±0.017	4.82±0.019
	Vanilla (IsaacGym → TestEnv)	100%±0.000%	**114**±0.720	**49.2**±0.104	2.92±0.021	5.07±0.016	94.3%±7.00%	169±5.76	72.0±0.692	3.26±0.076	5.86±0.101
	SysID	100%±0.000%	**115**±1.256	**49.1**±0.560	3.43±0.021	5.01±0.017	100%±0.000%	138±2.70	75.4±1.18	3.14±0.042	5.50±0.058
	DeltaDynamics	83.3%±0.000%	151±2.62	68.0±0.364	2.90±0.047	5.90±0.107	83.3%±0.000%	190±1.46	89.4±0.50	3.44±0.16	7.49±0.11
	ASAP	100%±0.000%	**112** ±1.648	**49.3** ±0.574	**2.53** ±0.019	**4.45** ±0.026	100%±0.000%	**126**±1.63	**71.2**±0.163	**2.81**±0.037	**5.13**±0.066
Hard	Oracle (IsaacGym → IsaacGym)	100%±0.000%	116±0.711	52.5±0.298	3.40±0.027	6.16 ±0.028	100%±0.000%	116±0.711	52.5±0.298	3.40±0.027	6.16 ±0.028
	Vanilla (IsaacGym → TestEnv)	100%±0.000%	148±0.845	**51.6**±0.137	4.41±0.055	6.88±0.064	82.9%±5.70%	175±9.77	80.7±1.69	3.87±0.175	7.19±0.199
	SysID	100%±0.000%	165±3.83	58.4±0.229	4.87±0.197	7.13±0.131	100%±0.000%	186±3.84	93.0±1.49	4.98±0.245	8.98±0.119
	DeltaDynamics	66.7%±0.000%	137±2.59	60.2±0.477	4.20±0.041	7.10±0.024	60.0%±5.70%	190±14.0	89.6±9.34	4.29±1.16	8.70±2.33
	ASAP	100%±0.000%	**129** ±1.57	56.5 ±1.15	**3.72** ±0.036	**6.52** ±0.042	100%±0.000%	**129**±2.31	**77.0**±1.07	**2.69**±0.040	**5.65**±0.073



ASAP: Aligning Simulation and Real-World Physics for Learning Agile Humanoid Whole-Body Skills. He et al.