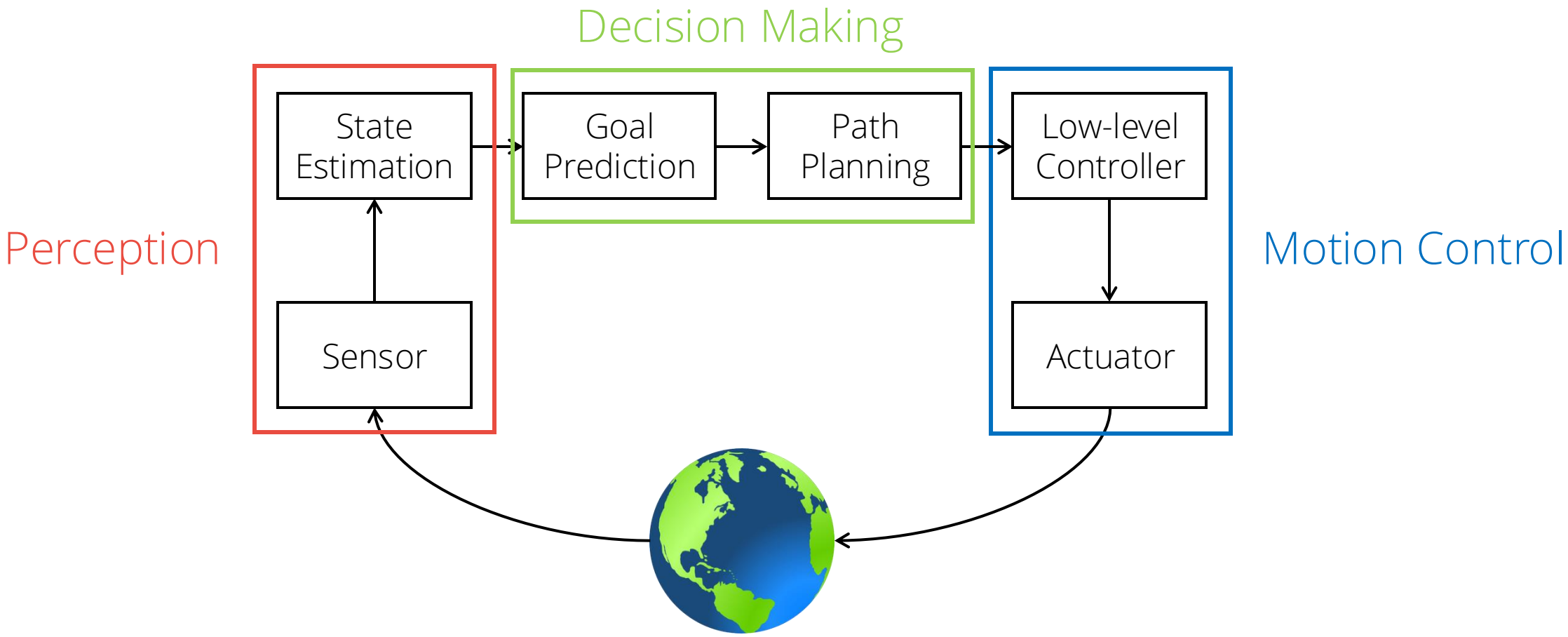
Robot Perception and Learning

Imitation learning

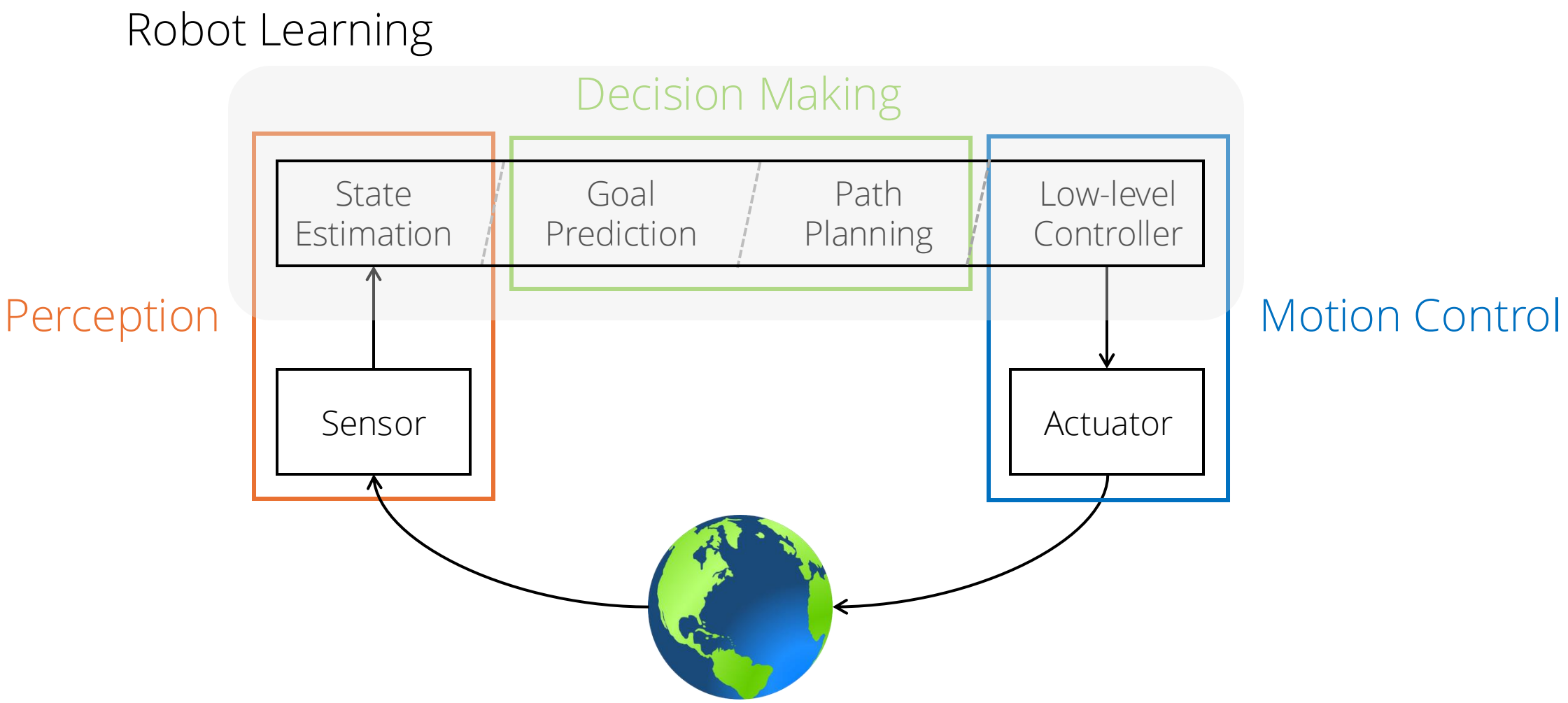
Tsung-Wei Ke

Fall 2025

We Discussed Three Individual Components in Robotics



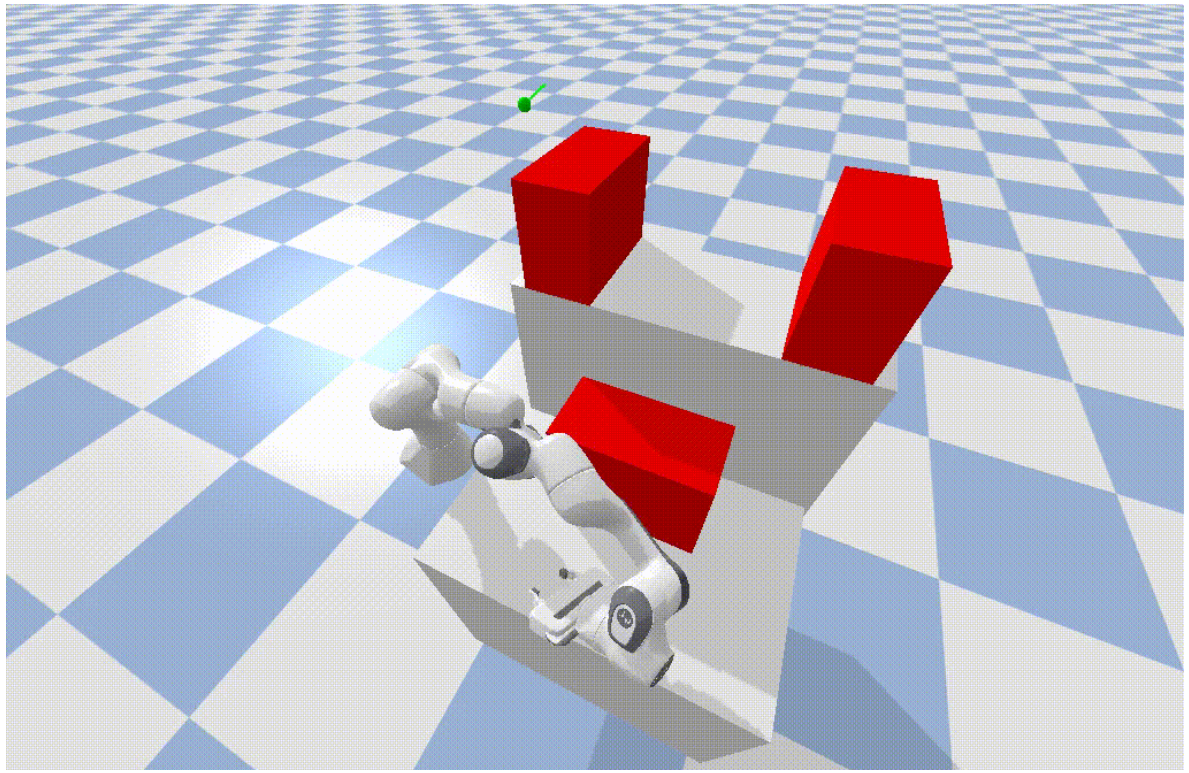
Robot Learning: Learning an End-to-end Framework for These Components

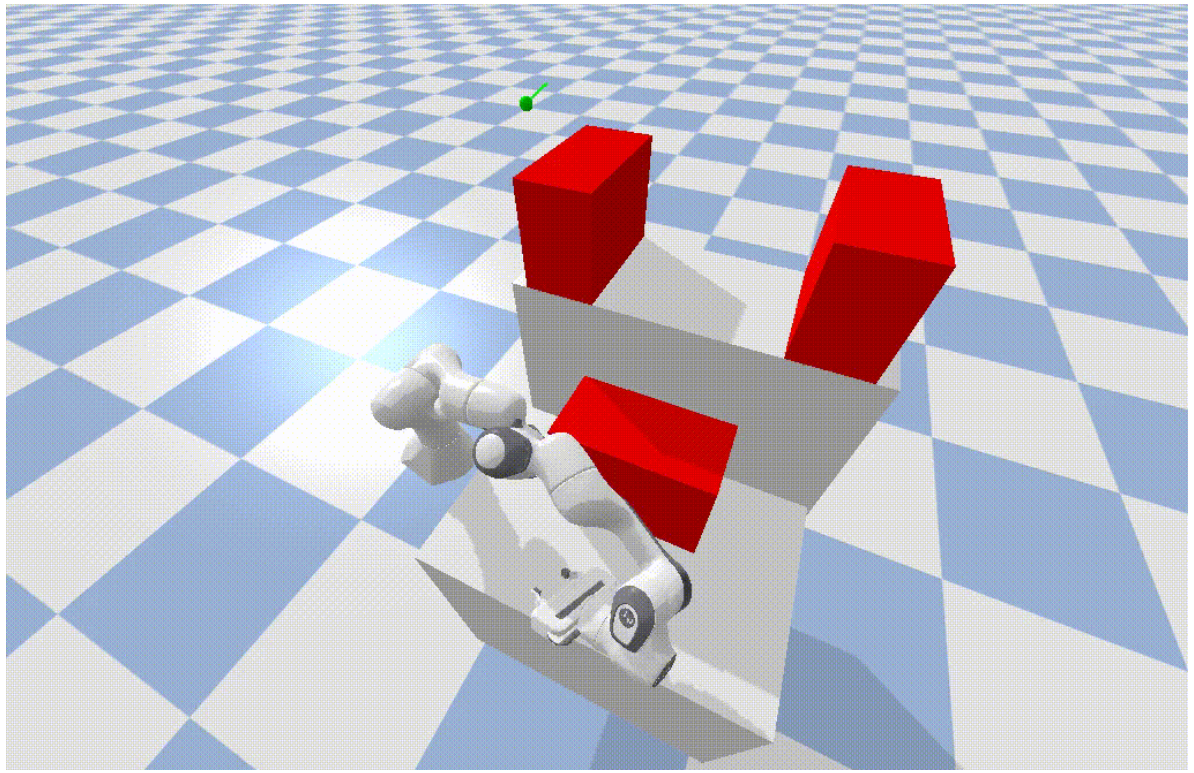


Robot Learning for Low-level Controller

OmniRetarget: Interaction-Preserving Data Generation for Humanoid Whole-Body Loco-Manipulation and Scene Interaction. Yang et al.

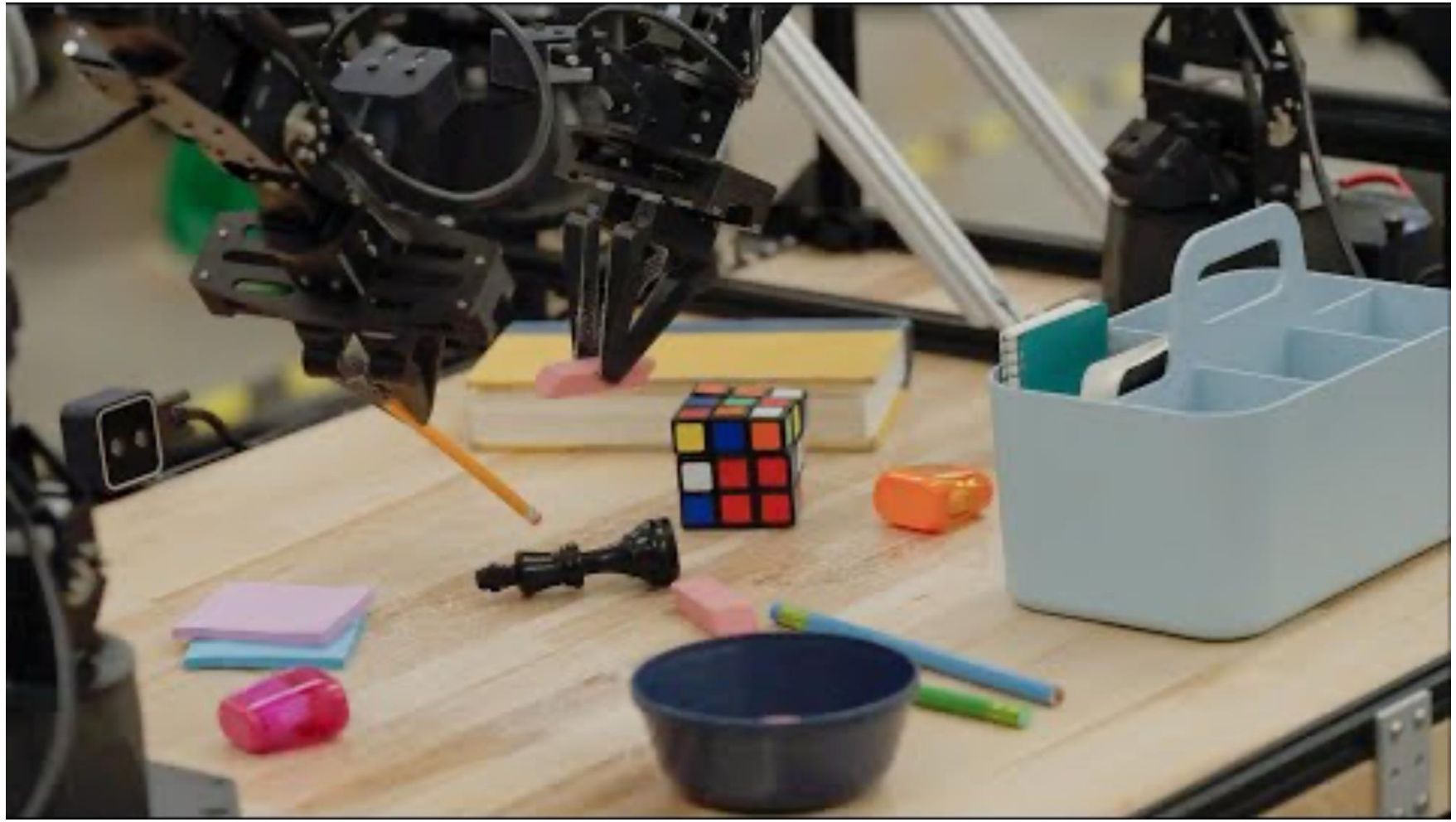
Robot Learning for Path Planning



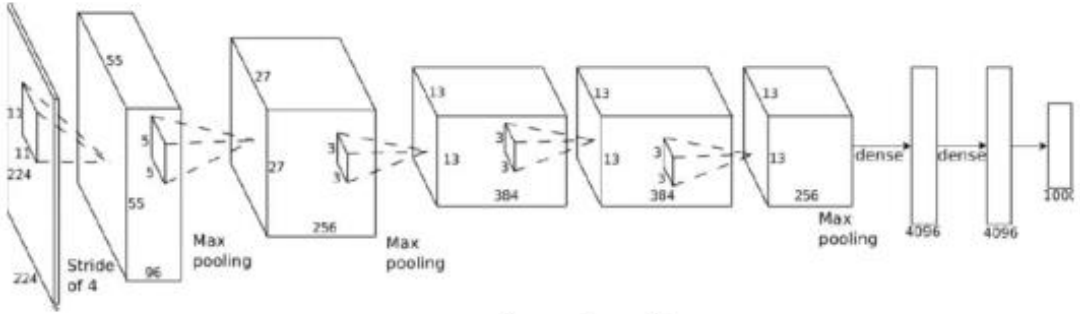


Motion Planning Diffusion: Learning and Planning of Robot Motions with Diffusion Models. Carvalho et al

Robot Learning for Goal (End-Effector Pose) Prediction



Policy Learning: Learn a Model that Predicts Actions from Observations



$\mathbf{o}_t$ $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ $\mathbf{a}_t$

$\mathbf{s}_t$ – state

$\mathbf{o}_t$ – observation → Obtained by your sensors

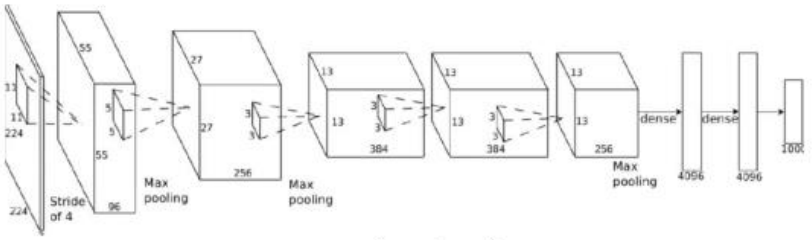
$\mathbf{a}_t$ – action → End-effector 6DoF pose / joint angles / torques?

$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ – policy

$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ – policy (fully observed)

Slide adapted from S. Levine

Policy Learning vs. Sequential Labeling

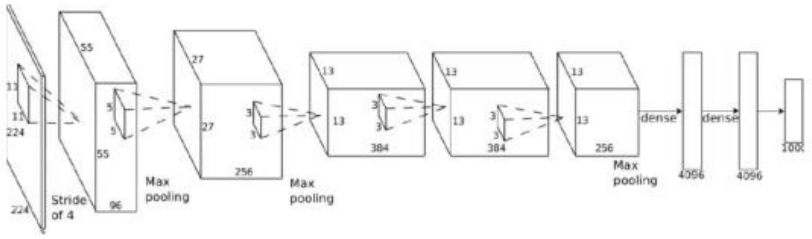


$\mathbf{o}_t$ $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ $\mathbf{a}_t$

$\mathbf{s}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{a}_t$ – action

$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ – policy
$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ – policy (fully observed)

Sequential labeling

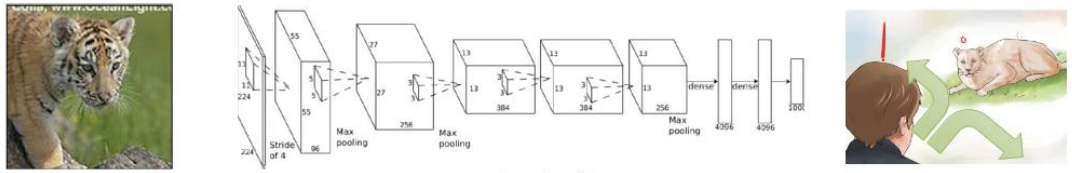


Tiger

o_t y_t

Policy Learning vs. Sequential Labeling

Behavior cloning

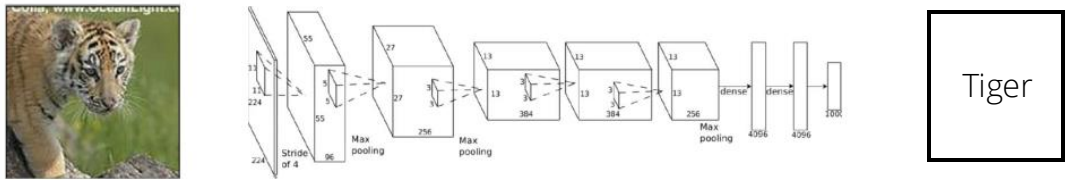


$\mathbf{o}_t$ $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ $\mathbf{a}_t$

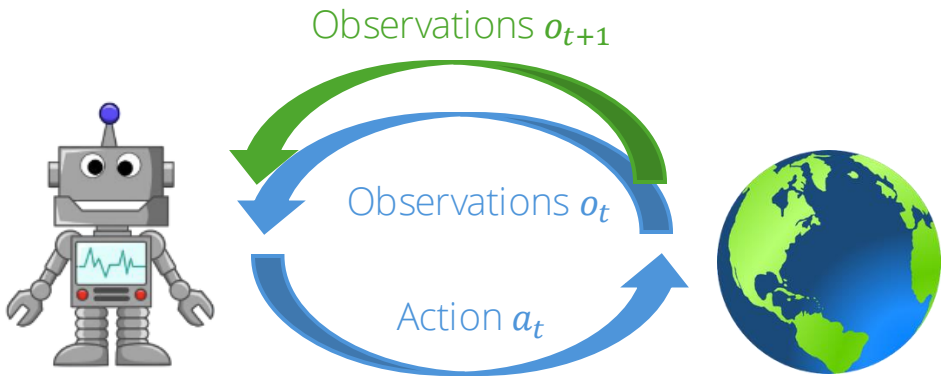
$\mathbf{s}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{a}_t$ – action

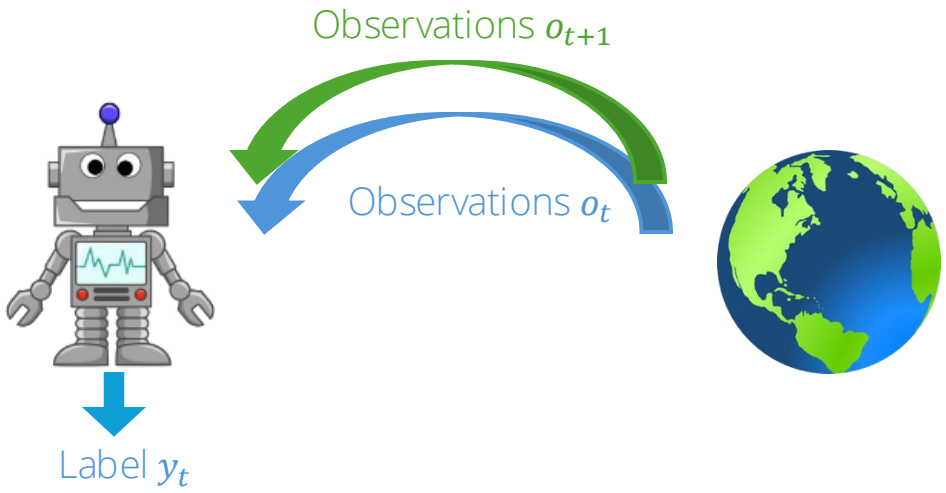
$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ – policy
$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ – policy (fully observed)

Sequential labeling



o_t y_t



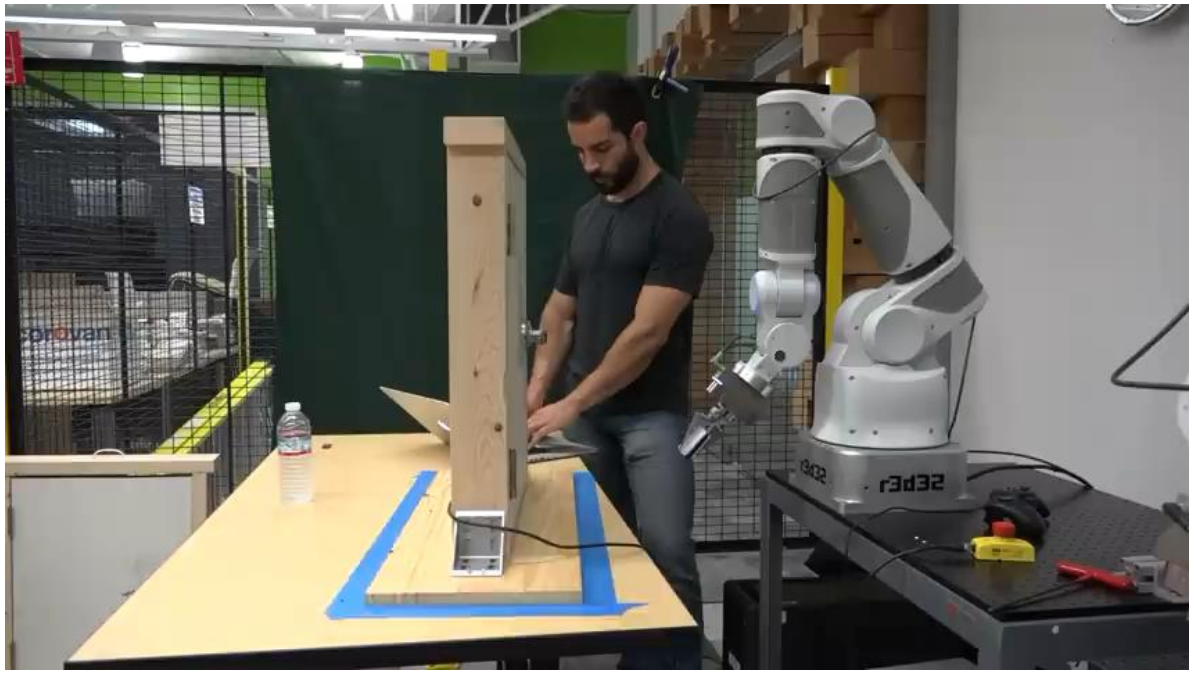


Slide adapted from S. Levine

Learning to Act by Cloning Behaviors

https://blog.ohiohealth.com/simple-ways-teach-family-heritage/

Collect Action Labels by Teleoperation

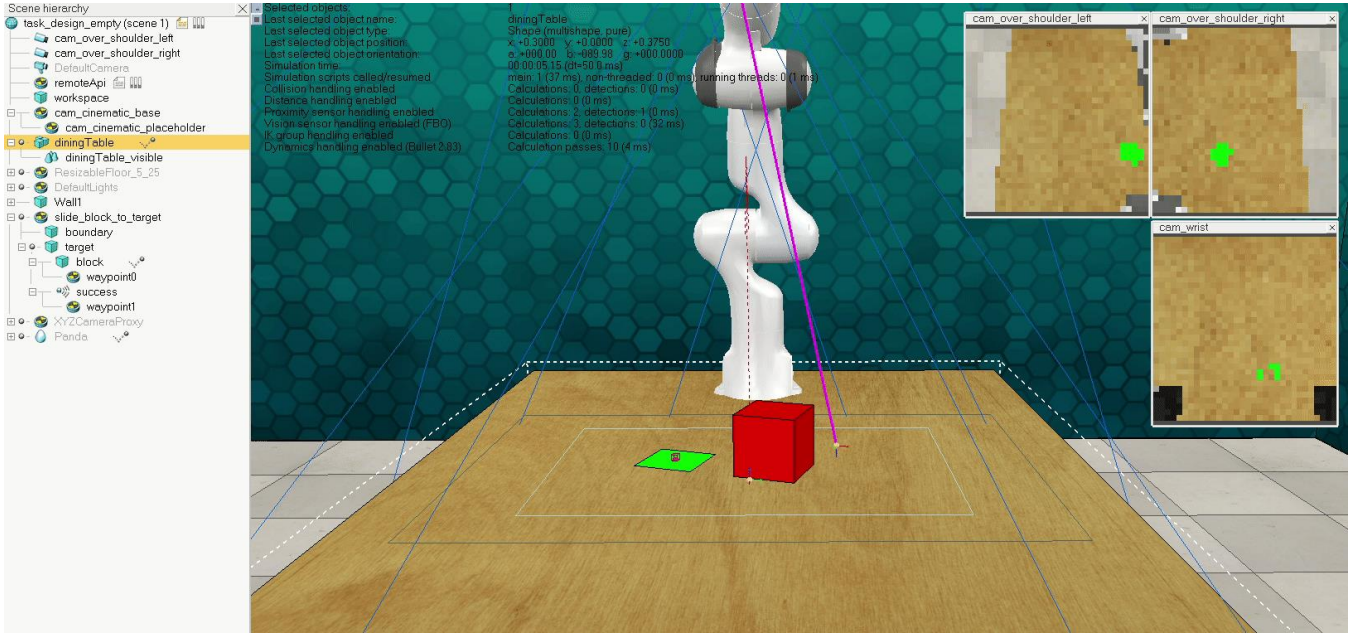


https://youtu.be/9-9udQtO1PY?si=AZCr7n_9V41BA2o9

https://youtu.be/Bhg3uOx9ZPw?si=et7L0endzGvGPJz-

Collect Action Labels by Goal-Pose Annotations and Motion Planners

CuRobo: CUDA Accelerated Robot Library



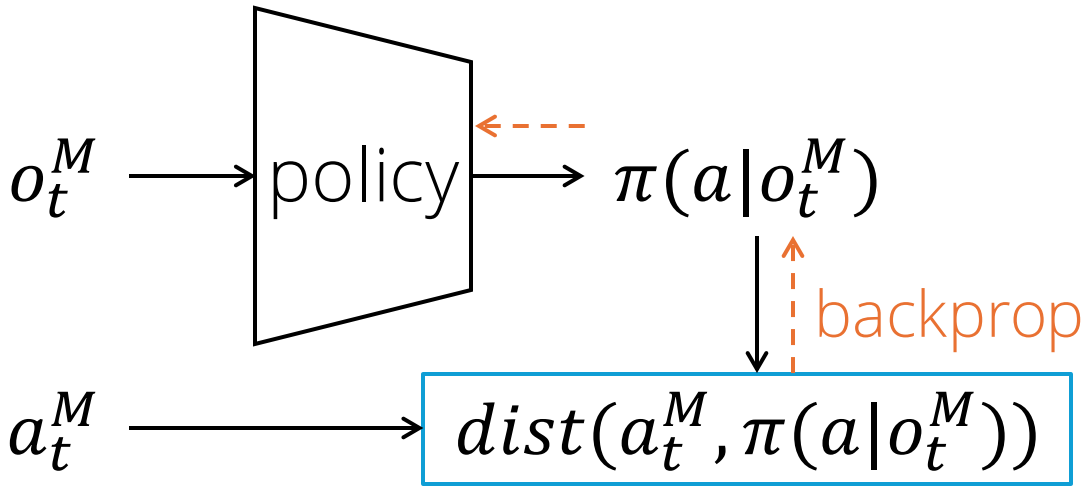
RLBench: Robot Learning Benchmark. James et al.

Behavior Cloning: Learning to Act by Imitation

Step 1: Collect demonstrations

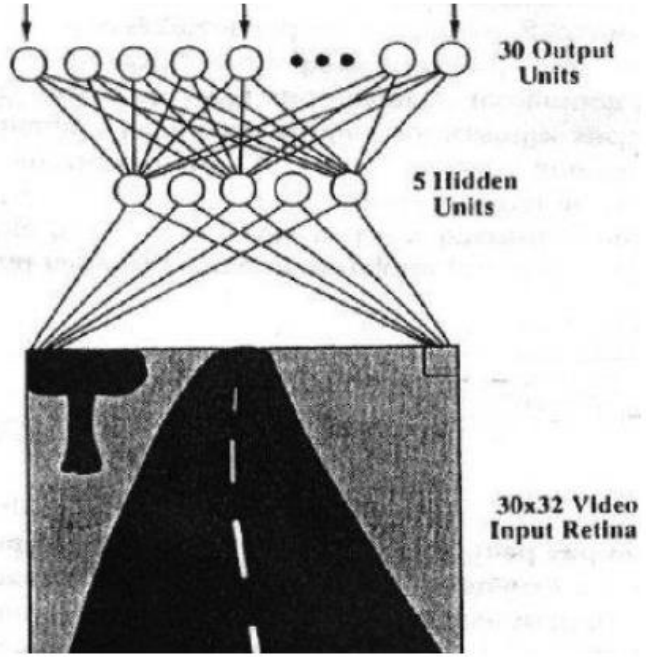
$$o_1^1, a_1^1, o_2^1, a_2^1, \dots, o_T^1, a_T^1$$
$$o_1^2, a_1^2, o_2^2, a_2^2, \dots, o_T^2, a_T^2$$
$$\dots$$
$$o_1^N, a_1^N, o_2^N, a_2^N, \dots, o_T^N, a_T^N$$

Step 2: Train the policy to imitate



Autonomous Driving with Imitation Learning

ALVINN: **A**utonomous **L**and **V**ehicle **I**n a **N**eural **N**etwork
1989



Autonomous Driving with Imitation Learning

Imitation Learning Looks Great in General

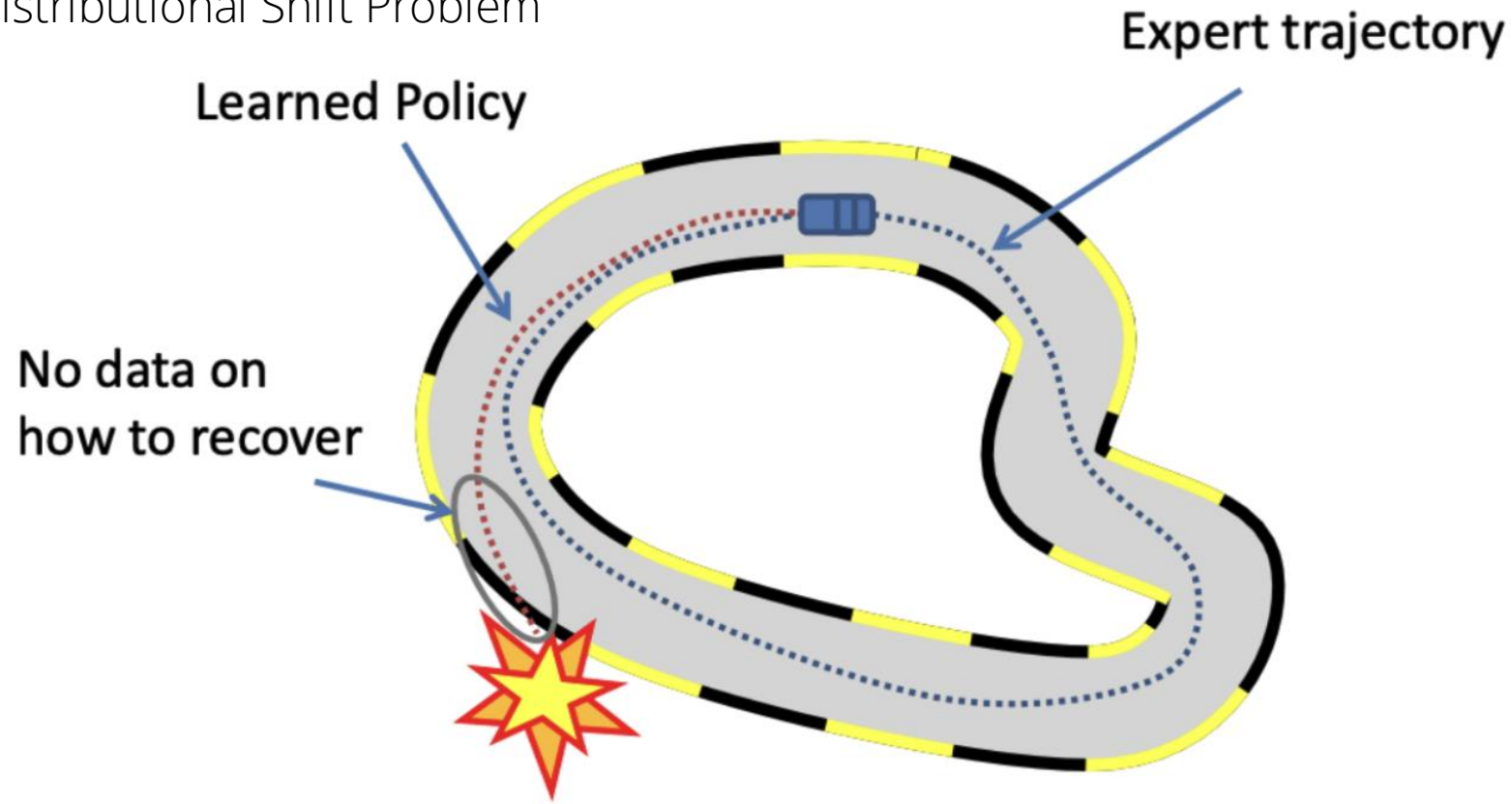
https://youtu.be/tlThdr3O5Qo?si=YBMonUrjnD0YG2y6

However, It Doesn't Always Work…

Video credit @AIDRIVRClips on Youtube

Co-Variate Shift Problem

Distributional Shift Problem

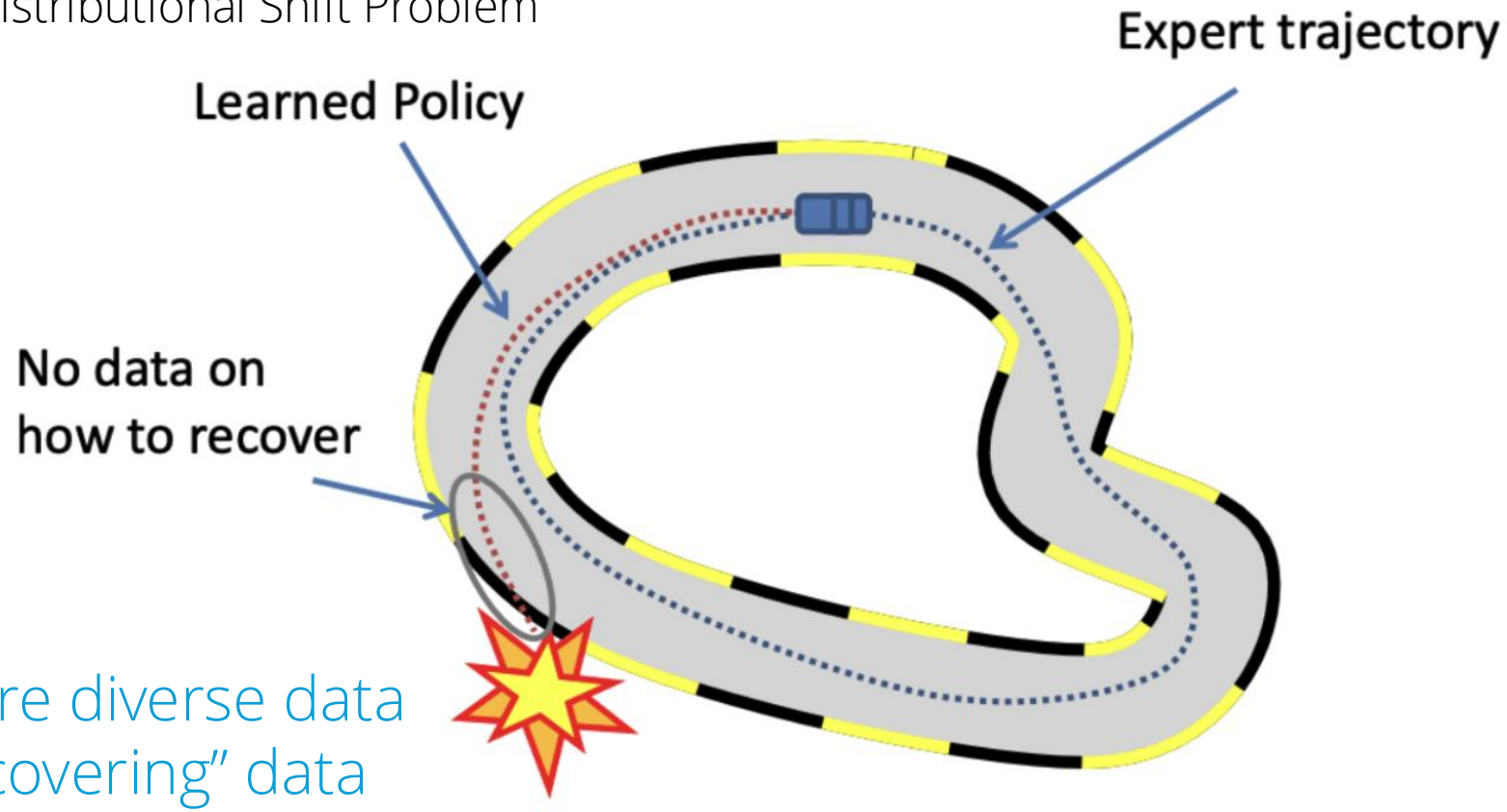


Any solution?

The “cloned” policy can go out-of-distribution!

Co-Variate Shift Problem

Distributional Shift Problem



Solution:

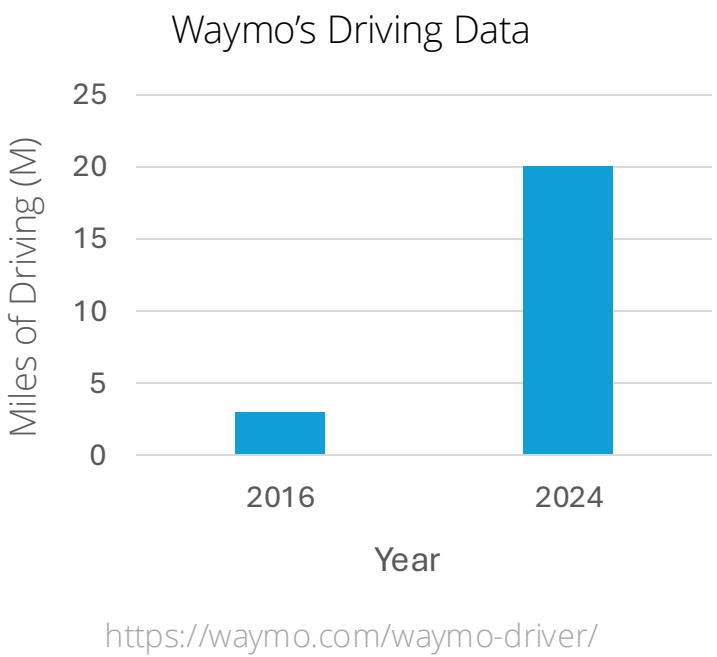
1. Collect more diverse data
2. Collect "recovering" data by augmentation

The "cloned" policy can go out-of-distribution!

Slide credit P. Agrawal

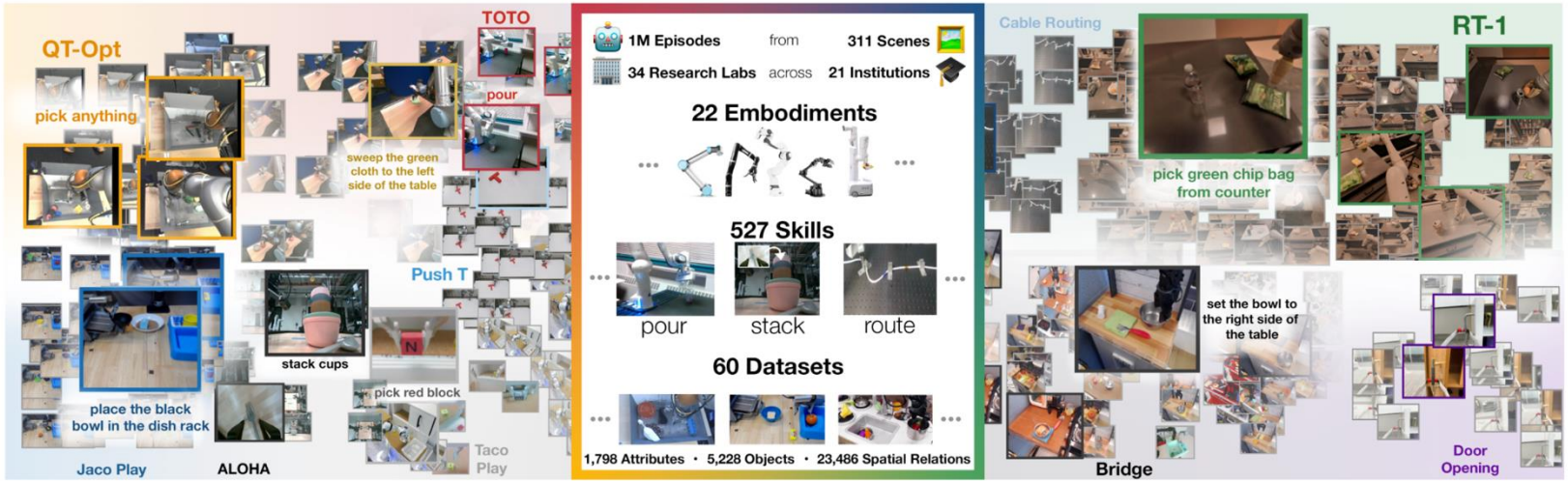
Slide adapted from John Schulman

Solution 1: Collect More Demonstration Data

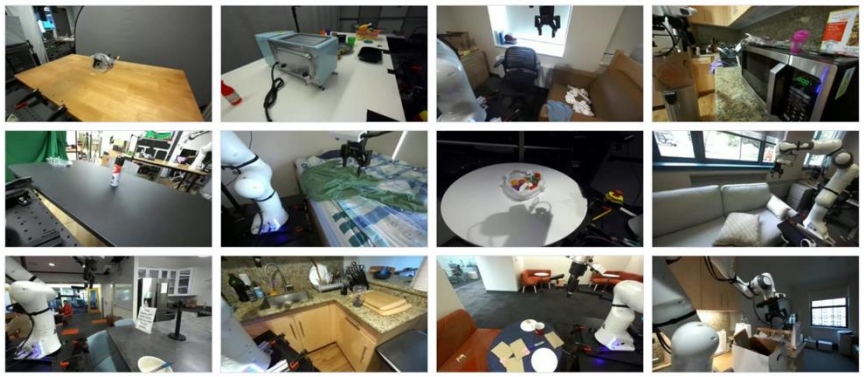


https://waymo.com/waymo-driver/

Open X-Emobiment



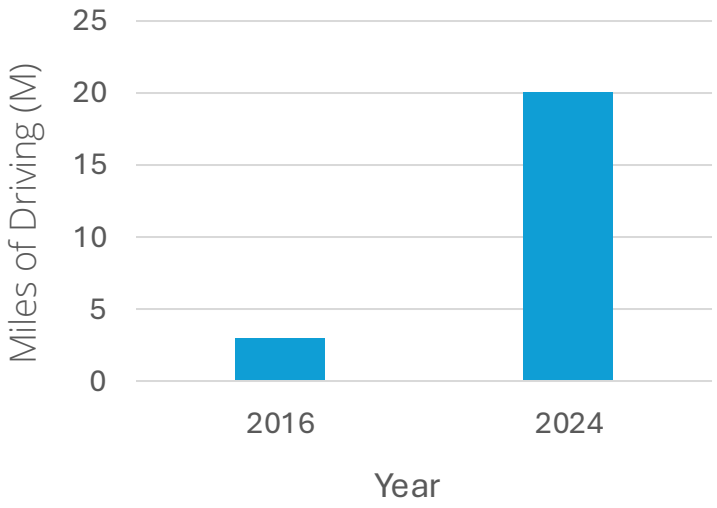
DROID



RoboCasa

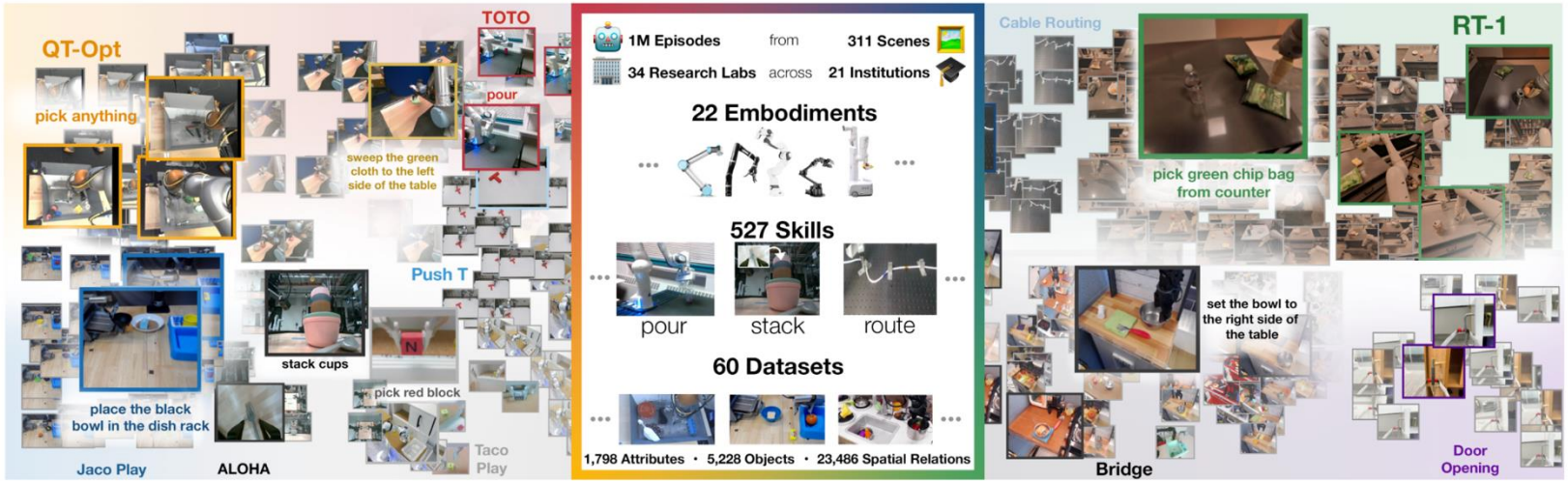
Solution 1: Collect More Demonstration Data

Waymo's Driving Data

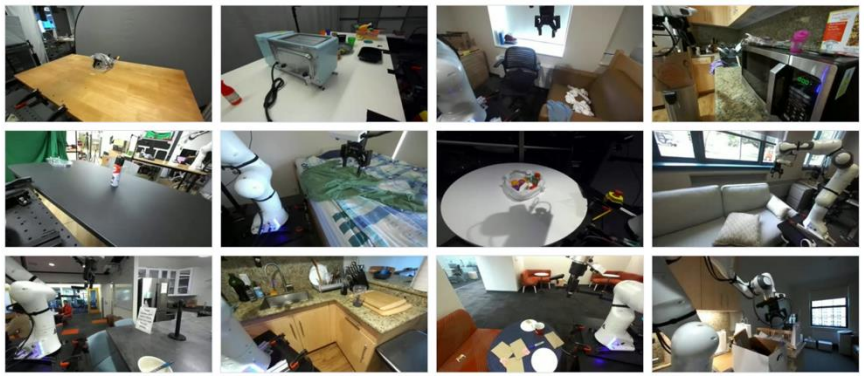


https://waymo.com/waymo-driver/

Open X-Embodiment



DROID

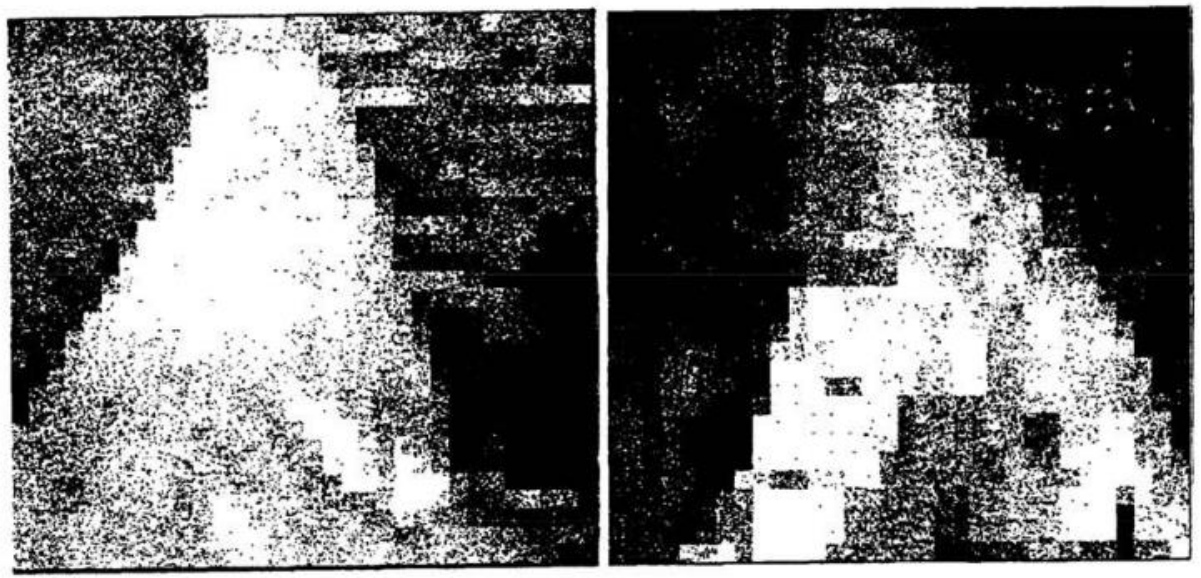


RoboCasa

Problems:

1. Reproducibility: How to replay corner cases?
2. Cost: How much does it take to cover all/most distributional drift?

Solution 1: Collect More Demonstration Data in Simulation

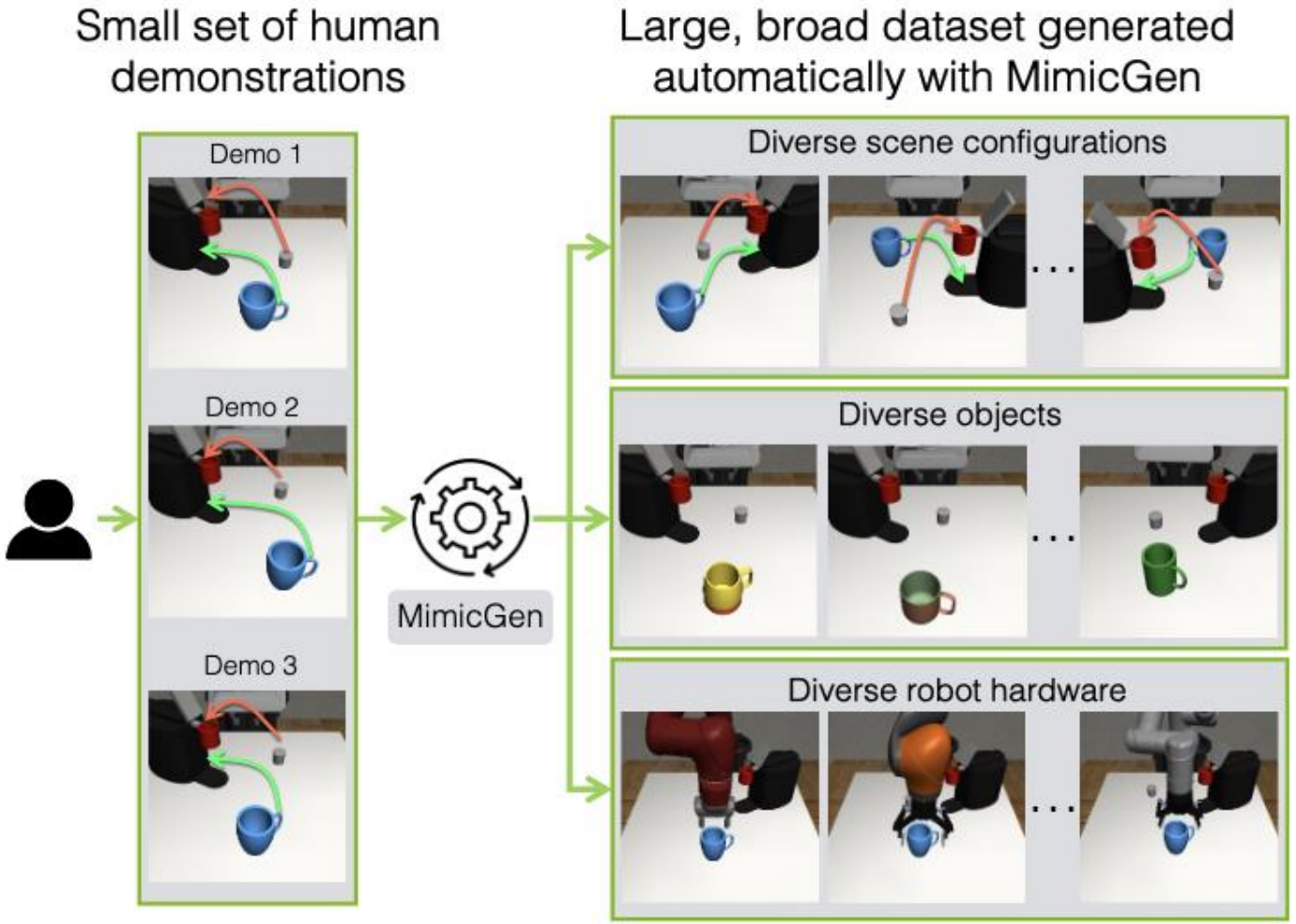


Real Road Image Simulated Road Image

ALVINN: An Autonomous Land Vehicle in a Neural Network. D. A. Pomerleau.

- Simulate road images and the corresponding ground-truth steering angle
- Online adaptation to human-driver steering angle control
- Neural-network based policies

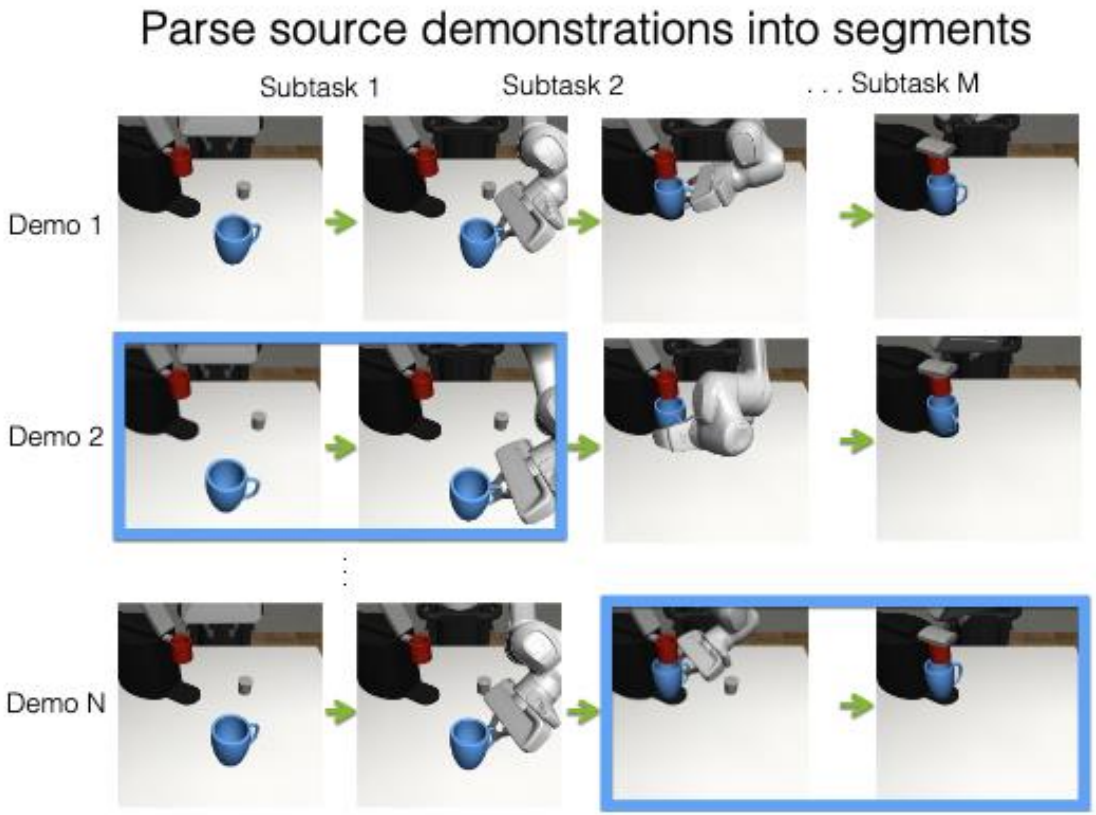
Solution 1: Collect More Demonstration Data by Augmentation

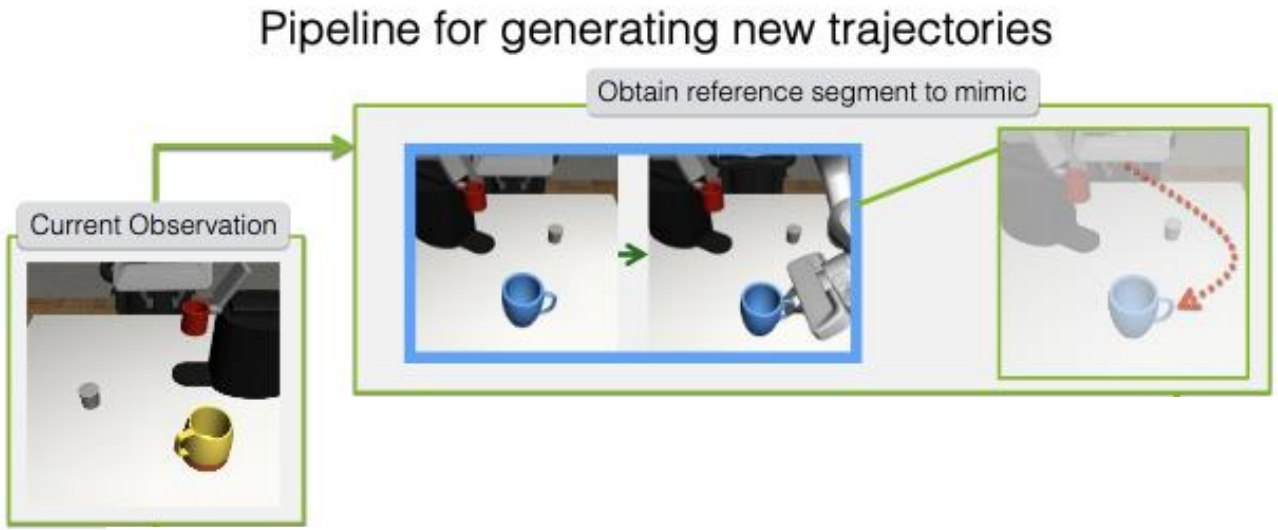


- Data augmentation based on geometrical priors:
  - ➢ Robot actions remain invariant to the object frame
- Pipeline overview:
  1. Sample a start state of a task
  2. Retrieve subtasks from demonstrations and adapt each subtask to the current scene
  3. Execute the synthetic demos. Add to training dataset if the execution succeed.

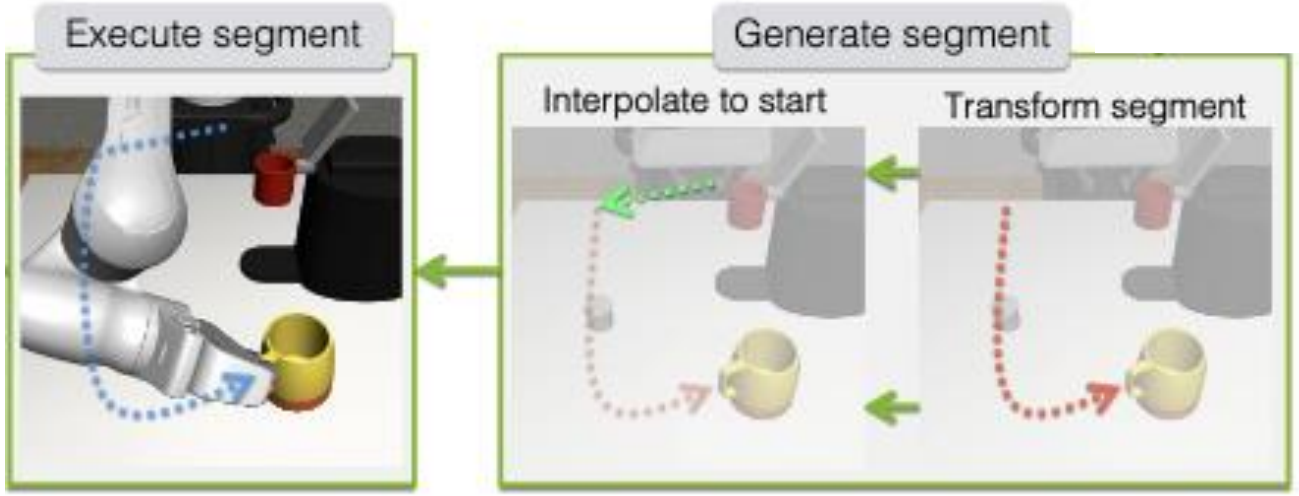
MimicGen: A Data Generation System for Scalable Robot Learning using Human Demonstrations. Mandlekar et al.

Step 1: Segment Demonstrations into Subtasks





Step 2: Adapt the Subtask Segment to the New Scene



- Let $\mathbf{T}_{\mathbf{W}}^{\mathbf{C_t}}$ denote the end-effector pose $\mathbf{C_t}$ with respect to world frame $\mathbf{W}$
- Let $\mathbf{T}_{\mathbf{W}}^{\mathbf{O}}$ denote the object pose $\mathbf{O}$ with respect to world frame $\mathbf{W}$
- Given a new scene with $\mathbf{T}_{\mathbf{W}}^{\mathbf{O}'}$, geometrical priors:

$$\mathbf{T}_{\mathbf{W}}^{\mathbf{C}_t'} = \mathbf{T}_{\mathbf{W}}^{\mathbf{O}} \left(\mathbf{T}_{\mathbf{W}}^{\mathbf{O}'}\right)^{-1} \mathbf{T}_{\mathbf{W}}^{\mathbf{C}_t}$$

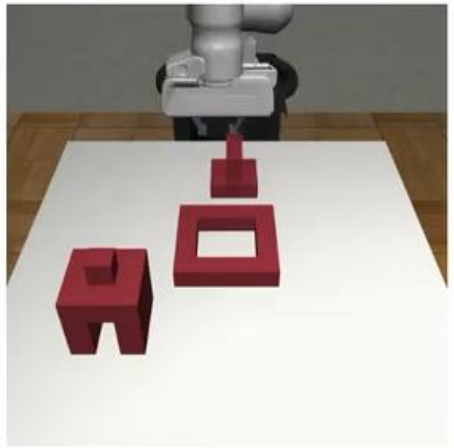
Why?

- $\mathbf{T}_{\mathbf{O}'}^{\mathbf{C}_t'} = \left(\mathbf{T}_{\mathbf{W}}^{\mathbf{O}'}\right)^{-1} \mathbf{T}_{\mathbf{W}}^{\mathbf{C}_t'}$
- $\mathbf{T}_{\mathbf{O}}^{\mathbf{C}_t} = \left(\mathbf{T}_{\mathbf{W}}^{\mathbf{O}}\right)^{-1} \mathbf{T}_{\mathbf{W}}^{\mathbf{C}_t}$

MimicGen: A Data Generation System for Scalable Robot Learning using Human Demonstrations. Mandlekar et al.

Solution 1: Collect More Demonstration Data by Augmentation

10 human demos



Human collects small number of teleoperated demonstrations

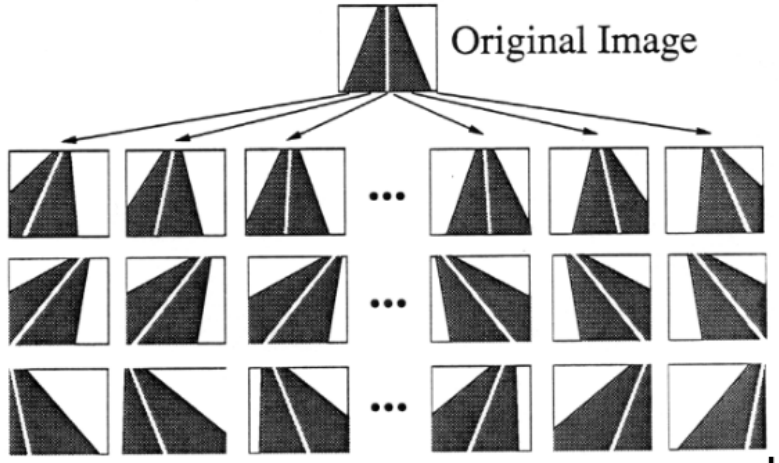
Problems: the policy still doesn't how to correct / recover from mistakes…
How to collect corrective data?

MimicGen: A Data Generation System for Scalable Robot Learning using Human Demonstrations. Mandlekar et al.

Solution 2: Collect Recovering Data by Augmentation

Simulating the Steering Errors

- Let humans drive as best they can.
- Increase training set variety by *artificially* shifting and rotating the video images, so that the vehicle appears at different orientations relative to the road.



- Generate 14 random shift/rotations for each image.
- A simple steering model is used to predict how a human driver would react to each transformation.

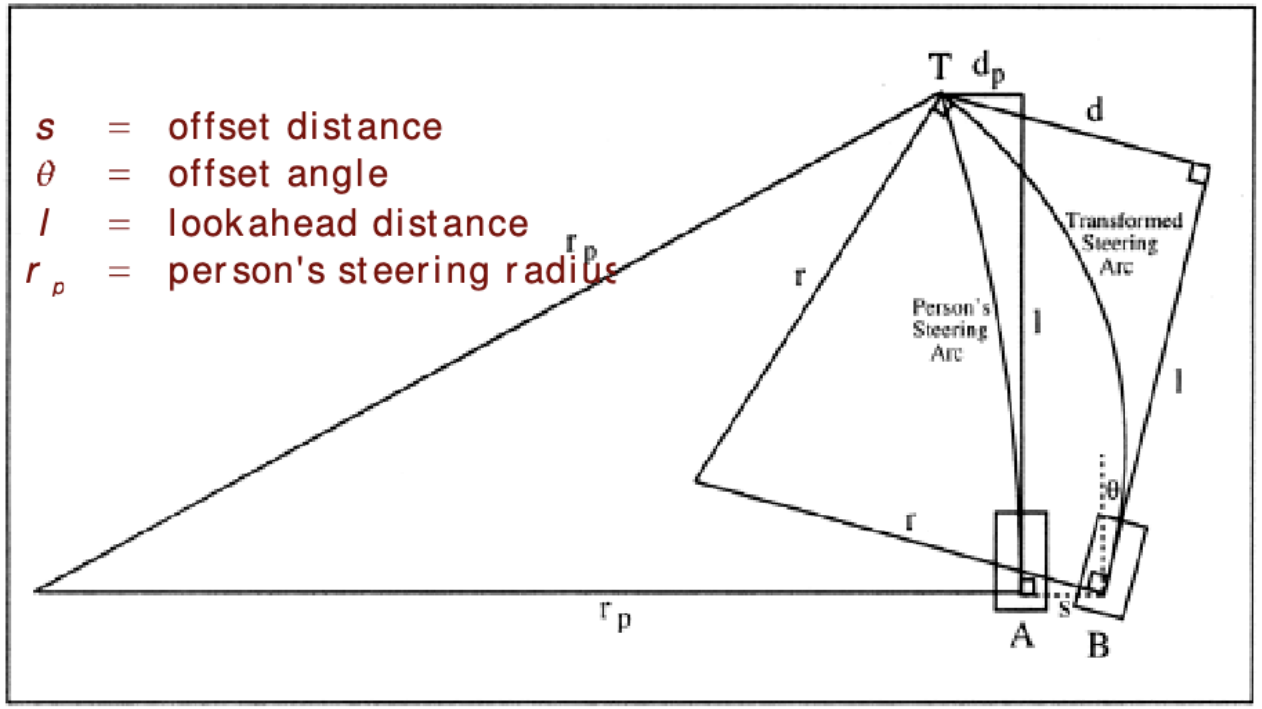
Interactive Systems Labs

Slide credit Alex Waibel

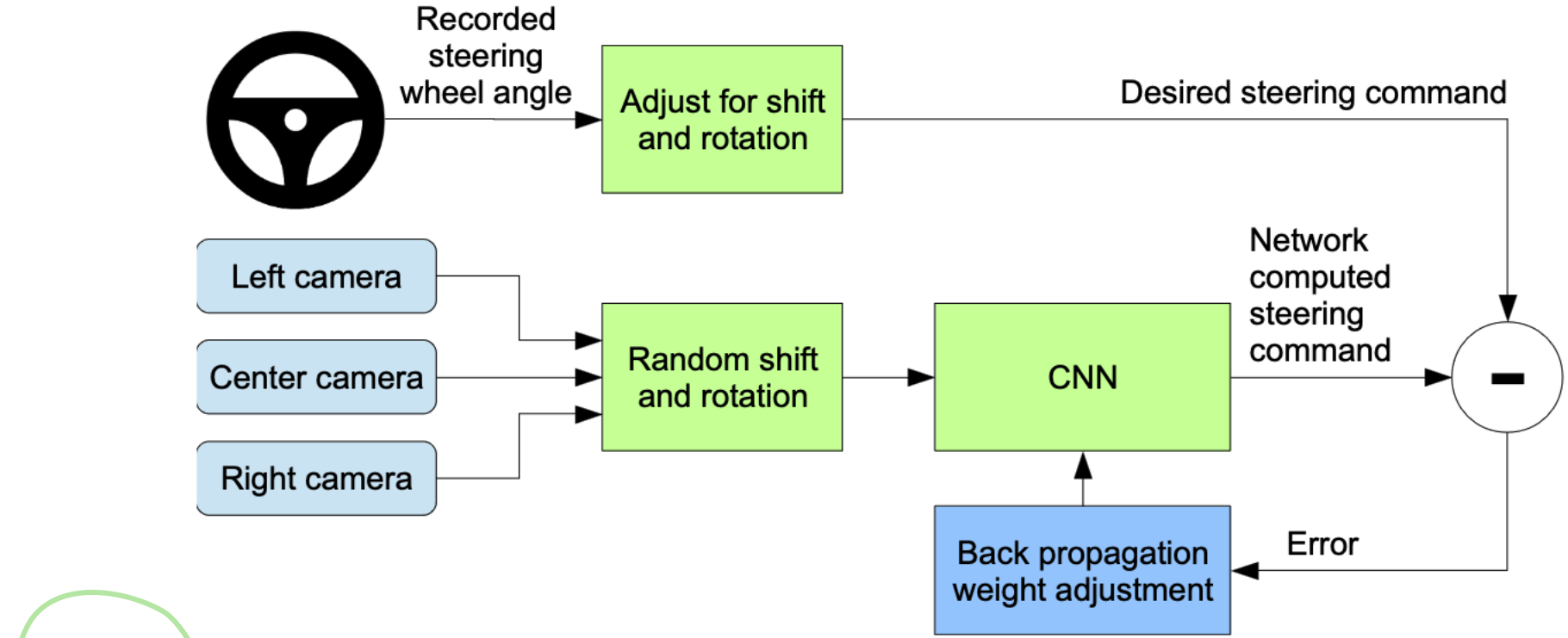
Solution 2: Collect Recovering Data by Augmentation

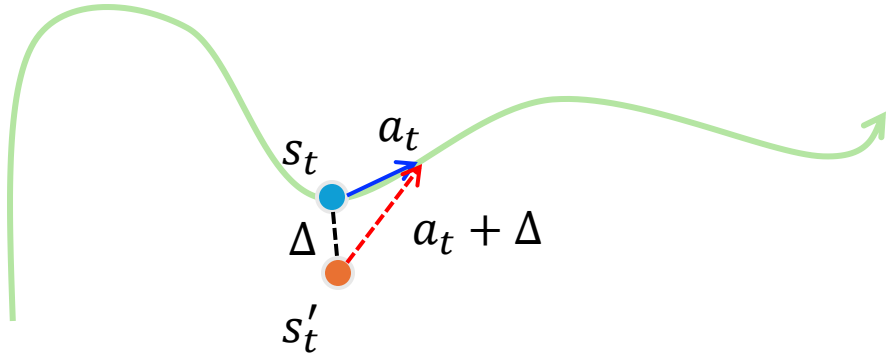
Estimating Correct Steering Direction

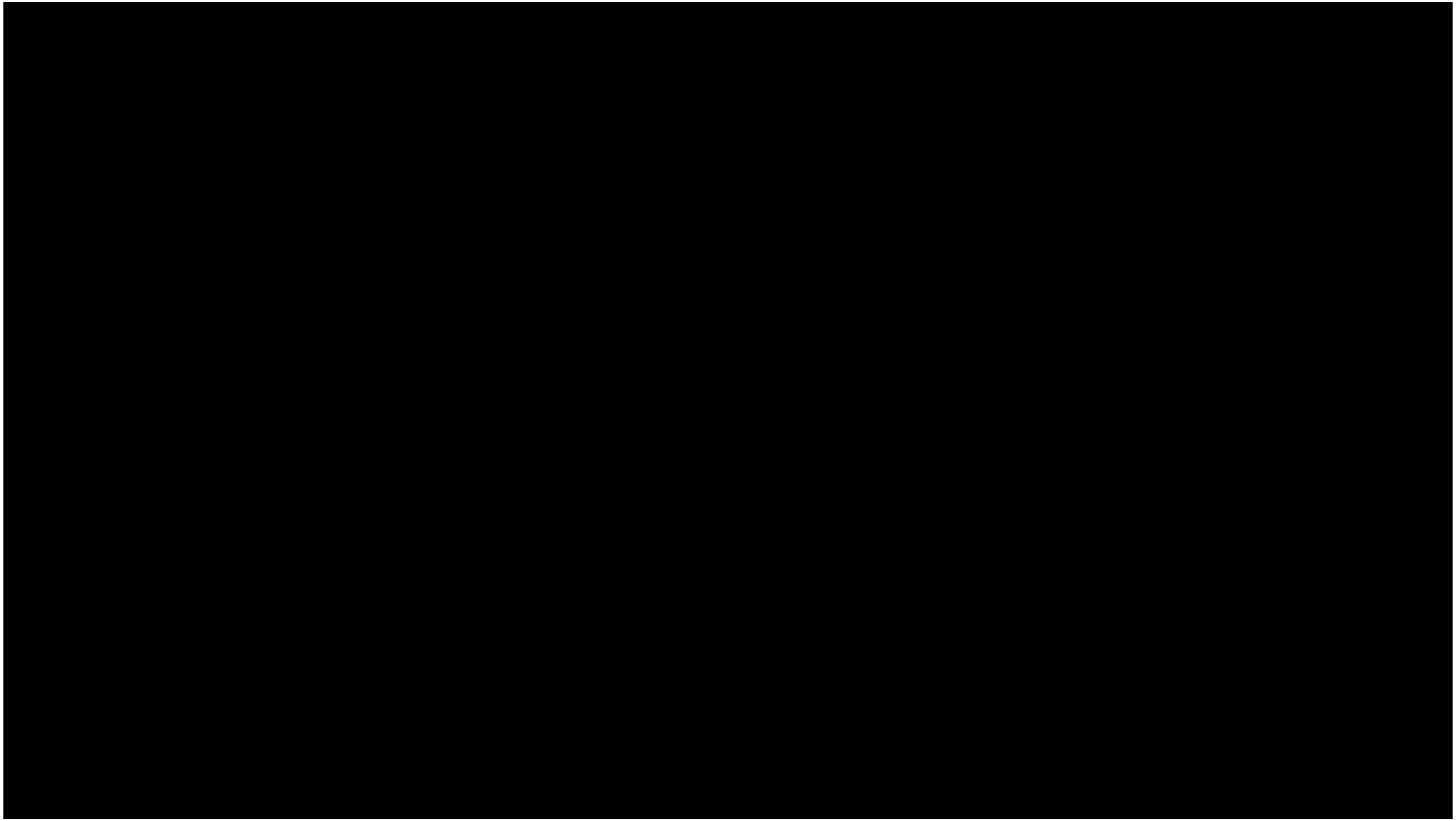
- "Pure pursuit" steering model generates a fairly good estimate of what the human driver would do.



Solution 2: Collect Recovering Data by Augmentation

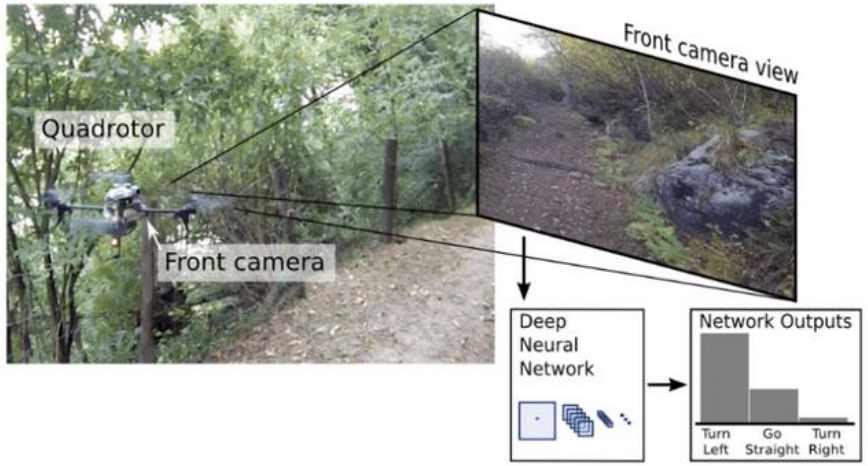


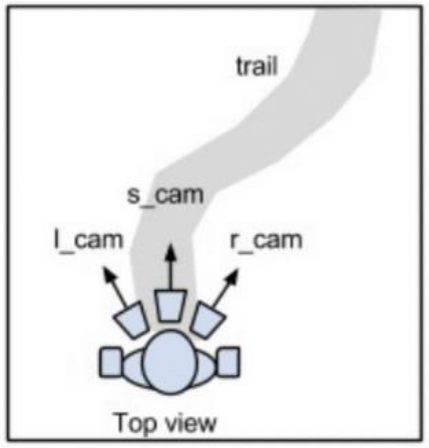




https://youtu.be/qhUvQiKec2U?si=Ko656GwtFFgDBwTZ

Solution 2: Collect Recovering Data by Augmentation



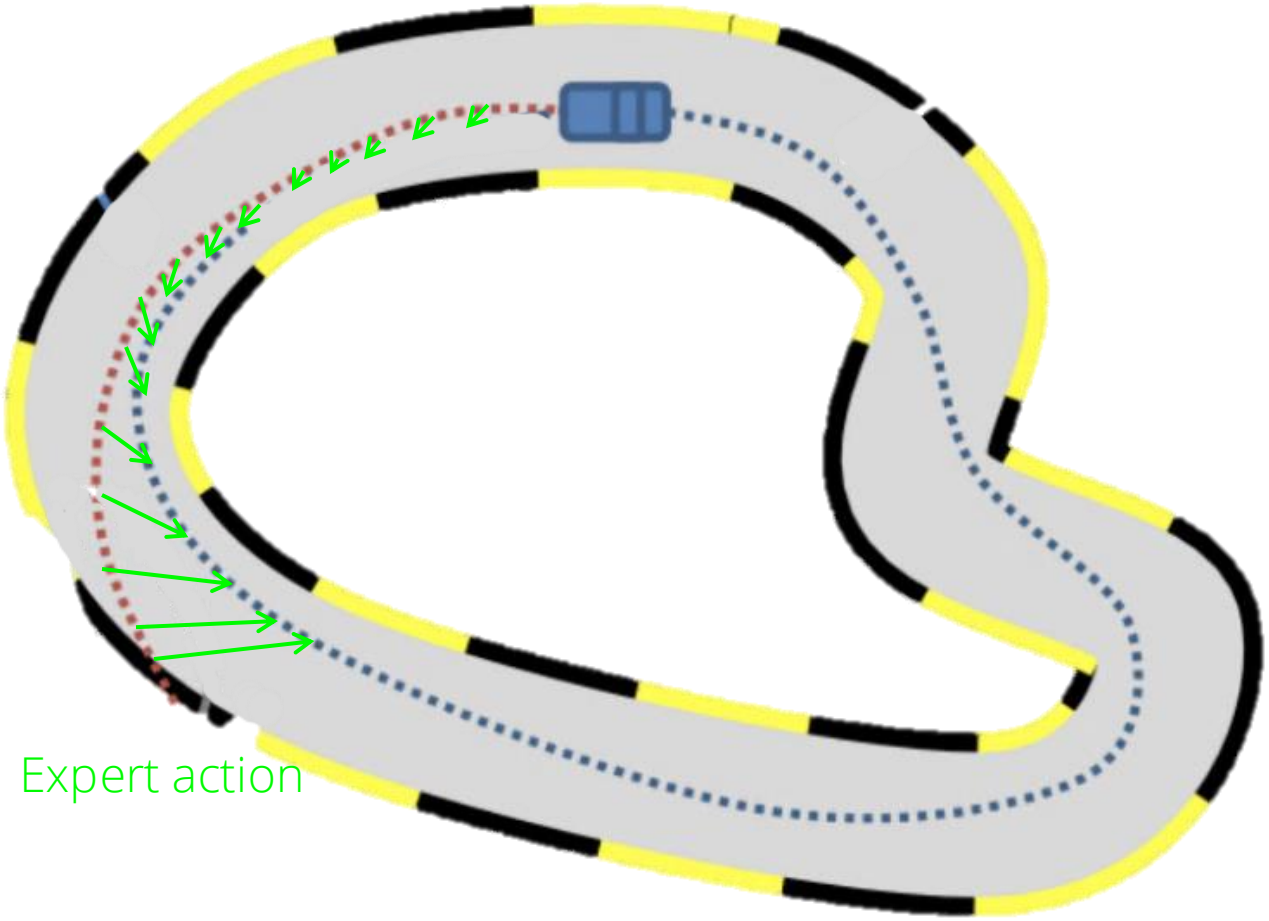


A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots. Giusti et al.

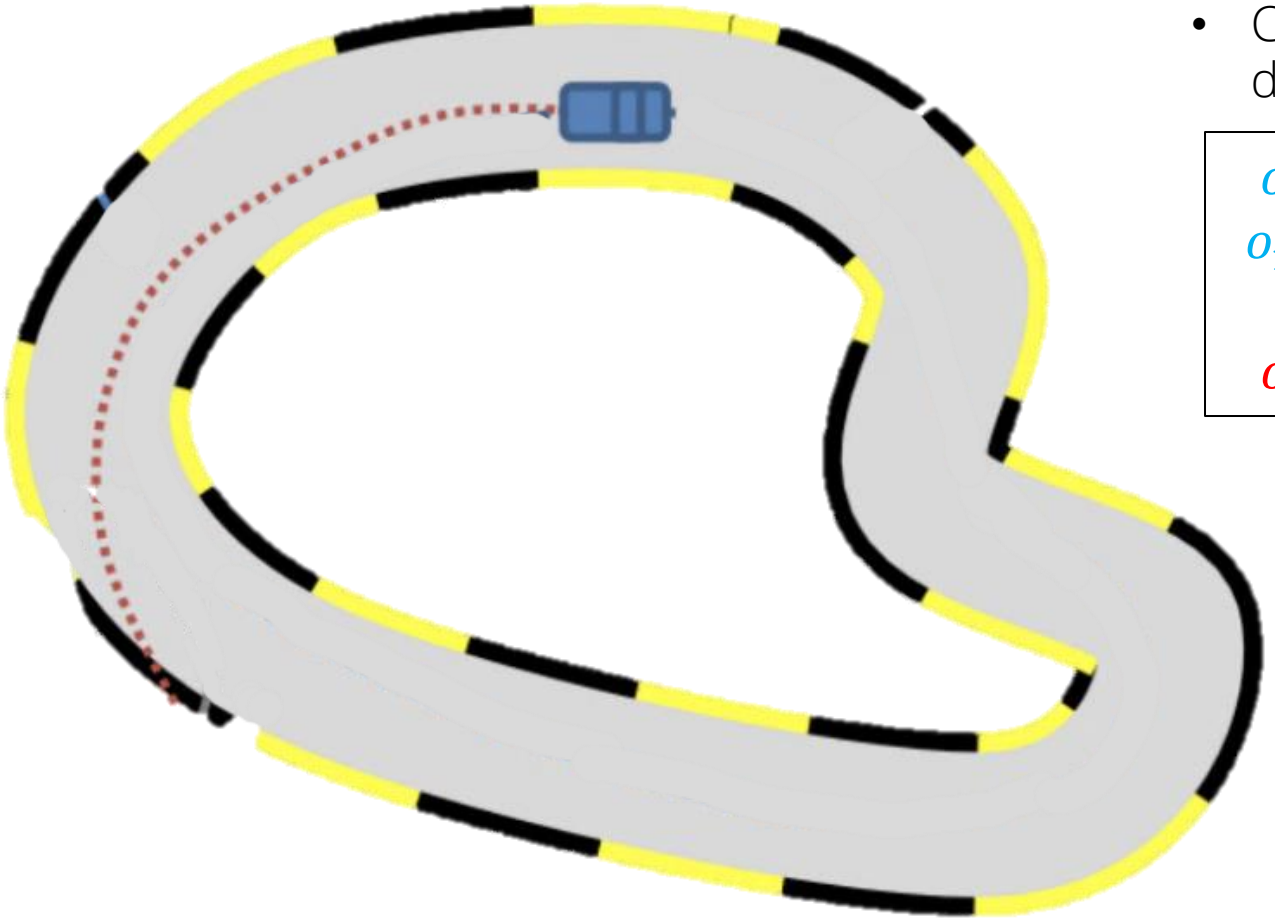
What are the Problems?

- Is geometric augmentation of the data always feasible?
  - How to recover from collision
  - How to correct erroneous manipulation of deformable objects
- Does geometric augmentation of the data suffice?

Solution 2: Collect Recovering Data with Expert in the Loop



Solution 2: Collect Recovering Data with Expert in the Loop

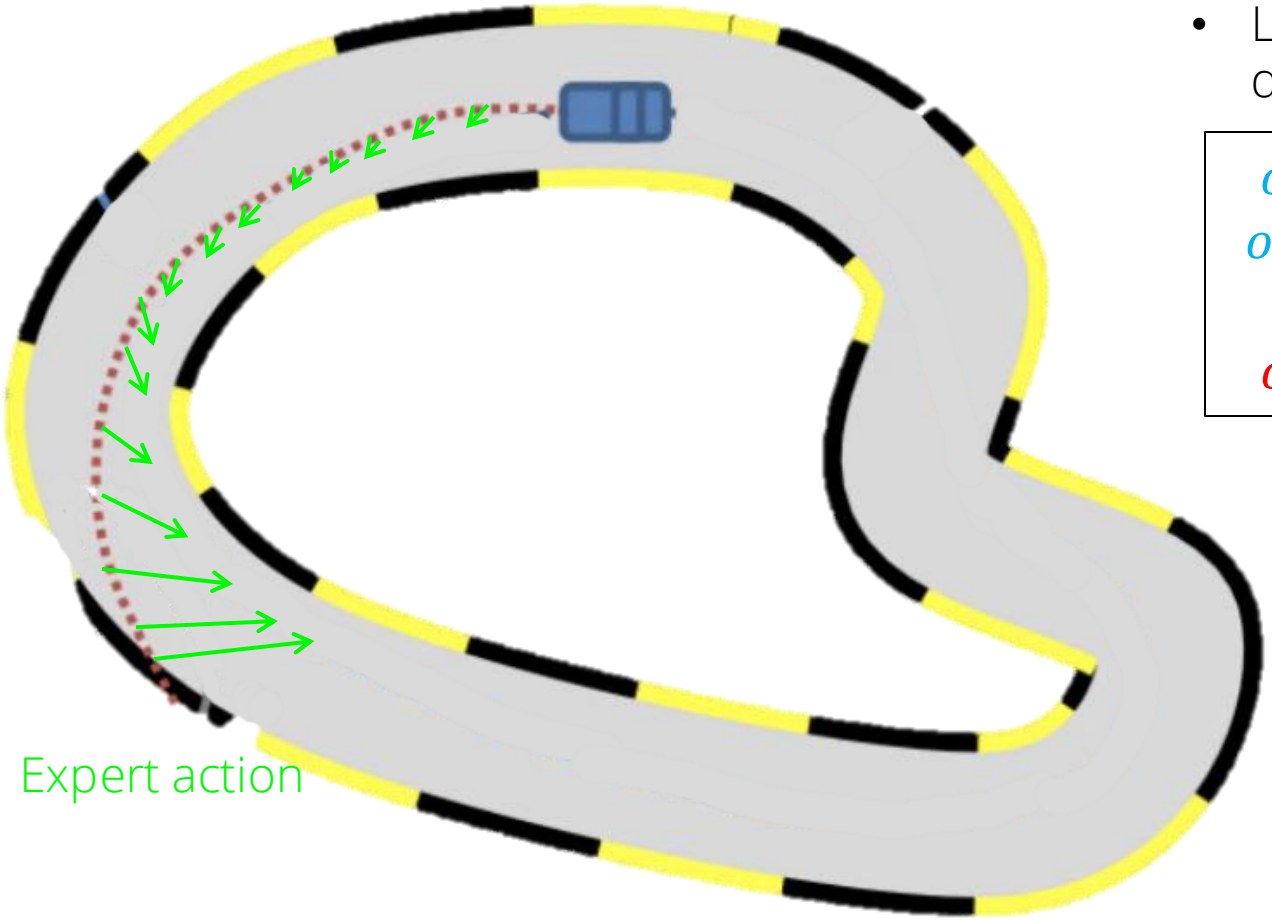


- Collect new experience by deploying trained policies:

$o_1^S, \hat{a}_1^S, o_2^S, \hat{a}_2^S, \ldots, o_T^S, \hat{a}_T^S$
$o_1^U, \hat{a}_1^U, o_2^U, \hat{a}_2^U, \ldots, o_T^U, \hat{a}_T^U$
...
$o_1^V, \hat{a}_1^V, o_2^V, \hat{a}_2^V \ldots, o_T^V, \hat{a}_T^V$

o_t^n: seen observation
a_t^n: seen action
o_t^n: unseen observation
$\hat{a}_t^n$: predicted action

Solution 2: Collect Recovering Data with Expert in the Loop



- Label new experience by querying experts:

$o_1^S, a_1^S, o_2^S, a_2^S, \ldots, o_T^S, a_T^S$
$o_1^U, a_1^U, o_2^U, a_2^U, \ldots, o_T^U, a_T^U$
...
$o_1^V, a_1^V, o_2^V, a_2^V \ldots, o_T^V, a_T^V$

o_t^n: seen observation
a_t^n: seen action
o_t^n: unseen observation
a_t^n: expert action

DAGGER

Initialize $\mathcal{D} \leftarrow \emptyset$.
Initialize $\hat{\pi}_1$ to any policy in Π.
for $i = 1$ **to** N **do**
  Let $\pi_i = \beta_i \pi^* + (1 - \beta_i)\hat{\pi}_i$.
  Sample T-step trajectories using π_i.
  Get dataset $\mathcal{D}_i = \{(s, \pi^*(s))\}$ of visited states by π_i and actions given by expert.
  Aggregate datasets: $\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{D}_i$.
  Train classifier $\hat{\pi}_{i+1}$ on $\mathcal{D}$.
end for
Return best $\hat{\pi}_i$ on validation.

Algorithm 3.1: DAGGER Algorithm.

New dataset:

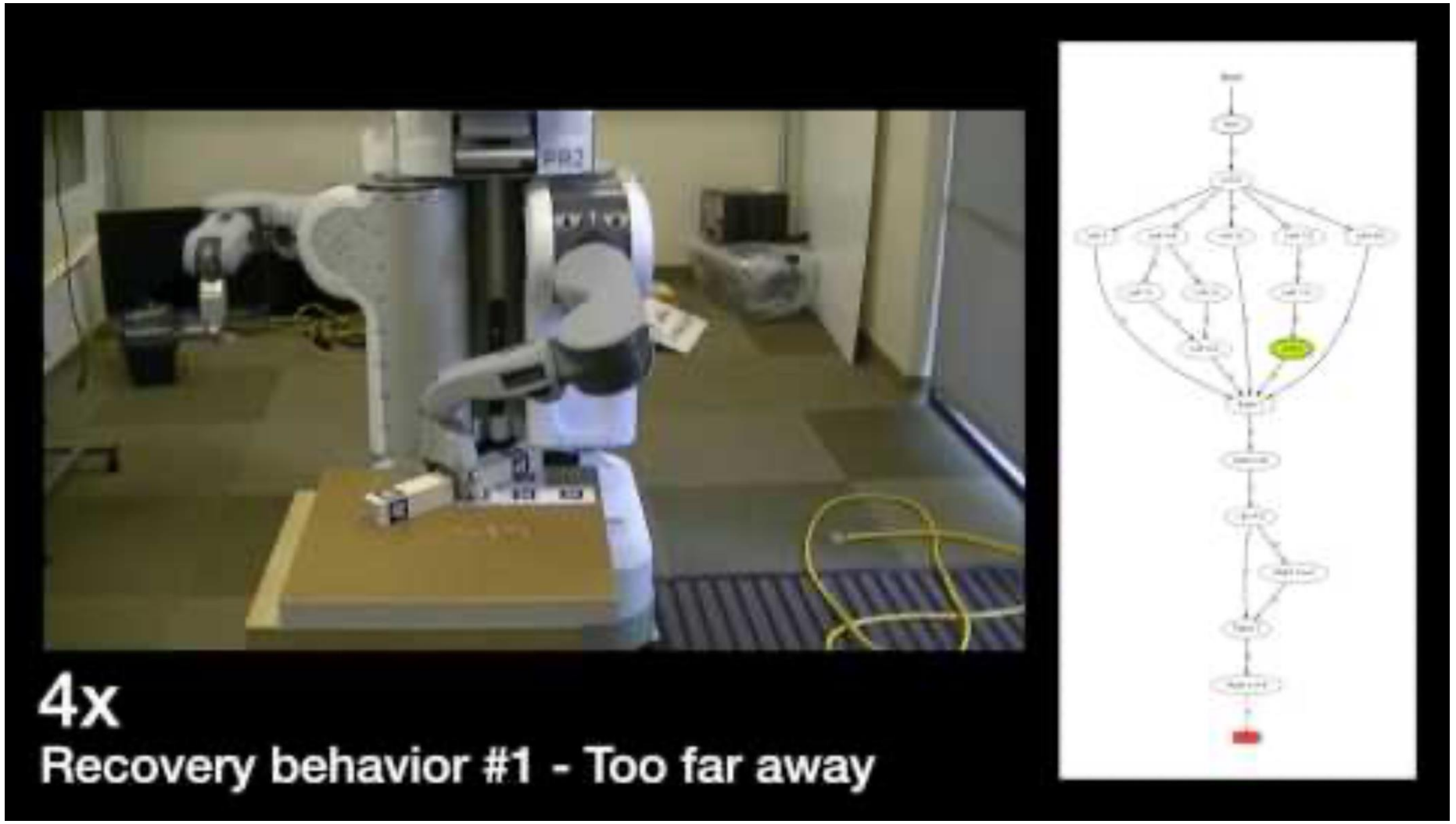
$o_1^1, a_1^1, o_2^1, a_2^1, \ldots, o_T^1, a_T^1$
$o_1^2, a_1^2, o_2^2, a_2^2, \ldots, o_T^2, a_T^2$
...
$o_1^N, a_1^N, o_2^N, a_2^N, \ldots, o_T^N, a_T^N$

+

$o_1^S, a_1^S, o_2^S, a_2^S, \ldots, o_T^S, a_T^S$
$o_1^U, a_1^U, o_2^U, a_2^U, \ldots, o_T^U, a_T^U$
...
$o_1^V, a_1^V, o_2^V, a_2^V \ldots, o_T^V, a_T^V$

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

DAGGER

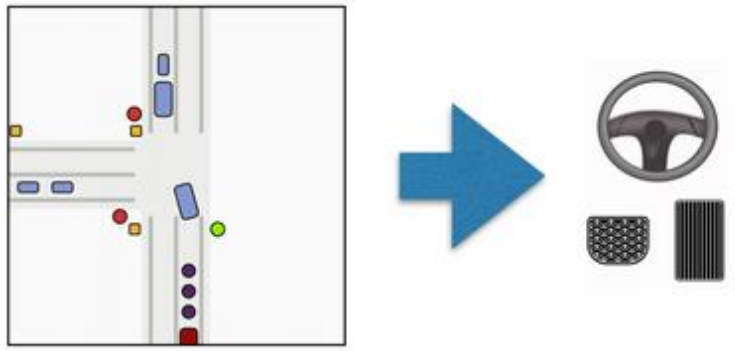


Slide credit S. Niekum

Experts can be Non-Human but Privileged Learned Policies

DAGGER: from a privileged teaching agent to an agent that drives from images.

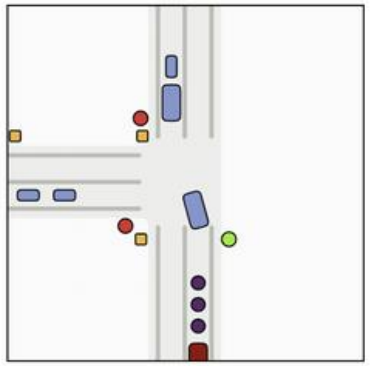
privileged agent

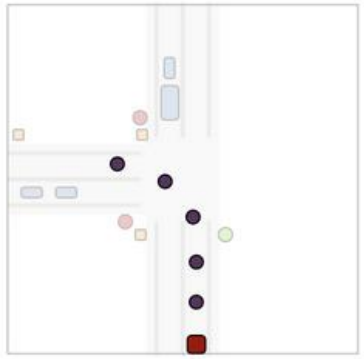


sensorimotor agent

Learning by Cheating, Chen et al., 2019

Privileged Agent vs. Sensorimotor Agent





Privileged Agent:

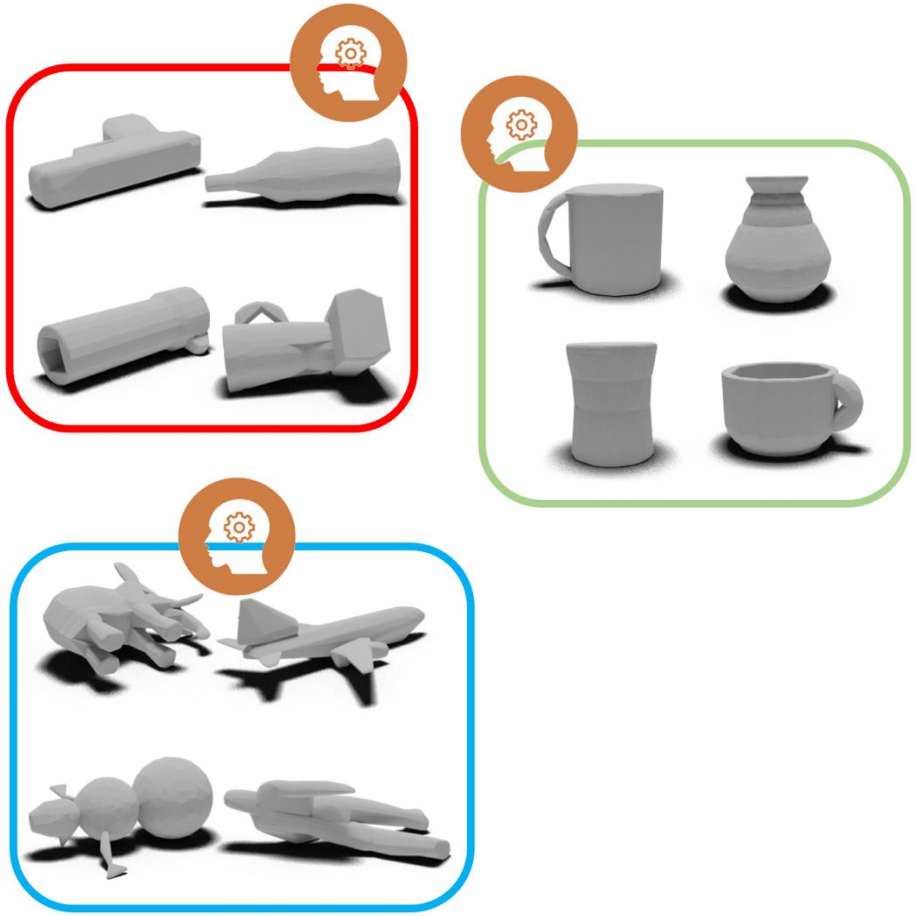
- Fully (perfectly) observes the world state
- Predicts high-level control (e.g. future waypoints for the car to follow)

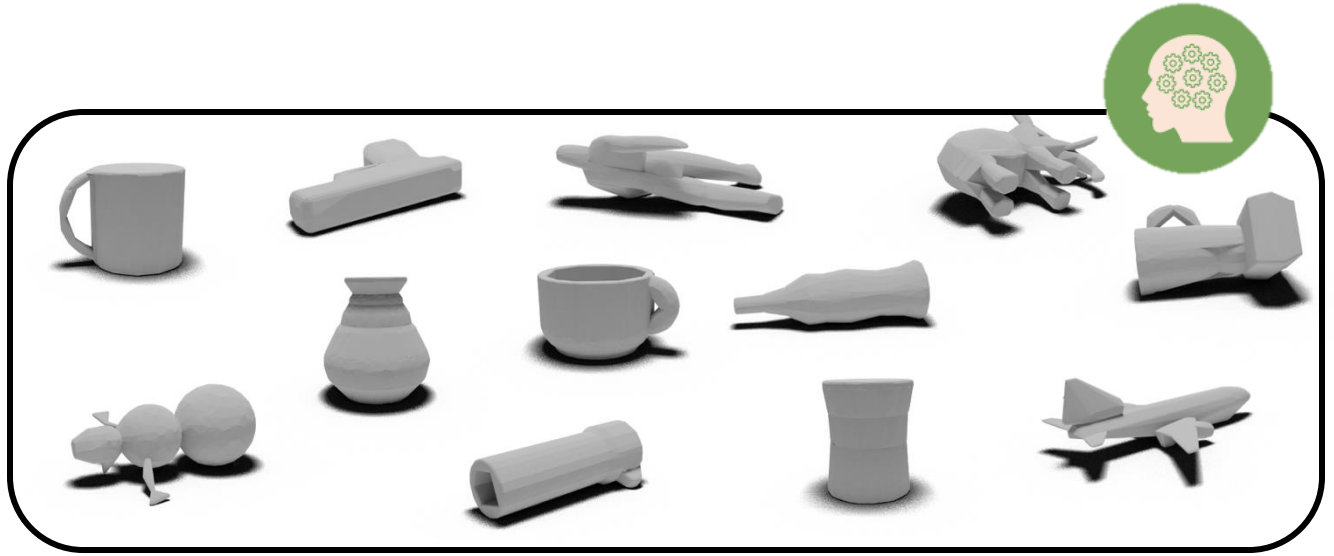
Sensorimotor Agent:

- Partially (imperfectly) observes the world state
- Predicts low-level control (e.g. steering command)

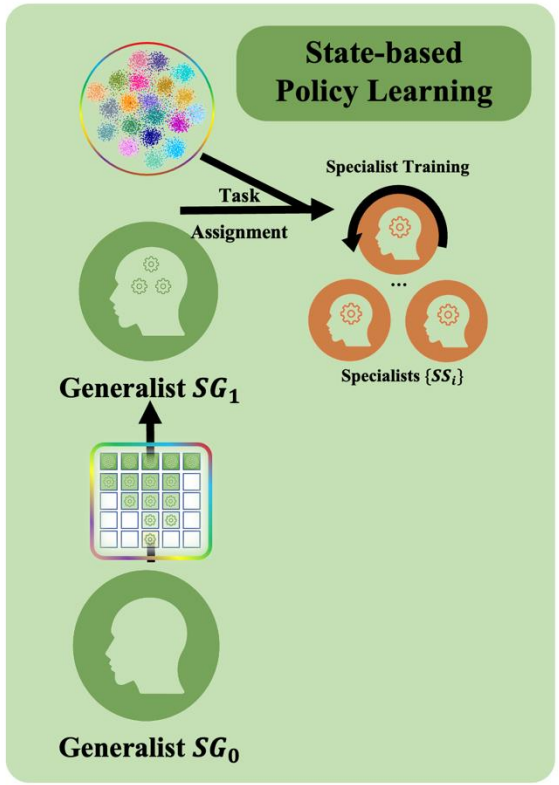
Slide credit K. Fragkiadaki

Experts can be Non-Human but Specialist Policies



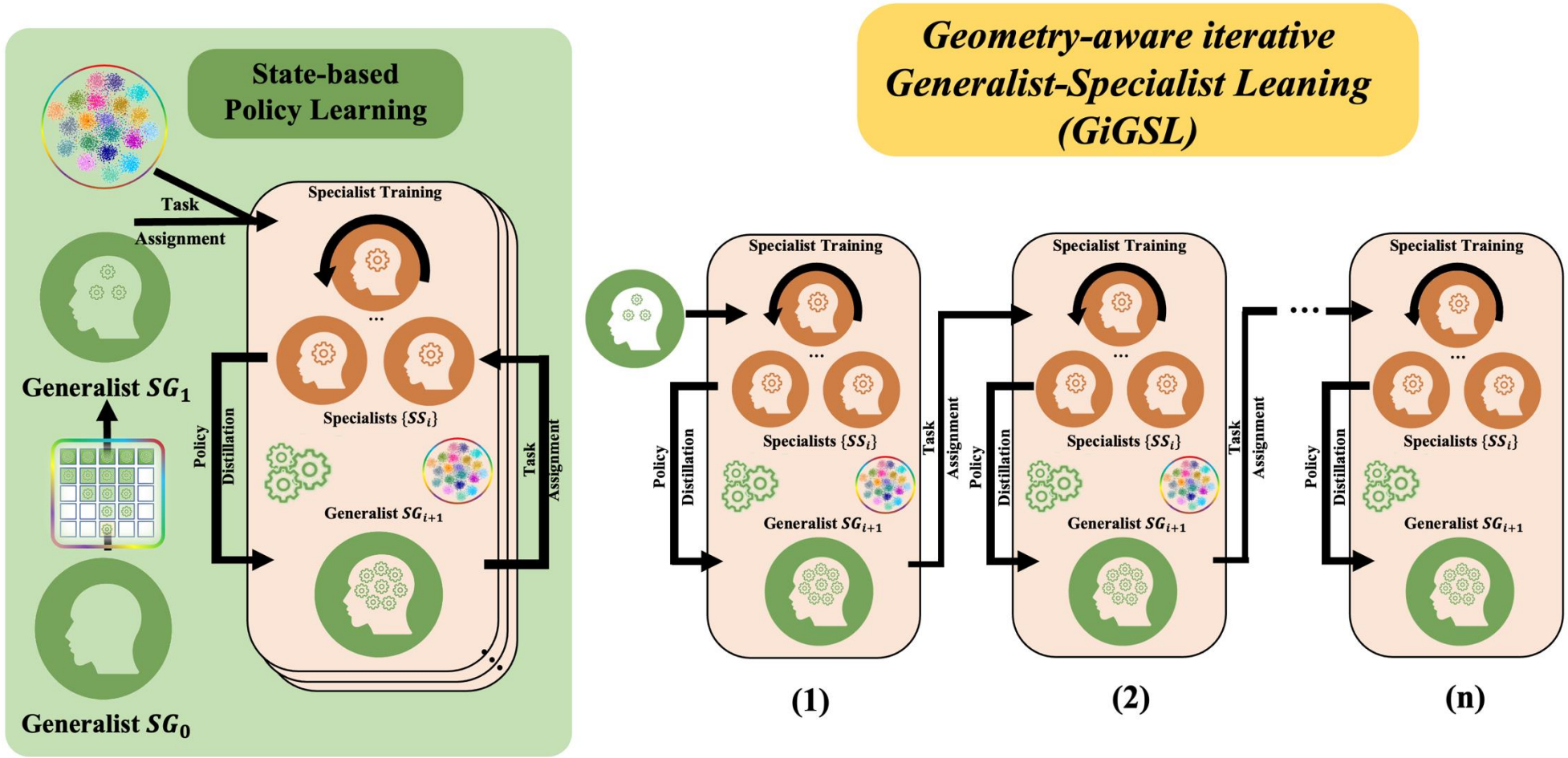


Bootstrapping from Specialist Agents Facilitates Large-scale Policy Learning

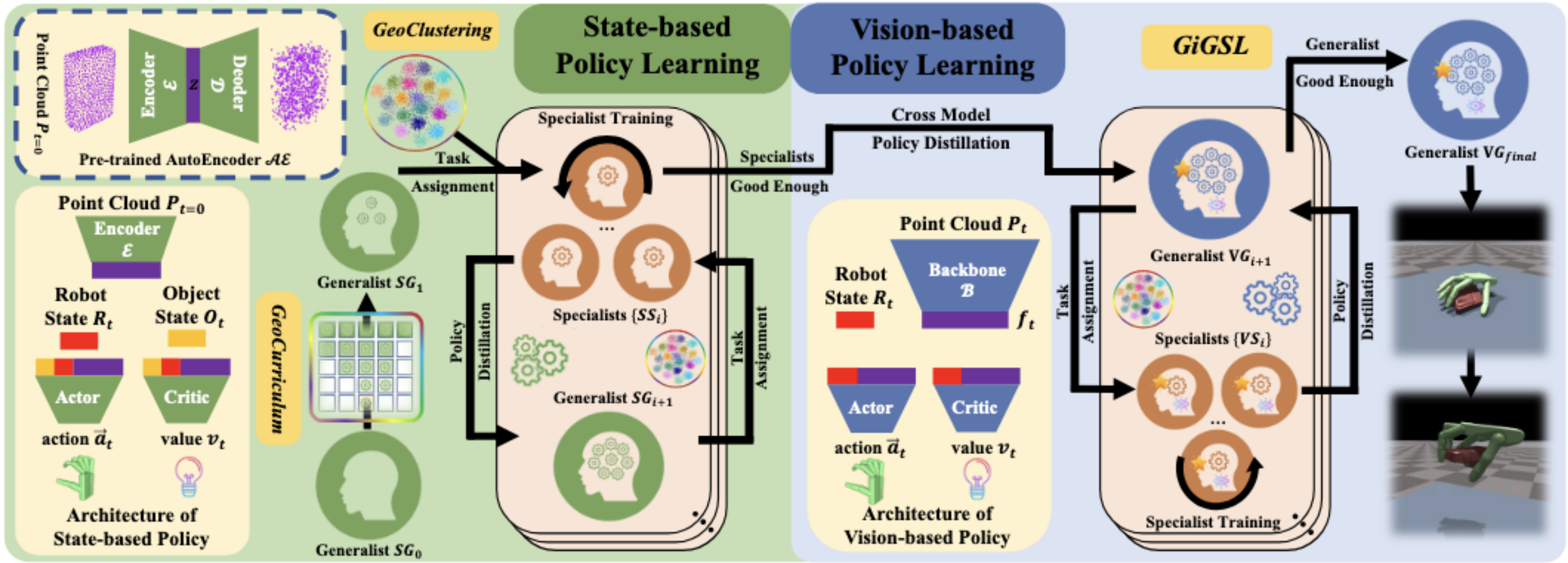


UniDexGrasp++: Improving Dexterous Grasping Policy Learning via Geometry-aware Curriculum and Iterative Generalist Specialist Learning. Wan et al.

Bootstrapping from Specialist Agents Facilitates Large-scale Policy Learning



Bootstrap Policy Learning from Specialist Agents and Privileged Agents



Is Behavior Cloning + DAGGER All You Need?

- Of course, No!

- More problems:
  - **Causal confusion**. Does your model look at the correct feature so that it generalizes to novel scene?

Causal Confusion: Look at Wrong Features

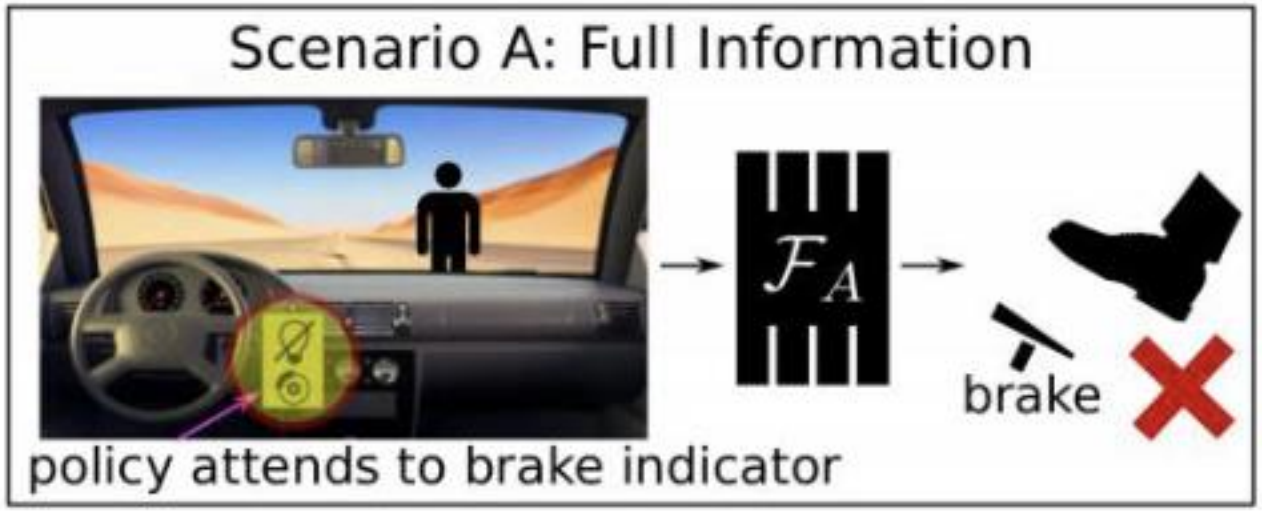


Image credit S. Levine

Causal Confusion: Look at Wrong Features

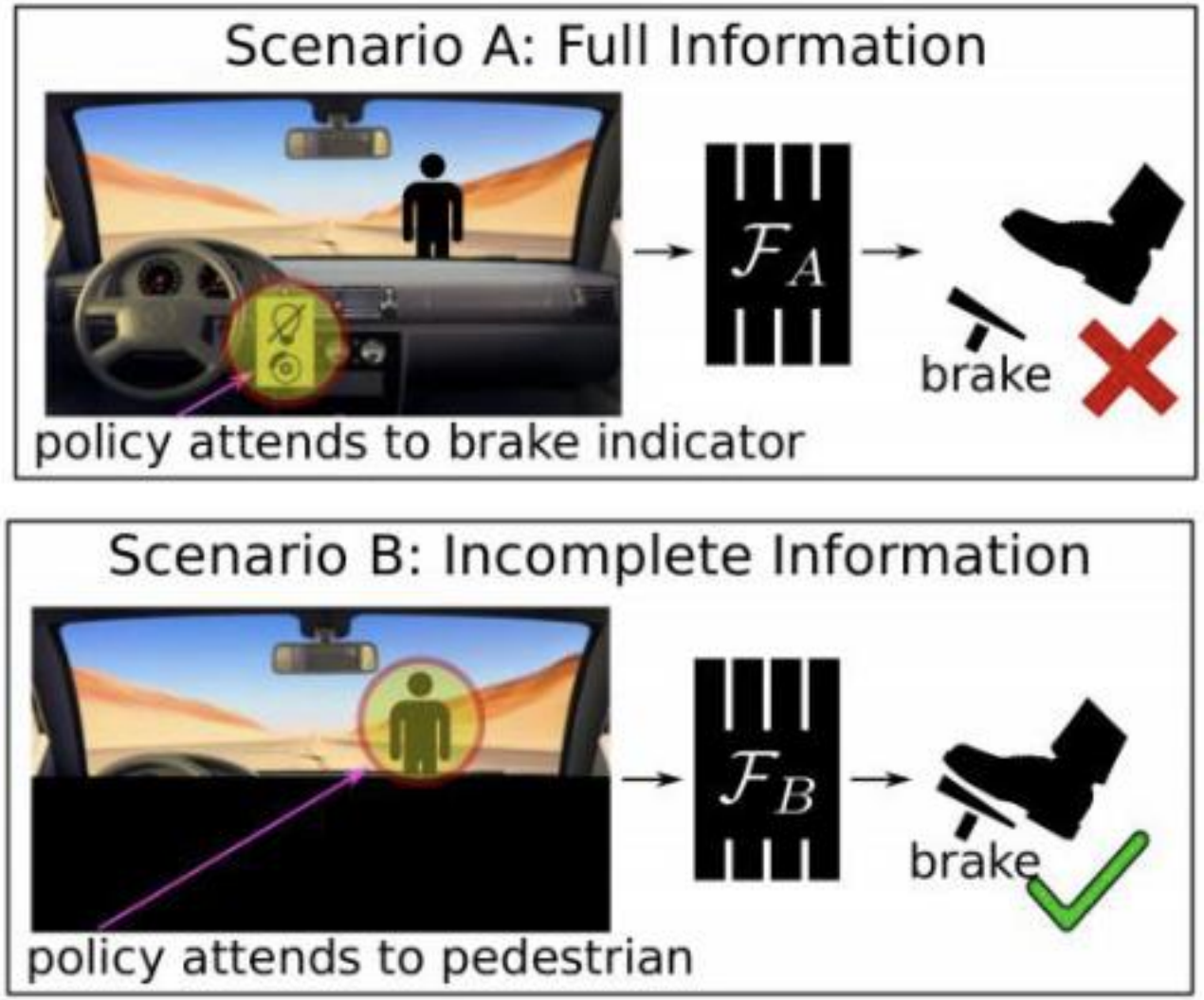


Image credit S. Levine

Is Behavior Cloning + DAGGER All You Need?

- Of course, No!

- More problems:
  - **Causal confusion**. Does your model look at the correct feature so that it generalizes to novel scene?
  - **Multi-modality behaviors**. What if there are multiple ways to achieve a goal? What is the better formulation of the model output?

Multi-modality Behaviors: The Same Task can be Done in Different Ways

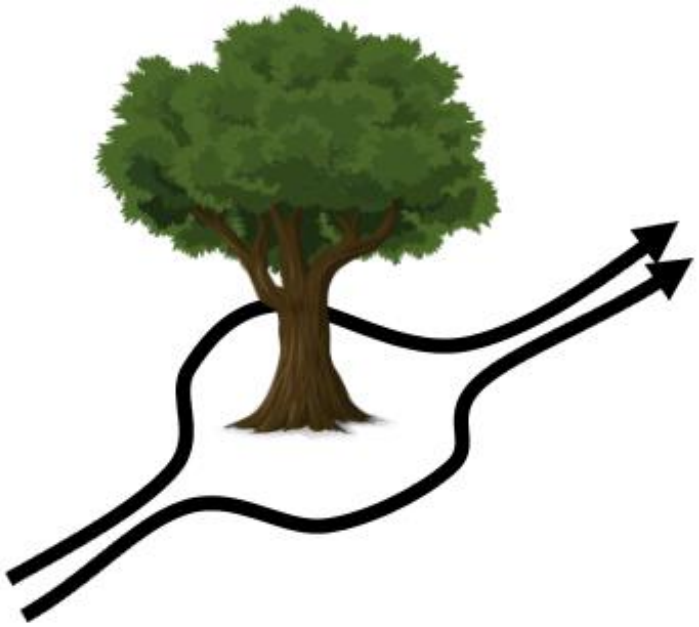
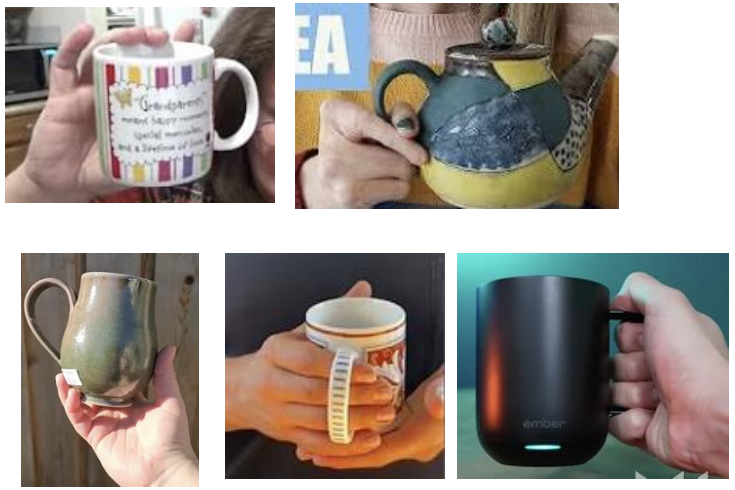


Image credit S. Levine]

Diffusion policy: Visuomotor policy learning via action diffusion

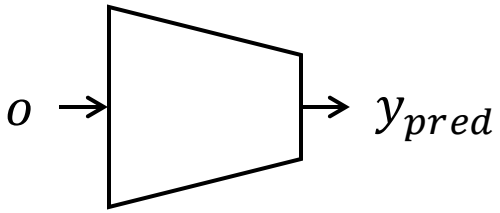


How to model multi-modal behaviors?

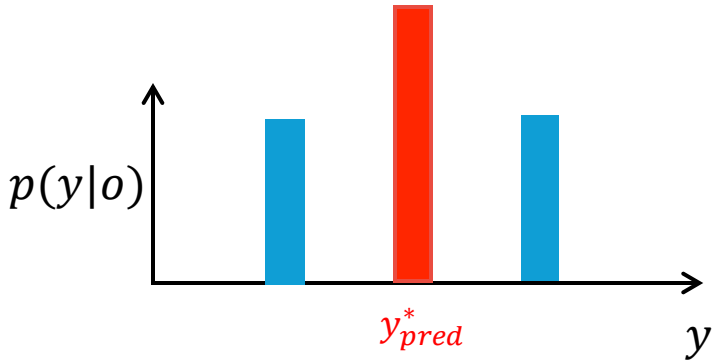
Deterministic Regression Approximates Action Average

Deterministic

- predicts the same output given the same input



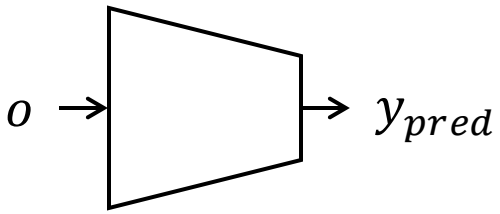
$$y^*_{pred} = \underset{y_{pred}}{\text{argmin}} \frac{1}{N} \Sigma_{n=1}^{N} |y_{pred} - y_n|^2$$



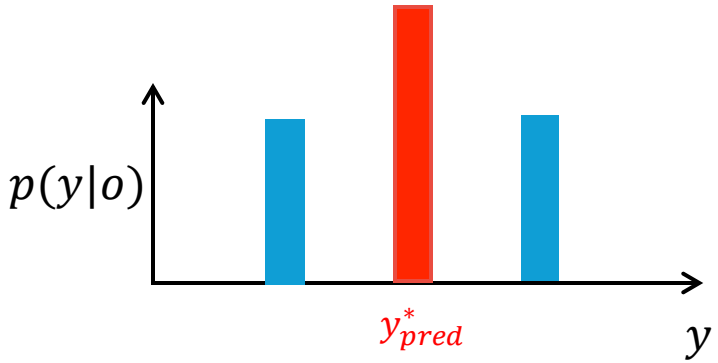
Probabilistic Regression Captures Distribution of Action

Deterministic

- predicts the same output given the same input

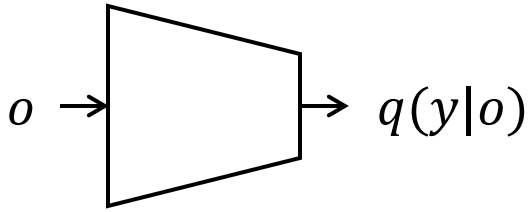


$$y^*_{pred} = \underset{y_{pred}}{\text{argmin}} \frac{1}{N} \Sigma_{n=1}^{N} |y_{pred} - y_n|^2$$

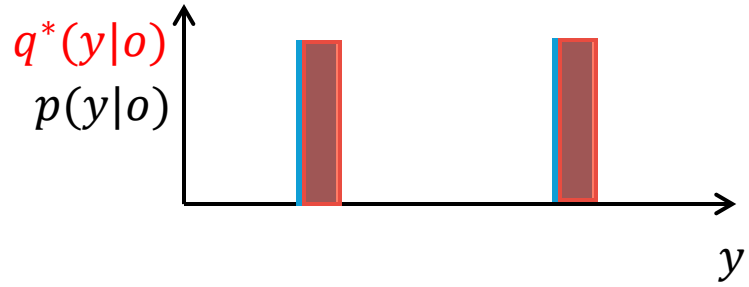


Probabilistic

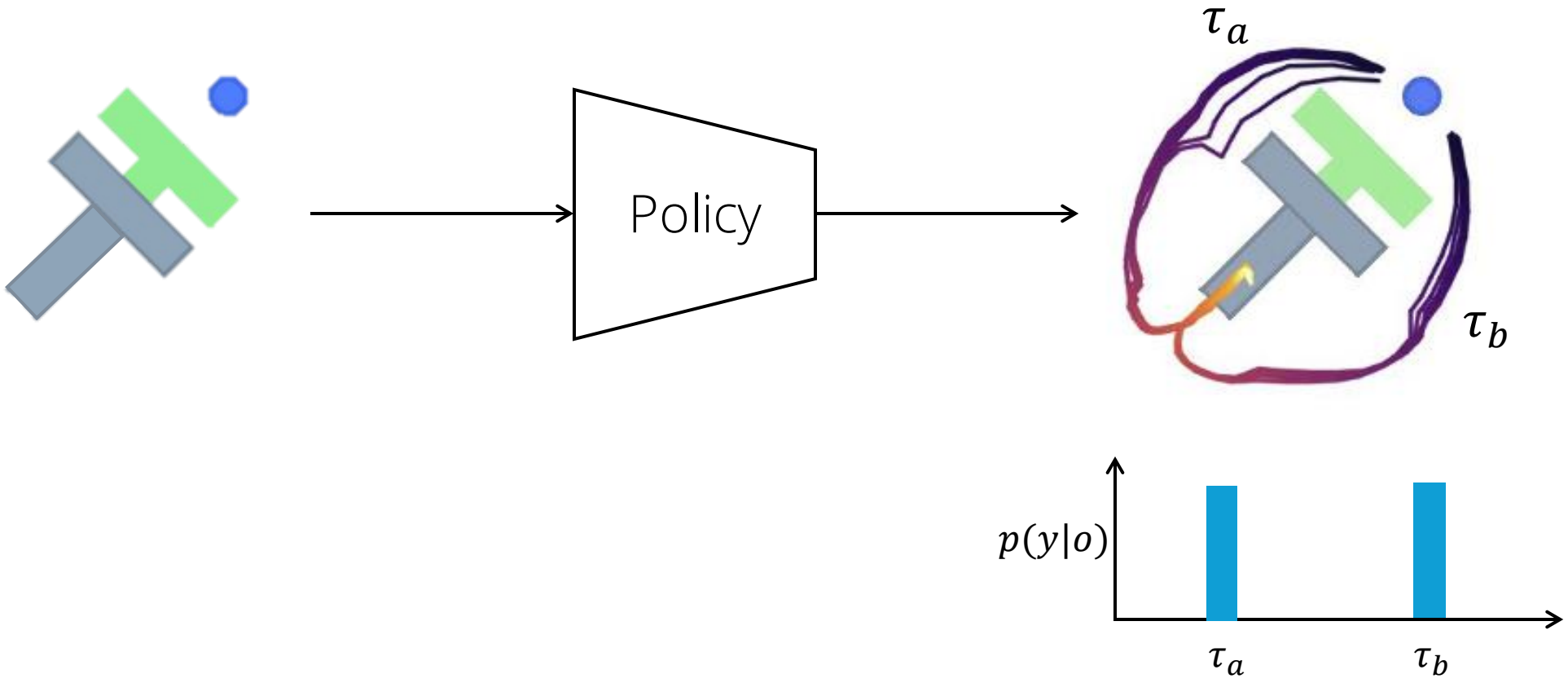
- predicts the same likelihood of all output given the same input



$$q^*(y|o) = \underset{q^*}{\text{argmin}}\, D_{KL}(q(y|o), p(y|o))$$



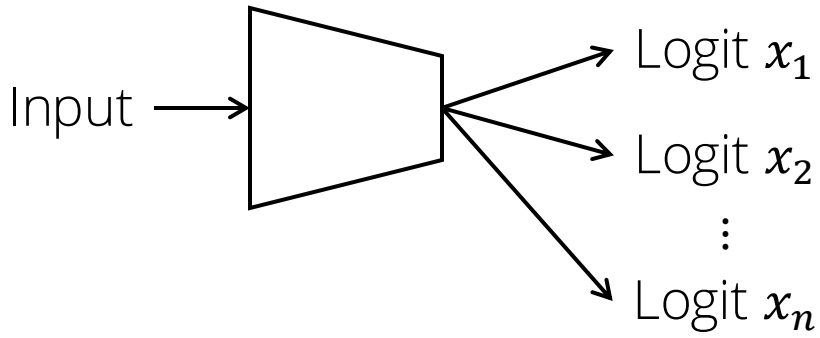
Model Multi-modality Behaviors with Probabilistic Models

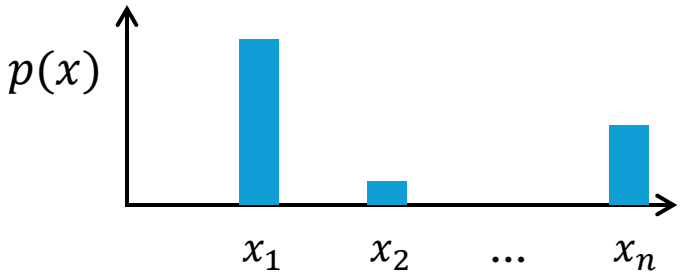


Choices of Probabilistic Generative Models

Categorical

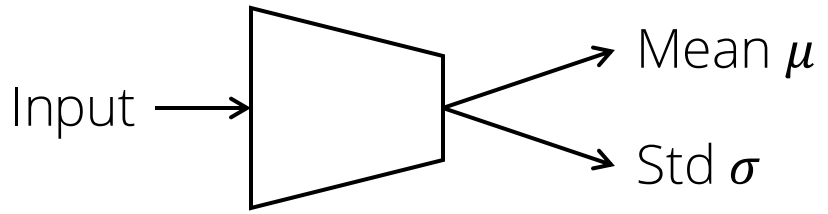
- Predict the categorical probability of sample "x"
- Most LLMs etc

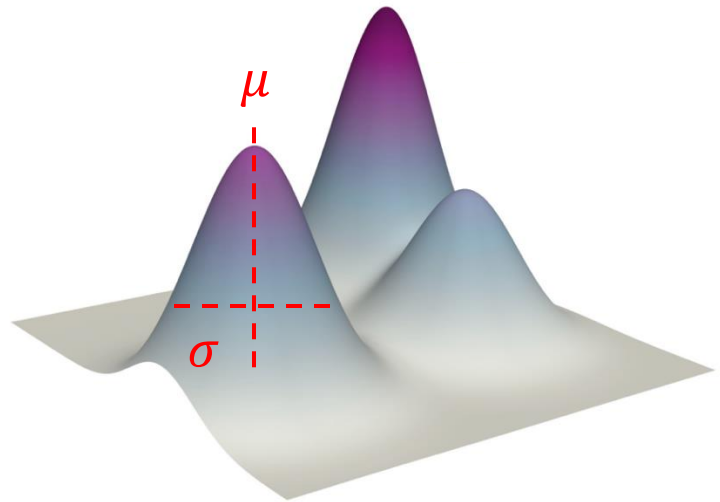




Gaussian

- Predict the mean and standard deviation to model the probability of sample "x"
- VAE, Diffusion models etc

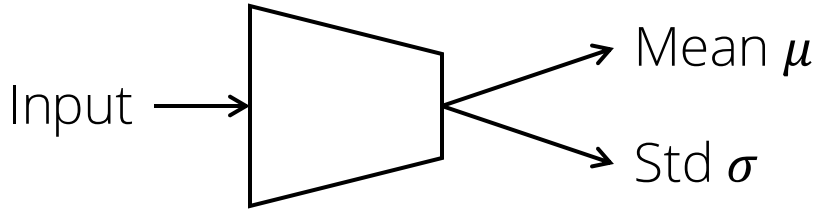


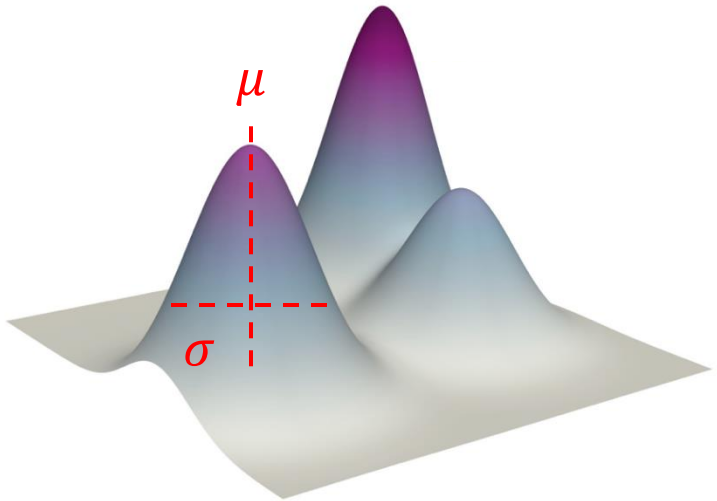


Choices of Probabilistic Generative Models

Gaussian

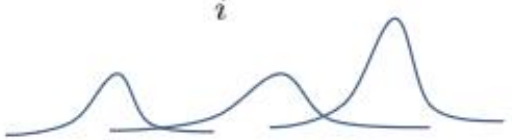
- Predict the mean and standard deviation to model the probability of sample "x"
- VAE, Diffusion models etc





Mixture of Gaussians

$$\pi(\mathbf{a}|\mathbf{o}) = \sum_i w_i \mathcal{N}(\mu_i, \Sigma_i)$$



$$w_1, \mu_1, \Sigma_1, \ldots, w_N, \mu_N, \sigma_N$$

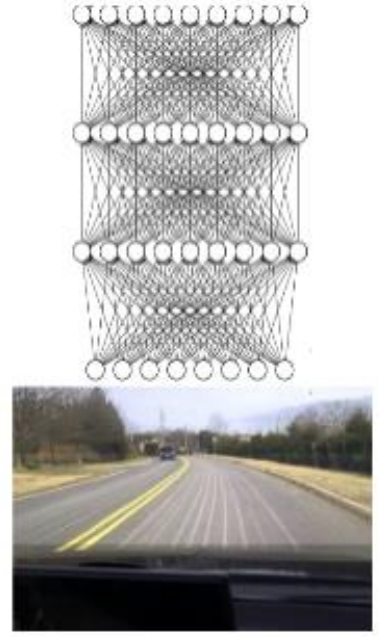
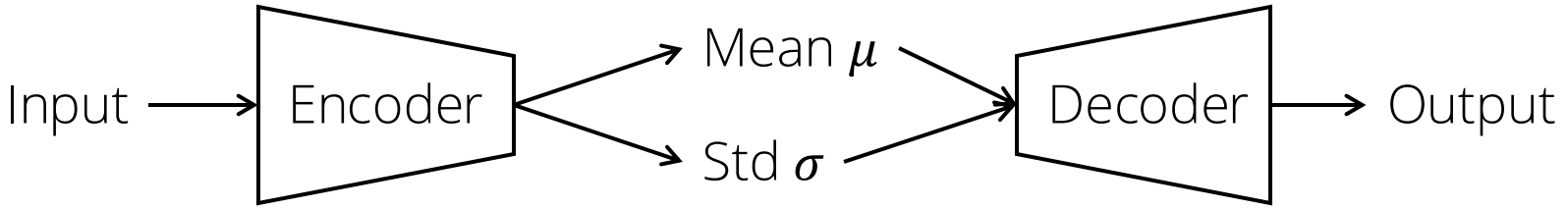


Image credit S. Levine

Example: Variational Auto-Encoder



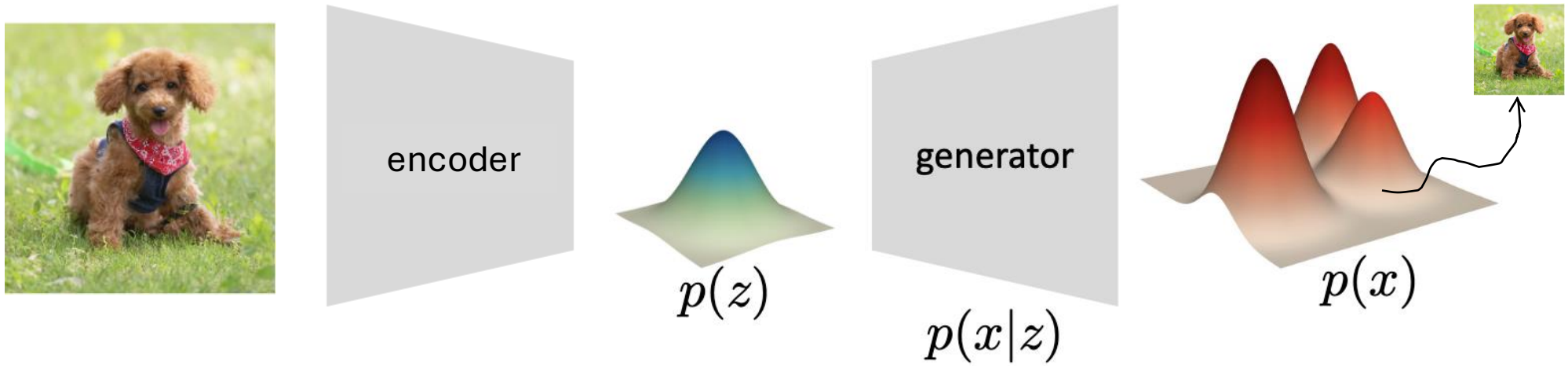


Image source: Kaiming He

Example: Denoising Diffusion Probabilistic Models

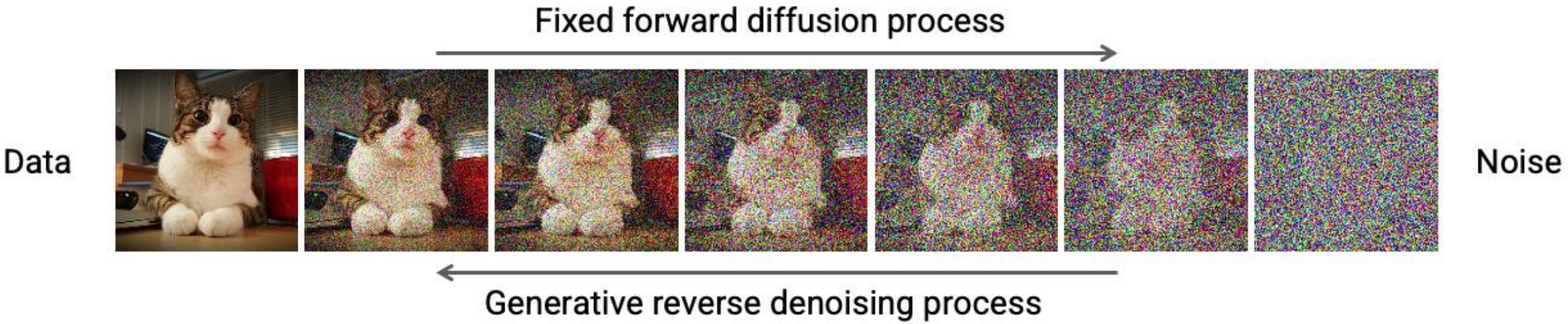
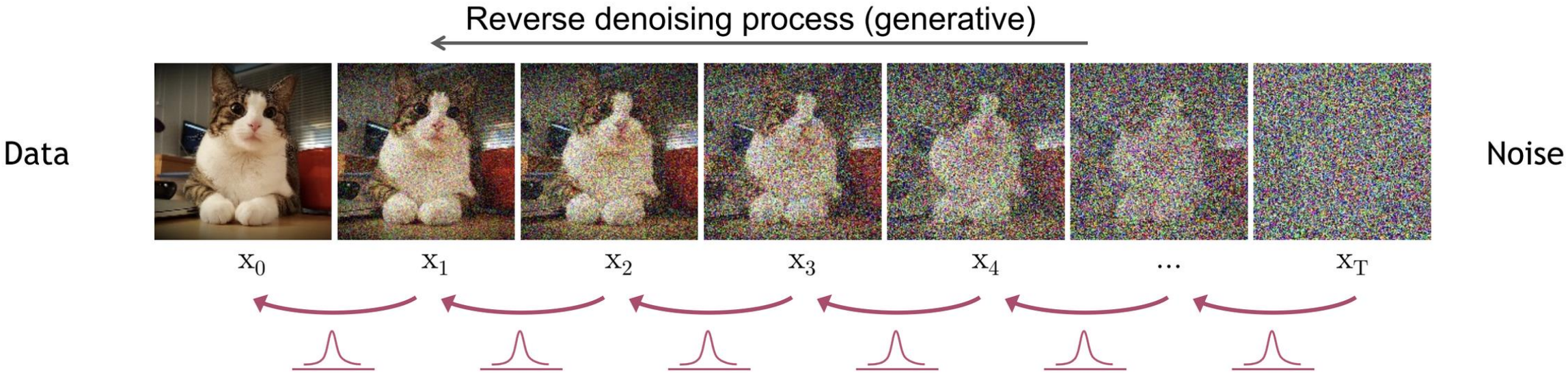


Image source: https://cvpr2023-tutorial-diffusion-models.github.io/

Denoising Diffusion Probabilistic Models (DDPM)



$$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$$

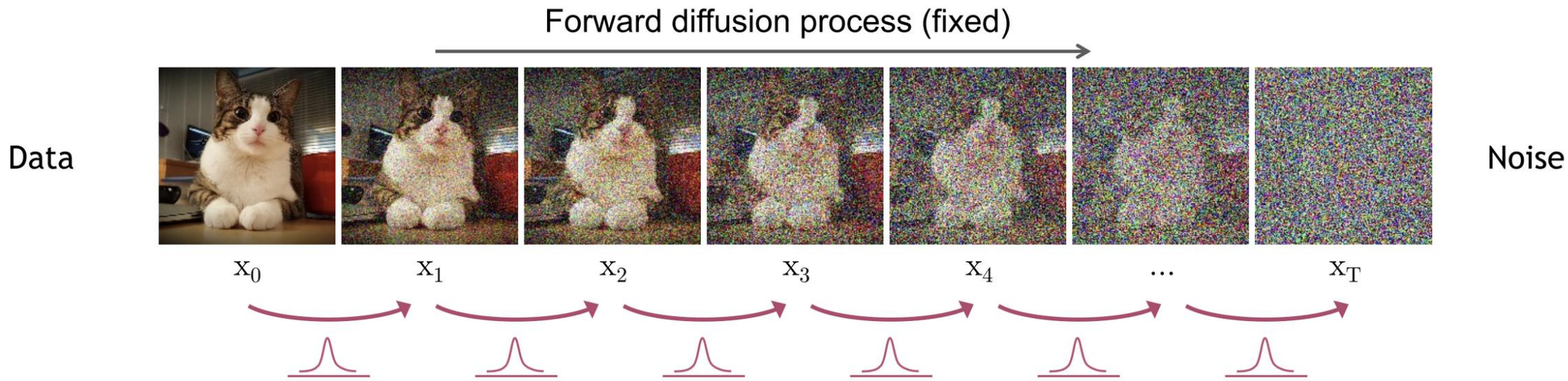
$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \underbrace{\mu_\theta(\mathbf{x}_t, t)}, \sigma_t^2\mathbf{I})$$

Trainable network
(U-net, Denoising Autoencoder)

$$\Rightarrow \quad p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

Image source: https://cvpr2023-tutorial-diffusion-models.github.io/

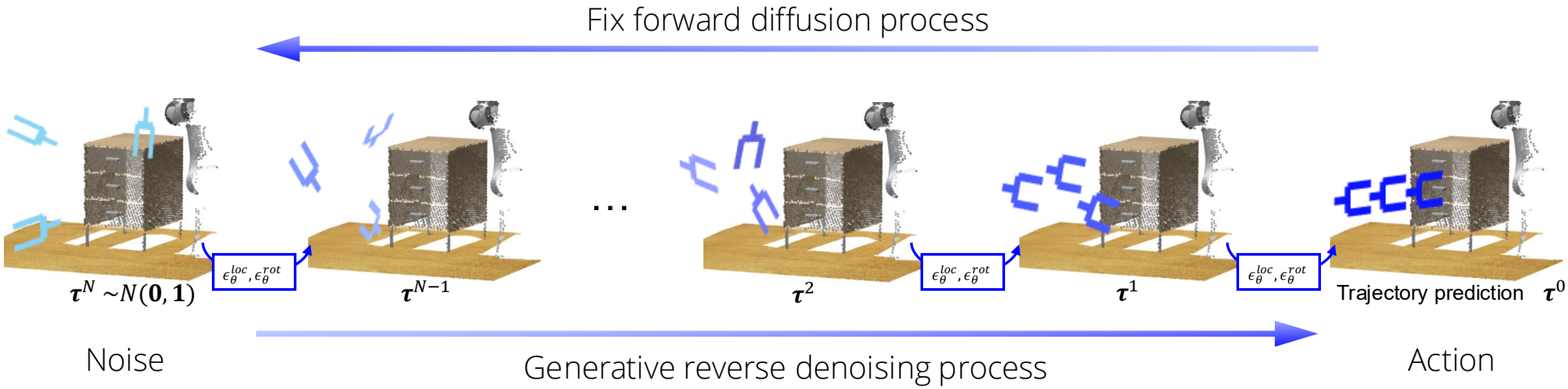
Denoising Diffusion Probabilistic Models (DDPM)



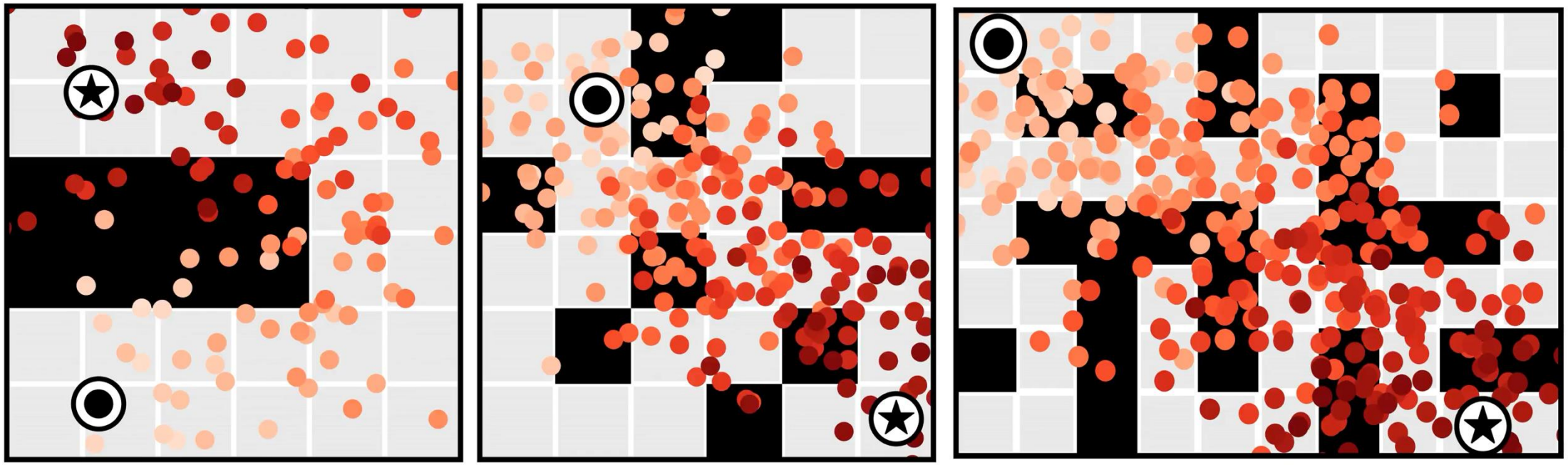
$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x_t}; \sqrt{1-\beta_t}\mathbf{x_{t-1}}, \beta_t\mathbf{I}) \quad \Rightarrow \quad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \qquad \text{(joint)}$$

Image source: https://cvpr2023-tutorial-diffusion-models.github.io/

Learning DDPM Models as Robot Policies



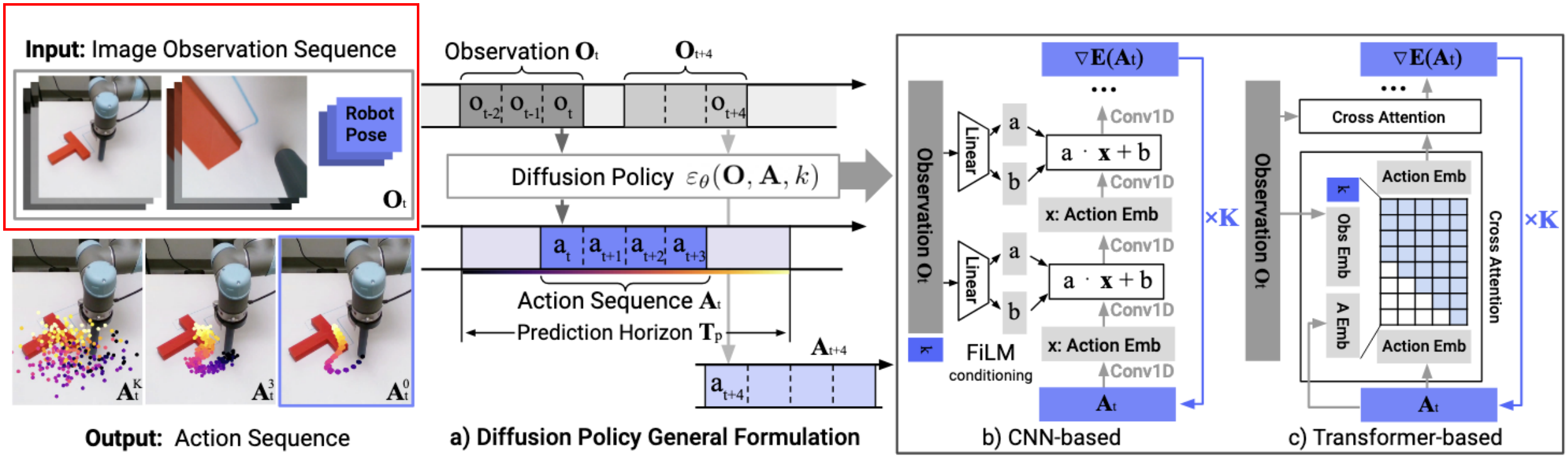
Learning DDPM Models as Robot Policies



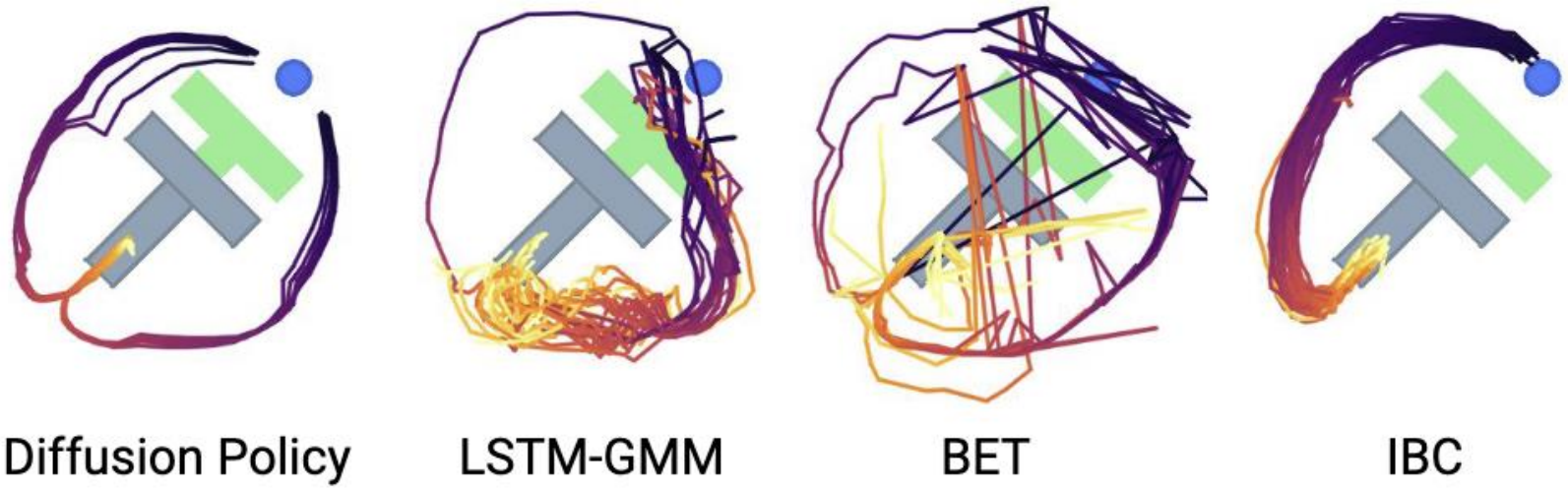
Planning with Diffusion for Flexible Behavior Synthesis. Janner et al.

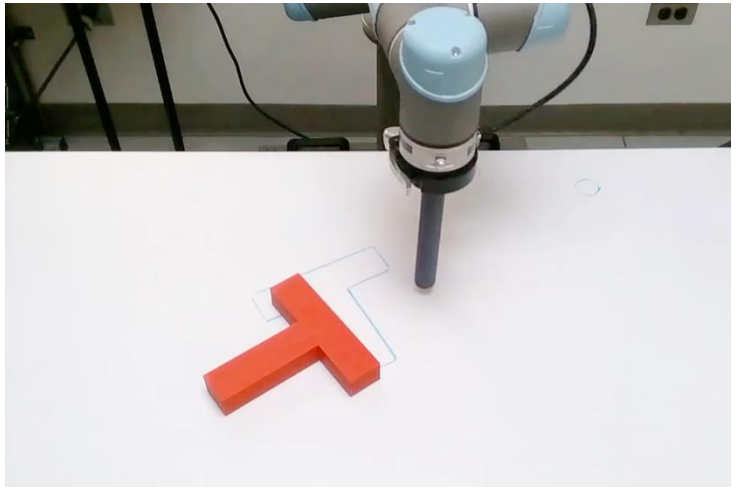
DDPM Policy with 2D Inputs

An input image is represented by a flattened feature vector

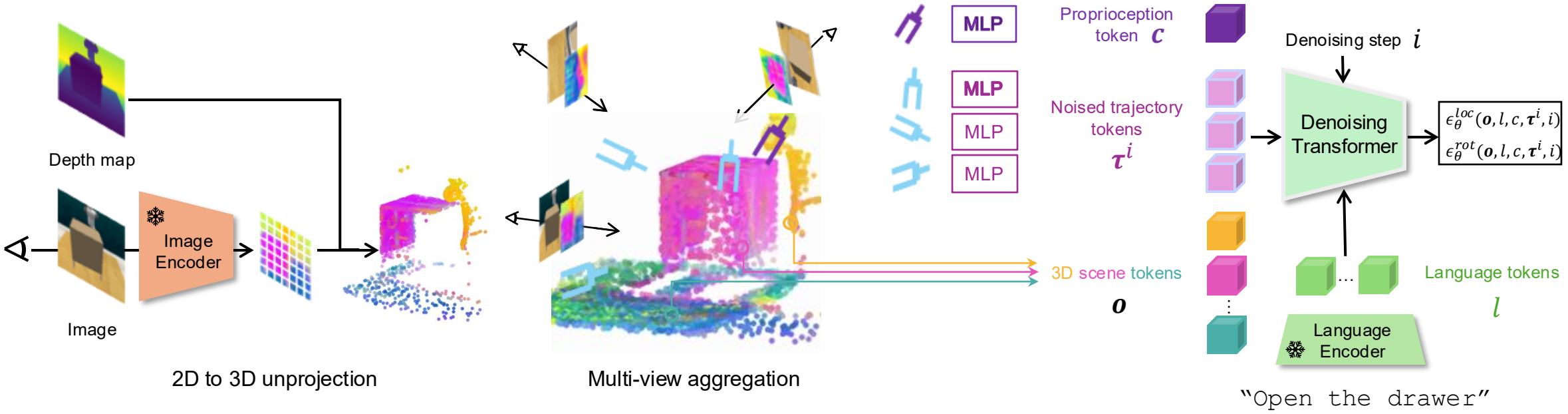


DDPM Policy with 2D Inputs





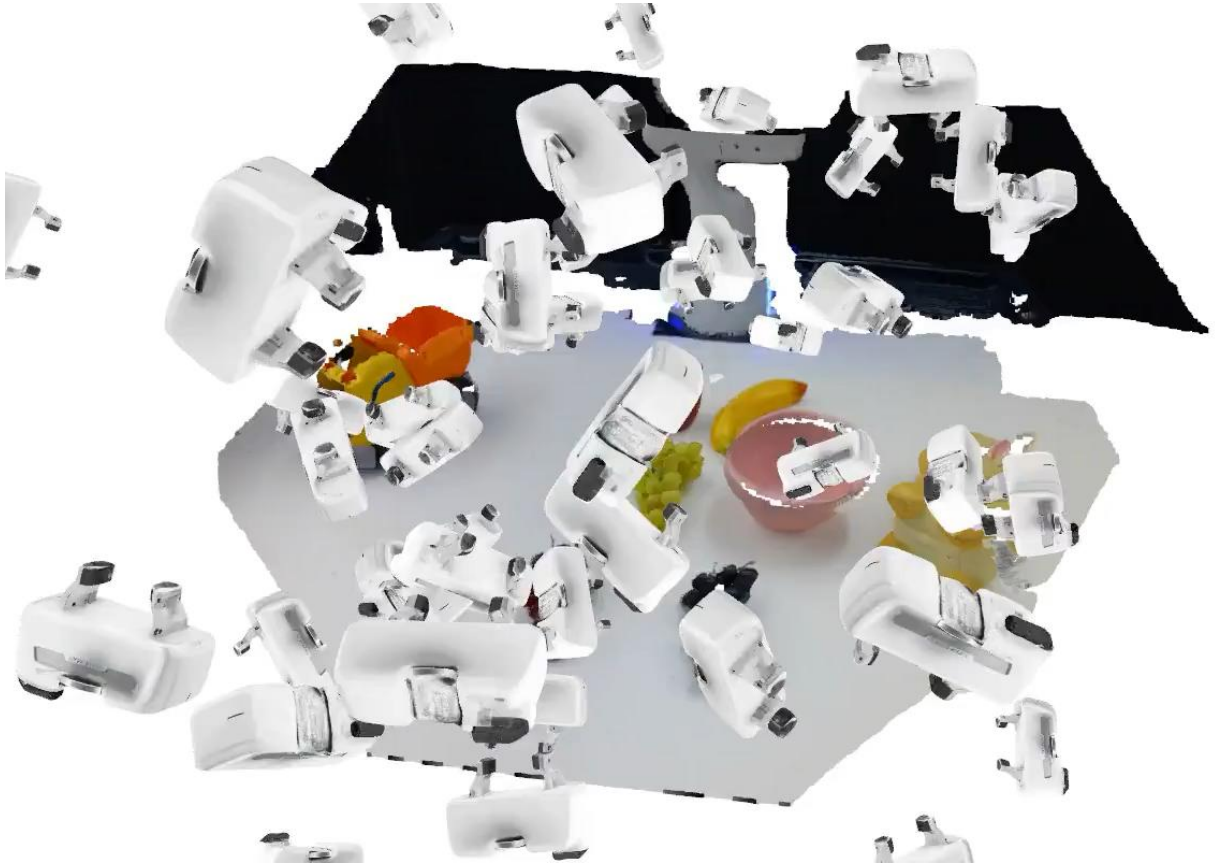
DDPM Policy with 3D Inputs



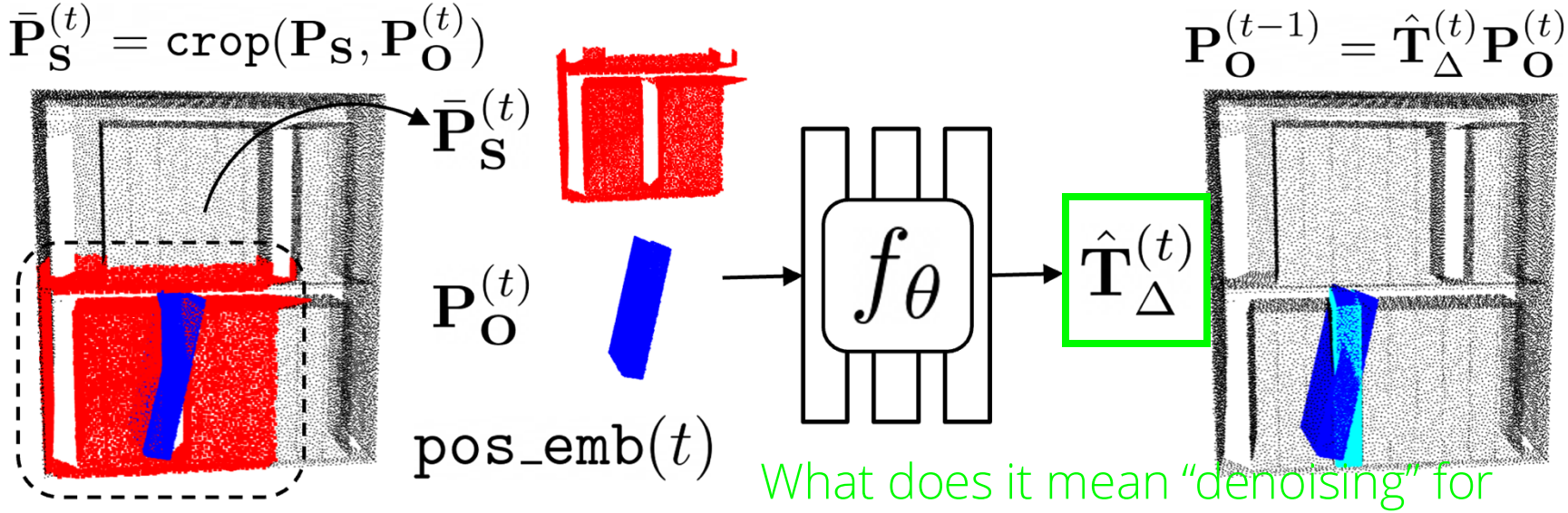
DDPM Policy with 3D Inputs

Calvin

Real-world



Can We Generate Other Pose Representations?



What does it mean "denoising" for geometric transformation?

$P_O^{(0)}$: (clean) object point cloud at diffusion step $\mathbf{0}$

$P_O^{(t)}$: (perturbed) object point cloud at diffusion step t

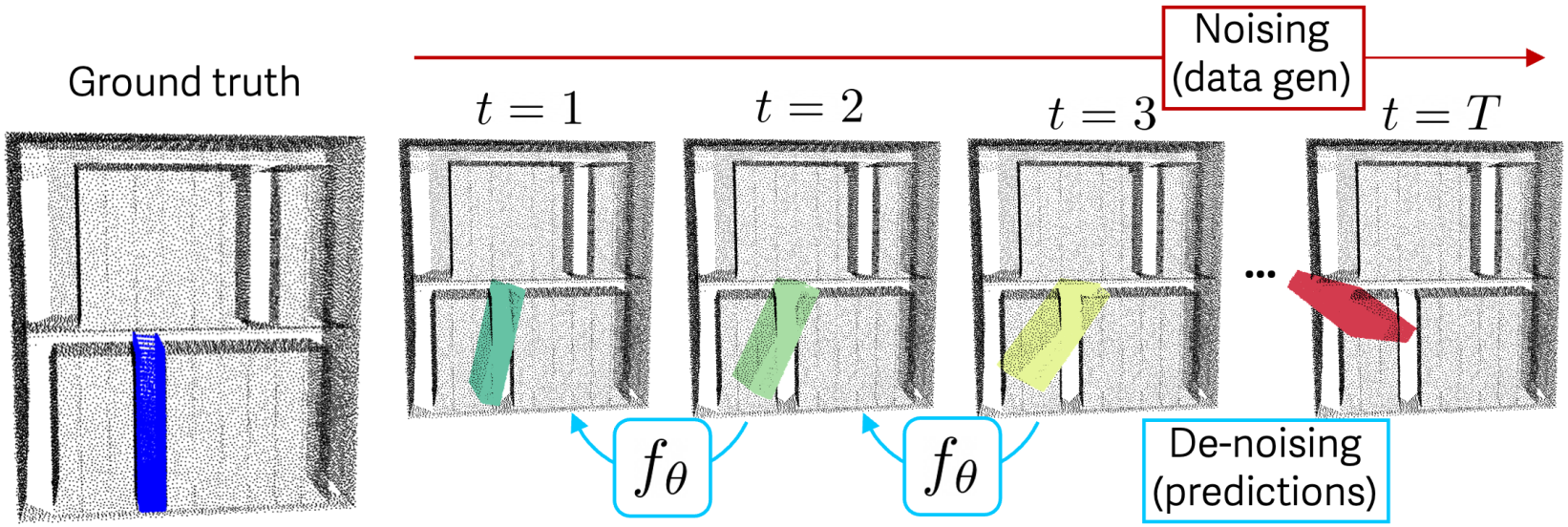
P_S: the whole scene point cloud

$P_S^{(t)}$: cropped scene point cloud around the object point cloud at diffusion step t

$\hat{T}_{\Delta}^{(t)}$: de-perturbed transformation from diffusion step t to $t-1$

Shelving, Stacking, Hanging: Relational Pose Diffusion for Multi-modal Rearrangement. Simeonov et al.

De-noising Process



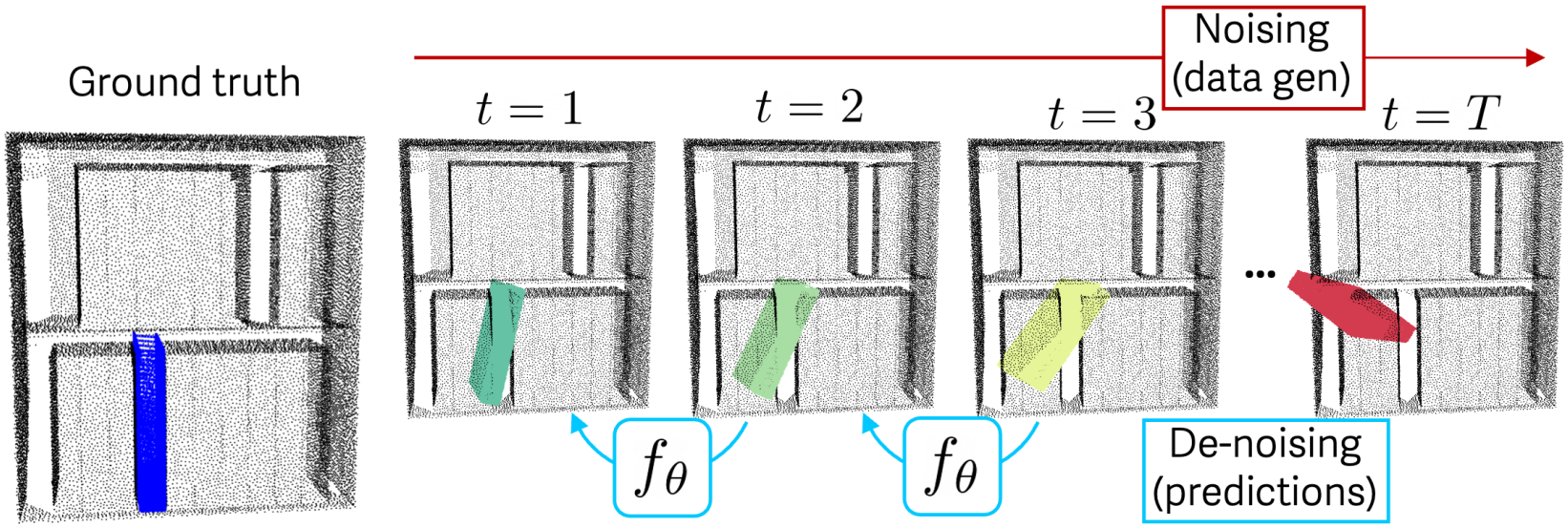
$$\mathbf{T}_{\Delta}^{(i)} = \mathbf{T}_{\Delta}^{\text{Rand}} f_\theta\left(\hat{\mathbf{P}}_{\mathbf{O}}^{(i)}, \mathbf{P}_{\mathbf{S}}, \texttt{pos\_emb}(t)\right) \qquad t = \texttt{i\_to\_t}(i)$$

$$\hat{\mathbf{T}}^{(i-1)} = \mathbf{T}_{\Delta}^{(i)} \hat{\mathbf{T}}^{(i)} \qquad \hat{\mathbf{P}}_{\mathbf{O}}^{(i-1)} = \mathbf{T}_{\Delta}^{(i)} \hat{\mathbf{P}}_{\mathbf{O}}^{(i)}$$

De-noising from diffusion step t to $t-1$

Shelving, Stacking, Hanging: Relational Pose Diffusion for Multi-modal Rearrangement. Simeonov et al.

De-noising Process



Remember DDPM?

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$$

$$\mathbf{T}_\Delta^{(i)} = \mathbf{T}_\Delta^{\text{Rand}} f_\theta\left(\hat{\mathbf{P}}_\mathbf{O}^{(i)}, \mathbf{P}_\mathbf{S}, \texttt{pos\_emb}(t)\right) \qquad t = \texttt{i\_to\_t}(i)$$

$$\hat{\mathbf{T}}^{(i-1)} = \mathbf{T}_\Delta^{(i)} \hat{\mathbf{T}}^{(i)} \qquad \hat{\mathbf{P}}_\mathbf{O}^{(i-1)} = \mathbf{T}_\Delta^{(i)} \hat{\mathbf{P}}_\mathbf{O}^{(i)}$$

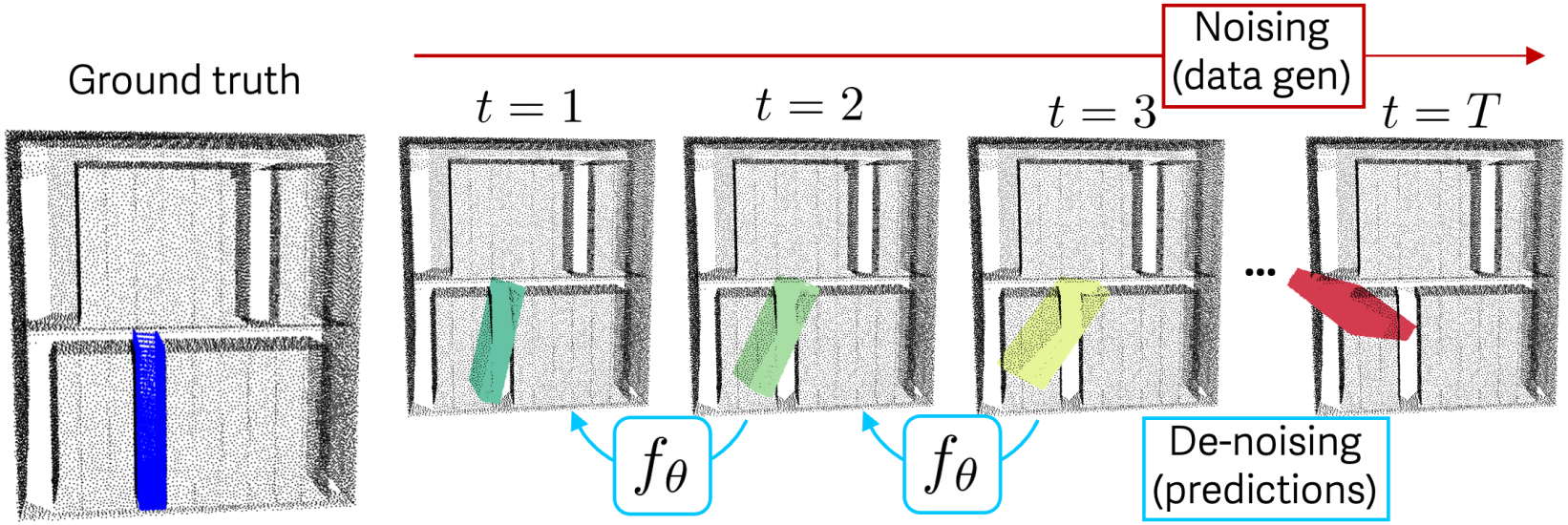
Shelving, Stacking, Hanging: Relational Pose Diffusion for Multi-modal Rearrangement. Simeonov et al.

How to Generate Training Labels?

Sample a random transformation at diffusion step T, and find the interpolated intermediate transformation

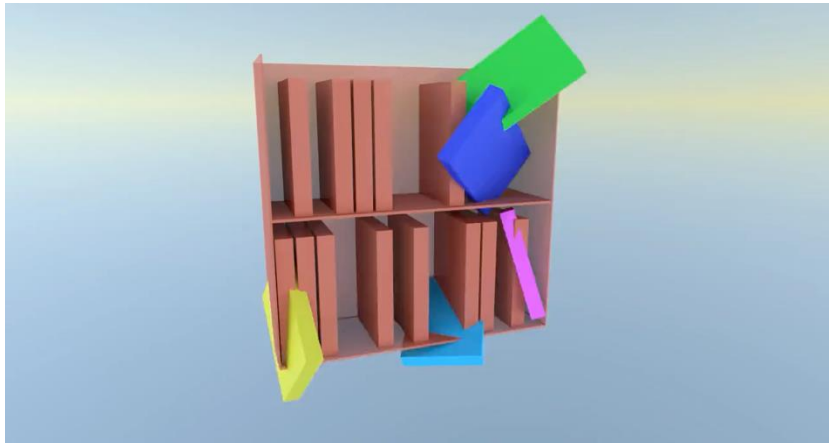
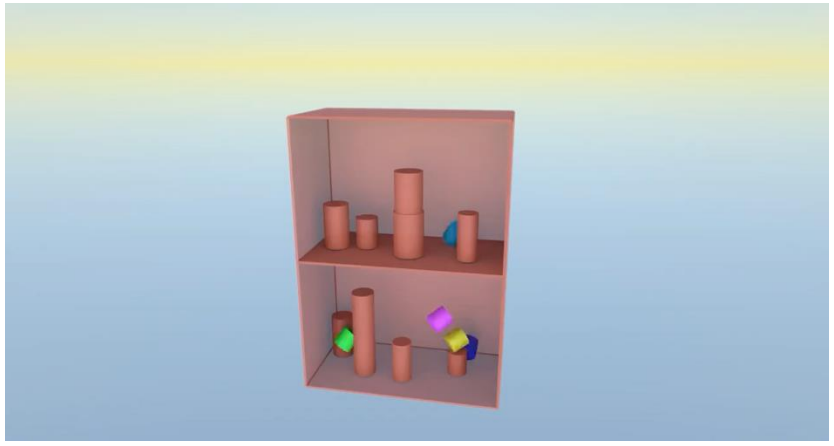
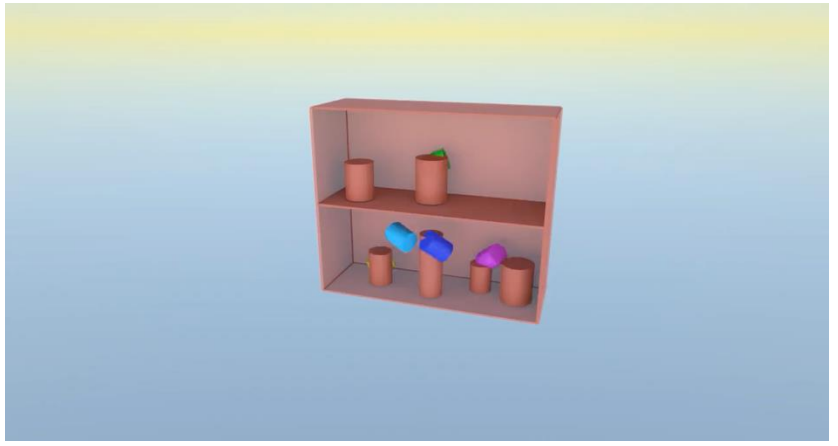
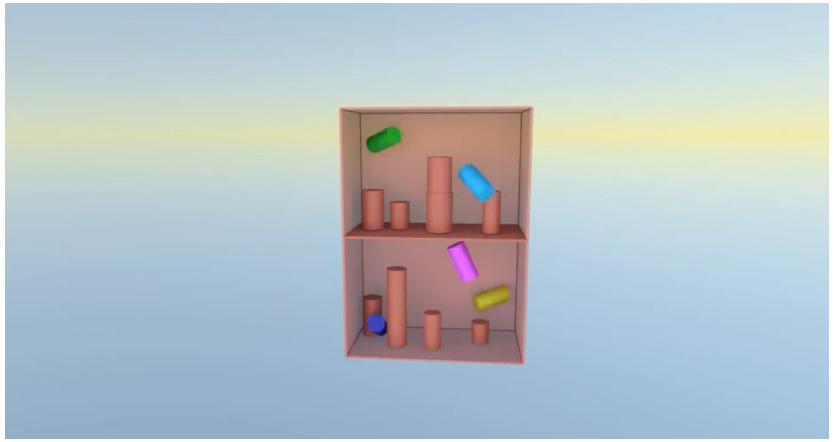
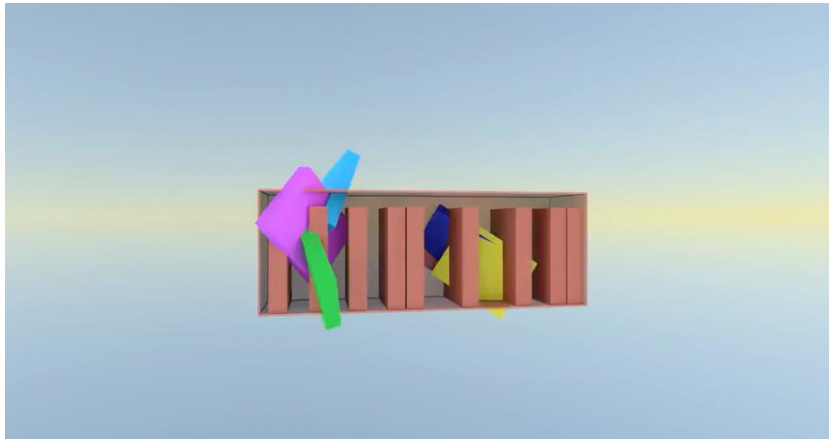
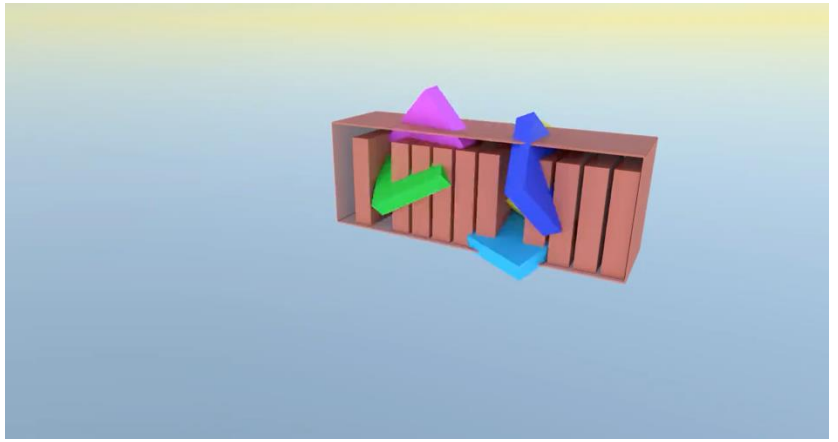
$$\{\mathbf{T}_{\text{interp}}^{(s)}\}_{s=1}^{t} = \texttt{interp}(\mathbf{T}_{\text{noise, inv}}^{(t)}, t)$$

$$\mathbf{T}_{\Delta,\text{GT}}^{(t)} = [\mathbf{T}_{\text{interp}}^{(t-1)}]^{-1}\mathbf{T}_{\text{interp}}^{(t)}$$



Shelving, Stacking, Hanging: Relational Pose Diffusion for Multi-modal Rearrangement. Simeonov et al.

Generated Results

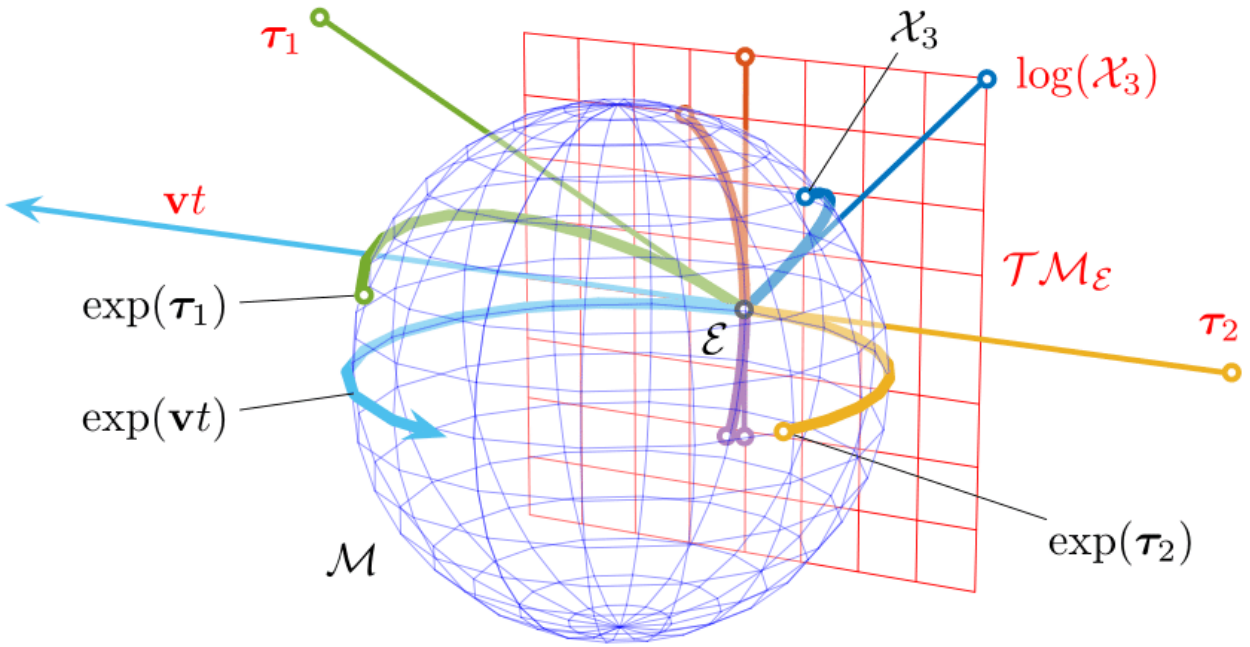







Shelving, Stacking, Hanging: Relational Pose Diffusion for Multi-modal Rearrangement. Simeonov et al.

Can We Generate SE(3) Representations?

$$q(\boldsymbol{H}|\boldsymbol{H}_\mu, \boldsymbol{\Sigma}) \propto \exp\left(-0.5 \ \left\|\text{Logmap}(\boldsymbol{H}_\mu^{-1}\boldsymbol{H})\right\|^2_{\boldsymbol{\Sigma}^{-1}}\right)$$

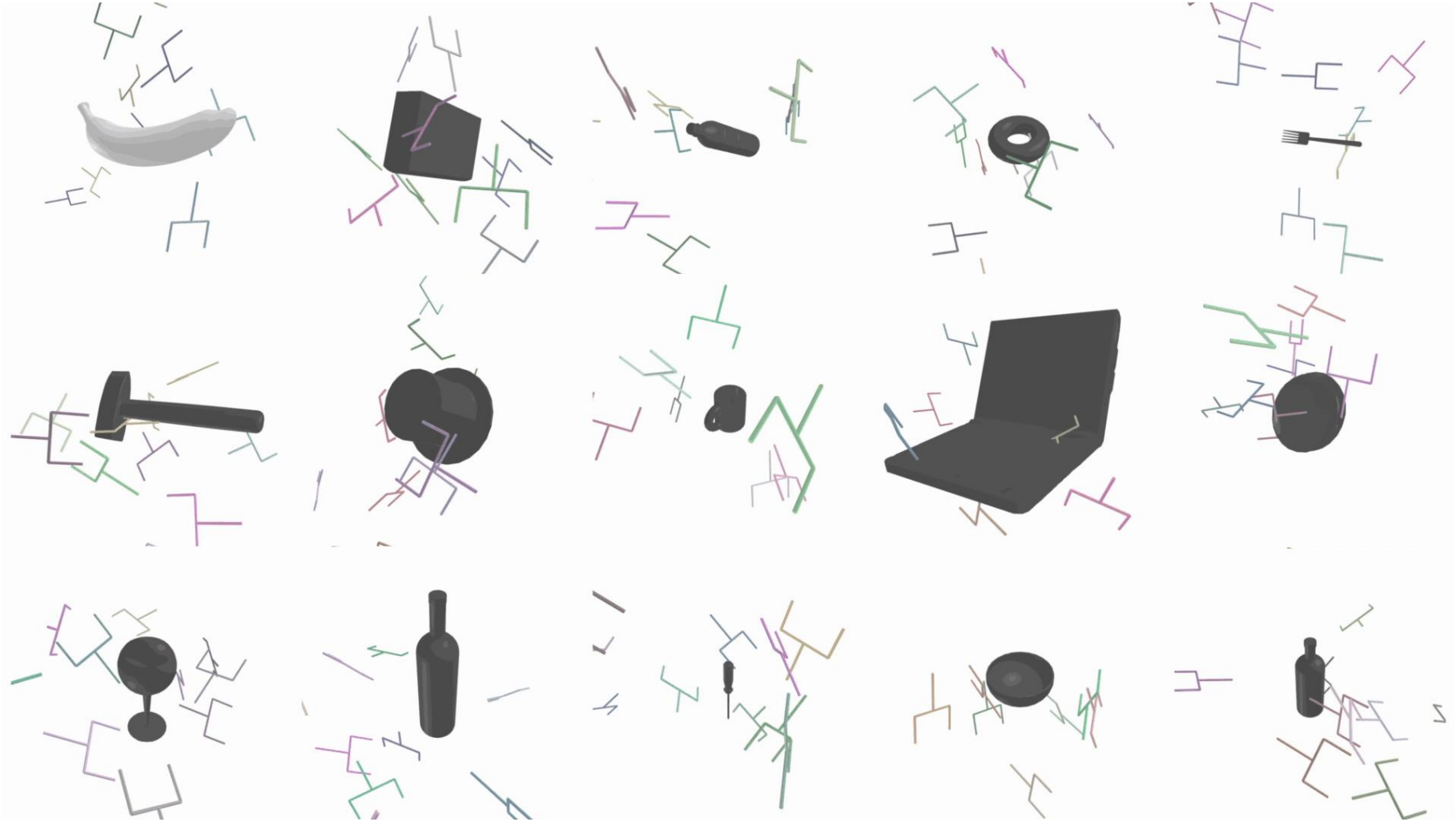
$H_\mu \in SE(3)$ is the mean
$\Sigma \in \mathbb{R}^{6\times6}$ is the covariance matrix



SE(3)-DiffusionFields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. Urain et al.

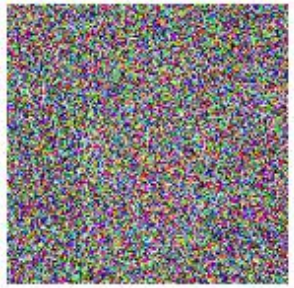
https://www.researchgate.net/figure/Representation-of-the-relation-between-the-Lie-group-and-the-Lie-algebra-The-Lie_fig1_329413756

Can We Generate SE(3) Representations?



SE(3)-DiffusionFields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. Urain et al.

Example: Flow Matching Policy

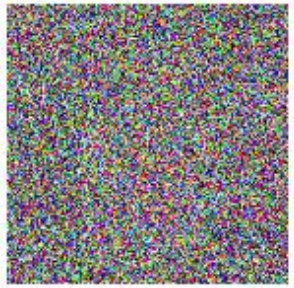


Noise

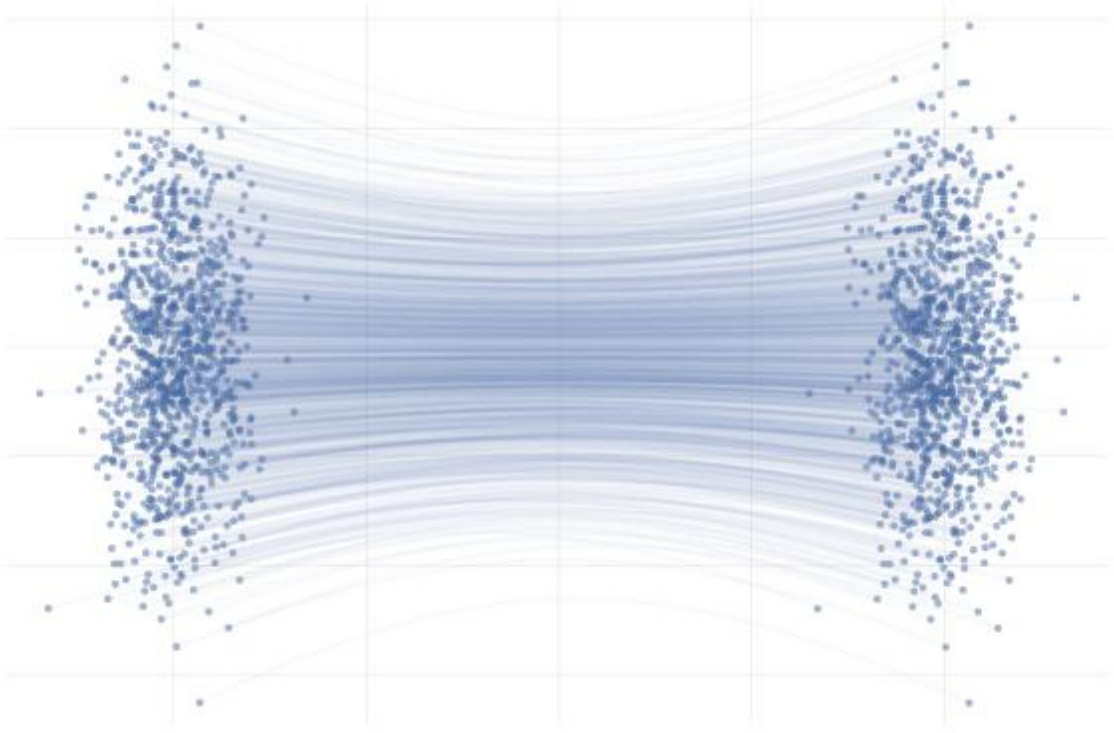
Data

Video source: https://mlg.eng.cam.ac.uk/blog/2024/01/20/flow-matching.html

Flow Matching Policy Learn Velocity Field to Transport One Distribution to Another



Noise



Data

Image source: https://mlg.eng.cam.ac.uk/blog/2024/01/20/flow-matching.html

Transportation Velocity Field is not Unique

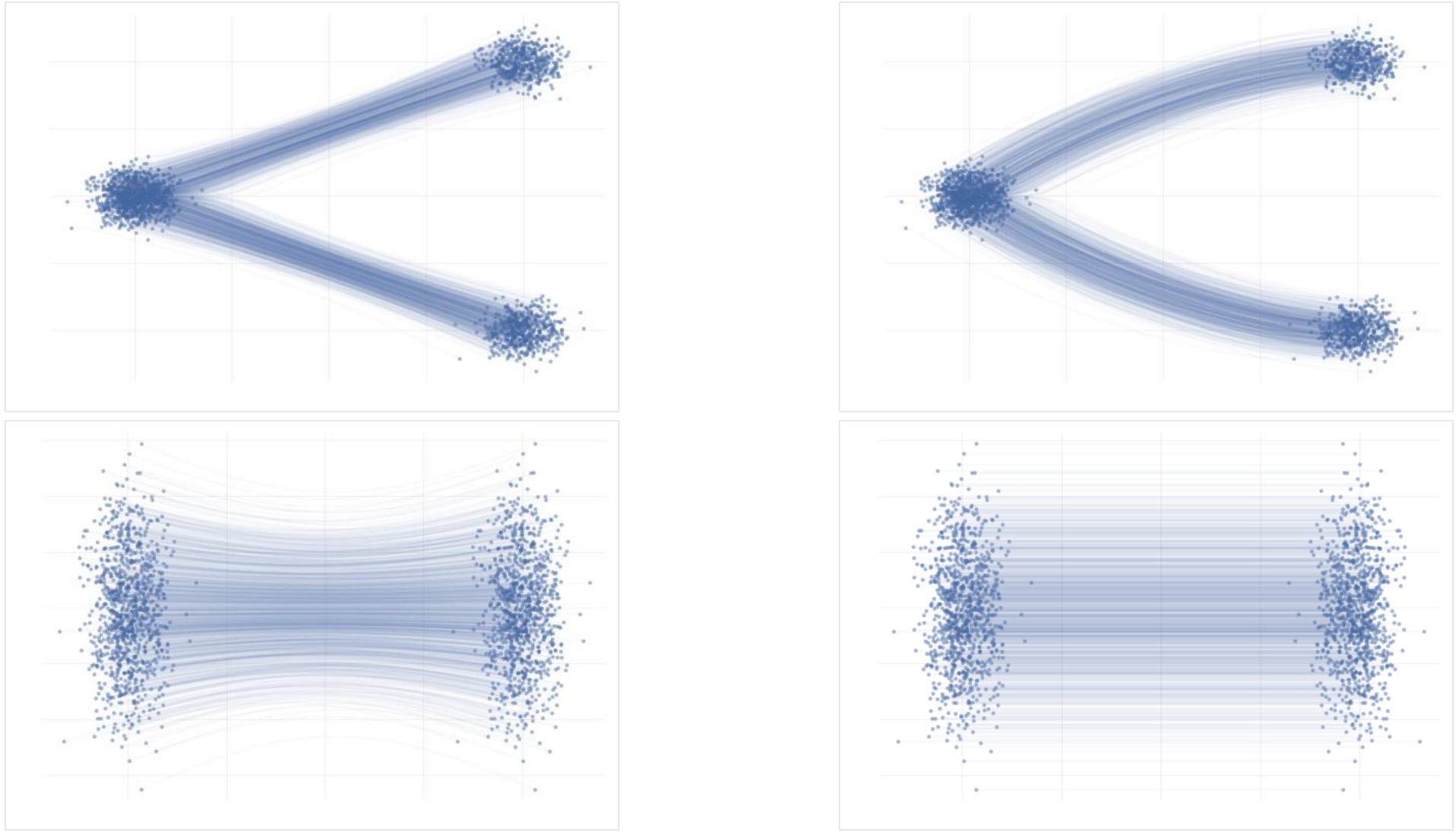


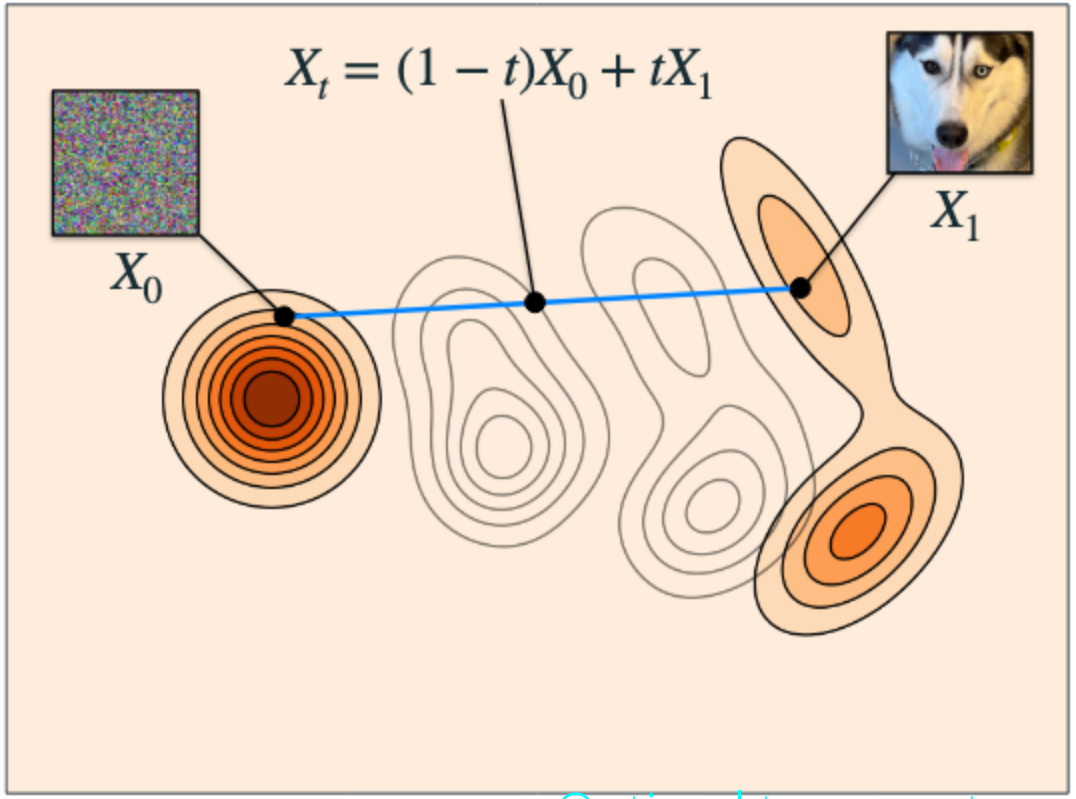
Image source: https://mlg.eng.cam.ac.uk/blog/2024/01/20/flow-matching.html

Simplest version of Flow Matching

- Arbitrary $X_0 \sim p, X_1 \sim q$
- Arbitrary coupling $(X_0, X_1) \sim \pi_{0,1}$

Why does it work?

- Build flow from conditional flows
- Regress conditional flows



Optimal transport

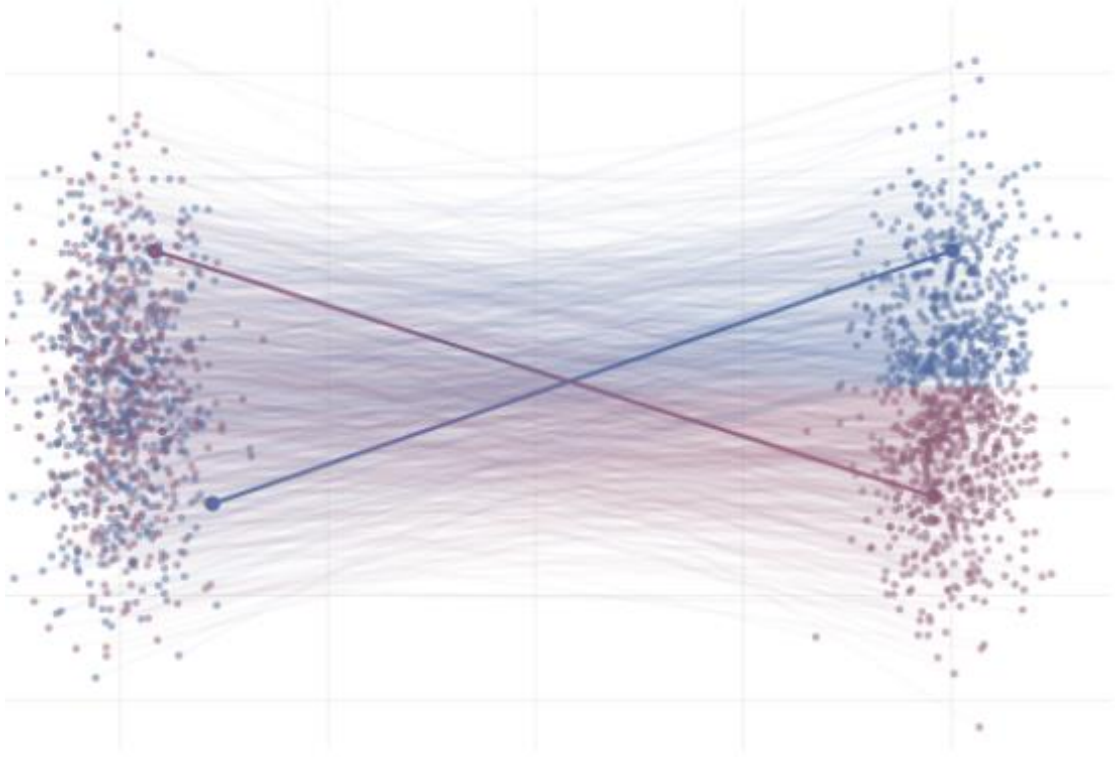
$$\mathbb{E}_{t,X_0,X_1}\left\| u_t^\theta(X_t) - (X_1 - X_0) \right\|^2$$

Neural network

Slide credit: https://neurips.cc/media/neurips-2024/Slides/99531.pdf

Random Coupling of Two Distributions Results in Curved Velocity Field

Training



Inference

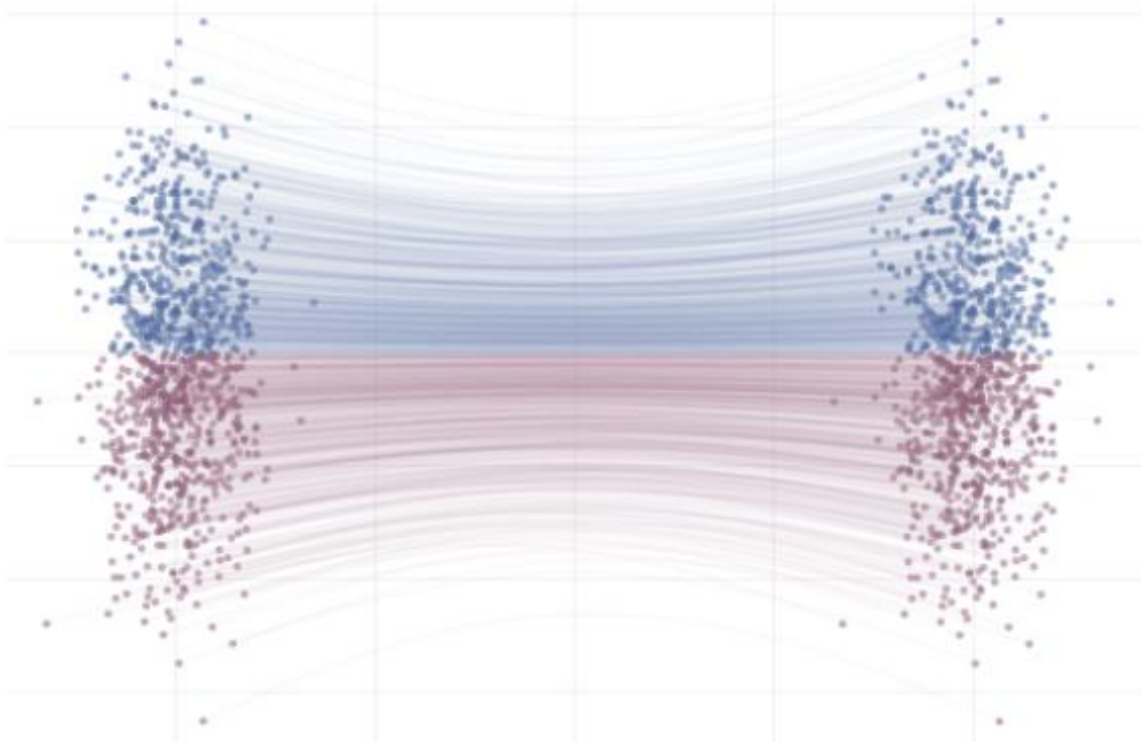
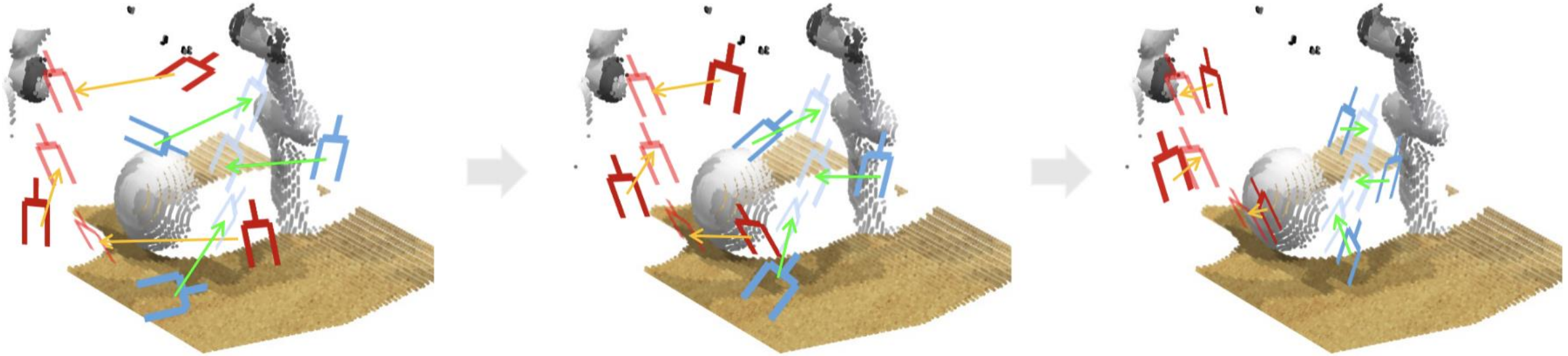


Image source: https://mlg.eng.cam.ac.uk/blog/2024/01/20/flow-matching.html

Learning Flow Matching Models as Robot Policies



3D FlowMatch Actor: Unified 3D Policy for Single- and Dual-Arm Manipulation. Gkanatsios et al.

Question

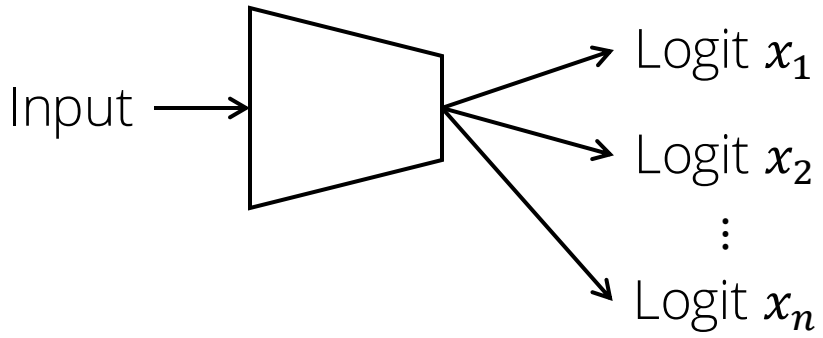
- How to reduce sampling steps during training?
- Is random coupling a good training recipe?
- What is the optimal prior distribution $\mathbf{X_0}$ to sample target distribution $\mathbf{X_1}$?

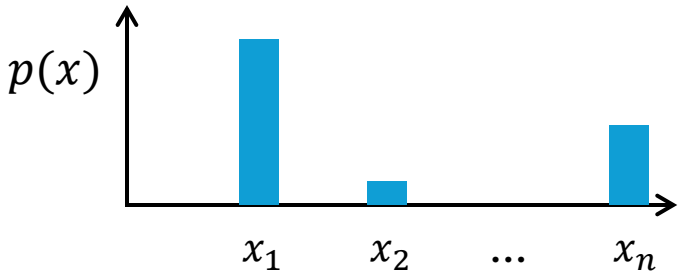
We'll have an additional video lecture by TAs (Jia-Wei and Chien-Sheng) on Flow Matching!

Choices of Probabilistic Generative Models

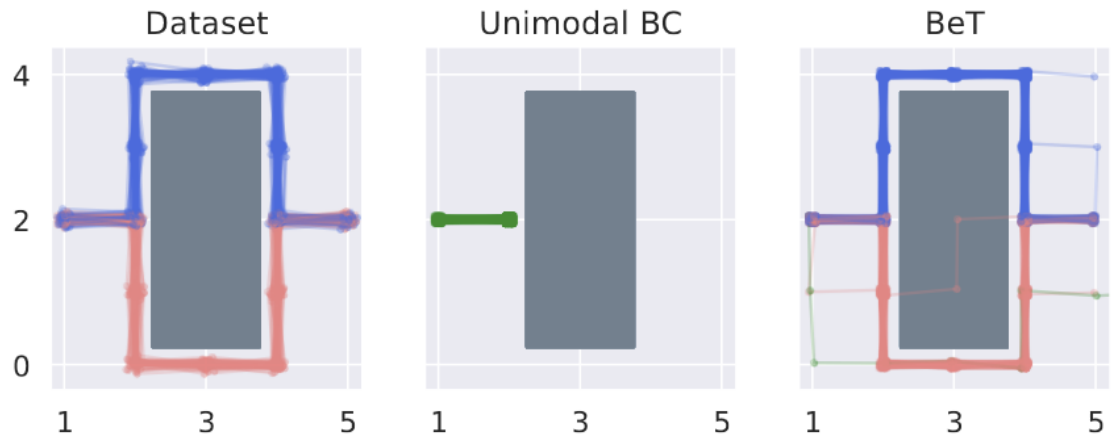
Categorical

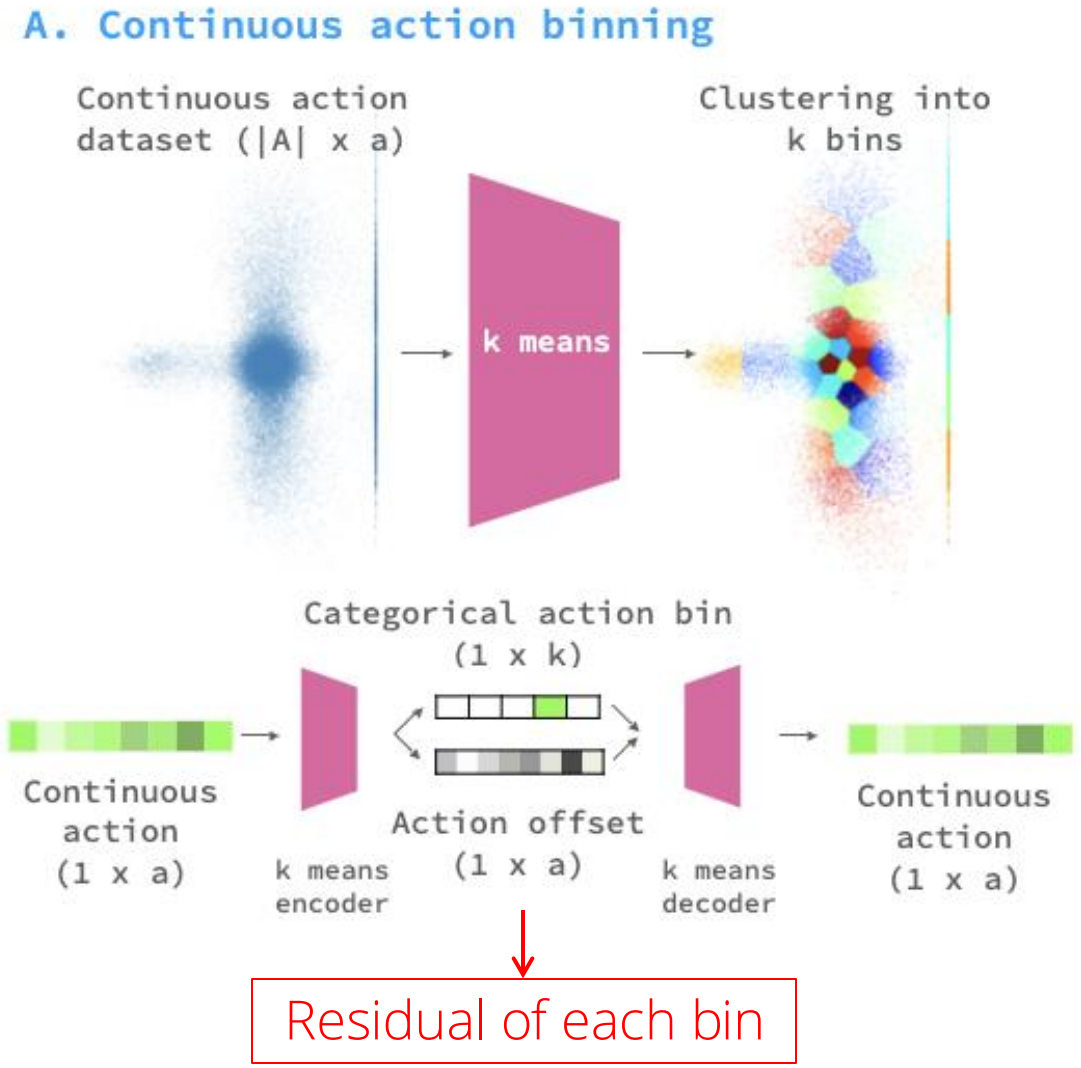
- Predict the categorical probability of sample "x"
- Most LLMs etc



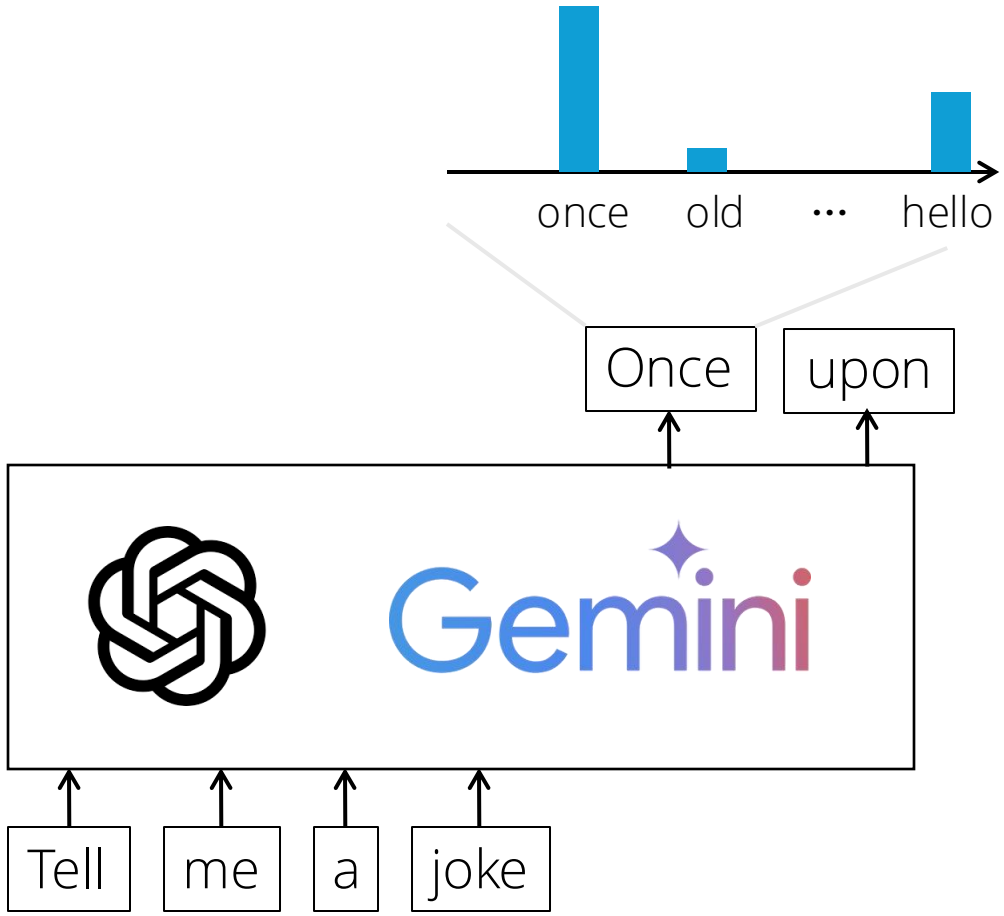


Example: Categorical Generative Models with Residuals





Example: Discrete Autoregressive Transformer as Categorical Generative Models



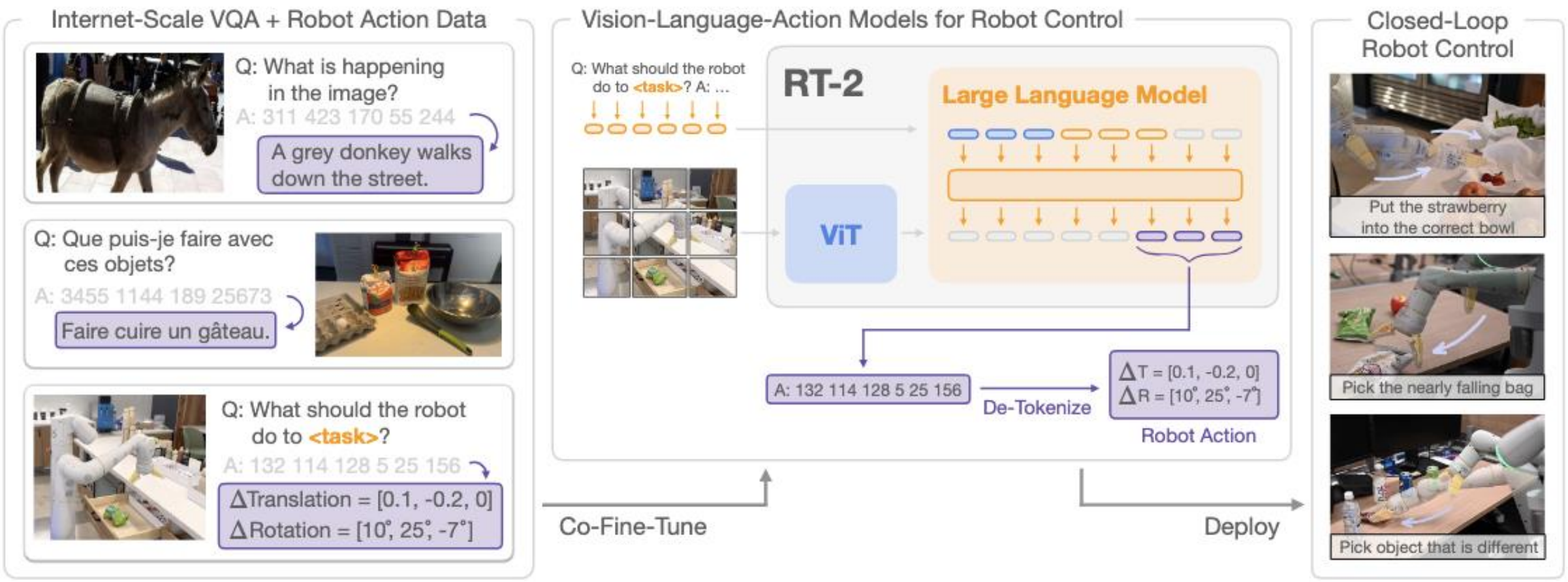
GPT-4.5

I'm going through a tough time after failing a test

Aw, I'm really sorry to hear that. Failing a test can feel pretty tough and discouraging, but remember, it's just one moment—not a reflection of who you are or your capabilities. Want to talk about what happened, or do you just need a distraction? I'm here either way.

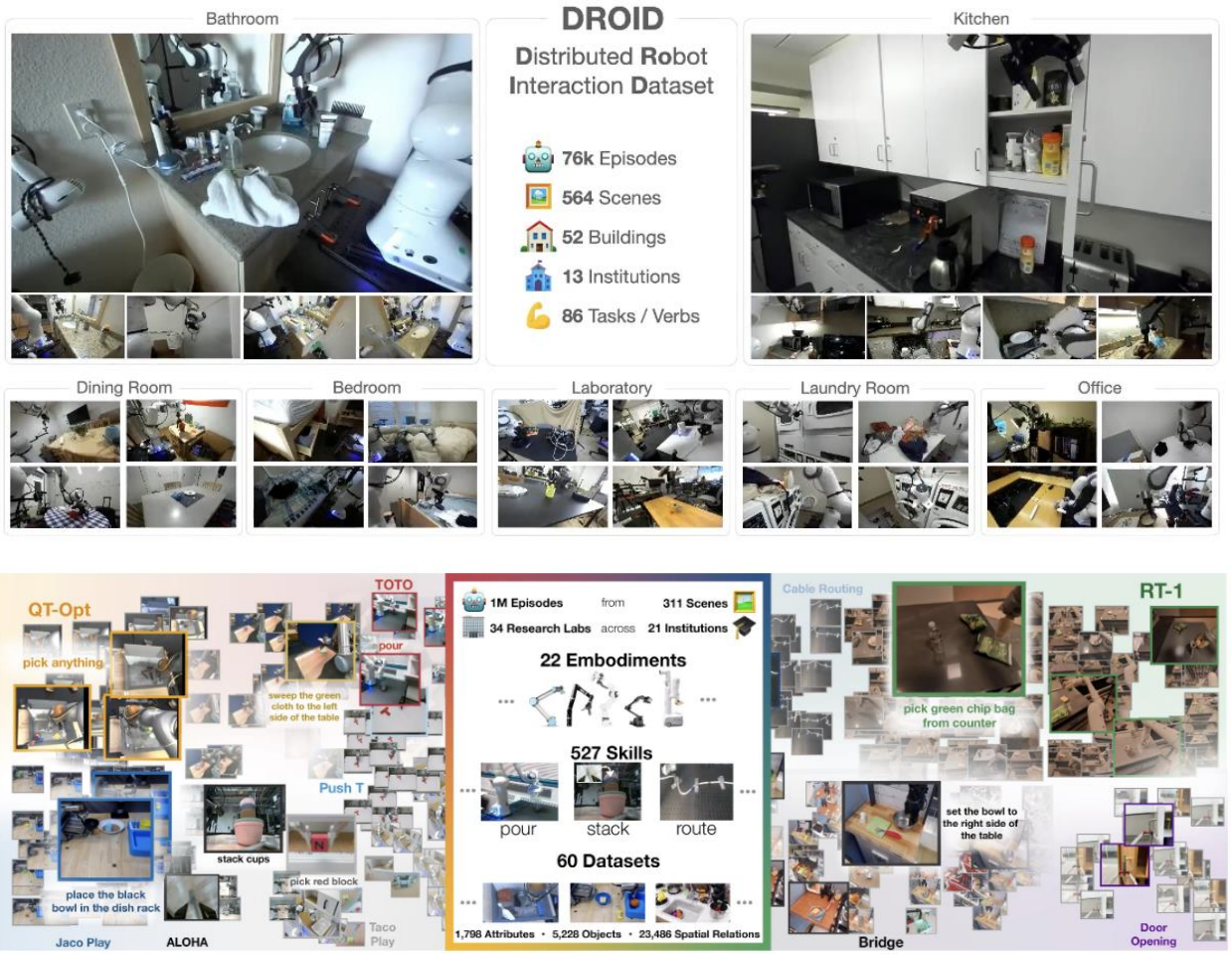
https://openai.com/index/introducing-gpt-4-5/

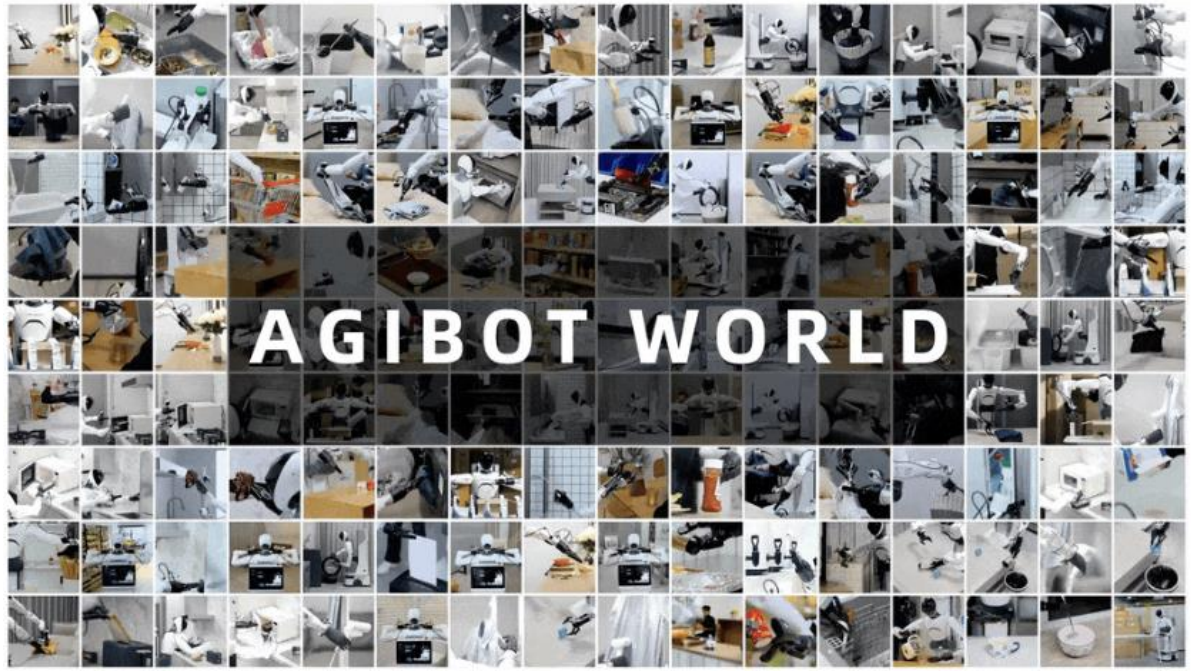
Vision-Language-Action Models (VLAs): Autoregressive Transformer as Robot Policies



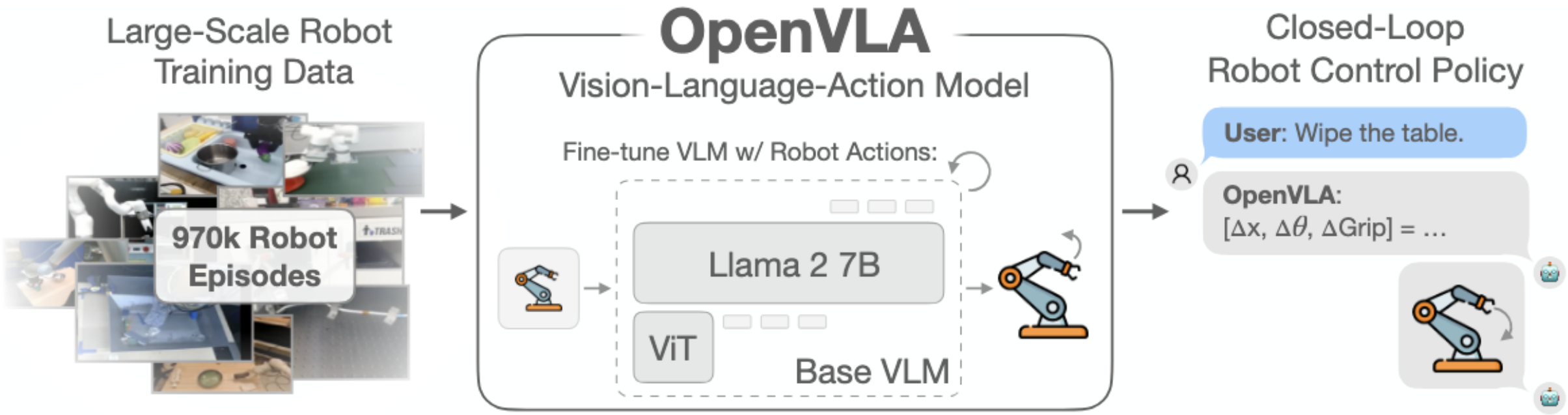
RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. Google DeepMind.

Large-Scale Robotic Datasets



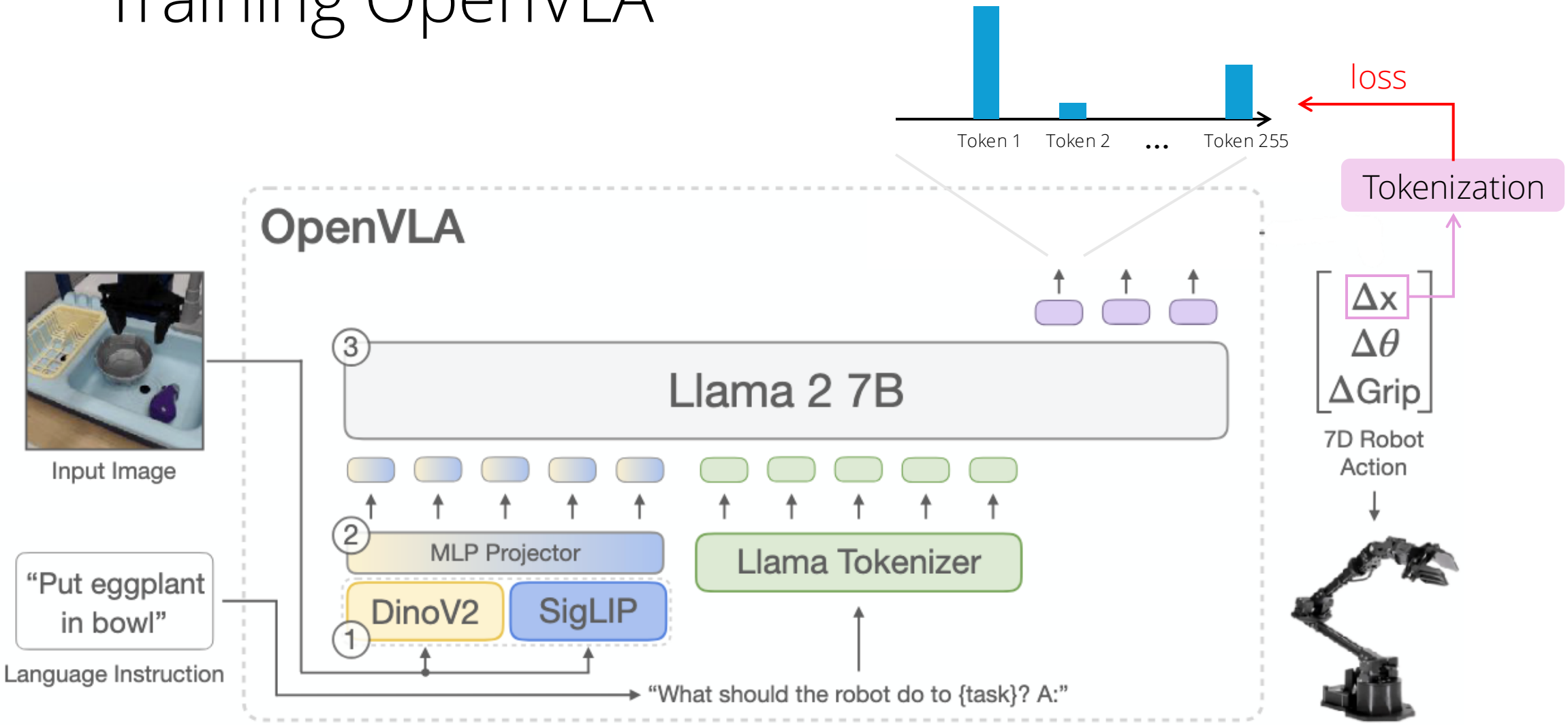


Vision-Language-Action Models (VLAs): Autoregressive Transformer as Robot Policies



OpenVLA: An Open-Source Vision-Language-Action Model. Kim et al.

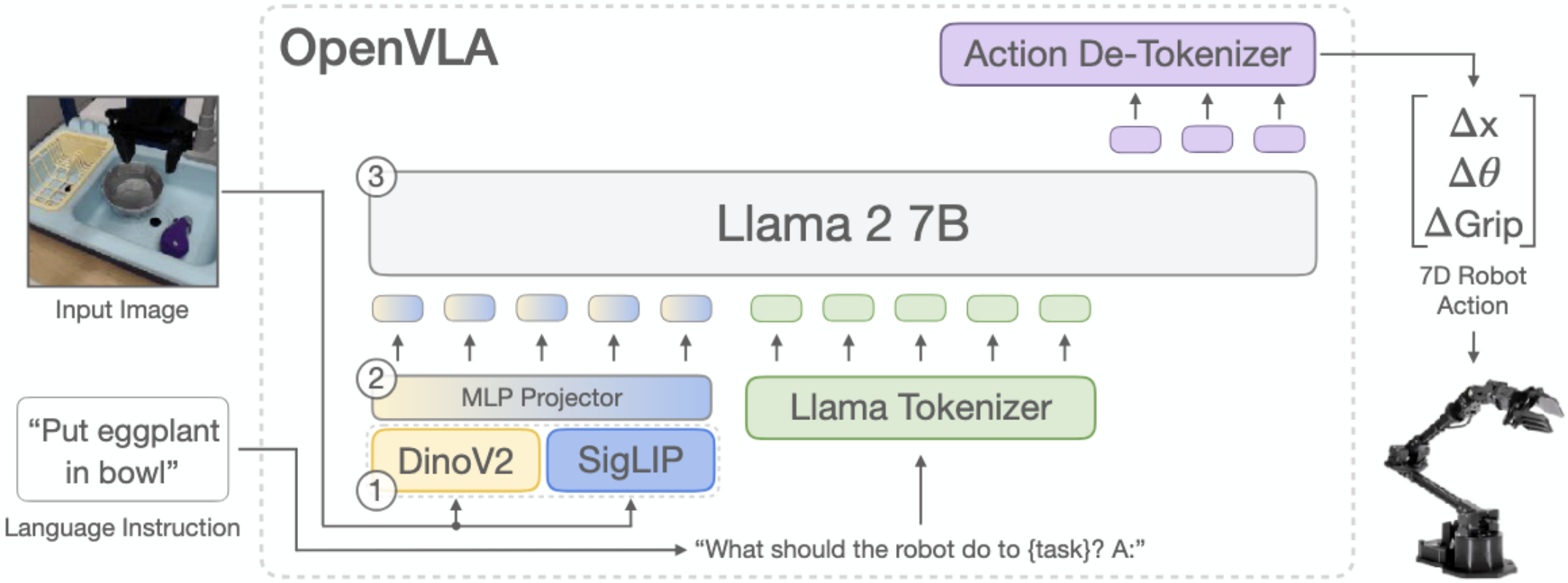
Training OpenVLA



How to tokenize real-valued actions?
Normalization and mapping to rarely used tokens

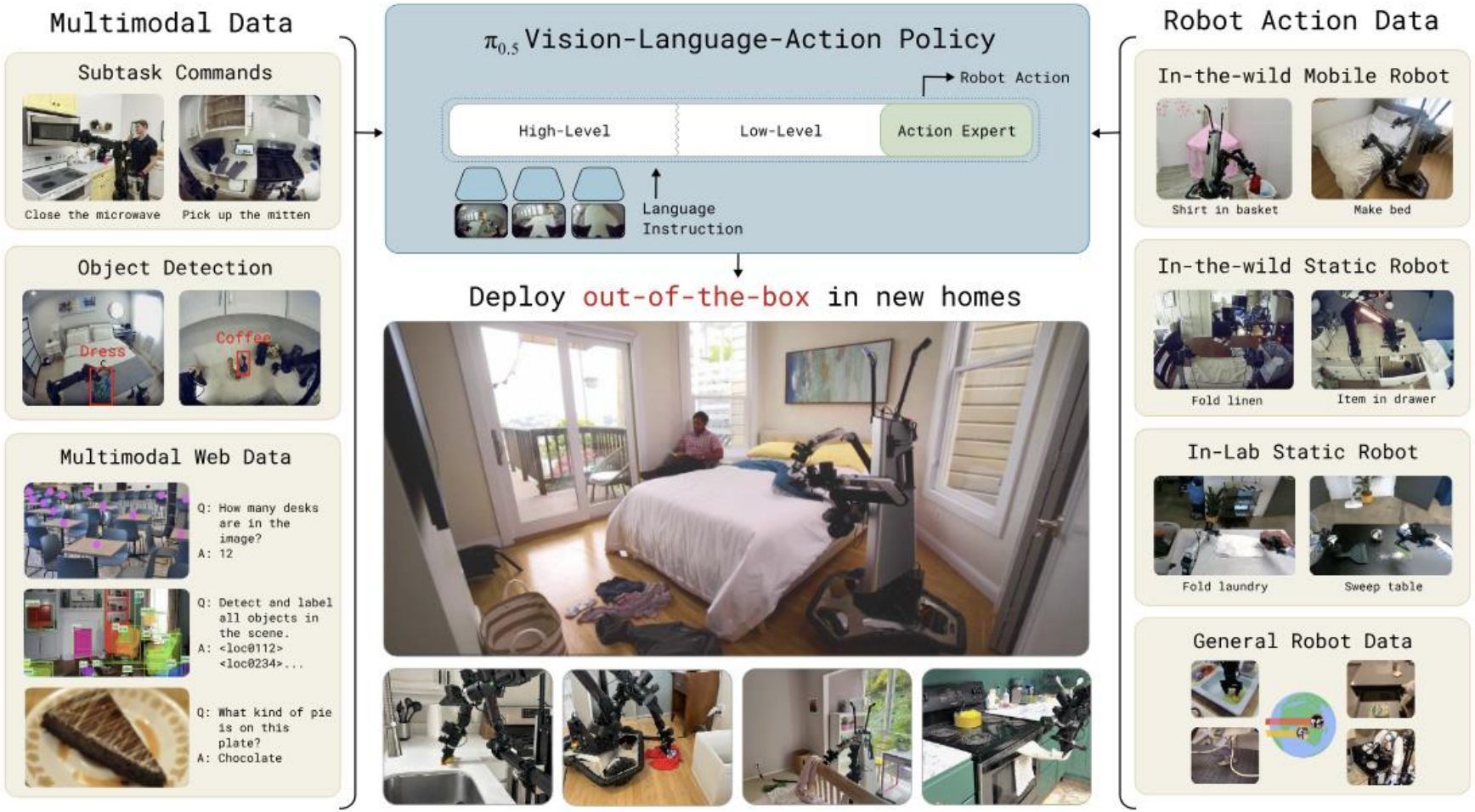
OpenVLA: An Open-Source Vision-Language-Action Model. Kim et al.

Inference with OpenVLA



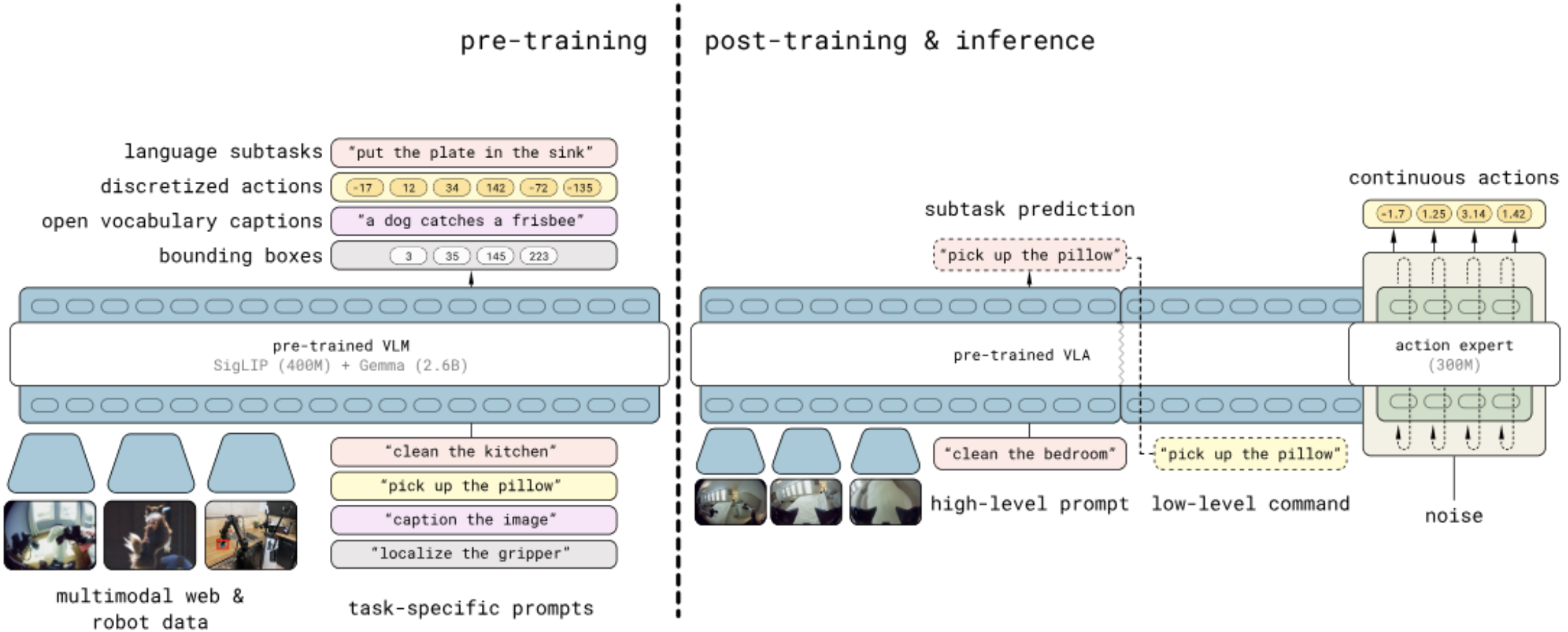
OpenVLA: An Open-Source Vision-Language-Action Model. Kim et al.

Autoregressive Transformers Jointly Handle Robotic, Vision, and Language Tasks



$\pi_{0.5}$: Vision-Language-Action Model with Open-World Generalization

Autoregressive Transformers Jointly Handle Robotic, Vision, and Language Tasks



Is Behavior Cloning + DAGGER All You Need?

- Of course, No!

- More problems:
  - **Causal confusion**. Does your model look at the correct feature so that it generalizes to novel scene?
  - **Multi-modality behaviors**. What if there are multiple ways to achieve a goal? What is the better formulation of the model output?
  - **No generalization to new scenes.** The model performs poorly in unseen / slightly difference scene

Robotics is Not Solved: SOTA Policies Don’t Generalize

Succeed in the real world

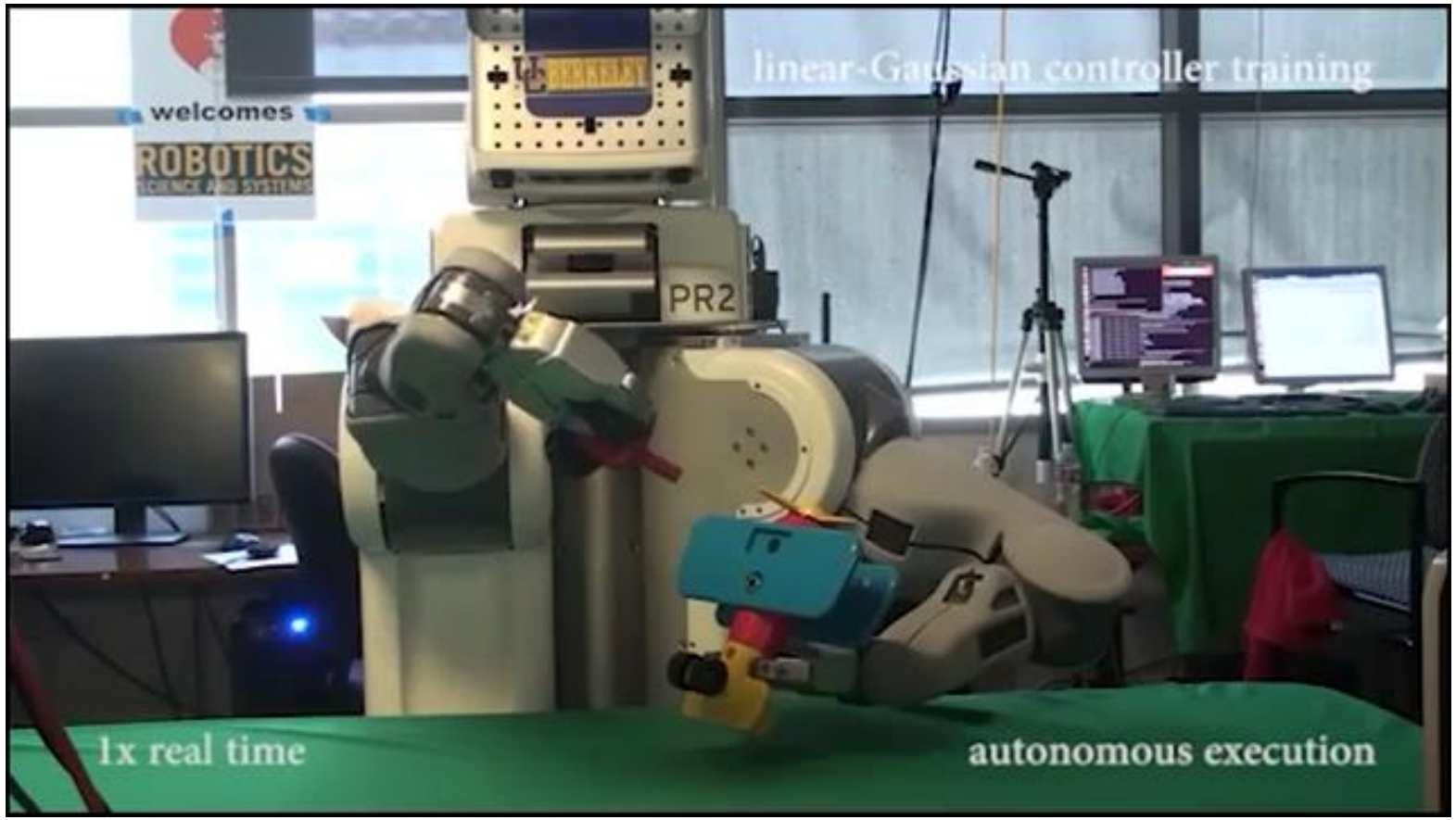
Fail in the digital twin

OpenVLA: An Open-Source Vision-Language-Action Model

Solution: Exploration in the Environment

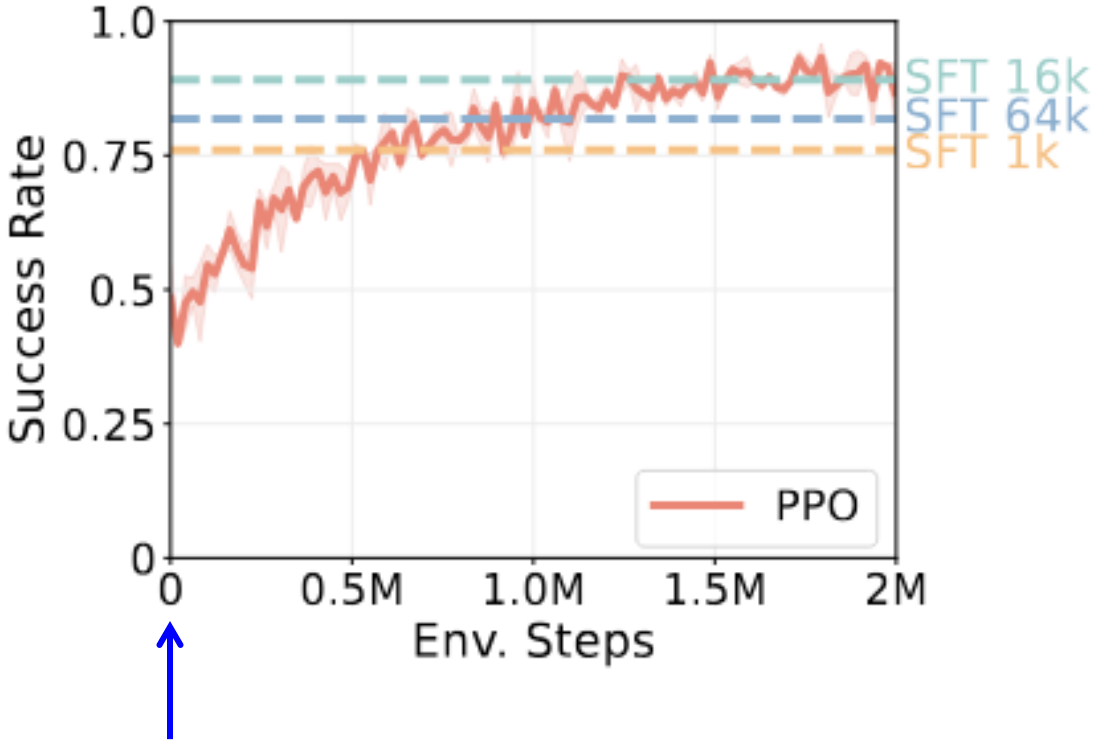
- Try out new behaviors with the hope of discovering optimal solutions

https://www.youtube.com/watch?v=kFPcoclV5Vo



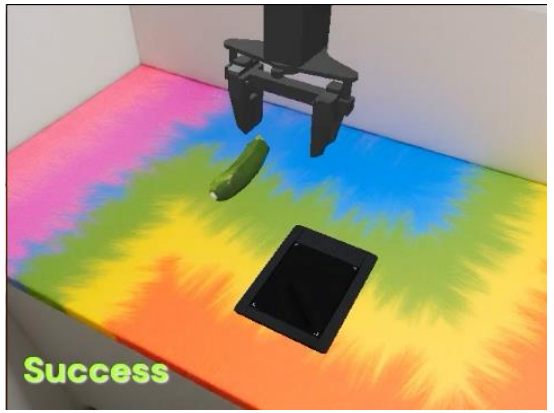
https://www.youtube.com/watch?v=JeVppkoloXs&t=11s

Exploration Improves Trained Policies



Before RL finetuning

We'll cover in the next week!



What Can RL Bring to VLA Generalization? An Empirical Study. Liu et al.

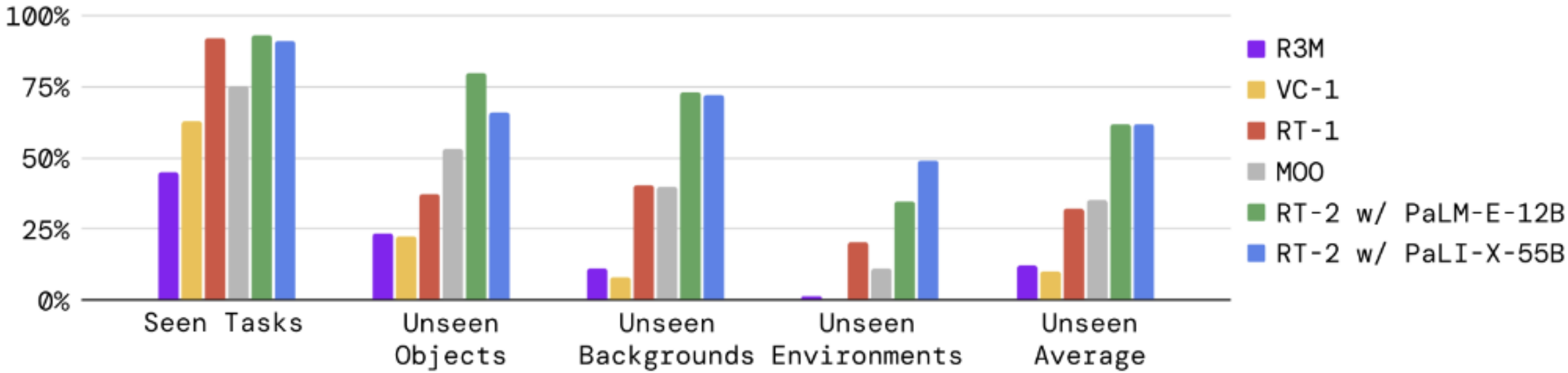
Is Behavior Cloning + DAGGER All You Need?

- Of course, No!

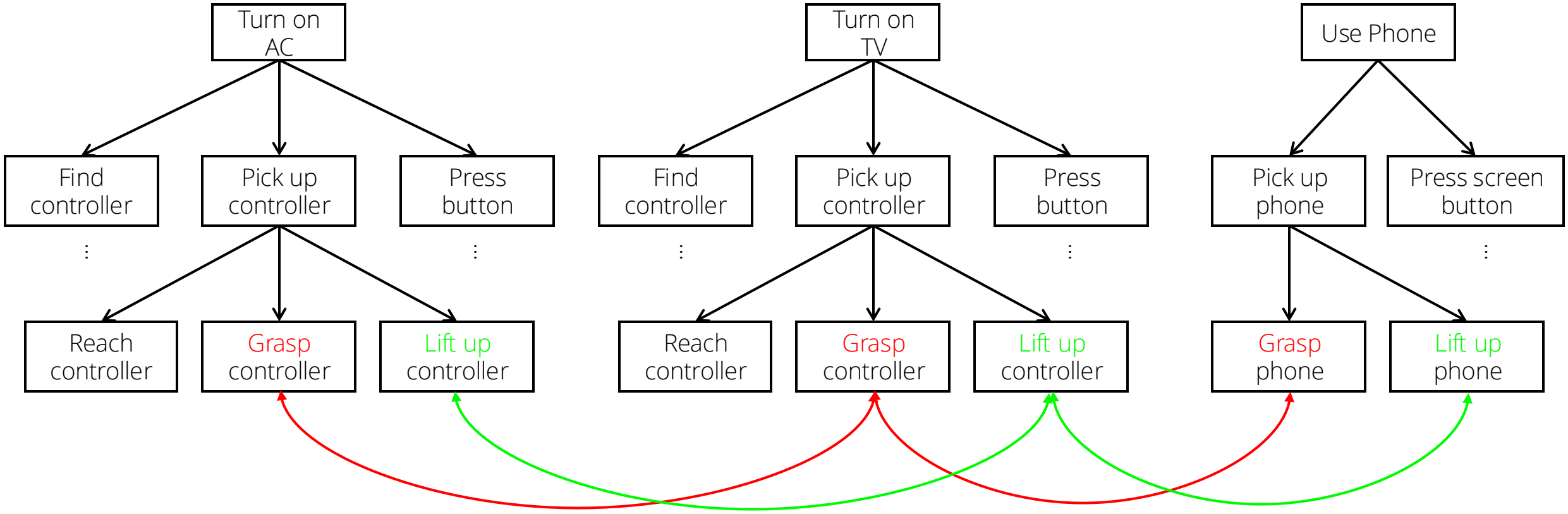
- More problems:
  - **Causal confusion**. Does your model look at the correct feature so that it generalizes to novel scene?
  - **Multi-modality behaviors**. What if there are multiple ways to achieve a goal? What is the better formulation of the model output?
  - **No generalization to new scenes.** The model performs poorly in unseen / slightly difference scene
  - **No generalization of new tasks.** Once your models learn to turn on the air conditioner, can it turn on the tv without/with minimal learning?

No generalization of new tasks

Even the largest and the best model don't generalize to unseen tasks...



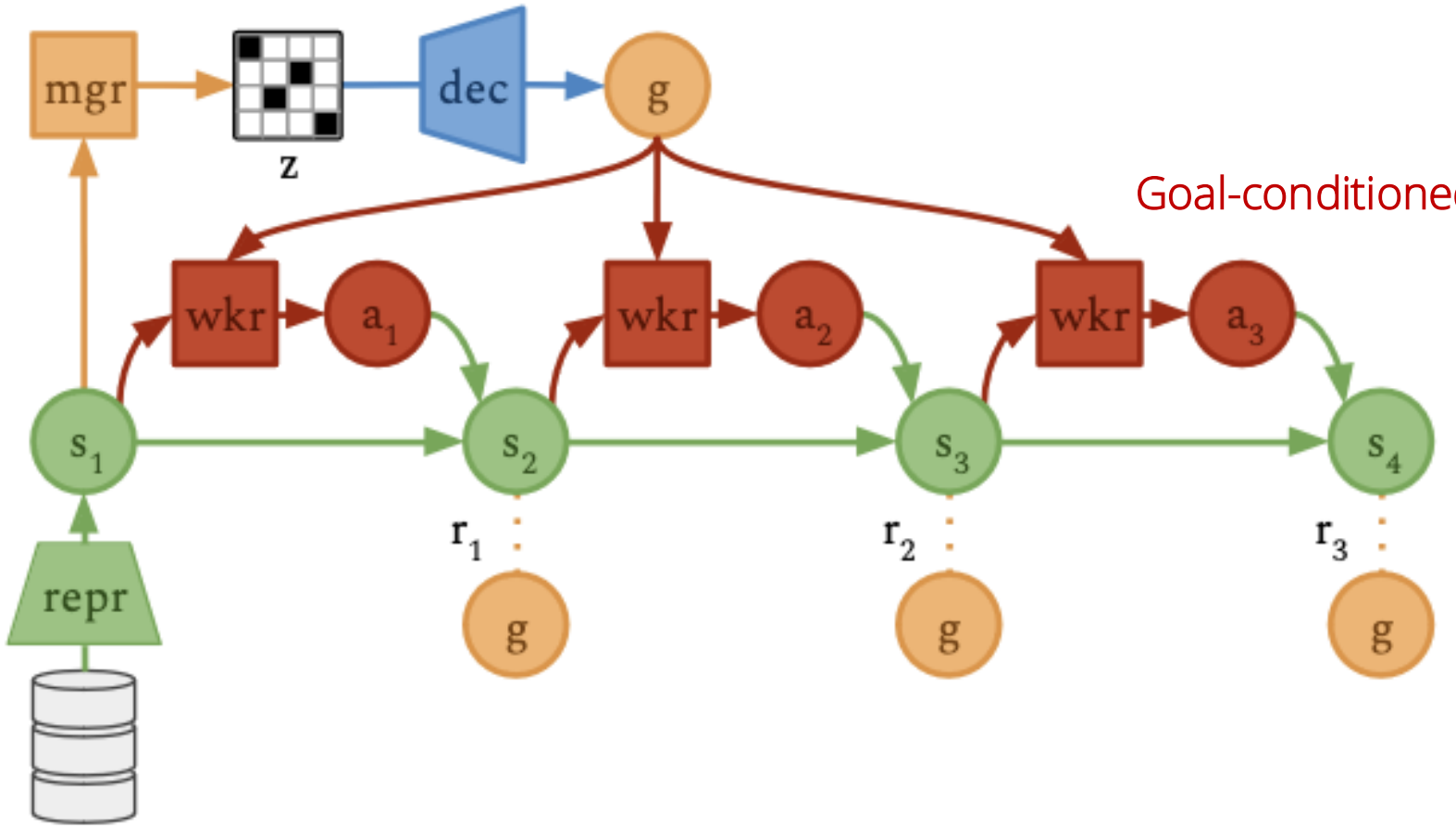
Knowledge Can be Shared Across Tasks



Lower-level skills are more similar among tasks
than higher-level ones

Learn to Achieve Subgoal

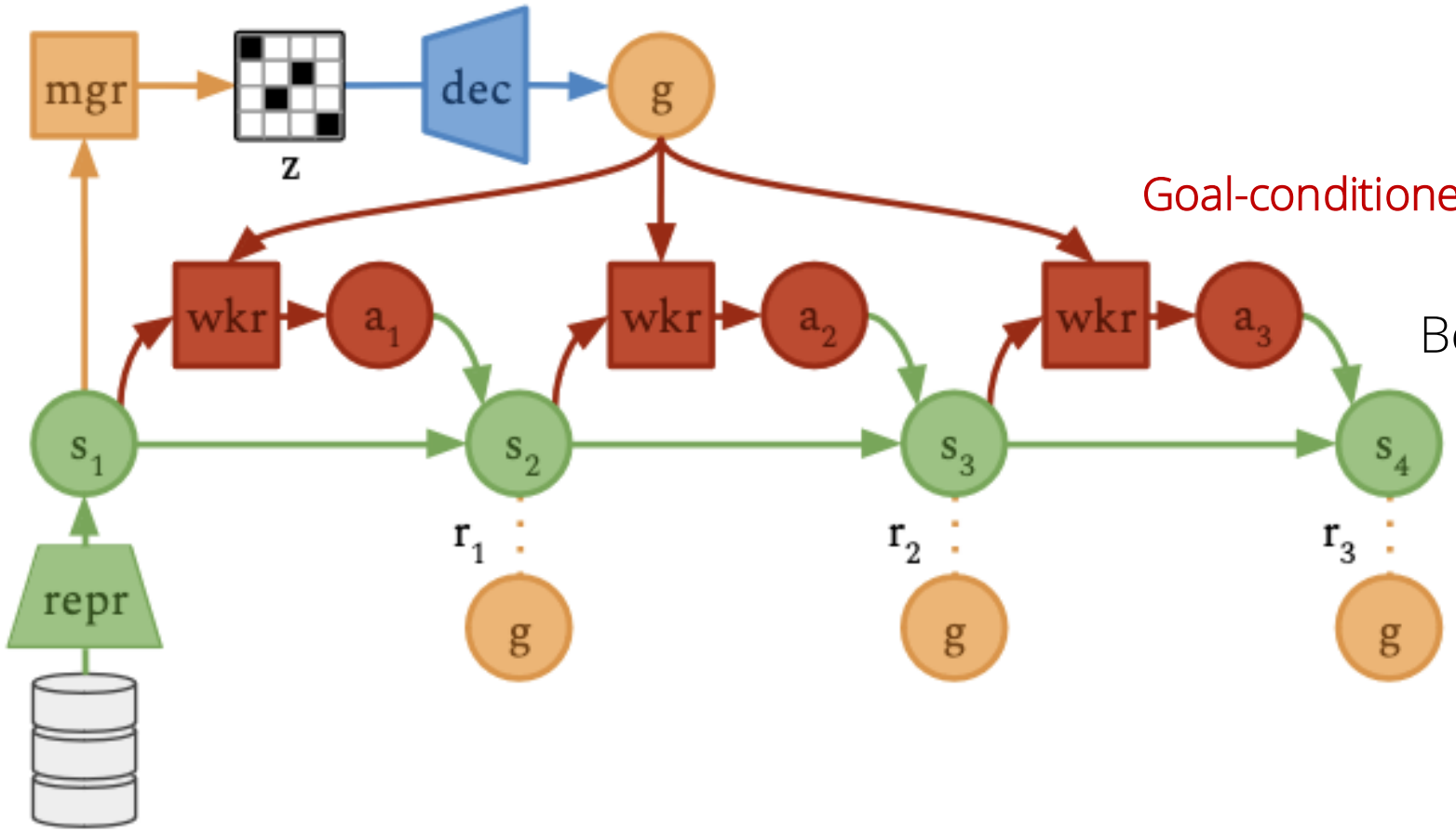
Subgoal E.g. Grasp controller



Goal-conditioned Action $\boldsymbol{p(a \mid s, g)}$

Learn to Achieve Subgoal

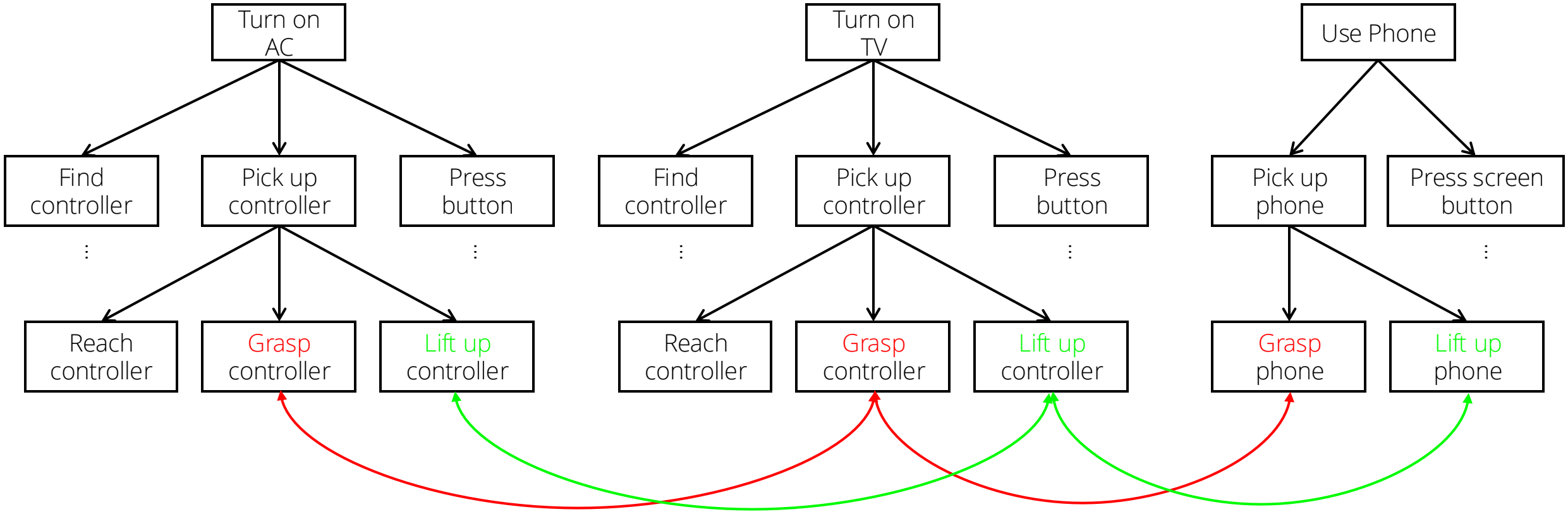
Subgoal E.g. Grasp controller



Goal-conditioned Action $\boldsymbol{p(a \mid s, g)}$

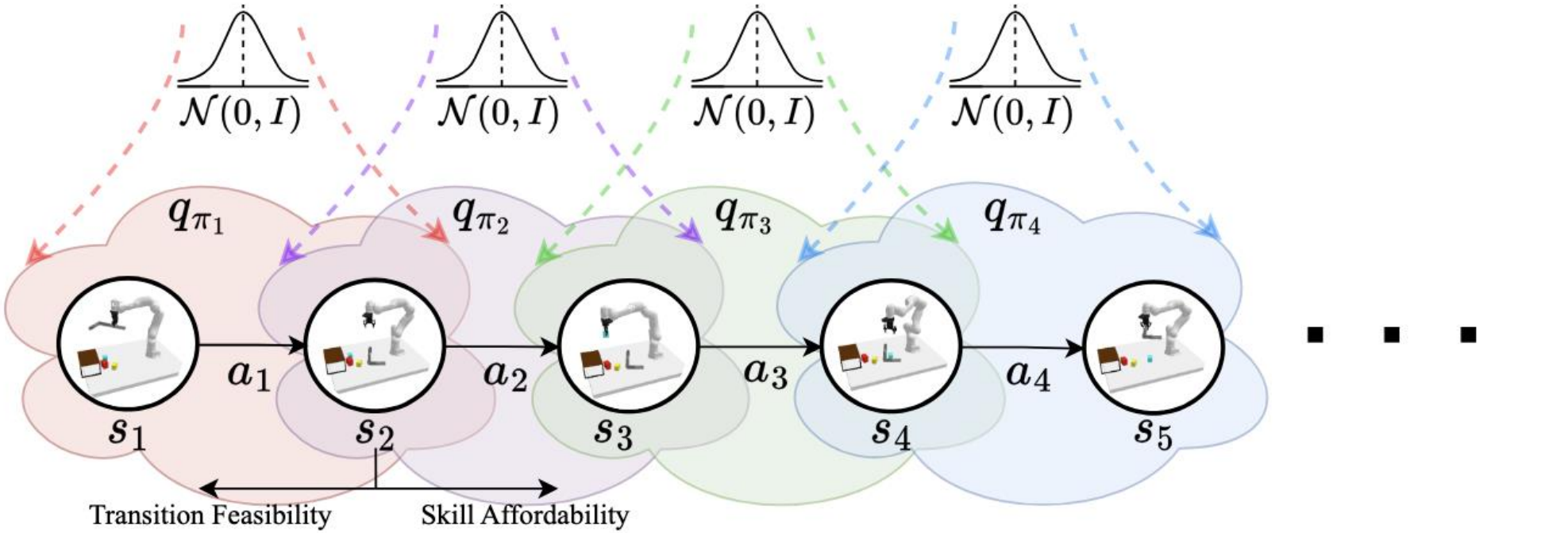
Better generalization if two tasks share the same subgoal

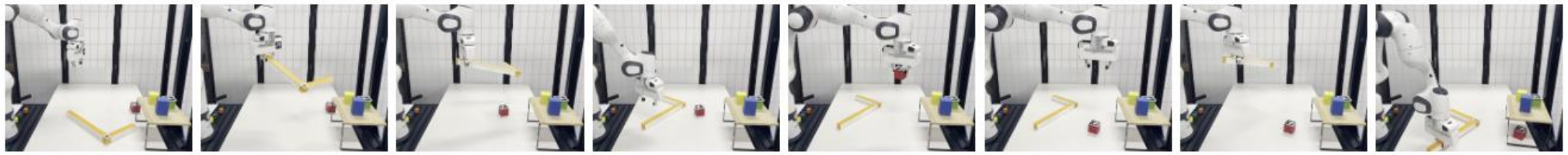
An Open Research Question: How to Generalize to Similar Skills?



Lower-level skills are more similar among tasks than higher-level ones

An Open Research Question: How to Learn New Tasks By Composing Multiple Learned Skills?





Generative Skill Chaining: Long-Horizon Skill Planning with Diffusion Models. Mishra et al.

Is Behavior Cloning + DAGGER All You Need?

- Of course, No!

- More problems:
  - **Causal confusion**. Does your model look at the correct feature so that it generalizes to novel scene?
  - **Multi-modality behaviors**. What if there are multiple ways to achieve a goal? What is the better formulation of the model output?
  - **No generalization to new scenes.** The model performs poorly in unseen / slightly difference scene
  - **No generalization of new tasks.** Once your models learn to turn on the air conditioner, can it turn on the tv without/with minimal learning?
  - **Sub-optimal demonstration.** What if the experts are mediocre? Can you be better than your teacher?