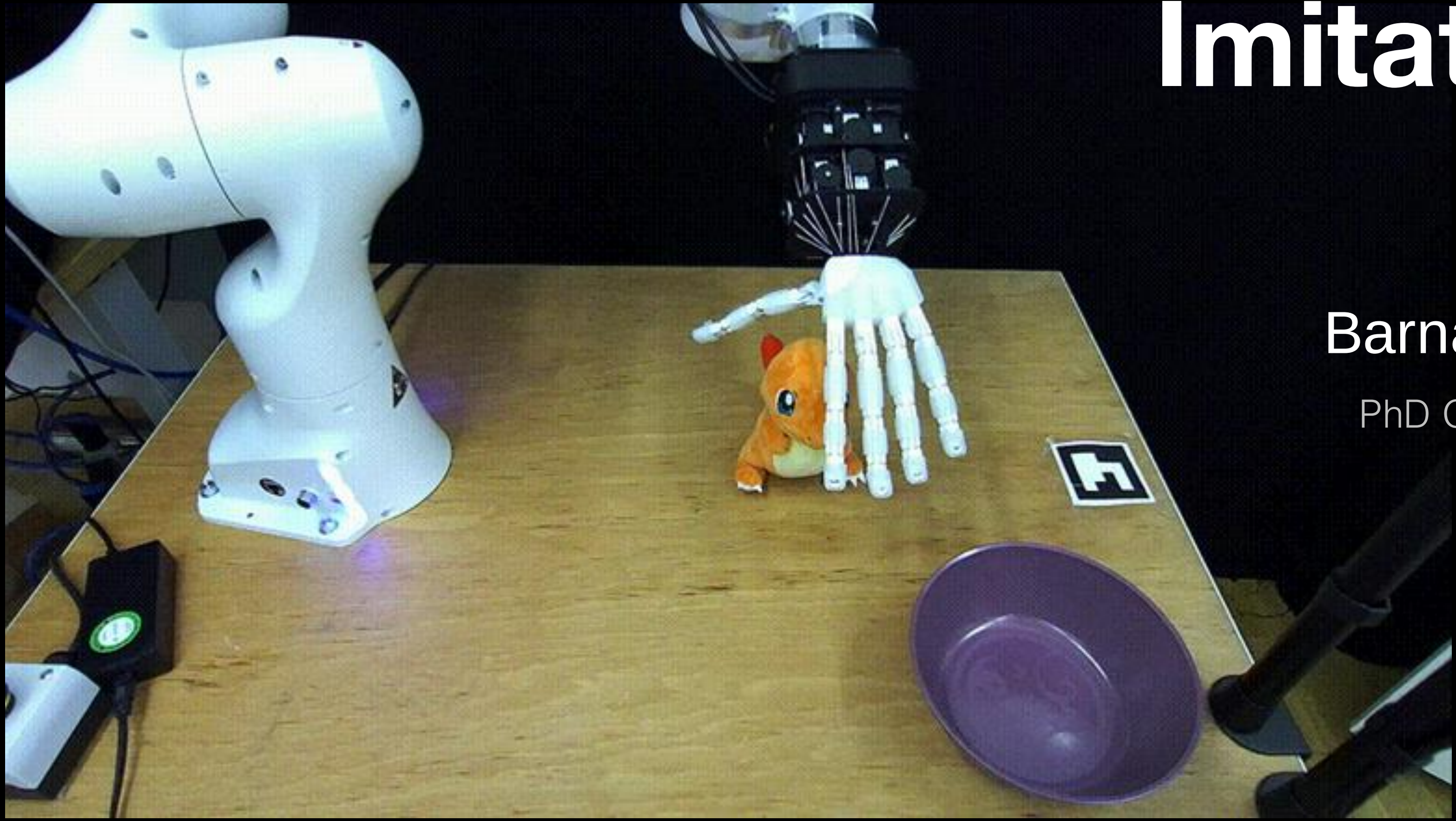
ETH zürich



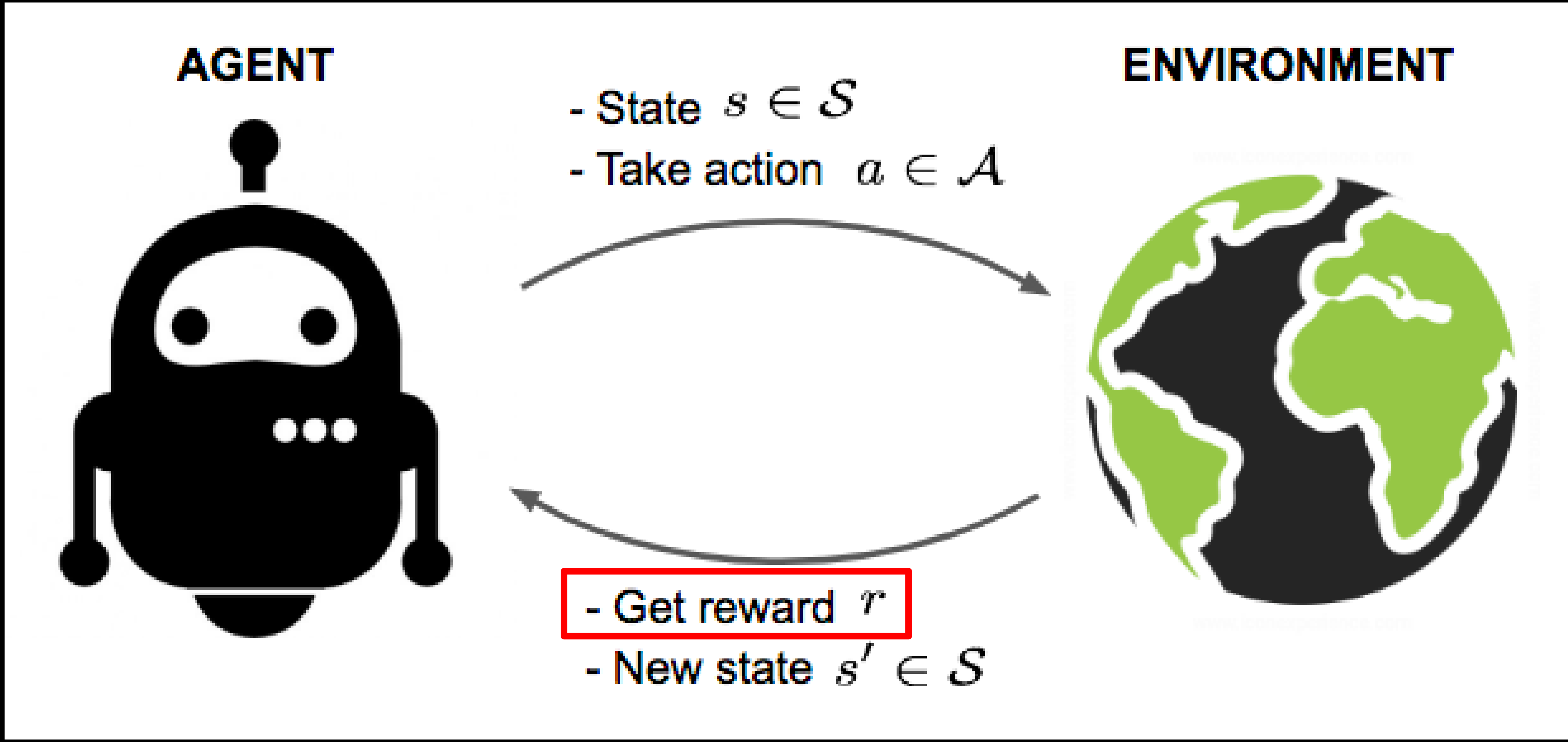
Imitation Learning

Barnabas Gavin Cangan

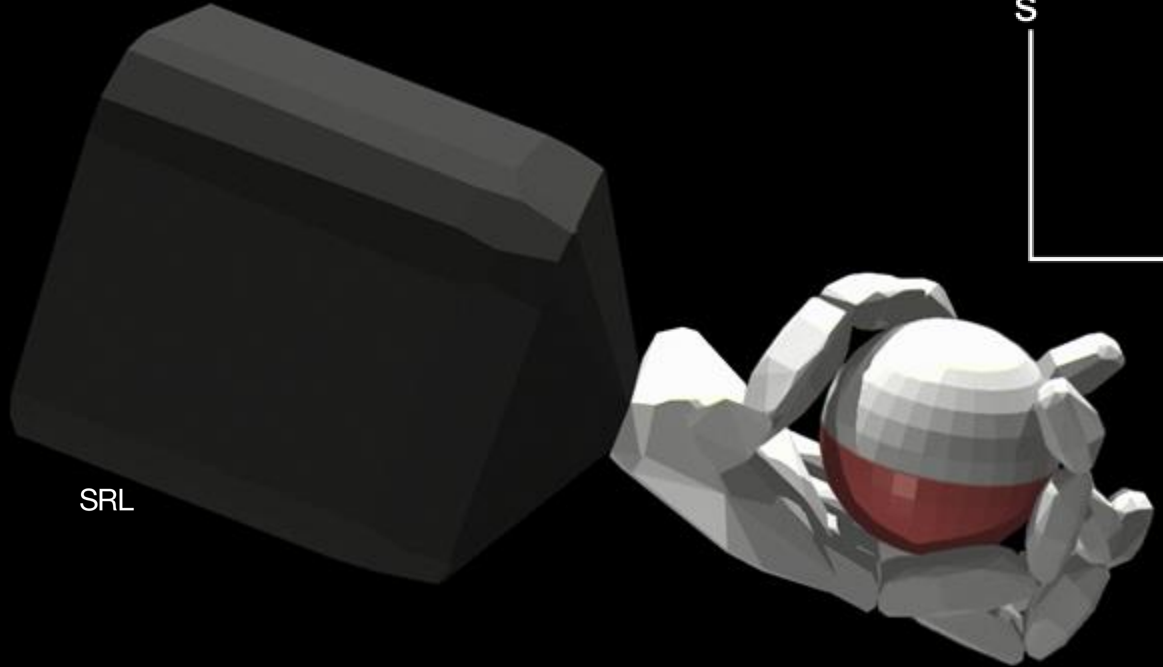
PhD Candidate, Soft Robotics Lab

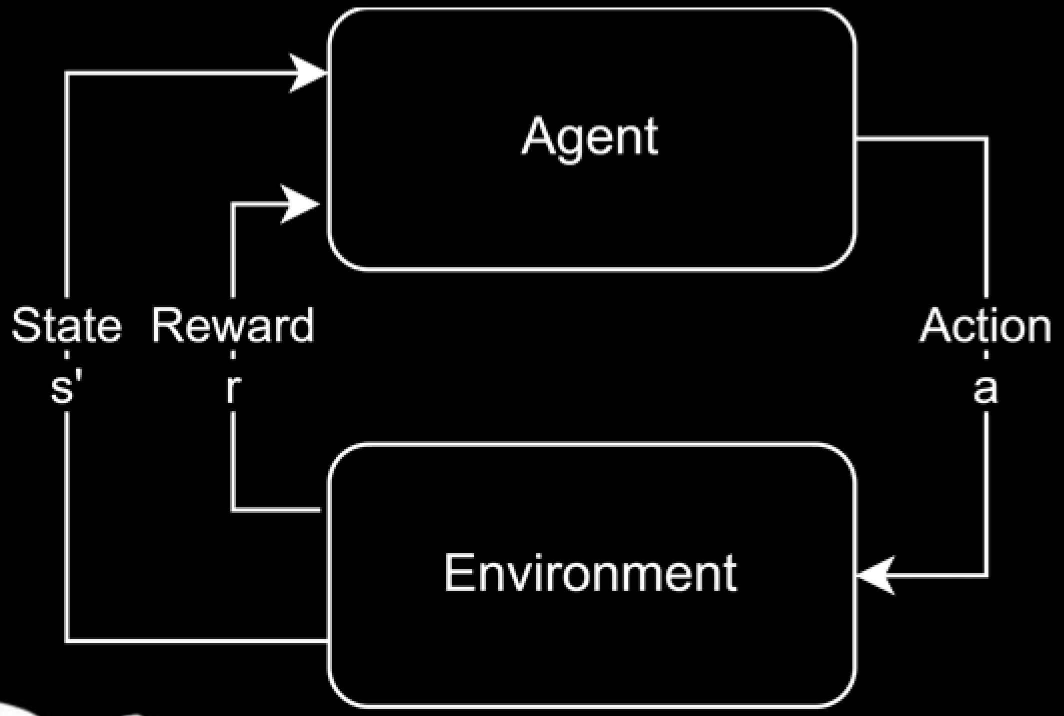
SoftRobotics Laboratory

Reinforcement Learning



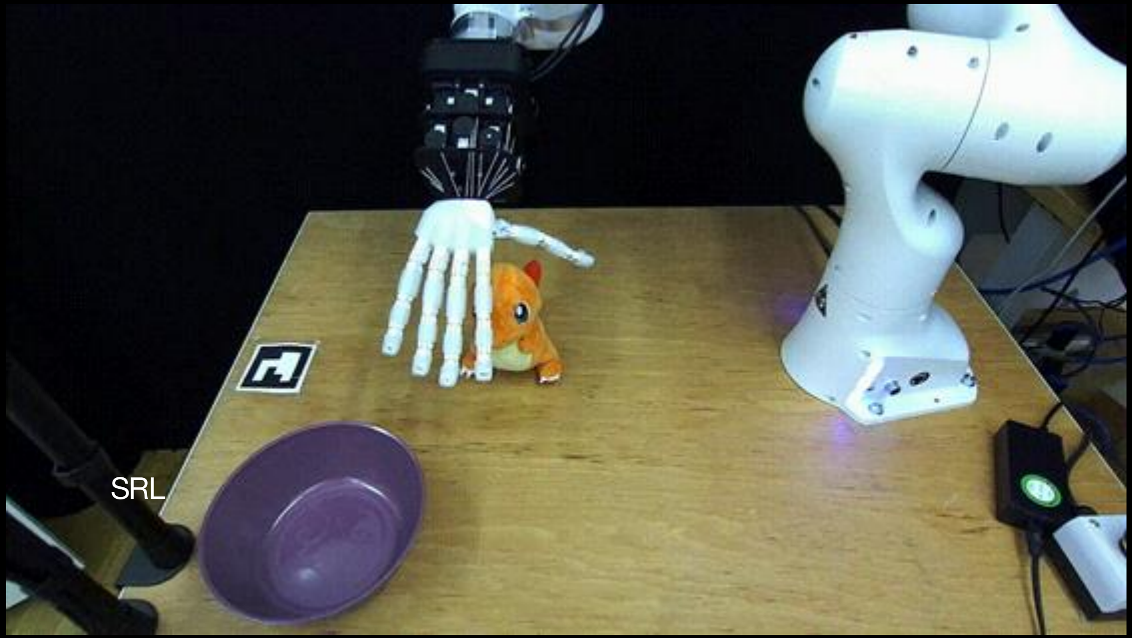
Altamimi, S., 2018. *QoE-Fair Video Streaming over DASH* (Doctoral dissertation, Université d'Ottawa/University of Ottawa).





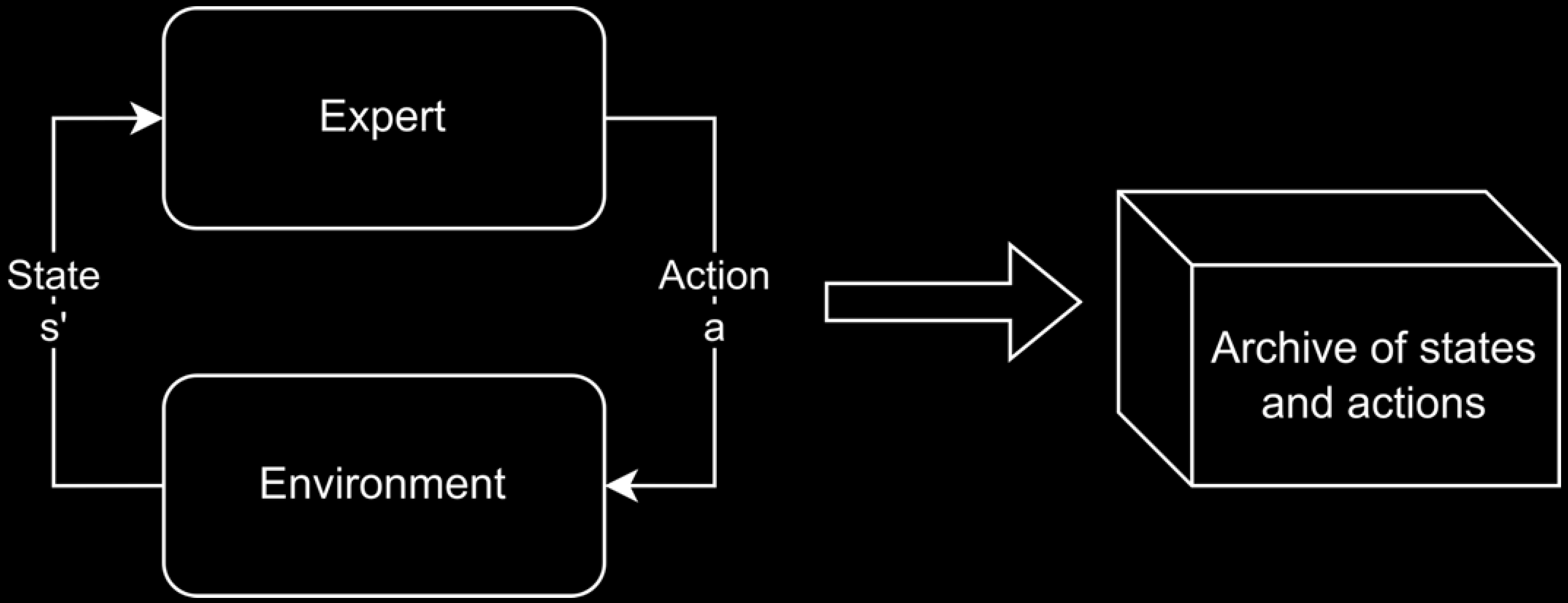
Reinforcement Learning (refer to tutorial…)

Differences with Reinforcement Learning



Reward function?

Differences with Reinforcement Learning



Vanilla Behavior Cloning

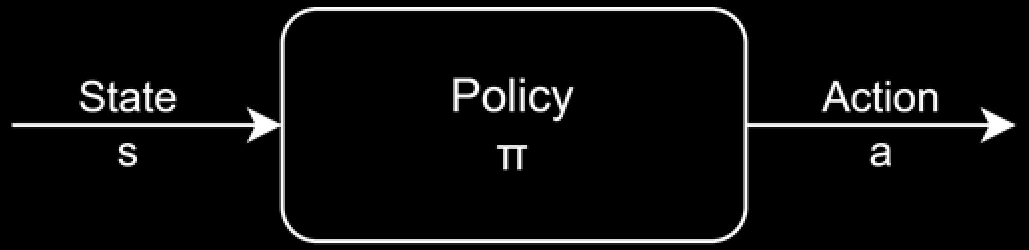
Data collection:
Record states, expert actions

Training:
learns mapping: observation → action
supervised learning to minimize:

$$L = ||a_{student}(o) - a_{expert}(o)||^2$$

Test:
observations → output actions

Limitations:
1. Very little generalization
2. Missing a bootstrapping equivalent



Why does this not work well? What are we missing?

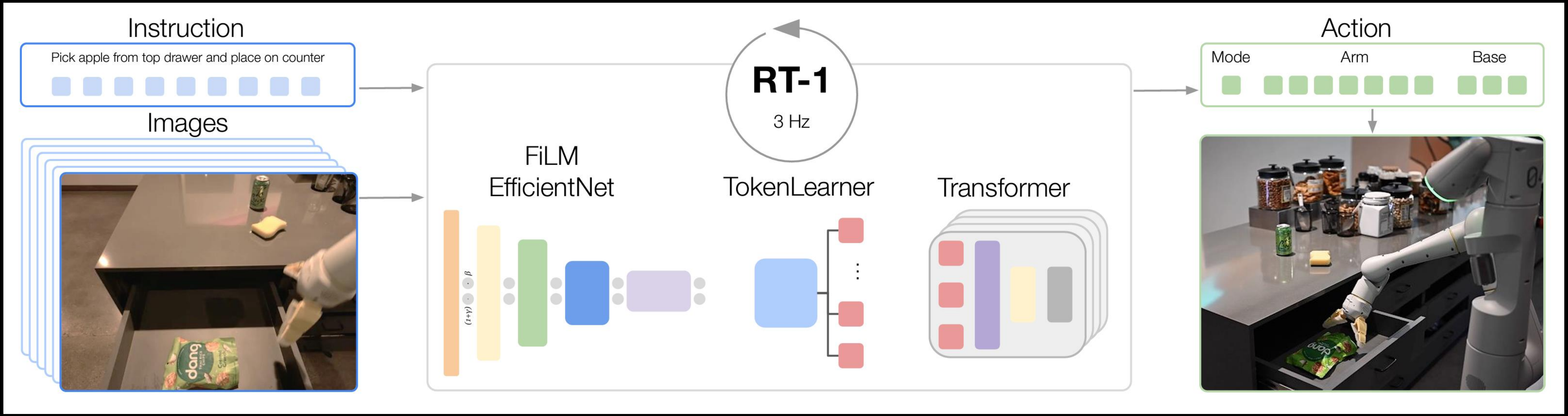
Robotics Transformer (RT-1)

3 Hz RGB images + language input + joint angles

700 tasks, 13 robots, 17 months → 130k episodes,

ETH zürich SoftRobotics Laboratory

Robotics Transformer (RT-1)



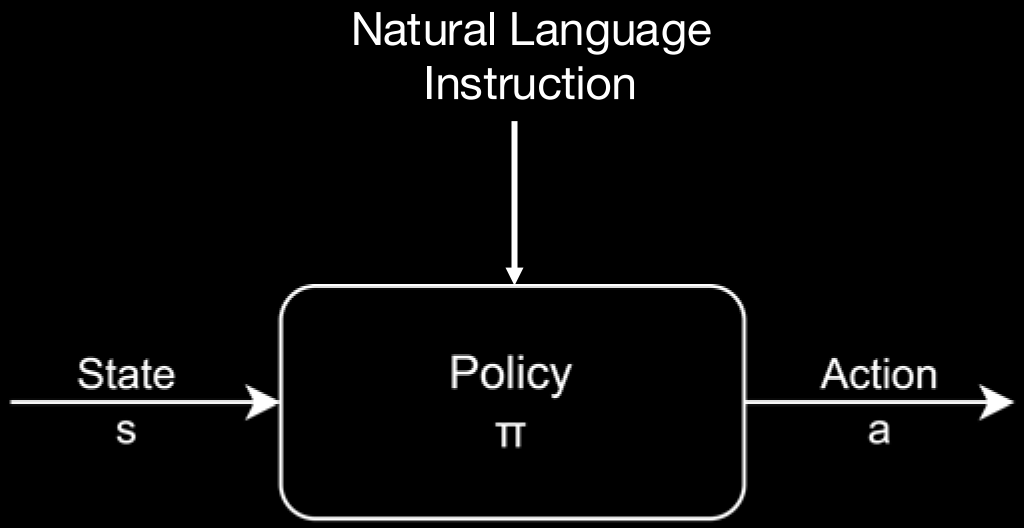
Robotics Transformer (RT-1)

Differences from "vanilla" BC

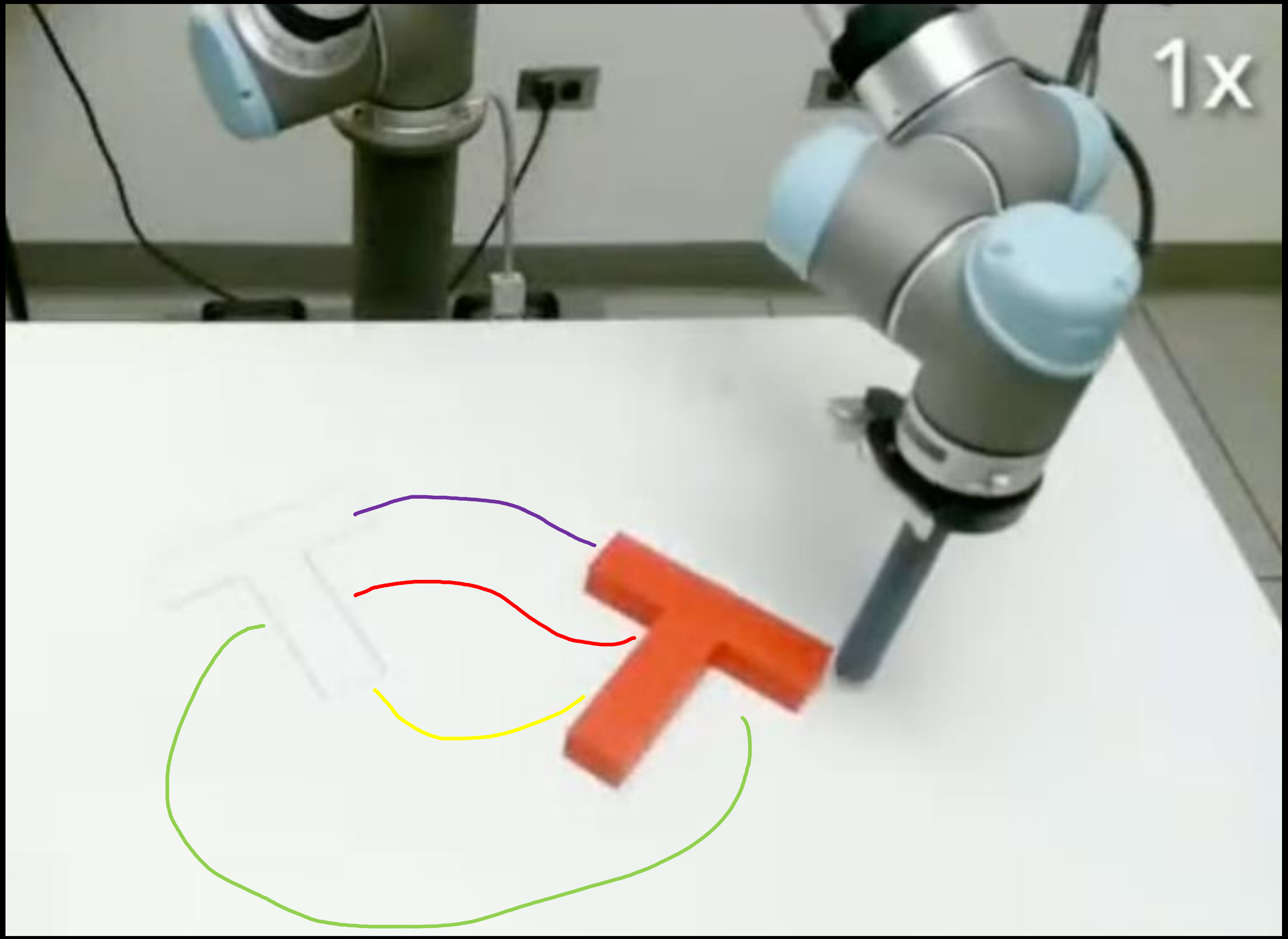
1. Uses transformers: manipulation as a sequence prediction task (like LLMs)
2. Vision Transformers ← state inputs + language inputs + images
3. Can be trained for hundreds of tasks → some transfer learning using shared attention
4. Temporal attention over historic states/actions → capture long-term dependencies
5. Action space discretized into tokens

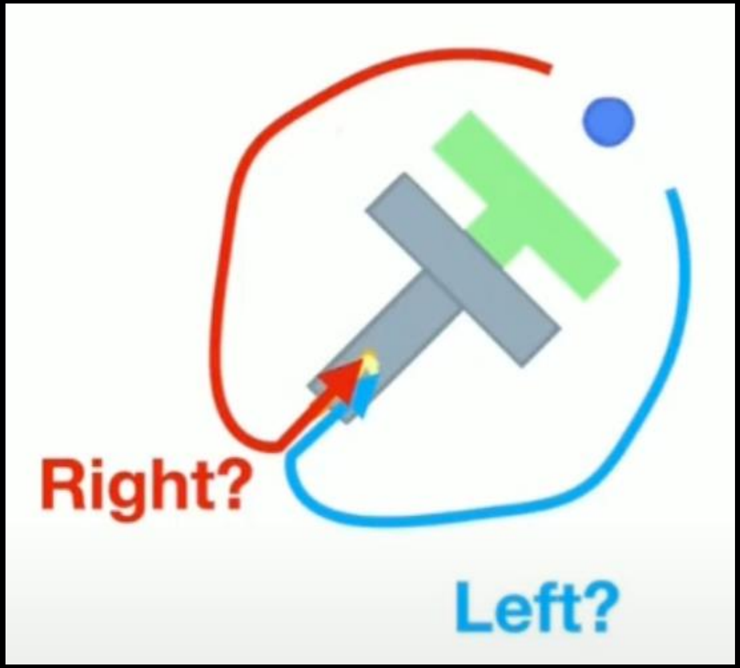
Limitations:

1. Massive dataset
2. "High quality" labeling

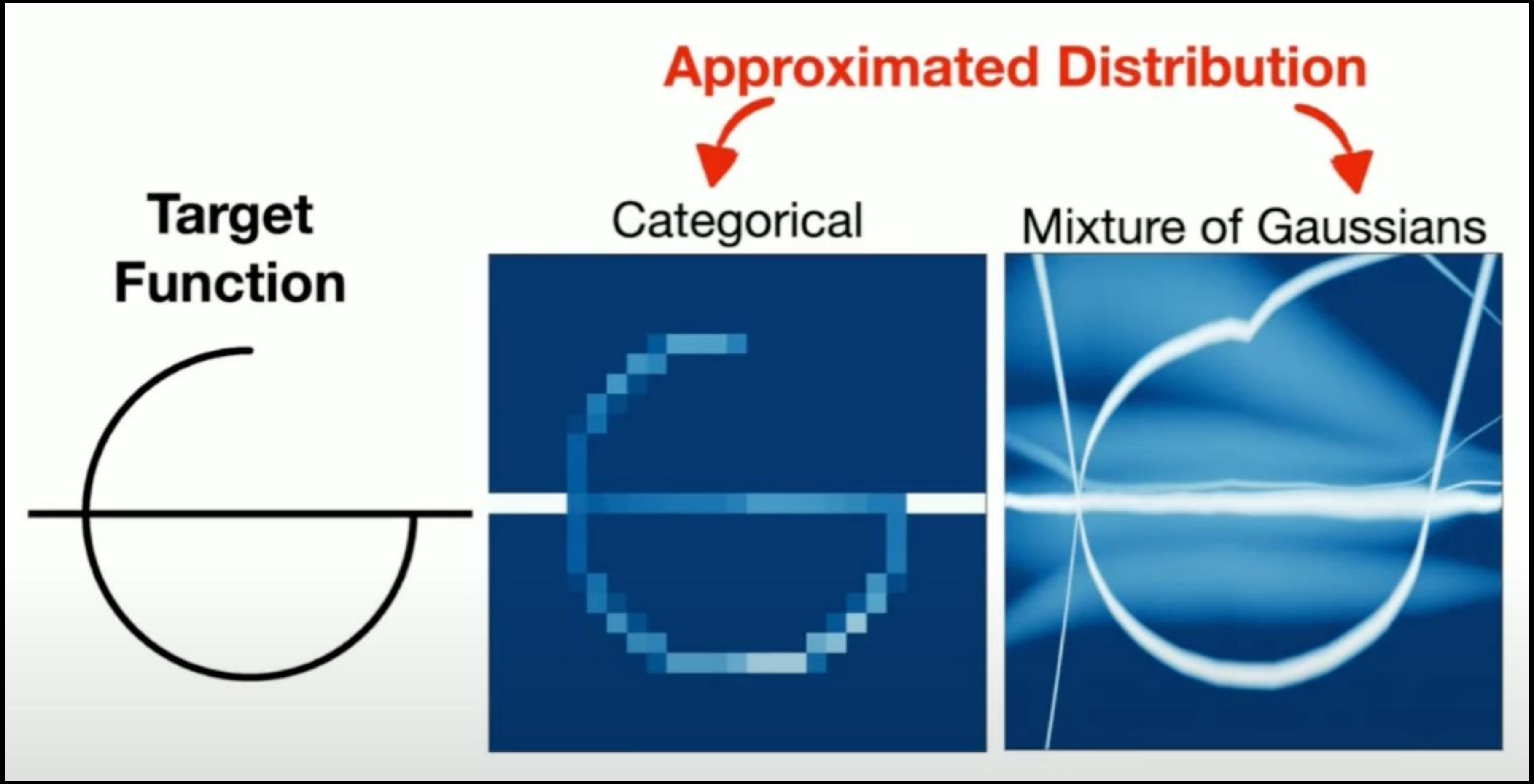


Multimodality





Multimodality

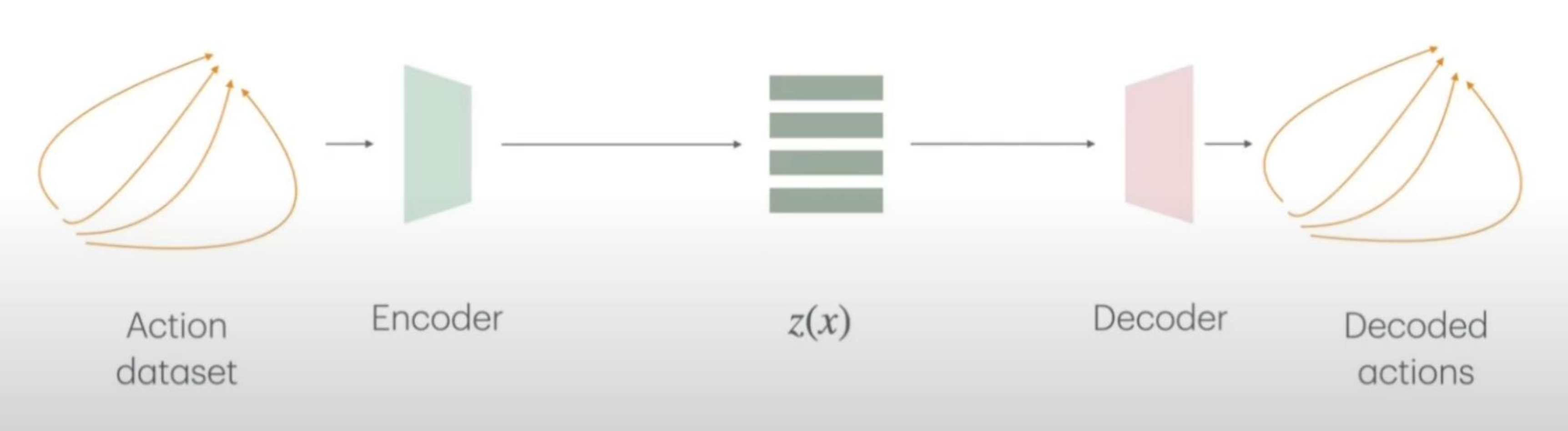


In what situation do you not care about multimodality?

Behavior Transformers (BeT)

Differences from RT-1

1. "modes" form clusters in latent space. Modes → style of doing a task
2. Keeps multimodal *(human recorded)* actions from collapsing



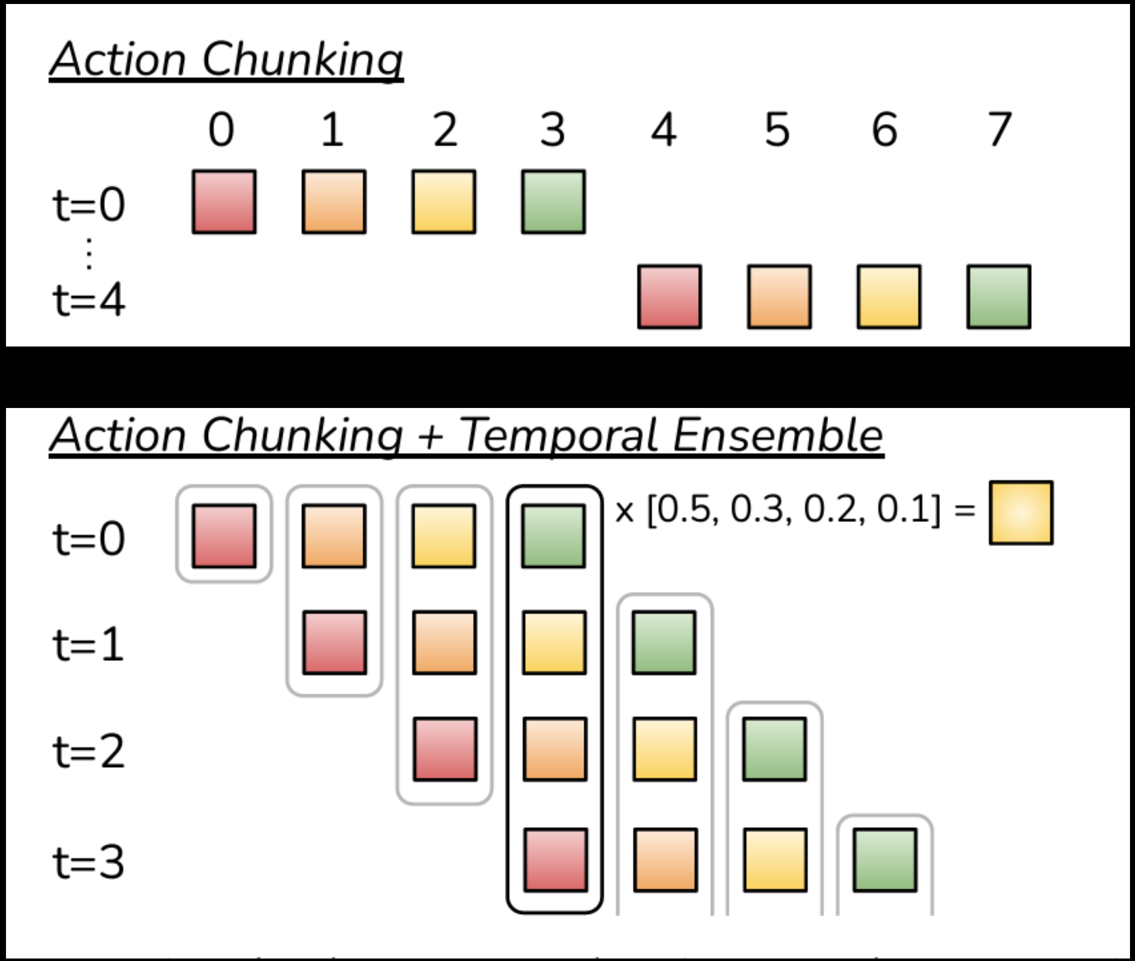
How do you select one mode vs another?

Action Chunking with Transformers

50 Hz RGB images + joint angles

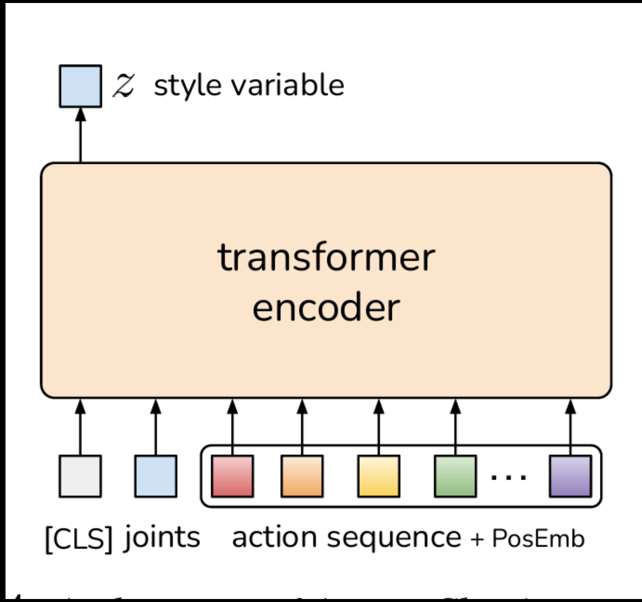
only 50 demonstrations!

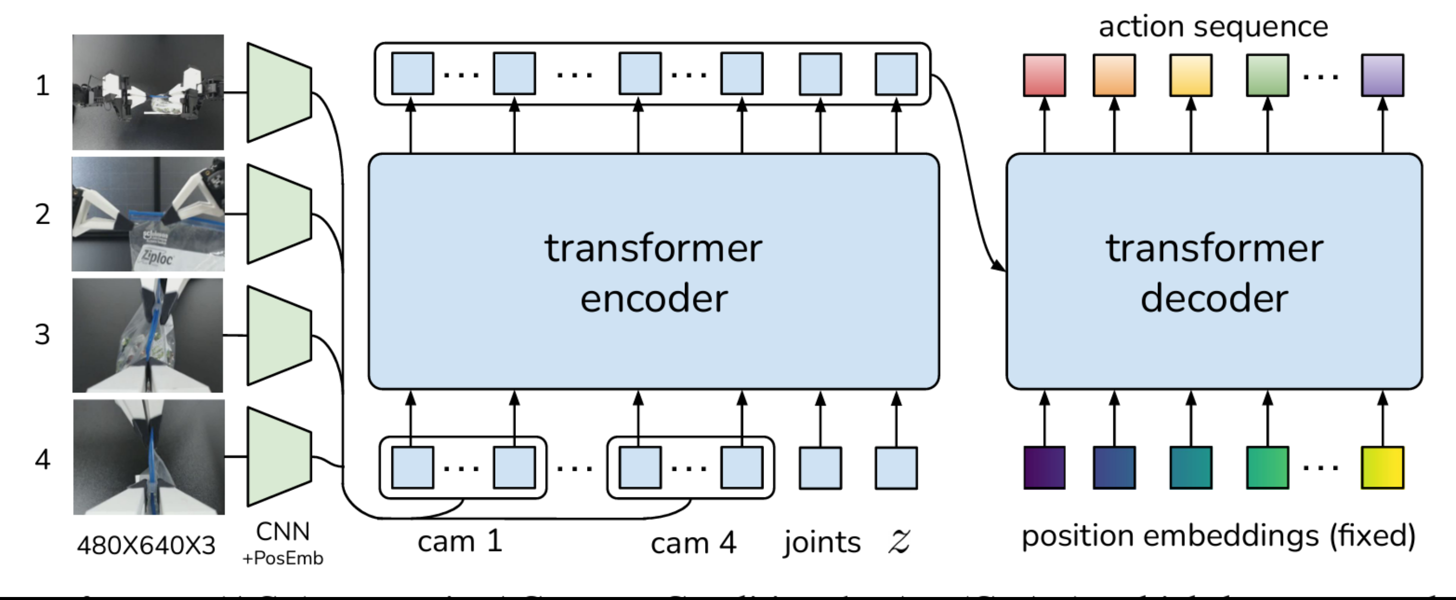
Action Chunking with Transformers



Action Chunking with Transformers

Architecture + Training





ETH zürich SoftRobotics Laboratory

Action Chunking with Transformers

Differences from previous works:

1. Predicts actions chunks, say k steps at a time
2. Chunk can be a meaningful action segment of a larger task
3. Temporal ensembling to minimize noise in action prediction

How does ACT handle multimodality?

Diffusion Policy

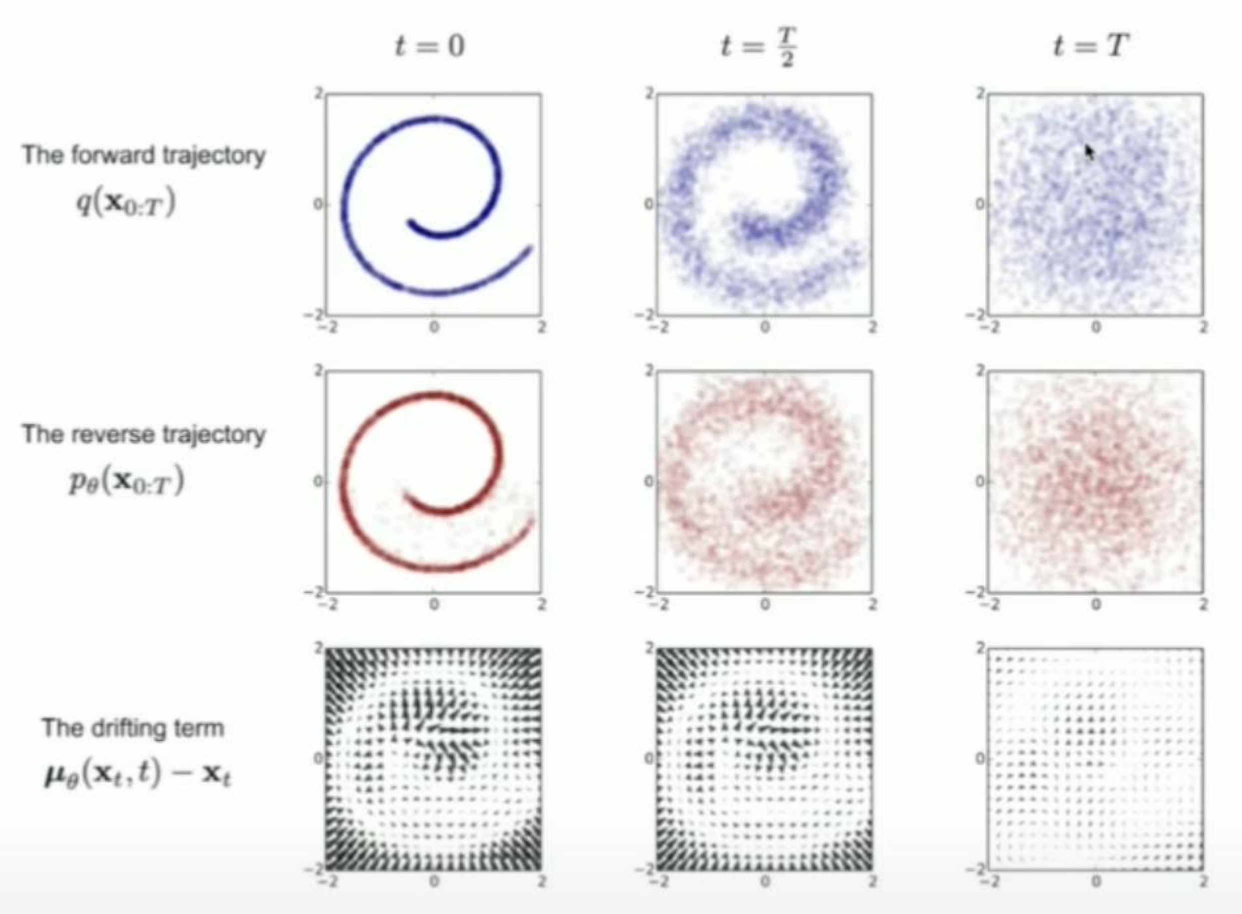
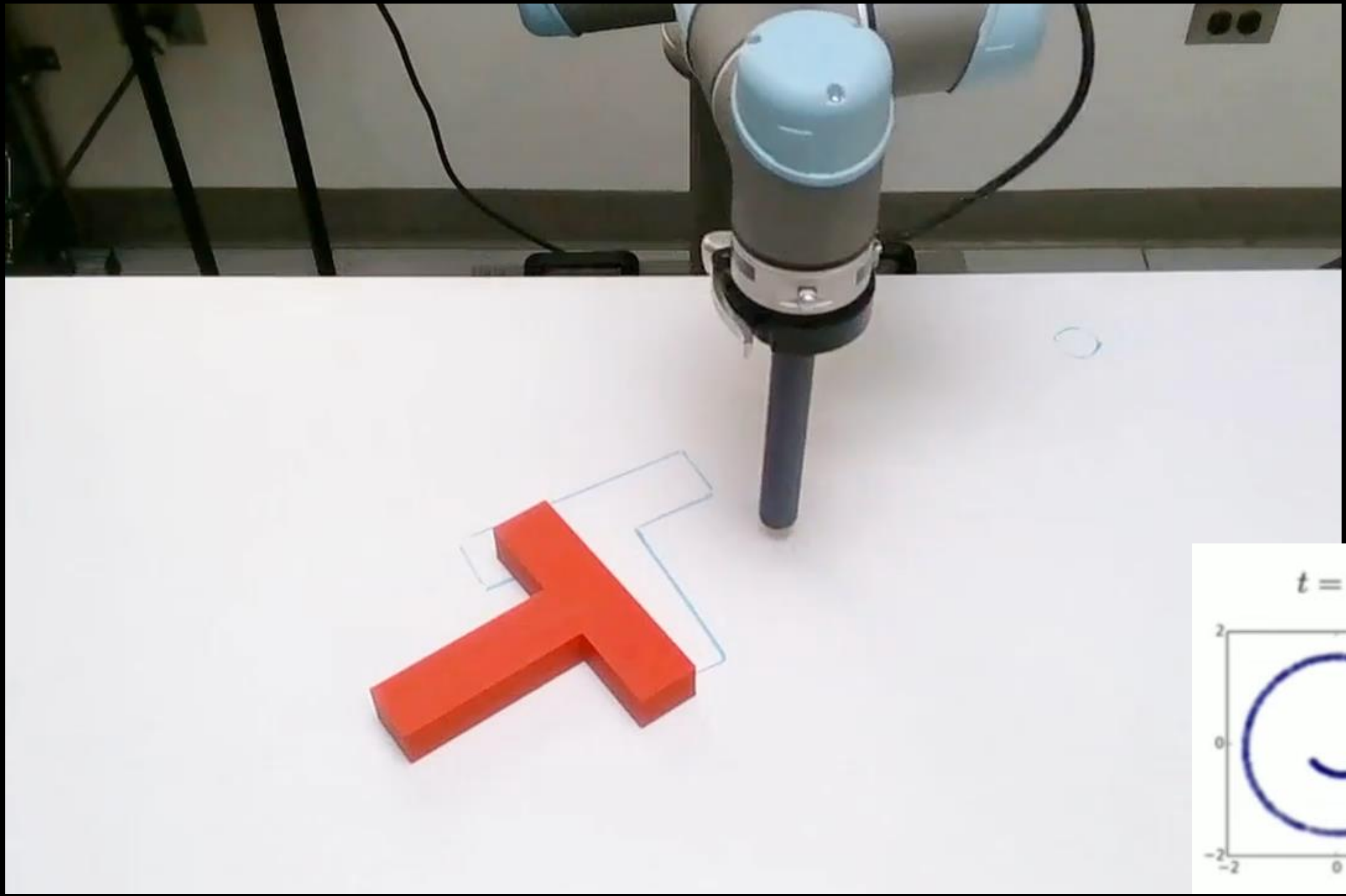
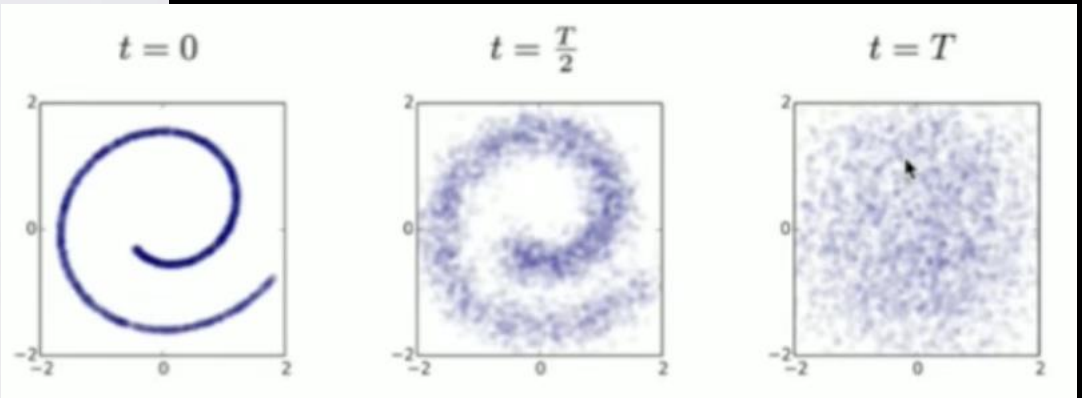


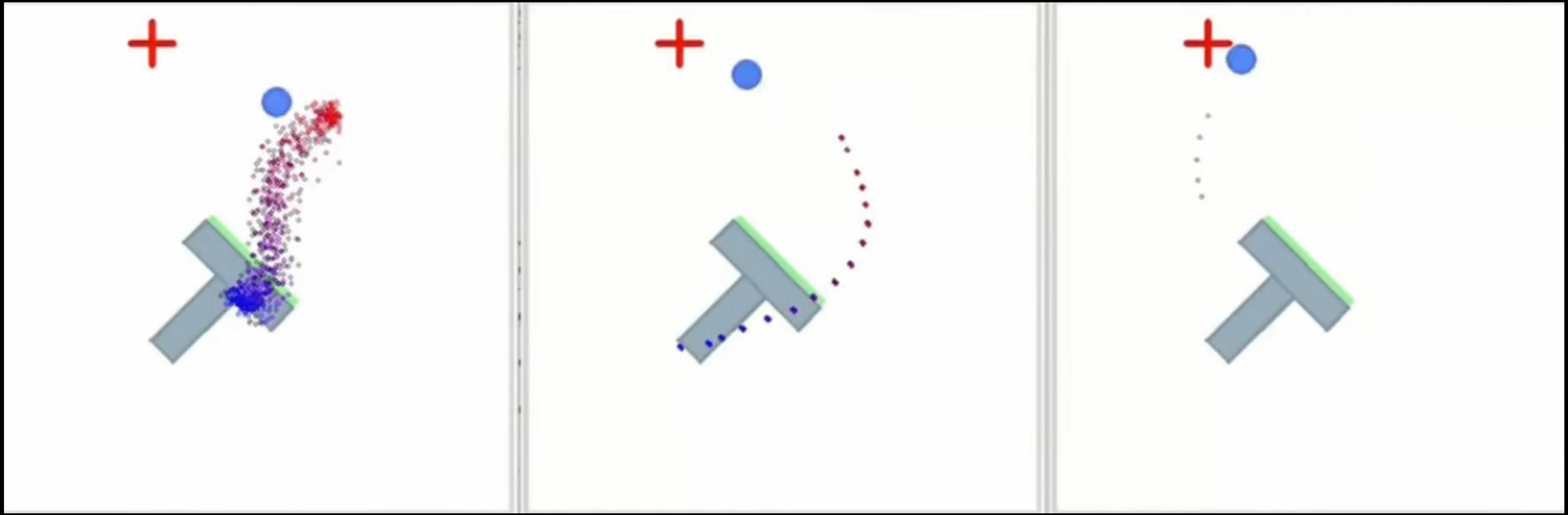
Fig. 3. An example of training a diffusion model for modeling a 2D swiss roll data. (Image source: Sohl-Dickstein et al., 2015)

Diffusion Policy





Comparison – Handling Multimodality in Data



Diffusion Policy ACT VQ-BeT

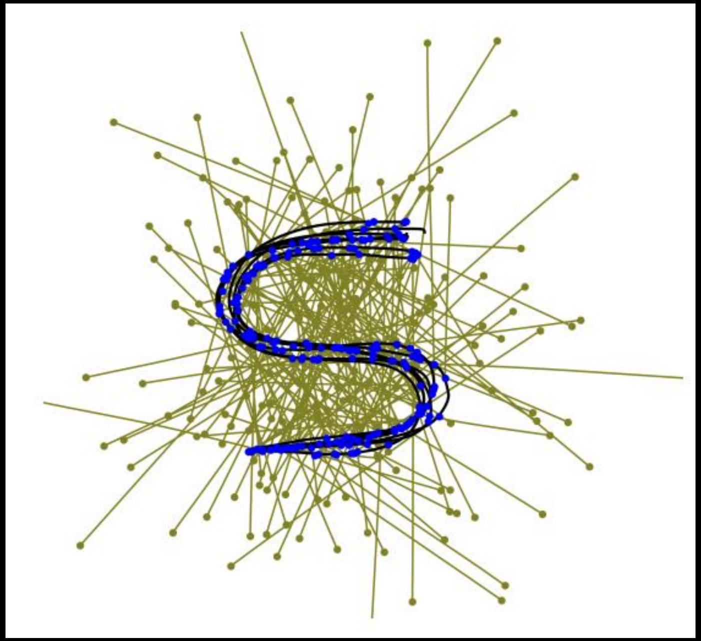
Flow Matching

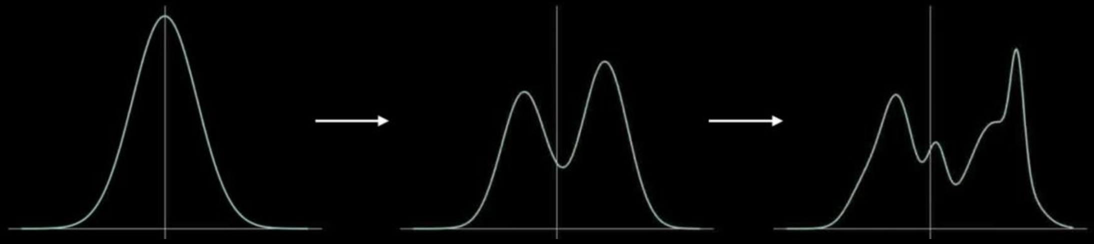
Speed up the inference further?

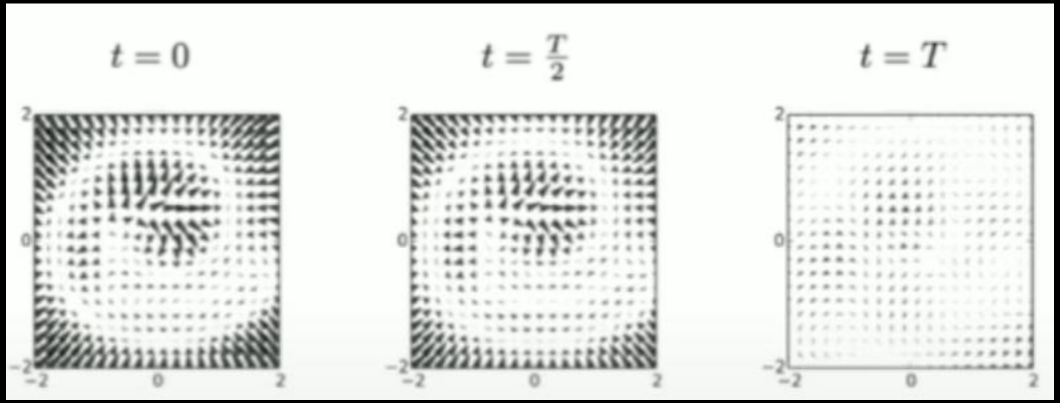
How do you make the trajectories smoother?

Improve training stability?

Flow Matching







Learning a deterministic continuous flow >> stochastic process

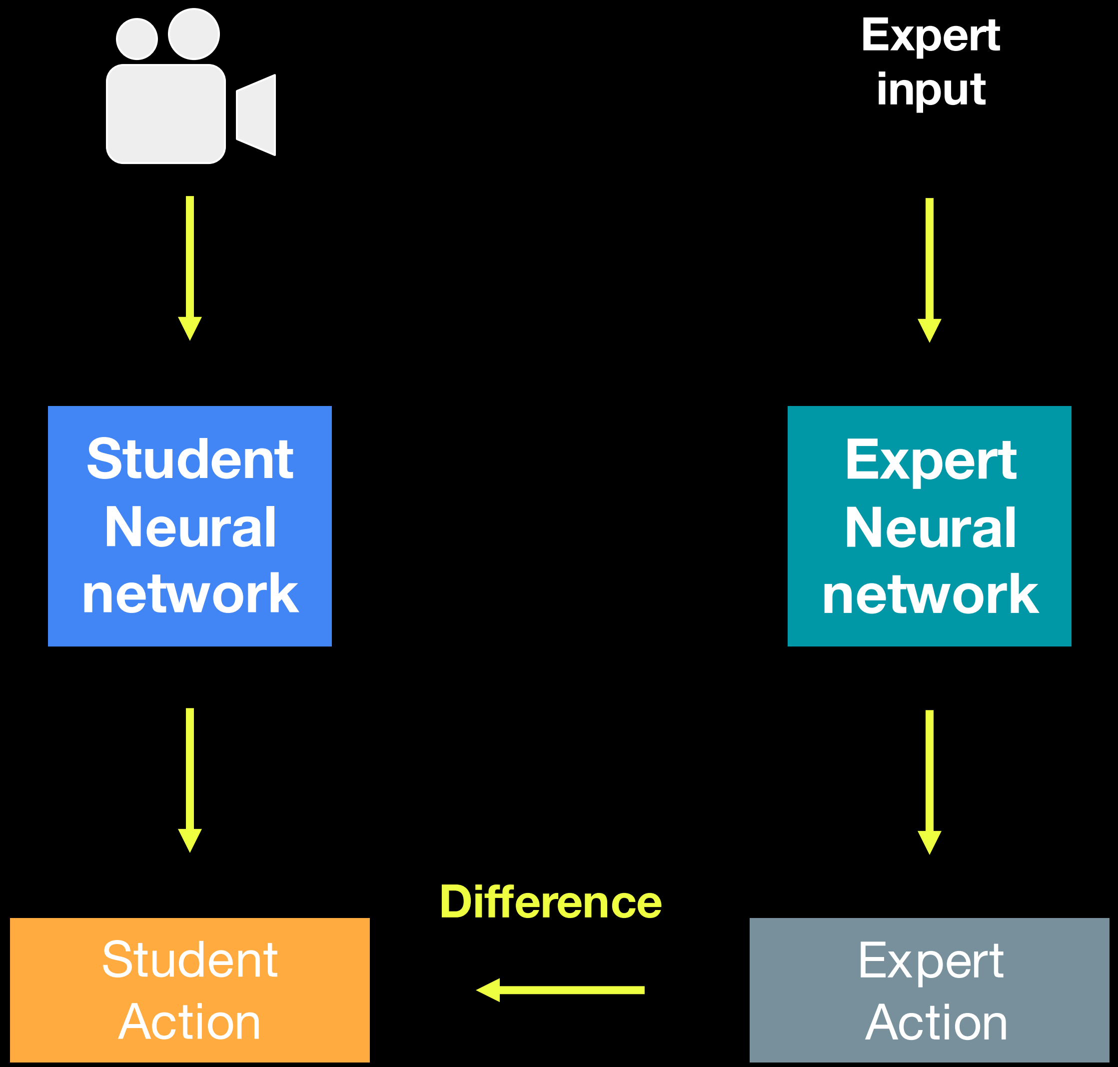
Flow Matching

diffusion policy,
200 steps, score 0.91

flow matching,
120 steps, score 1.0

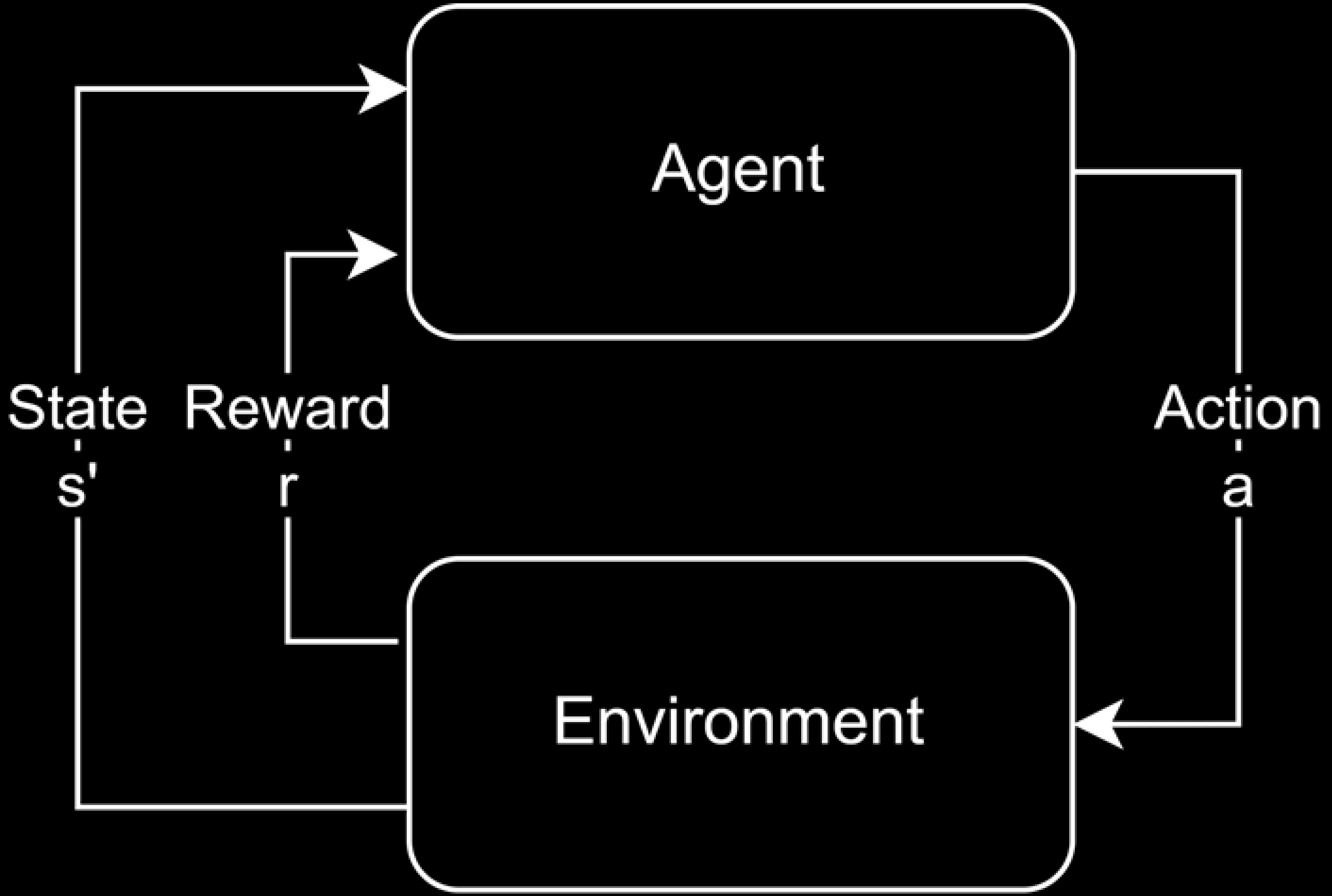
Backup slides

DAgger



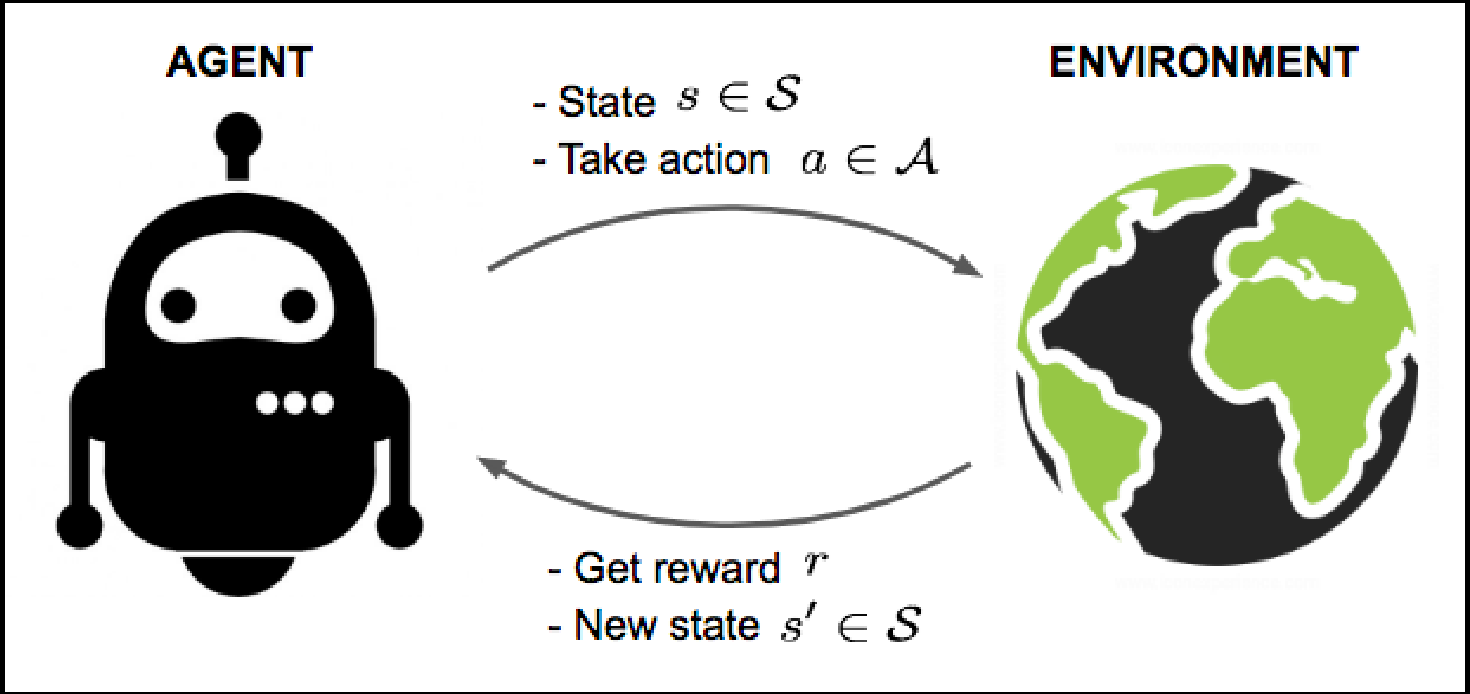
ETH zürich SoftRobotics Laboratory

Markov Process



Policy

High level intuition



Altamimi, S., 2018. *QoE-Fair Video Streaming over DASH* (Doctoral dissertation, Université d'Ottawa/University of Ottawa).

Reward and Discount Factor

Cumulative reward

$$R_t = \sum_{k=t}^{T} r_k\left(s_k, a_k\right)$$

Reward at timestep k

Action at timestep k

State at timestep k

Value and Q functions

Value function in states given policy π

Expected cumulative reward

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}\right] \qquad \forall s \in \mathbb{S}$$

Set of all possible states

Value and Q functions

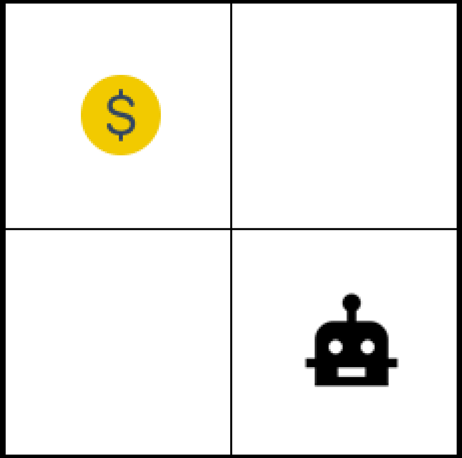
Q function in state s and action a given policy π

Expected cumulative reward

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty}\gamma^{k}r_{t+k+1} \mid s_t = s, a_t = a\right]$$

Given that in state s action a is applied

Value and Q functions



Original map

1.0	0.75
0.75	0.5

Value function for each cell

↑ 1.0	↓ 1.0	↑ 0.0	↓ 0.25
← 1.0	→ 1.0	← 0.75	→ 0.0
↑ 0.75	↓ 0.0	↑ 0.5	↓ 0.0
← 0.0	→ 0.25	← 0.5	→ 0.0

Q function for each cell and action

What is the point of a Q function & value function?