Week 15

Embodied AI & Robot Vision

[ECEA0649/ECE40049] Deep Learning for Image Processing | Spring 2026

Kihyun Na (Research Professor)
BK21 AI Education and Research Group &
Institute for Information and Communication Technology,
Handong Global University

HANDONG GLOBAL UNIVERSITY

Curriculum

** Adjusted based on survey results*

Wk 1	OT + Introduction	**Wk 9**	Foundation Models (CLIP, SAM) + Paper #1
Wk 2	DL Fundamentals Review	**Wk 10**	Diffusion Models + Paper #2
Wk 3	CNN	**Wk 11**	Diffusion Models (2) + Paper #3
Wk 4	From Sequence Modeling to Transformer	**Wk 12**	Vision-Language Models + Paper #4
Wk 5	Transformer in Vision	**Wk 13**	VLM Applications (Paper #5)
Wk 6	Detection & Segmentation	**Wk 14**	Video Understanding + Paper #6
Wk 7	Self-Supervised Learning	**Wk 15**	Embodied AI & Robot Vision + Paper #7
Wk 8	Review Literacy + Role Explanation	**Wk 16**	Miniconference (Final Project)

Lecture Lecture + Paper Miniconference

Today's Agenda

01 **From Perception to Action** — 5 min

What changes when output becomes action

02 **How Robots Decide** — 15 min

Questions & Taxonomy

03 **Imitation Learning & Diffusion Policy** — 15 min

Behavioral Cloning, Compounding error, action as generation

04 **VLA & Learning from Experience** — 15 min

From RT-1 to π, RL-finetuning

Part 1

From Perception to Action

What changes when output becomes action

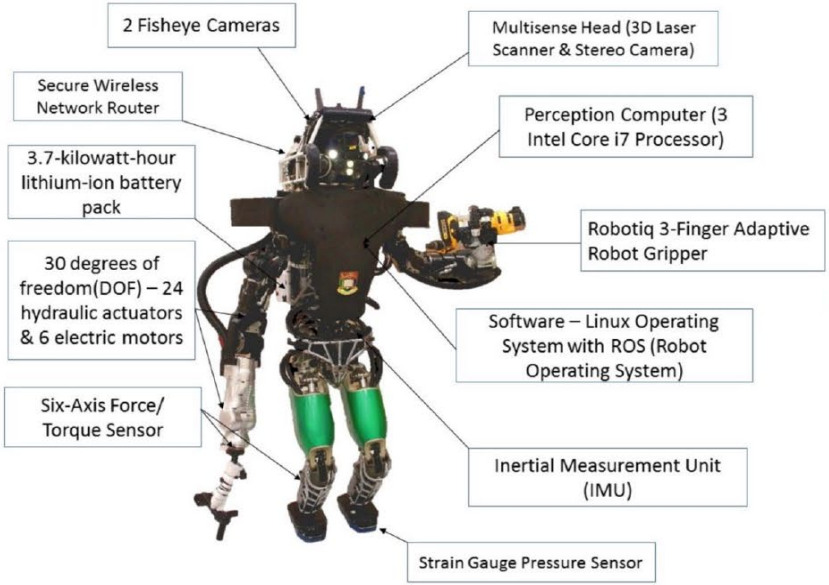
Perception for Robots

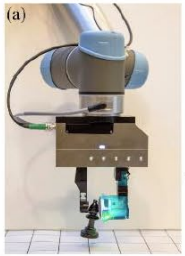
Robot vision is embodied, active, and environmentally situated!

- Embodied: Robots have physical bodies and experience the world

- Active: Robots are active perceivers.

- Situated: Robots are situated in the world.





[Source: HKU Advanced Robotics Laboratory]

What Changes with Action?

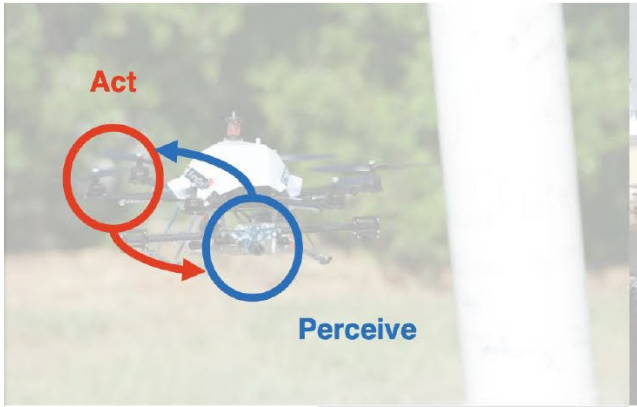
From open-loop prediction to a closed loop

Perception: x → y. The prediction is graded, the world stays put, samples are i.i.d.

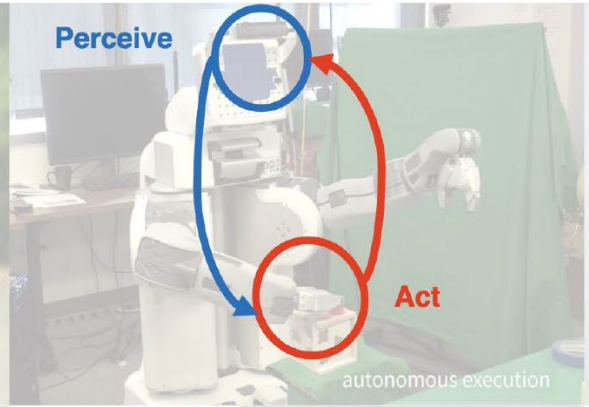
Action: o(t) → a(t) → the world changes → o(t+1). Your output becomes your next input — a closed loop.

Consequence 1 (i.i.d. breaks) Your mistakes change your data distribution, so errors compound instead of averaging out.

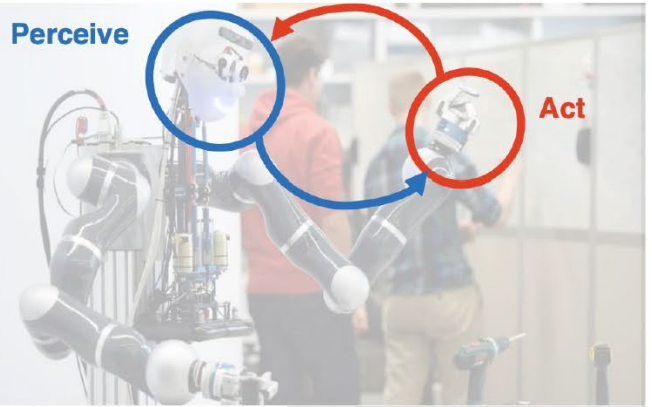
Consequence 2 (no labels) There is no 'correct action' dataset: only demonstrations to imitate, or rewards to chase.



[Sa et al. IROS 2014]



[Levine et al. JMLR 2016]



[Bohg et al. ICRA 2018]

Today's question: what happens when we act on what we see?

image → label becomes observation (perception) → action

From This Course to Robot Learning

Every perception tool reappears on the robot

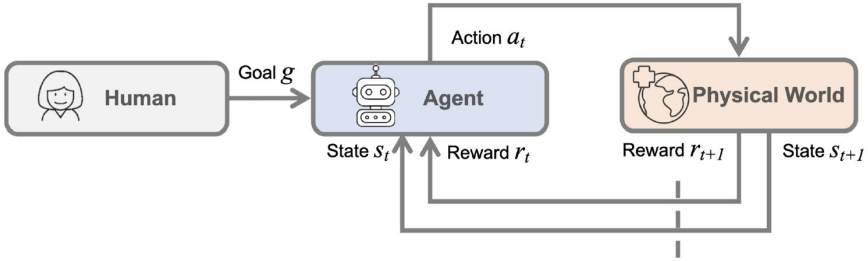
1	**Wk 7 · Self-Supervised Learning**	→ world models that learn from unlabeled video (V-JEPA)
2	**Wk 9 · Foundation Models**	→ the vision encoder inside VLA policies
3	**Wk 10–11 · Diffusion Models**	→ action experts that generate motion (Diffusion Policy, π0)
4	**Wk 12–13 · Vision-Language Models**	→ the backbone of Vision-Language-Action models
5	**Wk 14 · Video Understanding**	→ predicting how the world evolves (V-JEPA 2)

Part 2

How Robots Decide

Questions & Taxonomy

Terminology: Reinforcement Learning



state $\mathbf{s}_t$ - the state of the "world" at time t

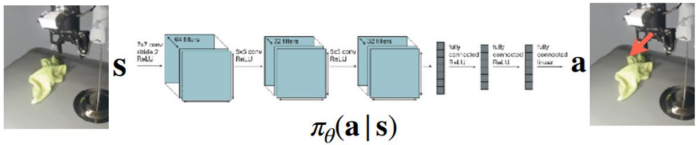
observation $\mathbf{o}_t$ - what the agent observes at time t

action $\mathbf{a}_t$ - the decision taken at time t

trajectory τ - sequence of states/observations and actions

$$(\mathbf{s}_1, \mathbf{a}_1, \mathbf{s}_2, \mathbf{a}_2, \ldots, \mathbf{s}_T, \mathbf{a}_T)$$

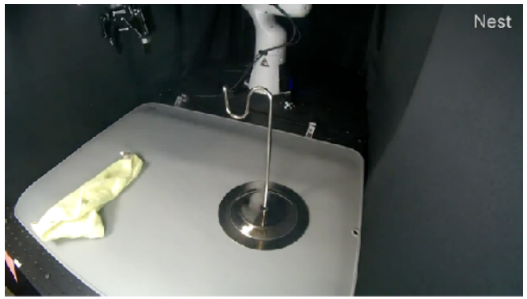
reward function $r(\mathbf{s}, \mathbf{a})$ - how good is $\mathbf{s}, \mathbf{a}$?



policy $\pi(\mathbf{a}_t\,|\,\mathbf{s}_t)$ or $\pi(\mathbf{a}_t\,|\,\mathbf{o}_{t-m:t})$ - behavior, usually what we are trying to learn

can be represented using a generative model

Examples of RL problem



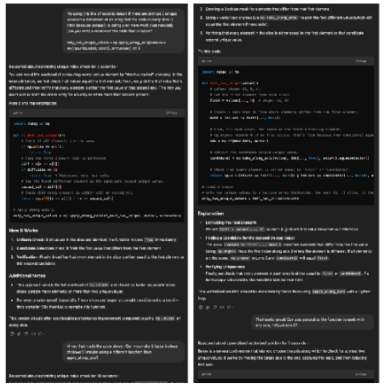
state $\mathbf{s}$ - RGB images, joint positions, joint velocities

action $\mathbf{a}$ - commanded next joint position

trajectory τ - 10-sec sequence of camera, joint readings, controls at 20 Hz

$$(\mathbf{s}_1, \mathbf{a}_1, \mathbf{s}_2, \mathbf{a}_2, \ldots, \mathbf{s}_T, \mathbf{a}_T),\ T = 200$$

reward $r(\mathbf{s}, \mathbf{a})$ = 1 if the towel is on the hook in state $\mathbf{s}$
0 otherwise



observation $\mathbf{o}$ - the user's most recent message

action $\mathbf{a}$ - chatbot's next message

trajectory τ - variable length conversation trace

$$(\mathbf{o}_1, \mathbf{a}_1, \mathbf{o}_2, \mathbf{a}_2, \ldots, \mathbf{o}_T, \mathbf{a}_T)$$

reward $r(\mathbf{s}, \mathbf{a})$ = 1 if the user gives upvote
-10 if the user downvotes
0 if no user feedback

Goal of Reinforcement Learning

maximize *expected* sum of rewards: $\max_{\theta} \mathbb{E}_{\tau \sim p_\theta(\tau)} \left[\sum_t^T r(\mathbf{s}_t, \mathbf{a}_t) \right]$

$$\underbrace{p(\mathbf{s}_1, \mathbf{a}_1, \ldots, \mathbf{s}_T, \mathbf{a}_T)}_{p_\theta(\tau)} = p(\mathbf{s}_1) \prod_{t=1}^{T} \pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t) p(\mathbf{s_{t+1}} \mid \mathbf{s}_t, \mathbf{a}_t)$$

How good is a particular policy?

value function $V^\pi(\mathbf{s})$ - future expected reward starting at $\mathbf{s}$ and following π

Q-function $Q^\pi(\mathbf{s}, \mathbf{a})$ - future expected reward starting at $\mathbf{s}$, taking $\mathbf{a}$, then following π

Type of Algorithms

maximize *expected* sum of rewards: $\max_{\theta} \mathbb{E}_{\tau \sim p_\theta(\tau)} \left[\sum_t^T r(\mathbf{s}_t, \mathbf{a}_t) \right]$

1. **Imitation learning**: mimic a policy that achieves high reward
2. **Policy gradients**: directly differentiate the above objective
3. **Actor-critic**: estimate value of the current policy and use it to make the policy better
4. **Value-based**: estimate value of the optimal policy
5. **Model-based**: learn to model the dynamics, and use it for planning or policy improvement

Two Questions

Q1 · What's the signal?

Imitation Learning: copy demonstrations. Supervised learning

Reinforcement Learning: trial & error guided by a reward signal.

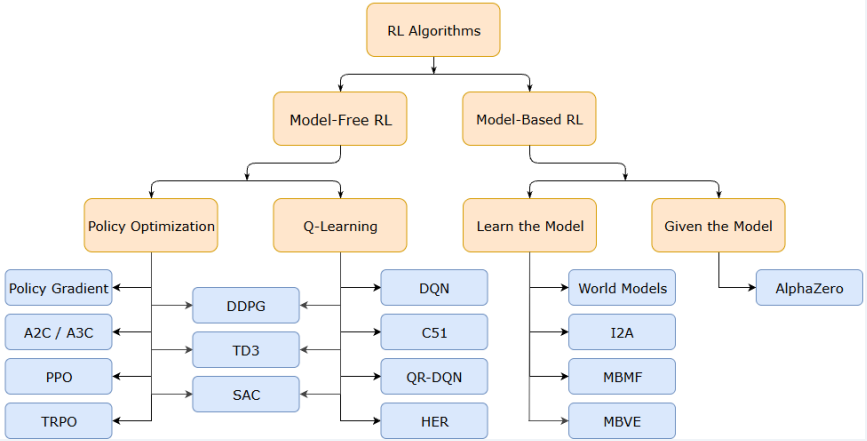
Our main axis today: it maps onto SFT → RLHF in LLMs

Q2 · World model or not?

Model-based: learn s' = f(s, a); can plan by simulating ahead before acting.

Model-free: learn the behavior directly; at test time, just execute.

c.f. Planning is not an axis.
It is how a model-based agent produces actions.



Source: OpenAI Spinning Up

Part 3

Imitation Learning & Diffusion Policy

Behavioral Cloning, Compounding error, action as generation

Imitation Learning

Data: Given trajectories collected by an expert

"*demonstrations*" $\mathscr{D} := \{(\mathbf{s}_1, \mathbf{a}_1, \ldots, \mathbf{s}_T)\}$

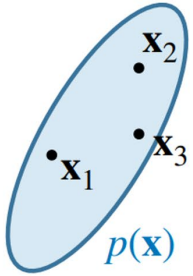
(sampled from some unknown policy π_{expert})

Training: For deterministic policy, supervised regression to the expert's actions

$$\min_{\theta} \frac{1}{|\mathscr{D}|} \sum_{(\mathbf{s},\mathbf{a}) \in \mathscr{D}} \|\mathbf{a} - \hat{\mathbf{a}}\|^2 \qquad \text{where } \hat{\mathbf{a}} = \pi_\theta(\mathbf{s})$$

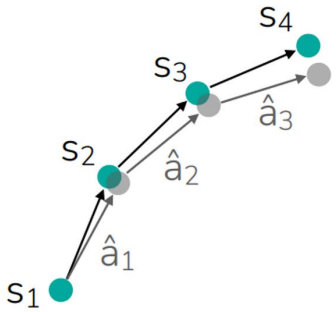
Compounding Errors

Supervised learning



Inputs independent of predicted labels $\hat{\mathbf{y}}$

Supervised learning of behavior



Predicted actions affect next state.

Errors can lead to drift away from the data distribution!

Errors can then compound!

$$p_{expert}(\mathbf{s}) \neq p_{\pi}(\mathbf{s})$$

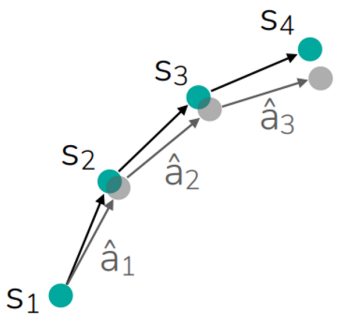
states visited by expert

states visited by learned policy π

"covariate shift"

Addressing Compounding Errors

Supervised learning of behavior



Predicted actions affect next state.

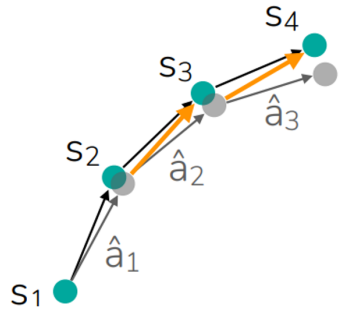
Errors can lead to drift away from the data distribution!

Errors can then compound!

$$p_{expert}(\mathbf{s}) \neq p_{\pi}(\mathbf{s})$$

Solutions?

1. Collect A LOT of demo data & hope for the best.
2. Collect corrective behavior data



1. Roll out learned policy π_θ: $\mathbf{s}'_1, \hat{\mathbf{a}}_1, \ldots, \mathbf{s}'_T$
2. Query expert action at visited states $\mathbf{a}^* \sim \pi_{expert}(\cdot \mid \mathbf{s}')$
3. Aggregate corrections with existing data $\mathscr{D} \leftarrow \mathscr{D} \cup \{(\mathbf{s}', \mathbf{a}^*)\}$
4. Update policy $\min_\theta \mathscr{L}(\pi_\theta, \mathscr{D})$

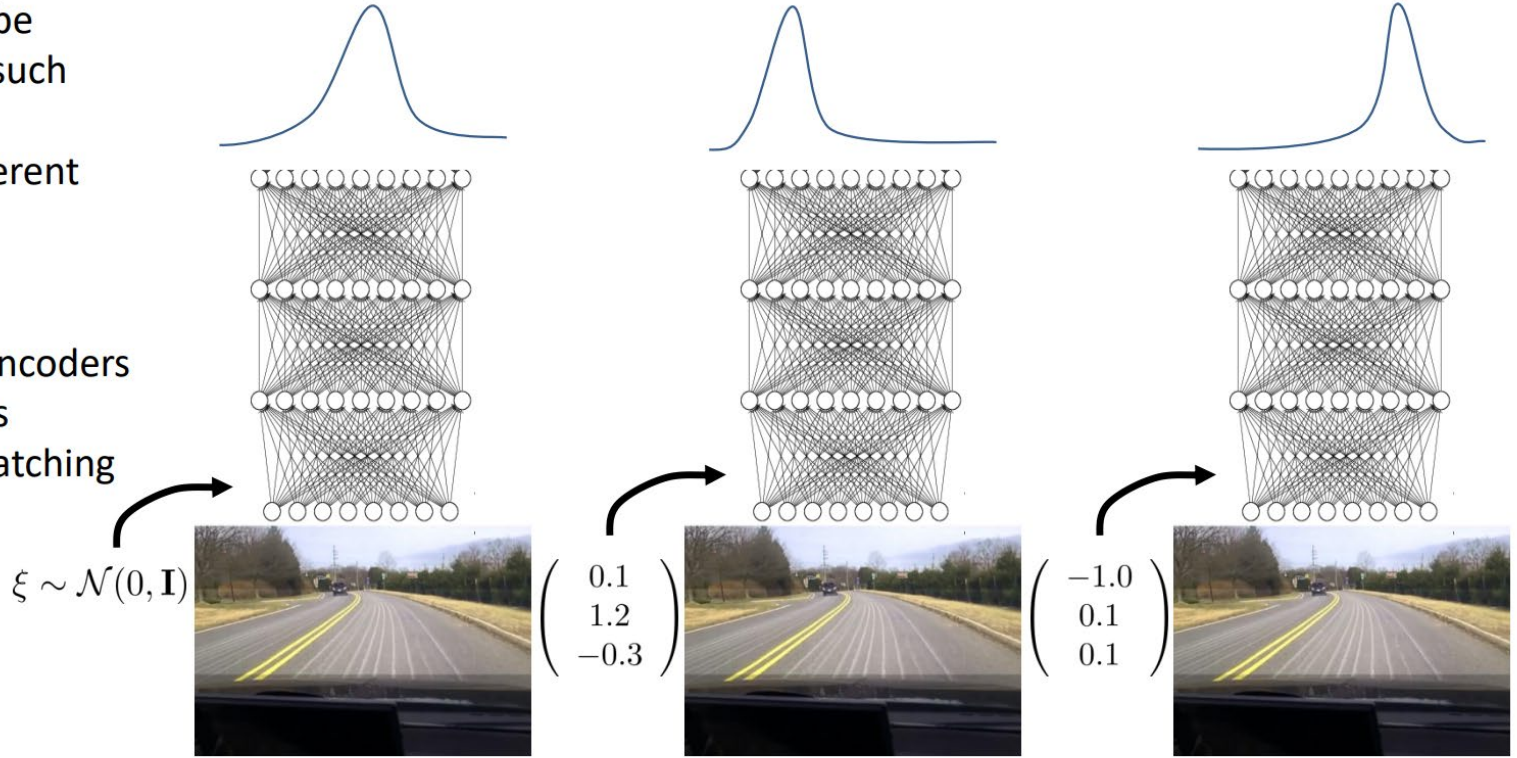
"dataset aggregation" (DAgger)

Expressive Continuous Distributions

Key challenge: the random noise must actually be used by the model, such that different noise samples map to different action modes

Some solutions:

- Variational autoencoders
- Normalizing flows
- Diffusion/flow matching



Policy training implementation using Diffusion

1. construct minibatch

   for each element in the batch j:

   sample $(\mathbf{o}_t^{(j)}, \mathbf{a}_t^{(j)})$ from dataset

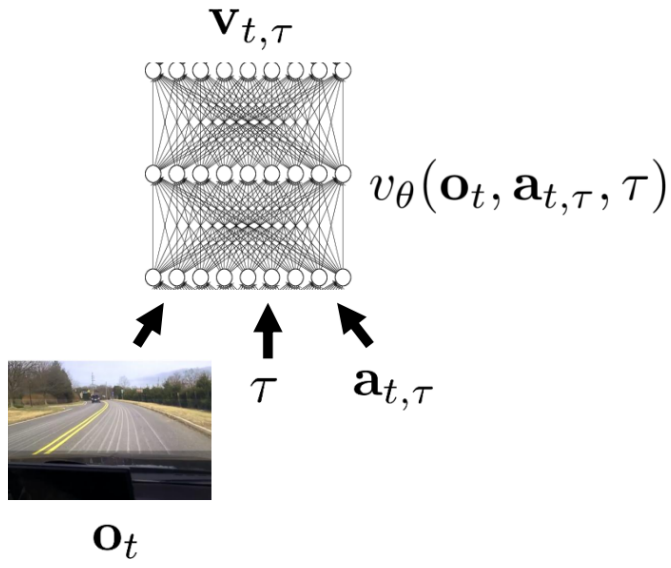
   sample $\mathbf{a}_{t,0}^{(j)} \sim \mathcal{N}(0, \mathbf{I})$

   sample $\tau^{(j)} \sim p(\tau)$ (e.g., $p(\tau) = \mathcal{U}(0, 1)$)

   compute $\mathbf{a}_{t,\tau}^{(j)} = \tau^{(j)}\mathbf{a}_t^{(j)} + (1 - \tau^{(j)})\mathbf{a}_{t,0}^{(j)}$

2. update $\theta \leftarrow \theta + \alpha\nabla_\theta\mathcal{L}$

   where $\mathcal{L} = \sum_{j=1}^{B} \|v_\theta(\mathbf{o}_t^{(j)}, \mathbf{a}_{t,\tau}^{(j)}, \tau^{(j)}) - (\mathbf{a}_t^{(j)} - \mathbf{a}_{t,0}^{(j)})\|^2$



Action Chunking

A small detail that helps quite a lot

standard policy:

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

execute $\mathbf{a}_t$ in the environment

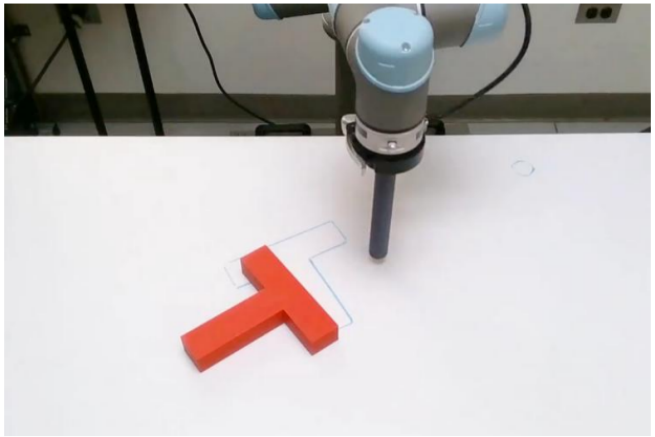
observe $\mathbf{o}_{t+1}$, repeat

action chunked policy:

sample $\mathbf{a}_{t:t+K} \sim \pi_\theta(\mathbf{a}_{t:t+K}|\mathbf{o}_t)$

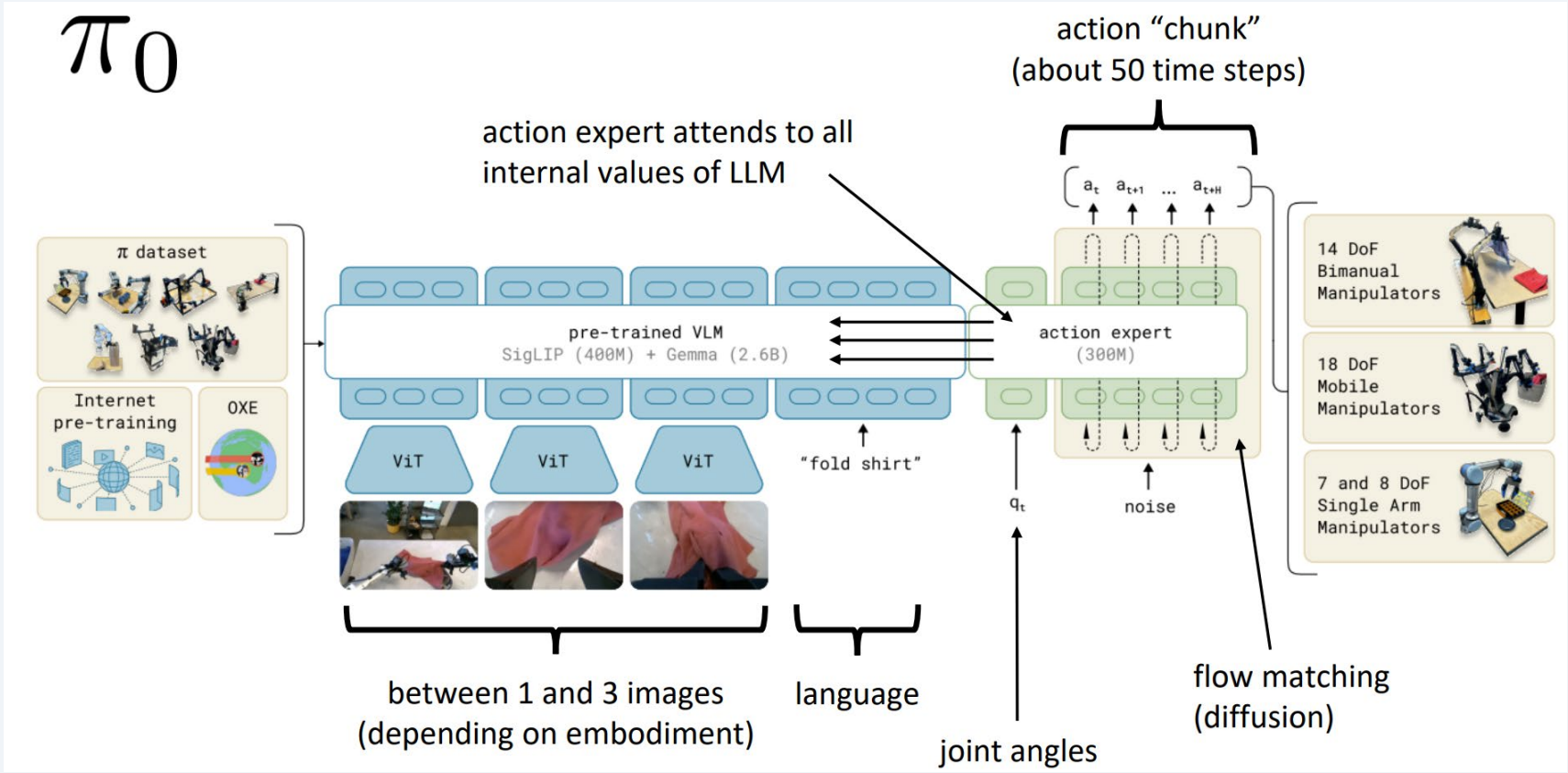
execute $\mathbf{a}_t, \mathbf{a}_{t+1}, ..., \mathbf{a}_{t+K}$ in the environment

observe $\mathbf{o}_{t+K+1}$, repeat



Chi et al. Diffusion policy, 2024

π_0: Diffusion + Chunking + Pre-training

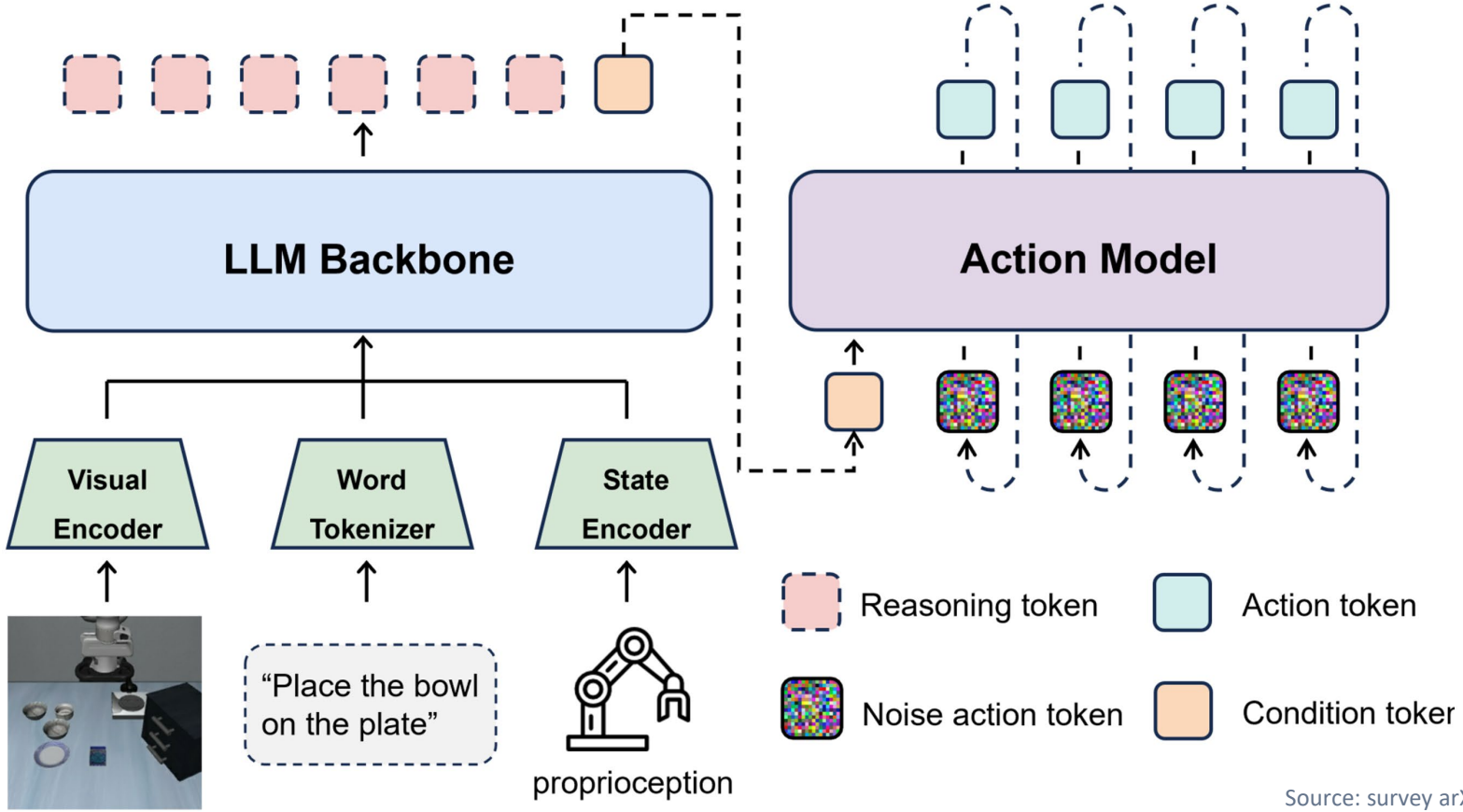


Part 4

VLA & Learning from Experience

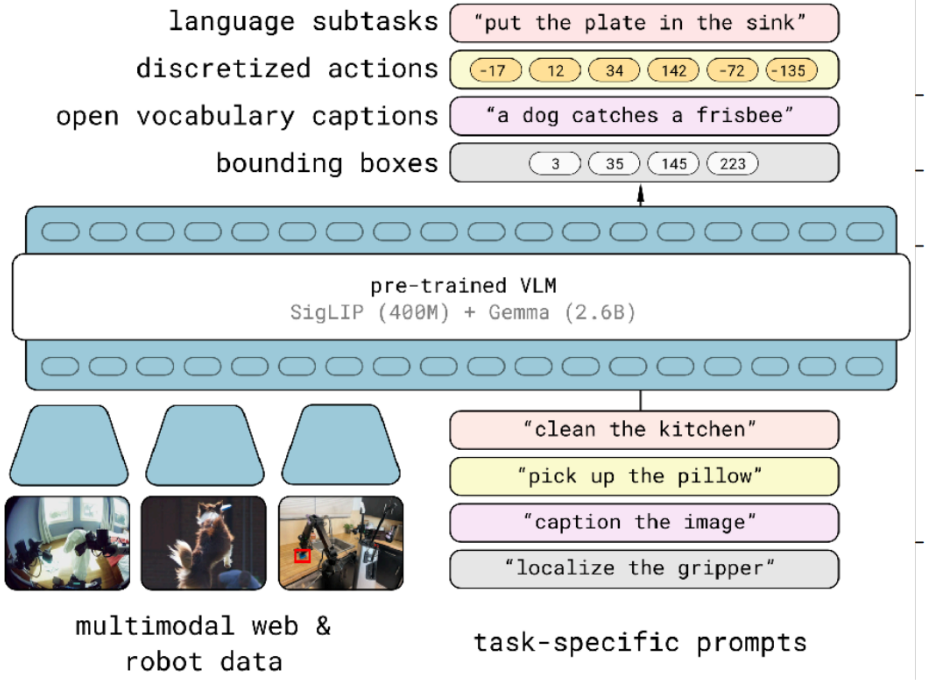
From RT-1 to π, RL-finetuning

Robot Foundation Model



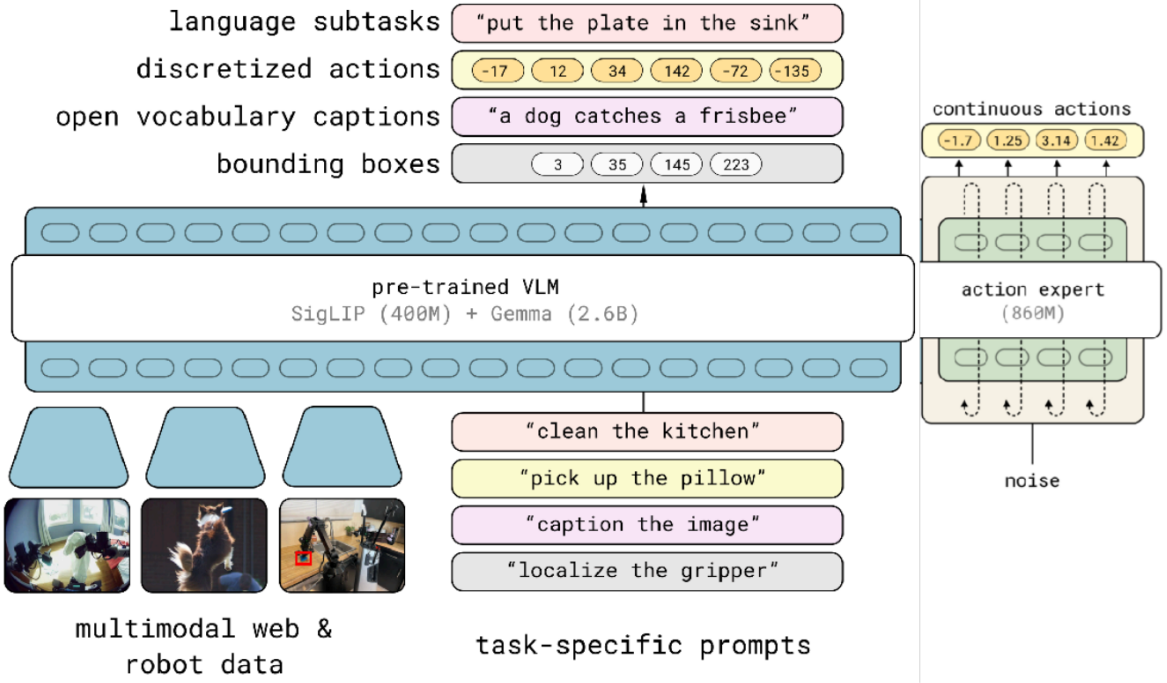
Source: survey arXiv 2510.17111

Robot Foundation Model = VLA



- start with pretrained vision-language model (VLM)
- alternative design: start with generative video model
- training data mixture often includes:
  - robot demonstration data (imitation learning)
  - VLM tasks (question answering, captioning, detection)
  - human video data (motion prediction)
- often includes an diffusion-based “action expert”

Robot Foundation Model = VLA

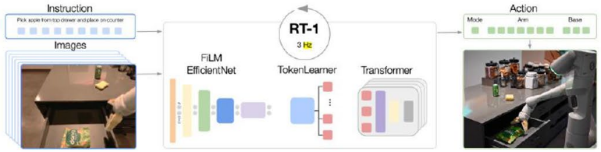


Action expert:

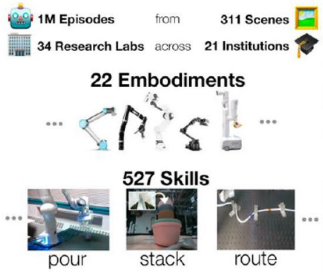
- uses diffusion or flow matching to predict continuous actions
- attends to all activations of LLM backbone
- designed to avoid multiple forward passes through entire backbone
- gradient is often not passed to backbone

Vision-Language Action Model Evolution

Action becomes discrete tokens

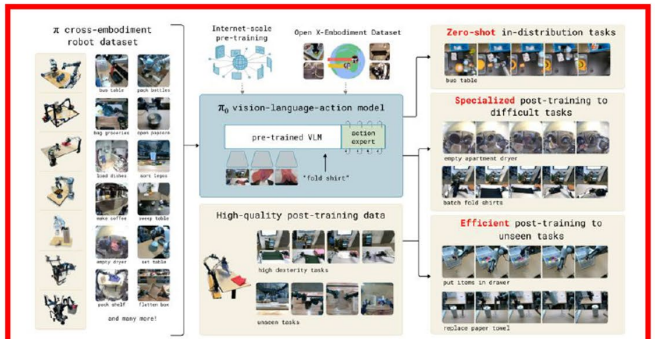


RT-1 (Dec. 2022)



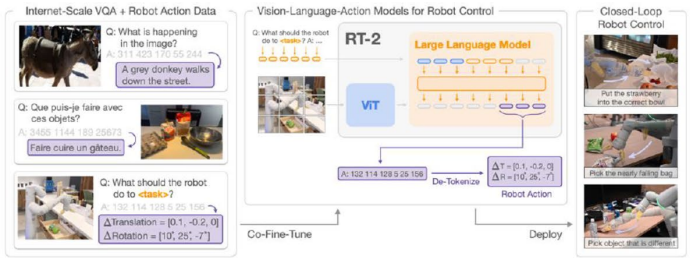
RT-X (Oct. 2023)

Flow-matching action experts



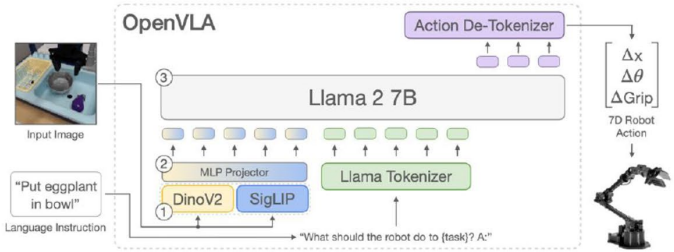
Pi-Zero (Oct. 2024)

RT-2 (Jul. 2023)



Web-scale VLM co-finetuned with robot data

OpenVLA (Jun. 2024)



Open-source 7B on 970k episodes

Helix (Figure)
Hi-Robot (PI)
Gemini Robotics
Pi-0.5 (PI)
GR00T (Nvidia)
DYNA-1
…

The 80% Ceiling — the LLM Parallel

How LLMs learn

Pretrain → SFT: imitate curated demonstrations (supervised).

→ **RLHF:** optimize a reward learned from human preferences.

→ **RLVR:** verifiable rewards
tests pass → r = 1 (o1, DeepSeek-R1).

Imitation plateaus; RL pushes past the plateau.

How robots learn

VLM pretrain → IL on demos: plateaus near ~80% success perfect demos never teach recovery.

→ **RL fine-tuning:** learn from the robot's own experience; autonomy needs 99%, not 80%.

→ **Verifiable / autonomous reward:** success detection labels experience at scale.

Same ladder, two embodiments.

Source: π*0.6 (arXiv 2511.14759)

RECAP: Three Kinds of Experience

1. Collect roll-outs & interventions

The robot runs; an expert steps in when it drifts

Autonomous roll-outs + corrections + the original demos

2. Update the value function

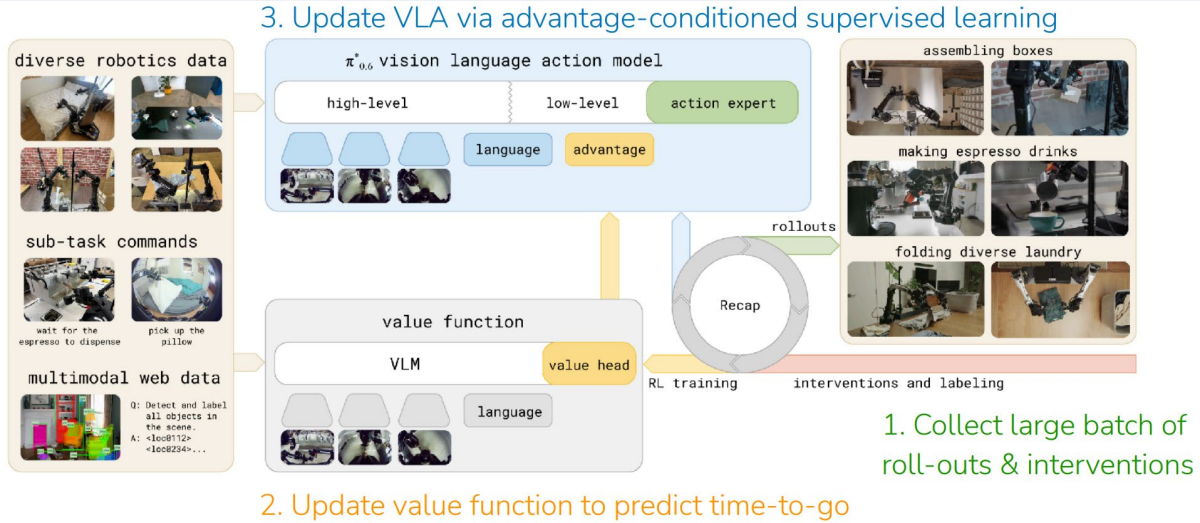
A VLM learns to predict time-to-go

Value = How long until success from here
Advantage = how much an action *shortens* it

3. Update the VLA Advantage-conditioned SL

Label actions good / bad

Just supervised learning on its own experience



Source: π*0.6 (arXiv 2511.14759)

Evaluation of the VLA

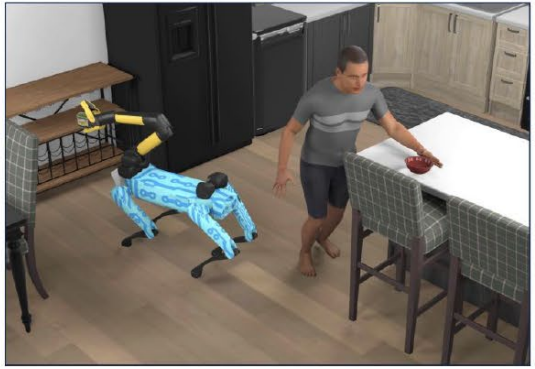
- Evaluation is primarily conducted in the real world
  - Real-world evaluation is costly and noisy
    - “We have large enough budget such that we can still make progress.”
  - Weak correlation between training loss and real-world success rate.
    - Training objectives vs task-specific metrics, training vs testing horizons

ALOHA 2

Evaluation of the VLA

- What about evaluation in simulation?
  - Sim-to-real gap: rigid / deformable / cloth
  - Efficient asset generation
  - Digitalization of the real world
  - Procedural generation of realistic and diverse scenes
  - Correlation between sim and real

ImageNet in Embodied AI?



Habitat 3.0

World Models: Mirror vs Map

Understand — the Mirror

A world model as representation: capture how the world works (Ha & Schmidhuber 2018; JEPA).

V-JEPA 2: 1M h unlabeled video (Wk 7 SSL + Wk 14) → <62 h robot data → zero-shot Franka planning.

2026: LeCun leaves Meta → AMI Labs raises $1B for JEPA-style WMs.

Predict — the Map

A world model as future simulator: imagine outcomes, then decide (LeCun; video generation à la Sora).

Open debate: does video generation learn physics, or memorize pixels?

The critique paper → seminar ammunition in the RL track.

2026: World Labs raises $1B; Genie 3 becomes Waymo's simulator.

Source: survey arXiv 2411.14499, V-JEPA 2 (arXiv 2506.09985), critique (arXiv 2507.05169)

Summary & Resources

What We Covered

Action closes the loop: i.i.d. breaks, and there is no label for the right action

Two questions map the field: what's the signal (imitate vs. reward)? Is there a world model?

Imitation Learning → compounding errors → DAgger, diffusion policies & action chunking

VLA = VLM + action expert, trained by imitation at scale (RT-1 → π0) and imitation plateaus, so RL fine-tuning follows

World models: understand (Mirror) vs. predict (Map): an open, billion-dollar debate

Further Reading

OpenAI Spinning Up “Kinds of RL Algorithms” - spinningup.openai.com

Stanford CS224R (Finn) - cs224r.stanford.edu
Berkeley CS285 (Levine) - rail.eecs.berkeley.edu/deeprlcourse

Diffusion Policy (arXiv 2303.04137)
π0 (arXiv 2410.24164)
π*0.6 / RECAP (arXiv 2511.14759)
π0.7 (arXiv 2604.15483)
V-JEPA 2 (arXiv 2506.09985)
World-model survey (arXiv 2411.14499)

Next Week Week 16: End of the Semester ☺

Acknowledgments

Some slides and figures in this course are adapted from or inspired by the following open resources.

Stanford CS224R: Deep Reinforcement Learning (Spring 2026)

Chelsea Finn et al. | cs224r.stanford.edu. Parts 2–3 slides adapted from Lectures 1–2

UC Berkeley CS285: Deep Reinforcement Learning

Sergey Levine | rail.eecs.berkeley.edu/deeprlcourse

OpenAI Spinning Up (RL taxonomy figure)

spinningup.openai.com

Physical Intelligence (π0 architecture & blog figures)

pi.website

Key papers: Diffusion Policy (Chi et al., 2303.04137), π0 (2410.24164), π*0.6 (2511.14759), V-JEPA 2 (2506.09985), World-model survey (2411.14499), Efficient VLA survey (251.17111)