From Demonstration to Generalist Manipulation 🤖

MoAI Seminar (9/21)

Jeongyoon Ko, CSE Department ‘25

Introduction:
Goal of the seminar

- Understand challenges in robotic manipulation
- What each of these three papers is actually trying to fix
  - **ALOHA**
  - **Open X-Embodiment**
  - **Octo**

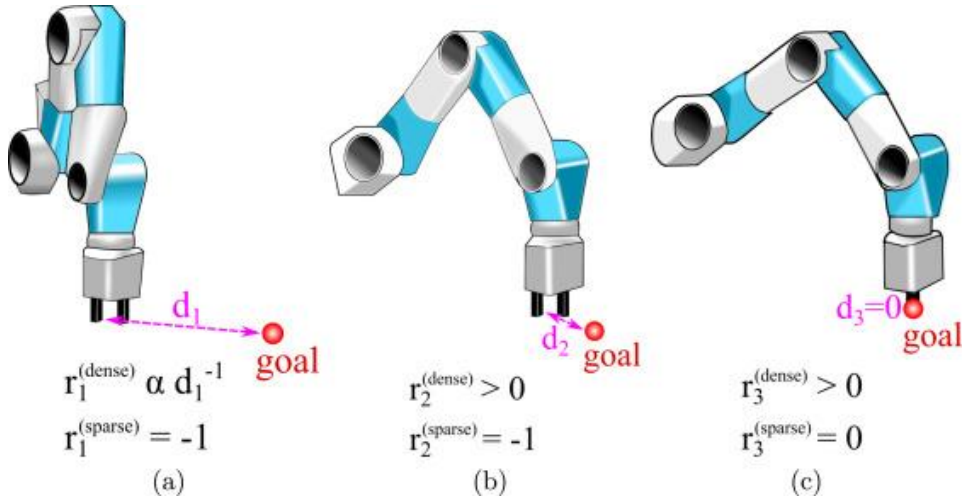
Introduction:
Roadmap

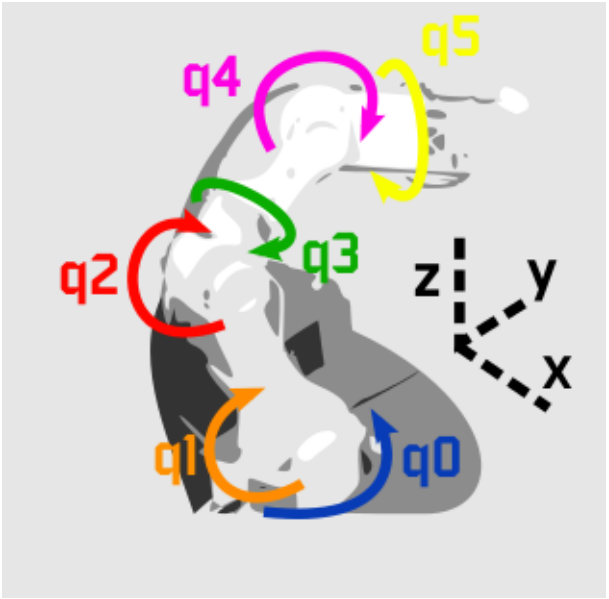
- **ALOHA**: how to get good demonstrations?
- **Open X-Embodiment**: how to scale them across robots?
- **Octo**: how to turn scaled data into one policy?

Introduction:
Why Manipulation Is Hard?

- 1) Contact-rich dynamics
  - friction, deformation, you can't simulate it perfectly

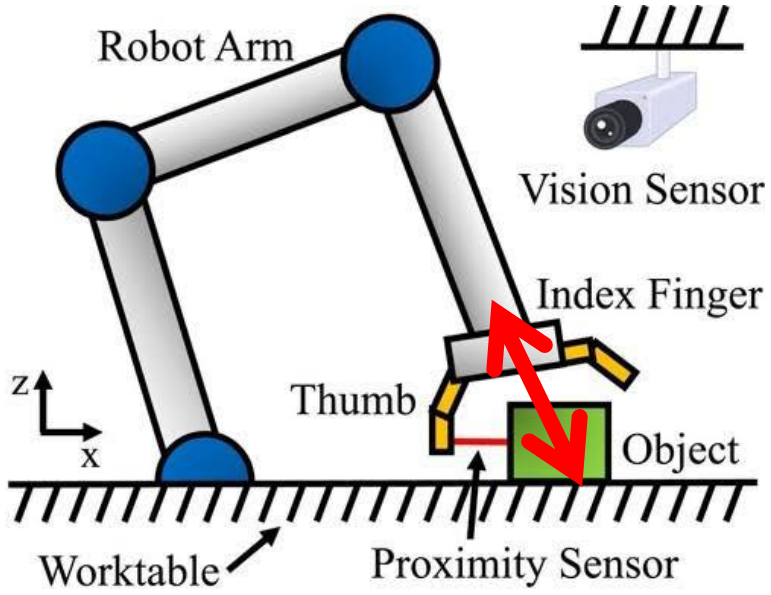
- 2) Actions are continuous and high-dimensional
  - not a handful of discrete choices





Introduction:
Why Manipulation Is Hard?

- 3) Long horizons
  - Early errors compound over time
- 4) Reward specification is difficult



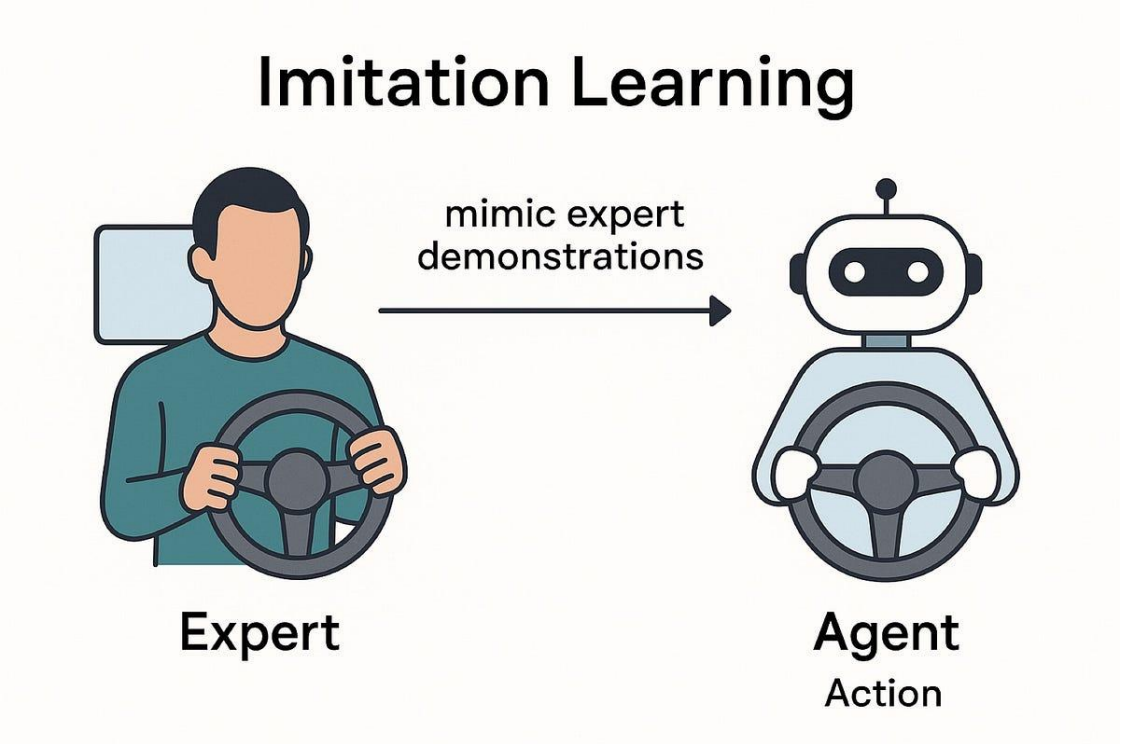
So: learn it from demonstrations instead of designing it by hand.

Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware 🏝️

Zhao et al., RSS 2023

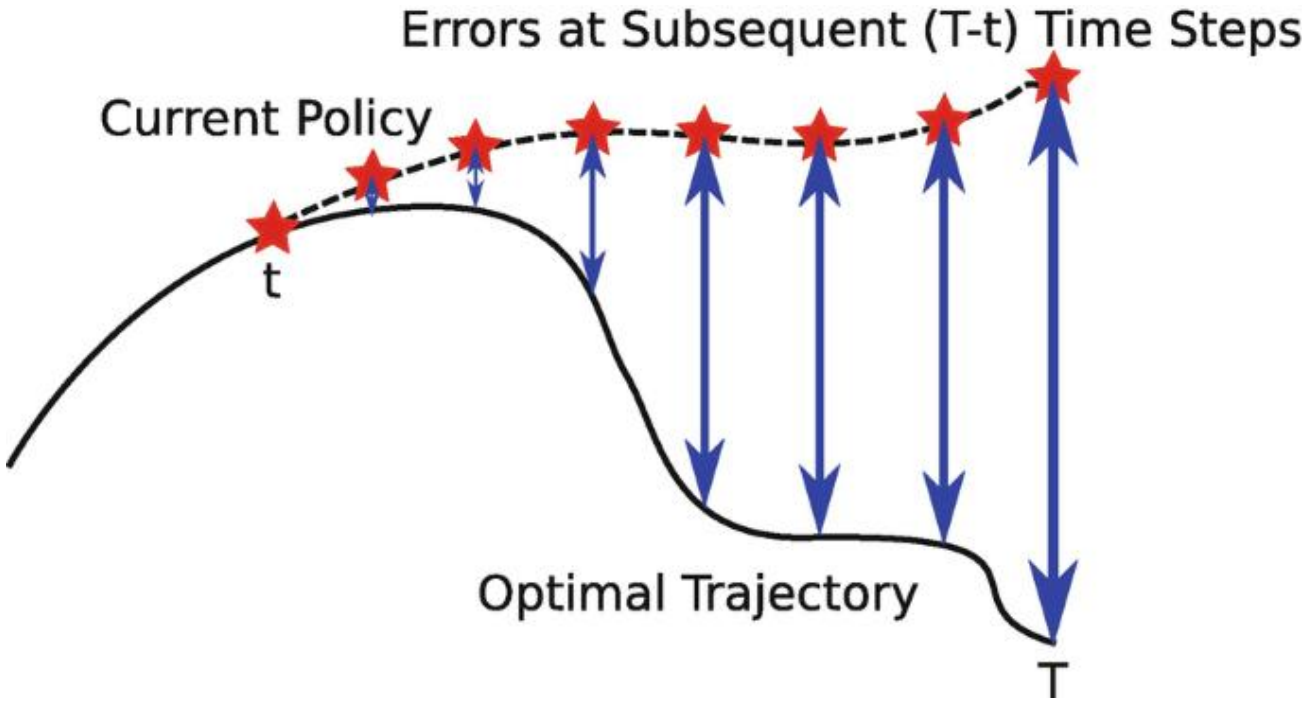
Preliminaries: Imitation Learning

- Learn $\pi_\theta(a_t \mid o_t)$ directly from expert trajectories
- Behavior cloning
  - supervised regression from observation to action



Preliminaries: Compounding Error

- Small prediction errors shift the state away from the training distribution
- The policy has not been trained on these off-distribution states, and **error accumulates**



Problem:
Two Things Stand in the Way

- 1) Fine-grained bimanual tasks require near-millimeter precision
- 2) Precise teleoperation hardware is **costly,** and plain behavior cloning is not robust over **long horizons**



Fine-grained manipulation needs both fixed.

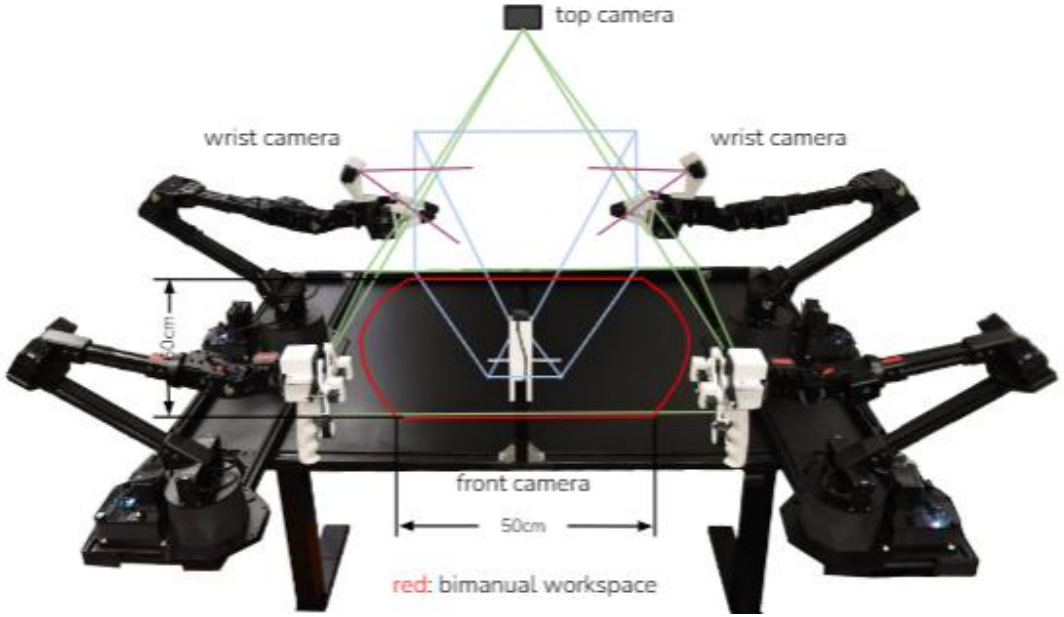
Contribution

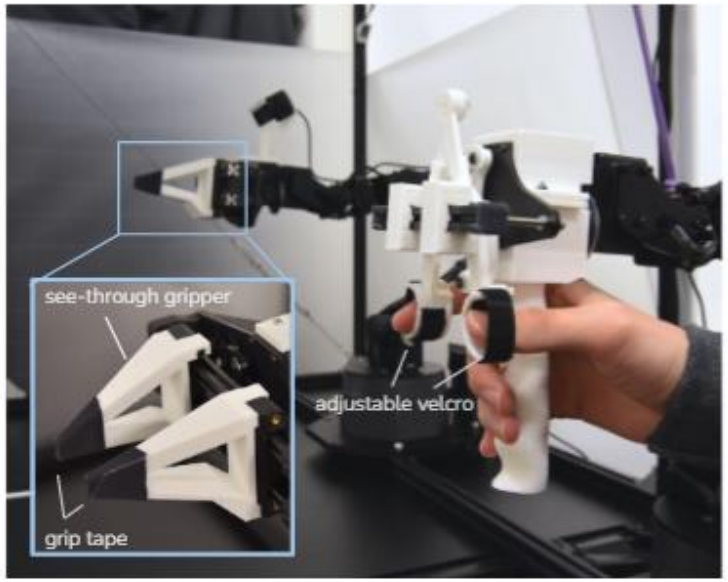
Contribution:

- A low-cost, reproducible bimanual teleoperation system
- ACT (Action Chunking with Transformers), an IL algorithm designed to counter compounding error

Methods:
Hardware

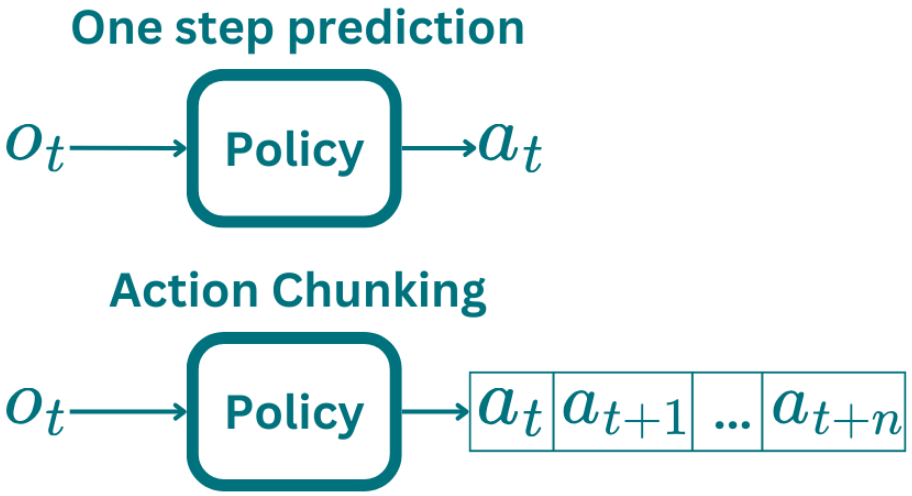
- Leader–follower teleoperation
  - a human operator drives low-cost "leader" arms
  - The "follower" arms replicate the motion, producing precise demonstration data





Methods:
Action Chunking

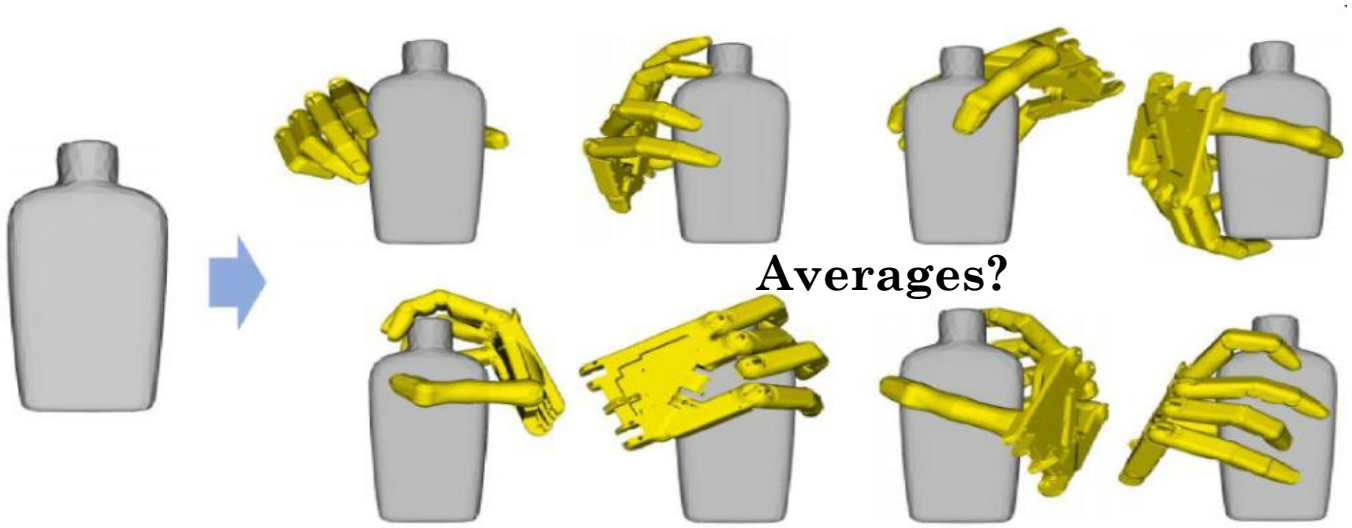
- The policy predicts a chunk of k future actions at each step, rather than a single action
  - This shortens the effective decision horizon and limits the accumulation of error





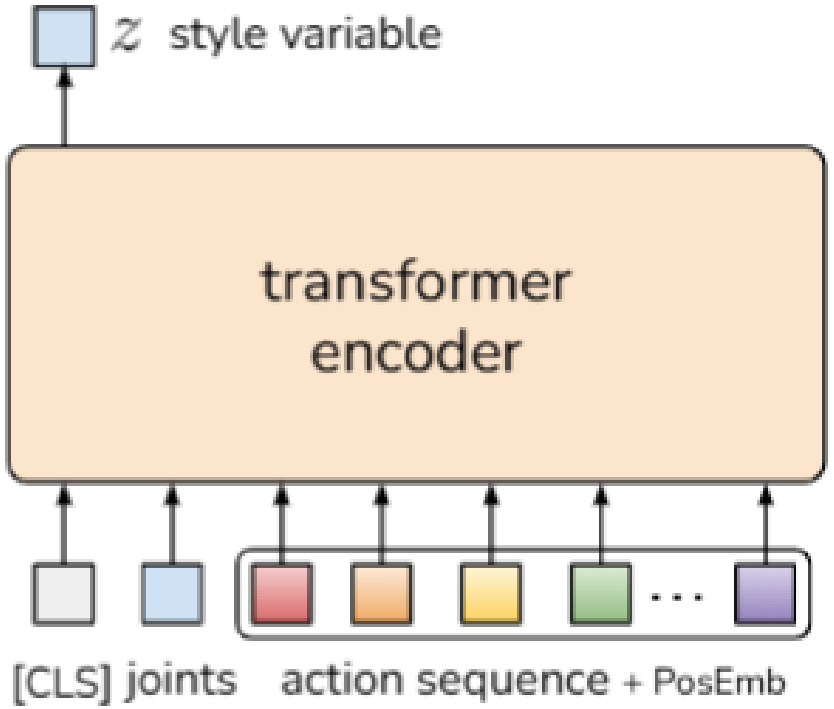
Methods:
Conditional VAE

- Human demonstrations are multimodal
  - the same task is performed with natural variation
  - A latent variable z allows the model to represent this variation, rather than averaging over it



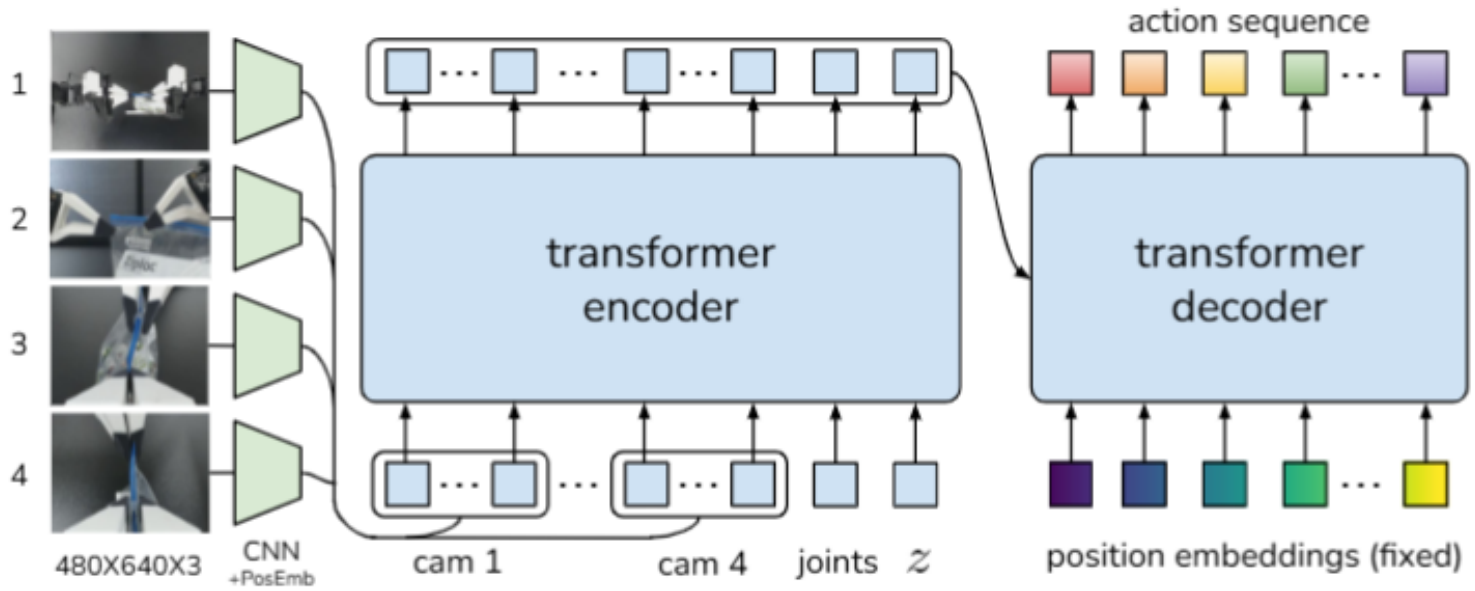
Methods:
Conditional VAE

- Human demonstrations are multimodal
  - the same task is performed with natural variation
  - A latent variable z allows the model to represent this variation, rather than averaging over it



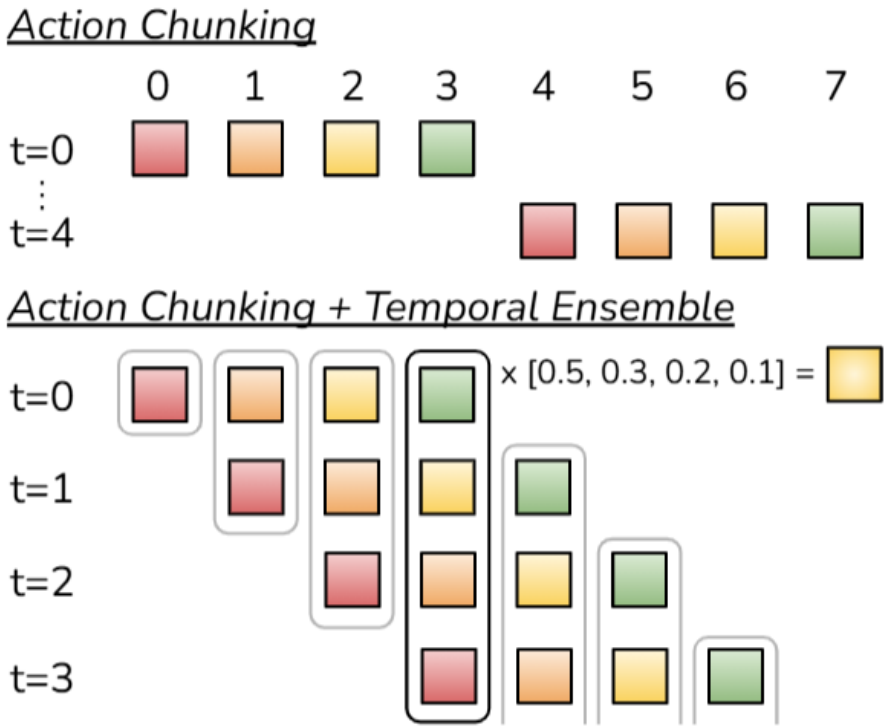
Methods:
Conditional VAE

- Encoder (training only)
  - compresses the demonstrated action sequence into a distribution over z
- Decoder
  - takes z together with the current observation, and outputs the action chunk



Methods:
Temporal Ensembling

- Action chunks predicted at consecutive timesteps overlap
  - Averaging the overlapping predictions yields smoother, more stable execution



Experiments: Setup

- Evaluated on 8 fine-grained bimanual tasks: 6 real-world with ALOHA, 2 simulated in MuJoCo
  - Real-world: **Slide Ziploc, Slot Battery, Open Cup, Thread Velcro, Prep Tape, Put On Shoe**



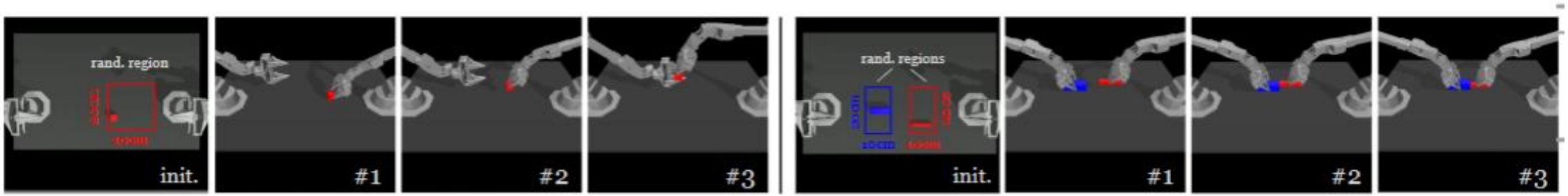
Slide Ziploc: Open the ziploc bag that is standing upright on the table. The bag is randomized along the 15cm white line



Open Cup: Pick up and open the lid of a translucent condiment cup. The cup is randomized along the 15cm white line. 1

Experiments: Setup

- Evaluated on 8 fine-grained bimanual tasks: 6 real-world with ALOHA, 2 simulated in MuJoCo
  - Simulated: **Cube Transfer, Bimanual Insertion**
  - Each real task learned from just 50 demonstrations, about 10–20 minutes of data



Left: Cube Transfer. Transfer the red cube to the other arm. The right arm touches *(#1)* and grasps *(#2)* the red cube, then hands it to the left arm.

Experiments: Result

- **ACT substantially outperforms** the baselines on precision-critical tasks
- Strong performance is achieved from a relatively **small number of demonstrations**

	Cube Transfer (sim)			**Bimanual Insertion** (sim)			**Slide Ziploc** (real)			**Slot Battery** (real)		
	Touched	Lifted	Transfer	Grasp	Contact	Insert	Grasp	Pinch	Open	Grasp	Place	Insert
BC-ConvMLP	34 | 3	17 | 1	1 | 0	5 | 0	1 | 0	1 | 0	0	0	0	0	0	0
BeT	60 | 16	51 | 13	27 | 1	21 | 0	4 | 0	3 | 0	8	0	0	4	0	0
RT-1	44 | 4	33 | 2	2 | 0	2 | 0	0 | 0	1 | 0	4	0	0	4	0	0
VINN	13 | 17	9 | 11	3 | 0	6 | 0	1 | 0	1 | 0	28	0	0	20	0	0
ACT (Ours)	**97 | 82**	**90 | 60**	**86 | 50**	**93 | 76**	**90 | 66**	**32 | 20**	**92**	**96**	**88**	**100**	**100**	**96**

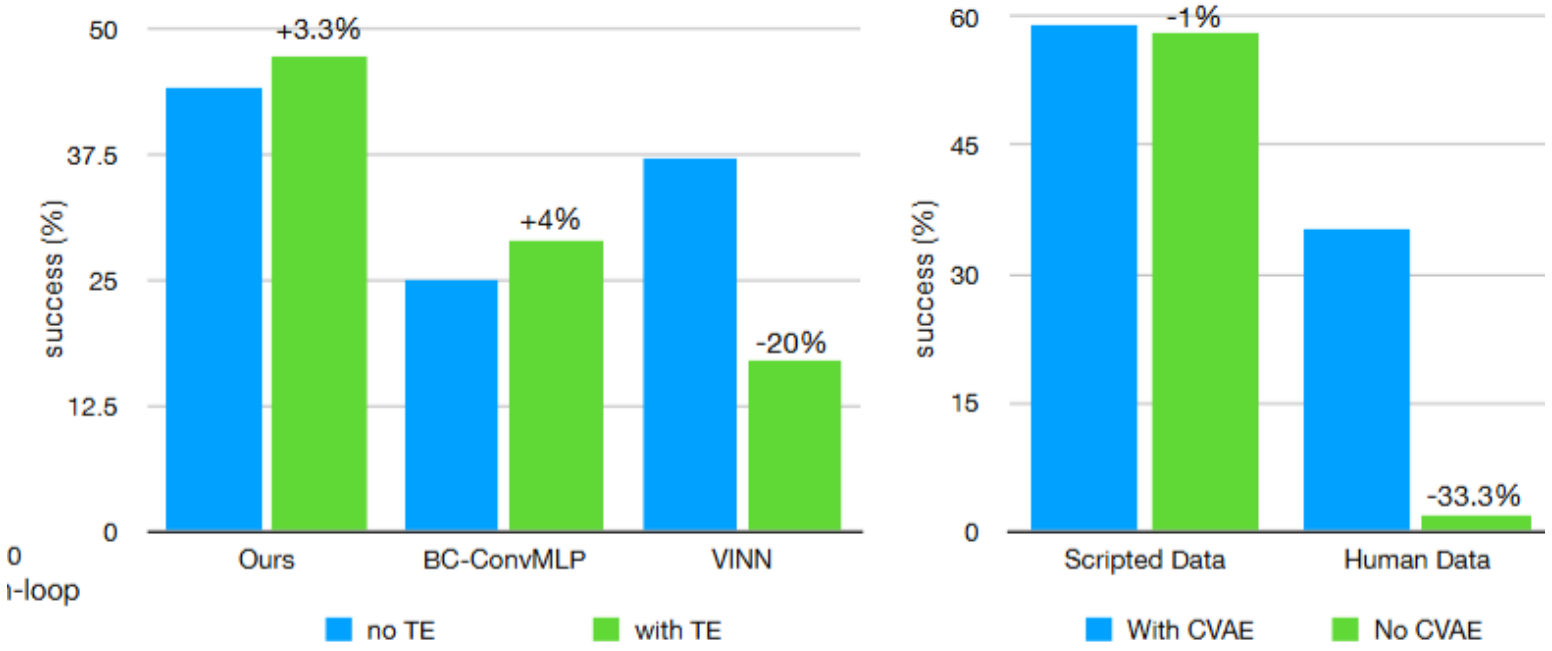
TABLE I: Success rate (%) for 2 simulated and 2 real-world tasks, comparing our method with 4 baselines. For the two simulated tasks, we report [training with scripted data | training with human data], with 3 seeds and 50 policy evaluations each. For the real-world tasks, we report training with human data, with 1 seed and 25 evaluations. Overall, ACT significantly outperforms previous methods.

	Open Cup (real)			**Thread Velcro** (real)			**Prep Tape** (real)				**Put On Shoe** (real)			
	Tip Over	Grasp	Open Lid	Lift	Grasp	Insert	Grasp	Cut	Handover	Hang	Lift	Insert	Support	Secure
BeT	12	0	0	24	0	0	8	0	0	0	12	0	0	0
ACT (Ours)	**100**	**96**	**84**	**92**	**40**	**20**	**96**	**92**	**72**	**64**	**100**	**92**	**92**	**92**

TABLE II: Success rate (%) for the remaining 3 real-world tasks. We only compare with the best performing baseline BeT.

Experiments: Result

- Removing action chunking degrades performance
- Removing temporal ensembling degrades performance as well; **both components contribute**



Conclusion

- Low-cost hardware combined with chunked imitation learning enables precise bimanual manipulation
- However, all data is collected from a single embodiment in a single lab.
- This raises the question of generalization: **what happens beyond this one robot?**

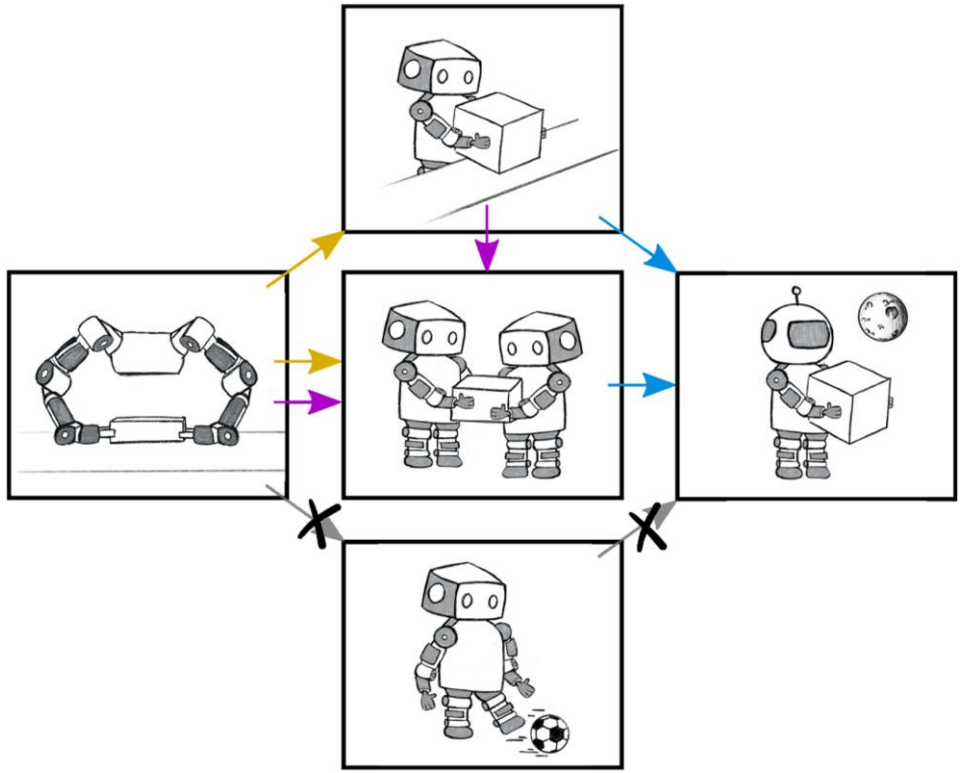
Robotic Learning Datasets and RT-X Models

Open X-Embodiment Collaboration, ICRA 2024 IEEE

ICRA Best Conference Paper Award

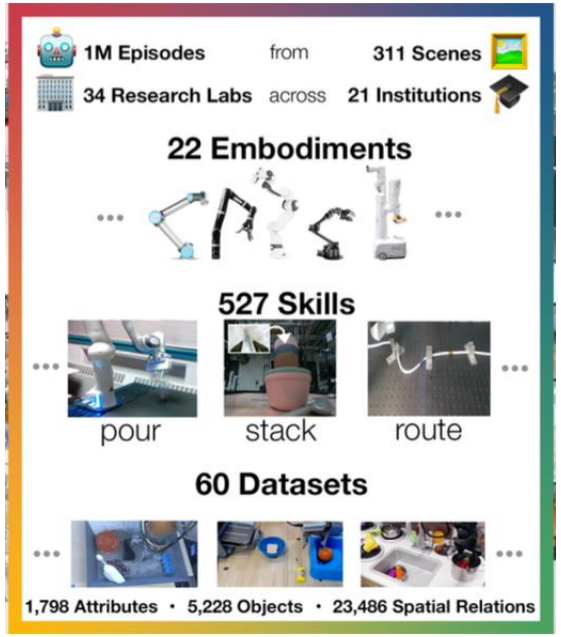
Open X-Embodiment:
The Scaling Problem: One Policy per Robot?

- Standard imitation learning trains one policy per robot
  - A new robot means new sensors, new action space
  - Can experience collected on one robot improve learning on another?



Open X-Embodiment:
The Scaling Problem: One Policy per Robot?

- Pools data from 21 institutions, 22 embodiments, into one standardized dataset
  - RT-1-X / RT-2-X, trained on the pooled data, show consistent positive transfer across robots
- **positive transfer is not yet a single, deployable generalist policy**

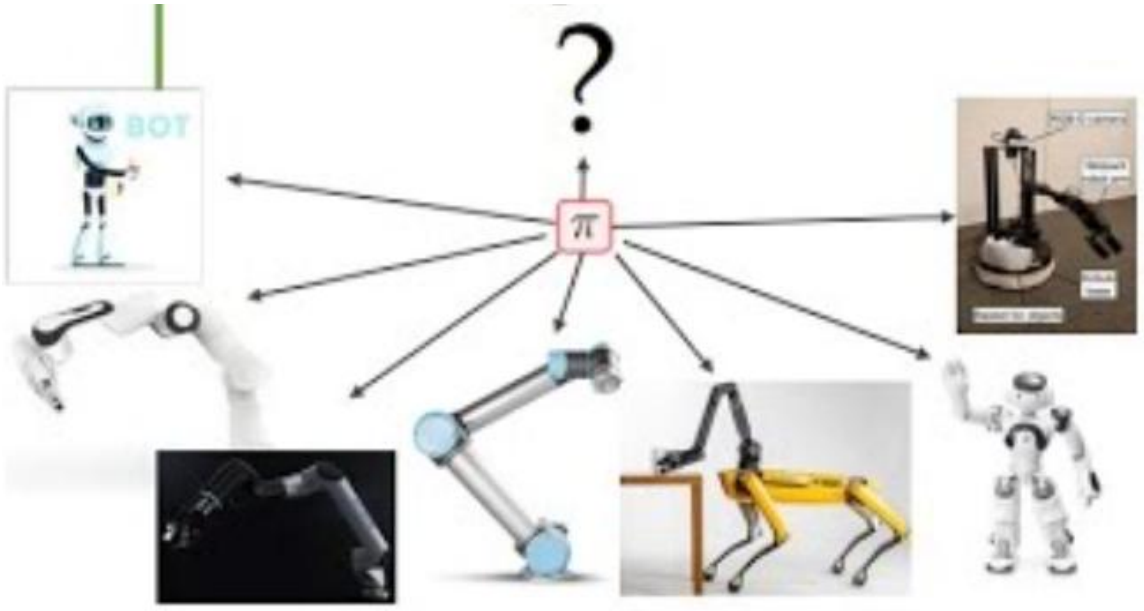


An Open-Source Generalist Robot Policy 🐙

Octo Model Team et al., RSS 2024

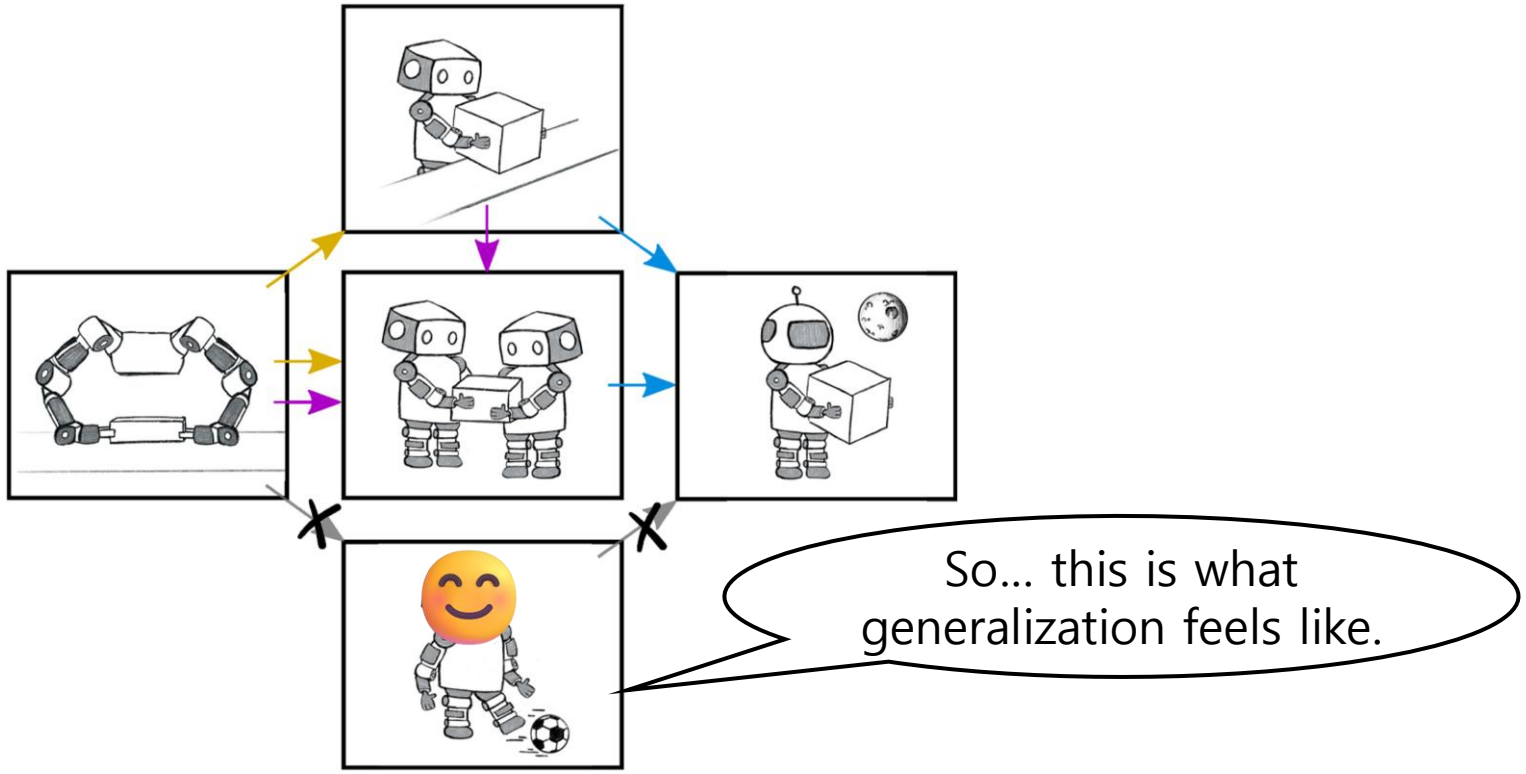
Preliminaries: Generalist Policy

- **A single pretrained policy** intended to transfer across tasks and robots
  - Not a policy that works zero-shot on every robot out of the box
  - Rather: pretrain once, then adapt cheaply (pretraining, then target-robot fine-tuning, then execution)



Preliminaries: Generalist Policy

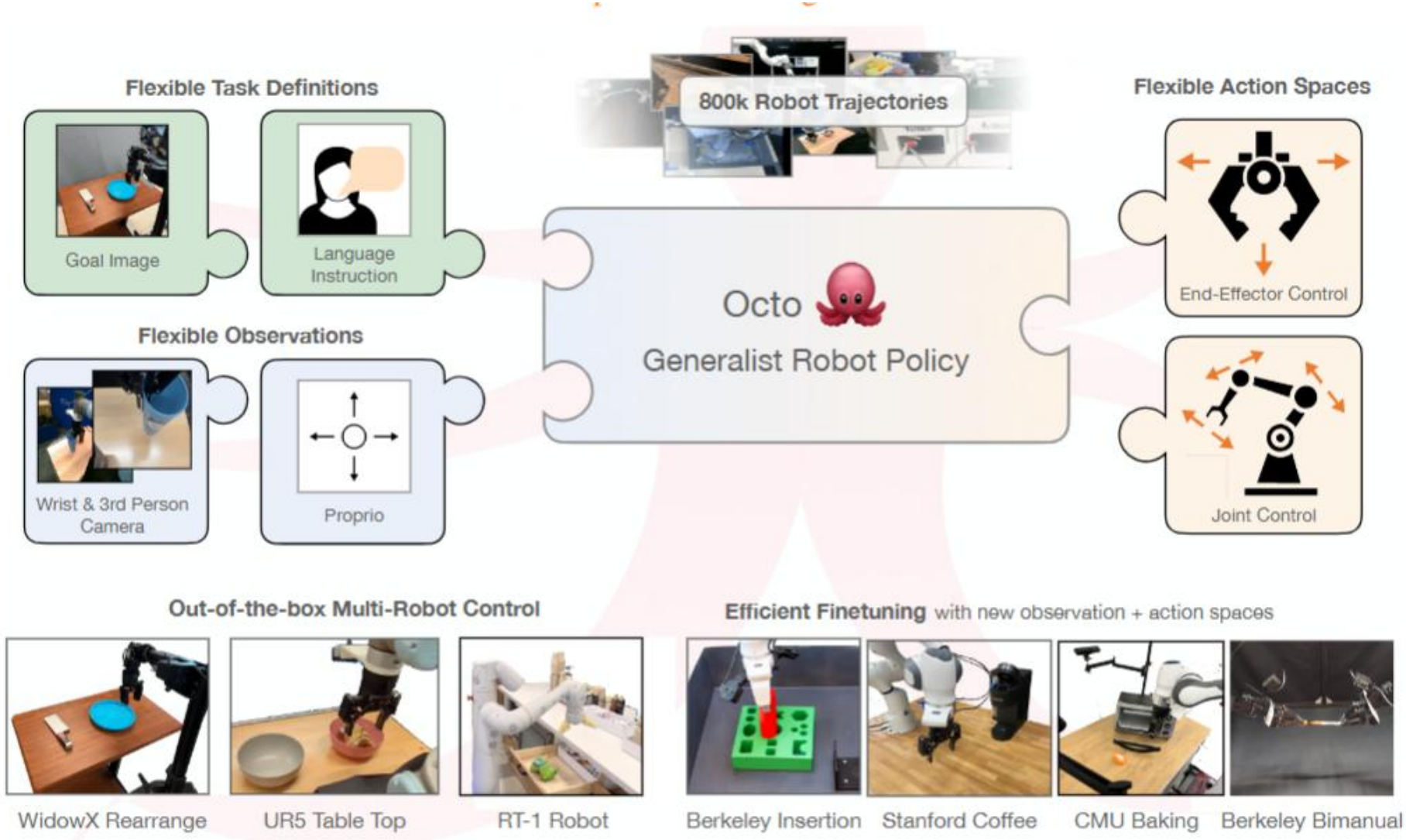
- **The goal**:
  - **transfer efficiently** to robots, sensors, and action spaces **the model never saw** during pretraining



Contribution

- **A transformer-based generalist policy**, pretrained on roughly 800k trajectories from Open X-Embodiment
- Fine-tuned to 9 robot platforms, several **with observation and action spaces absent** from pretraining

Contribution

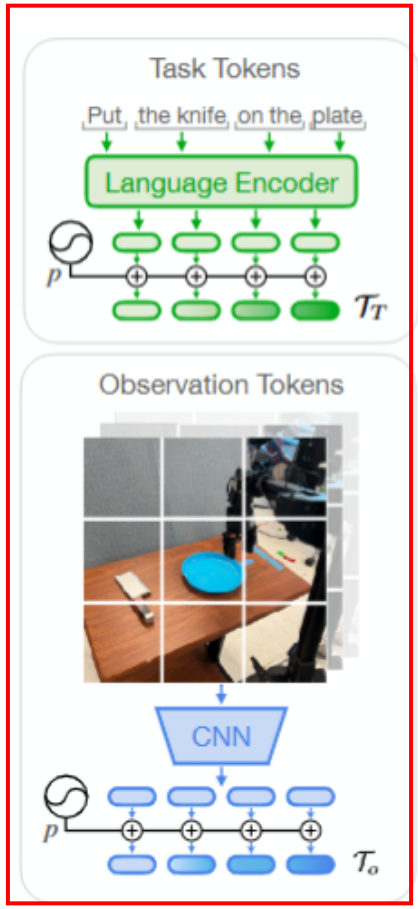


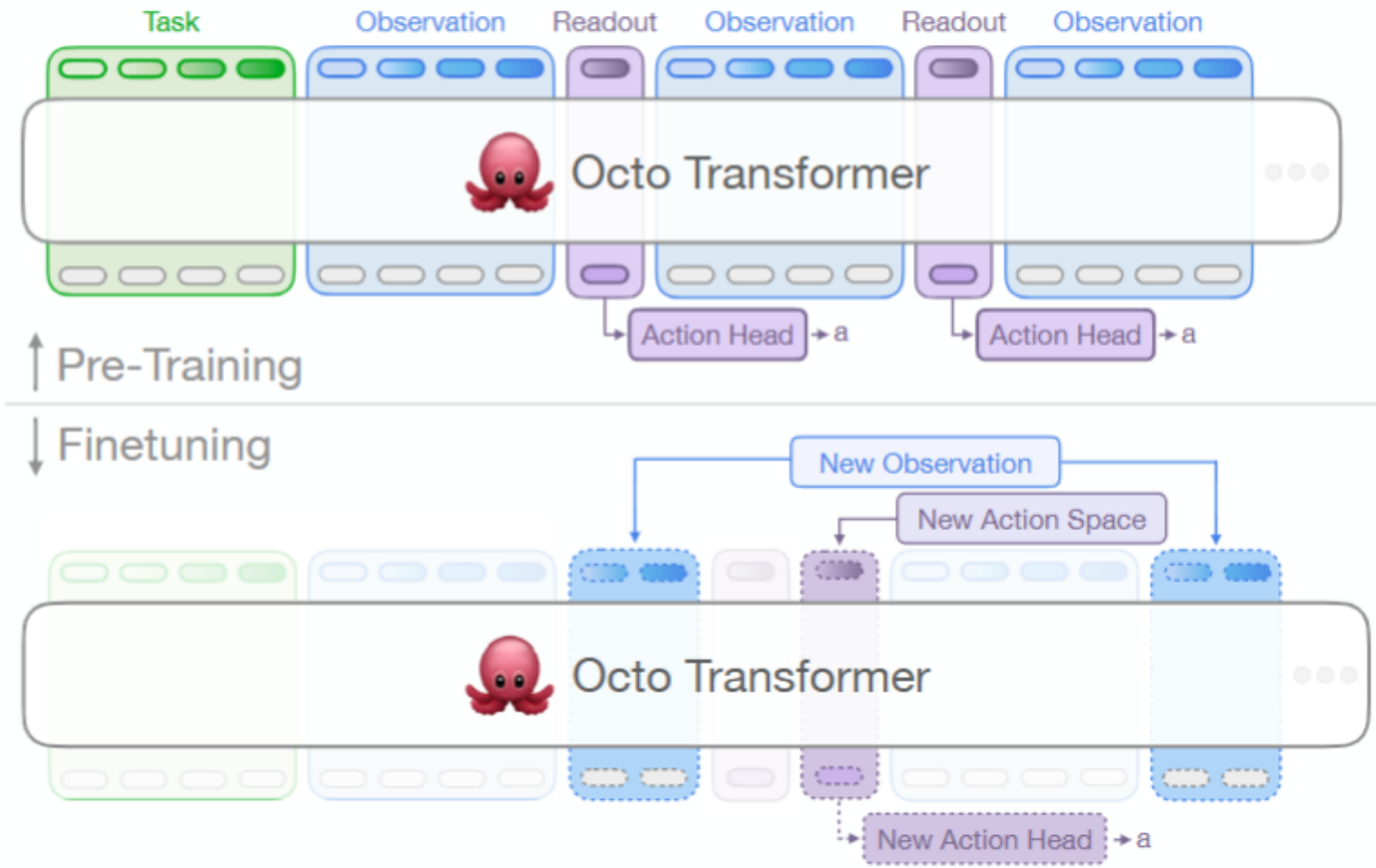
Methods:
A Common Policy Interface

- Robots differ in..
  - camera setup, image resolution, proprioceptive state, action dimension, gripper representation
- So $o^{(r)} \neq o^{(r')}$ and $a^{(r)} \neq a^{(r')}$ across robots
- How can one transformer process all of them?

Methods:
Shared Backbone, Robot-Specific Heads

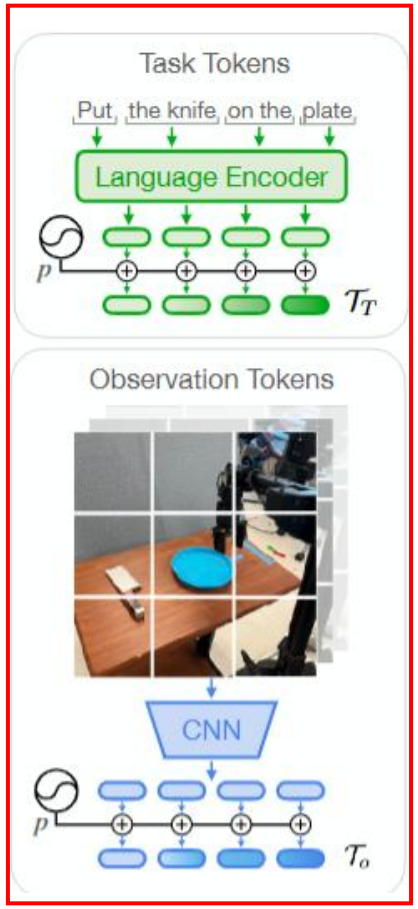
- 1) **Tasks and observations are tokenized** into a common sequence
  - heterogeneous robot data → common token representation





Methods:
Shared Backbone, Robot-Specific Heads

- 1) **Tasks and observations are tokenized** into a common sequence
  - Tasks can be given as language or goal image

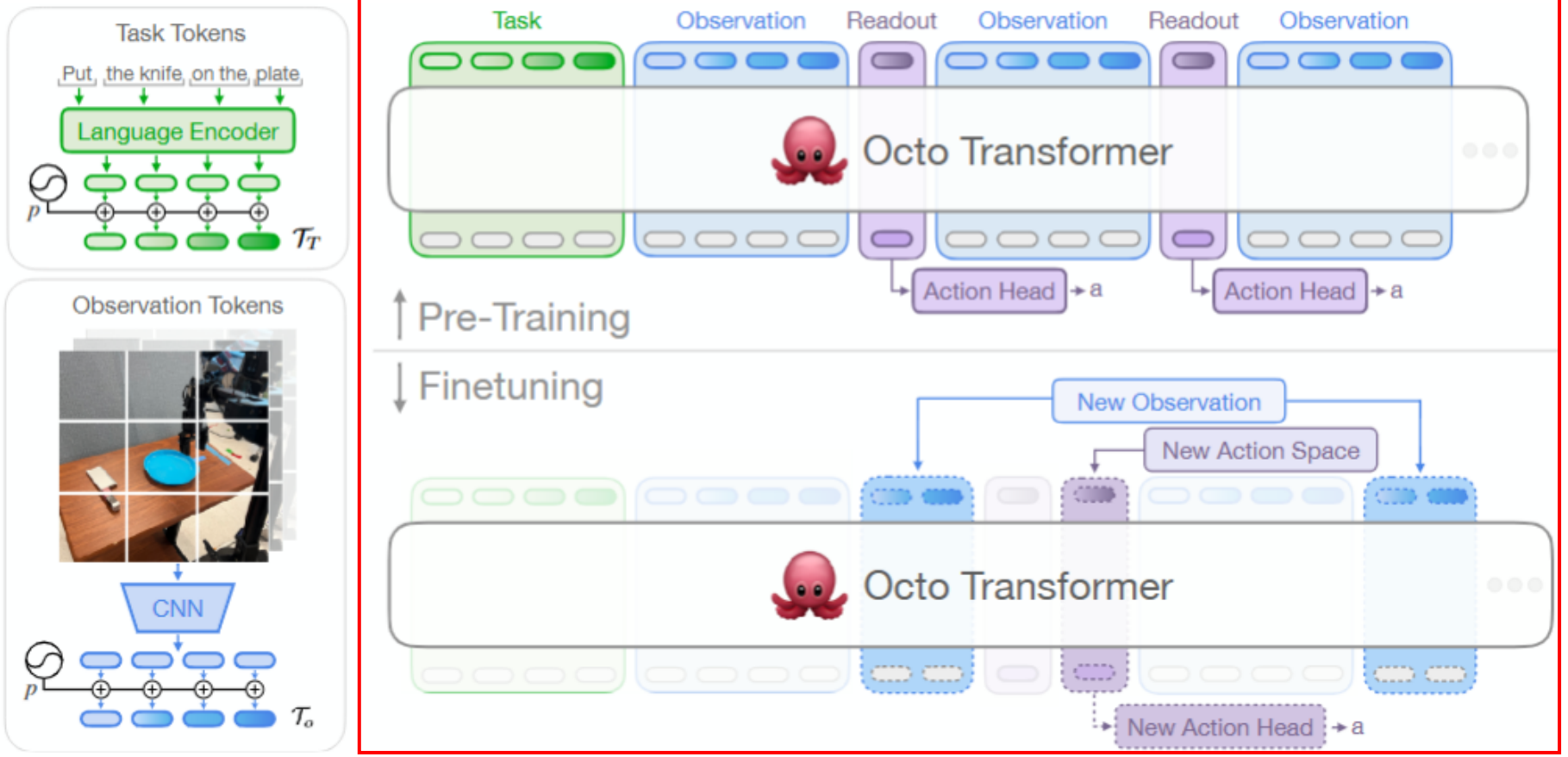


Task token
L=["pick","up","the","cup"] → task tokens

Observation token
Image → N visual tokens

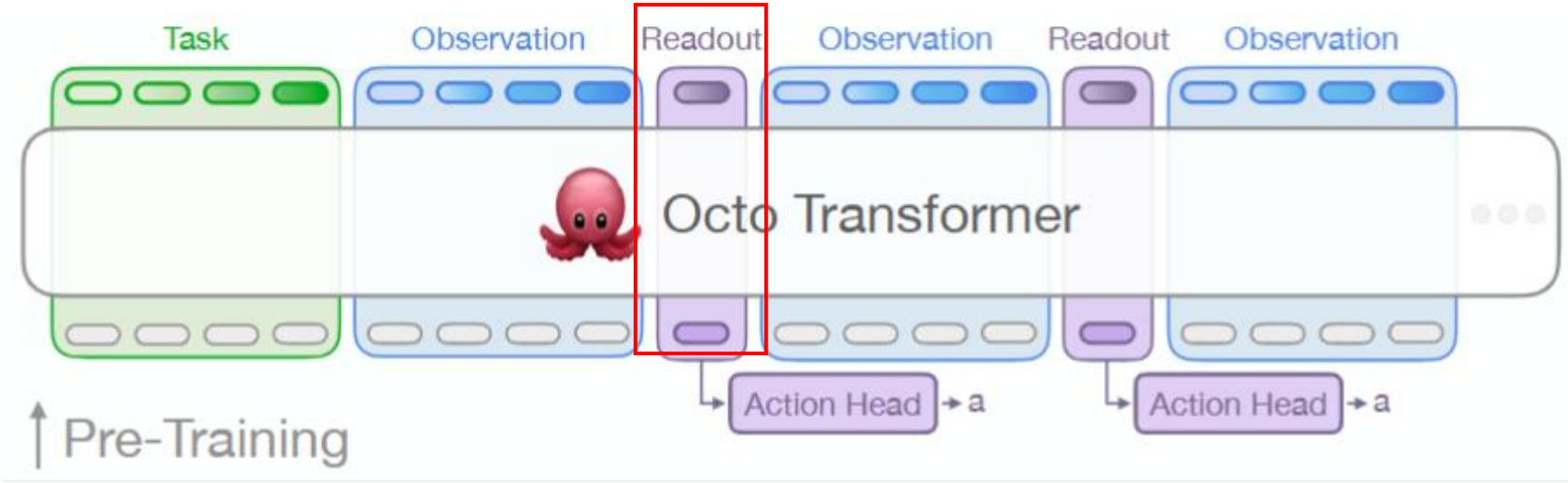
Methods:
Shared Backbone, Robot-Specific Heads

- 2) **Robot-specific action** head reads out the final prediction



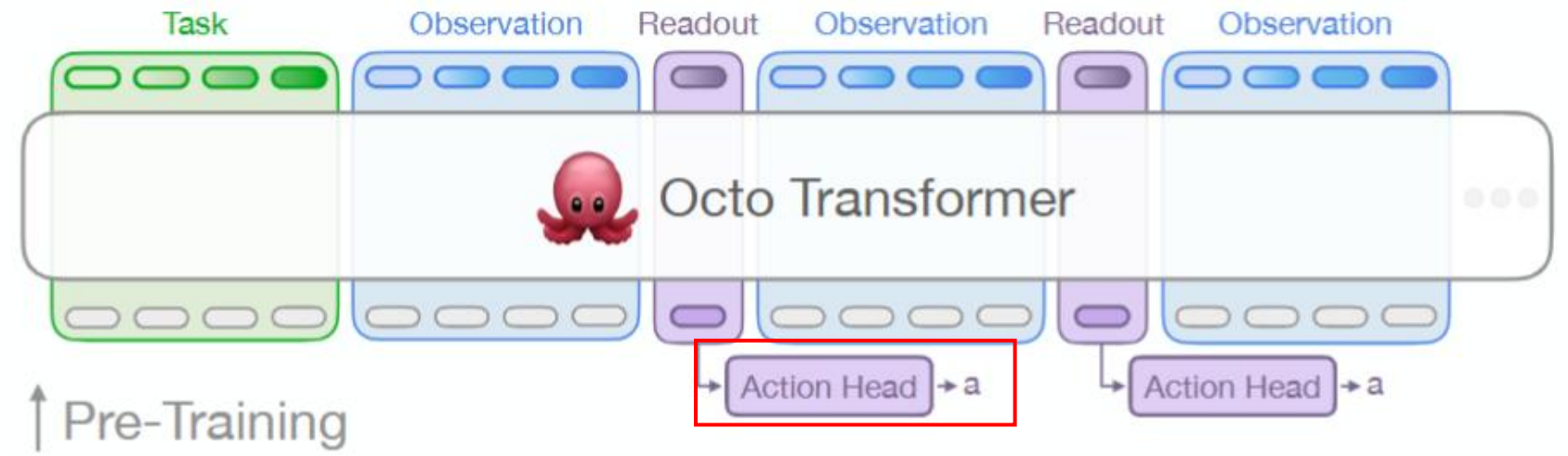
Methods:
Shared Backbone, Robot-Specific Heads

- 2) **Robot-specific action** head reads out the final prediction
  - Readout token
    - Reads a representation from the transformer via attention, producing an embedding that conditions the action head
    - Same role as the [CLS] token from BERT



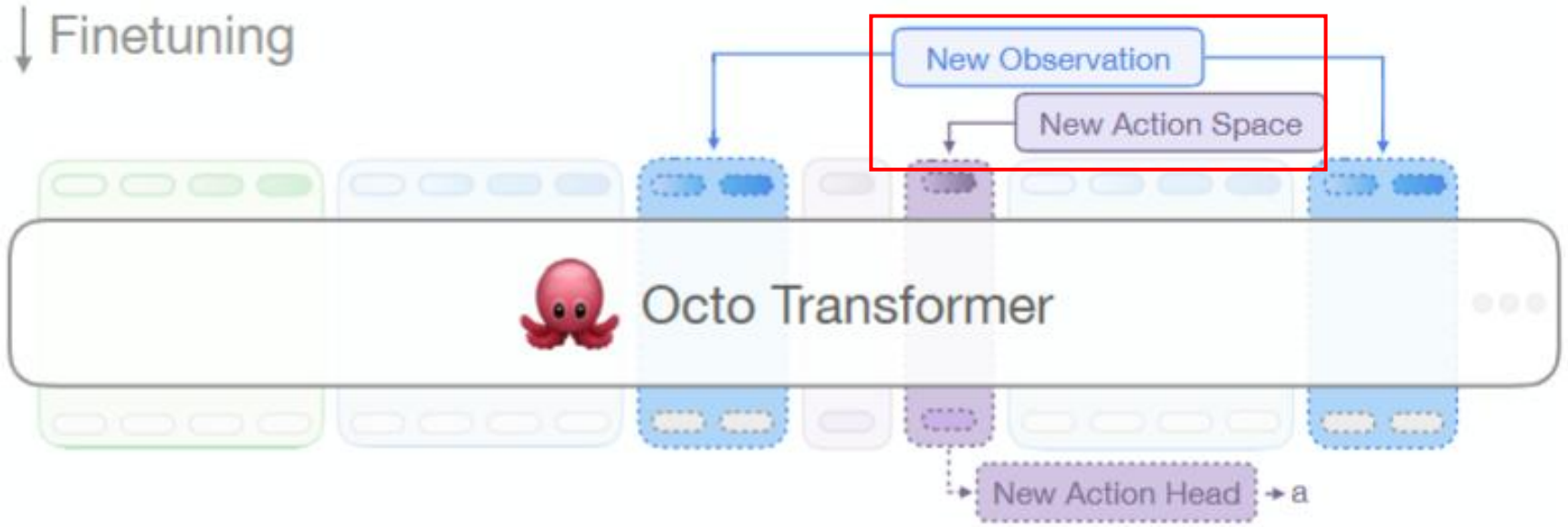
Methods:
Shared Backbone, Robot-Specific Heads

- 2) **Robot-specific action** head reads out the final prediction
  - Action head
    - Decodes that embedding into actions via a diffusion-based head
    - (not the paper's central contribution)



Methods:
Shared Backbone, Robot-Specific Heads

- 3) During **fine-tuning,**
  - Registers a new readout token + action head for the new robot, while the backbone carries over pretrained weights



Methods:
Why Split It This Way?

- Manipulation knowledge, e.g. "grasp the cup handle," can be shared across robots
- Low-level control, e.g. "move joint 3 by 0.17 rad," cannot
- **Octo's design**: **share the backbone, keep the interface robot-specific**

Experiments:
Zero-shot Evaluation

- Successfully adapted across all 9 platforms, including unseen action spaces

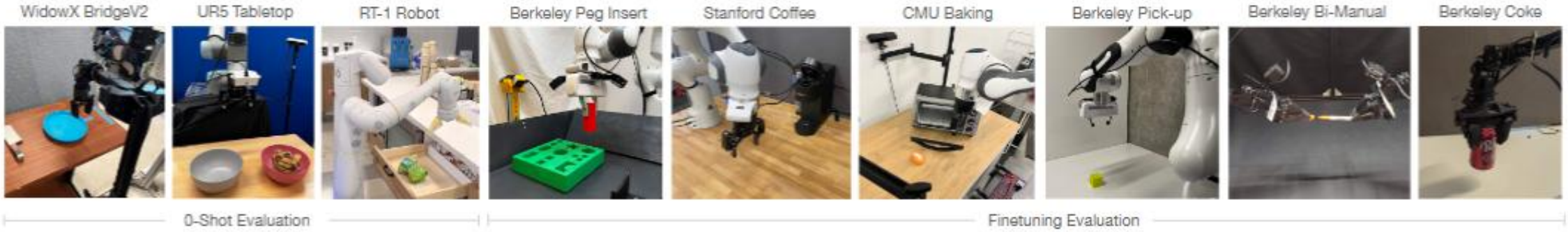


Fig. 7: **Evaluation Tasks.** Replicated from the main text for convenience. We evaluate Octo on 9 real robot setups across 4 institutions in zero-shot and finetuning scenarios.

Experiments:
Zero-shot Evaluation

- Compared against RT-1-X and RT-2-X on the same in-distribution language-conditioned tasks
  - Octo outperforms RT-1-X and performs comparably to the much larger RT-2-X

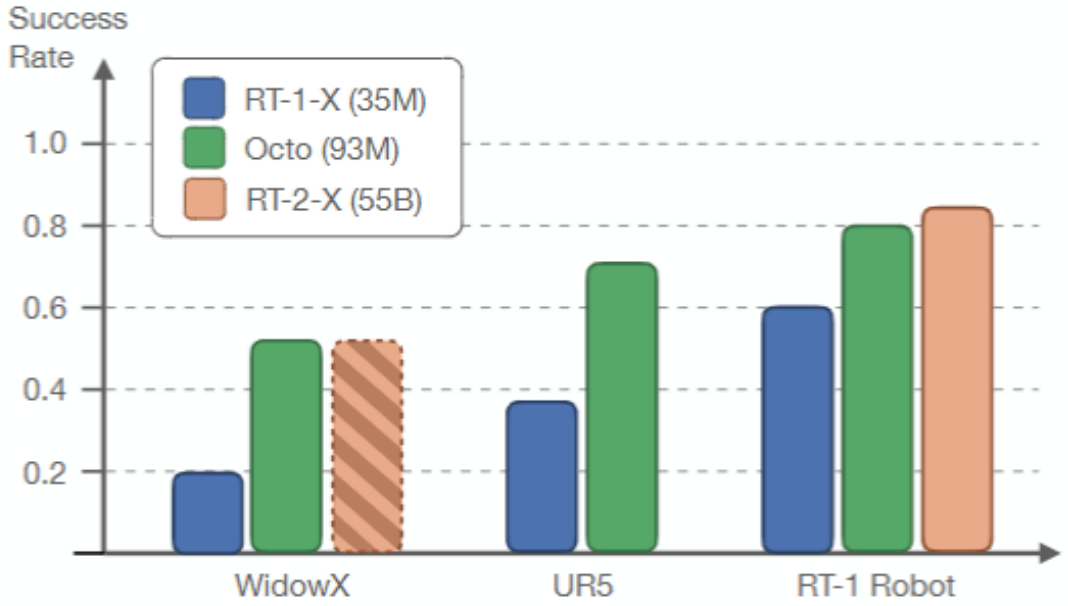


Fig. 5: Zero-Shot Evaluation. Out-of-the-box, Octo can control

Experiments:
Fine-tuning Evaluation

- 6 new domains, ~100 target demonstrations each, same hyperparameters
  - Compared against training from scratch and a pretrained visual representation (VC-1)

	Berkeley Insertion*	Stanford Coffee	CMU Baking	Berkeley Pick-Up†	Berkeley Coke	Berkeley Bimanual†	Average
ResNet+Transformer Scratch	10%	45%	25%	0%	20%	20%	20%
VC-1 [57]	5%	0%	30%	0%	10%	50%	15%
Octo (Ours)	**70%**	**75%**	**50%**	**60%**	**100%**	**80%**	**72%**

TABLE I: **Finetuning Evaluation.** Octo enables data-efficient finetuning to new domains and out-performs training from

Conclusion:
What Did Octo Actually Generalize?

- Zero-shot success was only ever measured within the pretraining distribution
  - **genuinely new robots always required fine-tuning !**

Conclusion:
What Did Octo Actually Generalize?

- **Open question:**
  - how deep does the transfer go? **low-level visual/action priors, or reusable manipulation concepts?**
  - e.g. low : image features, smooth action scales ..
    high : "grasp," "insert" ..

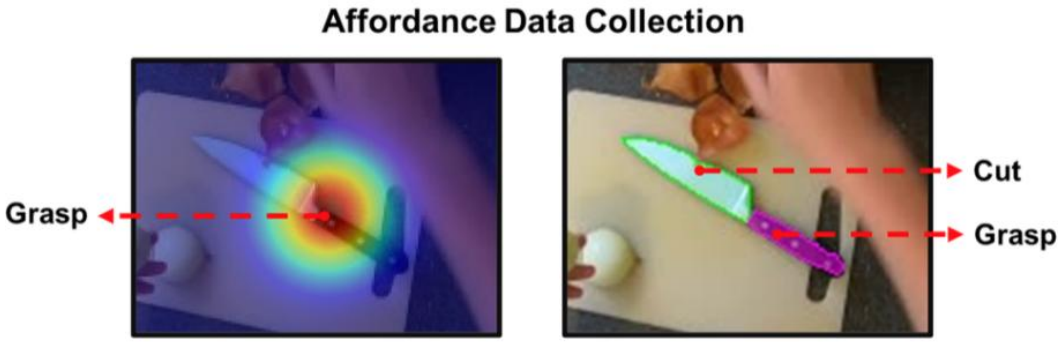
  - The paper doesn't test far enough to tell.

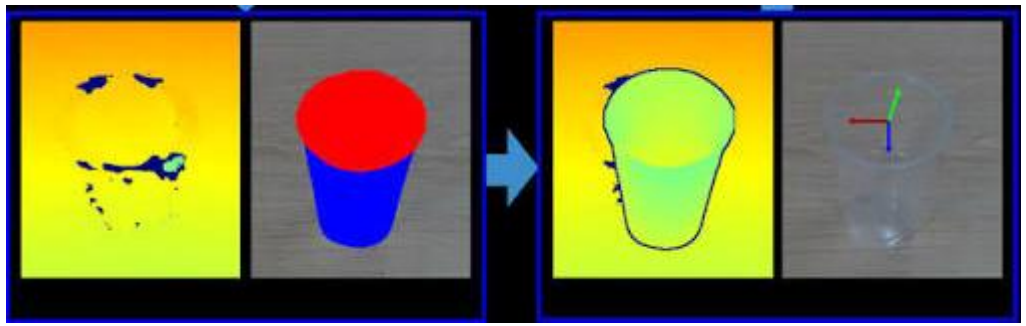
Synthesis

	ALOHA	**Open X-Embodiment**	**Octo**
Question	How do we collect manipulation data?	Can experience be shared ac ross robots?	Can one policy absorb that experience?
Scale	One robot	22 embodiments	800k trajectories
Key idea	Teleoperation + imitation	Cross-embodiment dataset	Shared pretrained policy
Main limitation	Narrow data distribution	No single shared policy yet	Still requires target-robot fine-tuning

Conclusion:
Closing Question

- one robot, one task, one dataset → many robots, many tasks, many datasets
- **Does more demonstration data yield general manipulation, or is a better abstraction still required?**
  - visual representations, task semantics, manipulation skills, contact strategies, or robot-independent affordances?





Thank You.

Questions welcome