Foundation Models for BC

ESE 6510

Antonio Loquercio

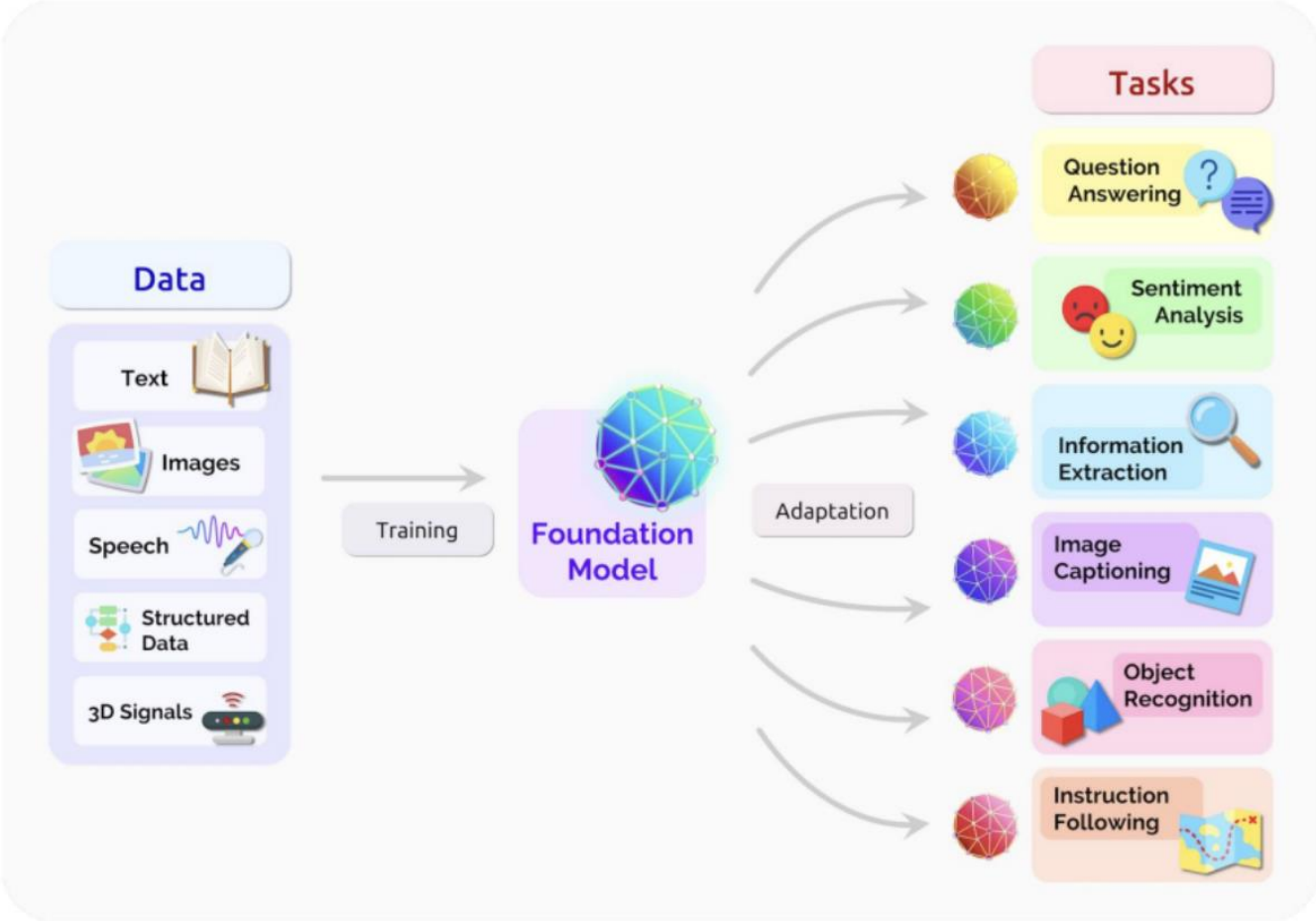
Source: Reddit

Behavioral Cloning: Agenda

- Theoretical Foundations
- Tools for Data Collection
- Algorithms
- **Leveraging foundation models**
- Challenges

What is a Foundation Model?

“A foundation model is any model that is trained on broad data (generally using self-supervision at scale) that can be adapted (zero-shot or fine-tuned) to a wide range of downstream tasks.”



Bommasani, et.al. On the Opportunities and Risks of Foundation Models, 2021

(Recap) What can I use Foundation Models for in Robotics?

- Planners (e.g., Palm-E).

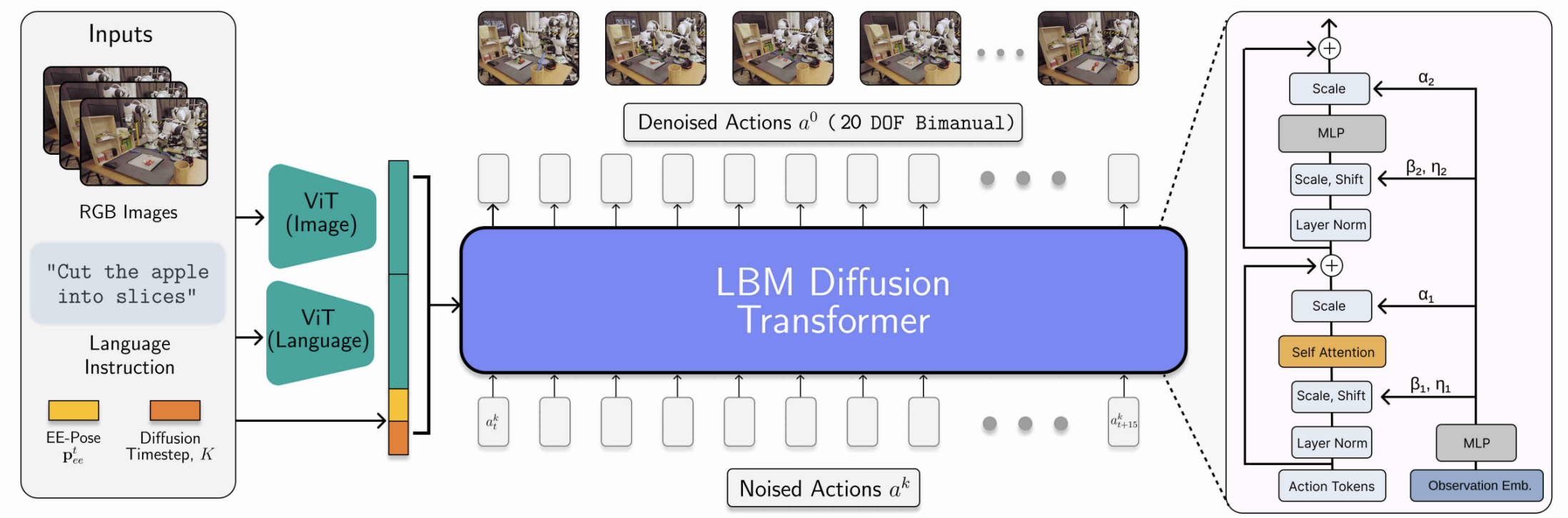
- Code Generators.

- Feature extractors
  - Producing goal-conditioned input features from which learning is easy/fast.

- Reward Models

- Data Generation

Foundation Models as Feature Extractors



A Careful Examination of Large Behavior Models for Multitask Dexterous Manipulation

Foundation Models as Feature Extractors

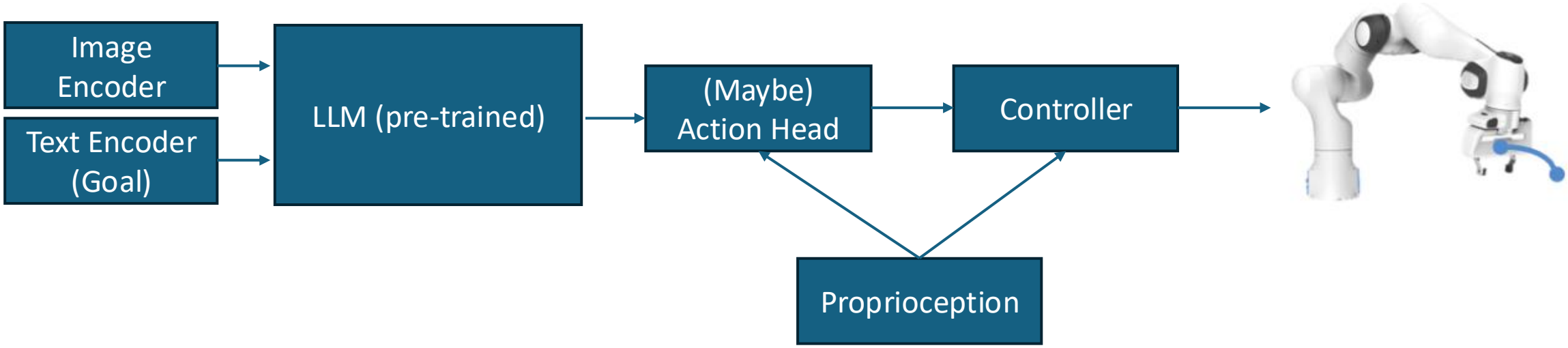
- Image Encoder
  - Often a combination of Dyno (low-level vision) and SigLIP (high-level vision).
  - Apparently works well even for other modalities, e.g., vision and touch.

- Text Encoder
  - use pre-trained tokenizers (e.g., T5, Llama, CLIP)

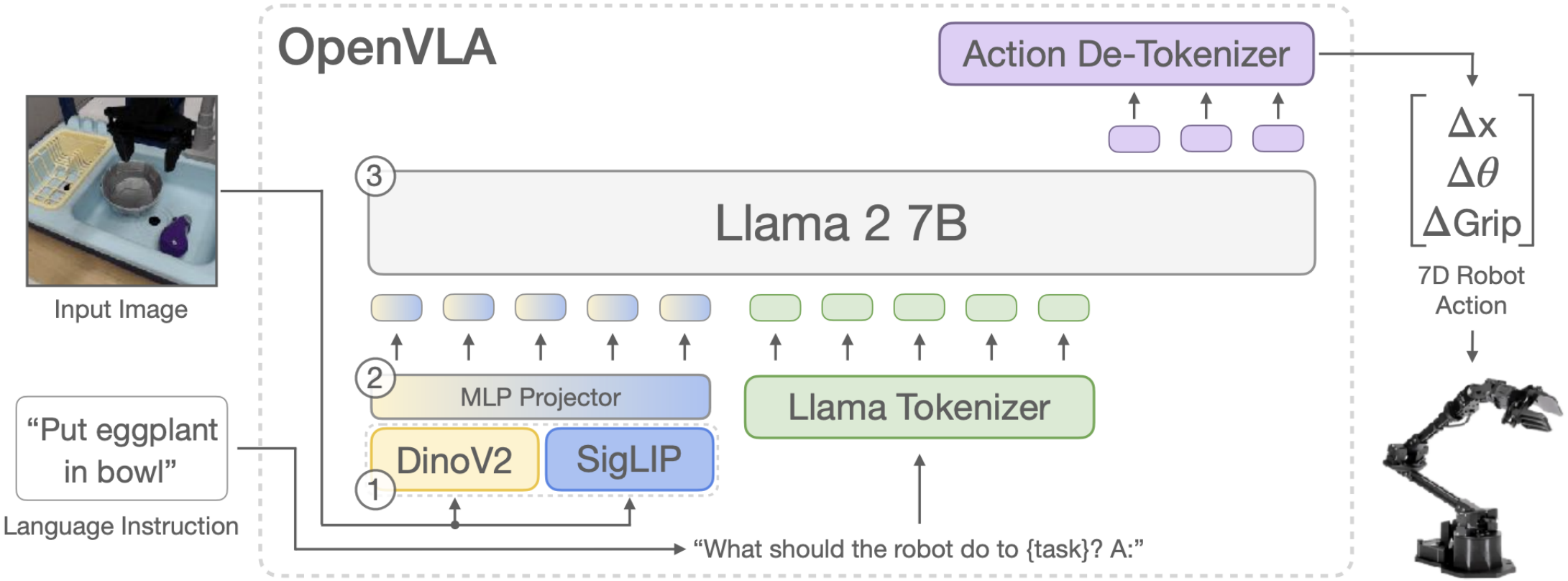
- General Idea:
  - Encoders are often heavy, so you don’t finetune them unless strictly necessary.

Vision-Language Action (VLA) Models

- The anatomy of a VLA

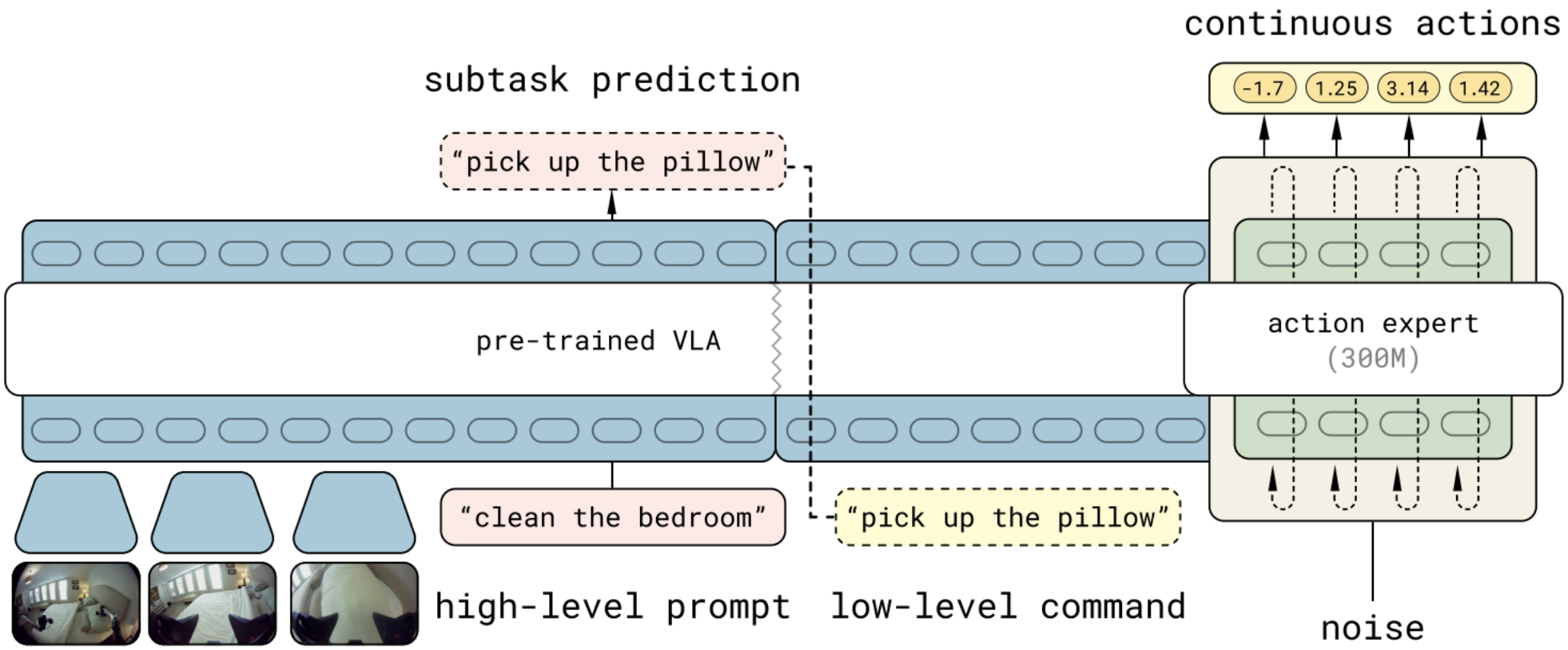


OpenVLA: No Action Head



OpenVLA: An Open-Source Vision-Language-Action Model, 2024

π: With Action Head



W/ Action Head is the most common choice in industry

- Nvidia Groot
- Figure Helix
- …

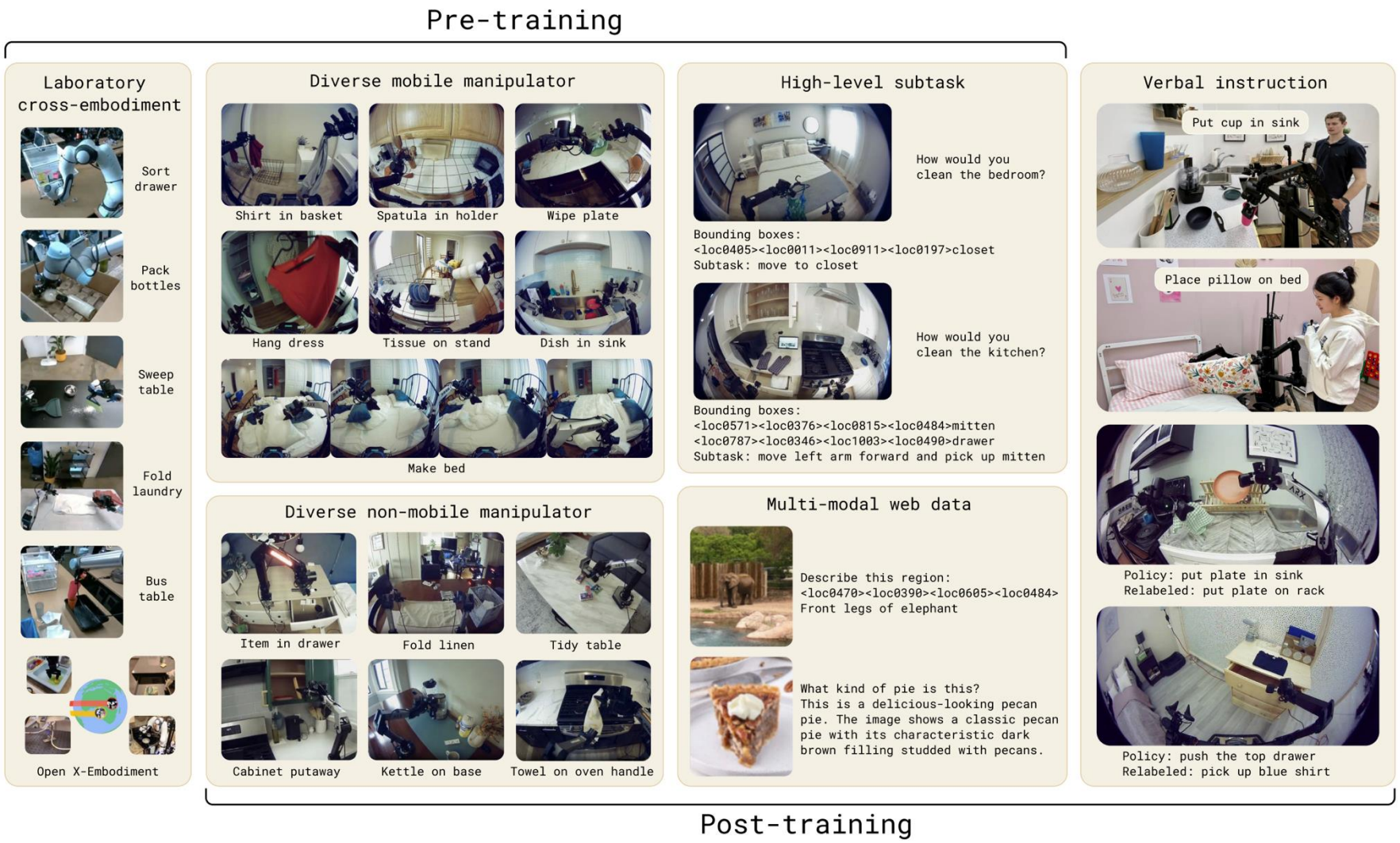
- Aside: Don’t call it system 1 and system 2, this is wrong.

VLA Training (still evolving, but at high level)

- Phase I
  - Pretrain on text, Visual Q&A data and general robot data.
  - Either from scratch, or finetuning an existing open source LLM (e.g., gemma).
  - Generally done without action head.

- Phase II
  - Add the action head and finetune the policy with robot data
  - Often repeated for different tasks

VLA Training Data



π0.5: a Vision-Language-Action Model with Open-World Generalization

Vision Language Action Model: General Wisdom

- The pre-training step really helps visual generalization.
- Training in different locations for the same task yields similar performance to training in the test locations.

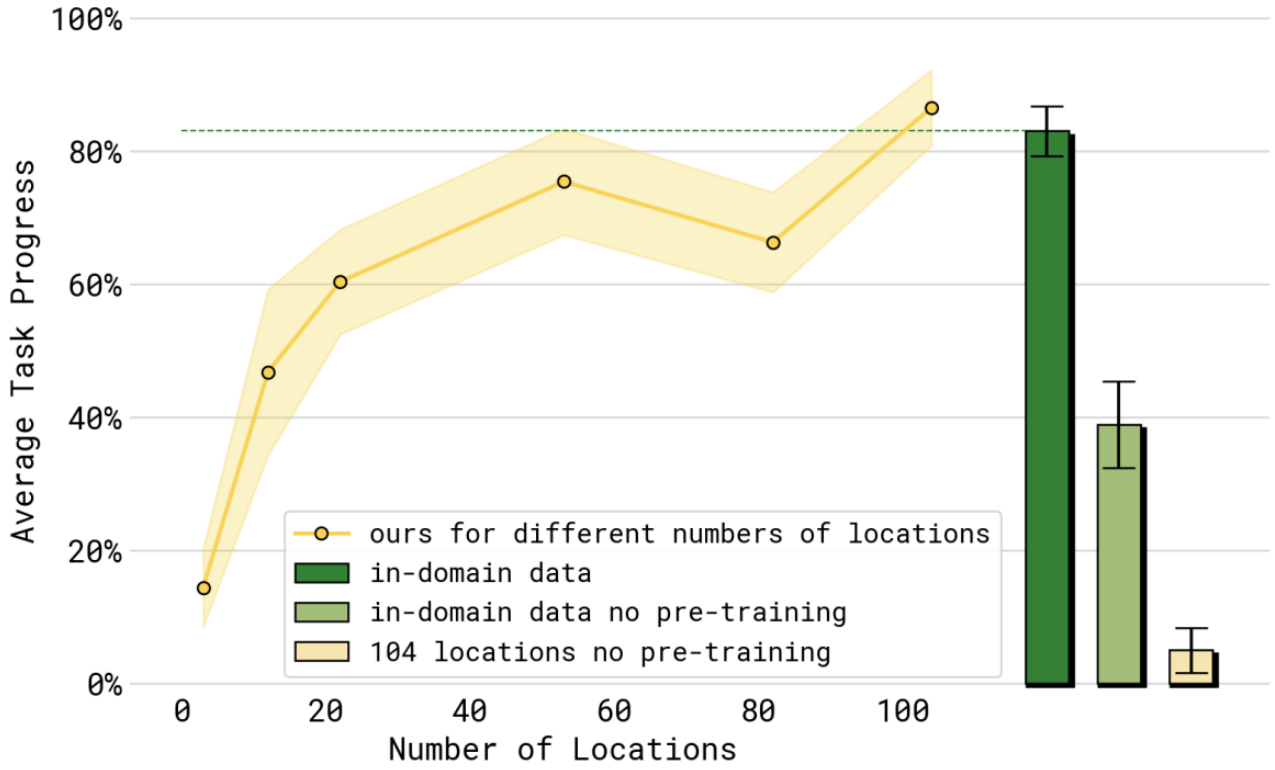


Fig. 8: **Evaluating performance with different numbers of locations.** Performance over the four test tasks — "dishes in sink", "items in drawer", "laundry basket", "make bed" — improves with more training environments. The dashed green line and green bar show a baseline model that includes the test homes in the training set. Compared to this model, our best model achieves similar performance, despite not seeing any data from the test homes.

Vision Language Action Model: General Wisdom

- Web-data at pretraining helps, but less than vision-language models.

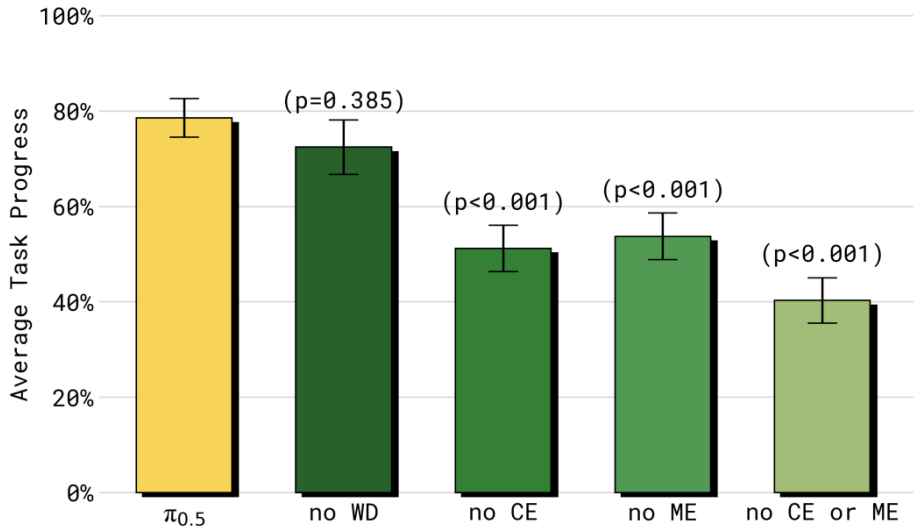


Fig. 10: **Training recipe ablations, mock homes.** We evaluate variants of our model that exclude different parts of the training mixture on all four test tasks (10 trials per policy and task). Including cross-embodiment data, both in diverse environments (ME) and for diverse tasks in laboratory settings (CE) is important for good performance, with large degradation when either or both of these data sources are removed. Web data (WD) does not make a significant difference in these experiments, but we will see in Figures 11 and 13 that it impacts object generalization and high-level performance.

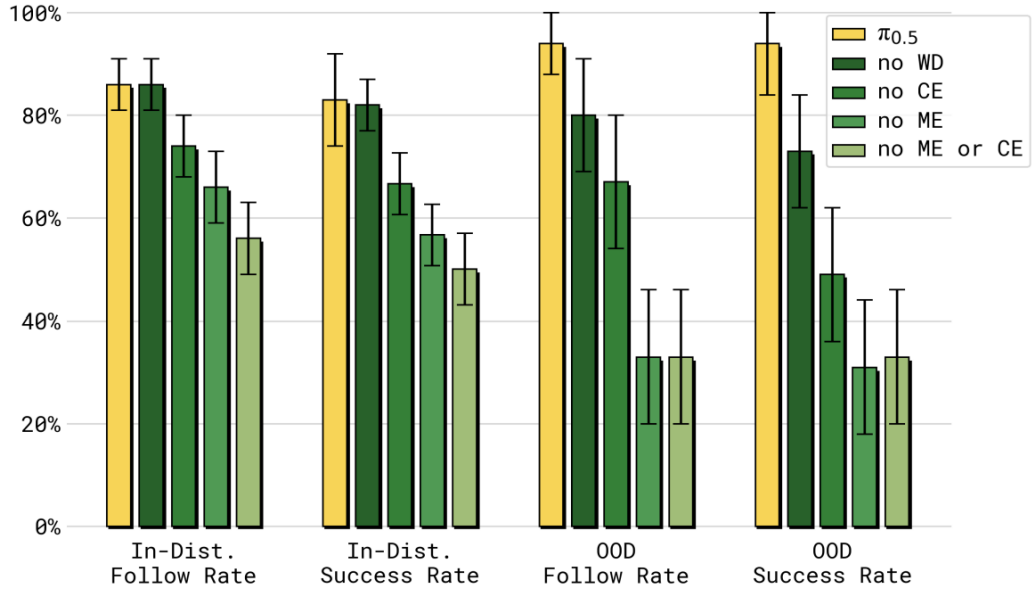
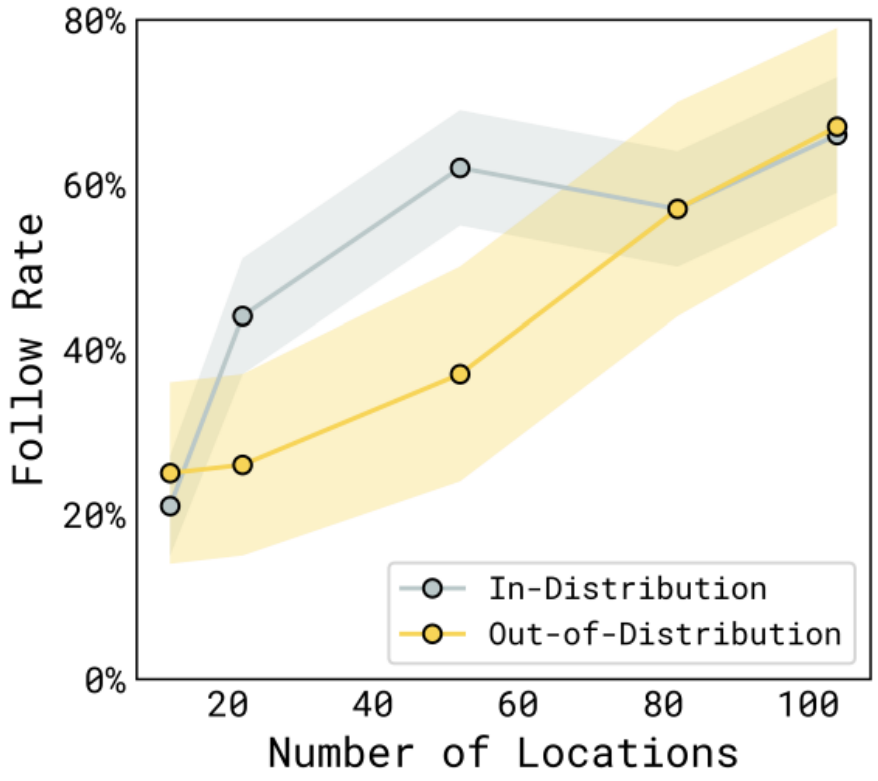
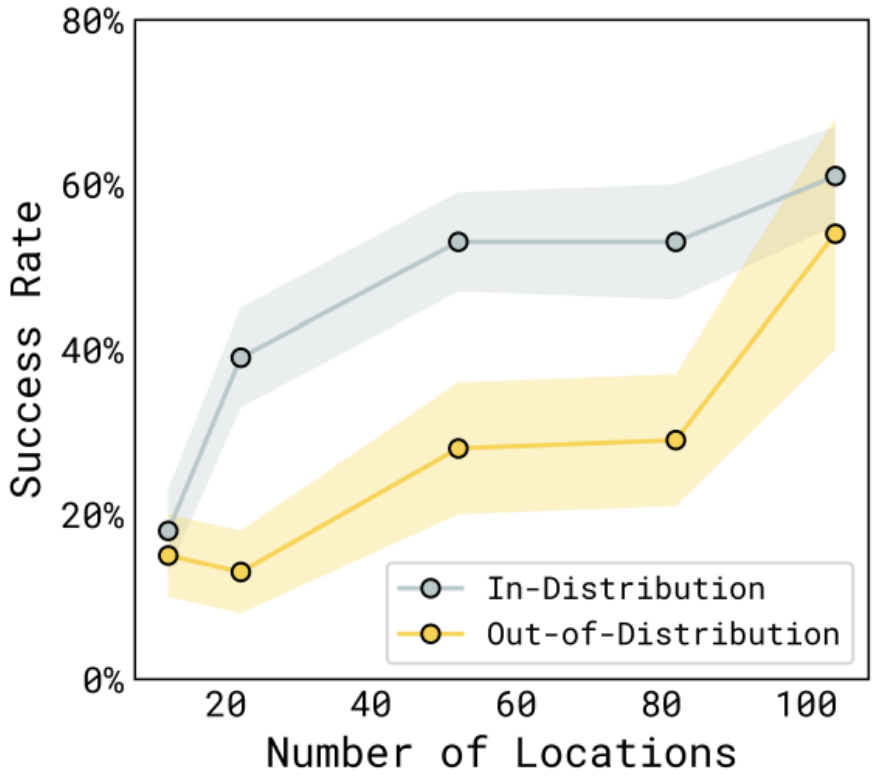


Fig. 11: **Training recipe ablations, language following.** Evaluating language following with in-distribution and out-of-distribution objects after training on different numbers of locations. Including web data (WD) is important for out-of-distribution (OOD) performance in particular. Cross-embodiment (CE) and diverse environment (ME) data both have a large impact on in-distribution and out-of-distribution performance.

Vision Language Action Model: General Wisdom

- Often the model does not do what you ask.
- Particularly an issue with small Phase II datasets.





Vision Language Action Model: General Wisdom

- Small technical decision matter a lot:
  - Number of diffusion steps
  - Data normalization
  - Data distribution
  - Grippers
  - …
- We’ll come back to this in the next class

(Recap) What can I use Foundation Models for in Robotics?

- Planners (e.g., Palm-E).
- Code Generators (see Yunzhu Li's Guest Lecture).
- Feature extractors
  - Producing goal-conditioned input features from which learning is easy/fast.
- Reward Models
- Data Generation

Foundation Models as Rewards

- Old idea
- Quite hard in practice
  - Reward Shaping?
  - Value functions barely generalize between policies, generalizing per task is even harder.
  - Look at the value function loss of your drone racing project…

Universal Value Function Approximators

Tom Schaul SCHAUL@GOOGLE.COM
Dan Horgan HORGAN@GOOGLE.COM
Karol Gregor KAROLG@GOOGLE.COM
David Silver DAVIDSILVER@GOOGLE.COM
Google DeepMind, 5 New Street Square, EC4A 3TW London

Abstract

Value functions are a core component of reinforcement learning systems. The main idea is to to construct a single function approximator $V(s;\theta)$ that estimates the long-term reward from any state s, using parameters θ. In this paper we introduce *universal* value function approximators (UVFAs) $V(s,g;\theta)$ that generalise not just over states s but also over goals g. We develop an efficient technique for supervised learning of UVFAs, by factoring observed values into separate embedding vectors for state and goal, and then learning a mapping from s and g to these factored embedding vectors. We show how this technique may be incorporated into a reinforcement learning algorithm that updates the UVFA solely from observed rewards. Finally, we demonstrate that a UVFA can successfully generalise to previously unseen goals.

how to evaluate or control a specific aspect of the environment (e.g. progress toward a waypoint). A collection of general value functions provides a powerful form of knowledge representation that can be utilised in several ways. For example, the *Horde* architecture (Sutton et al., 2011) consists of a discrete set of value functions ('demons'), all of which may be learnt simultaneously from a single stream of experience, by bootstrapping off-policy from successive value estimates (Modayil et al., 2014). Each value function may also be used to generate a policy or option, for example by acting greedily with respect to the values, and terminating at goal states. Such a collection of options can be used to provide a temporally abstract action-space for learning or planning (Sutton et al., 1999). Finally, a collection of value functions can be used as a predictive representation of state, where the predicted values themselves are used as a feature vector (Sutton & Tanner, 2005; Schaul & Ring, 2013).

In large problems, the value function is typically repre-

Foundation Models as Rewards: Formulations

- Dense Reward Prediction
  - Direct
  - Code Generation (Mainly in simulation)

- Sparse Reward Prediction
  - Success / Failure
  - Preference

- Value Prediction
  - “Time” to finish

- Often the foundation model is finetuned for the task of reward prediction, but it can do zero-shot with some tricks.

Foundation Models as Zero-Shot Dense Reward Models

$$R_{\text{CLIP}}(s) = \frac{\text{CLIP}_L(l) \cdot \text{CLIP}_I(\psi(s))}{\|\text{CLIP}_L(l)\| \cdot \|\text{CLIP}_I(\psi(s))\|}.$$

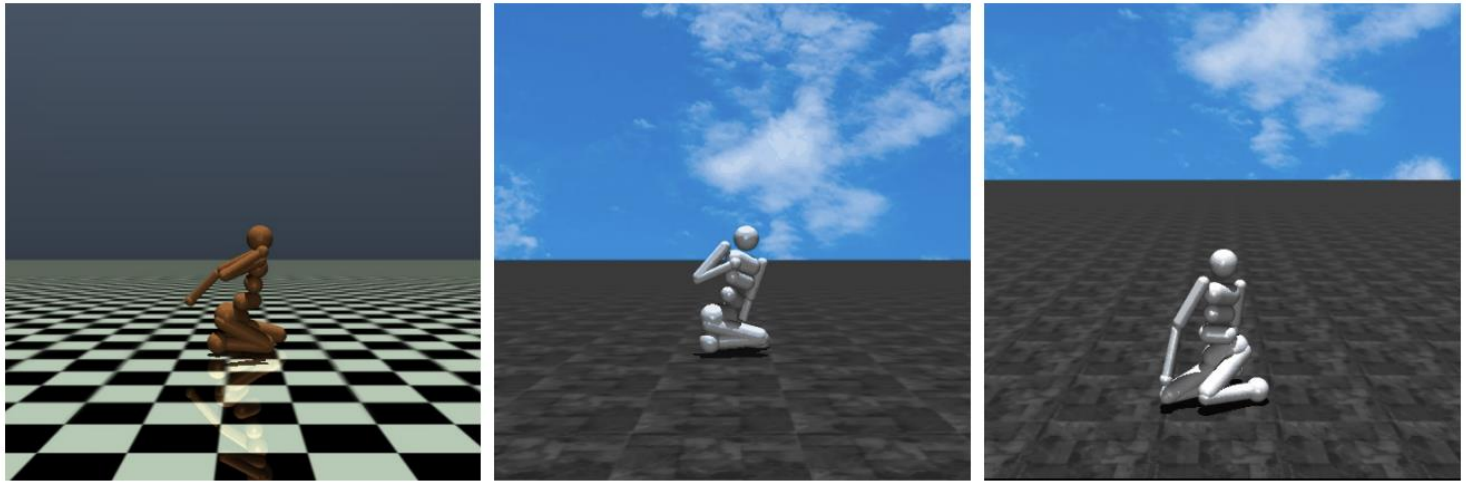
l = “stand up with raised hands”

Vision-Language Models are Zero-shot Reward Models for Reinforcement Learning, 2024

Foundation Models as Zero-Shot Dense Reward Models

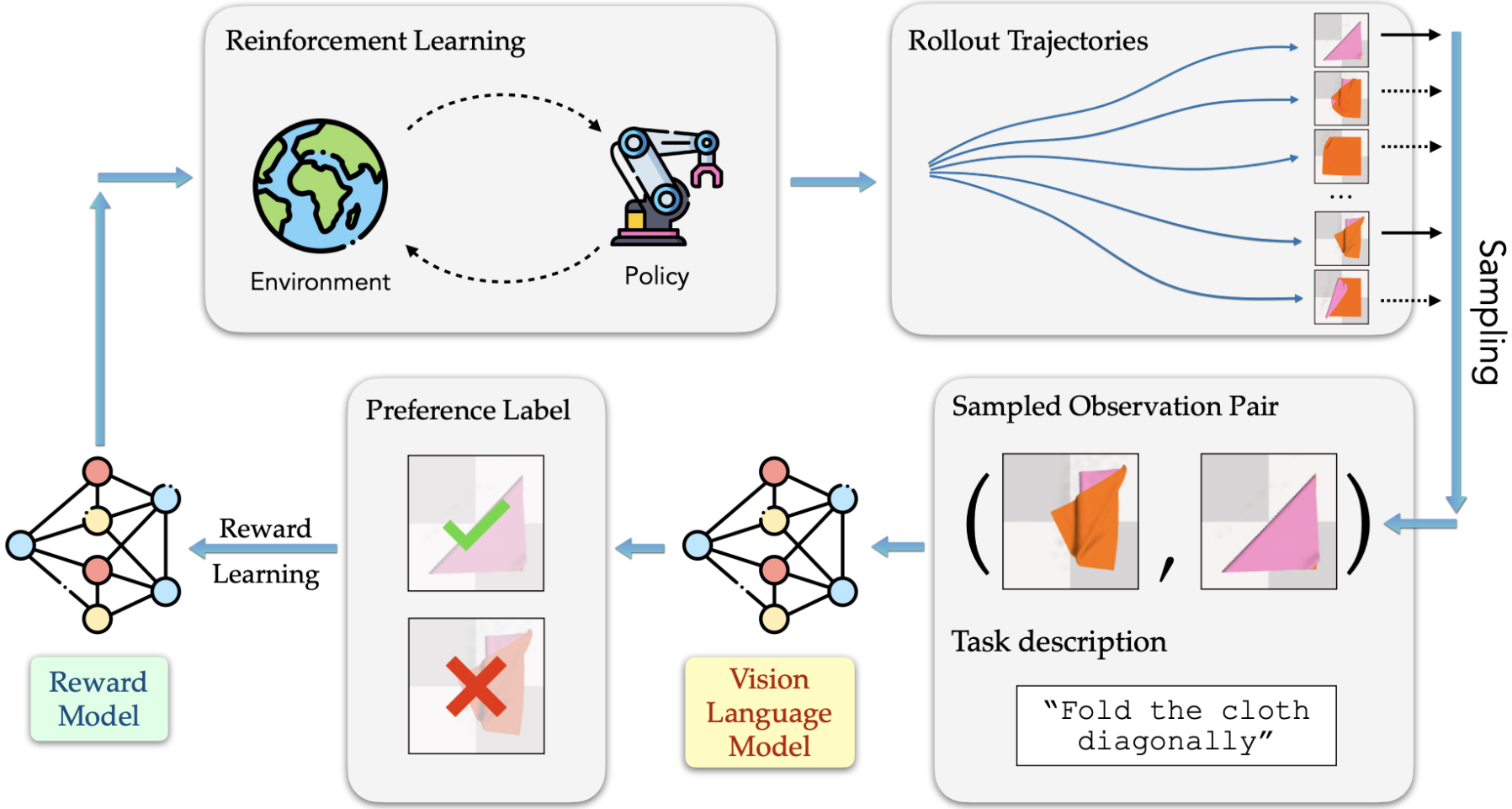
- Works okaish (cannot learn many tasks)
- Sensitive to prompting

Camera Angle	Textures	Success Rate
Original	Original	36%
Original	Modified	91%
Modified	Modified	100%



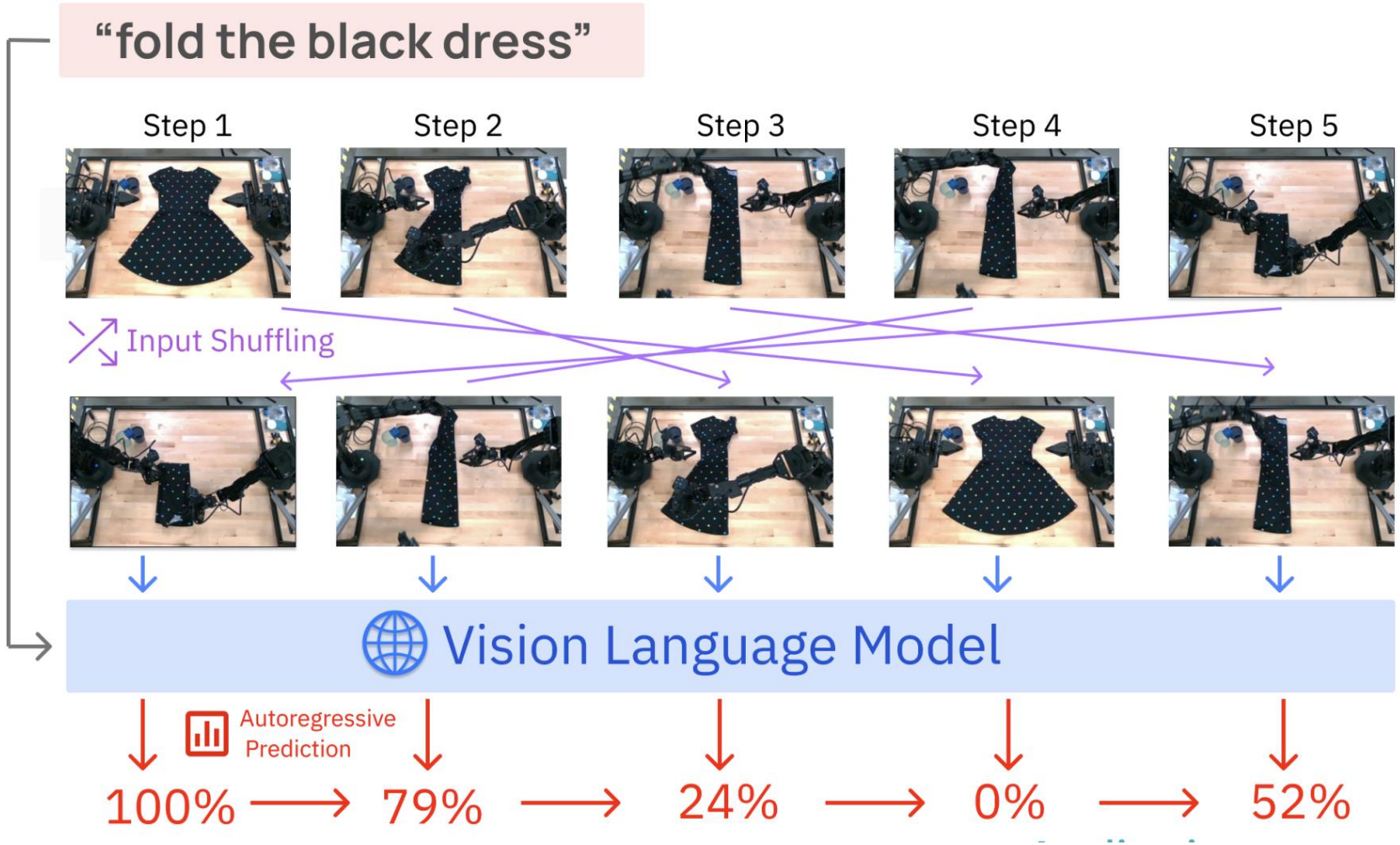
(a) Original (b) Modified textures (c) Modified textures & camera angle

Foundation Models as Zero-Shot Sparse Reward Models



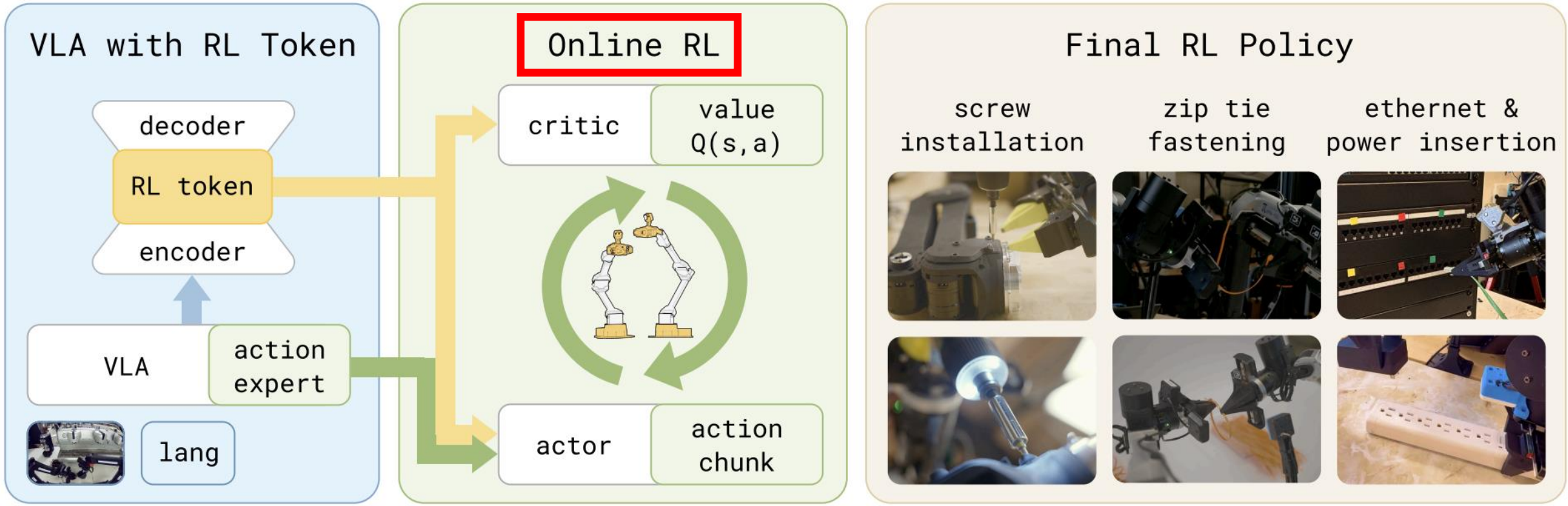
RL-VLM-F: Reinforcement Learning from Vision Language Foundation Model Feedback, 2024

Foundation Models as Zero-Shot Value Models



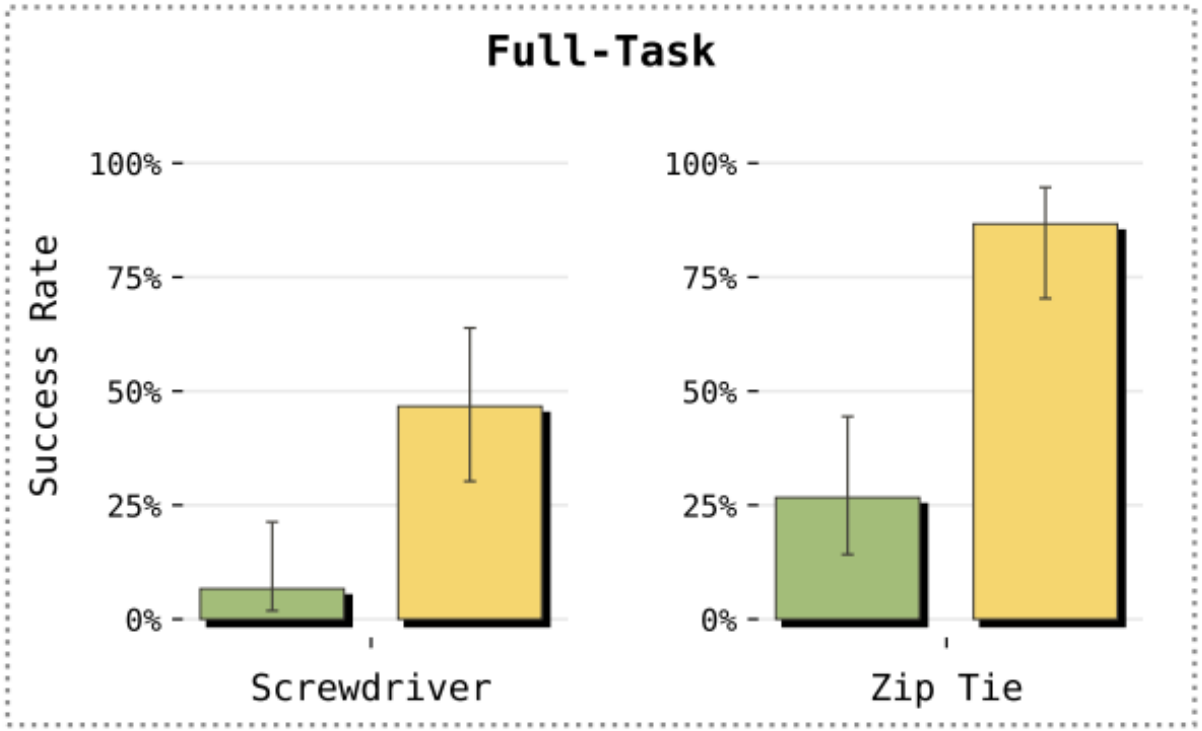
Vision-Language Models are In-Context Value Learners, 2025

Finetuning a Model for Value & Action Prediction

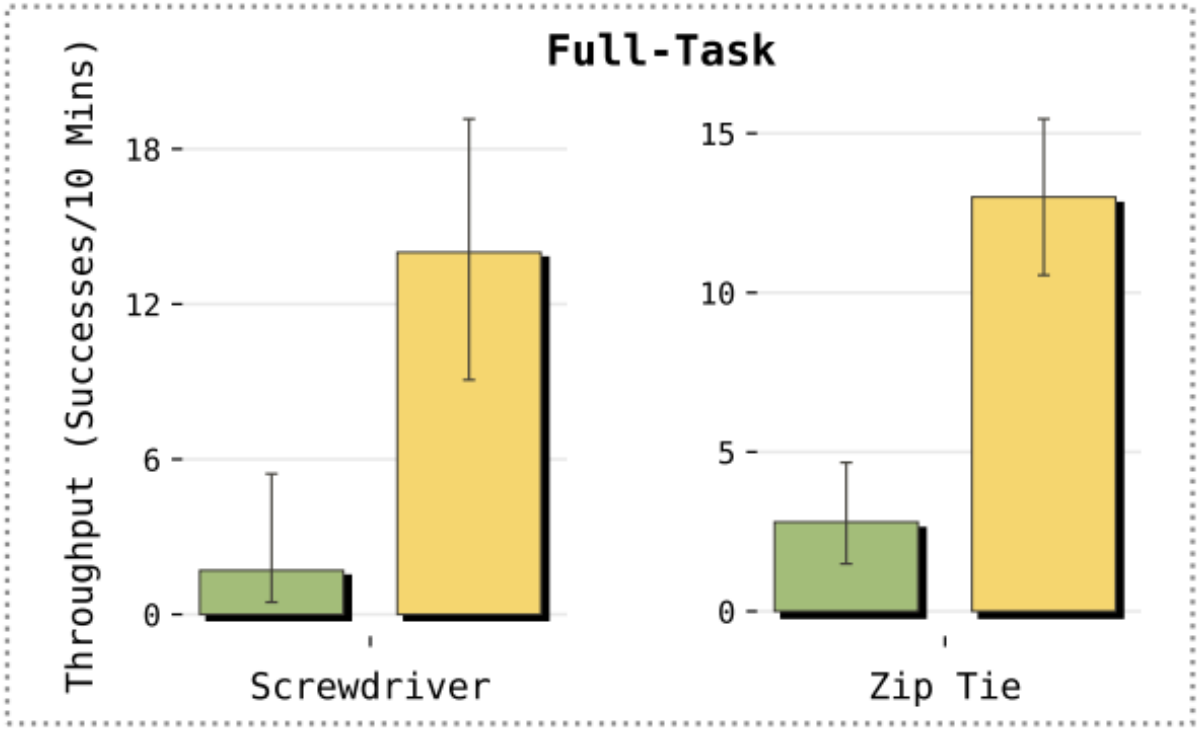


RL Token: Bootstrapping Online RL with Vision-Language-Action Models

Finetuning a Model for Value & Action Prediction



Increase Success Rate



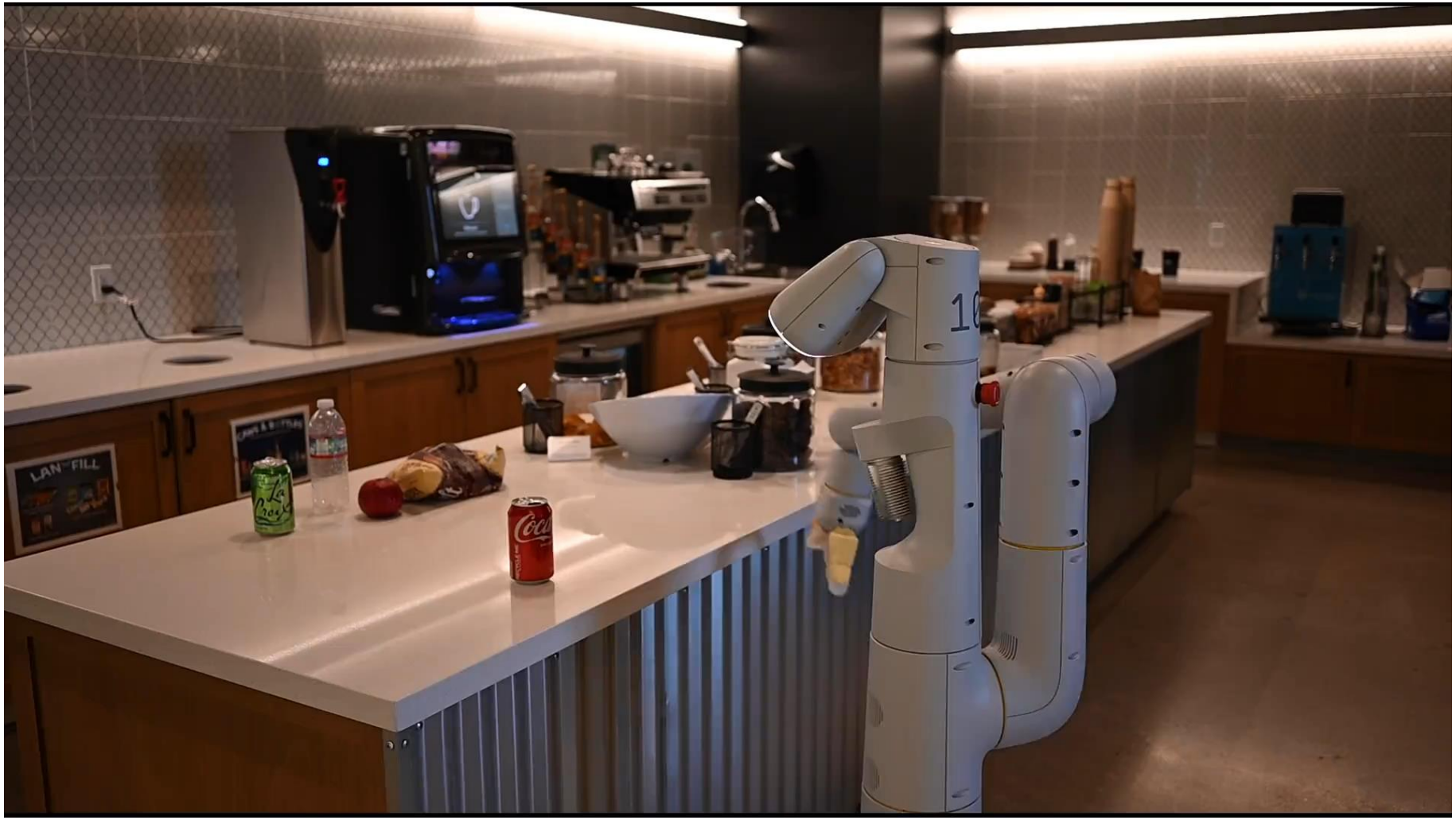
Increase Execution Speed

RL Token: Bootstrapping Online RL with Vision-Language-Action Models

(Recap) What can I use Foundation Models for in Robotics?

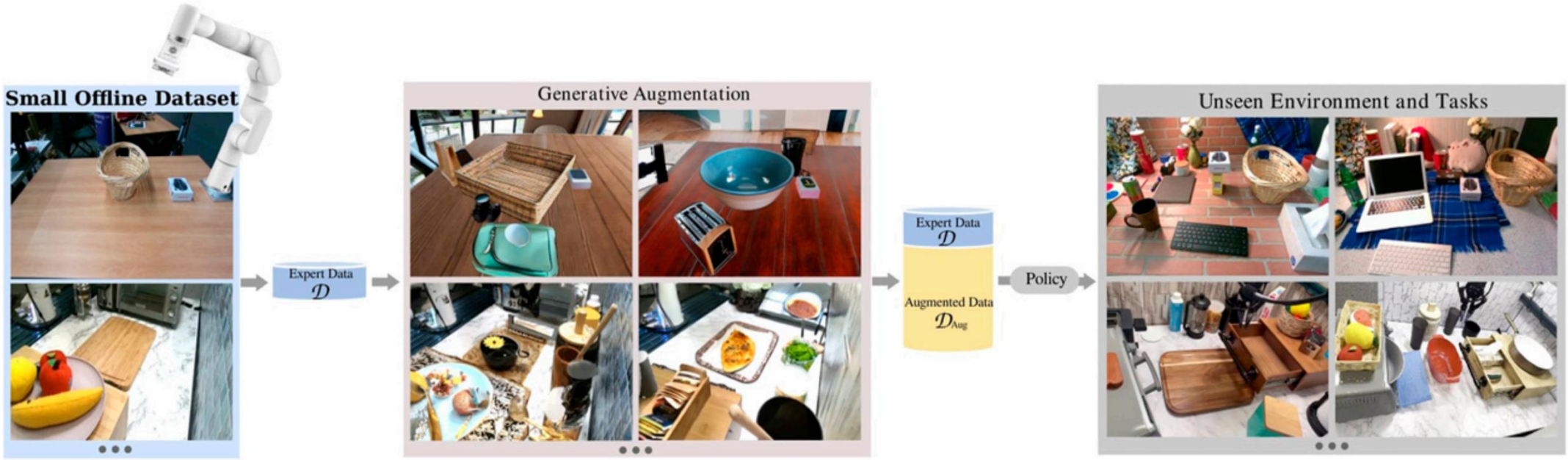
- Planners (e.g., Palm-E).
- Code Generators.
- Feature extractors
  - Producing goal-conditioned input features from which learning is easy/fast.
- Reward Models
- Data Generation

Foundation Models for Data Augmentation



Scaling Robot Learning with Semantically Imagined Experience, 2023

Foundation Models for Data Augmentation



Semantically controllable augmentations for generalizable robot learning, 2025

Foundation Models for Data Augmentation

- Mainly focuses on visual augmentation. But there are instances of action augmentation (e.g., moving objects and adapting the trajectory).

- Nice idea, but hard to do in practice:
  - Which objects are important for the task? Which one can be randomized?
  - How to make sure the action does not change? (think about lifting a full bottle vs an empty bottle)

- Not standard in SOTA robot learning pipelines (that I am aware of)

Behavioral Cloning: Agenda

- Theoretical Foundations
- Tools for Data Collection
- Algorithms
- Leveraging foundation models
- **Challenges**