Robotics

CS6960 MultiModal LLM Agents

Kenneth Marino

Slide Credits: Chelsea Finn, others

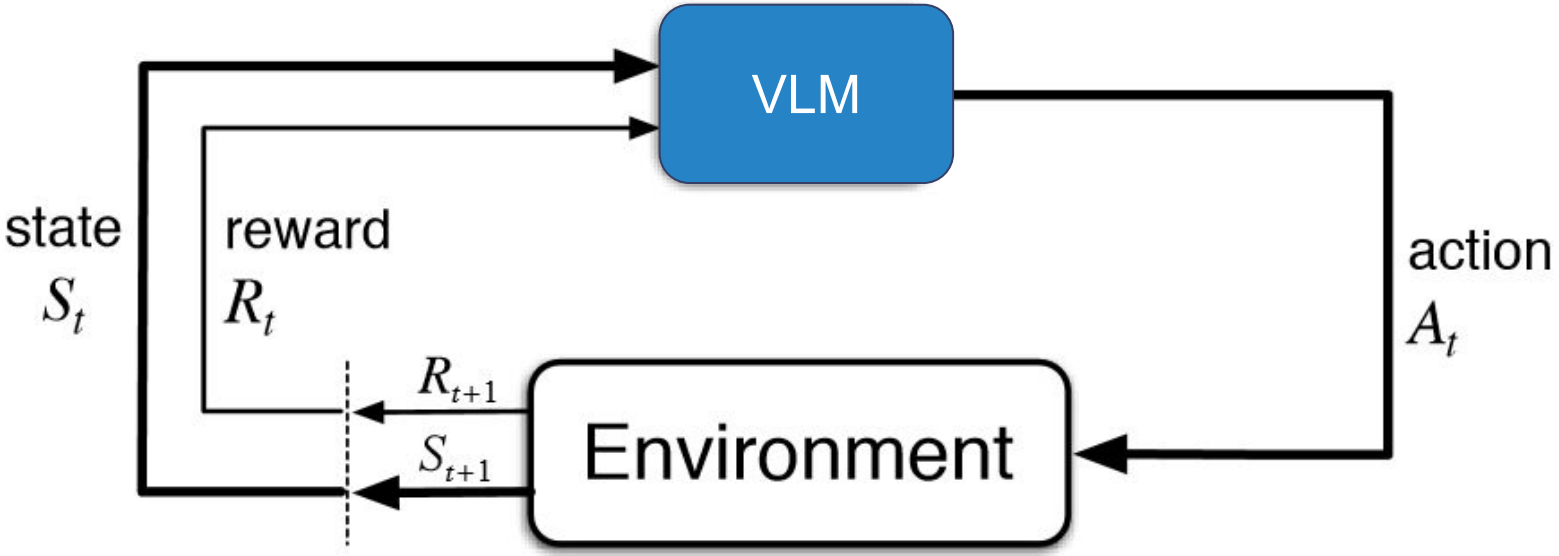
Announcements

- Projects
  - Keep making progress!
- Presentations
  - Only a few more left
- Two options
  - People present more (raise your hands if yes)
  - More time for project activities

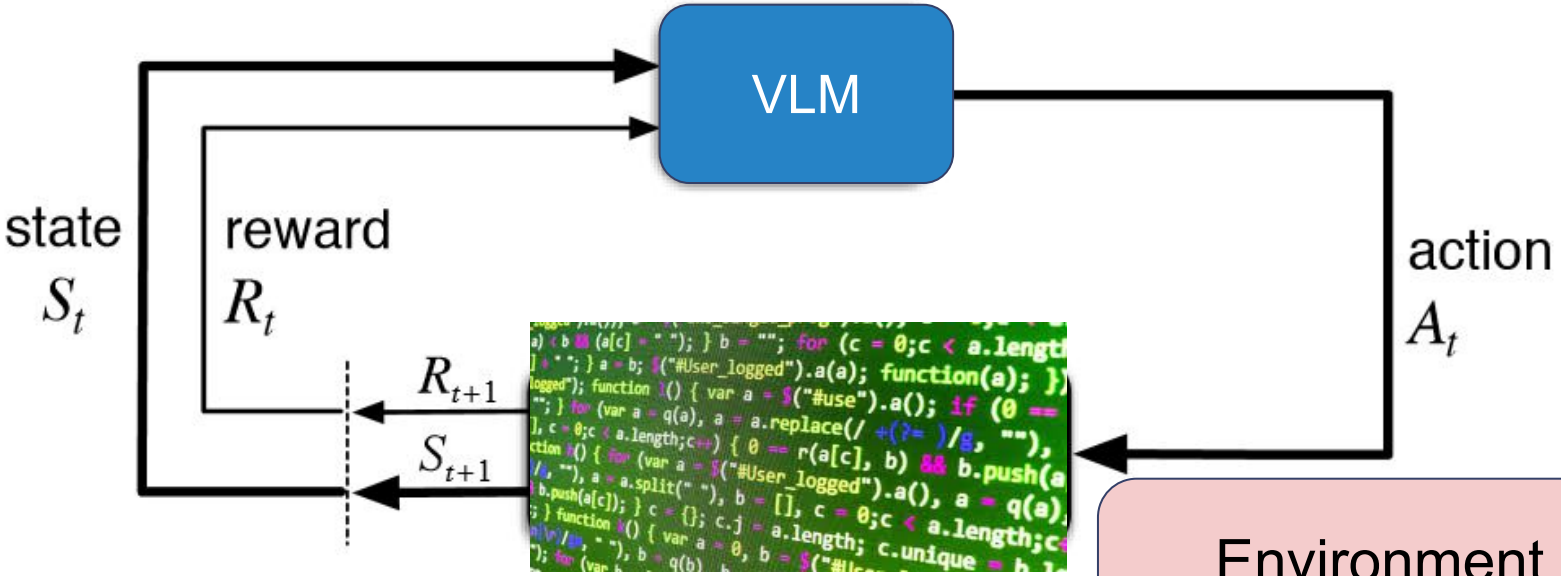
Any Questions

Final Topic: Robotics!

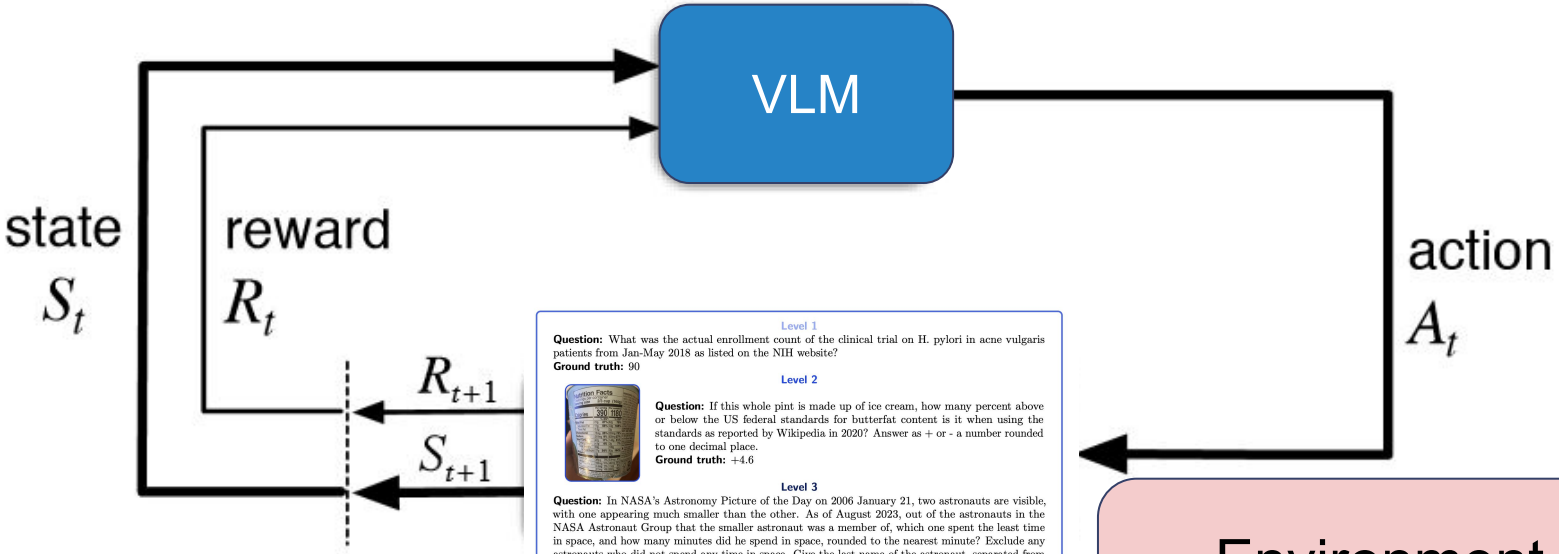
Recall: Familiar Picture



What does VLM agent actually look like?

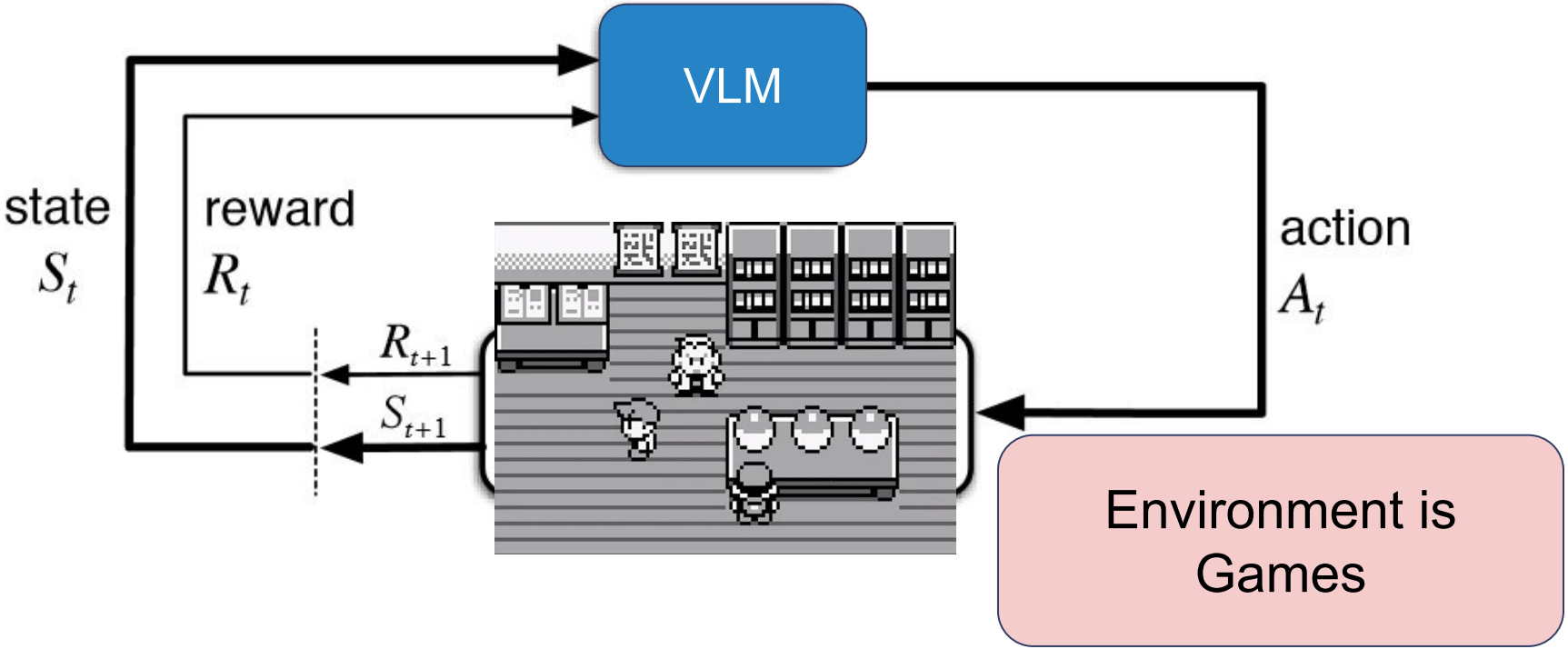


What does VLM agent actually look like?

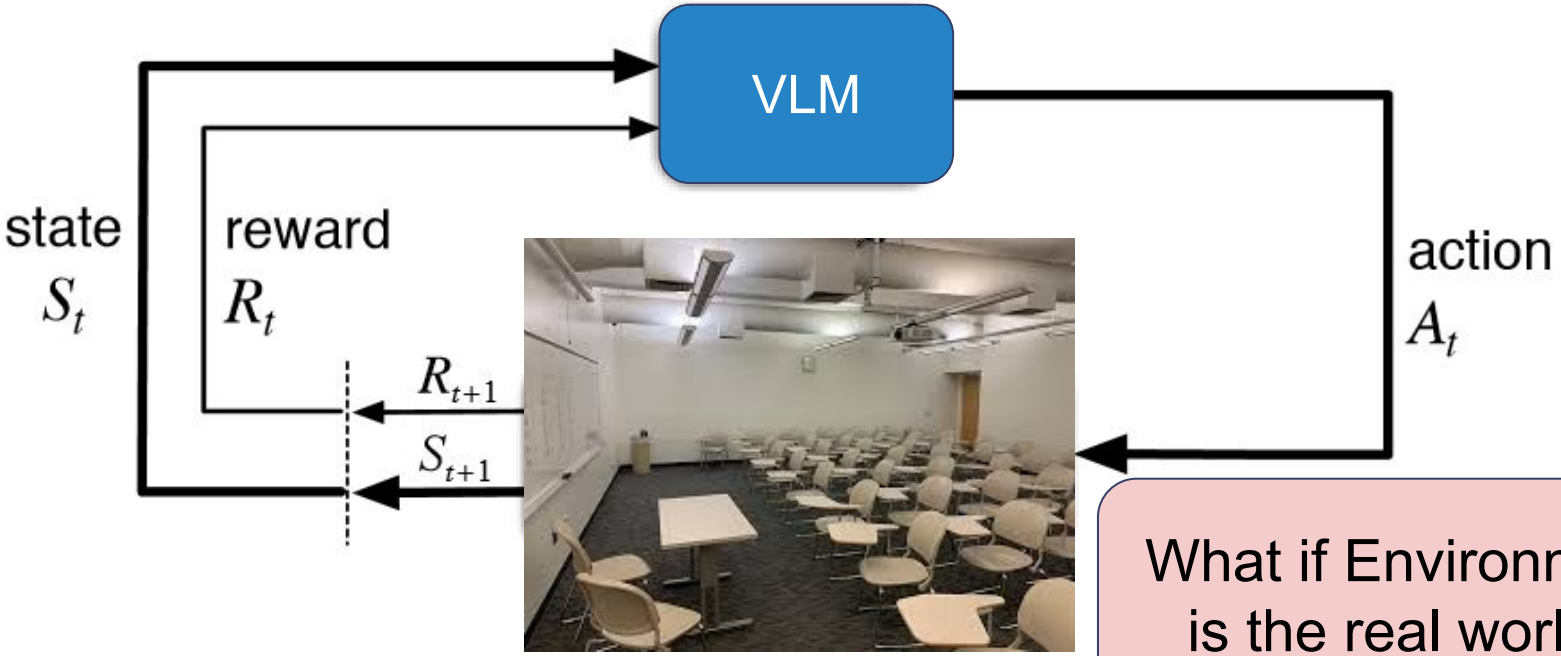


Environment is Assistant Tasks

What does VLM agent actually look like?

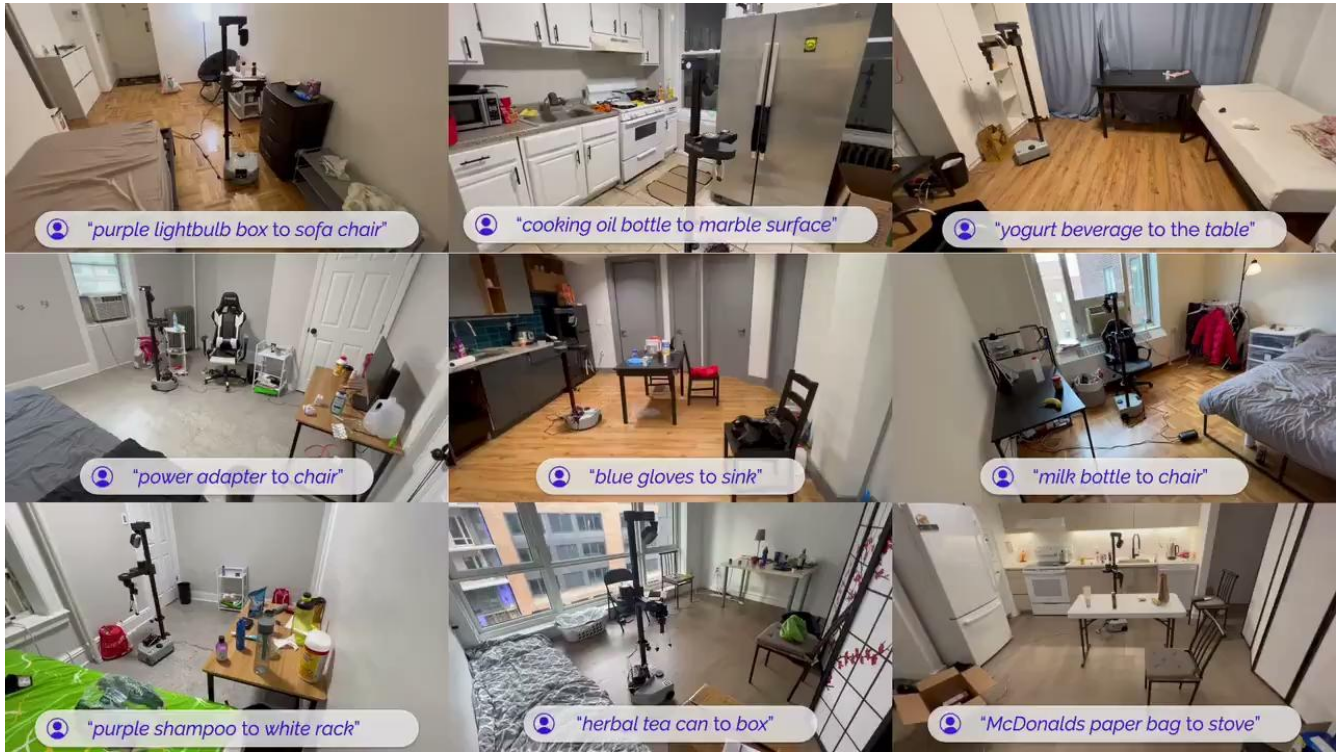


What does VLM agent actually look like?



What if Environment is the real world?

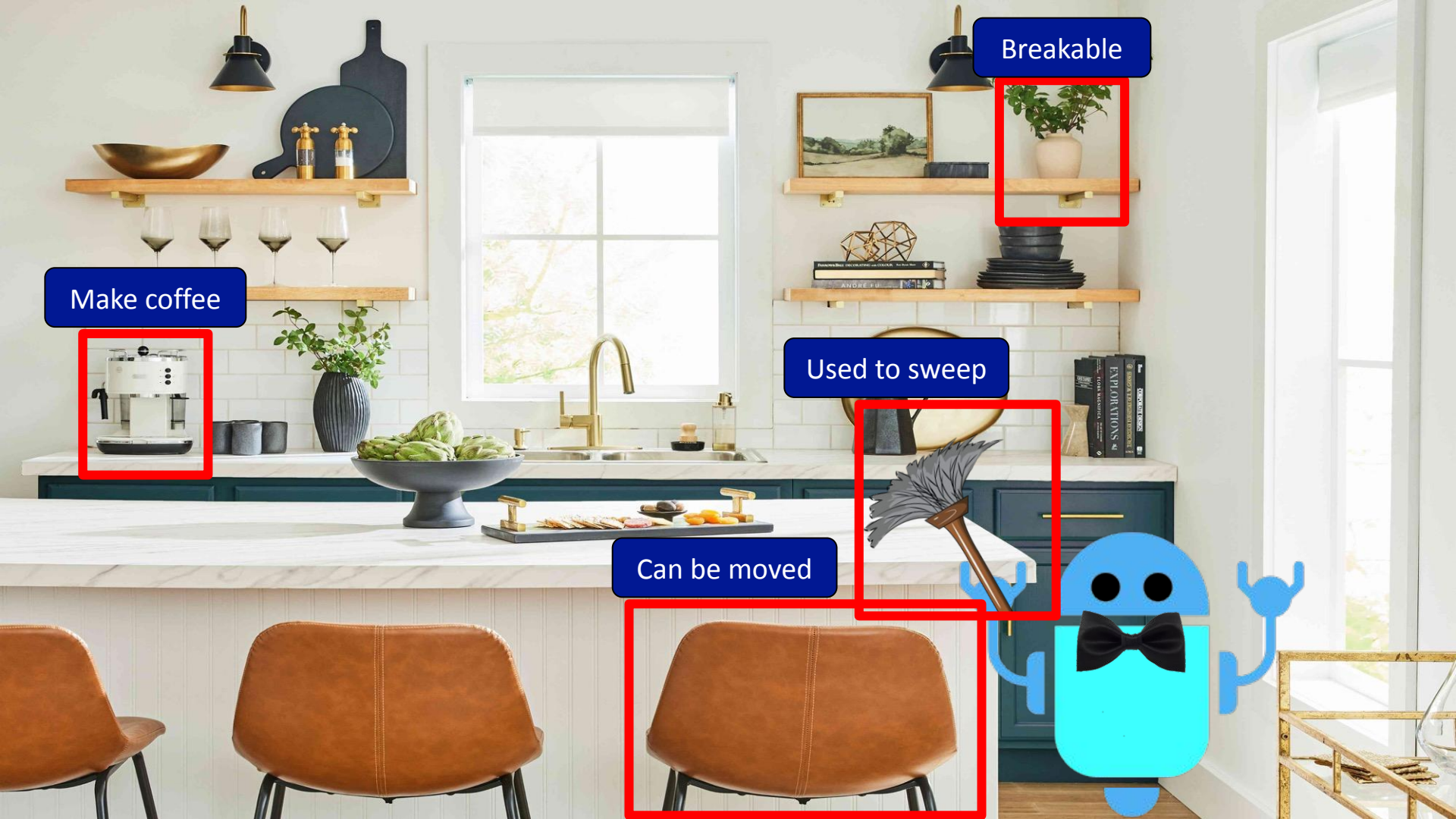
Robotics

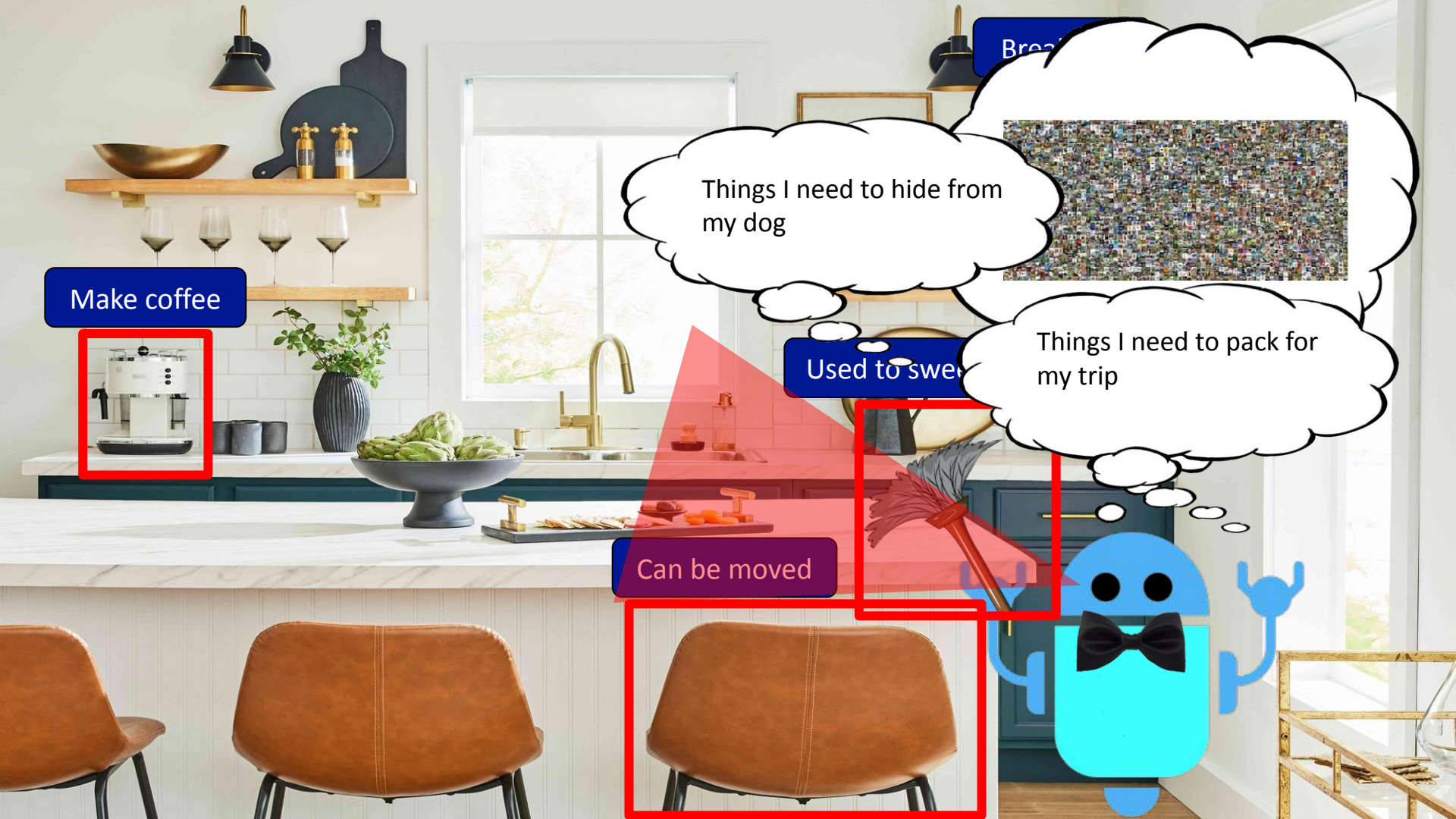

"purple lightbulb box to sofa chair"
"cooking oil bottle to marble surface"
"yogurt beverage to the table"
"power adapter to chair"
"blue gloves to sink"
"milk bottle to chair"
"purple shampoo to white rack"
"herbal tea can to box"
"McDonalds paper bag to stove"

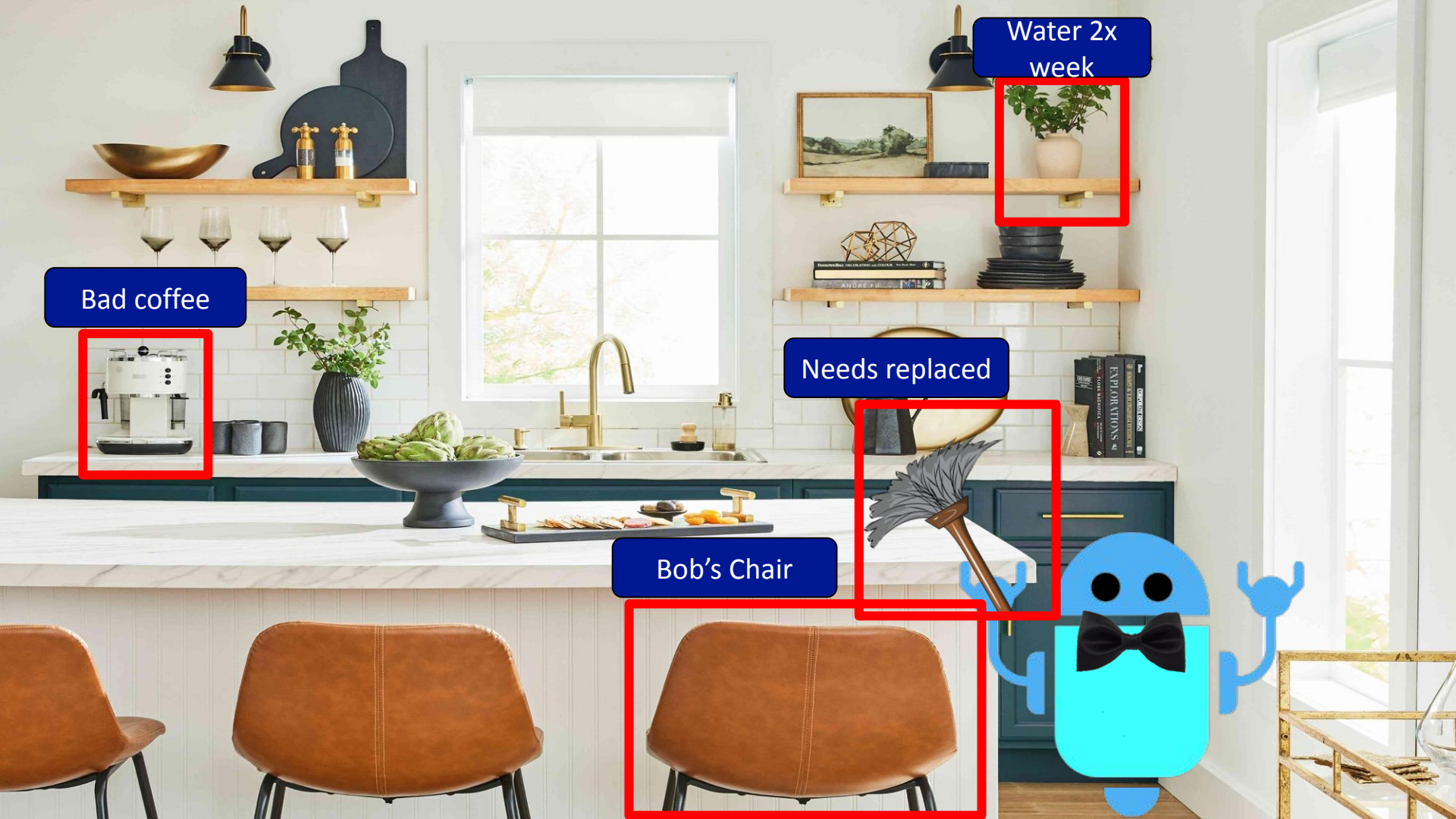
Robotics as a natural end-point for agents

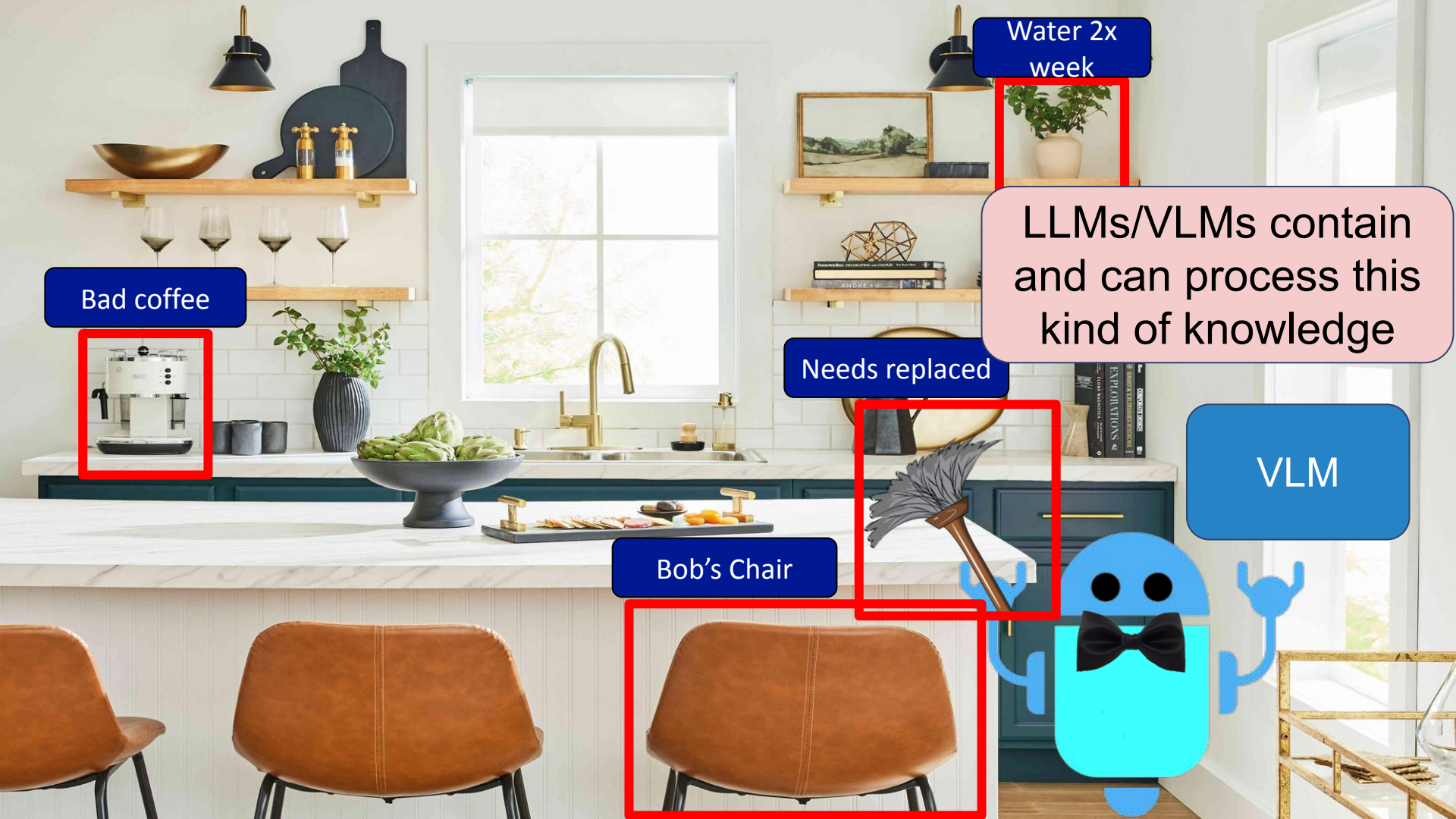
- Robotics is an open-ended environment
  - Have to do many different things
  - Interact with any kind of object
  - **Interact with people**
- Why might LLMs/Agents be useful
  - World Knolwedge!

Generally Capable Robots Require lots of Knowledge and Skills


Breakable
Make coffee
Used to sweep
Can be moved


Make coffee
Used to swe
Can be moved
Things I need to hide from my dog
Things I need to pack for my trip


Water 2x week
Bad coffee
Needs replaced
Bob's Chair


Water 2x week
LLMs/VLMs contain and can process this kind of knowledge
Bad coffee
Needs replaced
VLM
Bob's Chair

How to use agents for robotics

- Input space
  - Robot position
  - Sensors: proprioception, vision, maybe touch or other sensors

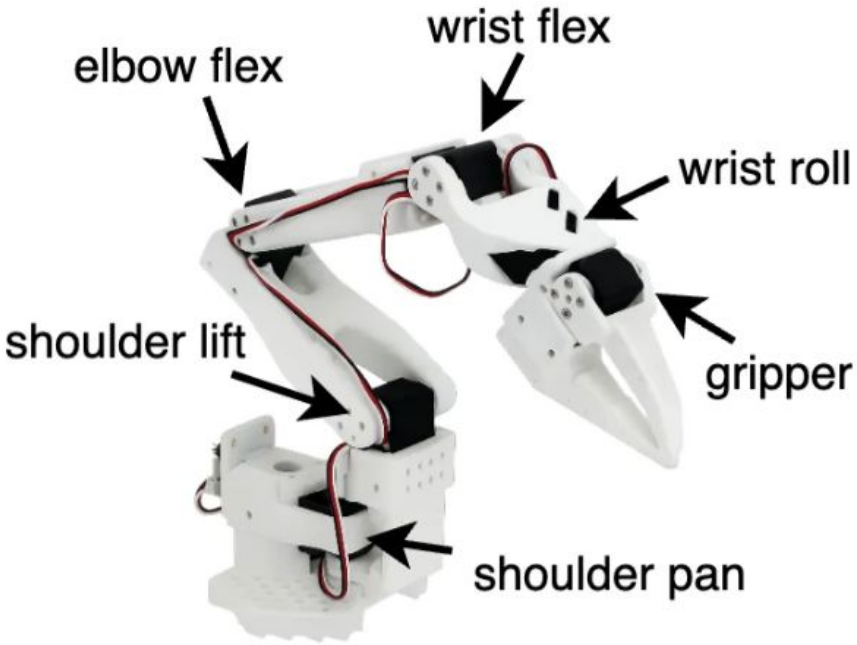
Robot Vision/Sensors

- Usually at least one overhead camera
- Sometimes cameras on each arm
- VLM can be very useful for real perception (need some kind of real vision system)

How to use agents for robotics

- Input space
  - Robot position
  - Sensors: proprioception, vision, maybe touch or other sensors
- Biggest issue action space
  - How do we get the agent to take robot actions?

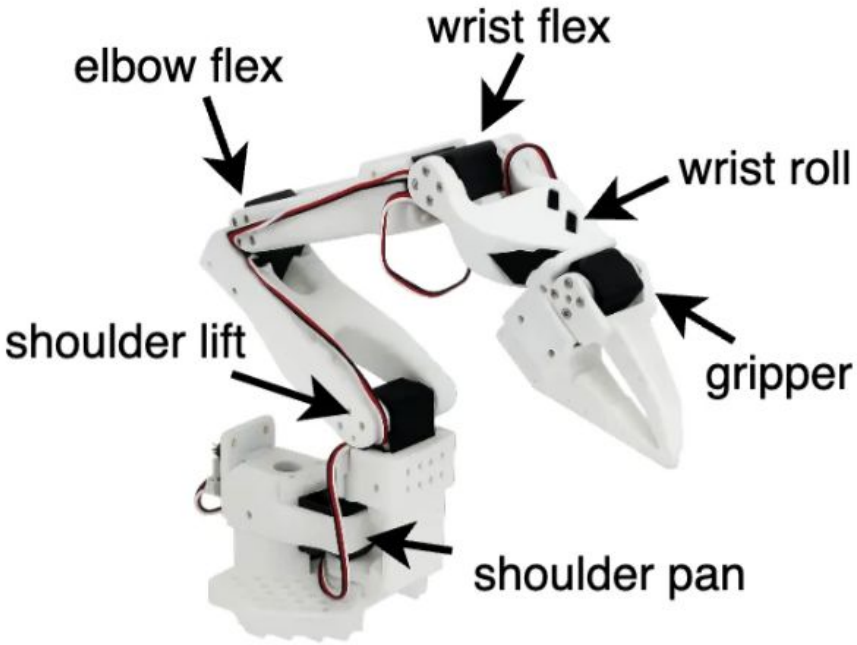
Robot Action Space



- E.g. 6 degrees of freedom arms
- For each actuator, can control either absolute position (θ) or torque (τ) to control

https://huggingface.co/spaces/lerobot/robot-learning-tutorial

Robot Action Space



- E.g. 6 degrees of freedom arms
- For each actuator, can control either absolute position (θ) or torque (τ) to control

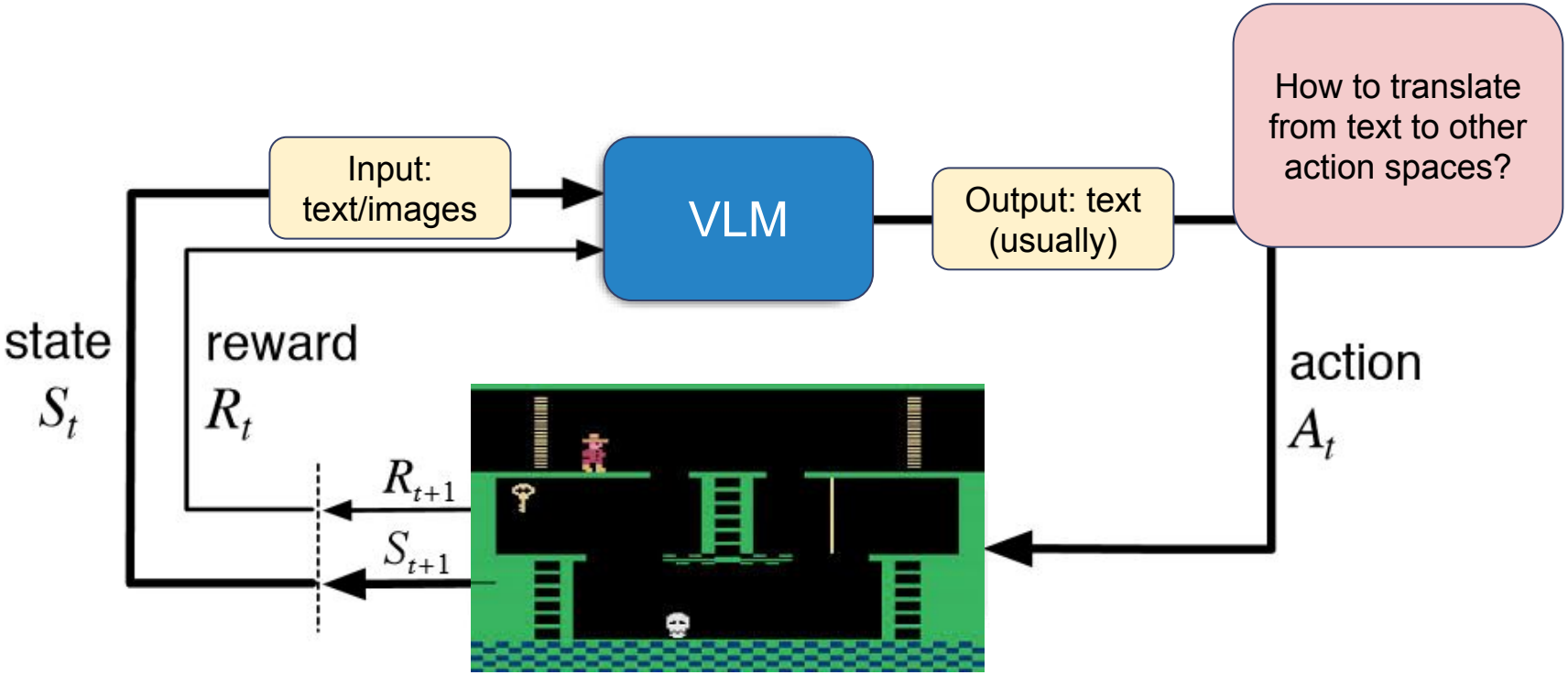
Problem: these are continuous values!

https://huggingface.co/spaces/lerobot/robot-learning-tutorial

How to use agents for robotics

- Input space
  - Robot position
  - Sensors: proprioception, vision, maybe touch or other sensors
- Biggest issue action space
  - How do we get the agent to take robot actions?
- A couple of options
  - Use tools/code as interface, let some other module handle actions
  - Output actions as strings
  - Train an action/specific module for actions

Recall: VLM to action space?



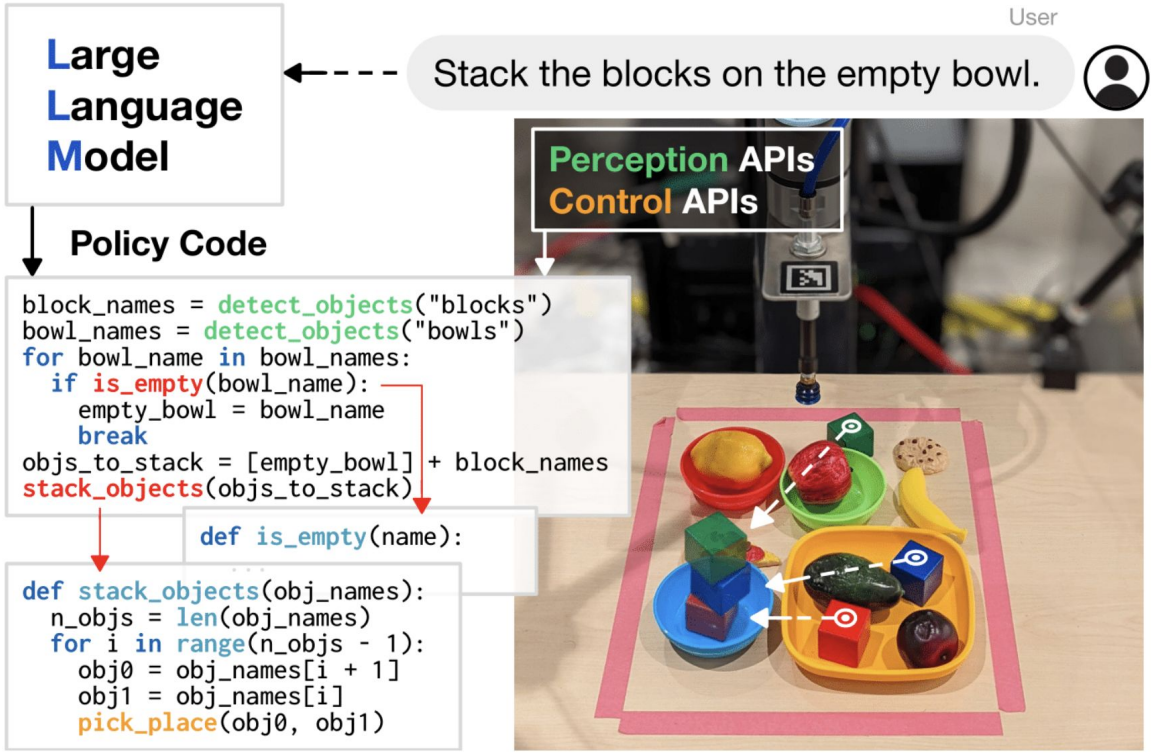
Code as Policies

Code as Policies: Language Model Programs for Embodied Control

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, Andy Zeng

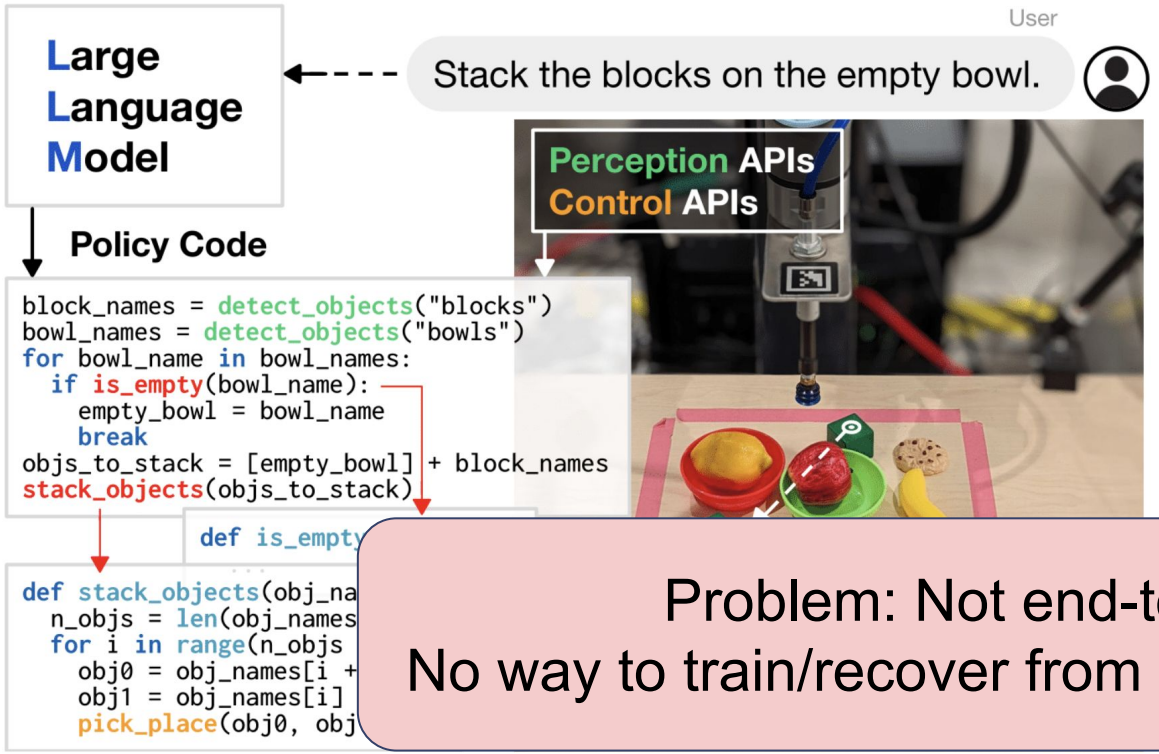
Broad idea: use code as interface to robot actions as “tools”

Code as Policies



- Task is decomposed into program
- Have Perception/Control functions pre-defined

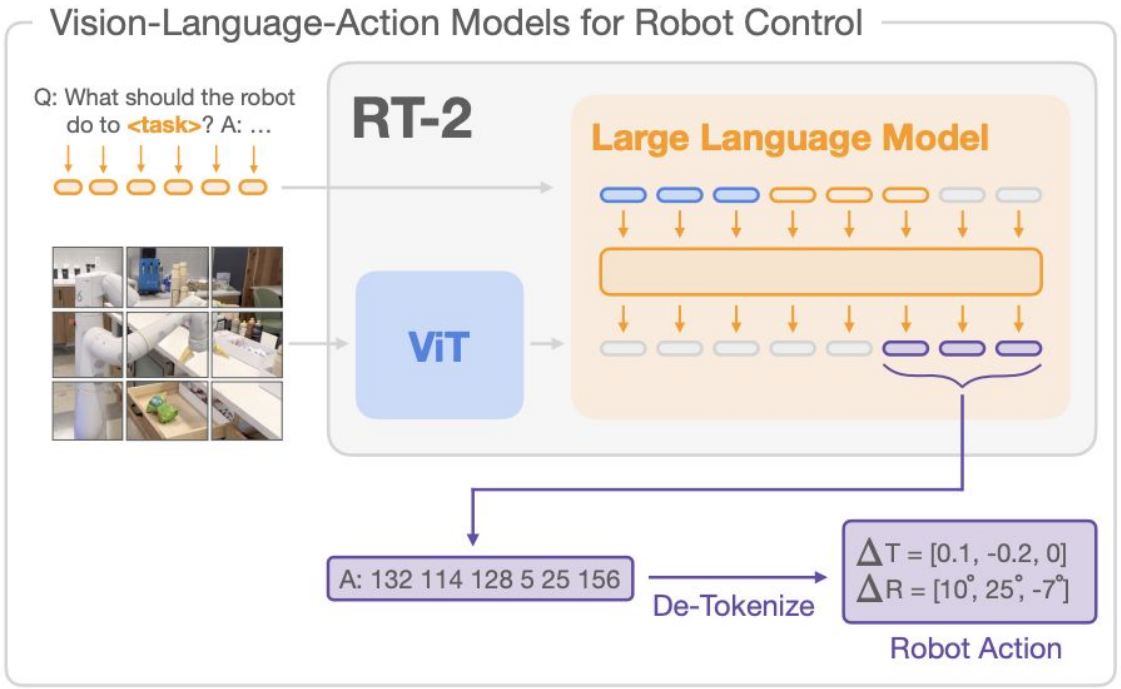
Code as Policies



- Task is decomposed into program
- Have Perception/Control functions pre-defined

Problem: Not end-to-end
No way to train/recover from low-level failure

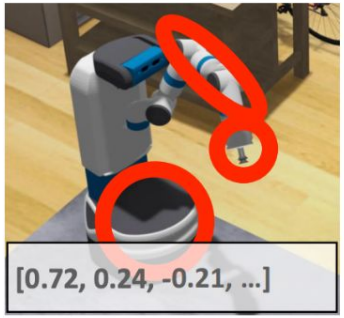
Just use language output?



- Literally just output numbers as output (like the strings)
- Usually rescalled to (e.g. 0 - 255)
- Still requires finetuning to get at all correct

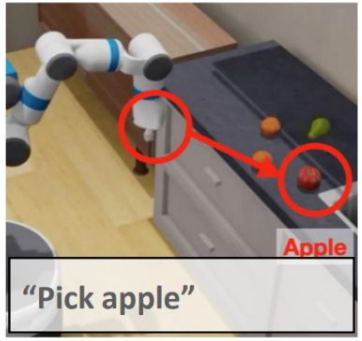
RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control

Action Tokenization



Action is a continuous vector

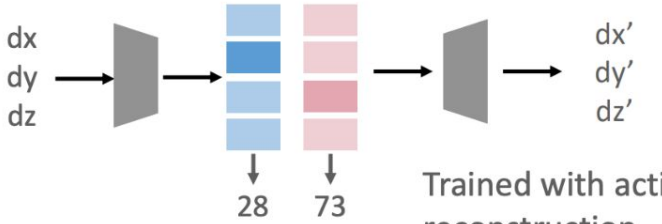
Example: end effector control
[dx, dy, dz]



Action is a selection from a set of discrete choices

Learned Tokenization

Residual VQ-VAE for discrete action tokenization



Trained with action reconstruction

Semantic Tokenization

“pick apple”

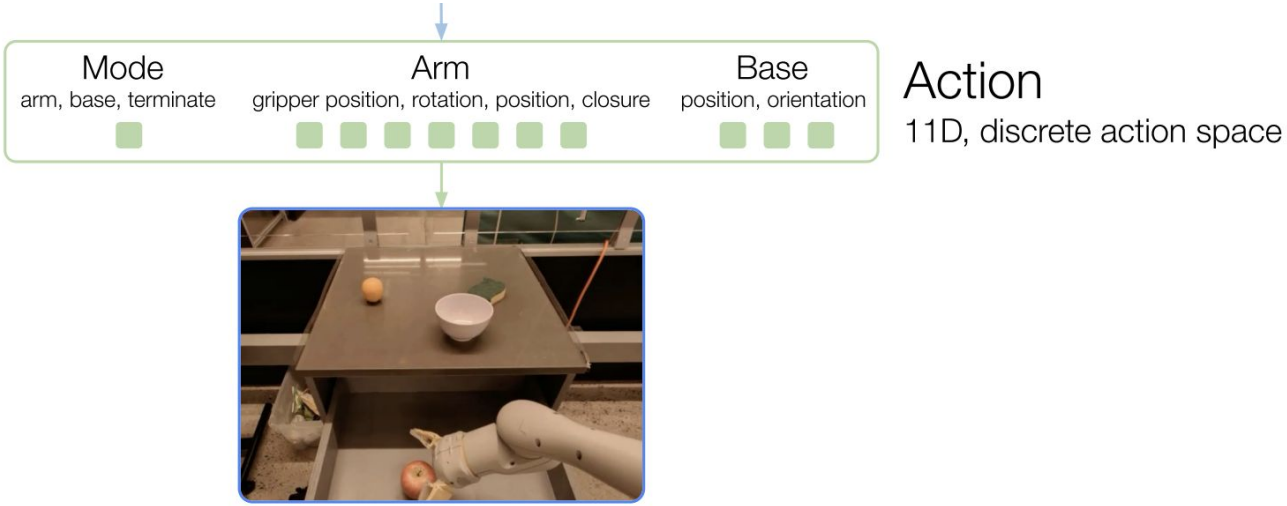
↓

[278,276]

Describe action with language and tokenize with LLM vocabulary

Szot et al., Grounding Multimodal Large Language Models in Actions. NeurIPS 2024.

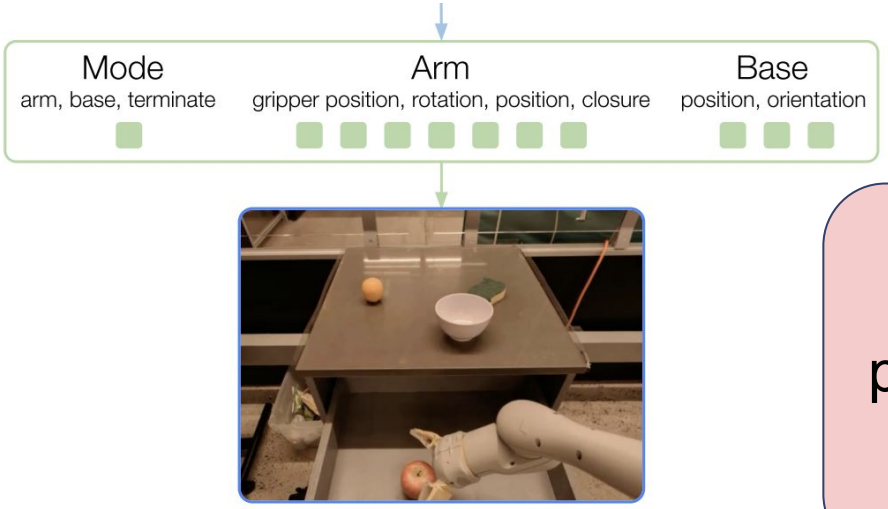
New Action Head



- Conceptually very easy, just directly output actions
- Requires training a new action head

RT-1: Robotics Transformer for Real-World Control at Scale

New Action Head



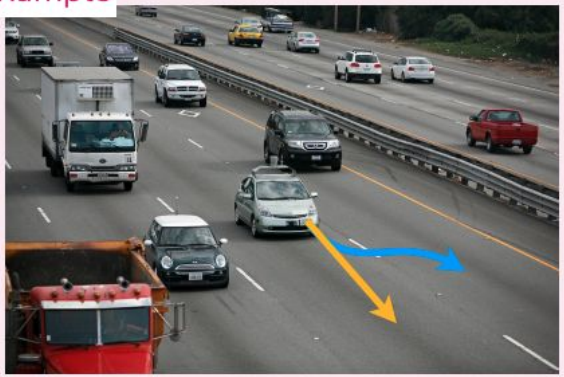
Does anyone recall the problem we discussed way back in RL lecture?

- Conceptually very easy, just directly output actions
- Requires training a new action head

RT-1: Robotics Transformer for Real-World Control at Scale

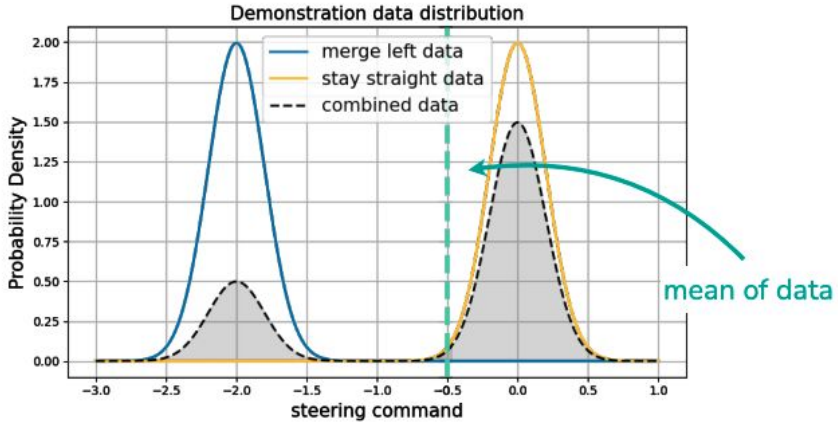
Recall: Multimodal Problem

Example



- Dataset from human drivers
- Sensor readings + steering commands

Question: what might policy trained with ℓ_2-regression do?



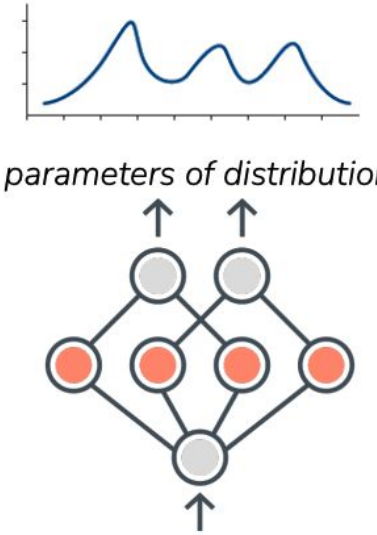
How can we represent more than the mean?

How often does this happen in practice? All the time!

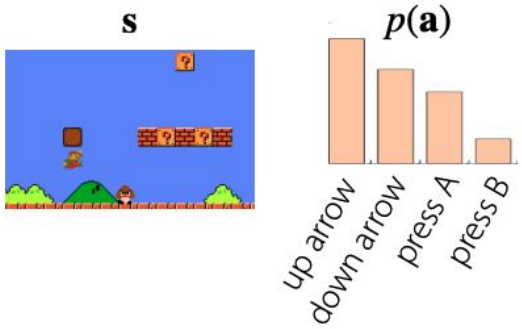
Esp. when data collected by multiple people.

Slide credit: Chelsea Finn

Recall: Learning Distributions with NNs



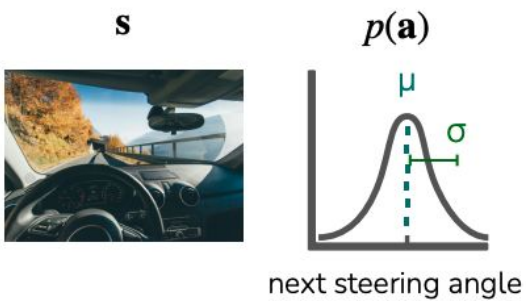
1D discrete actions



Neural net outputs $p(\text{up}), p(\text{down}), \ldots$ represent categorical distribution.

Maximally expressive

Continuous actions



Neural net outputs μ, σ to represent Gaussian distribution.

Not very expressive!

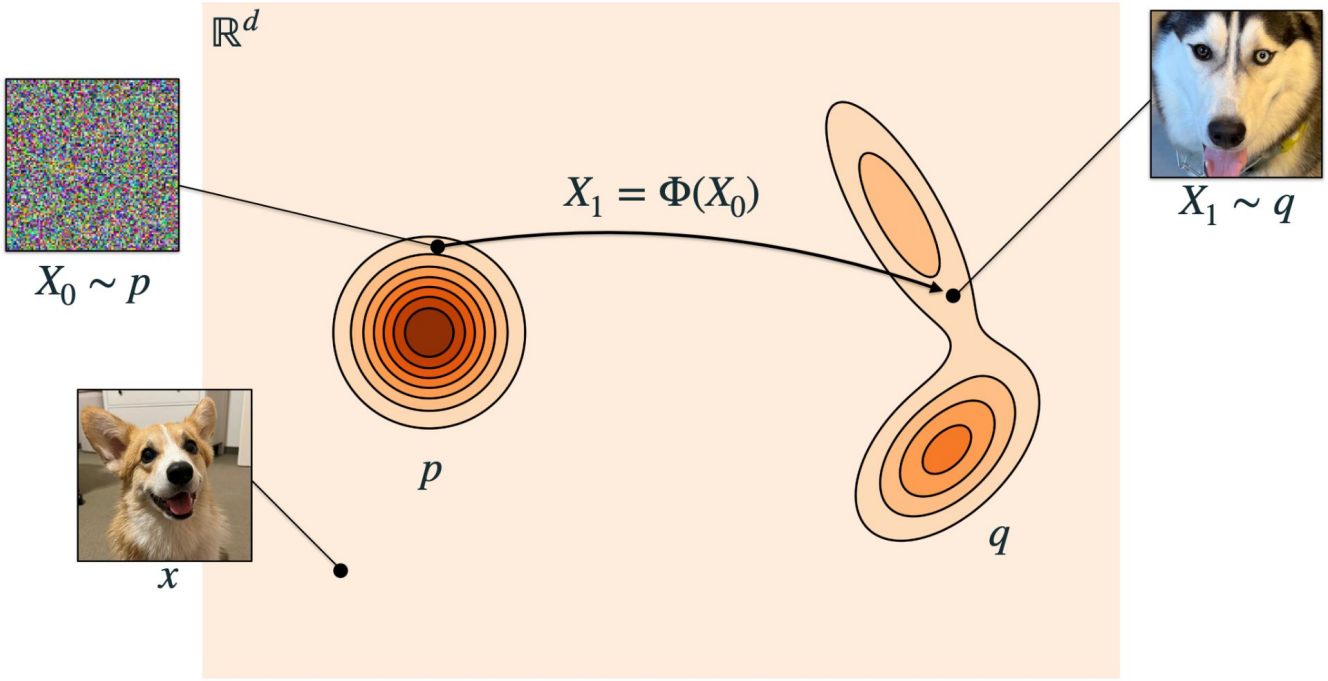
Slide credit: Chelsea Finn

Solution: Diffusion/Flow Matching!

- Mathematically quite involved
  - Neurips 2024 tutorial on this was really great for explaining this!
  - https://neurips.cc/media/neurips-2024/Slides/99531.pdf

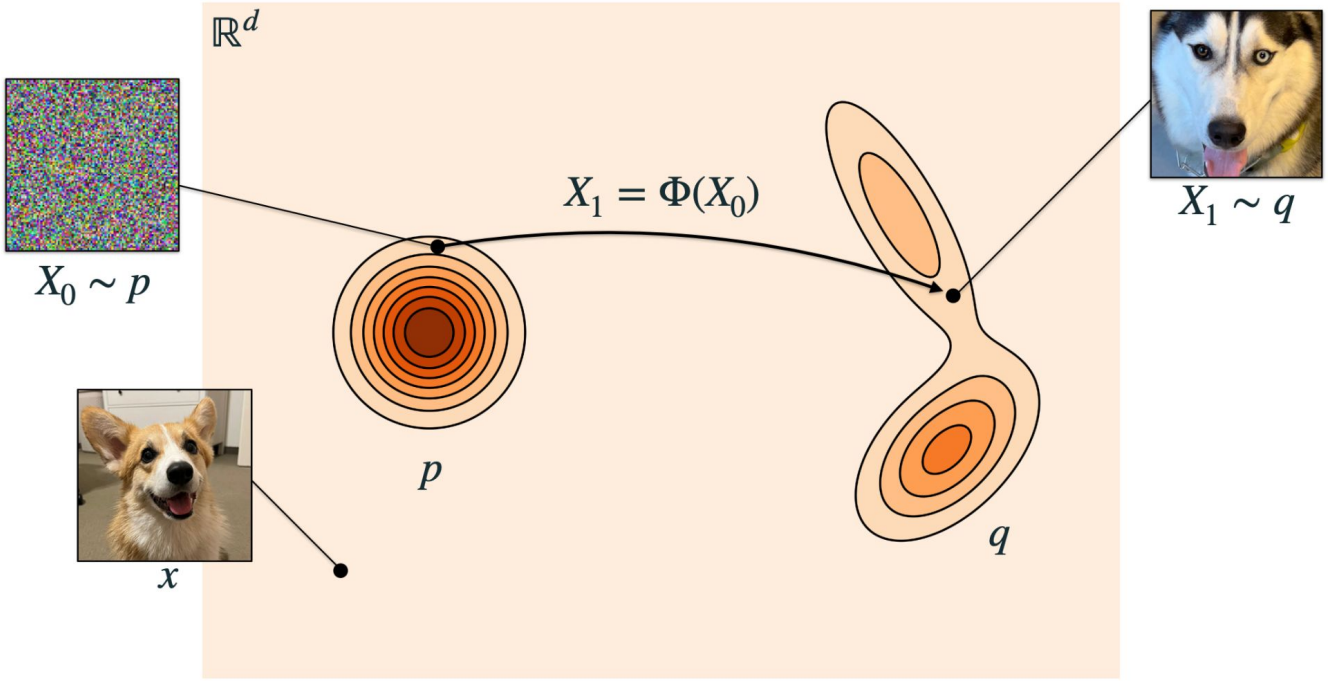
https://neurips.cc/media/neurips-2024/Slides/99531.pdf

Diffusion In a Nutshell



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

Diffusion In a Nutshell



How do we go from random noise to an image (or a robot torque)

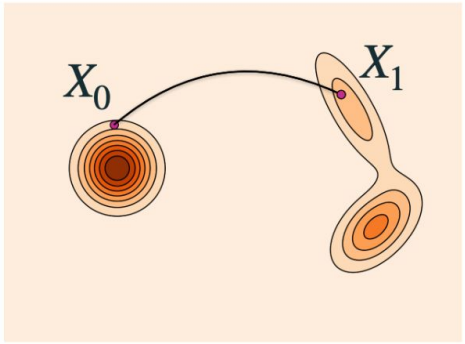
https://neurips.cc/media/neurips-2024/Slides/99531.pdf

Diffusion In a Nutshell

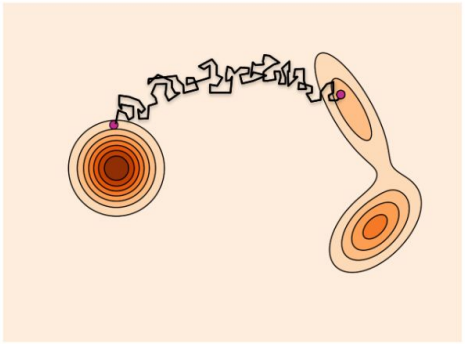
- Continuous-time Markov process $(X_t)_{0 \le t \le 1}$

$$X_{t+h} \leftarrow \Phi_{t+h|t}(X_t)$$

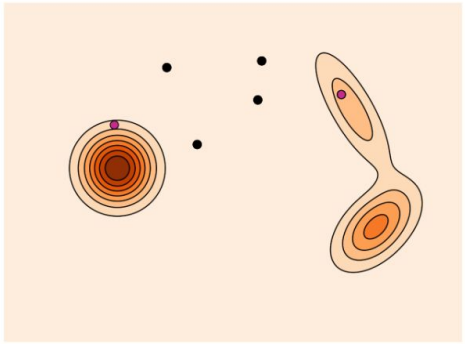
Answer: step-by-step



Flow



Diffusion



Jump

Diffusion/Flow Matching - Predicts iteratively

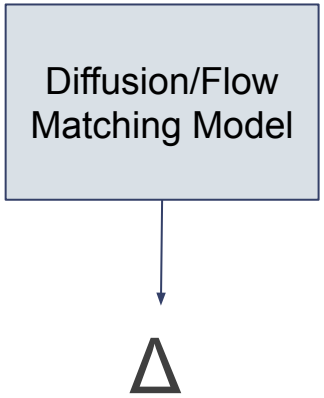
X_0

Start at some random place
in the space

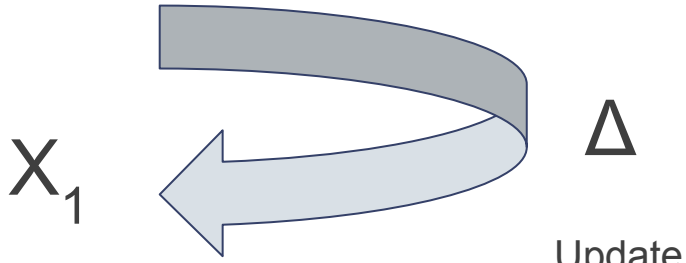
Diffusion/Flow Matching - Predicts iteratively

X_0

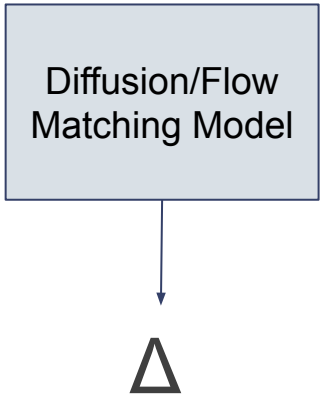
Start at some random place in the space



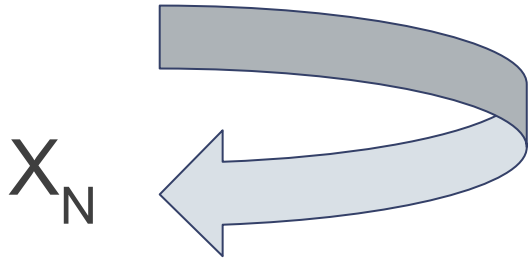
Diffusion/Flow Matching - Predicts iteratively



Start at some random place in the space



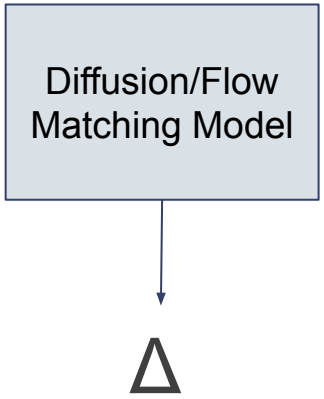
Diffusion/Flow Matching - Predicts iteratively



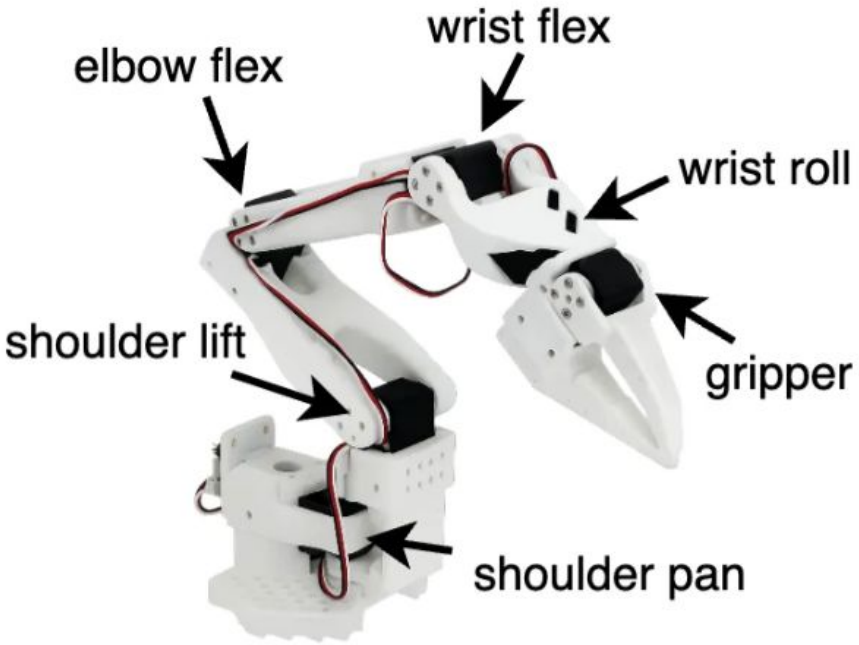
Start at some random place in the space

Δ

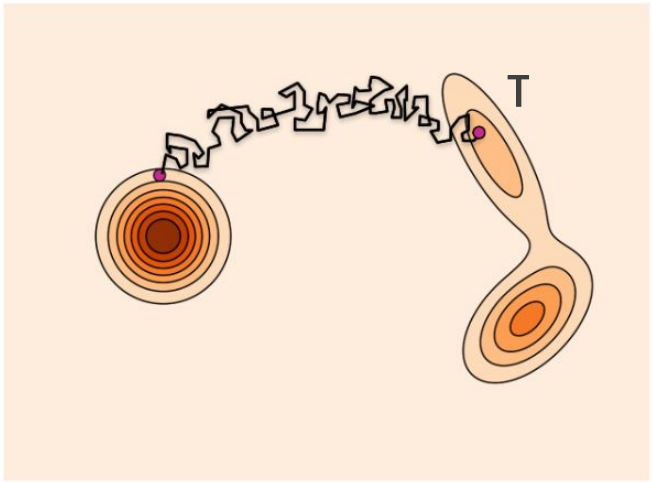
Continue until convergence



What this means for robotics

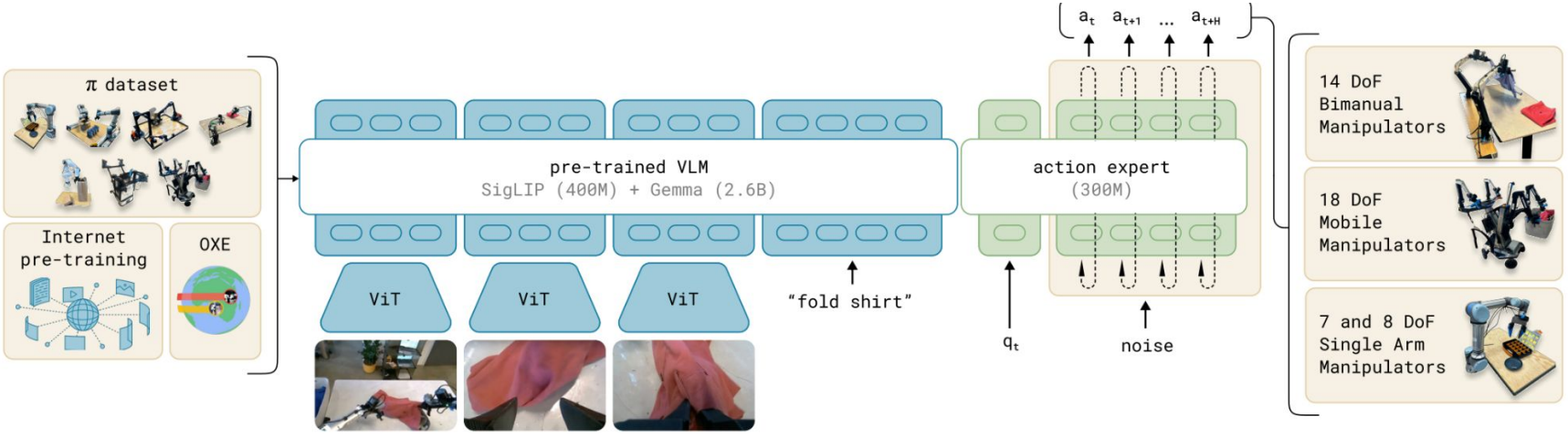


- Do multiple predictions to get final torque (τ) action



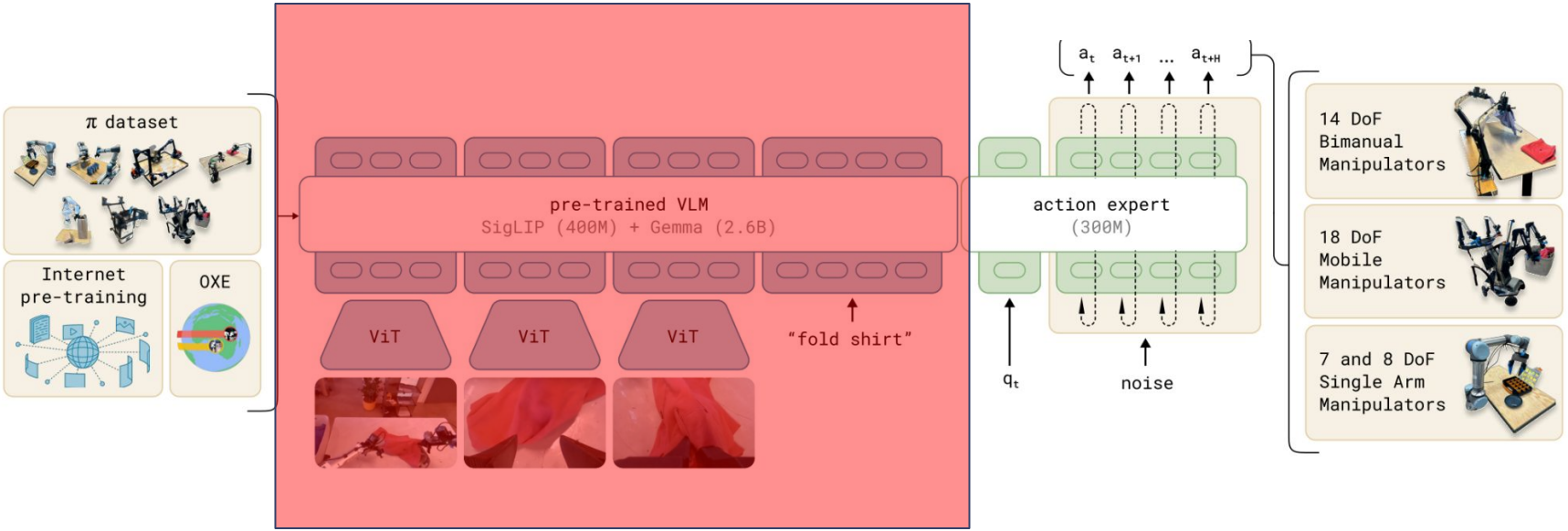
https://huggingface.co/spaces/lerobot/robot-learning-tutorial

Create VLAs by combining VLM w/ action head



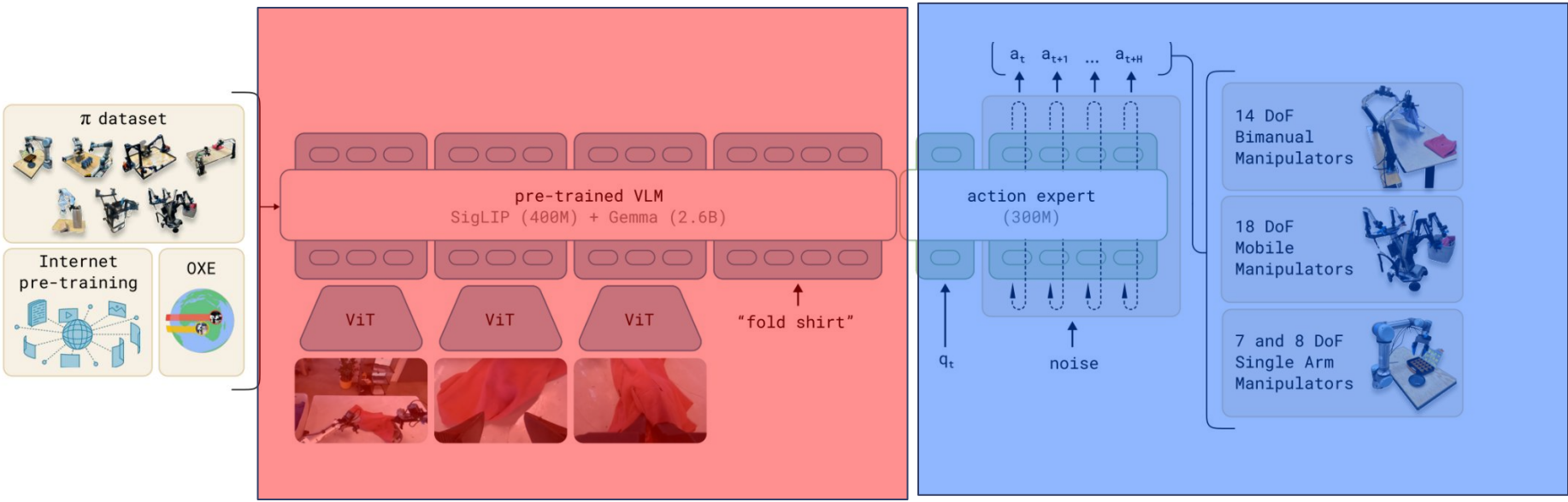
Create VLAs by combining VLM w/ action head

Start with some VLM, sometimes frozen



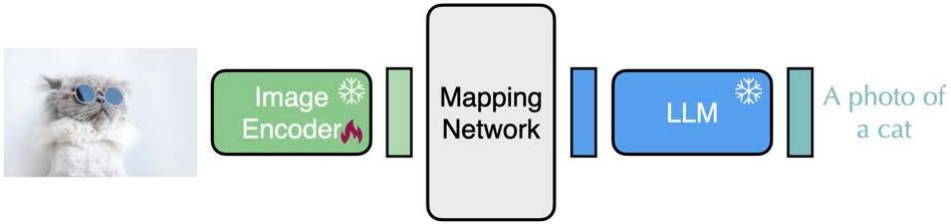
Create VLAs by combining VLM w/ action head

Start with some VLM, sometimes frozen



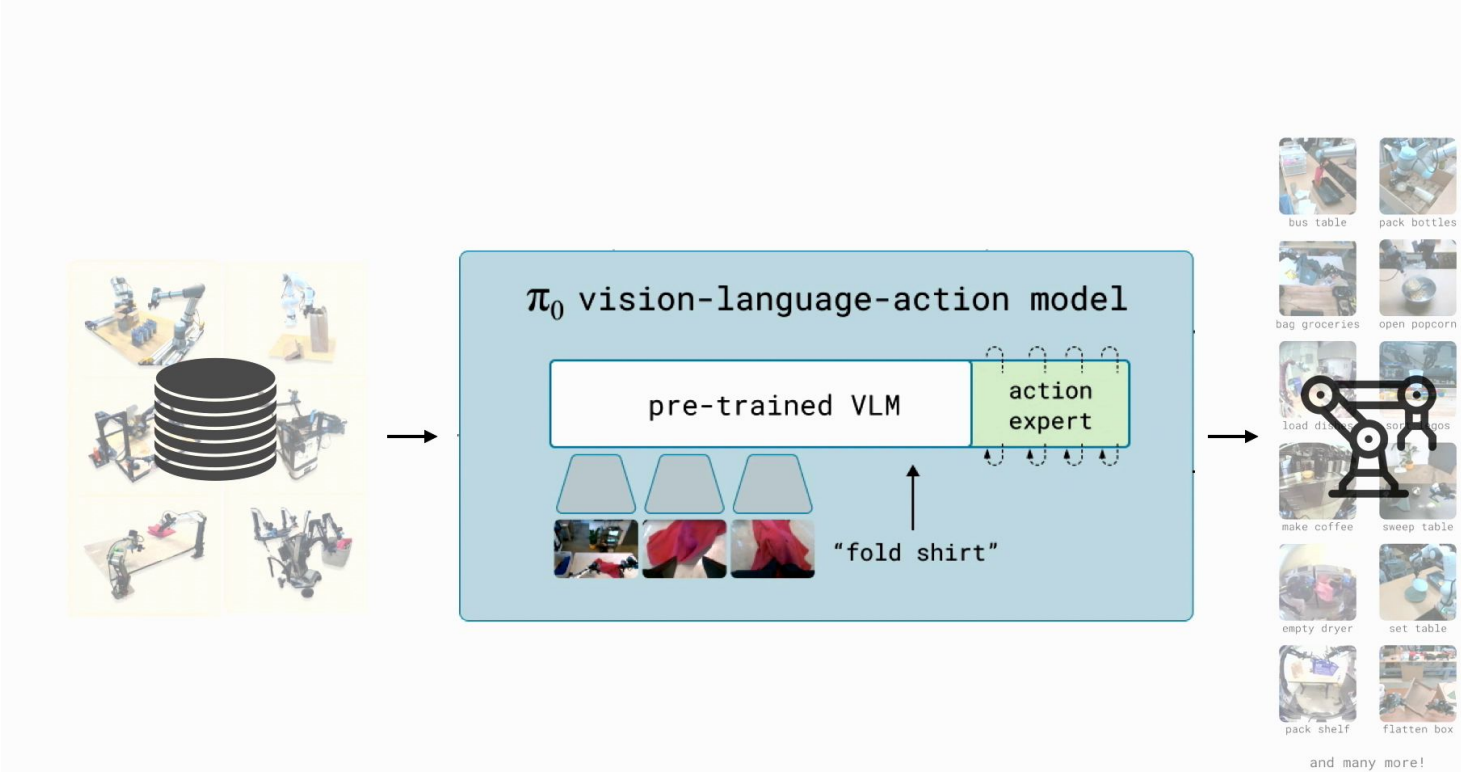
Add an action head (e.g. diffusion model) trained on robotics data

Recall: Starting from frozen backbones not new



VLMs from Pretrained Backbones

VLAs: One model to rule them all?



Slide: Chelsea Finn

VLAs: One model to rule them all?

Summary

- Robots the final boss of AI
  - Requires action and perception
  - Requires operating in the real wold
- LLM/VLM can give lots of knowledge about world/tasks
- Big question: action
  - Need to operate in much more complex/continuous action space
- Solutions
  - Output with tolls/code
  - Output with strings
  - Train new output head
  - Diffusion models
- Big picture: toward large foundation models for robotics?

Any Questions

Questions

Now for the presentations!