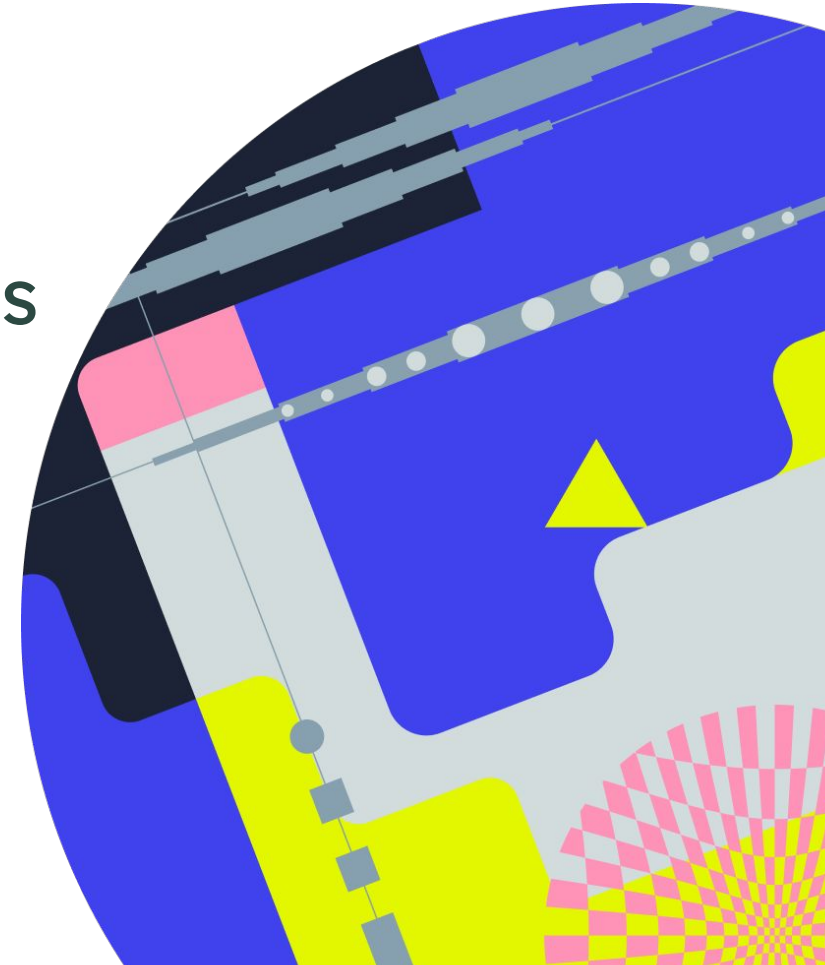
Школа анализа данных

Seminar 4: Action modality, LLM/VLM agents



Supplementary slides to Google Colab notebook

Andrey Spiridonov, Zinkovich Viktoriia

Recap of previous seminar

1. Deep Fusion

deeply fuses multimodal inputs within internal layers

1.1. **Standard** Cross-Attention (SC-DF)

OpenFlamingo

- perceiver resampler
- cross-attention
- tanh gating

1.2. **Custom** Layers (CL-DF)

MoE-LLaVA

- vision encoder MLP
- MoE layer
- Router

2. Early Fusion

multimodal inputs are fed to the model rather to its internals

2.1. **Non-tokenized** (NT-EF)

Qwen-VL

- 3 stage pre-training
- **encoder** for vision modality

Qwen-VL

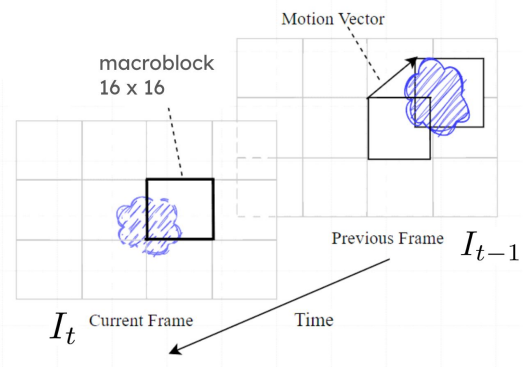
2.2. **Tokenized** (T-EF)

LaVIT

- **tokenizer** for visual modality
- token selector & merger

Recap of previous seminar

Video-LaVIT (Feb 2024) – employ the **MPEG-4** (1991) to divide the image to **keyframes** (primary semantics) and **motion** (temporal evolvement)



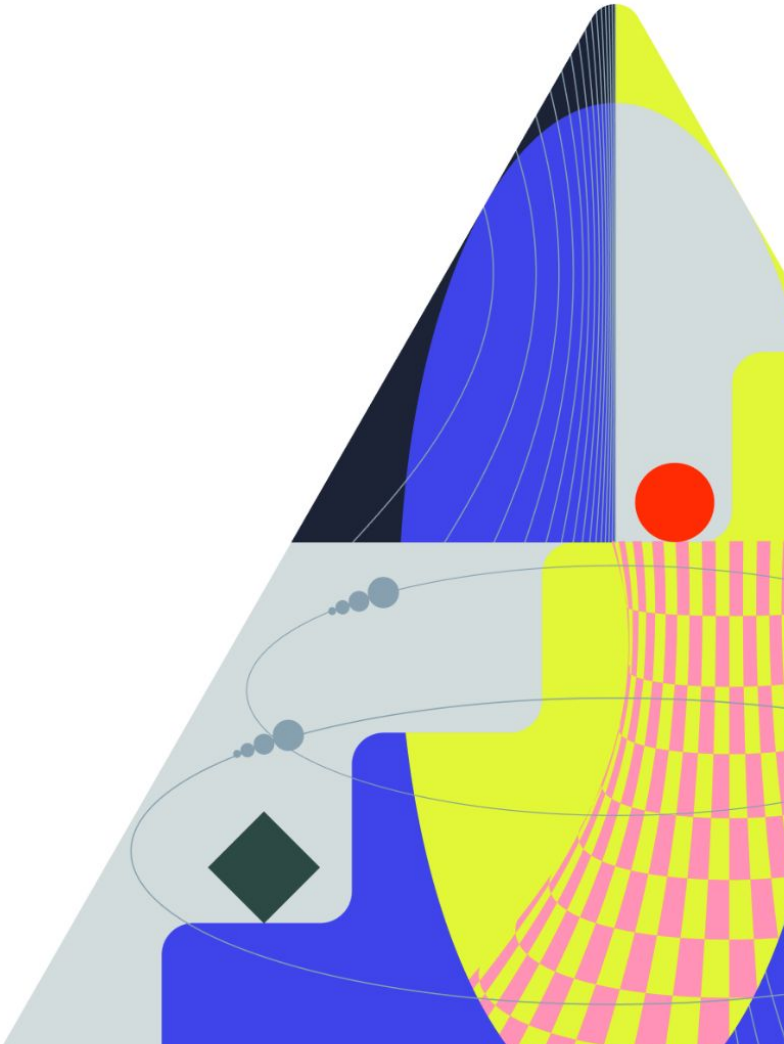
Text2Video

Image2Video

Part 1

Octo model

Open-Source Generalist
Robot Policy



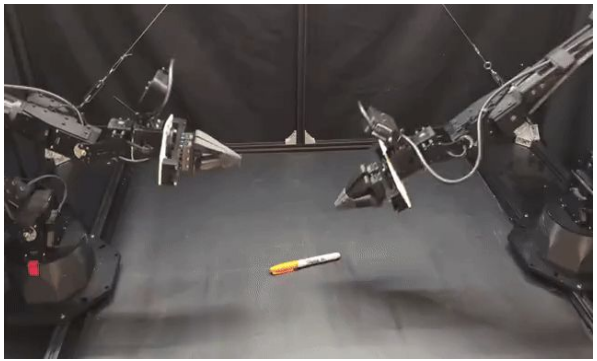
Octo: Introduction (May 2024)

Octo trains policies on robot datasets assembled across ***multiple*** **embodiments** and trained on a larger and **more diverse** robot data mix

[1] Dibya Ghosh et al. **Octo:** An Open-Source Generalist Robot Policy. **RSS, Oct 2024**. [link]

Octo: Introduction (May 2024)

Octo trains policies on robot datasets assembled across ***multiple*** **embodiments** and trained on a larger and **more diverse** robot data mix

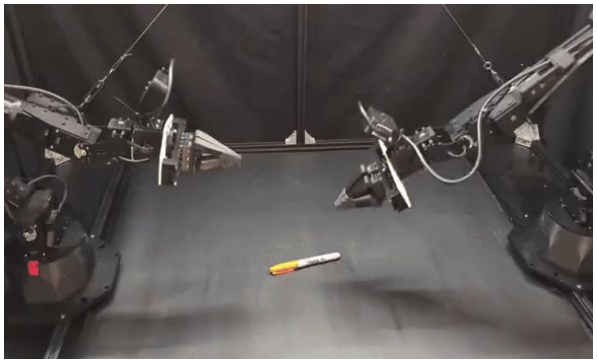


dual-arm robot, Stanford
Aloha

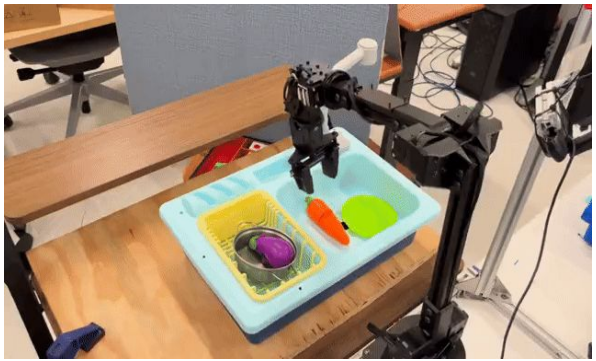
[1] Dibya Ghosh et al. **Octo:** An Open-Source Generalist Robot Policy. **RSS, Oct 2024**. [link]

Octo: Introduction (May 2024)

Octo trains policies on robot datasets assembled across ***multiple*** **embodiments** and trained on a larger and **more diverse** robot data mix



dual-arm robot, Stanford
Aloha

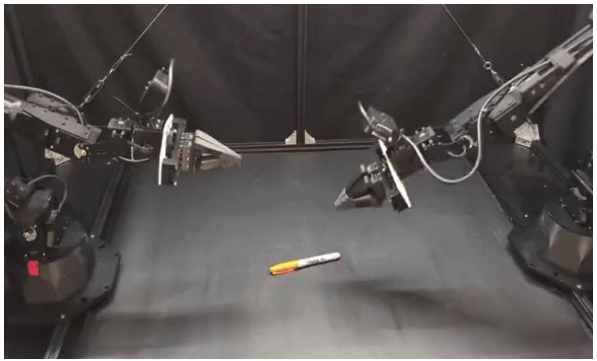


part of the Bridge Dataset
WidowX BridgeV2

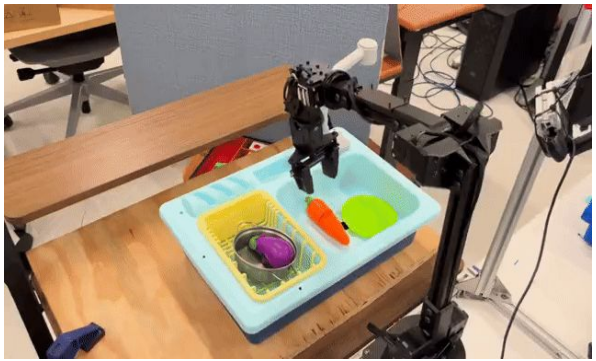
[1] Dibya Ghosh et al. **Octo:** An Open-Source Generalist Robot Policy. **RSS, Oct 2024**. [link]

Octo: Introduction (May 2024)

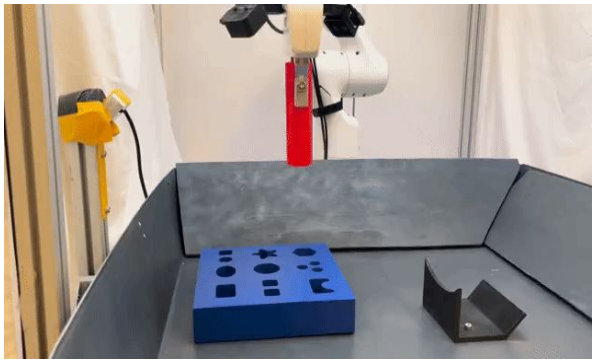
Octo trains policies on robot datasets assembled across ***multiple*** **embodiments** and trained on a larger and **more diverse** robot data mix



dual-arm robot, Stanford
Aloha



part of the Bridge Dataset
WidowX BridgeV2

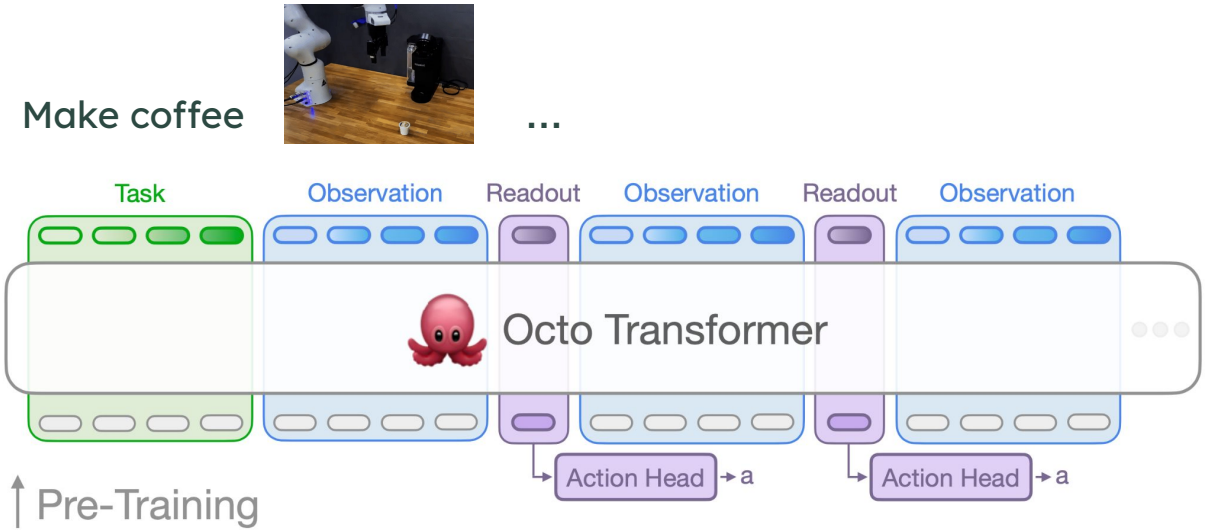


Functional manipulation benchmark
Berkley Peg insertion

[1] Dibya Ghosh et al. **Octo:** An Open-Source Generalist Robot Policy. **RSS, Oct 2024**. [link]

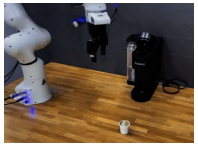
Octo pre-training: "Tokenizers"

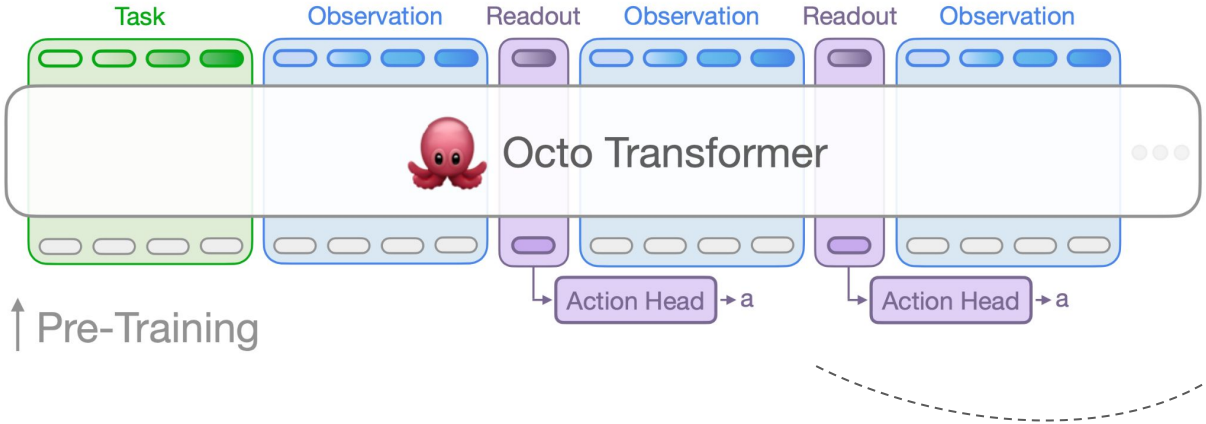
Core model is **transformer architecture** that maps arbitrary input tokens (created from observations and tasks) to output tokens (then decoded into actions)

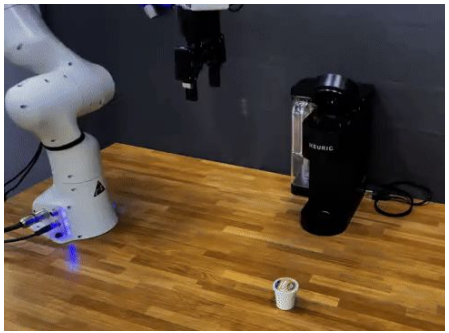


Octo pre-training: "Tokenizers"

Core model is **transformer architecture** that maps arbitrary input tokens (created from observations and tasks) to output tokens (then decoded into actions)

Make coffee  ...



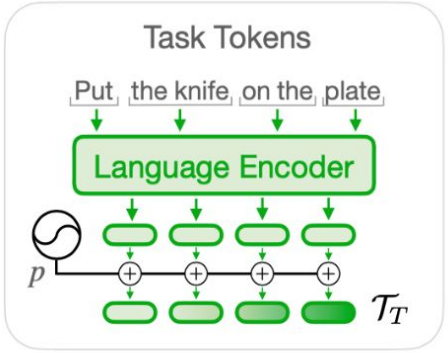


Octo pre-training: "Tokenizers"

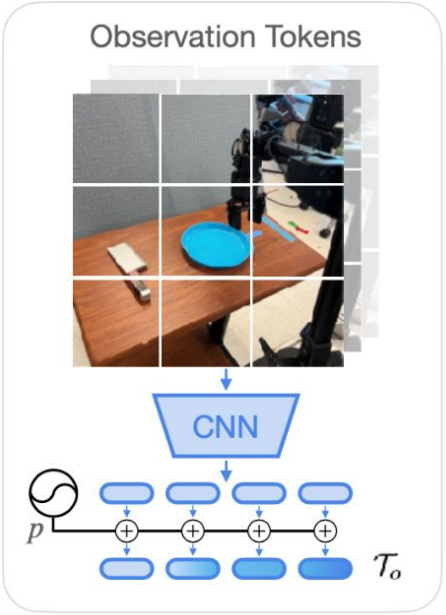
Obtain tokens through **modality-specific tokenizers** (actually "embeddings", not "tokens") and arrange all them sequentially $[\mathcal{T}_T, \mathcal{T}_{o,1}, \mathcal{T}_{o,2}, \dots]$

t5-base [111M]

passed through a pretrained transformer that produces a sequence of **language embedding tokens**



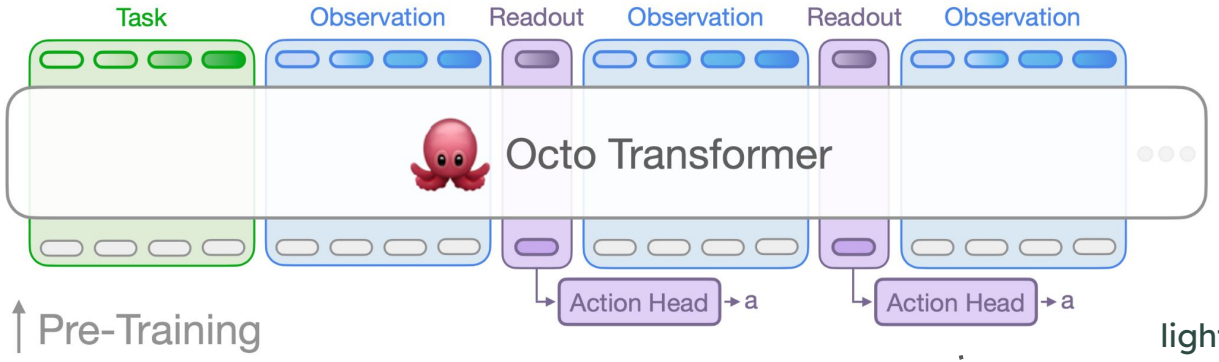
shallow CNN, **flattened patches**



Octo pre-training: Backbone

observation tokens can only attend causally to tokens **from the same or earlier time steps** as well as task tokens

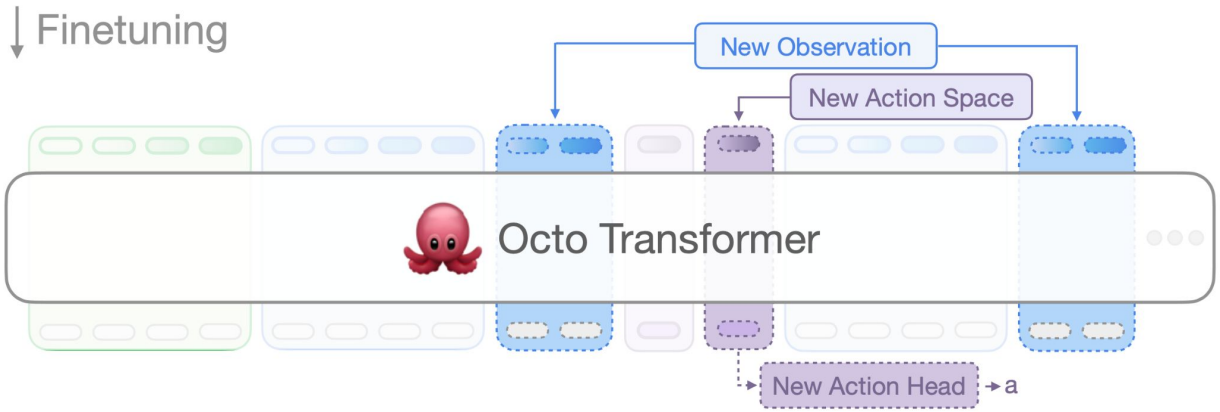
readout tokens is a compact vector embedding of the observation



lightweight **"action head"** that implements the **diffusion process** is applied to the embeddings for the readout tokens

Octo pre-training: Backbone

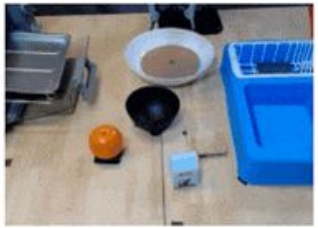
When adding new task, observations, loss functions, we can **fully retain the pretrained weights** of the transformer



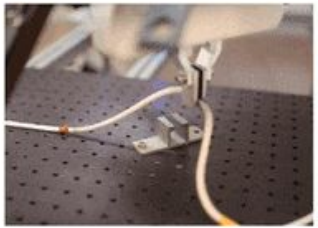
Only adding new encoders or parameters in new head → **generalist model**

Octo: Dataset

Octo was trained on **800k trajectories** from the **Open X-Embodiment dataset**
(from 1.5M robot episodes)



CLVR, USC



RAIL, UC Berkeley

CILVR, NYU

AUTOLab, UC Berkeley

AiS, University of Freiburg

25 datasets

several robot **embodiments**, scenes

sensors and labels

remove **repetitive** datasets

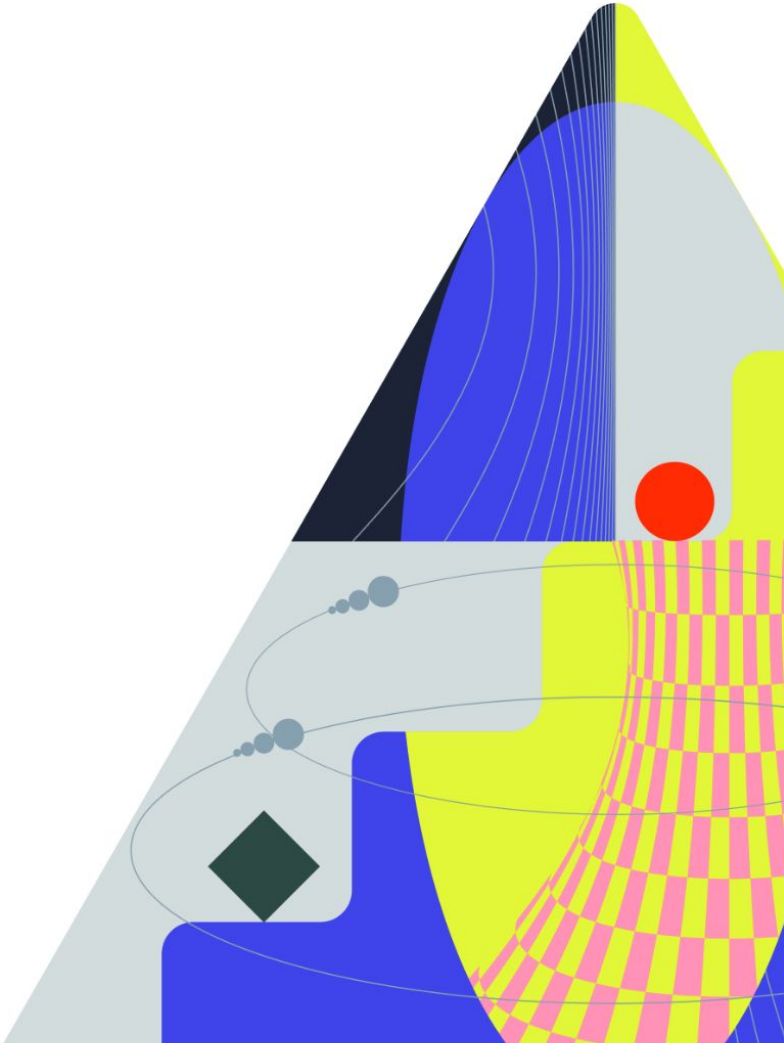
remove **low** image resolution

diverse dataset

Part 2

Agents

LLM & VLM guided agents



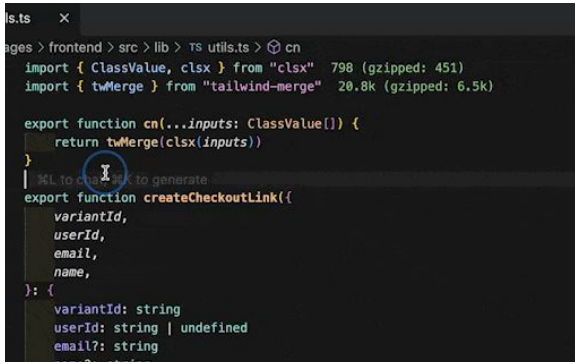
Who is the Agent?

A*gent* is a system capable of perceiving its **environment** and making **decisions** based on these perceptions to achieve specific **goals**

[1] Junlin Xie et al. **Large Multimodal Agents:** A Survey. **23 Feb 2024**. [link]

Who is the Agent?

A*gent* is a system capable of perceiving its **environment** and making **decisions** based on these perceptions to achieve specific **goals**

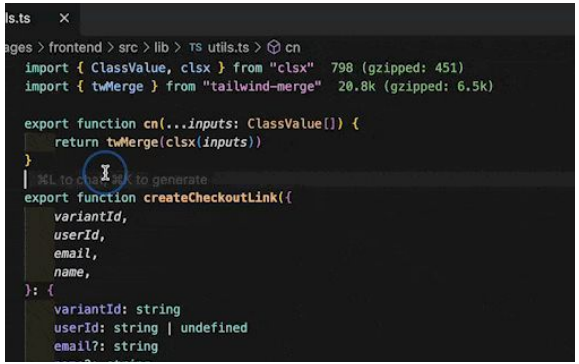


Cursor **Code Editor**

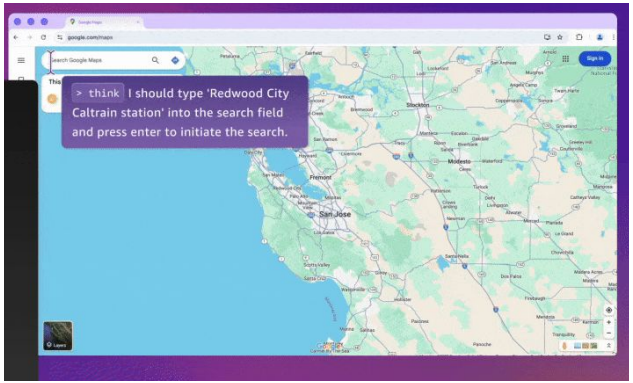
[1] Junlin Xie et al. **Large Multimodal Agents:** A Survey. **23 Feb 2024**. [link]

Who is the Agent?

A*gent* is a system capable of perceiving its **environment** and making **decisions** based on these perceptions to achieve specific **goals**



Cursor **Code Editor**

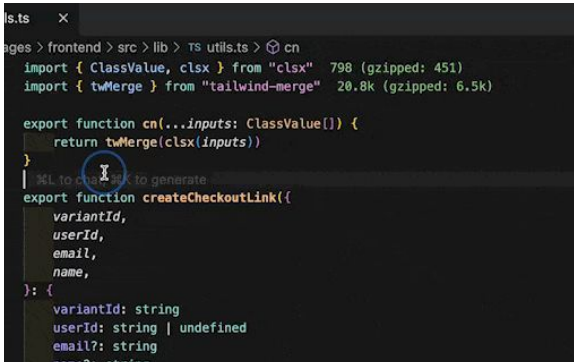


Amazon Nova **Web Agent**

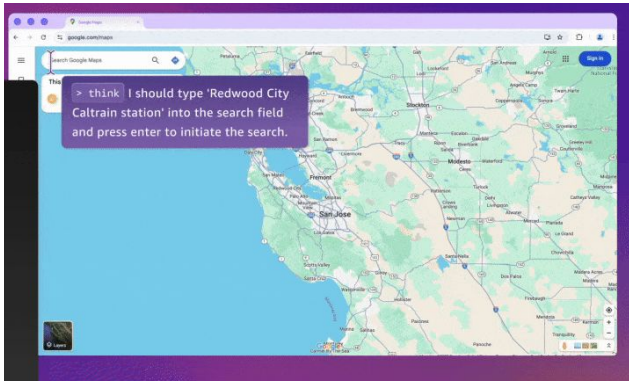
[1] Junlin Xie et al. **Large Multimodal Agents:** A Survey. **23 Feb 2024**. [link]

Who is the Agent?

A*gent* is a system capable of perceiving its **environment** and making **decisions** based on these perceptions to achieve specific **goals**



Cursor **Code Editor**



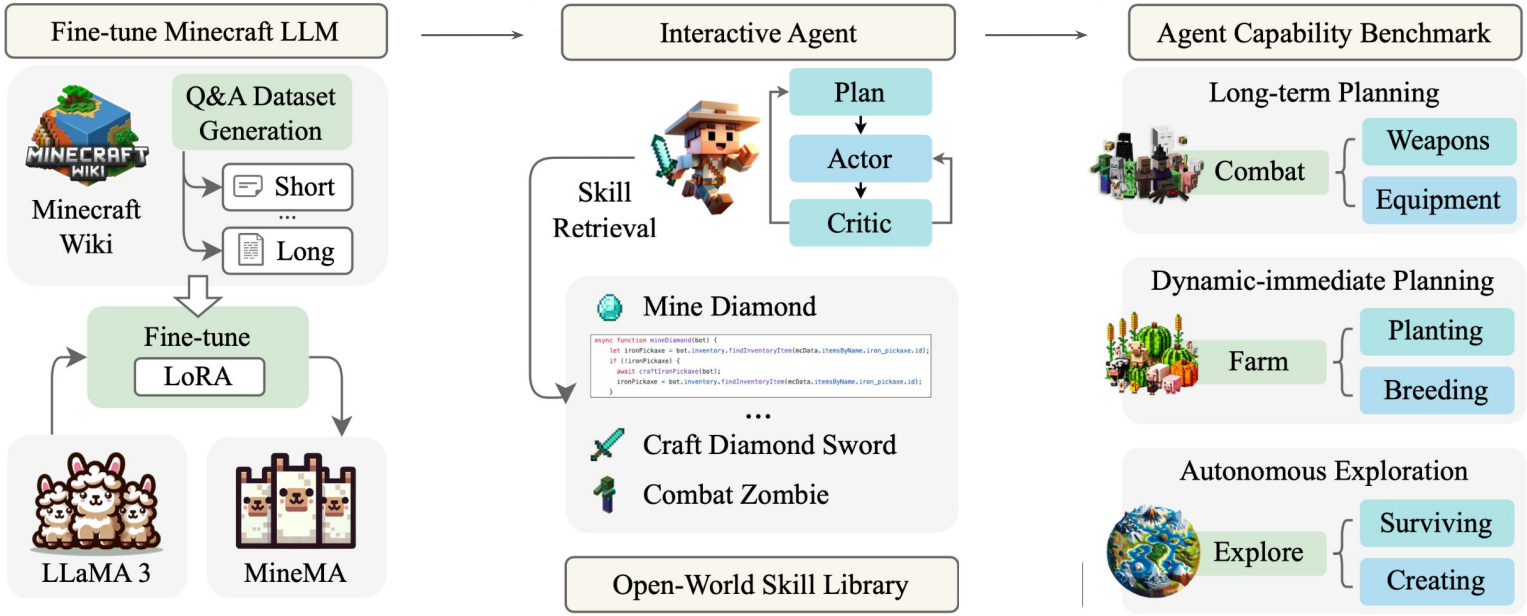
Amazon Nova **Web Agent**

Minecraft game

[1] Junlin Xie et al. **Large Multimodal Agents:** A Survey. **23 Feb 2024**. [link]

Odyssey: Pipeline (Oct 2024)

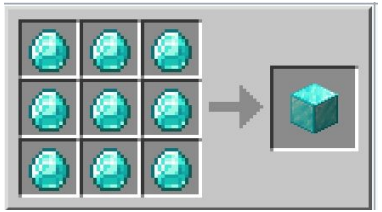
Odyssey — new framework that empowers **Large Language Model (LLM)-based agents** with open-world skills to explore the vast Minecraft world

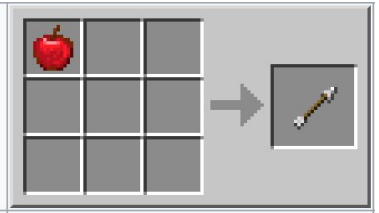


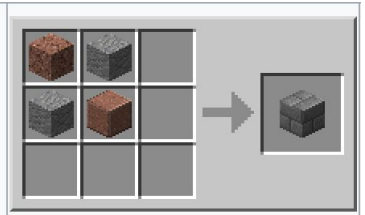
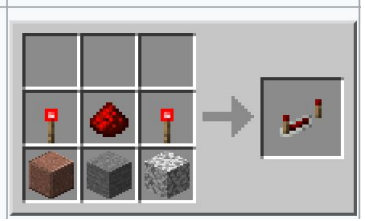
Odyssey: Minecraft Wiki

To improve agent performance in Minecraft, we fine-tune the **LLaMA-3 model** using a large-scale Q&A dataset with **390k+ instruction entries** sourced from the **Minecraft Wiki**

Crafting:



Apple	
Block of Gold + Apple	

Stone Bricks	Any Stone	
Redstone Repeater	Redstone Torch + Redstone Dust + Any Stone	

Odyssey: Minecraft Wiki

To improve agent performance in Minecraft, we fine-tune the **LLaMA-3 model** using a large-scale Q&A dataset with **390k+ instruction entries** sourced from the **Minecraft Wiki**

Passive mobs:

Hostile mobs:

Odyssey: Interactive Agent

Efficient **retrieval of skills** is provided by generating a description for each skill by calling the LLM – **Sentence Transformer** to encode each skill

collectItem.js

```
if (!mob) {
    bot.chat("Could not find a mob.");
    return false;
}
// kill the mob using the sword
await equipBestTool(bot, tool);
await killMob(bot, mob.name, 300);
// collect the dropped items
await bot.pathfinder.goto(new GoalBlock(mob.position.x,
mob.position.y, mob.position.z));
bot.chat("Collected dropped items.");
```

LLM-based agent employs a **planner-actor-critic** architecture to define which actions to do

- **40** primitive skills
- **183** compositional skills

Odyssey: Interactive Agent

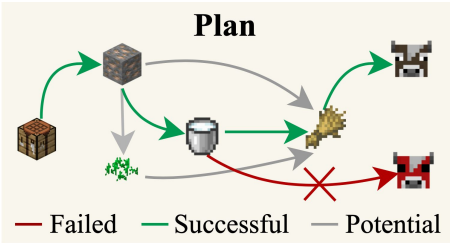
1 **LLM Planner** — breaks down high-level goals into specific low-level subgoals

a) **Ultimate goal** = I want to breed cow and collect items from it.

b) **State of the agent**

[**Position**]: x=2134.5, y=69.0, z=769.5
[**Time**] day
[**Nearby bocks**] dirt, grass, oak_log
[**Nearby entities**] horse, pig

c) **Achievements**



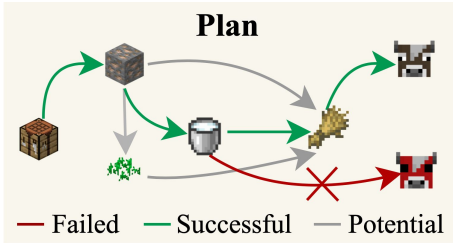
Odyssey: Interactive Agent

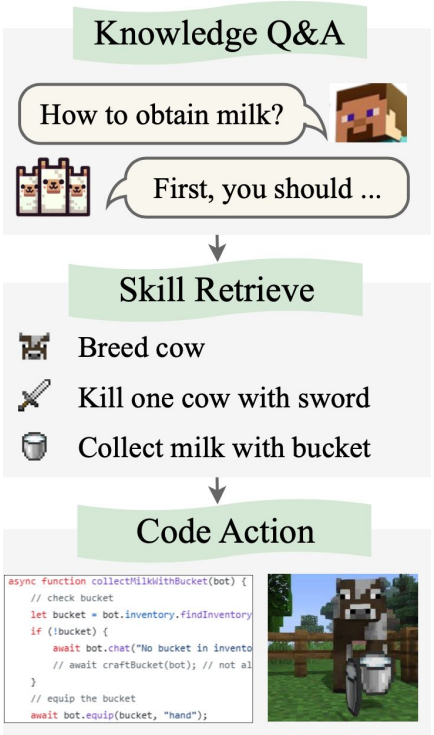
2 **LLM Actor —** invoked to sequentially execute the subgoals generated by the LLM planner within the Minecraft environment

a) Query context

b) Similarity matching

c) Skill Selection





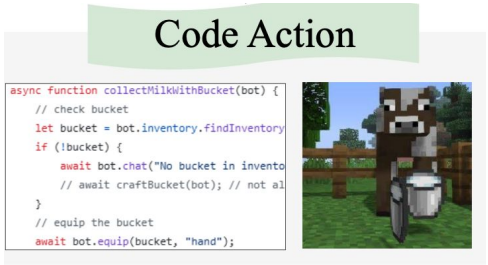
Odyssey: Interactive Agent

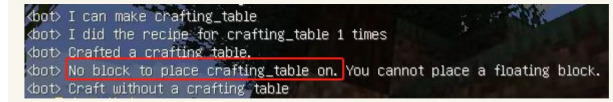
3 **LLM Critic —** noting successful outcomes and failure points, **refine strategy**

a) Execution Feedback

b) Self-validation

c) Self-reflection





Execution Feedback

[Lack of pre-requirements]
I cannot collect milk without a .
[Environment feedback]
I could not find a to collect milk.

Self-validation:

Observation
My subgoal is to:
collect milk
last_inventory (16/36): ...
cur_inventory (18/36): ...

Thought
Based on changes of my inventory, is my subgoal successful?

Self-reflection:

Rethink
You should analysis the reason why my subgoal is failed based on the logs provided.

Critic
Since you only have , you might need the to attract a for milk.

Odyssey: Examples

- Use **GPT-3.5 and GPT-4** for initial **data** generation
- All experiments are conducted with the open-source **LLaMA-3 model**
- Simulation environment is built on top of **Voyager**

Shear a Sheep

Odyssey: Examples

- Use **GPT-3.5 and GPT-4** for initial **data** generation
- All experiments are conducted with the open-source **LLaMA-3 model**
- Simulation environment is built on top of **Voyager**

Mining diamonds from scratch

Odyssey: Examples

- Use **GPT-3.5 and GPT-4** for initial **data** generation
- All experiments are conducted with the open-source **LLaMA-3 model**
- Simulation environment is built on top of **Voyager**

Craft sword and Combat a zombie