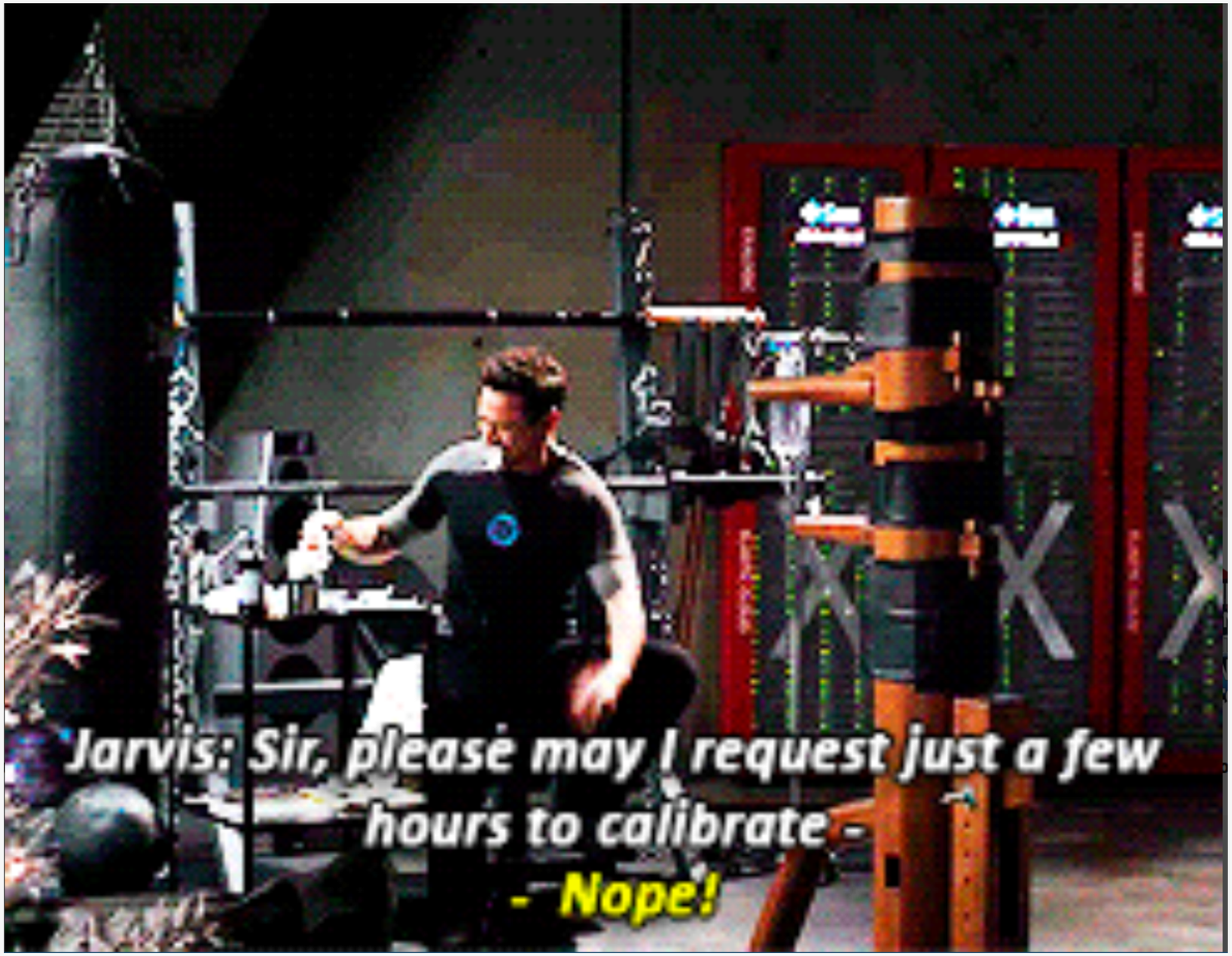
Good Afternoon


I know kung fu.


TADASHI HAMADA


Jarvis: Sir, please may I request just a few hours to calibrate -
- Nope!

MINEDOJO - Building Embodied Agents with Internet-Scale Knowledge

Aldrin Jenson
MDL19CS008

2022 Oct 31
arxiv.org/abs/2206.08853

H.1 Motivation

For what purpose was the dataset created? We create this internet-scale multimodal knowledge base to facilitate research towards open-ended, generally capable embodied agents.

What is an Agent?

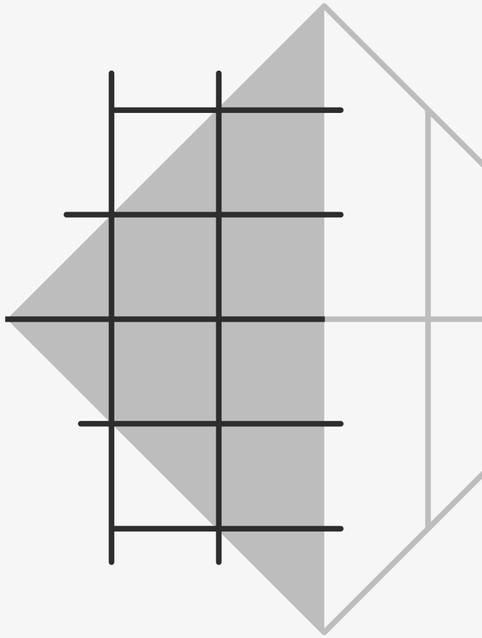
Anything that perceives environment through sensors and act upon that environment through actuators. An Agent runs in the cycle of perceiving, thinking, and acting.

An agent can be:

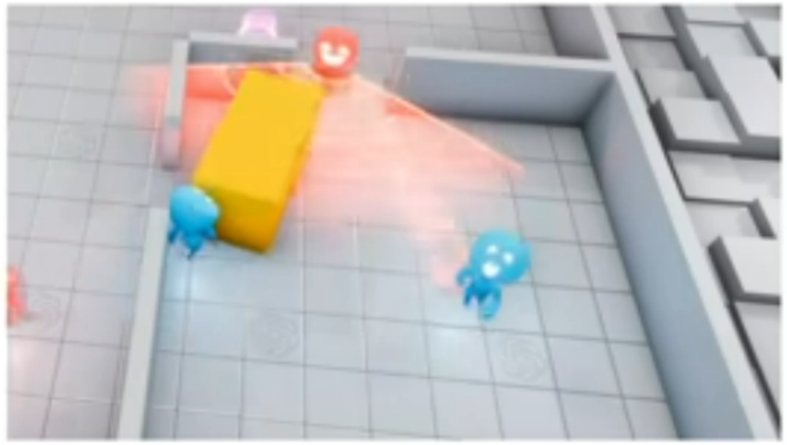
- **Human-Agent**: A human agent has eyes, ears, and other organs which work for sensors and hand, legs, vocal tract work for actuators.
- **Robotic Agent**: A robotic agent can have cameras, infrared range finder, NLP for sensors and various motors for actuators.
- **Software Agent**: Software agent can have keystrokes, file contents as sensory input and act on those inputs and display output on the screen.

What is an Embodied agent?

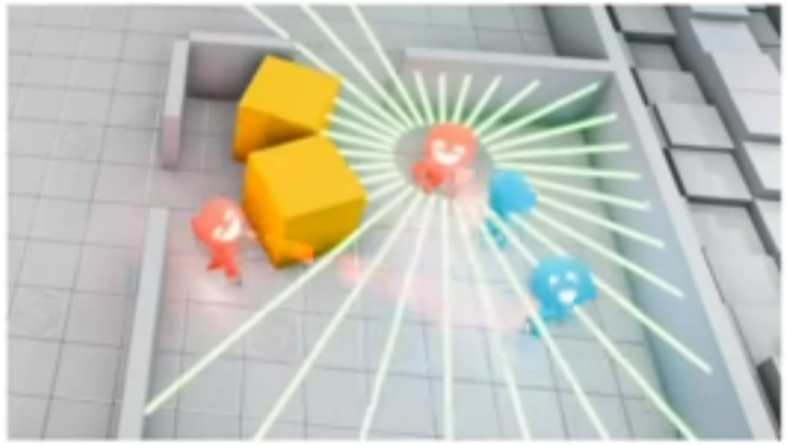
In artificial intelligence, an embodied agent, also sometimes referred to as an interface agent, is **an intelligent agent that interacts with the environment through a physical body within that environment.**



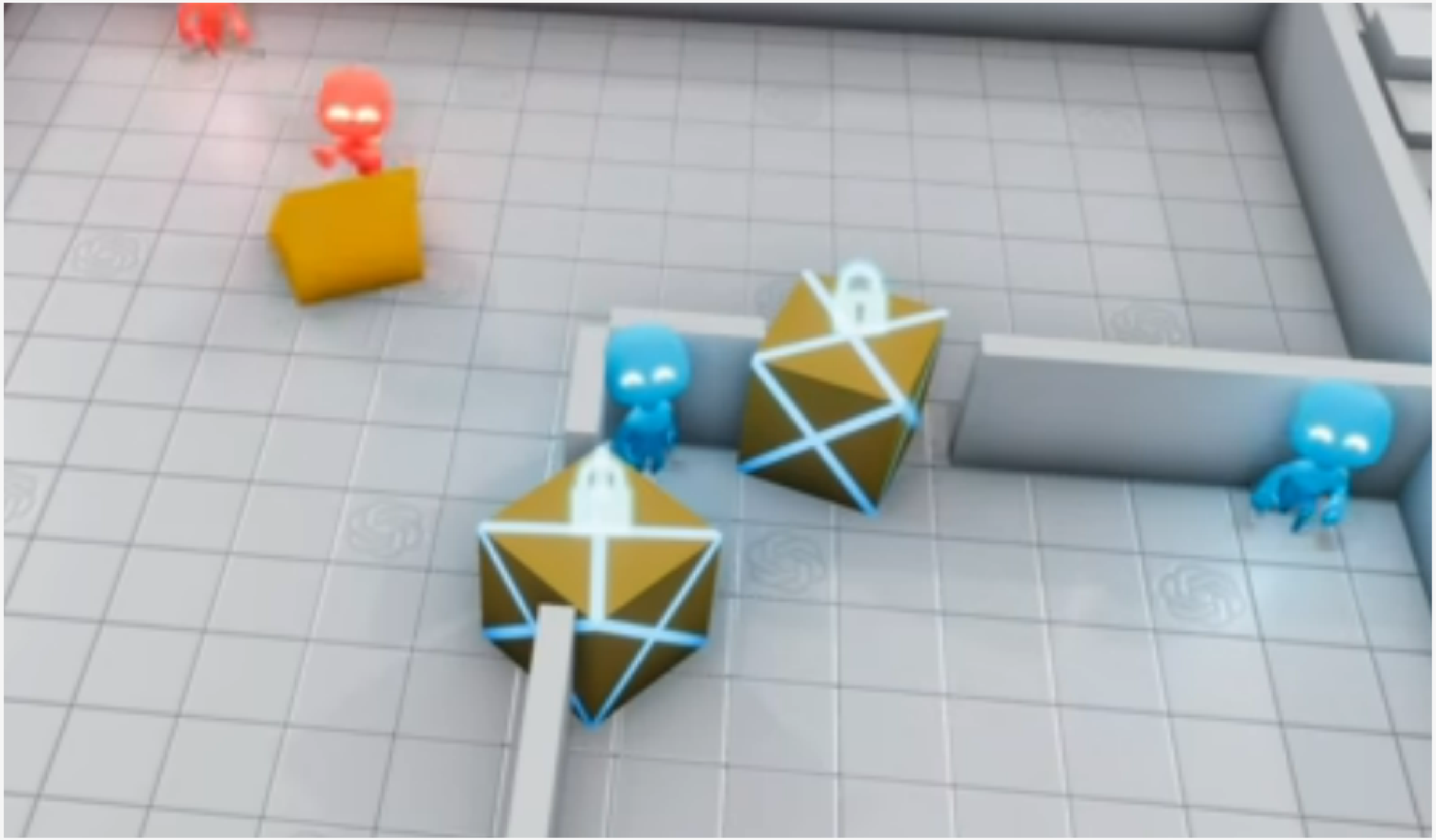
The agents can **move** by setting a force on themselves in the x and y directions as well as rotate along the



The agents can **see** objects in their line of sight and within a frontal cone.



The agents can **sense** distance to objects, walls, and other agents around them using a lidar-like



Three necessary pillars

For the emergence of generalist embodied agents

The environment should enable an unlimited variety of open-ended goals

A large-scale database of prior knowledge

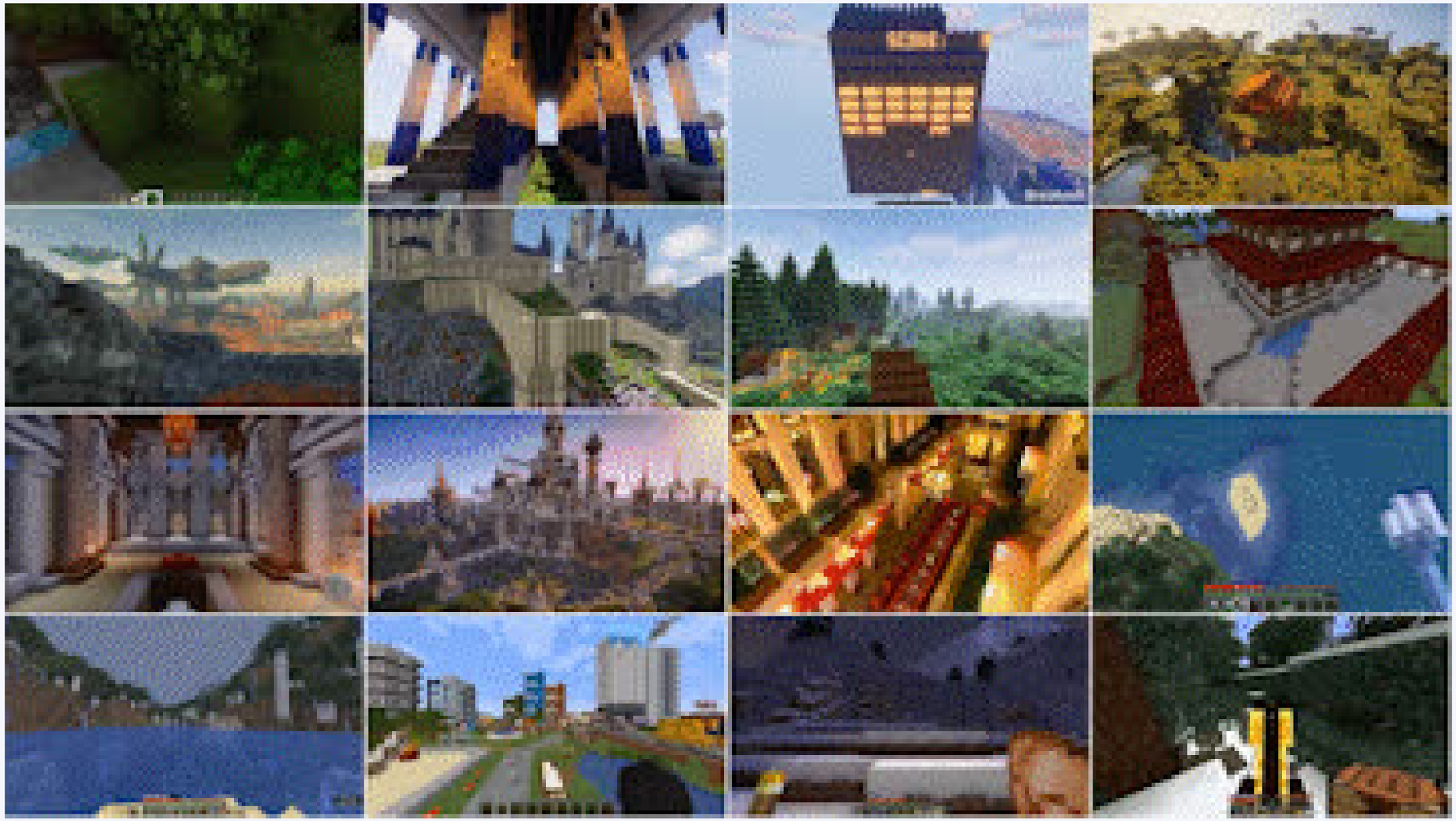
Flexible enough architecture to pursue any task in open-ended environments

Introducing

MineDOJO

A new framework to help the community
develop open-ended, generally-capable agents.

It is built on the popular **Minecraft game,** where a player explores a **procedurally generated 3D world** with **diverse types of terrains** to roam, materials to mine, tools to craft, structures to build, and wonders to discover. Unlike most other games Minecraft defines **no specific reward to maximize and no fixed storyline to follow**, making it **well suited** for developing open-ended environments for embodied AI research.



2 Related work

Open-ended Environments for Decision-making Agents. There are many environments developed with the goal of open-ended agent learning. Prior works include maze-style worlds [106, 114, 52], purely text-based game [60], grid worlds [17, 12], browser/GUI-based environments [93, 109], and indoor simulators for robotics [1, 92, 99, 27, 95, 84, 78]. These benchmarks have enabled and inspired new algorithmic innovations [23, 49, 115, 54]. Minecraft offers an exciting alternative for open-ended agent learning. It is a 3D visual world with procedurally generated landscapes and extremely flexible game mechanics that support an enormous variety of activities.

Minecraft for AI Research. The Malmo platform [51] is the first comprehensive release of a Gym-style agent API [10] for Minecraft. Based on Malmo, MineRL [41] provides a codebase and human play trajectories for the annual Diamond Challenge at NeurIPS [40, 42, 53]. MINEDOJO's simulator builds upon the pioneering work of MineRL, but greatly expands the API and benchmarking task suite. Other Minecraft benchmarks exist with different focuses. For example, CraftAssist [37] and IGLU [57] study agents with interactive dialogues. BASALT [89] applies human evaluation to 4 open-ended tasks. EvoCraft [38] is designed for structure building, and Crafter [43] optimizes for fast experimentation. Unlike prior works, MINEDOJO's core mission is to facilitate the development of generally capable embodied agents using internet-scale knowledge.

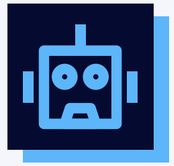
Internet-scale Multimodal Knowledge Bases. Big dataset such as Common Crawl [19], the Pile [30], LAION [85], YouTube-8M [2] and HowTo100M [67] have been fueling the success of large pre-trained language models [21, 80, 11] and multimodal models [103, 6, 68, 127, 7, 4, 120]. While generally useful for learning representations, these datasets are not specifically targeted at embodied agents. To provide agent-centric training data, RoboNet [20] collects video frames from 7 robot platforms, and Ego4D [36] recruits volunteers to record egocentric videos of household activities. In comparison, MINEDOJO's knowledge base is constructed without human curation efforts, much larger in volume, more diverse in data modalities, and comprehensively covers all aspects of the Minecraft environment.

Three necessary pillars

For the emergence of generalist embodied agents

The environment should enable an unlimited variety of open-ended goals

A large-scale database of prior knowledge



Flexible enough architecture to pursue any task in open-ended environments

730K YouTube videos
300K hours gameplay
2.2B words transcripts

~7K Wiki pages
Multimodal text, images, tables, and diagrams

340K Reddit posts
6.6M comments in r/Minecraft subreddit

Figure 2. The MineDojo framework takes advantage of an Internet-scale database to train an AI agent

GPT-3

GPT-3, or the third generation **Generative Pre-trained Transformer,** is a neural network machine learning model trained using internet data to generate any type of text. Developed by **OpenAI**, it requires a small amount of input text to generate large volumes of relevant and sophisticated machine-generated text.

B.3 GPT-3 Guidance

We leverage OpenAI's `GPT-3-davinci` API to automatically generate detailed guidance for a subset of the tasks. Inspired by [58], we adopt the following template to prompt GPT-3: `How to {task goal} in Minecraft? Let's think step by step.` Here are some examples:

The guidance for the task "*find material and craft a gold pickaxe*" is `1) Find a place with a lot of trees; 2) Cut down the trees and gather the wood; 3) Find a place with a lot of stone; 4) Mine the stone and gather the cobblestone; 5) Find a place with a lot of iron; 6) Mine the iron and gather the iron ingots; 7) Find a place with a lot of gold; 8) Mine the gold and gather the gold ingots; 9) Craft a gold pickaxe.`

The guidance for the task "*sail on boat with a sheep*" is `1) Find a boat; 2) Place the sheep in the boat; 3) Right-click on the boat with an empty hand to get in; 4) Use the WASD keys to move the boat. The sheep should stay in the boat.`

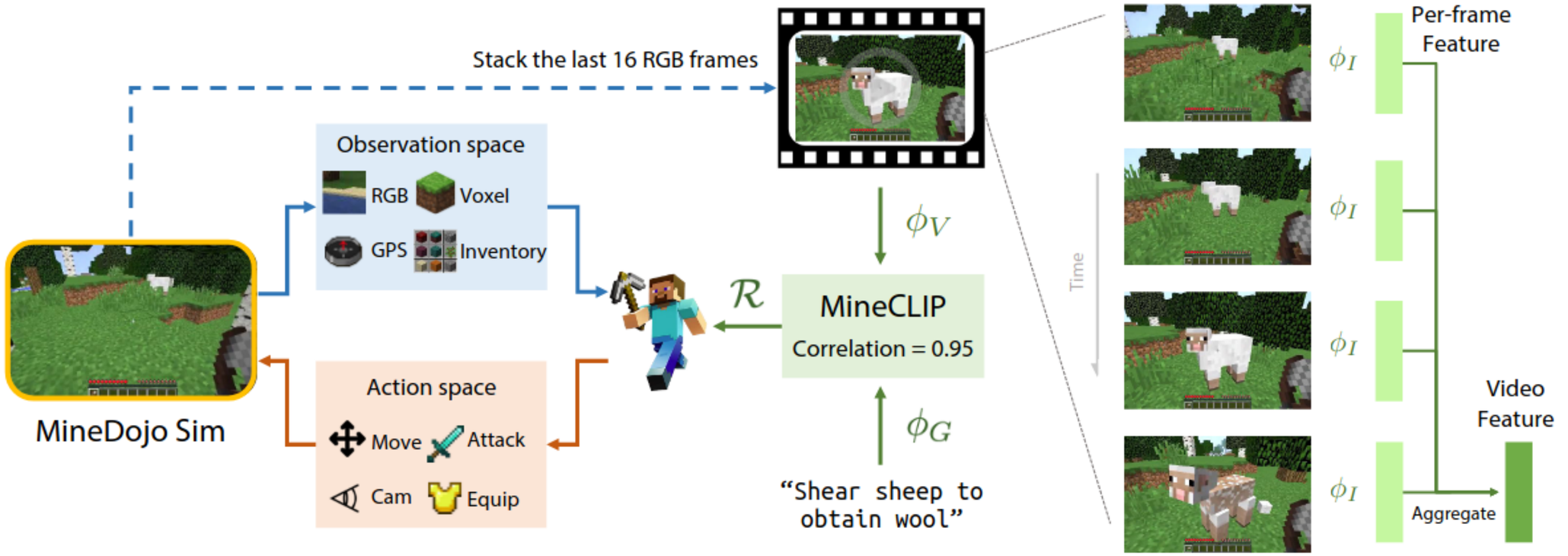


Figure 3: Algorithm design. MINECLIP is a contrastive video-language model pre-trained on MINEDOJO's massive Youtube database. It computes the correlation between an open-vocabulary language goal string and a 16-frame video snippet. The correlation score can be used as a learned dense reward function to train a strong multi-task RL agent.

Simulator

```yaml
survival_sword_food:
  category: survival
  prompt: survive as long as possible given a sword and some food

harvest_wool_with_shears_and_sheep:
  category: harvest
  prompt: harvest wool from a sheep with shears and a sheep nearby

techtree_from_barehand_to_wooden_sword:
  category: tech-tree
  prompt: find material and craft a wooden sword

combat_zombie_pigman_nether_diamond_armors_diamond_sword_shield:
  category: combat
  prompt: combat a zombie pigman in nether with a diamond sword,
    shield, and a full suite of diamond armors
```

Figure A.1: Example specifications. Please refer to `programmatic_tasks.yaml` in the supplementary for a complete listing.

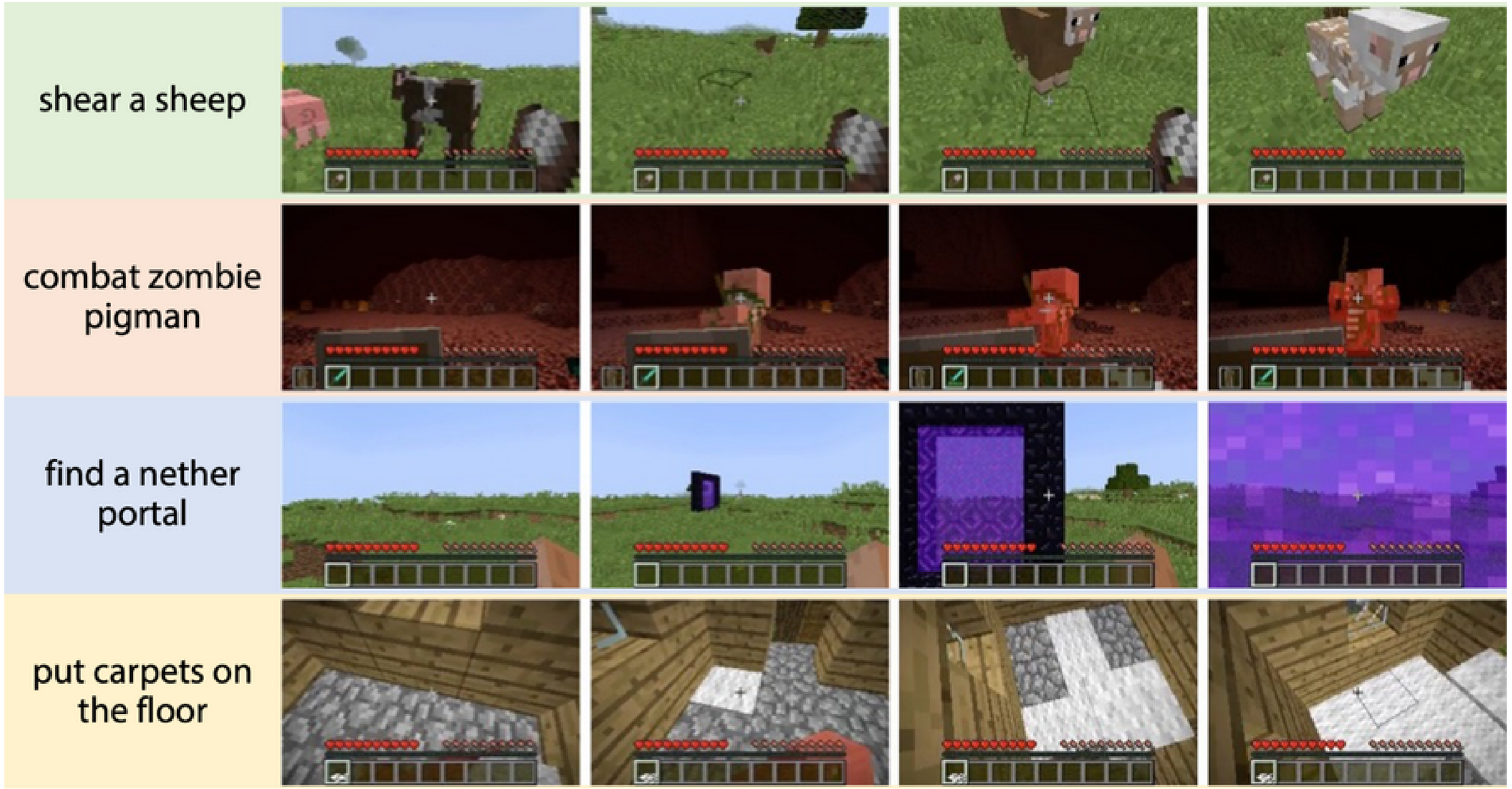


Figure 1. The NVIDIA AI agent follows the prompts within the MineDojo framework

Table 1: Our novel MINECLIP reward model is able to achieve competitive performance with manually written dense reward function for Programmatic tasks, and significantly outperforms the $\text{CLIP}_{\text{OpenAI}}$ method across all Creative tasks.

Group	Tasks	**Ours** (Attn)	**Ours** (Avg)	Manual Reward	Sparse-only	$\text{CLIP}_{\text{OpenAI}}$
	Milk Cow	**64.5 ± 37.1**	6.5 ± 3.5	62.8 ± 40.1	0.0 ± 0.0	0.0 ± 0.0
	Hunt Cow	**83.5 ± 7.1**	0.0 ± 0.0	48.3 ± 35.9	0.3 ± 0.4	0.0 ± 0.0
	Shear Sheep	12.1 ± 9.1	0.6 ± 0.2	**52.3 ± 33.2**	0.0 ± 0.0	0.0 ± 0.0
	Hunt Sheep	8.1 ± 4.1	0.0 ± 0.0	**41.9 ± 33.0**	0.3 ± 0.4	0.0 ± 0.0
	Combat Spider	80.5 ± 13.0	60.1 ± 42.5	**87.5 ± 4.6**	47.8 ± 33.8	0.0 ± 0.0
	Combat Zombie	47.3 ± 10.6	**72.3 ± 6.4**	49.8 ± 26.9	8.8 ± 12.4	0.0 ± 0.0
	Combat Pigman	1.6 ± 2.3	0.0 ± 0.0	**13.6 ± 9.8**	0.0 ± 0.0	0.0 ± 0.0
	Combat Enderman	0.0 ± 0.0	0.0 ± 0.0	0.3 ± 0.2	0.0 ± 0.0	0.0 ± 0.0
	Find Nether Portal	37.4 ± 40.8	**89.8 ± 5.7**	N/A	N/A	26.3 ± 32.6
	Find Ocean	33.4 ± 45.6	**54.3 ± 40.7**	N/A	N/A	9.9 ± 14.1
	Dig Hole	**91.6 ± 5.9**	88.1 ± 13.3	N/A	N/A	0.0 ± 0.0
	Lay Carpet	97.6 ± 1.9	**98.8 ± 1.0**	N/A	N/A	0.0 ± 0.0

Table 2: MINECLIP consistently agrees well with the ground-truth human judgement. Therefore it is accurate enough to serve as an automatic evaluation metric for open-ended Creative tasks.

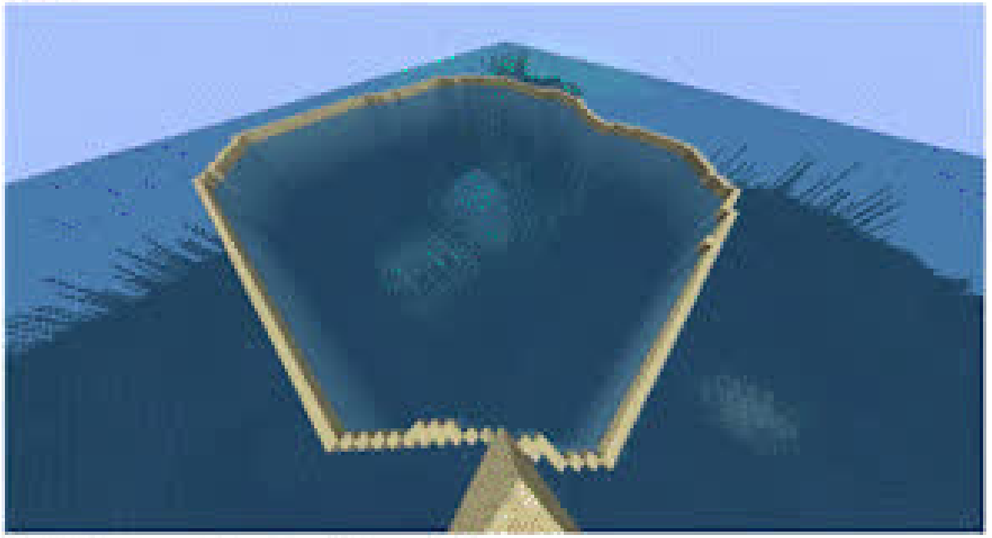
Tasks	Find Nether Portal	Find Ocean	Dig Hole	Lay Carpet
Ours (Attn)	98.7	**100.0**	99.4	97.4
Ours (Avg)	**100.0**	**100.0**	**100.0**	**98.4**
$\text{CLIP}_{\text{OpenAI}}$	48.7	98.4	80.6	54.1

Posted by u/BlockTrapKing 11 months ago

55

My first time encasing an Ocean Monument. Help! What is the best way to sponge it!

Help



49 Comments Award Share Save ...

With sponges

5x5 sections. Sponges only have so much power.

Sponges or a lot of dirt/sand

5x5 grid of sand squars. No joke but it let's you go down in a back and forth without fear of creating more source blocks

r/Minecraft · Posted by u/ERROR_171 2 months ago

689

does anyone know why some of my wheat won't grow?(farming)

Help

121 Comments Award Share Save ...

Gotta be the light add more torches

Light

It'll be your light level, as the wheat near the torch has grown. All you need to do is place more torches and you should see a difference.

light lvl too low

E.3 RL Training

PPO. We use the popular PPO algorithm [87] (Proximal Policy Optimization) as our RL training backbone. PPO is an on-policy method that optimizes for a surrogate objective while ensuring that the deviation from the previous policy is relatively small. PPO updates the policy network by

$$\underset{\theta}{\text{maximize}}\ \mathbb{E}_{s,a\sim\pi_{\theta_{\text{old}}}} L(s, a, \theta_{\text{old}}, \theta), \tag{1}$$

where

$$L(s, a, \theta_{\text{old}}, \theta) = \min\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)} A^{\pi_{\theta_{\text{old}}}}(s, a), \text{clip}\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)}, 1-\epsilon, 1+\epsilon\right) A^{\pi_{\theta_{\text{old}}}}(s, a)\right). \tag{2}$$

A is an estimator of the advantage function (GAE [86] in our case) and ϵ is a hyperparameter that controls the deviation between the new policy and the old one.

Self Imitation Learning. We apply self-imitation learning [74] (SI) to further improve sample efficiency because computing the reward with MINECLIP in the loop makes the training more expensive. Self-imitation learning is essentially supervised learning on a buffer $\mathcal{D}_{SI}$ of good trajectories generated by the agent's past self. In our case, the trajectories are generated by the behavior policy during PPO rollouts, and only added to $\mathcal{D}_{SI}$ if it is a *successful* trial or if the episodic return exceeds a certain threshold. Self imitation optimizes π_θ for the objective $\mathcal{J}_{SI} = \mathbb{E}_{s,a\sim\mathcal{D}_{SI}} \log \pi_\theta(a|s)$ with respect to θ.

We alternate between the PPO phase and the SI phase. A pseudocode of our interleaved training procedure is given in Algorithm 1. We use a *prioritized* strategy to sample trajectories from the buffer $\mathcal{D}_{SI}$. Specifically, we assign equal probability to all successful trajectories. Unsuccessful trajectories can still be sampled but with lower probabilities proportional to their episodic returns.

Three major contributions

Simulation platform with thousands of diverse open-ended tasks.

2 times greater than MINE RL

Internet-scale multimodal Minecraft knowledge base.

Novel algorithm for embodied agents with large-scale pre-training.

Conclusion

" In summary, this paper proposes an **open-ended task suite, internet-scale domain knowledge**, and agent learning with r**ecent advances on large pre-trained models.** MINEDOJO's simulator suite and knowledge base will be **open-sourced**. We hope that MINEDOJO will serve as an effective starter framework for the community to develop new algorithms and advance towards generally capable embodied agent"

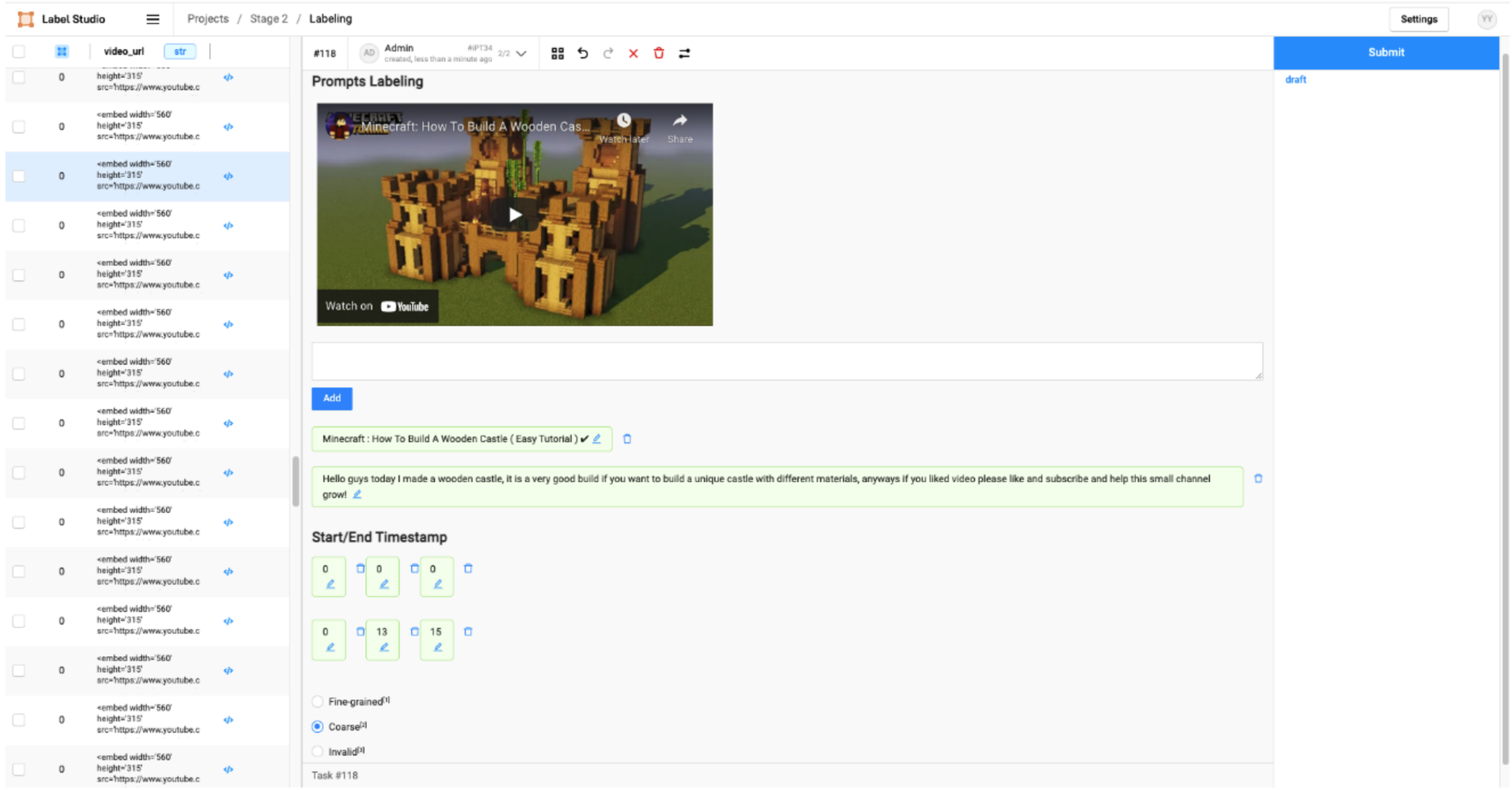


Figure A.2: Labeling UI to mine tasks from YouTube. A human annotator can choose to reject the video (*Invalid*), adjust the timestamps, select the title, or edit and expand the original description to be the new task goal.

Authors

Jim (Linxi) Fan

Guanzhi Wang*

Yunfan Jiang*

Ajay Mandlekar

Yuncong Yang

Haoyi Zhu

Andrew Tang

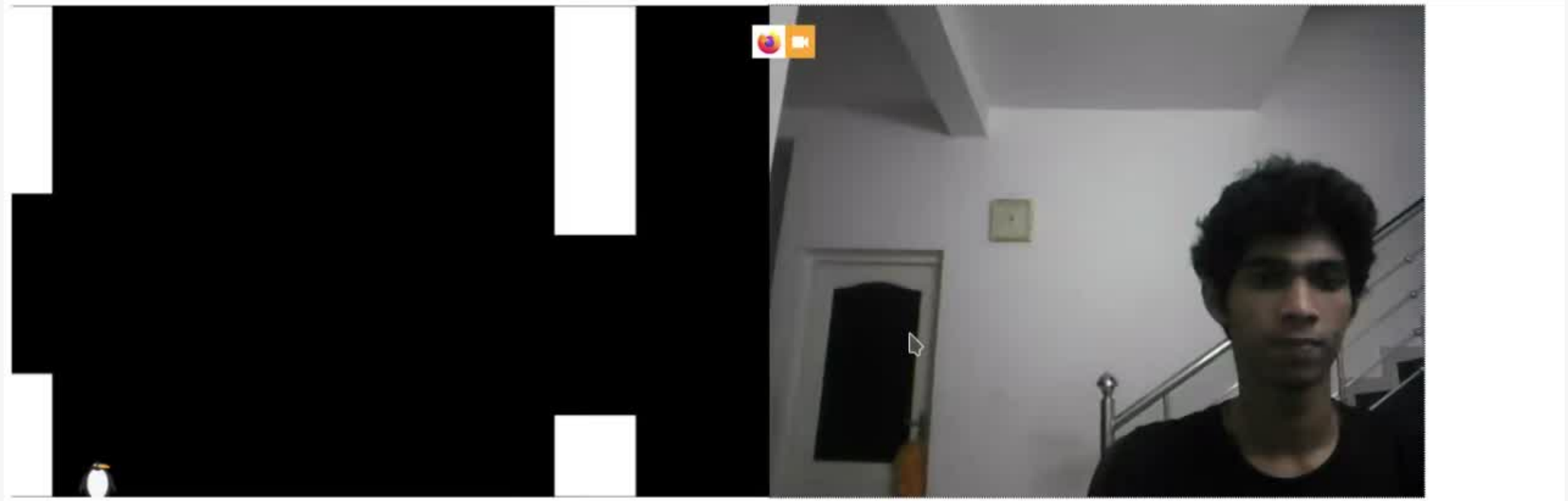
De-An Huang

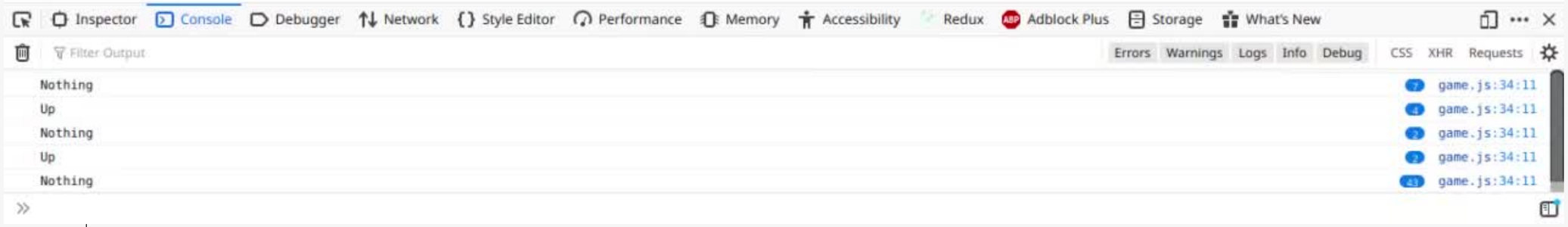
Yuke Zhu†

Anima Anandkumar†

https://minedojo.org/

My thoughts




Inspector
Console
Debugger
Network
Style Editor
Performance
Memory
Accessibility
Redux
Adblock Plus
Storage
What's New
Filter Output
Errors
Warnings
Logs
Info
Debug
CSS
XHR
Requests
Nothing
game.js:34:11
Up
game.js:34:11
Nothing
game.js:34:11
Up
game.js:34:11
Nothing
game.js:34:11

Thank you

References

- https://developer.nvidia.com/blog/building-generally-capable-ai-agents-with-minedojo/
- arxiv.org/abs/2206.08853
- https://www.youtube.com/user/keeroyz