MINEDOJO: Building Open-Ended Embodied Agents with Internet-Scale Knowledge

Linxi Fan[1], **Guanzhi Wang**[2*], **Yunfan Jiang**[3*], **Ajay Mandlekar**[1], **Yuncong Yang**[4], **Haoyi Zhu**[5], **Andrew Tang**[4], **De-An Huang**[1], **Yuke Zhu**[1 6†], **Anima Anandkumar**[1 2†]
[1]NVIDIA, [2]Caltech, [3]Stanford, [4]Columbia, [5]SJTU, [6]UT Austin
[*]Equal contribution [†]Equal advising
https://minedojo.org

Abstract

Autonomous agents have made great strides in specialist domains like Atari games and Go. However, they typically learn *tabula rasa* in isolated environments with limited and manually conceived objectives, thus failing to generalize across a wide spectrum of tasks and capabilities. Inspired by how humans continually learn and adapt in the open world, we advocate a trinity of ingredients for building generalist agents: 1) an environment that supports a multitude of tasks and goals, 2) a large-scale database of multimodal knowledge, and 3) a flexible and scalable agent architecture. We introduce MINEDOJO, a new framework built on the popular *Minecraft* game that features a simulation suite with thousands of diverse open-ended tasks and an internet-scale knowledge base with Minecraft videos, tutorials, wiki pages, and forum discussions. Using MINEDOJO's data, we propose a novel agent learning algorithm that leverages large pre-trained video-language models as a learned reward function. Our agent is able to solve a variety of open-ended tasks specified in free-form language without any manually designed dense shaping reward. We open-source the simulation suite, knowledge bases, algorithm implementation, and pretrained models (https://minedojo.org) to promote research towards the goal of generally capable embodied agents.

1 Introduction

Developing autonomous embodied agents that can attain human-level performance across a wide spectrum of tasks has been a long-standing goal for AI research. There has been impressive progress towards this goal, most notably in games [62, 66, 96] and robotics [53, 74, 110, 101, 80]. These embodied agents are typically trained *tabula rasa* in isolated worlds with limited complexity and diversity. Although highly performant, they are specialist models that do not generalize beyond a narrow set of tasks. In contrast, humans inhabit an infinitely rich reality, continuously learn from and adapt to a wide variety of open-ended tasks, and are able to leverage large amount of prior knowledge from their own experiences as well as others.

We argue that **three main pillars** are necessary for generalist embodied agents to emerge. First, the environment in which the agent acts needs to **enable an unlimited variety of open-ended goals** [88, 55, 92, 89]. Natural evolution is able to nurture an ever-expanding tree of diverse life forms thanks to the infinitely varied ecological settings that the Earth supports [89, 98]. This process has not stagnated for billions of years. In contrast, today's agent training algorithms cease to make new progress after convergence in narrow environments [62, 110]. Second, a **large-scale database of prior knowledge** is necessary to facilitate learning in open-ended settings. Just as humans frequently learn from the internet, agents should also be able to harvest practical knowledge encoded in large amounts of video demos [31, 59], multimedia tutorials [61], and forum discussions [97, 51, 41]. In a

36th Conference on Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks.

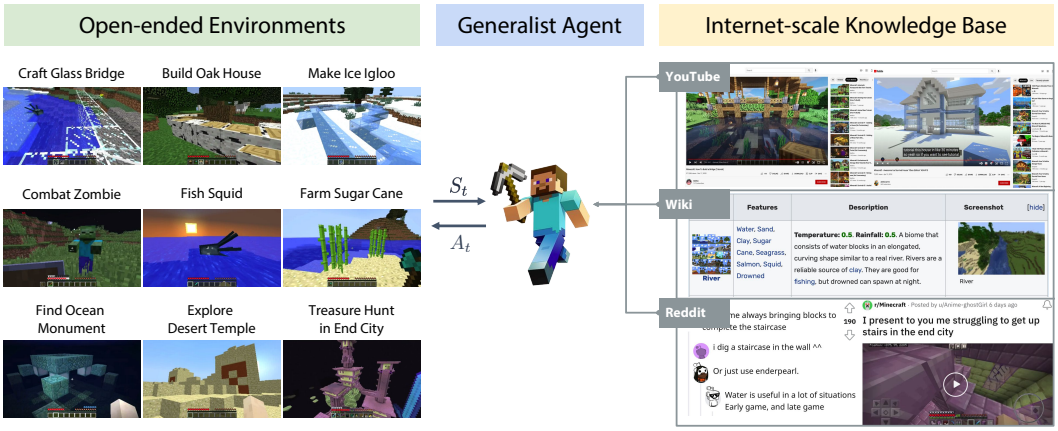

Figure 1: MINEDOJO is a novel framework for developing open-ended, generally capable agents that can learn and adapt continually to new goals. MINEDOJO features a benchmarking suite with ***thousands* of diverse open-ended tasks** specified in natural language prompts, and also provides an **internet-scale, multimodal knowledge base** of YouTube videos, Wiki pages, and Reddit posts. The database captures the collective experience and wisdom of millions of Minecraft gamers for an AI agent to learn from. Best viewed zoomed in.

complex world, it would be extremely inefficient for an agent to learn everything from scratch through trial and error. Third, the **agent's architecture** needs to be flexible enough to pursue any task in open-ended environments, and scalable enough to convert large-scale knowledge sources into actionable insights [16, 72]. This motivates the design of an agent that has a unified observation/action space, conditions on natural language task prompts, and adopts the Transformer pre-training paradigm [23, 68, 13] to internalize knowledge effectively.

In light of these three pillars, we introduce 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 [62, 66, 96], 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. We make the following three major contributions:

1. Simulation platform with thousands of diverse open-ended tasks. MINEDOJO provides convenient APIs on top of Minecraft that standardizes task specification, world settings, and agent's observation/action spaces. We introduce a benchmark suite that consists of thousands of natural language-prompted tasks, making it *two orders of magnitude* larger than prior Minecraft benchmarks like the MineRL Challenge [36, 49]. The suite includes long-horizon, open-ended tasks that cannot be easily evaluated through automated procedures, such as "*build an epic modern house with two floors and a swimming pool*". Inspired by the Inception score [73] and FID score [42] that are commonly used to assess AI-generated image quality, we introduce a novel agent evaluation protocol using a large video-language model pre-trained on Minecraft YouTube videos. This complements human scoring [78] that is precise but more expensive. Our learned evaluation metric has good agreement with human judgment in a subset of the full task suite considered in the experiments.

2. Internet-scale multimodal Minecraft knowledge base. Minecraft has more than 100 million active players [100], who have collectively generated an enormous wealth of data. They record tutorial videos, stream live play sessions, compile recipes, and discuss tips and tricks on forums. MINEDOJO features a massive collection of 730K+ YouTube videos with time-aligned transcripts, 6K+ free-form Wiki pages, and 340K+ Reddit posts with multimedia contents (Fig. 3). We hope that this enormous knowledge base can help the agent acquire diverse skills, develop complex strategies, discover interesting objectives, and learn actionable representations automatically.

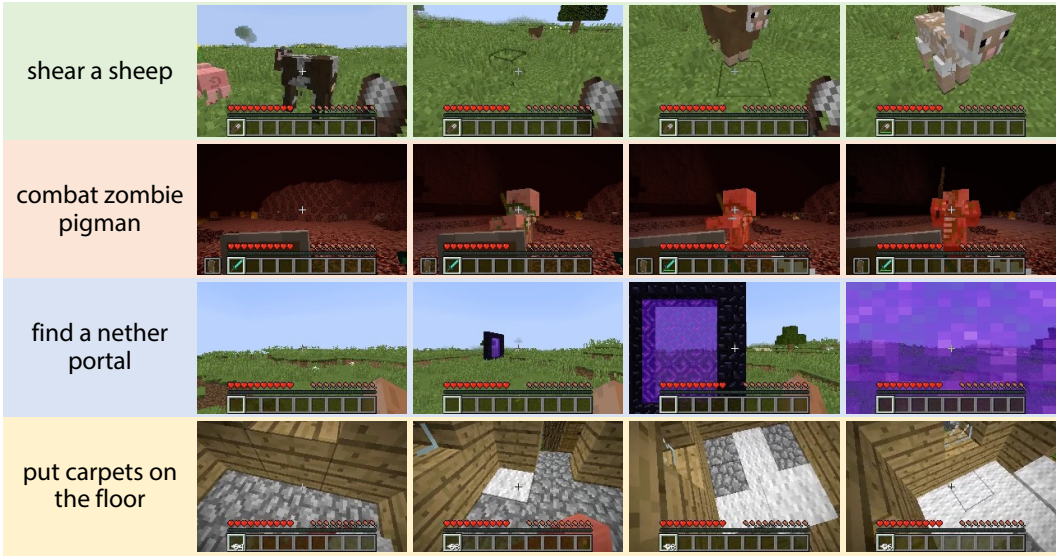

Figure 2: Visualization of our agent's learned behaviors on four selected tasks. Leftmost texts are the task prompts used in training. Best viewed on a color display.

3. Novel algorithm for embodied agents with large-scale pre-training. We develop a new learning algorithm for embodied agents that makes use of the internet-scale domain knowledge we have collected from the web. Using the massive volume of YouTube videos from MINEDOJO, we train a video-text contrastive model in the spirit of CLIP [69], which associates natural language subtitles with their time-aligned video segments. We demonstrate that this learned correlation score can be used effectively as an *open-vocabulary, massively multi-task reward function* for RL training. Our agent solves the majority of 12 tasks in our experiment using the learned reward model (Fig. 2). It achieves competitive performance to agents trained with meticulously engineered dense-shaping rewards, and in some cases outperforms them, with up to 73% improvement in success rates. For open-ended tasks that do not have a simple success criterion, our agents also perform well without any special modifications.

In summary, this paper proposes an open-ended task suite, internet-scale domain knowledge, and agent learning with recent advances on large pre-trained models [11]. We have open-sourced MINEDOJO's simulator, knowledge bases, algorithm implementations, pretrained model checkpoints, and task curation tools at `https://minedojo.org/`. 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.

2 MINEDOJO Simulator & Benchmark Suite

MINEDOJO offers a set of simulator APIs help researchers develop generally capable, open-ended agents in Minecraft. It builds upon the open-source MineRL codebase [36] and makes the following upgrades: 1) We provide **unified observation and action spaces** across all tasks, facilitating the development of multi-task and continually learning agents that can constantly adapt to new scenarios and novel tasks. This deviates from the MineRL Challenge design that tailors observation and action spaces to individual tasks; 2) Our simulation unlocks all three types of worlds in Minecraft, including the *Overworld*, the *Nether*, and the *End*, which **substantially expands the possible task space**, while MineRL only supports the Overworld natively; and 3) We provide convenient APIs to configure initial conditions and world settings to standardize our tasks.

With this MINEDOJO simulator, we define thousands of benchmarking tasks, which are divided into two categories: 1) *Programmatic tasks* that can be automatically assessed based on the ground-truth simulator states; and 2) *Creative tasks* that do not have well-defined or easily-automated success criteria, which motivates our novel evaluation protocol using a learned model (Sec. 4). To scale up

the number of Creative tasks, we mine ideas from YouTube tutorials and use OpenAI's GPT-3 [13] service to generate substantially more task definitions. Compared to Creative tasks, Programmatic tasks are simpler to get started, but tend to have restricted scope, limited language variations, and less open-endedness in general.

2.1 Task Suite I: Programmatic Tasks

We formalize each programmatic task as a 5-tuple: $T = (G, \mathcal{G}, \mathcal{I}, f_{\mathcal{S}}, f_{\mathcal{R}})$. G is an English description of the task goal, such as "*find material and craft a gold pickaxe*". $\mathcal{G}$ is a natural language guidance that provides helpful hints, recipes, or advice to the agent. We leverage OpenAI's `GPT-3-davinci` API to automatically generate detailed guidance for a subset of the tasks. For the example goal "*bring a pig into Nether*", GPT-3 returns: `1) Find a pig in the overworld; 2) Right-click on the pig with a lead; 3) Right-click on the Nether Portal with the lead and pig selected; 4) The pig will be pulled through the portal!` $\mathcal{I}$ is the initial conditions of the agent and the world, such as the initial inventory, spawn terrain, and weather. $f_{\mathcal{S}}$: $s_t \rightarrow \{0, 1\}$ is the success criterion, a deterministic function that maps the current world state s_t to a Boolean success label. $f_{\mathcal{R}}$: $s_t \rightarrow \mathbb{R}$ is an optional dense reward function. We only provide $f_{\mathcal{R}}$ for a small subset of the tasks in MINEDOJO due to the high costs of meticulously crafting dense rewards. For our current agent implementation (Sec. 4.1), we do not use detailed guidance. Inspired by concurrent works SayCan [3] and Socratic Models [107], one potential idea is to feed each step in the guidance to our learned reward model sequentially so that it becomes a stagewise reward function for a complex multi-stage task.

MINEDOJO provides 4 categories of programmatic tasks with 1,581 template-generated natural language goals to evaluate the agent's different capabilities systematically and comprehensively, including: 1) **Survival**: surviving for a designated number of days, 2) **Harvest**: finding, obtaining, cultivating, or manufacturing hundreds of materials and objects, 3) **Tech Tree**: crafting and using a hierarchy of tools, and 4) **Combat**: fighting various monsters and creatures that require fast reflex and martial skills. Each template has a number of variations based on the terrain, initial inventory, quantity, etc., which form a flexible spectrum of difficulty. In comparison, the NeurIPS MineRL Diamond challenge [36] is a subset of our programmatic task suite, defined by the task goal "*obtain 1 diamond*" in MINEDOJO.

2.2 Task Suite II: Creative Tasks

We define each creative task as a 3-tuple, $T = (G, \mathcal{G}, \mathcal{I})$, which differs from programmatic tasks due to the lack of straightforward success criteria. Inspired by model-based metrics like the Inception score [73] and FID score [42] for image generation, we design a novel task evaluation metric based on a pre-trained contrastive video-language model (Sec. 4.1). In the experiments, we find that the learned metric exhibits a high level of agreement with human evaluations (see Table 2).

We brainstormed and authored 216 Creative tasks, such as "*build a haunted house with zombie inside*" and "*race by riding a pig*". Nonetheless, such a manual approach is not scalable. Therefore, we develop two systematic approaches to extend the total number of task definitions to 1,560. This makes our Creative tasks *3 orders of magnitude* larger than Minecraft BASALT challenge [78], which has 4 Creative tasks.

Approach 1. Task Mining from YouTube Tutorial Videos. We identify our YouTube dataset as a rich source of tasks, as many human players demonstrate and narrate creative missions in the tutorial playlists. To collect high-quality tasks and accompanying videos, we design a 3-stage pipeline that makes it easy to find and annotate interesting tasks (see supplementary for details). Through this pipeline, we extract 1,042 task ideas from the common wisdom of a huge number of veteran Minecraft gamers, such as "*make an automated mining machine*" and "*grow cactus up to the sky*".

Approach 2. Task Creation by GPT-3. We leverage GPT-3's few-shot capability to generate new task ideas by seeding it with the tasks we manually authored or mined from YouTube. The prompt template is: `Here are some example creative tasks in Minecraft: {a few examples}. Let's brainstorm more detailed while reasonable creative tasks in Minecraft.` GPT-3 contributes 302 creative tasks after de-duplication, and demonstrates a surprisingly proficient understanding of Minecraft terminology.

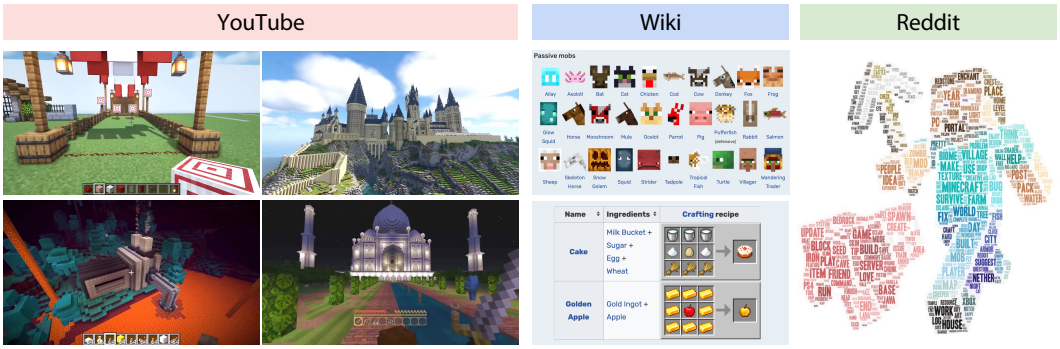

Figure 3: MINEDOJO's internet-scale, multimodal knowledge base. **Left, YouTube videos:** Minecraft gamers showcase the impressive feats they are able to achieve. Clockwise order: an archery range, Hogwarts castle, Taj Mahal, a Nether homebase. **Middle, Wiki:** Wiki pages contain multimodal knowledge in structured layouts, such as comprehensive catalogs of creatures and recipes for crafting. **Right, Reddit:** We create a word cloud from Reddit posts and comment threads. Gamers ask questions, share achievements, and discuss strategies extensively. Best viewed zoomed in.

2.3 Collection of Starter Tasks

We curate a set of 64 core tasks for future researchers to get started more easily. If their agent works well on these tasks, they can more confidently scale to the full benchmark. 1) **32 programmatic tasks**: 16 "standard" and 16 "difficult", spanning all 4 categories (survival, harvesting, combat, and tech tree). We rely on our Minecraft knowledge to decide the difficulty level. "Standard" tasks require fewer steps and lower resource dependencies to complete; and 2) **32 creative tasks**: 16 "standard" and 16 "difficult". Similarly, tasks labeled with "standard" are typically short-horizon tasks. We recommend that researchers run 100 evaluation episodes for each task and report the percentage success rate. The programmatic tasks have ground-truth success, while the creative tasks need our novel evaluation protocol (Sec. 5).

3 Internet-scale Knowledge Base

Two commonly used approaches [85, 96, 66, 27] to train embodied agents include training agents from scratch using RL with well-tuned reward functions for each task, or using a large amount of human-demonstrations to bootstrap agent learning. However, crafting well-tuned reward functions is challenging or infeasible for our task suite (Sec. 2.2), and employing expert gamers to provide large amounts of demonstration data would also be costly and infeasible [96].

Instead, we turn to the open web as an ever-growing, virtually unlimited source of learning material for embodied agents. The internet provides a vast amount of domain knowledge about Minecraft, which we harvest by extensive web scraping and filtering. We collect 33 years worth of YouTube videos, 6K+ Wiki pages, and millions of Reddit comment threads. Instead of hiring a handful of human demonstrators, we capture the collective wisdom of millions of Minecraft gamers around the world. Furthermore, language is a key and pervasive component of our database that takes the form of YouTube transcripts, textual descriptions in Wiki, and Reddit discussions. Language facilitates open-vocabulary understanding, provides grounding for image and video modalities, and unlocks the power of large language models [23, 82, 13] for embodied agents. To ensure socially responsible model development, we take special measures to filter out low-quality and toxic contents [11, 39] from our databases, detailed in the supplementary.

YouTube Videos and Transcripts. Minecraft is among the most streamed games on YouTube [30]. Human players have demonstrated a stunning range of creative activities and sophisticated missions that take hours to complete (examples in Fig. 3). We collect 730K+ narrated Minecraft videos, which add up to 33 years of duration and 2.2B words in English transcripts. In comparison, HowTo100M [59] is a large-scale human instructional video dataset that includes 15 years of experience in total – about half of our volume. The time-aligned transcripts enable the agent to ground free-form natural lan-

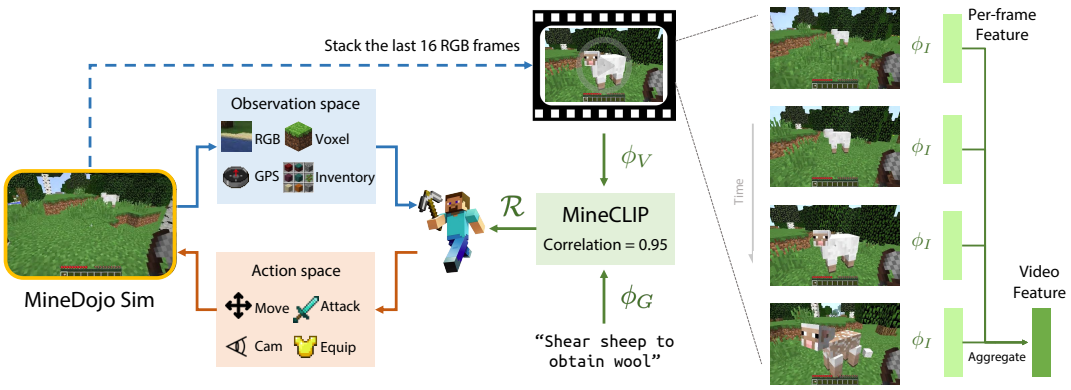

Figure 4: Algorithm design. MINECLIP is a contrastive video-language model pre-trained on MINEDOJO's massive Youtube database. It computes the correlation between an open-vocabulary language goal string and a 16-frame video snippet. The correlation score can be used as a learned dense reward function to train a strong multi-task RL agent.

guage in video pixels and learn the semantics of diverse activities without laborious human labeling. We operationalize this insight in our pre-trained video-language model (Sec. 4.1).

Minecraft Wiki. The Wiki pages cover almost every aspect of the game mechanics, and supply a rich source of unstructured knowledge in multimodal tables, recipes, illustrations, and step-by-step tutorials. We use Selenium [77] to scrape 6,735 pages that interleave text, images, tables, and diagrams. The pages are highly unstructured and do not share any common schema, as the Wiki is meant for human consumption rather than AI training. To preserve the layout information, we additionally save the screenshots of entire pages and extract 2.2M bounding boxes of the visual elements (visualization in supplementary). We do not use Wiki data in our current experiments. Since the Wiki contains detailed recipes for all crafted objects, they could be provided as input or training data for hierarchical planning methods and policy sketches [7]. Another promising future direction is to apply document understanding models such as LayoutLM [105, 104] and DocFormer [8] to learn actionable knowledge from these unstructured Wiki data.

Reddit. We scrape 340K+ posts along with 6.6M comments under the "r/Minecraft" subreddit. These posts ask questions on how to solve certain tasks, showcase cool architectures and achievements in image/video snippets, and discuss general tips and tricks for players of all expertise levels. We do not use Reddit data to train our current agent. A potential idea is to finetune large language models [23, 68] on our Reddit corpus to generate instructions and execution plans that are better grounded in the Minecraft domain. Concurrent works [3, 43, 107] have explored similar ideas and showed excellent results on robot learning, which is encouraging for more future research in MINEDOJO.

4 Agent Learning with Large-scale Pre-training

One of the grand challenges of embodied AI is to build a single agent that can complete a wide range of open-world tasks. The MINEDOJO framework aims to facilitate new techniques towards this goal by providing an open-ended task suite (Sec. 2) and large-scale internet knowledge base (Sec. 3). Here we take an initial step towards this goal by developing a proof of concept that demonstrates how a single language-prompted agent can be trained in MINEDOJO to complete several complex Minecraft tasks. To this end, we propose a novel agent learning algorithm that takes advantage of the massive YouTube data offered by MINEDOJO. We note that this is only one of the numerous possible ways to use MINEDOJO's internet database — the Wiki and Reddit corpus also hold great potential to drive new algorithm discoveries for the community in future works.

In this paper, we consider a multi-task reinforcement learning (RL) setting, where an agent is tasked with completing a collection of MINEDOJO tasks specified by language instructions (Sec. 2). Solving these tasks often requires the agent to interact with the Minecraft world in a prolonged fashion. Agents developed in popular RL benchmarks [91, 110] often rely on meticulously crafted dense and

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. Entries represent percentage success rates averaged over 3 seeds, each tested for 200 episodes. Success conditions are precise in Programmatic tasks, but estimated by MineCLIP for Creative tasks.

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

task-specific reward functions to guide random explorations. However, these rewards are hard or even infeasible to define for our diverse and open-ended tasks in MINEDOJO. To address this challenge, our key insight is to learn **a dense, language-conditioned reward function from in-the-wild YouTube videos and their transcripts**. Therefore, we introduce **MINECLIP**, a contrastive video-language model that learns to correlate video snippets and natural language descriptions (Fig. 4). MINECLIP is multi-task by design, as it is trained on open-vocabulary and diverse English transcripts.

During RL training, MINECLIP provides a high-quality reward signal *without* any domain adaptation techniques, despite the domain gap between noisy YouTube videos and clean simulator-rendered frames. MINECLIP eliminates the need to manually engineer reward functions for each and every MINEDOJO task. For Creative tasks that lack a simple success criterion (Sec. 2.2), MINECLIP also serves the dual purpose of an **automatic evaluation metric** that agrees well with human judgement on a subset of tasks we investigate (Sec. 4.2, Table 2). Because the learned reward model incurs a non-trivial computational overhead, we introduce several techniques to significantly improve RL training efficiency, making MINECLIP a practical module for open-ended agent learning in Minecraft (Sec. 4.2).

4.1 Pre-Training MINECLIP on Large-scale Videos

Formally, the learned reward function can be defined as $\Phi_{\mathcal{R}} : (G, V) \to \mathbb{R}$ that maps a language goal G and a video snippet V to a scalar reward. An ideal $\Phi_{\mathcal{R}}$ should return a high reward if the behavior depicted in the video faithfully follows the language description, and a low reward otherwise. This can be achieved by optimizing the InfoNCE objective [95, 40, 17], which learns to correlate positive video and text pairs [90, 5, 60, 4, 103].

Similar to the image-text CLIP model [69], MINECLIP is composed of a separate text encoder ϕ_G that embeds a language goal and a video encoder ϕ_V that embeds a moving window of 16 consecutive frames with 160×256 resolution (Fig. 4). Our neural architecture has a similar design as CLIP4Clip [58], where ϕ_G reuses OpenAI CLIP's pretrained text encoder, and ϕ_V is factorized into a frame-wise image encoder ϕ_I and a temporal aggregator ϕ_a that summarizes the sequence of 16 image features into a single video embedding. Unlike CLIP4Clip, we insert two extra layers of residual CLIP Adapter [29] after the aggregator ϕ_a to produce a better video feature, and finetune *only* the last two layers of the pretrained ϕ_I and ϕ_G.

From the MINEDOJO YouTube database, we follow the procedure in VideoCLIP [103] to sample 640K pairs of 16-second video snippets and time-aligned English transcripts, after applying a keyword filter. We train two MINECLIP variants with different types of aggregator ϕ_a: (1) MINECLIP[avg] does simple average pooling, which is fast but agnostic to the temporal ordering; (2) MINECLIP[attn] encodes the sequence by two transformer layers, which is relatively slower but captures more temporal

Table 2: MINECLIP agrees well with the ground-truth human judgment on the Creative tasks we consider. Numbers are F1 scores between MINECLIP's binary classification of tasks success and human labels (scaled to the percentage for better readability).

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

information, and thus produces a better reward signal in general. Details of data preprocessing, architecture, and hyperparameters are in the supplementary.

4.2 RL with MINECLIP Reward

We train a language-conditioned policy network that takes as input raw pixels and predicts discrete control. The policy is trained with PPO [76] on the MINECLIP rewards. In each episode, the agent is prompted with a language goal and takes a sequence of actions to fulfill this goal. When calculating the MINECLIP rewards, we concatenate the agent's latest 16 egocentric RGB frames in a temporal window to form a video snippet. MINECLIP handles all task prompts *zero-shot* without any further finetuning. In our experiments (Sec. 5), we show that MINECLIP provides effective dense rewards out of the box, despite the domain shift between in-the-wild YouTube frames and simulator frames. Besides regular video data augmentation, we do not employ any special domain adaptation methods during pre-training. Our finding is consistent with CLIP's strong zero-shot performances on robustness benchmarks in object recognition [69].

Compared to hard-coded reward functions in popular benchmarks [110, 91, 26], the MINECLIP model has 150M parameters and is thus much more expensive to query. We make several design choices to greatly accelerate RL training with MINECLIP in the loop: 1) The language goal G is fixed for a specific task, so the **text features ϕ_G can be precomputed** to avoid invoking the text encoder repeatedly; 2) Our agent's **RGB encoder reuses the pre-trained weights of ϕ_I** from MINECLIP. We do not finetune ϕ_I during RL training, which saves computation and endows the agent with good visual representations from the beginning; 3) MINECLIP's video encoder ϕ_V is factorized into an image encoder ϕ_I and a light-weight aggregator ϕ_a. This design choice enables **efficient image feature caching**. Consider two overlapping video sequences of 8 frames, `V[0:8]` and `V[1:9]`. We can cache the image features of the 7 overlapping frames `V[1]` to `V[7]` to maximize compute savings. If ϕ_V is a monolithic model like S3D [102] in VideoCLIP [103], then the video features from every sliding window must be recomputed, which would incur a much higher cost per time step; and 4) We leverage **Self-Imitation Learning** [65] to store the trajectories with high MINECLIP reward values in a buffer, and alternate between PPO and self-imitation gradient steps. It further improves sample efficiency as shown in the supplementary materials.

5 Experiments

We evaluate our agent-learning approach (Section 4) on 8 Programmatic tasks and 4 Creative tasks from the MINEDOJO benchmarking suite. We select these 12 tasks due to the diversity of skills required to solve them (e.g., harvesting, combat, building, navigation) and domain-specific entities (e.g., animals, resources, monsters, terrains, and structures). We split the tasks into 3 groups and train one multi-task agent for each group: `Animal-Zoo` (4 Programmatic tasks on hunting or harvesting resource from animals), `Mob-Combat` (Programmatic, fight 4 types of hostile monsters), and `Creative` (4 tasks).

In the experiments, we empirically check the quality of MINECLIP against manually written reward functions, and quantify how different variants of our learned model affect the RL performance. Table 1 presents our main results. Policy networks of all methods share the same architecture and are trained by PPO + Self-Imitation (Sec. 4.2). We compare the following methods: 1) **Ours (Attn)**: our agent trained with the MINECLIP[attn] reward model. For Programmatic tasks, we also add the final success condition as a binary reward. For Creative tasks, MINECLIP is the only source of reward; 2) **Ours (Avg)**: the average-pooling variant; 3) **Manual Reward**: hand-engineered dense reward using

Table 3: MINECLIP agents have stronger zero-shot visual generalization ability to unseen terrains, weathers, and lighting. Numbers outside parentheses are percentage success rates averaged over 3 seeds (each tested for 200 episodes), while those inside parentheses are relative performance changes.

	Tasks	**Ours** (Attn), train	**Ours** (Attn), unseen test	$\text{CLIP}_{\text{OpenAI}}$, train	$\text{CLIP}_{\text{OpenAI}}$, unseen test
	Milk Cow	64.5 ± 37.1	$\mathbf{64.8 \pm 31.3}(+\ \mathbf{0.8}\%)$	90.0 ± 0.4	$29.2 \pm 3.7\ (-67.6\%)$
	Hunt Cow	83.5 ± 7.1	$\mathbf{55.9 \pm 7.2}\ (\mathbf{-32.9}\%)$	72.7 ± 3.5	$16.7 \pm 1.6\ (-77.0\%)$
	Combat Spider	80.5 ± 13.0	$\mathbf{62.1 \pm 29.7}(\mathbf{-22.9}\%)$	79.5 ± 2.5	$54.2 \pm 9.6\ (-31.8\%)$
	Combat Zombie	47.3 ± 10.6	$\mathbf{39.9 \pm 25.3}(\mathbf{-15.4}\%)$	50.2 ± 7.5	$30.8 \pm 14.4(-38.6\%)$

ground-truth simulator states; 4) **Sparse-only**: the final binary success as a single sparse reward. Note that neither sparse-only nor manual reward is available for Creative tasks; and 5) **$\text{CLIP}_{\text{OpenAI}}$**: pre-trained OpenAI CLIP model that has **not** been finetuned on any MINEDOJO videos. All RL training details are presented in the supplementary. Fig. 2 visualizes the learned agent behavior in 4 of the considered tasks.

MINECLIP is competitive with manual reward. For Programmatic tasks (first 8 rows), RL agents guided by MINECLIP achieve competitive performance as those trained by manual reward. In three of the tasks, they even *outperform* the hand-engineered reward functions, which rely on privileged simulator states unavailable to MINECLIP. For a more statistically sound analysis, we conduct the Paired Student's *t*-test to compare the mean success rate of each task (pairing column 3 "Ours (Attn)" and column 5 "Manual Reward" in Table 1). The test yields p-value $0.3991 \gg 0.05$, which indicates that the difference between our method and manual reward is not considered statistically significant, and therefore they are comparable with each other. Because all tasks require nontrivial exploration, our approach also dominates the Sparse-only baseline. Note that the original OpenAI CLIP model fails to achieve any success. We hypothesize that the creatures in Minecraft look dramatically different from their real-world counterparts, which causes CLIP to produce *misleading* signals worse than no shaping reward at all. It implies the importance of finetuning on MINEDOJO's YouTube data.

MINECLIP provides automated evaluation. For Creative tasks (last 4 rows), there are no programmatic success criteria available. We convert MINECLIP into a binary success classifier by thresholding the reward value it outputs for an episode. To test the quality of MINECLIP as an automatic evaluation metric, we ask human judges to curate a dataset of 100 successful and 100 failed trajectories for each task. We then run both MINECLIP variants and $\text{CLIP}_{\text{OpenAI}}$ on the dataset and report the binary F1 score of their judgement against human ground-truth in Table 2. The results demonstrate that both MINECLIP[attn] and MINECLIP[avg] attain a very high degree of agreement with human evaluation results on this subset of the Creative task suite that we investigate. $\text{CLIP}_{\text{OpenAI}}$ baseline also achieves nontrivial agreement on Find Ocean and Dig Hole tasks, likely because real-world oceans and holes have similar texture. We use the `attn` variant as an automated success criterion to score the 4 Creative task results in Table 1. Our proposed method consistently learns better than $\text{CLIP}_{\text{OpenAI}}$-guided agents. It shows that MINECLIP is an effective approach to solving open-ended tasks when no straightforward reward signal is available. We provide further analysis beyond these 4 tasks in the supplementary.

MINECLIP shows good zero-shot generalization to significant visual distribution shift. We evaluate the learned policy without finetuning on a combination of unseen weather, lighting conditions, and terrains — 27 scenarios in total. For the baseline, we train agents with the original $\text{CLIP}_{\text{OpenAI}}$ image encoder (not trained on our YouTube videos) by imitation learning. The robustness against visual shift can be quantitatively measured by the relative performance degradation on novel test scenarios for each task. Table 3 shows that while all methods incur performance drops, agents with MINECLIP encoder is more robust to visual changes than the baseline across all tasks. Pre-training on diverse in-the-wild YouTube videos is important to boosting zero-shot visual generalization capability, a finding consistent with literature [69, 64].

6 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

[93, 98, 48], purely text-based game [54], grid worlds [18, 14], browser/GUI-based environments [81, 94], and indoor simulators for robotics [1, 80, 87, 26, 83, 74, 67]. 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. Prior methods in open-ended agent learning [25, 44, 99, 50, 22] do not make use of external knowledge, but our approach leverages internet-scale database to learn open-vocabulary reward models, thanks to Minecraft's abundance of gameplay data online.

Minecraft for AI Research. The Malmo platform [47] is the first comprehensive release of a Gym-style agent API [12] for Minecraft. Based on Malmo, MineRL [36] provides a codebase and human play trajectories for the annual Diamond Challenge at NeurIPS [35, 37, 49]. 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 [33] and IGLU [52] study agents with interactive dialogues. BASALT [78] applies human evaluation to 4 open-ended tasks. EvoCraft [34] is designed for structure building, and Crafter [38] 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. We include a feature comparison table of different Minecraft platforms for AI research in the supplementary.

Internet-scale Multimodal Knowledge Bases. Big dataset such as Common Crawl [20], the Pile [28], LAION [75], YouTube-8M [2] and HowTo100M [59] have been fueling the success of large pre-trained language models [23, 68, 13] and multimodal models [90, 5, 60, 109, 6, 4, 103]. While generally useful for learning representations, these datasets are not specifically targeted at embodied agents. To provide agent-centric training data, RoboNet [21] collects video frames from 7 robot platforms, and Ego4D [32] 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.

Embodied Agents with Large-scale Pre-training. Inspired by the success in NLP, embodied agent research [24, 10, 70, 19] has seen a surge in adoption of the large-scale pre-training paradigm. The recent advances can be roughly divided into 4 categories. 1) **Novel agent architecture**: Decision Transformer [16, 45, 108] applies self-attention to sequential decision making. GATO [71] and Unified-IO [57] learn a single model to accommodate different control interfaces. VIMA [46] unifies a wide range of robot manipulation tasks with multimodal prompting. 2) **Pre-training for better representations**: R3M [64] trains a general-purpose visual encoder for robot perception on Ego4D videos [32]. CLIPort [84] leverages the pre-trained CLIP model [69] to enable free-form language instructions for robot manipulation. 3) **Pre-training for better policies**: AlphaStar [96] achieves champion-level performance on StarCraft by imitating from numerous human demos. SayCan [3] leverages large language models (LMs) to ground value functions in the physical world. [56] and [72] directly reuse pre-trained LMs as policy backbone. VPT [9] is a concurrent work that learns an inverse dynamics model from human contractors to pseudo-label YouTube videos for behavior cloning. VPT is complementary to our approach, and can be finetuned to solve language-conditioned open-ended tasks with our learned reward model. 4) **Data-driven reward functions**: Concept2Robot [79] and DVD [15] learn a binary classifier to score behaviors from in-the-wild videos [31]. LOReL [63] crowd-sources humans labels to train language-conditioned reward function for offline RL. AVID [86] and XIRL [106] extract reward signals via cycle consistency. MINEDOJO's task benchmark and internet knowledge base are generally useful for developing new algorithms in all the above categories. In Sec. 4, we propose an open-vocabulary, multi-task reward model using MINEDOJO YouTube videos.

7 Conclusion

In this work, we introduce the MINEDOJO framework for developing generally capable embodied agents. MINEDOJO features a benchmarking suite of thousands of Programmatic and Creative tasks, and an internet-scale multimodal knowledge base of videos, wiki, and forum discussions. As an example of the novel research possibilities enabled by MINEDOJO, we propose MINECLIP as an effective language-conditioned reward function trained with in-the-wild YouTube videos. MINECLIP achieves strong performance empirically and agrees well with human evaluation results, making it a good automatic metric for Creative tasks. We look forward to seeing how MINEDOJO empowers the community to make progress on the important challenge of open-ended agent learning.

References

[1] Josh Abramson, Arun Ahuja, Iain Barr, Arthur Brussee, Federico Carnevale, Mary Cassin, Rachita Chhaparia, Stephen Clark, Bogdan Damoc, Andrew Dudzik, Petko Georgiev, Aurelia Guy, Tim Harley, Felix Hill, Alden Hung, Zachary Kenton, Jessica Landon, Timothy Lillicrap, Kory Mathewson, Soňa Mokrá, Alistair Muldal, Adam Santoro, Nikolay Savinov, Vikrant Varma, Greg Wayne, Duncan Williams, Nathaniel Wong, Chen Yan, and Rui Zhu. Imitating interactive intelligence. *arXiv preprint arXiv: Arxiv-2012.05672*, 2020.

[2] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv: Arxiv-1609.08675*, 2016.

[3] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, and Mengyuan Yan. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv: Arxiv-2204.01691*, 2022.

[4] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *arXiv preprint arXiv: Arxiv-2104.11178*, 2021.

[5] Jean-Baptiste Alayrac, Adrià Recasens, Rosalia Schneider, Relja Arandjelovic, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/0060ef47b12160b9198302ebdb144dcf-Abstract.html`.

[6] Elad Amrani, Rami Ben-Ari, Daniel Rotman, and Alex M. Bronstein. Noise estimation using density estimation for self-supervised multimodal learning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 6644–6652. AAAI Press, 2021. URL `https://ojs.aaai.org/index.php/AAAI/article/view/16822`.

[7] Jacob Andreas, Dan Klein, and Sergey Levine. Modular multitask reinforcement learning with policy sketches. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 166–175. PMLR, 2017. URL `http://proceedings.mlr.press/v70/andreas17a.html`.

[8] Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R. Manmatha. Docformer: End-to-end transformer for document understanding. *arXiv preprint arXiv: Arxiv-2106.11539*, 2021.

[9] Bowen Baker, Ilge Akkaya, Peter Zhokhov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *arXiv preprint arXiv: Arxiv-2206.11795*, 2022.

[10] Dhruv Batra, Angel X. Chang, Sonia Chernova, Andrew J. Davison, Jia Deng, Vladlen Koltun, Sergey Levine, Jitendra Malik, Igor Mordatch, Roozbeh Mottaghi, Manolis Savva, and Hao Su. Rearrangement: A challenge for embodied ai. *arXiv preprint arXiv: Arxiv-2011.01975*, 2020.

[11] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik

Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models. *arXiv preprint arXiv: Arxiv-2108.07258*, 2021.

[12] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv: Arxiv-1606.01540*, 2016.

[13] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[14] Tianshi Cao, Jingkang Wang, Yining Zhang, and Sivabalan Manivasagam. Babyai++: Towards grounded-language learning beyond memorization. *arXiv preprint arXiv: Arxiv-2004.07200*, 2020.

[15] Annie S. Chen, Suraj Nair, and Chelsea Finn. Learning generalizable robotic reward functions from in-the-wild human videos. In Dylan A. Shell, Marc Toussaint, and M. Ani Hsieh, editors, *Robotics: Science and Systems XVII, Virtual Event, July 12-16, 2021*, 2021. doi: 10.15607/RSS.2021.XVII.012. URL https://doi.org/10.15607/RSS.2021.XVII.012.

[16] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 15084–15097, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/7f489f642a0ddb10272b5c31057f0663-Abstract.html.

[17] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 2020. URL http://proceedings.mlr.press/v119/chen20j.html.

[18] Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. Babyai: A platform to study the sample efficiency of grounded language learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=rJeXCo0cYX.

[19] Jack Collins, Shelvin Chand, Anthony Vanderkop, and David Howard. A review of physics simulators for robotic applications. *IEEE Access*, 9:51416–51431, 2021.

[20] Common Crawl. Common crawl. https://commoncrawl.org/, 2012. Accessed: 2022-06-06.

[21] Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, volume 100 of *Proceedings of Machine Learning Research*, pages 885–897. PMLR, 2019. URL http://proceedings.mlr.press/v100/dasari20a.html.

[22] Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre M. Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/985e9a46e10005356bbaf194249f6856-Abstract.html.

[23] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv: Arxiv-1810.04805*, 2018.

[24] Jiafei Duan, Samson Yu, Hui Li Tan, Hongyuan Zhu, and Cheston Tan. A survey of embodied AI: from simulators to research tasks. *IEEE Trans. Emerg. Top. Comput. Intell.*, 6(2):230–244, 2022. doi: 10.1109/TETCI.2022.3141105. URL https://doi.org/10.1109/TETCI.2022.3141105.

[25] Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. Go-explore: a new approach for hard-exploration problems. *arXiv preprint arXiv: Arxiv-1901.10995*, 2019.

[26] Linxi Fan, Guanzhi Wang, De-An Huang, Zhiding Yu, Li Fei-Fei, Yuke Zhu, and Anima Anandkumar. Secant: Self-expert cloning for zero-shot generalization of visual policies. *arXiv preprint arXiv: Arxiv-2106.09678*, 2021.

[27] Florian Fuchs, Yunlong Song, Elia Kaufmann, Davide Scaramuzza, and Peter Dürr. Super-human performance in gran turismo sport using deep reinforcement learning. *IEEE Robotics Autom. Lett.*, 6(3):4257–4264, 2021. doi: 10.1109/LRA.2021.3064284. URL https://doi.org/10.1109/LRA.2021.3064284.

[28] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[29] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv: Arxiv-2110.04544*, 2021.

[30] Jordan Gerblick. Minecraft, the most-watched game on youtube, passes 1 trillion views, Dec 2021. URL https://www.gamesradar.com/minecraft-the-most-watched-game-on-youtube-passes-1-trillion-views/.

[31] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017.

[32] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina Gonzalez,

James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3,000 hours of egocentric video. *arXiv preprint arXiv: Arxiv-2110.07058*, 2021.

[33] Jonathan Gray, Kavya Srinet, Yacine Jernite, Haonan Yu, Zhuoyuan Chen, Demi Guo, Siddharth Goyal, C. Lawrence Zitnick, and Arthur Szlam. Craftassist: A framework for dialogue-enabled interactive agents. *arXiv preprint arXiv: Arxiv-1907.08584*, 2019.

[34] Djordje Grbic, Rasmus Berg Palm, Elias Najarro, Claire Glanois, and Sebastian Risi. *EvoCraft: A New Challenge for Open-Endedness*, pages 325–340. Springer International Publishing, 2021. doi: 10.1007/978-3-030-72699-7_21. URL `http://link.springer.com/content/pdf/10.1007/978-3-030-72699-7_21`.

[35] William H. Guss, Cayden Codel, Katja Hofmann, Brandon Houghton, Noboru Kuno, Stephanie Milani, Sharada Mohanty, Diego Perez Liebana, Ruslan Salakhutdinov, Nicholay Topin, Manuela Veloso, and Phillip Wang. The minerl 2019 competition on sample efficient reinforcement learning using human priors. *arXiv preprint arXiv: Arxiv-1904.10079*, 2019.

[36] William H. Guss, Brandon Houghton, Nicholay Topin, Phillip Wang, Cayden Codel, Manuela Veloso, and Ruslan Salakhutdinov. Minerl: A large-scale dataset of minecraft demonstrations. *arXiv preprint arXiv: Arxiv-1907.13440*, 2019.

[37] William H. Guss, Mario Ynocente Castro, Sam Devlin, Brandon Houghton, Noboru Sean Kuno, Crissman Loomis, Stephanie Milani, Sharada Mohanty, Keisuke Nakata, Ruslan Salakhutdinov, John Schulman, Shinya Shiroshita, Nicholay Topin, Avinash Ummadisingu, and Oriol Vinyals. The minerl 2020 competition on sample efficient reinforcement learning using human priors. *arXiv preprint arXiv: Arxiv-2101.11071*, 2021.

[38] Danijar Hafner. Benchmarking the spectrum of agent capabilities. *arXiv preprint arXiv: Arxiv-2109.06780*, 2021.

[39] Laura Hanu and Unitary team. Detoxify. Github. `https://github.com/unitaryai/detoxify`, 2020.

[40] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9726–9735. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.00975. URL `https://doi.org/10.1109/CVPR42600.2020.00975`.

[41] Matthew Henderson, Paweł Budzianowski, Iñigo Casanueva, Sam Coope, Daniela Gerz, Girish Kumar, Nikola Mrkšić, Georgios Spithourakis, Pei-Hao Su, Ivan Vulić, and Tsung-Hsien Wen. A repository of conversational datasets. *arXiv preprint arXiv: Arxiv-1904.06472*, 2019.

[42] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6626–6637, 2017. URL `https://proceedings.neurips.cc/paper/2017/hash/8a1d694707eb0fefe65871369074926d-Abstract.html`.

[43] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas

Jackson, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv: Arxiv-2207.05608*, 2022.

[44] Joost Huizinga and Jeff Clune. Evolving multimodal robot behavior via many stepping stones with the combinatorial multiobjective evolutionary algorithm. *Evolutionary computation*, 30 (2):131–164, 2022.

[45] Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 1273–1286, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/099fe6b0b444c23836c4a5d07346082b-Abstract.html.

[46] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts. *arXiv preprint arXiv: Arxiv-2210.03094*, 2022.

[47] Matthew Johnson, Katja Hofmann, Tim Hutton, and David Bignell. The malmo platform for artificial intelligence experimentation. *IJCAI*, 2016. URL https://dl.acm.org/doi/10.5555/3061053.3061259.

[48] Arthur Juliani, Ahmed Khalifa, Vincent-Pierre Berges, Jonathan Harper, Ervin Teng, Hunter Henry, Adam Crespi, Julian Togelius, and Danny Lange. Obstacle tower: A generalization challenge in vision, control, and planning. In Sarit Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 2684–2691. ijcai.org, 2019. doi: 10.24963/ijcai.2019/373. URL https://doi.org/10.24963/ijcai.2019/373.

[49] Anssi Kanervisto, Stephanie Milani, Karolis Ramanauskas, Nicholay Topin, Zichuan Lin, Junyou Li, Jianing Shi, Deheng Ye, Qiang Fu, Wei Yang, Weijun Hong, Zhongyue Huang, Haicheng Chen, Guangjun Zeng, Yue Lin, Vincent Micheli, Eloi Alonso, François Fleuret, Alexander Nikulin, Yury Belousov, Oleg Svidchenko, and Aleksei Shpilman. Minerl diamond 2021 competition: Overview, results, and lessons learned. *arXiv preprint arXiv: Arxiv-2202.10583*, 2022.

[50] Ingmar Kanitscheider, Joost Huizinga, David Farhi, William Hebgen Guss, Brandon Houghton, Raul Sampedro, Peter Zhokhov, Bowen Baker, Adrien Ecoffet, Jie Tang, Oleg Klimov, and Jeff Clune. Multi-task curriculum learning in a complex, visual, hard-exploration domain: Minecraft. *arXiv preprint arXiv: Arxiv-2106.14876*, 2021.

[51] Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. Abstractive summarization of reddit posts with multi-level memory networks. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2519–2531. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1260. URL https://doi.org/10.18653/v1/n19-1260.

[52] Julia Kiseleva, Ziming Li, Mohammad Aliannejadi, Shrestha Mohanty, Maartje ter Hoeve, Mikhail Burtsev, Alexey Skrynnik, Artem Zholus, Aleksandr Panov, Kavya Srinet, Arthur Szlam, Yuxuan Sun, Katja Hofmann, Michel Galley, and Ahmed Awadallah. Neurips 2021 competition iglu: Interactive grounded language understanding in a collaborative environment. *arXiv preprint arXiv: Arxiv-2110.06536*, 2021.

[53] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv: Arxiv-1712.05474*, 2017.

[54] Heinrich Küttler, Nantas Nardelli, Alexander Miller, Roberta Raileanu, Marco Selvatici, Edward Grefenstette, and Tim Rocktäschel. The nethack learning environment. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7671–7684. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/569ff987c643b4bedf504efda8f786c2-Paper.pdf.

[55] WB Langdon. Pfeiffer–a distributed open-ended evolutionary system. In *AISB*, volume 5, pages 7–13. Citeseer, 2005.

[56] Shuang Li, Xavier Puig, Chris Paxton, Yilun Du, Clinton Wang, Linxi Fan, Tao Chen, De-An Huang, Ekin Akyürek, Anima Anandkumar, Jacob Andreas, Igor Mordatch, Antonio Torralba, and Yuke Zhu. Pre-trained language models for interactive decision-making. *arXiv preprint arXiv: Arxiv-2202.01771*, 2022.

[57] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. *arXiv preprint arXiv: Arxiv-2206.08916*, 2022.

[58] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval. *arXiv preprint arXiv: Arxiv-2104.08860*, 2021.

[59] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. *arXiv preprint arXiv: Arxiv-1906.03327*, 2019.

[60] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9876–9886. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.00990. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Miech_End-to-End_Learning_of_Visual_Representations_From_Uncurated_Instructional_Videos_CVPR_2020_paper.html.

[61] Minecraft Wiki. Minecraft wiki. hhttps://minecraft.fandom.com/wiki/Minecraft_Wiki, 2016. Accessed: 2022-06-06.

[62] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv: Arxiv-1312.5602*, 2013.

[63] Suraj Nair, Eric Mitchell, Kevin Chen, Brian Ichter, Silvio Savarese, and Chelsea Finn. Learning language-conditioned robot behavior from offline data and crowd-sourced annotation. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *Conference on Robot Learning, 8-11 November 2021, London, UK*, volume 164 of *Proceedings of Machine Learning Research*, pages 1303–1315. PMLR, 2021. URL https://proceedings.mlr.press/v164/nair22a.html.

[64] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv: Arxiv-2203.12601*, 2022.

[65] Junhyuk Oh, Yijie Guo, Satinder Singh, and Honglak Lee. Self-imitation learning. In *International Conference on Machine Learning*, pages 3878–3887. PMLR, 2018.

[66] OpenAI, :, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique P. d. O. Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv: Arxiv-1912.06680*, 2019.

[67] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8494–8502. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00886. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Puig_VirtualHome_Simulating_Household_CVPR_2018_paper.html.

[68] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[69] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[70] Harish Ravichandar, Athanasios S Polydoros, Sonia Chernova, and Aude Billard. Recent advances in robot learning from demonstration. *Annual review of control, robotics, and autonomous systems*, 3:297–330, 2020.

[71] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent. *arXiv preprint arXiv: Arxiv-2205.06175*, 2022.

[72] Machel Reid, Yutaro Yamada, and Shixiang Shane Gu. Can wikipedia help offline reinforcement learning? *arXiv preprint arXiv: Arxiv-2201.12122*, 2022.

[73] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2226–2234, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/8a3363abe792db2d8761d6403605aeb7-Abstract.html.

[74] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[75] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

[76] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv: Arxiv-1707.06347*, 2017.

[77] Selenium WebDriver. Selenium webdriver. https://www.selenium.dev/, 2011. Accessed: 2022-06-06.

[78] Rohin Shah, Cody Wild, Steven H. Wang, Neel Alex, Brandon Houghton, William Guss, Sharada Mohanty, Anssi Kanervisto, Stephanie Milani, Nicholay Topin, Pieter Abbeel, Stuart Russell, and Anca Dragan. The minerl basalt competition on learning from human feedback. *arXiv preprint arXiv: Arxiv-2107.01969*, 2021.

[79] Lin Shao, Toki Migimatsu, Qiang Zhang, Karen Yang, and Jeannette Bohg. Concept2robot: Learning manipulation concepts from instructions and human demonstrations. *The International Journal of Robotics Research*, 40(12-14):1419–1434, 2021.

[80] Bokui Shen, Fei Xia, Chengshu Li, Roberto Martín-Martín, Linxi Fan, Guanzhi Wang, Claudia Pérez-D'Arpino, Shyamal Buch, Sanjana Srivastava, Lyne P. Tchapmi, Micael E. Tchapmi,

Kent Vainio, Josiah Wong, Li Fei-Fei, and Silvio Savarese. igibson 1.0: a simulation environment for interactive tasks in large realistic scenes. *arXiv preprint arXiv: Arxiv-2012.02924*, 2020.

[81] Tianlin Tim Shi, Andrej Karpathy, Linxi Jim Fan, Jonathan Hernandez, and Percy Liang. World of bits: an open-domain platform for web-based agents. *ICML*, 2017. URL `https://dl.acm.org/doi/10.5555/3305890.3306005`.

[82] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

[83] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. ALFRED: A benchmark for interpreting grounded instructions for everyday tasks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10737–10746. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.01075. URL `https://openaccess.thecvf.com/content_CVPR_2020/html/Shridhar_ALFRED_A_Benchmark_for_Interpreting_Grounded_Instructions_for_Everyday_Tasks_CVPR_2020_paper.html`.

[84] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. *arXiv preprint arXiv: Arxiv-2109.12098*, 2021.

[85] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv: Arxiv-1712.01815*, 2017.

[86] Laura Smith, Nikita Dhawan, Marvin Zhang, Pieter Abbeel, and Sergey Levine. Avid: Learning multi-stage tasks via pixel-level translation of human videos. *arXiv preprint arXiv: Arxiv-1912.04443*, 2019.

[87] Sanjana Srivastava, Chengshu Li, Michael Lingelbach, Roberto Martín-Martín, Fei Xia, Kent Elliott Vainio, Zheng Lian, Cem Gokmen, Shyamal Buch, C. Karen Liu, Silvio Savarese, Hyowon Gweon, Jiajun Wu, and Li Fei-Fei. BEHAVIOR: benchmark for everyday household activities in virtual, interactive, and ecological environments. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *Conference on Robot Learning, 8-11 November 2021, London, UK*, volume 164 of *Proceedings of Machine Learning Research*, pages 477–490. PMLR, 2021. URL `https://proceedings.mlr.press/v164/srivastava22a.html`.

[88] Russell K Standish. Open-ended artificial evolution. *International Journal of Computational Intelligence and Applications*, 3(02):167–175, 2003.

[89] Kenneth O Stanley, Joel Lehman, and Lisa Soros. Open-endedness: The last grand challenge you've never heard of. *O'Reilly Online,*, 2017.

[90] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning video representations using contrastive bidirectional transformer. *arXiv preprint arXiv: Arxiv-1906.05743*, 2019.

[91] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. Deepmind control suite. *arXiv preprint arXiv: Arxiv-1801.00690*, 2018.

[92] Tim Taylor, Mark Bedau, Alastair Channon, David Ackley, Wolfgang Banzhaf, Guillaume Beslon, Emily Dolson, Tom Froese, Simon Hickinbotham, Takashi Ikegami, et al. Open-ended evolution: Perspectives from the oee workshop in york. *Artificial life*, 22(3):408–423, 2016.

[93] Open Ended Learning Team, Adam Stooke, Anuj Mahajan, Catarina Barros, Charlie Deck, Jakob Bauer, Jakub Sygnowski, Maja Trebacz, Max Jaderberg, Michael Mathieu, Nat McAleese, Nathalie Bradley-Schmieg, Nathaniel Wong, Nicolas Porcel, Roberta Raileanu, Steph Hughes-Fitt, Valentin Dalibard, and Wojciech Marian Czarnecki. Open-ended learning leads to generally capable agents. *arXiv preprint arXiv: Arxiv-2107.12808*, 2021.

[94] Daniel Toyama, Philippe Hamel, Anita Gergely, Gheorghe Comanici, Amelia Glaese, Zafarali Ahmed, Tyler Jackson, Shibl Mourad, and Doina Precup. Androidenv: A reinforcement learning platform for android. *arXiv preprint arXiv: Arxiv-2105.13231*, 2021.

[95] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv: Arxiv-1807.03748*, 2018.

[96] Oriol Vinyals, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, Wojciech M Czarnecki, Andrew Dudzik, Aja Huang, Petko Georgiev, Richard Powell, et al. Alphastar: Mastering the real-time strategy game starcraft ii. *DeepMind blog*, 2, 2019.

[97] Michael Vølske, Martin Potthast, Shahbaz Syed, and Benno Stein. TL;DR: Mining Reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63, Copenhagen, Denmark, sep 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4508. URL `https://aclanthology.org/W17-4508`.

[98] Rui Wang, Joel Lehman, Jeff Clune, and Kenneth O. Stanley. Paired open-ended trailblazer (poet): Endlessly generating increasingly complex and diverse learning environments and their solutions. *arXiv preprint arXiv: Arxiv-1901.01753*, 2019.

[99] Rui Wang, Joel Lehman, Aditya Rawal, Jiale Zhi, Yulun Li, Jeff Clune, and Kenneth O. Stanley. Enhanced poet: Open-ended reinforcement learning through unbounded invention of learning challenges and their solutions. *arXiv preprint arXiv: Arxiv-2003.08536*, 2020.

[100] Wikipedia contributors. Minecraft — Wikipedia, the free encyclopedia, 2022. URL `https://en.wikipedia.org/w/index.php?title=Minecraft&oldid=1092238294`. [Online; accessed 9-June-2022].

[101] Fei Xia, William B. Shen, Chengshu Li, Priya Kasimbeg, Micael Tchapmi, Alexander Toshev, Li Fei-Fei, Roberto Martín-Martín, and Silvio Savarese. Interactive gibson benchmark (igibson 0.5): A benchmark for interactive navigation in cluttered environments. *arXiv preprint arXiv: Arxiv-1910.14442*, 2019.

[102] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 305–321, 2018.

[103] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. *arXiv preprint arXiv: Arxiv-2109.14084*, 2021.

[104] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. Layoutlmv2: Multimodal pre-training for visually-rich document understanding. *arXiv preprint arXiv: Arxiv-2012.14740*, 2020.

[105] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. *arXiv preprint arXiv: Arxiv-1912.13318*, 2019.

[106] Kevin Zakka, Andy Zeng, Pete Florence, Jonathan Tompson, Jeannette Bohg, and Debidatta Dwibedi. Xirl: Cross-embodiment inverse reinforcement learning. *arXiv preprint arXiv: Arxiv-2106.03911*, 2021.

[107] Andy Zeng, Adrian Wong, Stefan Welker, Krzysztof Choromanski, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Pete Florence. Socratic models: Composing zero-shot multimodal reasoning with language. *arXiv preprint arXiv: Arxiv-2204.00598*, 2022.

[108] Qinqing Zheng, Amy Zhang, and Aditya Grover. Online decision transformer. *arXiv preprint arXiv: Arxiv-2202.05607*, 2022.

[109] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. *arXiv preprint arXiv: Arxiv-2011.07231*, 2020.

[110] Yuke Zhu, Josiah Wong, Ajay Mandlekar, and Roberto Martín-Martín. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv: Arxiv-2009.12293*, 2020.

8 Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See supplementary materials.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See supplementary materials.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See supplementary material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We report standard deviation across multiple runs. See Sec 5
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See supplementary material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] See the "License" part in our online submission.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] As a URL, see `https://minedojo.org`.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See supplementary material.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See supplementary material.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]