GROOT: LEARNING TO FOLLOW INSTRUCTIONS BY WATCHING GAMEPLAY VIDEOS

Shaofei Cai[1,2], **Bowei Zhang**[3], **Zihao Wang**[1,2], **Xiaojian Ma**[5], **Anji Liu**[4], **Yitao Liang**[1*]
Team CraftJarvis
[1]Institute for Artificial Intelligence, Peking University
[2]School of Intelligence Science and Technology, Peking University
[3]School of Electronics Engineering and Computer Science, Peking University
[4]Computer Science Department, University of California, Los Angeles
[5]Beijing Institute for General Artificial Intelligence (BIGAI)
{caishaofei,zhangbowei,zhwang}@stu.pku.edu.cn
xiaojian.ma@ucla.edu,liuanji@cs.ucla.edu,yitaol@pku.edu.cn
https://craftjarvis.github.io/GROOT

ABSTRACT

We study the problem of building an agent that can follow open-ended instructions in open-world environments. We propose to follow reference videos as instructions, which offer expressive goal specifications while eliminating the need for expensive text-gameplay annotations. We implement our agent GROOT in a simple yet effective encoder-decoder architecture based on causal transformers. We evaluate GROOT against open-world counterparts and human players on a proposed **Minecraft SkillForge** benchmark. The Elo ratings clearly show that GROOT is closing the human-machine gap as well as exhibiting a 70% winning rate over the best generalist agent baseline. Qualitative analysis of the induced goal space further demonstrates some interesting emergent properties, including the goal composition and complex gameplay behavior synthesis.

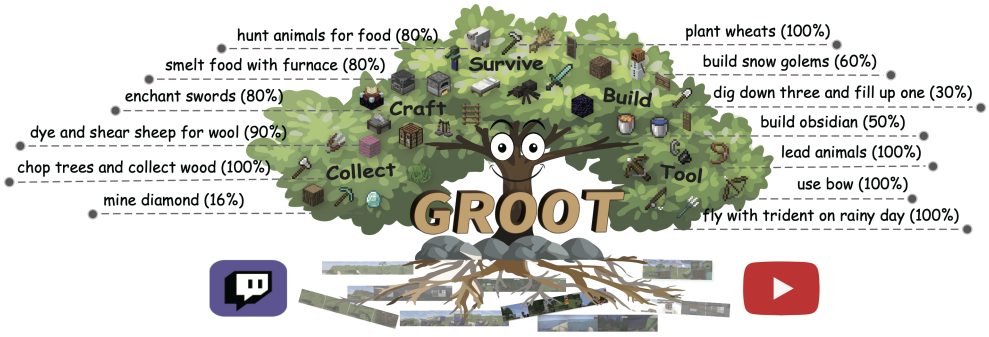


Figure 1: Through the cultivation of extensive gameplay videos, GROOT has grown a rich set of skill fruits (number denotes success rate; skills shown above do not mean to be exhaustive; kudos to the anonymous artist).

1 INTRODUCTION

Developing human-level embodied agents that can solve endless tasks in open-world environments, such as Minecraft (Johnson et al., 2016; Fan et al., 2022), has always been a long-term goal pursued in AI. Recent works have explored using Large Language Models (LLMs) to generate high-level plans, which guide the agent to accomplish challenging long-horizon tasks (Wang et al., 2023b;a; Zhu et al., 2023). However, a major gap between these LLM-based agents and generalist agents that can

*Corresponding Author.

complete endless amounts of tasks is the capability of their low-level controllers, which map the plans to motor commands. Recently developed controllers are only capable of completing a predefined and narrow set of programmatic tasks (Lin et al., 2021; Baker et al., 2022; Cai et al., 2023), which hinders LLM-based planning agents from unleashing their full potential. We attribute the limitation of these low-level controllers to how the goal is specified. Specifically, existing controllers use task indicator (Yu et al., 2019), future outcome (Chen et al., 2021; Lifshitz et al., 2023), and language (Brohan et al., 2022) to represent the goal. While it is easy to learn a controller with some of these goal specifications, they may not be expressive enough for diverse tasks. Taking future outcome goals as an example, an image of a desired house clearly lacks procedural information on how the house was built. One exception is language, but learning a controller that can receive language goals is prohibitively expensive as it requires a huge number of trajectory-text pairs with text that precisely depicts the full details of the gameplay, therefore preventing them from scaling up to more open-ended tasks.

Having observed the limitations of goal specification in the prior works, this paper seeks to find a balance between the capacity of goal specification and the cost of controller learning. Concretely, we propose to specify the goal as a reference gameplay video clip. While such video instruction is indeed expressive, there are two challenges: 1) How can the controller understand the actual goal being specified as the video itself can be ambiguous, i.e., a goal space or video instruction encoder has to be learned; 2) How to ultimately map such goal to actual motor commands? To this end, we introduce a learning framework that simultaneously produces a goal space and a video instruction following controller from gameplay videos. The fundamental idea is casting the problem as future state prediction based on past observations:

- The predicting model needs to identify which goal is being pursued from the past observations, which requires a good goal space (induced by a video instruction encoder);
- Since the transition dynamics model is fixed, a policy that maps both the state and the recognized goal to action is also needed by the predicting model when rolling the future state predictions.

Effectively, this results in the goal space and control policy we need. We introduce a variational learning objective for this problem, which leads to a combination of a cloning loss and a KL regularization loss. Based on this framework, we implement GROOT, an agent with an encoder-decoder architecture to solve open-ended Minecraft tasks by following video instructions. The video encoder is a non-causal transformer that extracts the semantic information expressed in the video and maps it to the latent goal space. The controller policy is a decoder module implemented by a causal transformer, which decodes the goal information in the latent space and translates it into a sequence of actions in the given environment states in an autoregressive manner.

To comprehensively evaluate an agent's mastery of skills, we designed a benchmark called **Minecraft SkillForge**. The benchmark covers six common Minecraft task groups: `collect`, `build`, `survive`, `explore`, `tool`, and `craft`, testing the agent's abilities in resource collection, structure building, environmental understanding, and tool usage, in a total of 30 tasks. We calculate Elo ratings among GROOT, several counterparts, and human players based on human evaluations. Our experiments showed that GROOT is closing the human-machine gap and outperforms the best baseline by 150 points (or 70% winning rate) in an Elo tournament system. Our qualitative analysis of the induced goal space further demonstrates some interesting emergent properties, including the goal composition and complex gameplay behavior synthesis.

To sum up, our main contributions are as follows:

- Start by maximizing the log-likelihood of future states given past ones, we have discovered the learning objectives that lead to a good goal space and ultimately the instruction-following controller from gameplay videos. It provides theoretical guidance for the agent architecture design and model optimization.
- Based on our proposed learning framework, we implemented a simple yet efficient encoder-decoder agent based on causal transformers. The encoder is responsible for understanding the goal information in the video instruction while the decoder as the policy emits motor commands.
- On our newly introduced benchmark, Minecraft SkillForge, GROOT is closing the human-machine gap and surpassing the state-of-the-art baselines by a large margin in the overall Elo rating comparison. GROOT also exhibits several interesting emergent properties, including goal composition and complex gameplay behavior synthesis.

2 PRELIMINARIES AND PROBLEM FORMULATION

Reinforcement Learning (RL) concerns the problem in which an agent interacts with an environment at discrete time steps, aiming to maximize its expected cumulative reward (Mnih et al., 2015; Schulman et al., 2017; Espeholt et al., 2018). Specifically, the environment is defined as a Markov Decision Process (MDP) $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, d_0 \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the transition dynamics, and d_0 is the initial state distribution. Our goal is to learn a policy $\pi(a|s)$ that maximizes the expected cumulative reward $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t]$, where $\gamma \in (0, 1]$ is a discount factor.

In goal-conditioned RL (GCRL) tasks, we are additionally provided with a goal $g \in \mathcal{G}$ (Andrychowicz et al., 2017; Ding et al., 2019; Liu et al., 2022; Cai et al., 2023; Jing et al., 2021; 2020; Yang et al., 2019). And the task becomes learning a goal-conditioned policy $\pi(a|s, g)$ that maximizes the expected return $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t^g]$, where r_t^g is the goal-specific reward achieved at time step t. Apart from being a new type of RL task, GCRL has been widely studied as a pre-training stage toward conquering more challenging environments/tasks (Aytar et al., 2018b; Baker et al., 2022; Zhang et al., 2022). Specifically, suppose we are provided with a good goal-condition policy, the goal can be viewed as a meta-action that drives the agent to accomplish various sub-tasks, which significantly simplifies tasks that require an extended horizon to accomplish. Further, when equipped with goal planners, we can achieve zero- or few-shot learning on compositional tasks that are beyond the reach of RL algorithms (Huang et al., 2022; Wang et al., 2023b;a; Zhu et al., 2023; Gong et al., 2023).

At the heart of leveraging such benefits, a key requirement is to have a properly-defined goal space that (i) has a wide coverage of common tasks/behaviors, and (ii) succinctly describes the task without including unnecessary information about the state. Many prior works establish goal spaces using guidance from other modalities such as language (Hong et al., 2020; Stone et al., 2023; Cai et al., 2023) or code (Wang et al., 2023a; Huang et al., 2023). While effective, the requirement on large-scale trajectory data paired with this auxiliary information could be hard to fulfill in practice. Instead, this paper studies the problem of simultaneously learning a rich and coherent goal space and the corresponding goal-conditioned policy, given a pre-trained inverse dynamic model and raw gameplay videos, i.e., sequences of states $\{s_{0:T}^{(i)}\}_i$ collected using unknown policies.

3 GOAL SPACE DISCOVERY VIA FUTURE STATE PREDICTION

This section explains our learning framework: discovering a "good" goal space as well as a video instruction following the controller through the task of predicting future states given previous ones. We start with an illustrative example in Minecraft (Johnson et al., 2016). Imagine that an agent is standing inside a grassland holding an axe that can be used to chop the tree in front of them. Suppose in the gameplay video, players either go straight to chop the tree or bypass it to explore the territory. In order to predict future frames, it is sufficient to know (i) which goal (chop tree or bypass tree) is being pursued by the agent, and (ii) what will happen if the agent chooses a particular option (i.e., transition dynamics). Apart from the latter information that is irrelevant to the past observations, we only need to capture the goal information, i.e., whether the agent decides to chop the tree or bypass the tree. Therefore, the task of establishing a comprehensive yet succinct goal space can be interpreted as predicting future states while conditioning on the transition dynamics of the environment.

Formally, our learning objective is to maximize the log-likelihood of future states given past ones: $\log p_\theta(s_{t+1:T}|s_{0:t})$. Define g as a latent variable conditioned on past states (think of it as the potential goals the agent is pursuing given past states), the evidence lower-bound of the objective given variational posterior $q_\phi(g|s_{0:T})$ is the following (see Appendix A for derivations):

$$\begin{aligned}\log p_\theta(s_{t+1:T}|s_{0:t}) &= \log \sum_g p_\theta(s_{t+1:T}, g|s_{0:t}) \\ &\geq \mathbb{E}_{g\sim q_\phi(\cdot|s_{0:T})} \left[\log p_\theta(s_{t+1:T}|s_{0:t}, g)\right] - D_{\mathrm{KL}}\left(q_\phi(g|s_{0:T}) \parallel p_\theta(g|s_{0:t})\right),\end{aligned}$$

where $D_{\mathrm{KL}}(\cdot\|\cdot)$ denotes the KL-divergence. Next, we break down the first term (i.e., $p_\theta(s_{t+1:T}|s_{0:t}, g)$) into components contributed by the (unknown) goal-conditioned policy $\pi_\theta(a|s, g)$

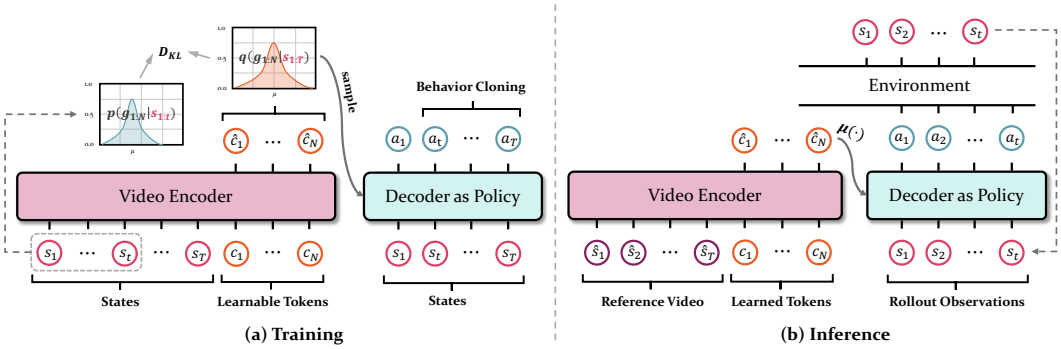


Figure 2: **GROOT agent architecture. Left:** In the training stage, a video encoder (non-causal transformer) learns to extract the semantic meaning and transfer the video (state sequence) into the goal embedding space. A goal-conditioned policy (causal transformer) is learned to predict actions following the given instructions. We learn the agent using behavior cloning under a KL constraint. **Right:** During inference, a reference video is passed into the encoder to generate the goal embeddings that drive the policy to interact with the environment.

and the transition dynamics $p_\theta(s_{t+1}|s_{0:t}, a_t)$:

$$\begin{aligned}\log p_\theta(s_{t+1:T}|s_{0:t}, g) &= \sum_{\tau=t}^{T} \log \sum_{a_\tau} \pi_\theta(a_\tau|s_{0:\tau}, g) \cdot p_\theta(s_{\tau+1}|s_{0:\tau}, a_\tau) \\ &\geq \sum_{\tau=t}^{T} \mathbb{E}_{a_\tau \sim p_\theta(a_\tau|s_{0:\tau+1})} \big[\log \pi_\theta(a_\tau|s_{0:\tau}, g) + C\big],\end{aligned}$$

where the constant C contains terms that depend solely on the environment dynamics and are irrelevant to what we want to learn (i.e., the goal space and the goal-conditioned policy). Bring it back to the original objective, we have

$$\log p(s_{t+1:T}|s_{0:t}) \geq \underbrace{\sum_{\tau=t}^{T-1} \mathbb{E}_{g\sim q_\phi(\cdot|s_{0:T}), a_\tau \sim p_\theta(\cdot|s_{0:\tau+1})} \left[\log \pi_\theta(a_\tau|s_{0:\tau}, g)\right]}_{\text{behaviour cloning}} - \underbrace{D_{\mathrm{KL}}\left(q_\phi(g|s_{0:T}) \parallel p_\theta(g|s_{0:t})\right)}_{\text{goal space constraint (KL regularization)}},$$

where $q_\phi(\cdot|s_{0:T})$ is implemented as a video encoder that maps the whole state sequence into the latent goal space. $p_\theta(\cdot|s_{0:\tau+1})$ is the inverse dynamic model (IDM) that predicts actions required to achieve a desired change in the states, which is usually a pre-trained model, details are in Appendix C. Thus, the objective can be explained as jointly learning a video encoder and a goal-controller policy through behavior cloning under succinct goal space constraints.

4 GROOT Architecture Design and Training Strategy

This section illustrates how to create an agent (we call it GROOT) that can understand the semantic meaning of a reference video and interact with the environment based on the aforementioned learning framework. According to the discussion in Section 3, the learnable parts of GROOT include the video encoder and the goal-conditioned policy. Recently, Transformer (Vaswani et al., 2017) has demonstrated effectiveness in solving sequential decision-making problems (Parisotto et al., 2019; Chen et al., 2021; Brohan et al., 2022). Motivated by this, we implement GROOT with transformer-based encoder-decoder architecture, as shown in Figure 2.

4.1 Video Encoder

The video encoder includes a Convolutional Neural Network (CNN) to extract spatial information from image states $s_{1:T}$ and a non-causal transformer to capture temporal information from videos. Specifically, we use a CNN backbone to extract visual embeddings $\{x_{1:T}\}$ for all frames. Additionally, motivated by Devlin et al. (2019); Dosovitskiy et al. (2020), we construct a set of learnable embeddings (or summary tokens), represented as $\{c_{1:N}\}$, to capture the semantic information present in the video.

The visual embeddings and summary tokens are passed to a non-causal transformer, resulting in the output corresponding to the summary tokens as $\{\hat{c}_{1:N}\}$

$$\begin{aligned} x_{1:T} &\leftarrow \texttt{Backbone}(s_{1:T}), \\ \hat{c}_{1:N} &\leftarrow \texttt{Transformer}([x_{1:T}, c_{1:N}]). \end{aligned} \tag{1}$$

Similar to VAE (Kingma & Welling, 2013), we assume that the latent goal space follows a Gaussian distribution, hence we use two fully connected layers, $\mu(\cdot)$ and $\sigma(\cdot)$, to generate the mean and standard deviation of the distribution, respectively. During training, we use the reparameterization trick to sample a set of embeddings $\{g_{1:N}\}$ from the distribution, where $g_t \sim \mathcal{N}(\mu(\hat{c}_t), \sigma(\hat{c}_t))$. During inference, we use the mean of the distribution as the goal embeddings, i.e. $g_t \leftarrow \mu(\hat{c}_t)$.

4.2 Decoder as Policy

To introduce our policy module, we start with VPT (Baker et al., 2022), a Minecraft foundation model trained with standard behavioral cloning. It is built on Transformer-XL (Dai et al., 2019) that can leverage long-horizon historical states and predict the next action seeing the current observation. However, the vanilla VPT architecture does not support instruction input. To condition the policy on goal embeddings, we draw the inspiration from Flamingo (Alayrac et al., 2022), that is, to insert *gated cross-attention dense* layers into every Transformer-XL block. The keys and values in these layers are obtained from goal embeddings, while the queries are derived from the environment states

$$\begin{aligned} \hat{x}^{(l)}_{1:t} &\leftarrow \texttt{GatedXATTN}(kv = g_{1:N}, q = x^{(l-1)}_{1:t}; \theta_l), \\ x^{(l)}_{1:t} &\leftarrow \texttt{TransformerXL}(qkv = \hat{x}^{(l)}_{1:t}; \theta_l), \\ \hat{a}_t &\leftarrow \texttt{FeedForward}(x^{(M)}_t), \end{aligned} \tag{2}$$

where the policy reuses the visual embeddings extracted by the video encoder, i.e., $x^{(0)}_{1:t} = x_{1:t}$, the policy consists of M transformer blocks, θ_l is the parameter of l-th block, $\hat{a}_t$ is the predicted action. Since our goal space contains information about how to complete a task that is richer than previous language-conditioned policy (Cai et al., 2023; Lifshitz et al., 2023), the cross-attention mechanism is necessary. It allows the GROOT to query the task progress from instruction information based on past states, and then perform corresponding behaviors to complete the remaining progress.

4.3 Training and Inference

The training dataset can be a mixture of Minecraft gameplay videos and offline trajectories. For those videos without actions, an inverse dynamic model (Baker et al., 2022) can be used to generate approximate actions. Limited by the computation resources, we truncated all the trajectories into segments with a fixed length of T without using any prior. We denote the final dataset as $\mathcal{D} = \{(x_{1:T}, a_{1:T})\}_M$, where M is the number of trajectories. We train GROOT in a fully self-supervised manner while the training process can be viewed as self-imitating, that is, training GROOT jointly using behavioral cloning and KL divergence loss

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[\sum_t -\log \pi_\theta(a_t | s_{1:t}, g) + \lambda_{KL} \sum_\tau D_{KL}\left(q_\phi(g|s_{0:T}) \parallel p_\theta(g|s_{0:\tau})\right) \right], \tag{3}$$

where λ_{KL} is the tradeoff coefficient, q_ϕ is a posterior visual encoder, p_θ is a prior video encoder with the same architecture, $g \sim q_\phi(\cdot|s_{0:T})$. More details are in the Appendix D.

5 Result

5.1 Performance on Mastering Minecraft Skills

Minecraft SkillForge Benchmark. In order to comprehensively evaluate the mastery of tasks by agents in Minecraft, we created a diverse benchmark called **Minecraft SkillForge**. It covers 30 tasks from 6 major categories of representative skills in Minecraft, including `collect`, `explore`, `craft`, `tool`, `survive`, and `build`. For example, the task "dig three down and fill one up" in the `build` category asks the agent to first dig three blocks of dirt, then use the dirt to fill the space above; The task of "building a snow golem" () requires the agent to sequentially stack 2 snow

(a) Elo Rating Comparison (b) Winning Rate of GROOT vs. Baselines (c) Success Rate Comparison

Figure 3: **Results on Minecraft SkillForge benchmark. Left:** Tournament evaluation of GROOT assessed by human players. GROOT performs better than state-of-the-art Minecraft agent STEVE-1. A 150-score gap corresponds to a 70% probability of winning. **Middle:** Winning rate of GROOT v.s. other agents on specific task categories. Colors from red to blue denote a decrease in the winning rate. Apart from the human player, GROOT surpasses all other baselines. **Right:** Success rate on 9 representative tasks. GROOT champions process-oriented tasks, such as `dig three down and fill one up` () and `build snow golems` ().

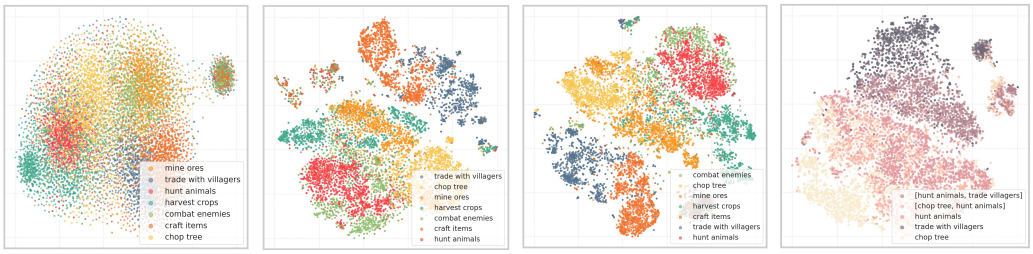
blocks () and 1 carved pumpkin (). We put the details of this benchmark in the Appendix H. Apart from some relatively simple or common tasks such as "collect wood" and "hunt animals", other tasks require the agent to have the ability to perform multiple steps in succession.

We compare GROOT with the following baselines: (a) VPT (Baker et al., 2022), a foundation model pre-trained on large-scale YouTube data, with three variants: VPT (fd), VPT(bc), and VPT(rl), indicating vanilla foundation model, behavior cloning finetuned model, and RL finetuned model; (b) STEVE-1 (Lifshitz et al., 2023), an instruction-following agent finetuned from VPT, with two variants: STEVE-1 (visual) and STEVE-1 (text) that receives visual and test instructions. More details are in Appendix F.1. *It is worth noting that GROOT was trained from scratch.*

Human Evaluation with Elo Rating. We evaluated the relative strength of agents by running an internal tournament and reporting their Elo ratings, as in Mnih et al. (2015). Before the tournament, each agent is required to generate 10 videos of length 600 on each task. Note that, all the reference videos used by GROOT are generated from another biome to ensure generalization. Additionally, we also invited 3 experienced players to do these tasks following the same settings. After the video collection, we asked 10 players to judge the quality of each pair of sampled videos from different agents. Considering the diversity of tasks, we designed specific evaluation criteria for every task to measure the quality of rollout trajectories. After 1500 comparisons, the Elo rating converged as in Figure 3 (left). Although there is a large performance gap compared with human players, GROOT has significantly surpassed the current state-of-the-art STEVE-1 series and condition-free VPT series on the overall tasks. Additional details are in Appendix G.

In Figure 3 (middle), we compare GROOT with other baselines in winning rate on six task groups. We found that except for the performance on `craft` tasks, where STEVE-1 (visual) outperforms our model, GROOT achieves state-of-the-art results. In particular, GROOT greatly outperforms other baselines by a large margin on `build` and `tool`. For `build`, the goal space needs to contain more detailed procedural information, which is the disadvantage of methods that use future outcomes as the goal. Moreover, such tasks are distributed sparsely in the dataset, or even absent in the dataset, which requires the agent to have strong generalization ability. As for `craft` group, GROOT is not superior enough, especially on the "crafting table" task. We attribute this to the wide task distribution in the dataset. Thus the future outcomes can prompt STEVE-1 to achieve a high success rate.

Programmatic Evaluation. To quantitatively compare the performance of the agents, we selected 9 representative tasks out of 30 and reported the success rate of GROOT, STEVE-1 (visual), and VPT (bc) on these tasks in Figure 3 (right). We found that, based on the success rate on tasks such as `dye and shear sheep`(), `enchat sword` (), `smelt food` (), `use bow` (), `sleep` (), and `lead animals` (), GROOT has already reached a level comparable to that of human players (100%). However, the success rates for `build snow golems` () and `build obsidian` () tasks are only 60% and 50%. By observing the generated videos, we



(a) Random Initialized **(b) GROOT w/o KL** **(c) GROOT w/ KL** **(d) Synthesized Videos**

Figure 4: **t-SNE visualization of the goal space.** Each color corresponds to a specific video category. **(Left):** Space of randomly initialized video encoder. All the videos are entangled together. **Middle:** Space of GROOT trained with self-supervised learning w/ and w/o KL regularization, respectively. The videos are clustered based on their semantics. Visualization shows the subtle differences between the two. **Right:** Synthesized videos using concatenation manner. The concatenated videos lay on the position between the source videos.

found that GROOT cannot precisely identify the items in Hotbar (such as buckets, lava buckets, snow blocks, and pumpkin heads), resulting in a low probability of switching to the correct item. STEVE-1 also has the same problem. This may be due to the current training paradigm lacking strong supervisory signals at the image level. Future work may introduce auxiliary tasks such as vision-question answering (VQA) to help alleviate this phenomenon. Details are in Appendix F.3.

5.2 Properties of Learned Goal Space

This section studies the properties of learned goal space. We used the t-SNE algorithm (van der Maaten & Hinton, 2008) to visualize the clustering effect of reference videos encoded in goal space, as in Figure 4. We select 7 kinds of videos, including `craft items`, `combat enemies`, `harvest crops`, `hunt animals`, `chop trees`, `trade with villagers`, and `mine ores`. These videos are sampled from the contractor data (Baker et al., 2022) according to the meta information (details are in Appendix F.2). Each category contains 1k video segments. As a control group, in Figure 4 (left), we showed the initial goal space of the video encoder (with a pre-trained EfficientNet-B0 (Tan & Le, 2019) as the backbone) before training. We found that the points are entangled together. After being trained on offline trajectories, as in Figure 4 (middle), it well understands reference videos and clusters them according to their semantics. This proves that it is efficient to learn behavior-relevant task representations using our self-supervised training strategy. The clustering effect is slightly better with KL regularization, though the difference is not very significant. Inevitably, there are still some videos from different categories entangled together. We attribute this to the possibility of overlap in the performed behaviors of these videos. For example, `chop trees` and `harvest crops` both rely on a sequential of "attack" actions.

Condition on Concatenated Videos. We also study the possibility of conditioning the policy on concatenated videos. First, we collect 3 kinds of source videos, including `chop trees`, `hunt animals`, and `trade with villagers`. We randomly sampled two videos from sources of `chop trees` and `hunt animals`, downsampled and concatenated them into a synthetic video, denoted as `[chop trees, hunt animals]`. By the same token, we can obtain `[hunt animals, trade with villagers]`. We visualize these videos together with the source videos in Figure 4 (right). We found that the source videos lie far away from each other while the concatenated videos are distributed between their source videos. Based on this intriguing phenomenon, we infer that concatenated videos may prompt GROOT to solve both tasks simultaneously. To verify this, we evaluate GROOT on three kinds of reference videos, i.e., `chop trees`, `hunt animals`, and `[chop trees, hunt animals]`. We launched GROOT in the forest and in the animal plains, respectively. The collected wood and killed mobs are reported in Figure 5. We found that although the concatenated video may not be as effective as raw video in driving an agent to complete a single task (60% of the performance of raw video), it does possess the ability to drive the agent to perform multiple tasks. This is an important ability. As discussed in Wang et al. (2023b), sometimes the high-level planner will propose multiple candidate goals, it will be efficient if the low-level controller can automatically determine which to accomplish based on the current observation.

Ablation on KL Divergence Loss. To investigate the role of KL loss in training, we evaluated GROOT (w/ KL) and its variant (w/o KL) on three tasks: `collect seagrass` (), `collect`

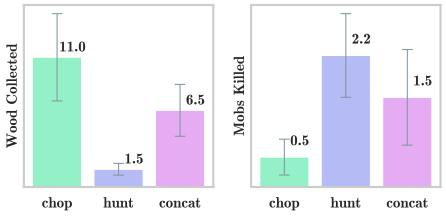


Figure 5: **Comparison on using raw and concatenated reference videos as conditions. Left:** Collected wood in the forest biome. **Right:** Killed mobs in the plains biome. "concat" denotes the reference video is `[chop trees, hunt animals]`. Statistics are measured over 10 episodes.

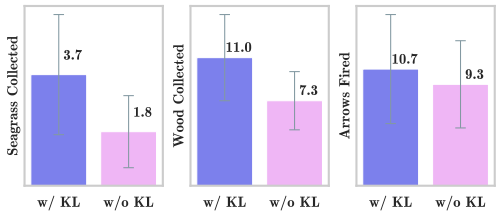


Figure 6: **Ablation study on KL loss.** After being jointly trained with KL loss, GROOT can collect $2\times$ more seagrass ( ) underwater and $1.5\times$ wood ( ) in the forest while the difference is not as impressive on the `use bow` ( ) task. Statistics are measured over 10 episodes.

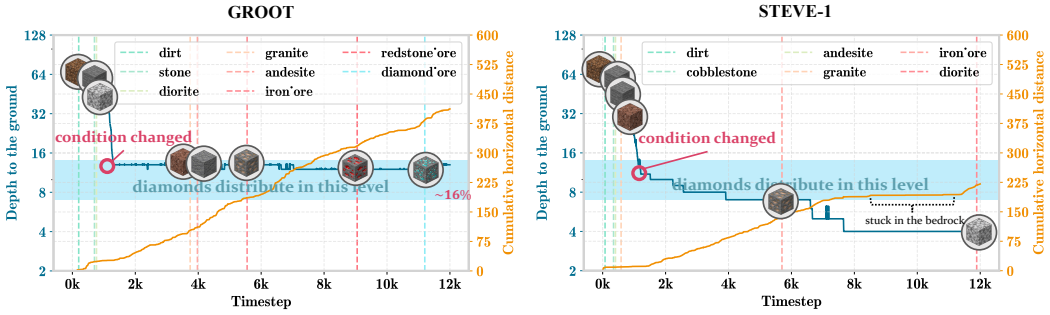


Figure 7: **Results on solving challenging `obtain diamond` task.** The vertical dashed lines represent the time when a certain item is first obtained. **Left:** GROOT first dug down to the depth of 12 and then mined horizontally to obtain diamonds with an average success rate of 16%. **Right:** STEVE-1 quickly dug down to the specific depth but struggled to maintain its height.

`wood` ( ), and `use bow` ( ). As shown in Figure 6, we found that introducing the constraint of KL loss improved agent performance by $2\times$ and $1.5\times$ in the first two tasks, whereas there was no significant effect in the `use bow` task. This may be because the first two tasks require the agent to generalize the corresponding skills to different terrains (e.g. locating trees in the environment for collecting wood and sinking to specific locations for collecting seagrass). Therefore, it puts higher demands on the agent's ability to generalize in the goal space, and this is exactly the role played by the KL loss. The `use bow` task is relatively simple in comparison because it only requires charging and shooting the arrow, without considering environmental factors.

5.3 Combining Skills for Long-horizon Tasks

In this section, we explore whether GROOT can combine skills to solve long-horizon tasks, which is key to its integration with a high-level planner. Taking the task of mining diamonds as an example, prior knowledge is that diamond ores are generally distributed between the 7th and 14th floors underground, and the probability of appearing in other depths is almost zero. Therefore, the agent needs to first dig down to the specified depth (12) and then maintain horizontal mining. To achieve this, we designed two reference videos, each 128 frames long. One describes the policy of starting from the surface and digging down, and the other demonstrates the behaviors of horizontal mining. We show an example in Figure 7 (left). In the beginning, GROOT quickly digs down to the specified depth and then switches to horizontal mining mode. It maintains the same height for a long time and found diamonds at 11k steps. In addition, we compared STEVE-1 (visual) under the same setting in Figure 7 (right). After switching to the horizontal mining prompt, STEVE-1 maintains its height for a short time before it stuck in the bedrock layer (unbreakable in survival mode), greatly reducing the probability of finding diamonds. This indicates that our goal space is expressive enough to instruct the way of mining, and the policy can follow the instructions persistently and reliably. In contrast, STEVE-1, which relies on future outcomes as a condition, was unable to maintain its depth, despite attempts at various visual prompts. We conducted 25 experiments each on GROOT and STEVE-1, with success rates of 16% and 0% for finding diamonds. Additional details are in the Appendix F.4.

6 Related Works

Pre-train Policy on Offline Data. Pre-training neural networks on web-scale data has been demonstrated as an effective training paradigm in Nature Language Processing (Brown et al., 2020) and Computer Vision (Kirillov et al., 2023). Inspired by this, researchers tried to transfer the success to the field of decision-making from pre-training visual representations and directly distilling the policy from offline data. As the former, Aytar et al. (2018a); Bruce et al. (2023) leveraged temporal information present in videos as the supervision signal to learn visual representations. The representations are then used to generate intrinsic rewards for boosting downstream policy learning, which still requires expensive online interactions with the environment. Schmidhuber (2019); Chen et al. (2021) leveraged scalable offline trajectories to train optimal policy by conditioning it on cumulated rewards. Laskin et al. (2022) proposed to learn an in-context policy improvement operator that can distill an RL algorithm in high data efficiency. Reed et al. (2022) learned a multi-task agent Gato by doing behavior cloning on a large-scale expert dataset. By serializing task data into a flat of sequence, they use the powerful transformer architecture to model the behavior distribution. However, these methods either require elaborated reward functions or explicit task definitions. This makes it hard to be applied to open worlds, where tasks are infinite while rewards are lacking. Another interesting direction is to use pre-trained language models for reasoning and vision language models for discrimination, to guide the policy in life-long learning in the environment (Di Palo et al., 2023).

Condition Policy on Goal Space. Researchers have explored many goal modalities, such as language (Khandelwal et al., 2021), image (Du et al., 2021), and future video (Xie et al., 2023), to build a controllable policy. Brohan et al. (2022) collected a large-scale dataset of trajectory-text pairs and trained a transformer policy to follow language instructions. Despite the language being a natural instruction interface, the cost of collecting paired training data is expensive. As a solution, Majumdar et al. (2022) sorted to use hindsight relabeling to first train a policy conditioned on the target image, then aligned text to latent image space, which greatly improves training efficiency. Lifshitz et al. (2023) moved a big step on this paradigm by replacing the target image with a 16-frame future video and reformulating the modality alignment problem into training a prior of latent goal given text.

Build Agents in Minecraft. As a challenging open-world environment, Minecraft is attracting an increasing number of researchers to develop AI agents on it, which can be divided into plan-oriented (Wang et al., 2023b;a) and control-oriented methods (Baker et al., 2022; Cai et al., 2023; Lifshitz et al., 2023) based on their emphasis. Plan-oriented agents aim to reason with Minecraft knowledge and decompose the long-horizon task into sub-tasks followed by calling a low-level controller. Control-oriented works follow the given instructions and directly interact with the environments using low-level actions (mouse and keyboard). Baker et al. (2022) pre-trained the first foundation model VPT in Minecraft using internet-scale videos. Although it achieves the first obtaining diamond milestone by fine-tuning with RL, it does not support instruction input. Lifshitz et al. (2023) created the first agent that can solve open-ended tasks by bridging VPT and MineCLIP (Fan et al., 2022). However, its goal space is not expressive enough and prevents it from solving multi-step tasks.

7 Limitations and Conclusion

Although GROOT has demonstrated powerful capabilities in expressing open-ended tasks in the form of video instructions, training such a goal space remains highly challenging. We found that GROOT is quite sensitive to the selection of reference videos, which we attribute to the fact that the goal space trained from an unsupervised perspective may not be fully aligned with the human intention for understanding the semantics of the reference video. Therefore, it would be a promising research direction in the future to use SFT (supervised fine-tuning, Sanh et al. (2021)) and RLHF (Ziegler et al., 2019) to align the pre-trained goal space with human preference.

We propose a paradigm for learning to follow instructions by watching gameplay videos. We prove that video instruction is a good form of goal space that not only expresses open-ended tasks but can be trained through self-imitation (once the IDM is available to label pseudo actions for raw gameplay videos). Based on this, we built an encoder-decoder transformer architecture agent named GROOT in Minecraft. Without collecting any text-video data, GROOT demonstrated extraordinary instruction-following ability and crowned the Minecraft SkillForge benchmark. Additionally, we also demonstrate its potential as a planner downstream controller in the challenging `obtain diamond` task. We believe that this training paradigm can be generalized in other complex open-world environments.

ACKNOWLEDGEMENTS

This work is funded in part by the National Key R&D Program of China #2022ZD0160301, a grant from CCF-Tencent Rhino-Bird Open Research Fund.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. *ArXiv*, abs/2204.14198, 2022. URL https://api.semanticscholar.org/CorpusID:248476411. 5, 18

Marcin Andrychowicz, Dwight Crow, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Joshua Tobin, P. Abbeel, and Wojciech Zaremba. Hindsight experience replay. *ArXiv*, abs/1707.01495, 2017. URL https://api.semanticscholar.org/CorpusID:3532908. 3

Yusuf Aytar, Tobias Pfaff, David Budden, Tom Le Paine, Ziyun Wang, and Nando de Freitas. Playing hard exploration games by watching youtube. In *Neural Information Processing Systems*, 2018a. URL https://api.semanticscholar.org/CorpusID:44061126. 9

Yusuf Aytar, Tobias Pfaff, David Budden, Tom Le Paine, Ziyun Wang, and Nando de Freitas. Playing hard exploration games by watching youtube. In *Neural Information Processing Systems*, 2018b. URL https://api.semanticscholar.org/CorpusID:44061126. 3

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*, abs/2206.11795, 2022. URL https://api.semanticscholar.org/CorpusID:249953673. 2, 3, 5, 6, 7, 9, 16, 17, 18, 19, 20, 21

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J. Joshi, Ryan C. Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S. Ryoo, Grecia Salazar, Pannag R. Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Anand Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Ho Vuong, F. Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale. *ArXiv*, abs/2212.06817, 2022. URL https://api.semanticscholar.org/CorpusID:254591260. 2, 4, 9, 17

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020. URL https://api.semanticscholar.org/CorpusID:218971783. 9

Jake Bruce, Ankit Anand, Bogdan Mazoure, and Rob Fergus. Learning about progress from experts. In *International Conference on Learning Representations*, 2023. URL https://api.semanticscholar.org/CorpusID:259298702. 9

Shaofei Cai, Zihao Wang, Xiaojian Ma, Anji Liu, and Yitao Liang. Open-world multi-task control through goal-aware representation learning and adaptive horizon prediction. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13734–13744, 2023. URL https://api.semanticscholar.org/CorpusID:256194112. 2, 3, 5, 9, 16, 32

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, P. Abbeel, A. Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In *Neural Information Processing Systems*, 2021. URL https://api.semanticscholar.org/CorpusID:235294299. 2, 4, 9

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Jan 2019. doi: 10.18653/v1/p19-1285. URL http://dx.doi.org/10.18653/v1/p19-1285. 5, 18

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019. URL https://api.semanticscholar.org/CorpusID:52967399. 4

Norman Di Palo, Arunkumar Byravan, Leonard Hasenclever, Markus Wulfmeier, Nicolas Heess, and Martin Riedmiller. Towards a unified agent with foundation models. In *Workshop on Reincarnating Reinforcement Learning at ICLR 2023*, 2023. 9

Yiming Ding, Carlos Florensa, Mariano Phielipp, and P. Abbeel. Goal-conditioned imitation learning. *ArXiv*, abs/1906.05838, 2019. URL https://api.semanticscholar.org/CorpusID:189762519. 3

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2020. URL https://api.semanticscholar.org/CorpusID:225039882. 4

Heming Du, Xin Yu, and Liang Zheng. Vtnet: Visual transformer network for object goal navigation. *ArXiv*, abs/2105.09447, 2021. URL https://api.semanticscholar.org/CorpusID:234790212. 9

Lasse Espeholt, Hubert Soyer, Rémi Munos, Karen Simonyan, Volodymyr Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. *ArXiv*, abs/1802.01561, 2018. URL https://api.semanticscholar.org/CorpusID:3645060. 3

Linxi (Jim) Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *ArXiv*, abs/2206.08853, 2022. URL https://api.semanticscholar.org/CorpusID:249848263. 1, 9, 16, 20, 31

Ran Gong, Qiuyuan Huang, Xiaojian Ma, Hoi Vo, Zane Durante, Yusuke Noda, Zilong Zheng, Song-Chun Zhu, Demetri Terzopoulos, Li Fei-Fei, et al. Mindagent: Emergent gaming interaction. *arXiv preprint arXiv:2309.09971*, 2023. 3

William H. Guss, Brandon Houghton, Nicholay Topin, Phillip Wang, Cayden Codel, Manuela M. Veloso, and Ruslan Salakhutdinov. Minerl: A large-scale dataset of minecraft demonstrations. In *International Joint Conference on Artificial Intelligence*, 2019. URL https://api.semanticscholar.org/CorpusID:199000710. 16

Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. Vln-bert: A recurrent vision-and-language bert for navigation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1643–1653, 2020. URL https://api.semanticscholar.org/CorpusID:227228335. 3

Wenlong Huang, F. Xia, Ted Xiao, Harris Chan, Jacky Liang, Peter R. 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. In *Conference on Robot Learning*, 2022. URL https://api.semanticscholar.org/CorpusID:250451569. 3

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *ArXiv*, abs/2307.05973, 2023. URL https://api.semanticscholar.org/CorpusID:259837330. 3

Mingxuan Jing, Xiaojian Ma, Wenbing Huang, Fuchun Sun, Chao Yang, Bin Fang, and Huaping Liu. Reinforcement learning from imperfect demonstrations under soft expert guidance. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 5109–5116, 2020. 3

Mingxuan Jing, Wenbing Huang, Fuchun Sun, Xiaojian Ma, Tao Kong, Chuang Gan, and Lei Li. Adversarial option-aware hierarchical imitation learning. In *International Conference on Machine Learning*, pp. 5097–5106. PMLR, 2021. 3

Matthew Johnson, Katja Hofmann, Tim Hutton, and David Bignell. The malmo platform for artificial intelligence experimentation. In *International Joint Conference on Artificial Intelligence*, 2016. URL https://api.semanticscholar.org/CorpusID:9953039. 1, 3, 16

Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: Clip embeddings for embodied ai. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14809–14818, 2021. URL https://api.semanticscholar.org/CorpusID:244346010. 9

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013. URL https://api.semanticscholar.org/CorpusID:216078090. 5

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. *ArXiv*, abs/2304.02643, 2023. URL https://api.semanticscholar.org/CorpusID:257952310. 9

Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, DJ Strouse, Steven Stenberg Hansen, Angelos Filos, Ethan Brooks, Maxime Gazeau, Himanshu Sahni, Satinder Singh, and Volodymyr Mnih. In-context reinforcement learning with algorithm distillation. *ArXiv*, abs/2210.14215, 2022. URL https://api.semanticscholar.org/CorpusID:253107613. 9

Shalev Lifshitz, Keiran Paster, Harris Chan, Jimmy Ba, and Sheila A. McIlraith. Steve-1: A generative model for text-to-behavior in minecraft. *ArXiv*, abs/2306.00937, 2023. URL https://api.semanticscholar.org/CorpusID:258999563. 2, 5, 6, 9, 16, 18, 19, 31

Zichuan Lin, Junyou Li, Jianing Shi, Deheng Ye, Qiang Fu, and Wei Yang. Juewu-mc: Playing minecraft with sample-efficient hierarchical reinforcement learning. *arXiv preprint arXiv:2112.04907*, 2021. 2

Minghuan Liu, Menghui Zhu, and Weinan Zhang. Goal-conditioned reinforcement learning: Problems and solutions. *ArXiv*, abs/2201.08299, 2022. URL https://api.semanticscholar.org/CorpusID:246063885. 3

Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. Zson: Zero-shot object-goal navigation using multimodal goal embeddings. *ArXiv*, abs/2206.12403, 2022. URL https://api.semanticscholar.org/CorpusID:250048645. 9

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Kirkeby Fidjeland, Georg Ostrovski, Stig Petersen, Charlie Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015. URL https://api.semanticscholar.org/CorpusID:205242740. 3, 6

Emilio Parisotto, H. Francis Song, Jack W. Rae, Razvan Pascanu, Çaglar Gülçehre, Siddhant M. Jayakumar, Max Jaderberg, Raphael Lopez Kaufman, Aidan Clark, Seb Noury, Matthew M. Botvinick, Nicolas Manfred Otto Heess, and Raia Hadsell. Stabilizing transformers for reinforcement learning. In *International Conference on Machine Learning*, 2019. URL https://api.semanticscholar.org/CorpusID:204578308. 4

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022. 9

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Stella Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. Multitask prompted training enables zero-shot task generalization, 2021. 9

Juergen Schmidhuber. Reinforcement learning upside down: Don't predict rewards - just map them to actions. *ArXiv*, abs/1912.02875, 2019. URL https://api.semanticscholar.org/CorpusID:208857600. 9

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347, 2017. URL https://api.semanticscholar.org/CorpusID:28695052. 3

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, L. Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Vedavyas Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy P. Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489, 2016. URL https://api.semanticscholar.org/CorpusID:515925. 21

Austin Stone, Ted Xiao, Yao Lu, Keerthana Gopalakrishnan, Kuang-Huei Lee, Quan Ho Vuong, Paul Wohlhart, Brianna Zitkovich, F. Xia, Chelsea Finn, and Karol Hausman. Open-world object manipulation using pre-trained vision-language models. *ArXiv*, abs/2303.00905, 2023. URL https://api.semanticscholar.org/CorpusID:257280290. 3

Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *ArXiv*, abs/1905.11946, 2019. URL https://api.semanticscholar.org/CorpusID:167217261. 7, 17

Maxim Tkachenko, Mikhail Malyuk, Andrey Holmanyuk, and Nikolai Liubimov. Label Studio: Data labeling software, 2020-2022. URL https://github.com/heartexlabs/label-studio. Open source software available from https://github.com/heartexlabs/label-studio. 22

Laurens van der Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008. URL https://api.semanticscholar.org/CorpusID:5855042. 7

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. URL https://api.semanticscholar.org/CorpusID:13756489. 4

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi (Jim) Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *ArXiv*, abs/2305.16291, 2023a. URL https://api.semanticscholar.org/CorpusID:258887849. 1, 3, 9, 16

Zihao Wang, Shaofei Cai, Anji Liu, Xiaojian Ma, and Yitao Liang. Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. *ArXiv*, abs/2302.01560, 2023b. URL https://api.semanticscholar.org/CorpusID:256598146. 1, 3, 7, 9, 16, 32

Zhihui Xie, Zichuan Lin, Deheng Ye, Qiang Fu, Wei Yang, and Shuai Li. Future-conditioned unsupervised pretraining for decision transformer. In *International Conference on Machine Learning*, 2023. URL https://api.semanticscholar.org/CorpusID:258947476. 9

Chao Yang, Xiaojian Ma, Wenbing Huang, Fuchun Sun, Huaping Liu, Junzhou Huang, and Chuang Gan. Imitation learning from observations by minimizing inverse dynamics disagreement. *Advances in neural information processing systems*, 32, 2019. 3

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan C. Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. *ArXiv*, abs/1910.10897, 2019. URL https://api.semanticscholar.org/CorpusID:204852201. 2

Qihang Zhang, Zhenghao Peng, and Bolei Zhou. Learning to drive by watching youtube videos: Action-conditioned contrastive policy pretraining. In *European Conference on Computer Vision*, 2022. URL https://api.semanticscholar.org/CorpusID:250626771. 3, 17

Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyuan Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, Y. Qiao, Zhaoxiang Zhang, and Jifeng Dai. Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory. *ArXiv*, abs/2305.17144, 2023. URL https://api.semanticscholar.org/CorpusID:258959262. 1, 3

Daniel M. Ziegler, Nisan Stiennon, Jeff Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *ArXiv*, abs/1909.08593, 2019. URL https://api.semanticscholar.org/CorpusID:202660943. 9