ROCKET-1: Master Open-World Interaction with Visual-Temporal Context Prompting

Shaofei Cai
Peking University
caishaofei@stu.pku.edu.cn

Zihao Wang
Peking University
zhwang@stu.pku.edu.cn

Kewei Lian
Peking University
lkwkwl@pku.edu.cn

Zhancun Mu
Peking University
muzhancun@pku.edu.cn

Xiaojian Ma
BIGAI
xiaojian.ma@ucla.edu

Anji Liu
University of California, Los Angeles
liuanji@cs.ucla.edu

Yitao Liang*
Peking University
yitaol@pku.edu.cn

Figure 1: **Our pipeline solves creative tasks, such as *get the obsidian* in the original Minecraft version, using the action space identical to human players (mouse & keyboard).** We present a novel instruction interface, *visual-temporal context prompting*, under which we learn a spatial-sensitive policy, ROCKET-1. VLMs identify regions of interest within each observation, effectively guiding ROCKET-1 interactions.

*Corresponding Author.

Abstract

Vision-language models (VLMs) have excelled in multimodal tasks, but adapting them to embodied decision-making in open-world environments presents challenges. A key issue is the difficulty in smoothly connecting individual entities in low-level observations with abstract concepts required for planning. A common approach to address this problem is through the use of hierarchical agents, where VLMs serve as high-level reasoners that break down tasks into executable sub-tasks, typically specified using language and imagined observations. However, language often fails to effectively convey spatial information, while generating future images with sufficient accuracy remains challenging. To address these limitations, we propose visual-temporal context prompting, a novel communication protocol between VLMs and policy models. This protocol leverages object segmentation from both past and present observations to guide policy-environment interactions. Using this approach, we train ROCKET-1, a low-level policy that predicts actions based on concatenated visual observations and segmentation masks, with real-time object tracking provided by SAM-2. Our method unlocks the full potential of VLMs' visual-language reasoning abilities, enabling them to solve complex creative tasks, especially those heavily reliant on spatial understanding. Experiments in Minecraft demonstrate that our approach allows agents to accomplish previously unattainable tasks, highlighting the effectiveness of visual-temporal context prompting in embodied decision-making. Codes and demos will be available on the project page: `https://craftjarvis.github.io/ROCKET-1`.

1 Introduction

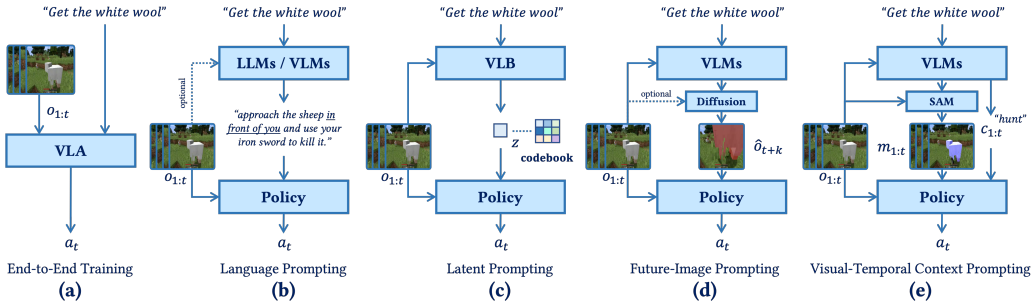

Figure 2: **Different pipelines in solving embodied decision-making tasks. (a)** End-to-end pipeline modeling token sequences of language, observations, and actions. **(b)** Language prompting: VLMs decompose instructions for language-conditioned policy execution. **(c)** Latent prompting: maps discrete behavior tokens to low-level actions. **(d)** Future-image prompting: fine-tunes VLMs and diffusion models for image-conditioned control. **(e)** Visual-temporal prompting: VLMs generate segmentations and interaction cues to guide ROCKET-1.

Pre-trained foundation vision-language models (VLMs) [30, 1] have shown impressive performance in reasoning, visual question answering, and task planning [5, 12, 32, 9], primarily due to training on internet-scale multimodal data. Recently, there has been growing interest in transferring these capabilities to embodied decision-making in open-world environments. Existing approaches can be broadly categorized into (i) end-to-end and (ii) hierarchical approaches. End-to-end approaches, such as RT-2 [5], Octo [24], LEO [16], and OpenVLA [28], aim to enable VLMs to interact with environments by collecting robot manipulation trajectory data annotated with text. This data is then tokenized to fine-tune VLMs into vision-language-action models (VLAs) in an end-to-end manner, as illustrated in Figure 2(a). However, collecting such annotated trajectory data is resource-intensive and difficult to scale. Moreover, introducing the action modality risks compromising the foundational abilities of VLMs.

Hierarchical agent architectures typically consist of a high-level reasoner and a low-level policy, which can be trained independently. In this architecture, the "communication protocol" between components defines the capability limits of the agent. Alternative approaches [32, 31, 12] leverage VLMs'

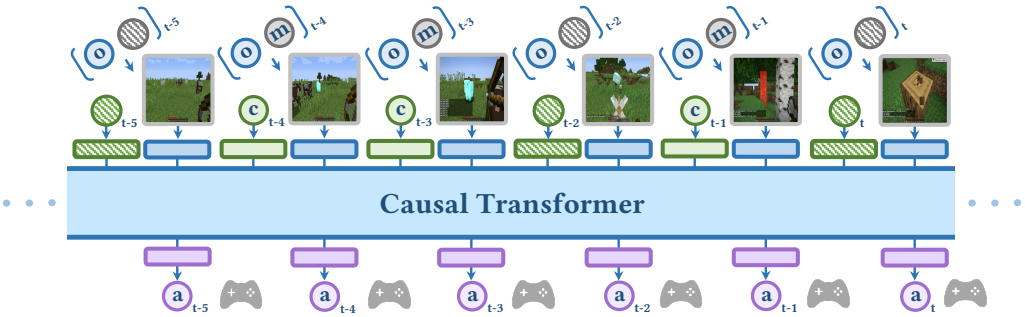

Figure 3: **ROCKET-1 architecture.** ROCKET-1 processes interaction types (c), observations (o), and object segmentations (m) to predict actions (a) using a causal transformer. Observations and segmentations are concatenated and passed through a visual backbone for deep fusion. Interaction types and segmentations are randomly dropped with a set probability during training.

reasoning abilities to zero-shot decompose tasks into language-based sub-tasks, with a separate language-conditioned policy executing them in the environment, refer to Figure 2(b). However, language instructions often fail to effectively convey spatial information, limiting the tasks agents can solve. For example, when multiple homonymous objects appear in an observation image, distinguishing a specific one using language alone may require extensive spatial descriptors, increasing data collection complexity and learning difficulty for the language-conditioned policy. To address this issue, approaches like STEVE-1 [19], GROOT-1 [7], and MineDreamer [39] propose using a purely vision-based interface to convey task information to the low-level policy. MineDreamer, in particular, uses hindsight relabeling to train an image-conditioned policy [19] for interaction, while jointly fine-tuning VLMs and diffusion models to generate goal images that guide the policy, shown in Figure 2(d). Although replacing language with imagined images as the task interface simplifies data collection and policy learning, predicting future observations requires building a world model, which still faces challenges such as hallucinations, temporal inconsistencies, low-quality generation, and limited temporal scope.

In human task execution, such as object grasping, people do not pre-imagine holding an object but maintain focus on the target object while approaching its affordance. When the object is obscured, humans rely on memory to recall its location and connect past and present visual scenes. This use of visual-temporal context enables humans to solve tasks effectively in novel environments. Building on this idea, we propose a novel communication protocol called **visual-temporal context prompting**, as shown in Figure 2(e). This allows users/reasoners to apply object segmentation to highlight regions of interest in past and current visual observations and convey interaction-type cues via a set of skill primitives. Based on this, we learn **ROCKET-1**, a low-level policy that uses visual observations and reasoner-provided segmentations as task prompts to predict actions causally. Specifically, a transformer [10] models dependencies between observations, essential for representing tasks in partially observable environments. As a bonus feature, ROCKET-1 can enhance its object-tracking capabilities during inference by integrating the state-of-the-art video segmentation model, SAM-2 [26], in a plug-and-play fashion. Additionally, we propose a **backward trajectory relabeling** method, which efficiently generates segmentation annotations in reverse temporal order using SAM-2, facilitating the creation of training datasets for ROCKET-1. Finally, we develop a hierarchical agent architecture leveraging visual-temporal context prompting, which perfectly inherits the vision-language reasoning capabilities of foundational VLMs. Experiments in Minecraft demonstrate that our pipeline enables agents to complete tasks previously unattainable by other methods, while the hierarchical architecture effectively solves long-horizon tasks.

Our main contributions are threefold: **(1)** We present **visual-temporal context prompting**, a novel protocol that effectively communicates spatial and interaction cues in hierarchical agent architecture. **(2)** We learn **ROCKET-1**, the first segmentation-conditioned policy in Minecraft, capable of interacting with nearly all the objects. **(3)** We develop **backward trajectory relabeling** method that can automatically detect and segment desired objects in collected trajectories with pre-trained SAMs for training ROCKET-1.

2 Preliminaries

Offline Reinforcement Learning. We model the open-world interaction problem as a Markov Decision Process (MDP) $\langle \mathcal{O}, \mathcal{A}, \mathcal{P}, \mathcal{C}, \mathcal{M}, \mathcal{R} \rangle$, where $\mathcal{O}$ and $\mathcal{A}$ represent the observation and action spaces, $\mathcal{P} : \mathcal{O} \times \mathcal{A} \times \mathcal{O} \to \mathbb{R}^+$ describes the environment dynamics, $\mathcal{C}$ is the set of interaction types, and $\mathcal{M}$ is the segmentation mask space. The binary reward function $\mathcal{R} : \mathcal{O} \times \mathcal{A} \times \mathcal{C} \times \mathcal{M} \to \{0, 1\}$ determines whether the policy has completed the specified interaction with the object indicated by the segmentation mask at each time step. The objective of reinforcement learning is to learn a policy that maximizes the expected cumulative reward, $\mathbb{E}\left[\sum_{t=1}^{T} r_t\right]$, where r_t is the reward at time step t. Our proposed *backward trajectory relabeling* method ensures that each trajectory attains a positive reward based on current object segmentations. This allows us to discard the rewards and learn a conditioned policy $\pi(a|o, c, m)$ directly using behavior cloning. In the offline setting, agents do not interact with the environment but rely on a fixed, limited dataset of trajectories. This setting is harder as it removes the ability to explore the environment and gather additional feedback.

Vision Language Models. Vision-Language Models (VLMs) are machine learning models capable of processing both image and language modalities. Recent advances in generative pretraining have led to the emergence of conversational models like Gemini [30], GPT-4o [1], and Molmo [11], which are trained on large-scale multimodal data and can reason and generate human-like responses based on text and images. Models such as Palm-E [12] have demonstrated strong abilities in embodied question-answering and task planning. However, standalone VLMs cannot often interact directly with environments. Some approaches use VLMs to generate language instructions for driving low-level controllers, but these methods struggle with expressing spatial information. This work focuses on releasing VLMs' spatial understanding in embodied decision-making scenarios. Molmo can accurately identify correlated objects in images using a list of (x, y) coordinates, as demonstrated in `https://molmo.allenai.org`.

Segment Anything Models. The Segment Anything Model (SAM, [18]), introduced by Meta, is a segmentation model capable of interactively segmenting objects based on point or bounding box prompts, or segmenting all objects in an image at once. It demonstrates impressive zero-shot generalization in both real-world and video game environments. Recently, Meta introduced SAM-2 [26], extending segmentation to the temporal domain. With SAM-2, users can prompt object segmentation with points or bounding boxes on a single video frame, and the model will track the object forward or backward in time, refer to `https://ai.meta.com/sam2`. Remarkably, SAM-2 continues tracking even if the object disappears and reappears, making it well-suited for partially observable open-world environments. In addition, we find the SAM models can be equipped with a text prompt module, enabling them to ground text-based concepts in visual images, as seen in grounded SAM [21].

3 Methods

Overview. Our work focuses on addressing complex interactive tasks in open-world environments like Minecraft. We leverage VLMs' visual-language reasoning capabilities to decompose tasks into multiple steps and determine object interactions based on environmental observations. For example, the *"build nether portal"* task requires a sequence of block placements at specific locations. A controller is also needed to map these steps into low-level actions. To convey spatial information accurately, we propose a visual-temporal context prompting protocol and a low-level policy, ROCKET-1. Pretrained VLMs process a sequence of frames $o_{1:t}$ and a language-based task description to generate object segmentations $m_{1:t}$ and interaction types $c_{1:t}$, representing the interaction steps. The learned ROCKET-1 $\pi(a_t|o_{1:t}, m_{1:t}, c_{1:t})$ interprets these outputs to interact with the environment in real-time. In this section, we outline ROCKET-1 's architecture and training methods, the dataset collection process, and a pipeline integrating ROCKET-1 with state-of-the-art VLMs.

ROCKET-1 Architecture. To train ROCKET-1, we prepare interaction trajectory data in the format: $\tau = (o_1, m_1, c_1, a_1, \cdots, o_T, m_T, c_T, a_T)$, where $o_t \in \mathbb{R}^{3xHxW}$ is the visual observation at time t, $m_t \in \{0, 1\}^{1xHxW}$ is a binary mask highlighting the object in o_t for future interaction, $c_t \in \mathbb{N}$ denotes the interaction type, and a_t is the action. If both m_t and c_t are zeros, no region is highlighted at o_t. As shown in Figure 3, ROCKET-1 is formalized as a conditioned policy, $\pi(a_t|o_{1:t}, m_{1:t}, c_{1:t})$, which takes a sequence of observations and object-segmented interaction regions to causally predict

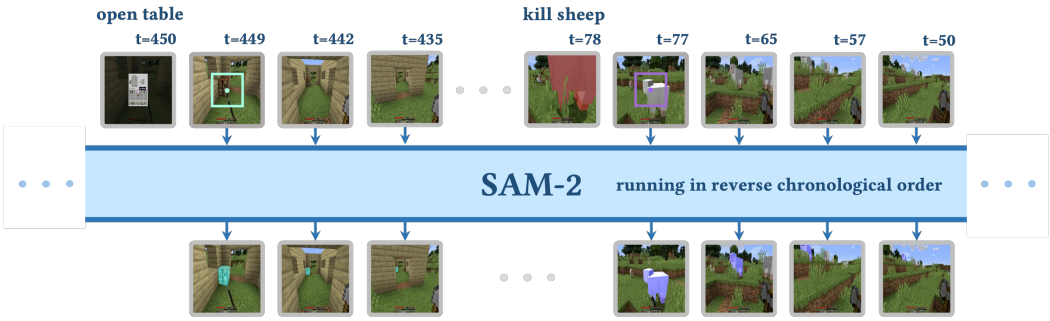

Figure 4: **Trajectory relabeling pipeline in Minecraft.** A bounding box and point selection are applied to the image center in the frame preceding the interaction event to identify the interaction object. SAM-2 is then run in reverse temporal order for a specified duration, with the interaction type remaining consistent throughout.

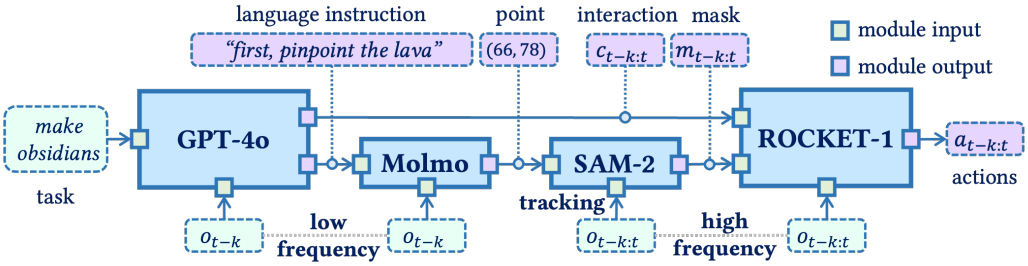

Figure 5: **A hierarchical agent structure based on our proposed visual-temporal context prompting.** A GPT-4o model decomposes complex tasks into steps based on the current observation, while the Molmo model identifies interactive objects by outputting points. SAM-2 segments these objects based on the point prompts, and ROCKET-1 uses the object masks and interaction types to make decisions. GPT-4o and Molmo run at low frequencies, while SAM-2 and ROCKET-1 operate at the same frequency as the environment.

actions. To effectively encode spatial information, inspired by [37], we concatenate the observation and object segmentation pixel-wise into a 4-channel image, which is processed by a visual backbone for deep fusion, followed by an attention pooling layer:

$$h_t \leftarrow \texttt{Backbone}([o_t, m_t]), \tag{1}$$

$$x_t \leftarrow \texttt{AttentionPooling}(h_t). \tag{2}$$

We extend the input channels of the first convolution layer in the pre-trained visual backbone from 3 to 4, initializing the new parameters to zero to minimize impact during early training. A TransformerXL [10, 3] module is then used to model temporal dependencies between observations and incorporate interaction type information to predict the next action $\hat{a}_t$:

$$\hat{a}_t \leftarrow \texttt{TransformerXL}(c_1, x_1, \cdots, c_t, x_t). \tag{3}$$

We delay the integration of interaction type information c_t until after fusing m_t and o_t, enabling the backbone to share knowledge across interaction types and mitigating data imbalance. Behavior cloning loss is used for optimization. However, this approach risks making a_t overly dependent on m_t and c_t, reducing the model's temporal reasoning capability. To address this, we propose randomly dropping segmentations with a certain probability, forcing the model to infer user intent from past inputs (visual-temporal context). The final optimization objective is:

$$\mathcal{L} = -\sum_{t=1}^{|\tau|} \log \pi(a_t | o_{1:t}, m_{1:t} \odot w_{1:t}, c_{1:t} \odot w_{1:t}), \tag{4}$$

where $w_t \sim \text{Bernoulli}(1-p)$ represents a mask, with p denoting the dropping probability, $\odot$ denotes the product operation over time dimension.

Backward Trajectory Relabeling. We seek to build a dataset for training ROCKET-1. The collected trajectory data τ typically contains only observations and actions. To generate object segmentations and interaction types for each frame, we propose a hindsight relabeling technique [2] combined with

Table 1: **Hyperparameters for training ROCKET-1.**

Hyperparameter	**Value**
Input Image Size	224×224
Visual Backbone	EfficientNet-B0 (4 channels)
Policy Transformer	TransformerXL
Number of Policy Blocks	4
Hidden Dimension	1024
Trajectory Chunk size	128
Dropout Rate p	0.75
Optimizer	AdamW
Weight Decay	0.001
Learning Rate	0.00004
Warmup Steps	2000

an object tracking model [26] for efficient data labeling. We first define a set of interactions $\mathcal{C}$ and identify frames where interaction events occur, detected using a pre-trained vision-language model, such as [1]. Then, we traverse the trajectory in reverse order, segmenting interacting objects in frame t via an open-vocabulary grounding model [21]. Finally, SAM-2 [26] is used to track and generate segmentations for frames $t-1, t-2, \ldots, t-k$, where k is the window length.

For Minecraft, we use contractor data [3] from OpenAI, consisting of 1.6 billion frames of human gameplay. This dataset includes meta information for each frame, recording interaction events such as *kill entity*, *mine block*, *use item*, *interact*, *craft*, and *switch*, eliminating the need for vision-language models to detect events. We observed that interacting objects are often centered in the previous frame, allowing the use of a fixed-position bounding box and point with the SAM model for segmentation, replacing open-vocabulary grounding models. We also introduced an additional interaction type, *navigate*, where significant player displacement over a trajectory identifies the most prominent object in the final frame as the target. As shown in Figure 4, the entire labeling process can be automated using SAM-2.

Integration with High-level Reasoner. Completing complex long-horizon tasks in open-world environments requires agents to have strong commonsense knowledge and visual-language reasoning, both of which are strengths of modern VLMs. As shown in Figure 5, we design a hierarchical agent architecture comprising GPT-4o [1], Molmo [11], SAM-2 [26], and ROCKET-1. GPT-4o decomposes tasks into object interactions based on an observation o_{t-k}, leveraging its extensive knowledge and reasoning abilities. Since GPT-4o cannot directly output the object masks, Molmo generates (x, y) coordinates for the described objects. SAM-2 then produces the object mask m_{t-k} from these coordinates and efficiently tracks objects $m_{t-k+1:t}$ in subsequent observations. ROCKET-1 uses the generated masks $m_{t-k:t}$ and interaction types $c_{t-k:t}$ from GPT-4o to engage with the environment. Due to the high computational cost, GPT-4o and Molmo run at lower frequencies, while SAM-2 and ROCKET-1 operate at the environment's frequency.

4 Results and Analysis

We first provide a detailed overview of the experimental setup, including the benchmarks, baselines, and implementation details. We then explore ROCKET-1 's performance on both short-horizon and long-horizon tasks. Finally, we conduct comprehensive ablation studies to validate the rationale behind our design choices.

4.1 Experimental Setup

Implementation Details. Briefly, we present ROCKET-1 's model architecture, hyperparameters, and optimizer configurations in Table 1. During training, each complete trajectory is divided into 128-length segments to reduce memory requirements. During inference, ROCKET-1 can access up to 128 frames of past observations. Most training parameters follow the settings from prior works such as [7, 8, 3].

Environment and Benchmark. We use the unmodified Minecraft 1.16.5 [15, 20] as our testing environment, which accepts mouse and keyboard inputs as the action space and outputs a 640×360

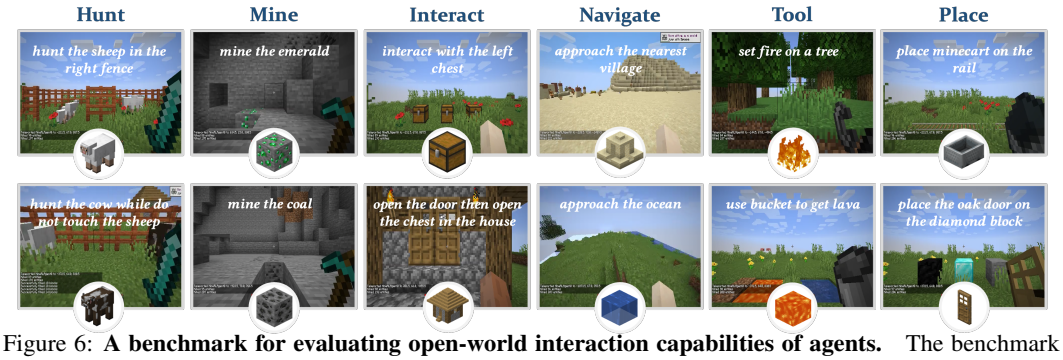

Figure 6: **A benchmark for evaluating open-world interaction capabilities of agents.** The benchmark contains six interaction types in Minecraft, totaling 12 tasks. Unlike previous benchmarks, these tasks emphasize interacting with objects at specific spatial locations. For example, in *"hunt the sheep in the right fence,"* the task fails if the agent kills the sheep on the left side. Some tasks, such as *"place the oak door on the diamond block,"* never appear in the training set. It is also designed to evaluate zero-shot generalization capabilities.

Table 2: **Results on the Minecraft Interaction benchmark.** Each task is tested 32 times, and the average success rate (in %) is reported as the final result. "Human" indicates instructions provided by a human. The subscript "+" indicates the improvement over the previous best.

Method	Prompt	Hunt		Mine		Interact		Navigate		Tool		Place		Overall
VPT-bc	N/A	13	16	0	13	3	31	0	9	0	0	0	0	7
STEVE-1	Human	0	6	0	69	0	3	0	31	91	6	16	0	19
GROOT-1	Human	9	22	0	6	3	6	0	3	47	13	3	0	9
ROCKET-1	Molmo	$\mathbf{91}_{+82}$	$\mathbf{84}_{+62}$	$\mathbf{78}_{+78}$	$\mathbf{75}_{+6}$	$\mathbf{81}_{+78}$	$\mathbf{50}_{+19}$	$\mathbf{78}_{+78}$	$\mathbf{97}_{+66}$	$\mathbf{94}_{+3}$	$\mathbf{91}_{+78}$	$\mathbf{72}_{+56}$	$\mathbf{91}_{+91}$	$\mathbf{82}_{+63}$
ROCKET-1	Human	$\mathbf{94}_{+85}$	$\mathbf{91}_{+69}$	$\mathbf{91}_{+91}$	$\mathbf{94}_{+25}$	$\mathbf{94}_{+91}$	$\mathbf{91}_{+60}$	$\mathbf{97}_{+97}$	$\mathbf{97}_{+66}$	$\mathbf{97}_{+6}$	$\mathbf{97}_{+84}$	$\mathbf{94}_{+78}$	$\mathbf{97}_{+97}$	$\mathbf{95}_{+76}$

RGB image as the observation. To comprehensively evaluate the agent's interaction capabilities, as shown in Figure 6, we introduce the **Minecraft Interaction Benchmark**, consisting of six categories and a total of 12 tasks, including *Hunt*, *Mine*, *Interact*, *Navigate*, *Tool*, and *Place*. This benchmark emphasizes object interaction and spatial localization skills. For example, in the *"hunt the sheep in the right fence"* task, success requires the agent to kill sheep within the right fence, while doing so in the left fence results in failure. In the *"place the oak door on the diamond block"* task, success is achieved only if the oak door is adjacent to the diamond block on at least one side.

Baselines. We compare our methods with the following baselines: (1) VPT [3]: A foundational model pre-trained on large-scale YouTube data, with three variants—VPT (fd), VPT (bc), and VPT (rl)—representing the vanilla foundational model, behavior-cloning finetuned model, and RL-finetuned model, respectively. In this study, we utilize the VPT (bc) variant. (2) STEVE-1 [19]: An instruction-following agent finetuned from VPT, capable of solving various short-horizon tasks. We select the text-conditioned version of STEVE-1 for comparison. (3) GROOT-1 [7]: A reference-video conditioned policy designed to perform open-ended tasks, trained on 2,000 hours of long-form videos using latent variable models.

Table 3: **Comparison of hierarchical architectures with different communication protocols.** All the seven tasks are long-horizon and require complex reasoning capabilities.

Method	Communication Protocol	Policy							
DEPS	language	STEVE-1	95%	75%	15%	2%	15%	0%	0%
MineDreamer*	future image	STEVE-1	95%	-	-	-	0%	0%	0%
OmniJarvis	latent code	GROOT-1	95%	90%	20%	8%	40%	0%	0%
Ours	visual-temporal context	ROCKET-1	**100%**	**100%**	**45%**	**25%**	**75%**	**50%**	**70%**

Figure 7: **Screenshots of our agent when completing long-horizon tasks.**

4.2 Performance on Short-Horizon Tasks

To quantitatively assess ROCKET-1 's performance on short-horizon tasks, we evaluated it on the Minecraft Interaction Benchmark, with results as illustrated in Table 2. Since ROCKET-1 operates as a low-level policy, it requires a high-level reasoner to provide prompts within a visual-temporal context, driving ROCKET-1 's interactions with the environment. We tested two reasoners: (1) A skilled Minecraft human player, who can provide prompts to ROCKET-1 at any interaction moment, serving as an oracle reasoner that demonstrates the upper bound of ROCKET-1 's capabilities. (2) A Molmo 72B model [11], where a predefined Molmo prompt is set for each task to periodically select points in the observation as prompts, which are then processed into object segmentations by the SAM-2 model [26]. Between Molmo's invocations, SAM-2's tracking capabilities offer object segmentations to guide ROCKET-1. For all baselines, humans provide prompts. We found that ROCKET-1 + Molmo consistently outperformed all baselines across all tasks, notably achieving a 91% success rate in the *"place oak door on the diamond block"* task, which no baseline could complete. Furthermore, given that this task is unlikely to appear in the training set, our pipeline demonstrates strong zero-shot generalization capabilities.

4.3 Performance on Long-Horizon Tasks

We compared hierarchical agent architectures based on different communication protocols: (1) language-based approaches, exemplified by DEPS [32]; (2) future-image-based methods, represented by MineDreamer [39]; (3) latent-code-based methods, as in OmniJarvis [34]; and (4) our proposed approach based on visual-temporal context, as illustrated in the Figure 5. For MineDreamer, we used the planner provided by DEPS and MineDreamer as the controller to complete the long-horizon experiment. We evaluated these methods on seven crafting or mining tasks, each requiring long-horizon planning: obtaining a wooden pickaxe (3600), furnace (6000), shears (12000), diamond (24000), steak (6000), obsidian (24000), and pink wool (6000), where the numbers in parentheses represent the time limit for each task. In the first five tasks, the agent starts from scratch, while for the obsidian task, we provided an empty bucket and a diamond pickaxe in advance, and for the pink wool task, we provided shears beforehand. Taking the obsidian task as an example, the player must first locate a nearby water source, fill the bucket, find a nearby lava pool, pour the water to form obsidian, and finally switch to the diamond pickaxe to mine the obsidian. We found that our approach significantly improved success rates on the first five tasks, particularly achieving a 35% increase in the steak task. For the last two tasks, all previous baseline methods failed, while our approach achieved a 75% success rate on the wool dyeing task. Screenshots are presented in Figure 7.

4.4 What Matters for ROCKET-1?

We conduct ablation studies on short-horizon tasks of Minecraft Interaction benchmark: "Hunt right sheep ( )" and "Mine emerald ( )".

SAM-2 Models. The SAM-2 model acts as a proxy segmentation generator when the high-level reasoner fails to provide timely object segmentations. We evaluate the impact of different SAM-2 model sizes on task performance and inference speed, as shown in Table 4. Results indicate that with

Table 4: **Comparison between different SAM-2 variants.** We studied the impact of SAM-2 models of different sizes on the agent's object-tracking capability (metric: success rate) and inference speed (metric: frame-per-second, FPS) on "Hunt right sheep" and "Mine emerald ore" tasks. "P-GAP" indicates the number of frames between prompts generated by Molmo.

Variants	**P-GAP**	**FPS** ↑	↑	↑
baseline (w/o sam2)	3	0.9	84%	**82**%
baseline (w/o sam2)	30	**9.2**	0%	3%
+sam2_tiny	30	5.4	84%	69%
+sam2_small	30	5.1	88%	50%
+sam2_base_plus	30	3.0	88%	63%
+sam2_large	30	2.4	**91**%	78%

Table 5: **Comparison of interaction-type information fusion at different positions.**

Interaction Type Fusion	**Hunt** ( )↑	**Mine** ( ) ↑
in transformer layer	**91**%	**78**%
in visual backbone	72%	69%

low-frequency prompts from the high-level reasoner (Molmo 72B) at 1.5 (game frequency is 20), SAM-2 greatly improves task success rates. While "sam2_hiera_large" is the best, increasing the SAM-2 model size yields performance gains at the cost of higher time.

Interaction-Type Information Fusion Location. We modified the visual backbone's input layer from 3 to 4 channels, allowing ROCKET-1 to integrate object segmentation information. For fusing interaction-type information, we explored two approaches: (1) keeping the object segmentation channel binary and encoding interaction types via an embedding layer for fusion in TransformerXL, and (2) directly encoding interaction types into the object segmentation for fusion within the visual backbone. As shown in Table 5, the first approach significantly outperformed the second, as it allows the visual backbone to share knowledge across different interaction types and focus on recognizing objects of interest without being affected by imbalanced interaction-type distributions.

5 Related Works

Instructions for Multi-Task Policy. The task instruction of a multi-task policy directly determines the types of tasks the policy can solve. Most current approaches [4, 5, 22, 6, 16] use natural language to describe task details and collect large amounts of text-demonstration data pairs to train a language-conditioned policy for interaction with the environment. Although natural language can express a wide range of tasks, it struggles to represent spatial relationships effectively. Additionally, gathering text-annotated demonstration data is costly, limiting the scalability of these methods. Alternatives, such as [19, 23, 29], use images to drive goal-conditioned policies, typically learning through hindsight relabeling in a self-supervised manner. While this reduces the need for annotated data, future images are often insufficiently expressive, making it difficult to capture detailed task execution processes. Methods like [7, 17] propose using reference videos to describe tasks, offering strong expressiveness but suffering from ambiguity, which may lead to inconsistencies between policy interpretation and human understanding, raising safety concerns. [14] suggests representing tasks with rough robot arm trajectories, enabling novel task completion but only in fully observable environments, limiting its applicability in open-world settings. CLIPort [27], which addresses pick-and-place tasks by controlling the robot's start and end positions using heatmaps, bears some resemblance to our proposed visual-temporal context prompting method. However, CLIPort focuses solely on the pick-and-place task solutions in a fully observable environment.

Agents in Minecraft. Minecraft offers a highly open sandbox environment with complex tasks and free exploration, ideal for testing AGI's adaptability and long-term planning abilities. Its rich interactions and dynamic environment simulate real-world challenges, making it an excellent testbed for AGI. One line of research focuses on low-level control policies in Minecraft. [3] annotated a large YouTube Minecraft video dataset with actions and trained the first foundation agent in the domain

using behavior cloning, but it lacks instruction-following capabilities. [6] employs a goal-sensitive backbone and horizon prediction module to enhance multi-task execution in partially observable environments, but it only solves tasks seen during training. [13] fine-tunes a vision-language alignment model MineCLIP using YouTube video data, and incorporates it into a reward shaping mechanism for training a multi-task agent, though task transfer still requires extensive environment interaction. [19] uses hindsight-relabeling to learn an image-goal-conditioned policy and aligns image and text spaces via MineCLIP, but this approach is limited to short-horizon tasks. [7] uses reference videos as task representations and trains a video-conditioned policy through a latent variable model, but generating environment-consistent reference videos remains a challenge. Another research focus integrates vision-language models for long-horizon task planning in Minecraft [36, 35, 25, 38]. DEPS [32], the first to apply large language models in Minecraft, uses a four-step process to decompose tasks, achieving the diamond mining challenge with minimal training. Voyager [31] highlights LLM-based agents' autonomous exploration and skill-learning abilities. Jarvis-1 [33] extends DEPS with multimodal memory, improving long-horizon task success rates by recalling past experiences. OmniJarvis [34] learns a behavior codebook using self-supervised methods to jointly model language, images, and actions. MineDreamer [39] fine-tunes VLMs and a diffusion model to generate goal images for task execution, though it faces challenges with image quality and consistency.

6 Conclusions

In this paper, we propose a novel hierarchical agent architecture to address the challenges of open-world interaction. To bridge spatial information gaps, we introduce a visual-temporal context prompting method that communicates intent between the high-level reasoner and the low-level policy. We develop ROCKET-1, an object-segmentation-conditioned policy responsible for real-time interaction with objects in the environment. This architecture can also leverage SAM-2 plug-and-play to enhance the policy's object-tracking capabilities. Extensive experiments in Minecraft demonstrate that our approach successfully harnesses the powerful visual-language reasoning abilities of VLMs for embodied decision-making, achieving significantly superior open-world interaction performance compared to baseline methods.

7 Limitations

Although ROCKET-1 significantly enhances interaction capabilities in Minecraft, it cannot engage with objects that are outside its field of view or have not been previously encountered. For instance, if the reasoner instructs ROCKET-1 to eliminate a sheep that it has not yet seen, the reasoner must indirectly guide ROCKET-1 's exploration by providing segmentations of other known objects. This limitation reduces ROCKET-1 's efficiency in completing simple tasks and necessitates frequent interventions from the reasoner, leading to increased computational overhead.

References

[1] O. J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, et al. Gpt-4 technical report. 2023. URL https://api.semanticscholar.org/CorpusID:257532815.

[2] M. Andrychowicz, D. Crow, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba. Hindsight experience replay. *ArXiv*, abs/1707.01495, 2017. URL https://api.semanticscholar.org/CorpusID:3532908.

[3] B. Baker, I. Akkaya, P. Zhokhov, J. Huizinga, J. Tang, A. Ecoffet, B. Houghton, R. Sampedro, and J. Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *ArXiv*, abs/2206.11795, 2022. URL https://api.semanticscholar.org/CorpusID:249953673.

[4] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. J. Joshi, R. C. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. S. Ryoo, G. Salazar, P. R. Sanketi, K. Sayed, J. Singh, S. A. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. H. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich.

Rt-1: Robotics transformer for real-world control at scale. *ArXiv*, abs/2212.06817, 2022. URL https://api.semanticscholar.org/CorpusID:254591260.

[5] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[6] S. Cai, Z. Wang, X. Ma, A. Liu, and Y. 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)*, pages 13734–13744, 2023. URL https://api.semanticscholar.org/CorpusID:256194112.

[7] S. Cai, B. Zhang, Z. Wang, X. Ma, A. Liu, and Y. Liang. Groot: Learning to follow instructions by watching gameplay videos. In *The Twelfth International Conference on Learning Representations*, 2023.

[8] S. Cai, B. Zhang, Z. Wang, X. Ma, A. Liu, and Y. Liang. GROOT-1.5: Learning to follow multi-modal instructions from weak supervision. In *Multi-modal Foundation Model meets Embodied AI Workshop @ ICML2024*, 2024. URL https://openreview.net/forum?id=zxdi4Kdfjq.

[9] Y. Cheng, C. Zhang, Z. Zhang, X. Meng, S. Hong, W. Li, Z. Wang, Z. Wang, F. Yin, J. Zhao, et al. Exploring large language model based intelligent agents: Definitions, methods, and prospects. *arXiv preprint arXiv:2401.03428*, 2024.

[10] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. 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.

[11] M. Deitke, C. Clark, S. Lee, R. Tripathi, Y. Yang, J. S. Park, et al. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models, 2024. URL https://arxiv.org/abs/2409.17146.

[12] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

[13] L. J. Fan, G. Wang, Y. Jiang, A. Mandlekar, Y. Yang, H. Zhu, A. Tang, D.-A. Huang, Y. Zhu, and A. Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *ArXiv*, abs/2206.08853, 2022. URL https://api.semanticscholar.org/CorpusID:249848263.

[14] J. Gu, S. Kirmani, P. Wohlhart, Y. Lu, M. G. Arenas, K. Rao, W. Yu, C. Fu, K. Gopalakrishnan, Z. Xu, et al. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches. *arXiv preprint arXiv:2311.01977*, 2023.

[15] W. H. Guss, B. Houghton, N. Topin, P. Wang, C. Codel, M. M. Veloso, and R. 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] J. Huang, S. Yong, X. Ma, X. Linghu, P. Li, Y. Wang, Q. Li, S.-C. Zhu, B. Jia, and S. Huang. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*, 2023.

[17] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. *ArXiv*, abs/2202.02005, 2022. URL https://api.semanticscholar.org/CorpusID:237257594.

[18] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. B. Girshick. Segment anything. *ArXiv*, abs/2304.02643, 2023. URL https://api.semanticscholar.org/CorpusID:257952310.

[19] S. Lifshitz, K. Paster, H. Chan, J. Ba, and S. 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.

[20] H. Lin, Z. Wang, J. Ma, and Y. Liang. Mcu: A task-centric framework for open-ended agent evaluation in minecraft. *arXiv preprint arXiv:2310.08367*, 2023.

[21] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[22] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, 2023.

[23] A. Majumdar, G. Aggarwal, B. Devnani, J. Hoffman, and D. Batra. Zson: Zero-shot object-goal navigation using multimodal goal embeddings. *ArXiv*, abs/2206.12403, 2022. URL `https://api.semanticscholar.org/CorpusID:250048645`.

[24] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[25] Y. Qin, E. Zhou, Q. Liu, Z. Yin, L. Sheng, R. Zhang, Y. Qiao, and J. Shao. Mp5: A multi-modal open-ended embodied system in minecraft via active perception. *arXiv preprint arXiv:2312.07472*, 2023.

[26] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL `https://arxiv.org/abs/2408.00714`.

[27] M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on robot learning*, pages 894–906. PMLR, 2022.

[28] A. Stone, T. Xiao, Y. Lu, K. Gopalakrishnan, K.-H. Lee, Q. H. Vuong, P. Wohlhart, B. Zitkovich, F. Xia, C. Finn, and K. Hausman. Open-world object manipulation using pre-trained vision-language models. *ArXiv*, abs/2303.00905, 2023. URL `https://api.semanticscholar.org/CorpusID:257280290`.

[29] P. Sundaresan, Q. Vuong, J. Gu, P. Xu, T. Xiao, S. Kirmani, T. Yu, M. Stark, A. Jain, K. Hausman, et al. Rt-sketch: Goal-conditioned imitation learning from hand-drawn sketches. 2024.

[30] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[31] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. J. Fan, and A. Anandkumar. Voyager: An open-ended embodied agent with large language models. *ArXiv*, abs/2305.16291, 2023. URL `https://api.semanticscholar.org/CorpusID:258887849`.

[32] Z. Wang, S. Cai, G. Chen, A. Liu, X. S. Ma, and Y. Liang. Describe, explain, plan and select: interactive planning with llms enables open-world multi-task agents. *Advances in Neural Information Processing Systems*, 36, 2023.

[33] Z. Wang, S. Cai, A. Liu, Y. Jin, J. Hou, B. Zhang, H. Lin, Z. He, Z. Zheng, Y. Yang, et al. Jarvis-1: Open-world multi-task agents with memory-augmented multimodal language models. *arXiv preprint arXiv:2311.05997*, 2023.

[34] Z. Wang, S. Cai, Z. Mu, H. Lin, C. Zhang, X. Liu, Q. Li, A. Liu, X. Ma, and Y. Liang. Omnijarvis: Unified vision-language-action tokenization enables open-world instruction following agents. *arXiv preprint arXiv:2407.00114*, 2024.

[35] Z. Wang, A. Liu, H. Lin, J. Li, X. Ma, and Y. Liang. Rat: Retrieval augmented thoughts elicit context-aware reasoning in long-horizon generation. *arXiv preprint arXiv:2403.05313*, 2024.

[36] H. Yuan, C. Zhang, H. Wang, F. Xie, P. Cai, H. Dong, and Z. Lu. Plan4mc: Skill reinforcement learning and planning for open-world minecraft tasks. *ArXiv*, abs/2303.16563, 2023. URL `https://api.semanticscholar.org/CorpusID:257805102`.

[37] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3836–3847, October 2023.

[38] S. Zheng, Y. Feng, Z. Lu, et al. Steve-eye: Equipping llm-based embodied agents with visual perception in open worlds. In *The Twelfth International Conference on Learning Representations*, 2023.

[39] E. Zhou, Y. Qin, Z. Yin, Y. Huang, R. Zhang, L. Sheng, Y. Qiao, and J. Shao. Minedreamer: Learning to follow instructions via chain-of-imagination for simulated-world control. *arXiv preprint arXiv:2403.12037*, 2024.