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

Shaofei Cai[1], **Zihao Wang**[1], **Kewei Lian**[1], **Zhancun Mu**[1]
Xiaojian Ma[3], **Anji Liu**[2], **Yitao Liang**[1]*
[1]Peking University, [2]University of California, Los Angeles
[3]Beijing Institute for General Artificial Intelligence (BIGAI)
{caishaofei, zhwang, lkwkwl, muzhancun}@stu.pku.edu.cn
xiaojian.ma@ucla.edu,liuanji@cs.ucla.edu,yitaol@pku.edu.cn

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. 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

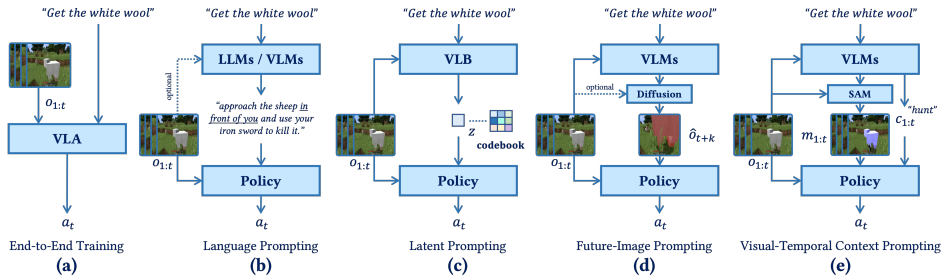

Figure 1: **Different pipelines in solving embodied decision-making tasks.** **(a)** End-to-end pipeline modeling token sequences of language, observations, and actions. **(b)** Language prompting: VLMs decompose instructions for language-conditioned policy execution. **(c)** Latent prompting: maps discrete 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.

*Corresponding Author.

Pre-trained foundation vision-language models (VLMs) [17, 1] have shown impressive performance in reasoning, visual question answering, and task planning [4, 8, 19, 5], 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 [4], Octo [14], LEO [10], and OpenVLA [16], aim to enable VLMs to interact with environments by collecting robot manipulation trajectory data annotated with text. However, collecting such annotated trajectory data is resource-intensive and difficult to scale. 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 [19, 18, 8] leverage VLMs' 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 1(b). However, language instructions often fail to effectively convey spatial information, limiting the tasks agents can solve. To address this issue, approaches like STEVE-1 [11] and MineDreamer [21] (Figure 1 d) propose using a purely vision-based interface to convey task information to the low-level policy. Although replacing language with imagined images as the intention interface simplifies data collection and policy learning, predicting future observations requires building a world model, which still faces challenges such as hallucinations, temporal inconsistencies, and limited temporal scope.

We propose a novel communication protocol called **visual-temporal context prompting**, as shown in Figure 1(e). This allows users/reasoners to apply object segmentation to highlight regions of interest in past observations and convey interaction-type cues via skill primitives. Based on this, we learn **ROCKET-1**, a low-level policy that uses visual observations and reasoner-provided segmentations as prompts to predict actions. Specifically, a transformer [6] 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 [15], 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.

2 Methods

Our work focuses on addressing complex interactive tasks in open-world environments like Minecraft. 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.

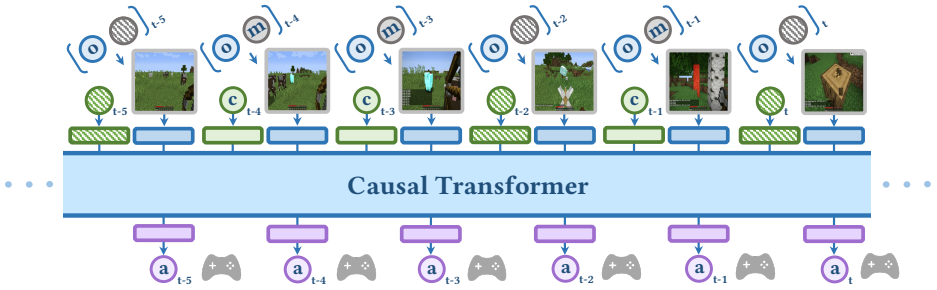

Figure 2: **ROCKET-1 architecture.** ROCKET-1 processes interaction types (c), observations (o), and object segmentations (m) to predict actions (a) using a causal transformer. Observations and segmentations are concatenated and passed through a visual backbone for deep fusion. Interaction types and segmentations are randomly dropped with a set probability during training.

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 $o_t \in \mathbb{R}^{3\times H\times W}$ is the visual observation at time

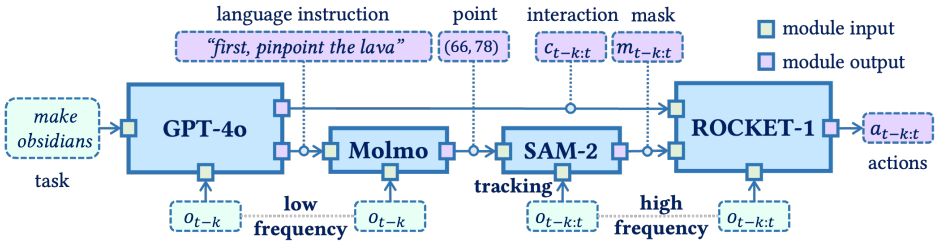

Figure 3: **A hierarchical agent structure based on our proposed visual-temporal context prompting.** A GPT-4o model decomposes complex tasks into steps based on the current observation, while the Molmo model identifies interactive objects by outputting points. SAM-2 segments these objects based on the 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.

t, $m_t \in \{0,1\}^{1\times H\times W}$ 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 2, 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 actions. To effectively encode spatial information, inspired by [20], 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. 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 [6, 3] module is then used to model temporal dependencies between observations and incorporate interaction type information to predict the next action $\hat{a}_t$. 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:

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

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 an object tracking model [15] 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 [13]. Finally, SAM-2 [15] 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. 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 3, we design a hierarchical agent architecture comprising GPT-4o [1], Molmo [7], SAM-2 [15], 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

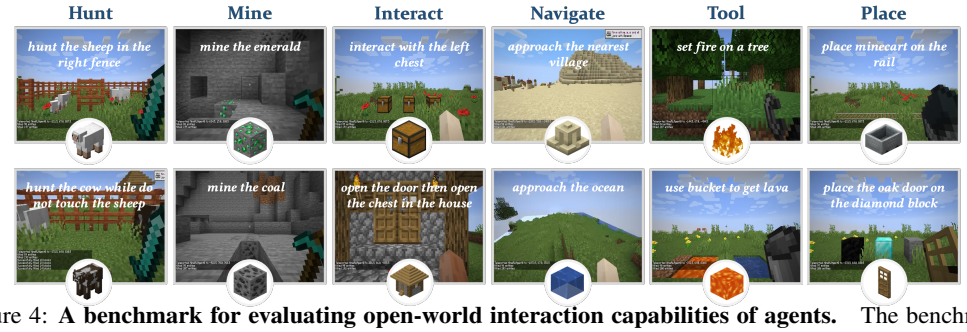

Figure 4: **A benchmark for evaluating open-world interaction capabilities of agents.** The benchmark contains six interaction types in Minecraft, totaling 12 tasks. Unlike previous benchmarks, these tasks emphasize interacting with objects at specific spatial locations.

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.

3 Results and Analysis

Table 1: **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.

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	**91**	**84**	**78**	**75**	**81**	**50**	**78**	**97**	**94**	**91**	**72**	**91**	**82**
ROCKET-1	Human	**94**	**91**	**91**	**94**	**94**	**91**	**97**	**97**	**97**	**97**	**94**	**97**	**95**

Environment and Benchmark. We use the unmodified Minecraft 1.16.5 [9, 12] as our testing environment, which accepts mouse and keyboard inputs as the action space and outputs a 640×360 RGB image as the observation. To comprehensively evaluate the agent's interaction capabilities, as shown in Figure 4, 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.

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 1. 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 [7], 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 [15]. 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, no baseline could complete.

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, 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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`.

[19] 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.

[20] 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.

[21] 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.