ROCKET-2: Steering Visuomotor Policy via Cross-View Goal Alignment

Shaofei Cai[1], **Zhancun Mu**[1], **Anji Liu**[2] **and Yitao Liang** ✉[1]
[1]Peking University, [2]University of California, Los Angeles, All authors are affiliated with Team CraftJarvis

We aim to develop a goal specification method that is semantically clear, spatially sensitive, and intuitive for human users to guide agent interactions in embodied environments. Specifically, we propose a novel cross-view goal alignment framework that allows users to specify target objects using segmentation masks from their own camera views rather than the agent's observations. We highlight that behavior cloning alone fails to align the agent's behavior with human intent when the human and agent camera views differ significantly. To address this, we introduce two auxiliary objectives: cross-view consistency loss and target visibility loss, which explicitly enhance the agent's spatial reasoning ability. According to this, we develop ROCKET-2, a state-of-the-art agent trained in Minecraft, achieving an improvement in the efficiency of inference 3× to 6×. We show ROCKET-2 can directly interpret goals from human camera views for the first time, paving the way for better human-agent interaction. The project page is available at https://craftjarvis.github.io/ROCKET-2/.

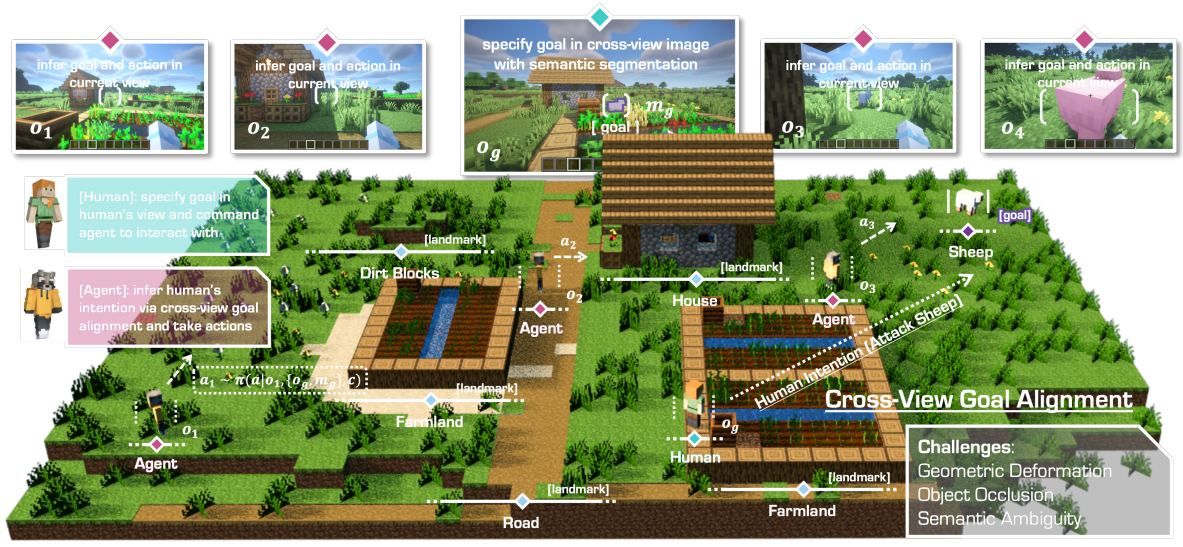

Figure 1 | **Cross-view Goal Alignment as a Human-Friendly Goal Specification Method**. The target object can be specified using a segmentation mask from the human's camera view, while the agent learns to align with human intent and take actions based on its own observations. Visual landmarks serve as key cues to establish spatial relationships across camera views.

1. Introduction

Learning an agent to achieve desired goals is a long-standing challenge in the field of embodied intelligence, with significant implications for the development of robots (Brohan et al., 2022, 2023; Jang et al., 2022) and virtual players (Wang et al., 2023a,b,c). A key challenge is to find goal representations that are (i) flexible for human users to specify and (ii) expressive and precise to capture as many tasks as possible. Most current approaches address only one of these aspects. For example, traditional works (Brohan et al., 2022; Driess et al., 2023; Lynch et al., 2023) focus on training agents to follow language instructions. As pointed out in Cai et al. (2024b); Sundaresan et al. (2024), while language is intuitive, it relies on numerous prepositions to express spatial relationships, which can be vague and inefficient. Furthermore, language also suffers from the gen-

Corresponding author(s): Yitao Liang
Shaofei Cai, Zhancun Mu <{caishaofei, muzhancun}@stu.pku.edu.cn>, Anji Liu <liuanji@cs.ucla.edu>, Yitao Liang <yitaol@pku.edu.cn>

eralization problem of novel visual concepts (Cai et al., 2023). Realizing these limitations, some works attempted to introduce visual modalities into goal representations. For example, Sundaresan et al. (2024) employs hand-drawn target layouts in robot manipulation environments to represent human intent; Gu et al. (2023) uses end-effector trajectory sketches for fine-grained control of robot arms; and ROCKET-1 (Cai et al., 2024b) specifies the objects to interact with by applying segmentation masks to the agent's observations. These methods have greatly improved the expressiveness of spatial relationships and generalization across tasks. However, both trajectory sketches and object segmentation are closely tied to the agent's current observation, causing issues in partially observable 3D worlds. These include: (i) goals need to be generated in real-time as the agent's camera view changes; (ii) goals cannot be specified when the target is occluded.

To strike a balance between expressiveness and flexibility, we propose an innovative and user-friendly cross-view goal specification method. It allows human users to specify the target object using segmentation masks from their own camera view, rather than the agent's camera view. The agent is then trained to align with human intent and take actions based on its own observations via imitation learning. Decoupling the goal specification from the camera view of the agent will significantly enhance the efficiency of human-agent interaction. However, the partial observability of open worlds makes aligning goals across camera views challenging. This involves handling occlusion, geometric deformation, and the distinction of objects of similar look. In Figure 1, we show an agent in the left corner and a human player standing on a farmland. The human intends to command the agent to hunt a sheep near the house, even though the agent cannot initially observe the target sheep. To achieve this, the agent must establish spatial relationships using shared visual landmarks between the human's camera view and its own. We find that relying solely on a behavior cloning loss is insufficient.

To address these challenges, we highlight an important property of behavior datasets (Cai et al., 2024b): **The target object remains consistent across camera views in an interaction trajectory**. Motivated by this, we propose two auxiliary objective functions: *cross-view consistency loss* and *target visibility loss*, to explicitly enhance the agent's ability to align goals across camera views. Specifically, cross-view consistency loss requires the agent to accurately predict the target object's centroid point w.r.t. its camera view, while target visibility loss helps the agent determine whether the target object is occluded. To further leverage temporal consistency, we use a causal Transformer (Vaswani et al., 2017) in the architecture to model the relationship between past predictions and the current observation. It encourages the agent to maintain tracking even when the target is occluded. By combining these auxiliary losses with behavior cloning loss, we develop ROCKET-2, a state-of-the-art agent in Minecraft. Our experiments show that ROCKET-2 can autonomously track the target object as the camera changes, eliminating the need for SAM's (Ravi et al., 2024) real-time semantic segmentation, speeding up inference 3× to 6× compared to ROCKET-1. To our knowledge, we are the first to demonstrate that agents can interpret intentions from a human's camera view and make decisions to achieve expected goals in the embodied world. Extensive visualization experiments and case studies offer deeper insights into its behavior.

Our contributions are threefold: **(1)** We introduce a user-friendly interface that allows humans to specify goals using segmentation masks from their camera view, paving the way for better human-agent interaction. **(2)** We propose *cross-view consistency loss* and *target visibility loss* to explicitly enhance the agent's ability to align targets across camera views and improve its steerability. **(3)** We train ROCKET-2, an agent that autonomously tracks targets, eliminating the need for real-time goal segmentation and significantly speeding up inference.

2. Related Works

Partial Observability. We address policy learning in partially observable 3D worlds (Cai et al., 2024a; Savva et al., 2019), where the agent receives only egocentric images rather than the full

environmental state. Since the movement of the camera view is part of the action space, the policy must actively explore to locate key objects, a common challenge in visual navigation (Savva et al., 2019) and FPS games (Pearce and Zhu, 2022). While some methods (Huang et al., 2023; Jiang et al., 2024) incorporate 3D point clouds to provide global context, such data is often unavailable. More commonly, memory-based architectures integrate historical observations to build implicit 3D perception. For example, Gadre et al. (2022); Zhao et al. (2023) use RNN to avoid redundant exploration, and Baker et al. (2022) employs TransformerXL to retain achievements and complete the long-horizon diamond challenge in Minecraft (Guss et al., 2019). Building on this, we require the policy to align objects across camera views to ensure consistent tracking of targets as the camera view changes. Although cross-view alignment has been explored in computer vision for tasks like BEV segmentation (Borse et al., 2023) and pedestrian re-identification (Xu et al., 2019), we are the first to study its application in open-world decision-making.

Goal-Conditioned Imitation Learning. GCIL refers to algorithms that optimize conditional policies through imitation learning, primarily using behavior cloning loss (Pomerleau, 1988). The policy conditions can take various forms, such as language (Brohan et al., 2022, 2023; Lynch et al., 2023), images (Lifshitz et al., 2023; Majumdar et al., 2022; Sundaresan et al., 2024), videos (Cai et al., 2023, 2024c), or trajectory sketches (Gu et al., 2023; Wang et al.). Compared to traditional imitation learning, GCIL provides a more explicit target during training, reducing the complexity of modeling the entire behavior space and making the policy steerable during inference.

Humans often use language to express their intentions, leading many studies (Brohan et al., 2022; Driess et al., 2023; Padalkar et al., 2023; Wang et al., 2023c) to focus on learning language-conditioned policies. However, language goals are often ambiguous and struggle to capture spatial details (Cai et al., 2024b; Gu et al., 2023). In tasks like navigation (Majumdar et al., 2022) and object manipulation (Wang et al.), image-based goal modalities have been explored, where users provide target images to guide the policy. While effective at conveying spatial information, images often over-specify details, making policies sensitive to irrelevant factors like lighting, object appearance, or background textures. Sundaresan et al. (2024) addresses these issues by replacing the image with hand-drawn sketches. However, generating sketches that align with the current state and the desired goal is non-trivial. Gu et al. (2023) further proposes using trajectory sketches for finer control and better generalization across tasks, but this approach is not applicable in partially observable 3D worlds.

ROCKET-1 (Cai et al., 2024b) tackles interaction problems in 3D worlds by training a visuomotor policy to identify interaction targets based on semantic segmentations in the visual context. While it resolves traditional goal images' ambiguity and generation challenges, it relies on SAM-2 (Ravi et al., 2024) to track goals and segment them during inference, severely limiting real-time performance. We propose a cross-view segmentation-conditioned policy that, unlike ROCKET-1, enables the policy to align the goal across camera views by itself and removes the need for real-time segmentation.

Hindsight Trajectory Relabeling. There are two main approaches to collecting labeled trajectory data: (1) providing instructions for contractors to collect trajectories in real-time, ensuring a causal link between actions and labels but incurring high costs and limited scalability; and (2) gathering large amounts of trajectories and generating labels through post-processing, known as *hindsight trajectory relabeling*. While the first approach (Lynch et al., 2023; Padalkar et al., 2023) produces higher-quality data, its cost constraints have led most research to adopt the second, more scalable one. Andrychowicz et al. (2017) was the first to reinterpret a trajectory's behavior using its final frame, greatly improving data utilization and inspiring subsequent research on goal-conditioned policies. Lifshitz et al. (2023) extended the approach by using the last 16 frames of a trajectory as the more expressive goal. Sundaresan et al. (2024) introduced hand-drawn sketches as a goal modality, reducing semantic ambiguity. Gu et al. (2023) converted

the robotic end-effector moving sketch to a 2D image as the goal, providing richer procedural details. Cai et al. (2024b) introduced backward trajectory relabeling, which first identifies interaction objects and events, then utilizes object tracking models (Ravi et al., 2024) to generate frame-level segmentations. This data supports training segmentation-conditioned policies. Our paper explores using this dataset to train policies for cross-view goal alignment.

3. Methods

In this section, we first introduce the problem of cross-view segmentation-conditioned policy, discussing it from the perspective of imitation learning. Next, we describe the process of generating cross-view trajectories annotated with semantic segmentation. We then present two auxiliary objectives designed to enhance cross-view object alignment in 3D scenes: the *cross-view consistency loss* and the *target visibility loss*. Finally, we detail the architecture of ROCKET-2 and outline the overall optimization objectives.

Problem Statement. Our goal is to learn a goal-conditioned visuomotor policy, which allows humans to specify goal objects for interaction using semantic segmentation across camera views. Formally, we aim to learn a policy $\pi_{\text{cross}}(a_t|o_{1:t}, \{o_g, m_g\}, c_g)$, where a_t represents the action at time t, c_g denotes the type of interaction. In the Minecraft environment, an action corresponds to raw mouse and keyboard inputs. $o_t \in \mathbb{R}^{H\times W\times 3}$ denotes the environment observation at time t, and $o_g \in \mathbb{R}^{H\times W\times 3}$ represents an observation of the local environment from a specific camera view. Generally, o_g and o_t have some visual content overlap. $m_g \in \{0,1\}^{H\times W\times 1}$ is a segmentation mask for o_g, highlighting the target object within the camera view o_g. During inference, users select a view o_g containing the desired object from historical observations returned by the environment and generates its corresponding semantic segmentation m_g. To train such visuomotor policy, we assume access to a dataset $\mathcal{D}_{\text{cross}} = \{c^n, (o_t^n, a_t^n, m_t^n)_{t=1}^{L(n)}\}_{n=1}^N$ consisting of N successful demonstration episodes, $L(n)$ is the length of episode n. **Within each episode, if m_t is non-empty, all (o_t, m_t) pairs indicate the same object.** Consequently, we can arbitrarily pick one observation frame as the goal view condition for the entire trajectory.

Cross-View Dataset Generation. Without loss of generality, we use the Minecraft world as an example to illustrate the data generation process. Manually collecting datasets that meet the requirements is highly expensive. Thus, we employ the *backward trajectory relabeling* technique proposed in Cai et al. (2024b) to automate the annotation of the OpenAI Contractor Dataset (Baker et al., 2022), which consists of free-play trajectories from human players: $\mathcal{D}_{\text{raw}} = \{(o_t^n, a_t^n)_{t=1}^{L(n)}\}_{n=1}^N$. Specifically, for any given episode n, we first detect all frames o_j^n where interaction events occur, identify the interaction type c_j^n, and localize the interacted object near frame j using bounding boxes and point-based prompts. The SAM-2 (Ravi et al., 2024) model is then employed to generate the segmentation mask m_j^n for the object. Starting from frame j, we traverse the trajectory backward and use the SAM-2 model to continuously generate segmentation masks for the object in real-time until either a new interaction event is encountered or a maximum tracking length is reached. Let i denote the end frame. The resulting trajectory clip is then added to the training dataset: $\mathcal{D}_{\text{cross}} \leftarrow \mathcal{D}_{\text{cross}} \cup \{c_j, (o_t^n, a_t^n, m_t^n)_{t=i}^j\}$. This ensures that every extracted clip is associated with a consistent interaction intent. The generated data encompasses the fundamental interaction types in Minecraft, including *use*, *break*, *approach*, *craft*, and *kill entity*. Among these, *approach* is a unique event, identified by detecting trajectory clips where the displacement exceeds a specified threshold. The object located at the center of the clip's final frame is designated as the goal of the *approach* event.

Cross-View Consistency Loss. Accurately interpreting the cross-view goal requires the policy to possess cross-view visual object alignment ability in 3D scenes. To achieve this, the model must fully exploit visual cues from different camera views, such as scene layout and landmark buildings, while being robust to challenges like occlusion, shape variations, and changes in distance. We observe that relying solely on behav-

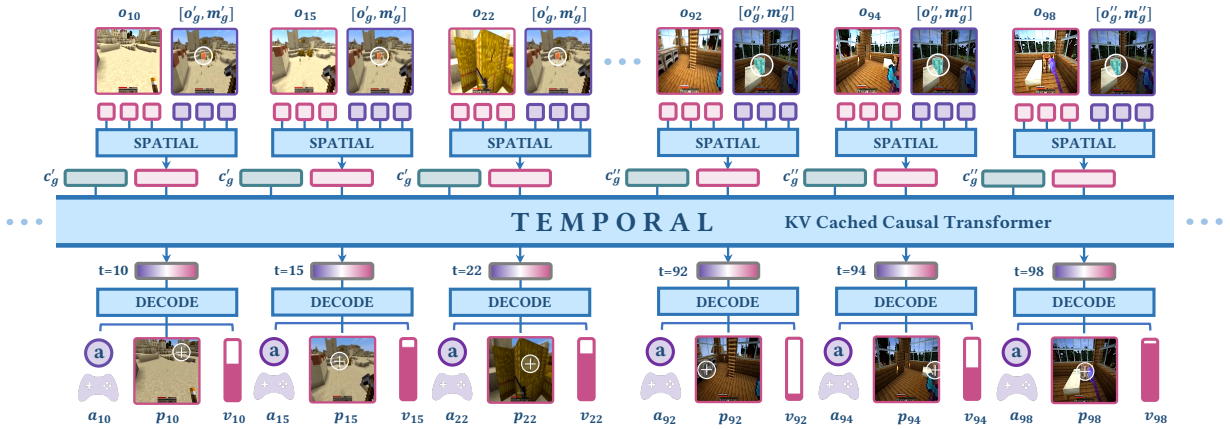

Figure 2 | **ROCKET-2 Architecture.** It consists of three parts: (1) a non-causal transformer for spatial fusion, which establishes the relationship between the agent's and human's camera views; (2) a causal transformer for temporal fusion, ensuring consistency for goal tracking; (3) a decoder module, made of a feedforward neural network (FFN), which predicts goal-related visuals cues and actions.

ior cloning loss (Pomerleau, 1988) is insufficient. Therefore, we propose a *cross-view consistency loss*. Since the segmentation across different camera views corresponds to the same object, we train the model to condition on the segmentation from one camera view to generate the segmentation for another camera view, thereby directly enhancing the model's 3D spatial perception. To reduce computational complexity, we opt to predict the centroid of the segmentation mask instead of the complete mask, formally expressed as: $\pi_{\text{cross}}(p_t \mid o_{1:t}, \{o_g, m_g\}, c_g)$, where

$$p_t = \frac{\sum_{i=1}^{H}\sum_{j=1}^{W}(i,j)\cdot m_t(i,j)}{\sum_{i=1}^{H}\sum_{j=1}^{W} m_t(i,j)}. \tag{1}$$

It is worth noting that incorporating the historical observation sequence $o_{1:t-1}$ as input is essential, especially when there is limited shared visual content between o_t and o_g. *This historical sequence acts as a smooth bridge to facilitate alignment.* Since the goal object represented by the segmentation corresponds to the target of the policy's interaction, this auxiliary task aligns the policy's actions with its visual focus, effectively improving task performance.

Target Visibility Loss. Due to the partial observability in 3D environments, it is common for target objects in interaction trajectories to disappear from the field of view and reappear later. During such intervals, the segmentation mask for the missing object is empty. To leverage this information, we propose training the model to predict whether the target object is currently visible, formulated as: $\pi_{\text{cross}}(v_t \mid o_{1:t}, \{o_g, m_g\})$, where v_t is a binary indicator for empty segmentation masks. On the one hand, accurately predicting object visibility helps the policy better match the target object, avoiding a simple appearance similarity measurement between two frames. On the other hand, visibility information guides the policy to make more reasonable decisions, such as confidently approaching the goal object when it is visible or actively adjusting its camera to explore when the target is absent.

ROCKET-2 Architecture. Lets a training trajectory n be denoted as $(c_g, \{o_t, m_t\}_{t=1}^{L(n)})$. A cross view index g is sampled from $\{i | i \in [1, L(n)], m_i \neq \phi\}$. We resize all visual observations and their segmentation masks to 224×224. For encoding the visual observation o_t, we utilize a DINO-pretrained (Caron et al., 2021) 3-channel ViT-B/16 (Dosovitskiy et al., 2020) (16 is the patch size), which outputs a token sequence of length 196, denoted as $\{\hat{o}_t^i\}_{i=1}^{196}$. Similarly, we encode the segmentation mask m_t using a 1-input-channel ViT-tiny/16, yielding $\{\hat{m}_t^i\}_{i=1}^{196}$. The ViT-base/16 encoder is frozen during training for efficiency, while the ViT-tiny/16 is trainable. To ensure spatial alignment, we fuse the cross-view condition (o_g, m_g) by concatenating the feature channels:

$$h_g^i = \text{FFN}(\text{concat}([\hat{o}_g^i \parallel \hat{m}_g^i])). \tag{2}$$

Given the ability of self-attention mechanisms to capture spatial details across views, we concatenate the token sequences from two views into a single sequence of length 392. A non-causal Transformer encoder module is applied (Vaswani et al., 2017) for spatial fusion, obtaining a frame-level representation x_t:

$$x_t \leftarrow \text{SpatialFusion}(\{\hat{o}_t^i\}_{i=1}^{196}, \{h_g^i\}_{i=1}^{196}). \quad (3)$$

Subsequently, we leverage a causal TransformerXL (Dai et al., 2019) architecture to capture temporal information across the sequence:

$$f_t \leftarrow \text{TransformerXL}(\{x_i\}_{i=1}^{t}, c_g). \quad (4)$$

Finally, a simple linear layer maps f_t to predict the action $\hat{a}_t$, centroid $\hat{p}_t$, and visibility $\hat{v}_t$. The loss function for episode n is defined as:

$$\mathcal{L}(n) = \sum_{t=1}^{L(n)} -a_t^n \log \hat{a}_t^n - p_t^n \log \hat{p}_t^n - v_t^n \log \hat{v}_t^n. \quad (5)$$

4. Experiments

We aim to address the following questions: **(1)** How does ROCKET-2 perform in terms of both accuracy and efficiency during inference? **(2)** Can ROCKET-2 follow the intention of a human from a cross-camera view? **(3)** Under what circumstances does ROCKET-2 fail to work? **(4)** How important are landmarks in cross-view goal alignment? **(5)** Can ROCKET-2 interpret goal views from cross-episode scenarios? **(6)** Which modules contribute effectively to training ROCKET-2?

4.1. Experimental Setup

Implementation Details. We present the model architecture, hyperparameters, and optimizer configurations of ROCKET-2 in Table 1. During training, each trajectory is divided into segments of length 128 to reduce memory requirements. We initialize the view backbone that is used to encode RGB images with DINO weights and freeze it for training efficiency. During inference, ROCKET-2 can access up to 128 key-value attention caches of past observations. Most training parameters follow those from prior works such as Baker et al. (2022); Cai et al. (2024b).

Figure 3 | **The Evaluation Metric is Spatial-Sensitive.** ✓ and × indicate the correct and incorrect objects for interaction, respectively. None of the task configurations were seen during the training.

Table 1 | **Detailed Training Hyperparameters.**

Hyperparameter	Value
Input Image Size	224×224
Hidden Dimension	1024
View Backbone	ViT-base/16 (DINO-v1)
Mask Backbone	ViT-tiny/16 (1-channel)
Spatial Transformer	PyTorch Transformer
Number of Spatial Blocks	4
Temporal Transformer	TransformerXL
Number of Temporal Blocks	4
Trajectory Chunk size	128
Optimizer	AdamW
Learning Rate	0.00004

Environment and Benchmark. We use the original Minecraft 1.16.5 (Cai et al., 2024a; Guss et al., 2019; Lin et al., 2023) as our testing environment, which accepts mouse and keyboard inputs and outputs a 640×360 RGB image at each step. Following Cai et al. (2024b), we employ the Minecraft Interaction Benchmark to evaluate the agent's interaction capabilities. This benchmark includes six categories and a total of 12 tasks, covering all basic Minecraft interaction types: *Hunt*, *Mine*, *Interact*, *Navigate*, *Tool*, and *Place*. As this benchmark emphasizes object interaction and spatial localization, its evaluation criteria are more stringent than those in Lifshitz et al. (2023) and Cai et al. (2023). We show some examples in Figure 3. In the *"hunt the sheep in the right fence"* task, success requires the agent to kill the sheep within the right fence, while killing it in the left fence results in failure. Similarly, in the *"place the oak door on the diamond block"* task, success is only achieved if the oak door is adjacent to the

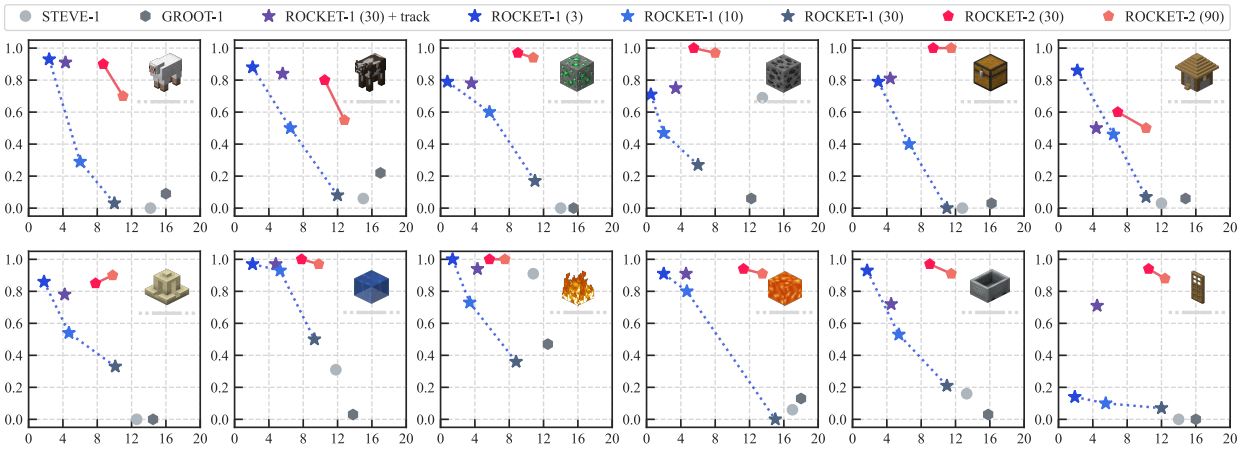

Figure 4 | **Performance-Efficiency Comparison on the Minecraft Interaction Benchmark.** The x-axis represents inference speed (FPS), and the y-axis shows the interaction success rate. Numbers in parentheses indicate the Molmo invocation interval, where larger values mean higher FPS. "+ track" denotes real-time SAM-2 segmentation between Molmo calls, increasing inference time (applicable only to ROCKET-1). In most cases, ROCKET-2 achieves 3× to 6× faster while matching or surpassing ROCKET-1's peak performance.

diamond block on at least one side.

Baselines. We compare our ROCKET-2 with the following instruction-following baselines: (1) STEVE-1 (Lifshitz et al., 2023): An instruction-following agent fine-tuned from VPT (Baker et al., 2022), capable of solving various short-horizon tasks. We use the text-conditioned version of STEVE-1 for comparison. (2) GROOT-1 (Cai et al., 2023): A reference-video-conditioned policy designed for open-ended tasks, trained on 2,000 hours of long-form videos using latent variable models. (3) ROCKET-1 (Cai et al., 2024b): A segmentation-conditioned policy capable of mastering 12 interaction tasks. While it achieves a high interaction success rate, its reliance on SAM-2's real-time tracking during inference creates an efficiency bottleneck.

4.2. Performance-Efficiency Analysis

We demonstrate that cross-view goal specification (implemented as ROCKET-2) significantly improves inference speed while maintaining high interaction success. Following Cai et al. (2024b), we evaluate STEVE-1, GROOT-1, and variants of ROCKET-1 and our ROCKET-2 on the *Minecraft Interaction Benchmark*. Due to the limited spatial reasoning ability, STEVE-1 and GROOT-1 achieve success rates below 20% on most tasks. We illustrate the inference pipeline of ROCKET-series agents in Table 2. Automated evaluation of ROCKETs relies on Molmo (Deitke et al., 2024) and SAM (Ravi et al., 2024) to generate a segmentation mask for the target object in the given views. In general, ROCKET-1 requires object masks for all agent observations, whereas ROCKET-2 only needs one or a few object masks. We observe that ROCKET-1 attains over 80% success with high-frequency Molmo point corrections (every 3 frames) but suffers from slow inference. Lowering Molmo's frequency greatly degrades ROCKET-1's performance, with tasks like "collecting lava" failing entirely at 30-frame intervals. While one can enable SAM's tracking mode to provide dense goal signals, it remains computationally expensive. In contrast, our ROCKET-2 decouples the goal specification from the agent view, it does not need frequent segmentation mask modification

Table 2 | **ROCKET-Series Inference Pipeline Details.** Molmo can pinpoint the target object based on the task prompt. SAM uses the point to generate object mask m_t w.r.t. o_t and supports real-time object tracking.

Model	Inference Pipeline
R1(3)	$\mathbf{m_1} \leftarrow \text{SAM}(o_1, \text{Molmo}(o_1, \text{prompt}))$ $\pi_{R1}(a_t \vert o_1, \mathbf{m_1}, o_2, o_3, o_4, \mathbf{m_4}, o_5, o_6, o_7, \mathbf{m_7}, \cdots)$
R1(30)+track	$\mathbf{m_{1:30}} \leftarrow \text{SAM}(o_{1:30}, \text{Molmo}(o_1, \text{prompt}))$ $\pi_{R1}(a_t \vert o_1, \mathbf{m_1}, o_2, \mathbf{m_2}, o_3, \mathbf{m_3}, o_4, \mathbf{m_4}, \cdots)$
R2(60)	$\mathbf{m_1} \leftarrow \text{SAM}(o_1, \text{Molmo}(o_1, \text{prompt}))$ $\pi_{R2}(a_t \vert o_1, \mathbf{m_1}, o_2, o_3, o_4, \cdots, o_{60}, o_{61}, \mathbf{m_{61}}, \cdots)$

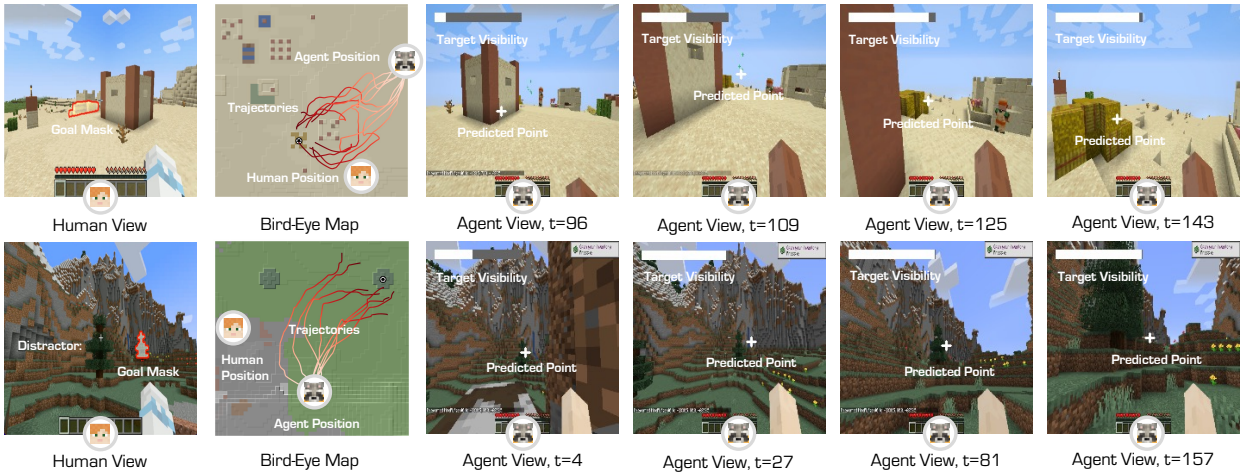

Figure 5 | **Case Study of Human-Agent Interaction.** We demonstrate how a human interacts with ROCKET-2, leveraging its spatial reasoning abilities. **(Top Row)** The human specifies a hay bale ( ) that is not visible to ROCKET-2. By exploring the area around the visible landmark (house), ROCKET-2 successfully locates the goal. **(Bottom Row)** The human specifies a target tree in the presence of a tree distractor. ROCKET-2 accurately identifies the correct tree by reasoning about spatial relationships and landmarks. The agent's trajectories are visualized in bird's-eye view maps.

and can autonomously track the target object. It achieves comparable or superior performance to ROCKET-1 with a 3× to 6× inference speedup.

4.3. Intuitive Human-Agent Interaction

In Figure 5, we present two case studies illustrating ROCKET-2 interprets human intent under the cross-view goal specification interface. The first case (top row) involves a task requiring the agent to approach a hay bale ( ) located behind a house ( ). From the human view, both the house and the hay bale are visible, whereas ROCKET-2 initially observes only the house. A key challenge arises from the differing camera views: the human and ROCKET-2 perceive the scene from opposite sides of the house. To analyze the agent's behavior, we visualize both its camera views and its trajectories on a bird's-eye map. We observe that ROCKET-2 effectively infers the hay bale's potential location and successfully navigates toward it. This is reflected in the increasing target visibility score and the movement of the predicted point. Interestingly, the bird's-eye view reveals that ROCKET-2 approaches the target from both sides of the house, demonstrating diversity in route selection. The second case (bottom row) showcases ROCKET-2 's ability to distinguish between a distractor and the human-specified goal object, despite their visual similarity. This finding highlights that ROCKET-2 's spatial reasoning extends beyond object appearance and incorporates scene alignment for goal inference.

4.4. Analyzing Failure Cases

We analyze failure cases in ROCKET-2 's task execution and identify three main issues: (1) *Prediction Drift*: When pursuing distant targets for extended periods, the predicted point gradually shifts away from the object. Since distant targets rely on temporal consistency from memory for recognition, but the model was only trained with memory lengths up to 128, it faces long-sequence generalization challenges during inference. (2) *Distance Perception Error*: When the goal and agent camera views differ significantly, the agent sometimes stops one step before reaching the target, leading to interaction failure. We observe that updating the goal view to the agent's current observation alleviates this issue, likely due to the greater cross-view discrepancy encountered during inference compared to training. (3) *Action Jitter*: When inferring the original version of ROCKET-2, we observe significant action jitter, this could cause failures in precise interactions such as placing blocks. We find that incorporating previous actions during training and inference

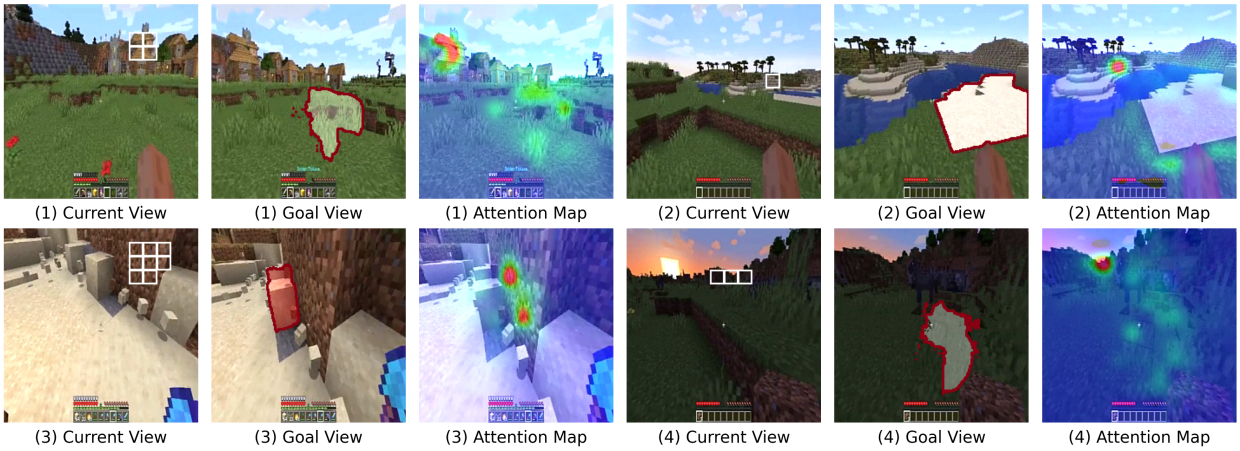

Figure 6 | **Visualization Analysis of Cross-View Alignment.** The vision patches (identified by white grid) represent a chosen background landmark in the agent's current view (instead of the goal object). We generated an attention map with the **spatial fusion transformer** using these patches as queries and the goal view patches as keys and values. We found that ROCKET-2 perfectly aligned with the selected landmarks across views.

greatly improves action smoothness.

4.5. Landmarks Attention Visualization

Prominent non-goal objects, referred to as "landmarks", play a crucial role in assisting humans or agents in localizing goal objects within a scene. For instance, when multiple objects with similar appearances are present, spatial relationships between the goal and landmarks can aid in distinction. In this subsection, we aim to explore whether ROCKET-2 implicitly learns landmark alignment by visualizing the attention weights of its spatial transformer.

Specifically, we prepare a current view observation and a third view with goal segmentation. Before being fed into the spatial transformer, both views are encoded into $14 \times 14 = 196$ tokens: $\{\hat{o}_t^i\}_{i=1}^{196}$ and $\{h_g^i\}_{i=1}^{196}$ (notations are consistent with Sec. 3). We inspect the softmax-normalized attention map of the first self-attention layer in the spatial transformer, denoted as $\{a_{i,j}\}_{i,j=1}^{392}$, where $a_{i,197:392}$ represents the attention map generated by using patch i from the current view as the query and all patches from the third view as keys and values. This map is overlaid on the third view (goal view) to reflect its responsiveness to patch i in the current view. Since landmarks may span multiple patches, we aggregate the response maps of different patches to form the final attention map $\{m_i\}_{i=1}^{196}$:

$$m_i = \frac{1}{|L|} \sum_{x \in L} a_{x,i+196}, \tag{6}$$

where L denotes the set of patches in the current view representing a specific landmark. Notably, the selected landmarks do not overlap with the goal segmentation. As shown in Figure 6, we present four sets of data covering *villages*, *plains*, *deserts*, and *forest* terrains. In the first plot, the white grid indicates the selected landmark patches, while the third plot shows the third view response to the chosen landmarks. Our findings reveal that ROCKET-2 effectively matches cross-view consistency even under significant geometric deformations and distance variations. Surprisingly, in data point (4), even subtle forest depressions at a considerable distance are accurately matched.

4.6. Cross-Episode Goal Alignment

We observe that ROCKET-2 exhibits cross-episode generalization capabilities. As shown in Figure 7, the selected goal views come from different episodes, each generated with a unique world seed. In the top-row example, the goal view is from a "bridge-building" episode set in the *savanna* biome, where the player is placing a dirt block to build the bridge. After feeding

Figure 7 | **Cross-Episode Generalization.** The goal view does not exist within the agent's world but originates from a different episode. We observe that the agent attempts to infer the semantic information underlying the goal specification.

Table 3 | **Ablation Study on Auxiliary Objectives.** The final loss function for each row is the cumulative sum of all loss functions from the preceding rows.

Training Variants				**Avg.**
behavior cloning	0.52	0.78	0.65	0.65
+target visibility	0.63	0.83	0.68	0.71
+cross-view consistency	**0.85**	**0.97**	**1.00**	**0.94**

forward the goal view, we place ROCKET-2 in a *forest* biome and observe its behavior. We find that it first exhibits pillar-jumping behavior, and after placing many blocks, it begins to build the bridge horizontally. Although it ultimately failed to build the perfect bridge, the emergent behavior still indicates that ROCKET-2 attempts to understand the underlying semantic information when there is no landmark match across views. In the bottom row, the goal view is taken from a Minecraft creative mode, observing a house from the sky— a view never seen during training. We find that ROCKET-2 explores its environment and successfully identifies a visually similar house. This demonstrates ROCKET-2 's robustness to a variety of goal views.

4.7. Ablation Studies on Auxiliary Objectives

To evaluate the impact of auxiliary losses on model performance, we define three variants: (1) only *behavior cloning loss*, (2) + *target visibility loss*, and (3) the full version with + *cross-view consistency loss*. We conduct experiments on three tasks: *Navigate to House in a Village*( ), *Mine Emerald*( ), and *Interact with the Left Chest*( ). We find that the BC-only variant achieves an average success rate of only 65%, demonstrating that the action signal is insufficient for learning spatial alignment. Adding *target visibility loss* improves performance by 6%, while further incorporating *cross-view consistency loss* boosts the success rate to 94%. This proves that leveraging temporal consistency and introducing vision-based auxiliary losses can greatly enhance cross-view goal alignment and inference-time decision-making capabilities.

5. Conclusions and Limitations

To improve human-agent interaction in embodied worlds, we propose a cross-view goal specification approach. Since behavior cloning alone fails to align the agent with human views, we introduce cross-view consistency and target visibility losses to enhance alignment. ROCKET-2 achieves state-of-the-art performance on the *Minecraft Interaction Benchmark* with a 3× to 6× efficiency boost. Visualizations and case studies validate our method. We also observe that ROCKET-2 struggles with visual reasoning when the discrepancy between the agent's and human's views is large. As memory length increases, the predicted points exhibit noticeable drift. We attribute this to the relabeled dataset, whose memory window is short and view variation is limited. Enhancing data quality could help address this issue.

6. Acknowledgements

This work is funded by the National Science and Technology Major Project #2022ZD0114902.

References

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. 3, 4, 6, 7

S. Borse, M. Klingner, V. R. Kumar, H. Cai, A. Almuzairee, S. Yogamani, and F. Porikli. X-align: Cross-modal cross-view alignment for bird's-eye-view segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3287–3297, 2023. 3

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. 1, 3

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. 1, 3

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. 2, 3, 6, 7

S. Cai, Z. Mu, K. He, B. Zhang, X. Zheng, A. Liu, and Y. Liang. Minestudio: A streamlined package for minecraft ai agent development. 2024a. URL https://api.semanticscholar.org/CorpusID:274992448. 2, 6

S. Cai, Z. Wang, K. Lian, Z. Mu, X. Ma, A. Liu, and Y. Liang. Rocket-1: Master open-world interaction with visual-temporal context prompting. *arXiv preprint arXiv:2410.17856*, 2024b. 1, 2, 3, 4, 6, 7

S. Cai, B. Zhang, Z. Wang, H. Lin, X. Ma, A. Liu, and Y. Liang. Groot-2: Weakly supervised multi-modal instruction following agents. *arXiv preprint arXiv:2412.10410*, 2024c. 3

M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 5

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. 6

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. 7

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. 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. 5

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. 1, 3

S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song. Clip on wheels: Zero-shot object navigation as object localization and exploration. *arXiv preprint arXiv:2203.10421*, 3(4): 7, 2022. 3

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. 2, 3

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. 3, 6

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. 3

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. 1

J. Jiang, Y. Yang, Y. Deng, C. Ma, and J. Zhang. Bevnav: Robot autonomous navigation via spatial-temporal contrastive learning in bird's-eye view. *IEEE Robotics and Automation Letters*, 2024. 3

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. 3, 6, 7

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. 6

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. 1, 3

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. 3

A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, A. Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. 3

T. Pearce and J. Zhu. Counter-strike deathmatch with large-scale behavioural cloning. In *2022 IEEE Conference on Games (CoG)*, pages 104–111. IEEE, 2022. 3

D. A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988. 3, 5

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. 2, 3, 4, 7

M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347, 2019. 2, 3

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. 1, 2, 3

A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, 2017.

URL https://api.semanticscholar.org/CorpusID:13756489. 2, 6

C. Wang, L. Fan, J. Sun, R. Zhang, L. Fei-Fei, D. Xu, Y. Zhu, and A. Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. In *7th Annual Conference on Robot Learning*. 3

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, 2023a. URL https://api.semanticscholar.org/CorpusID:258887849. 1

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, 2023b. 1

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*, 2023c. 1, 3

D. Xu, J. Chen, C. Liang, Z. Wang, and R. Hu. Cross-view identical part area alignment for person re-identification. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2462–2466, 2019. doi: 10.1109/ICASSP.2019.8683137. 3

Q. Zhao, L. Zhang, B. He, H. Qiao, and Z. Liu. Zero-shot object goal visual navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2025–2031. IEEE, 2023. 3