Steering Visuomotor Policy in Open Worlds via Cross-View Goal Alignment

Shaofei Cai[1], **Zhancun Mu**[1], **Anji Liu**[2], **Yitao Liang**[1]

[1]Institute for Artificial Intelligence, Peking University
[2]School of Computing, National University of Singapore
{caishaofei, muzhancun}@stu.pku.edu.cn, anjiliu@nus.edu.sg, yitaol@pku.edu.cn

Abstract

We aim to develop a goal specification method that is semantically clear, spatially sensitive, domain-agnostic, and intuitive for human users to guide agent interactions in 3D environments. Specifically, we propose a novel cross-view goal alignment framework that allows users to specify target objects using segmentation masks from their 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 $3\times$ to $6\times$ compared to ROCKET-1. We show that ROCKET-2 can directly interpret goals from human camera views, enabling better human-agent interaction. Remarkably, ROCKET-2 demonstrates zero-shot generalization capabilities: despite being trained exclusively on the Minecraft dataset, it can adapt and generalize to other 3D environments like Doom, DMLab, and Unreal through a simple action space mapping.

Code — https://github.com/CraftJarvis/ROCKET-2

1 Introduction

Learning an agent to achieve desired goals is a long-standing challenge in the field of decision making, with significant implications for the development of robots (Brohan et al. 2022, 2023; Jang et al. 2022) and virtual players (Wang et al. 2023c,d,b). 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, many 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 Sundaresan et al. (2024); Cai et al. (2025a), 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 generalization 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. 2025a) specifies the objects to interact with by applying segmentation masks to the agent's perception. These methods greatly improved the expressiveness of spatial relationships and generalization across tasks. However, both trajectory sketches and object mask are closely tied to the agent's real-time 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 observation. The agent is then trained to align with human intent and take actions based on its pixel perception 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. 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. 2025a): *The target object remains consistent across camera views in a short interaction window*. 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 given its camera view, while target visibility loss helps the agent determine whether the target object 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

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

the camera changes, eliminating the need for SAM's (Ravi et al. 2024) per-frame goal instance segmentation, speeding up inference $3\times$ to $6\times$ compared to ROCKET-1. We observe that ROCKET-2 can interpret intentions from a human's camera view and make decisions to achieve expected goals in the 3D world. Notably, by simply mapping Minecraft's action space to that of other 3D environments, such as Doom (Kempka et al. 2016), DMLab (Beattie et al. 2016), and Unreal (Zhong et al. 2024), we are surprised to find that ROCKET-2 *could successfully perform basic navigation and object interaction tasks in a zero-shot generalization manner.* This demonstrates that our cross-view alignment scheme is domain-agnostic and holds large potential for further scaling across a wider range of environments. Our contributions are threefold:

- We introduce a user-friendly interface that allows humans to specify goals using instance mask from human users' own views.
- By introducing *cross-view consistency loss* and *target visibility loss*, we train ROCKET-2, which autonomously tracks the goal, eliminating the need for per-frame segmentation, and greatly speeding up inference.
- We demonstrate that ROCKET-2, trained solely on Minecraft data, can zero-shot generalize to other 3D environments, which opens new avenues for discovering universal decision representations in 3D environments.

2 Related Works

2.1 Partial Observability in 3D Open World

We address policy learning in partially observable 3D environments (Savva et al. 2019; Cai et al. 2024), where the agent perceives only egocentric views and must actively explore to locate key objects (Pearce and Zhu 2022). Some methods leverage 3D point clouds for global context (Huang et al. 2023; Jiang et al. 2024), but such data is often unavailable. More commonly, memory-based architectures are used to aggregate past observations: RNNs help avoid redundant exploration (Zhao et al. 2023; Gadre et al. 2022), and TransformerXL enables long-horizon planning in Minecraft (Baker et al. 2022; Guss et al. 2019). To mitigate partial observability, some works (Krantz et al. 2023, 2022) use instance images to specify goals, but these require unoccluded, centered views where the object occupies most of the image—constraints that limit their applicability. Instead, we require the policy to align object references across egocentric views, enabling robust target tracking under camera motion. While cross-view alignment has been explored in computer vision for BEV segmentation (Borse et al. 2023) and re-identification (Xu et al. 2019), we are the first to apply it to open-world policy learning.

2.2 Goal-Conditioned Imitation Learning

GCIL optimizes conditional policies via behavior cloning (Pomerleau 1988), using goal representations such as language (Brohan et al. 2022, 2023; Lynch et al. 2023), images (Majumdar et al. 2022; Lifshitz et al. 2023; Sundaresan et al. 2024), videos (Cai et al. 2023, 2025b), or trajectory sketches (Wang et al. 2023a; Gu et al. 2023). Compared to standard imitation learning, GCIL provides clearer targets, simplifies behavior modeling, and improves policy controllability. Language goals are commonly used (Padalkar et al. 2023; Driess et al. 2023; Wang et al. 2023d) but often fail to convey spatial details (Cai et al. 2025a; Gu et al. 2023). Image goals (Majumdar et al. 2022; Wang et al. 2023a) better ground spatial intent but are sensitive to lighting, textures, and other irrelevant features. Sketches (Sundaresan et al. 2024) reduce visual sensitivity but are difficult to generate. Trajectory sketches (Gu et al. 2023) improve control and generalization, but assume full observability, limiting their use in 3D worlds. We instead propose aligning goal-relevant objects across views, enabling spatially grounded yet robust goal conditioning in partially observable environments.

3 Method

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.

3.1 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 interaction event, such as *break item*, *kill entity*, and *approach*. 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 share 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 generate 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 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.*

3.2 Cross-View Dataset Preparation

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. (2025a) to automate the

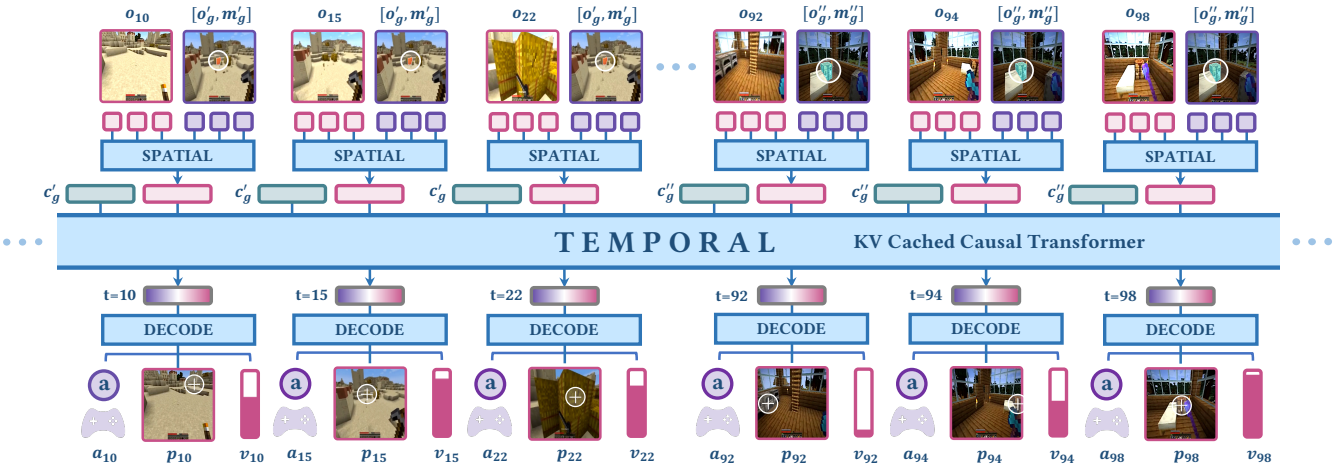

Figure 1: Overview of our approach. We address the challenge of steering visuomotor policy via cross-view goal alignment. Our method allows humans to specify goals from their camera view while the agent acts based on its own observations.

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*.

3.3 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 behavior 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 $p_t = \frac{\sum_{i=1}^{H}\sum_{j=1}^{W}(i,j)\cdot m_t(i,j)}{\sum_{i=1}^{H}\sum_{j=1}^{W} m_t(i,j)}$. It is worth noting that incorporating the historical observations $o_{1:t-1}$ 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 mask 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.

3.4 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 reasonable decisions, such as confidently approaching the goal when it is visible or actively adjusting its camera to explore when the target is absent.

3.5 ROCKET-2 Architecture

Let 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 $\{\hat{o}_t^i\}_{i=1}^{196}$. Inspired by (Zhong et al. 2025), we encode the

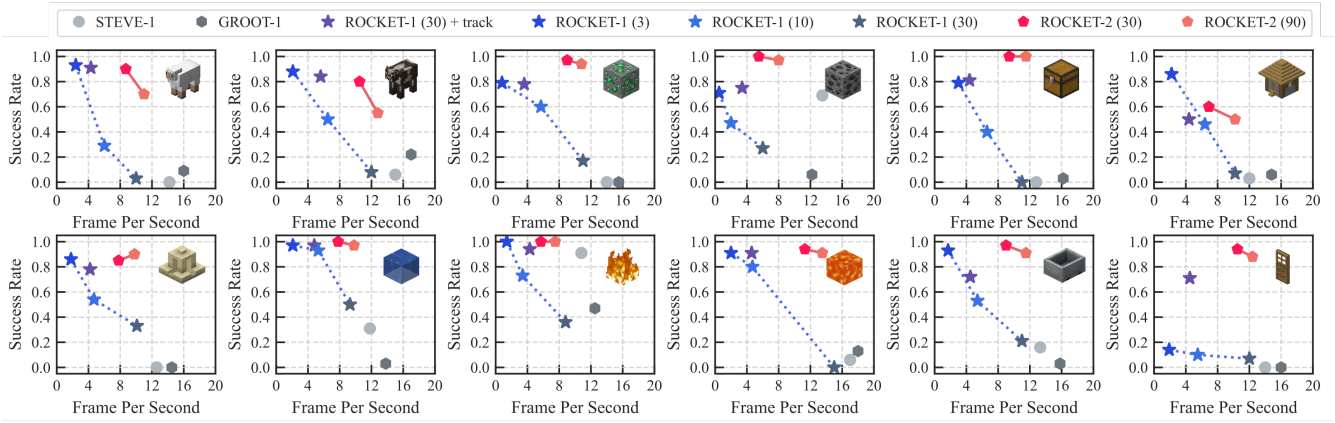

Figure 2: Minecraft Interaction Benchmark Results. Our ROCKET-2 achieves 3× to 6× faster inference, matching or surpassing ROCKET-1's peak performance in most tasks. Numbers in parentheses indicate Molmo invocation interval (steps); larger values mean higher FPS. + *track* denotes real-time SAM-2 tracking between Molmo calls.

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:

$$h_g^i = \text{FFN}(\text{concat}([\hat{o}_g^i \parallel \hat{m}_g^i])). \quad (1)$$

Given the ability of self-attention mechanisms to capture spatial details across views, we concatenate the token sequences from two views into a 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:

$$x_t \leftarrow \text{SpatialFusion}(\{\hat{o}_t^i\}_{i=1}^{196}, \{h_g^i\}_{i=1}^{196}). \quad (2)$$

Then, we leverage a causal TransformerXL (Dai et al. 2019) architecture to capture temporal information among frames:

$$f_t \leftarrow \text{TransformerXL}(\{x_i\}_{i=1}^{t}, c_g). \quad (3)$$

Finally, a light network maps f_t to predict action $\hat{a}_t$, centroid $\hat{p}_t$, and visibility $\hat{v}_t$. The loss function for episode n follows:

$$\mathcal{L}(n) = \sum_{t=1}^{L(n)} -a_t^n \log \hat{a}_t^n - p_t^n \log \hat{p}_t^n - v_t^n \log \hat{v}_t^n. \quad (4)$$

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 intent of a human from a cross-camera view in Minecraft? **(3)** Can ROCKET-2 adapt and generalize to other 3D environments? **(4)** How important are landmarks in cross-view goal alignment for ROCKET-2? **(5)** Can ROCKET-2 interpret goal views from cross-episode scenarios? **(6)** Which modules contribute effectively to training ROCKET-2?

4.1 Experimental Setup

We include all the details, such as hyperparameters, benchmarks, and case studies, in the appendix.

Figure 3: The Evaluation Metric is Spatial-Sensitive. ✓ and × indicate the correct and incorrect objects for interaction, respectively. None of the task is trained.

Environment and Benchmark We use Minecraft v1.16.5 (Guss et al. 2019; Cai et al. 2024) as the testing environment, which accepts mouse and keyboard inputs and returns a 640 × 360 RGB image per step. Following Cai et al. (2025a), 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 present three examples in Figure 3, more can be found in appendix. In the *"hunt the sheep in the right fence"* task, success requires the agent to kill the sheep within the right fence, while killing sheep in the left fence results in failure.

Baselines We compare our ROCKET-2 with the following instruction-following baselines: (1) STEVE-1 (Lifshitz et al. 2023): A text-conditioned agent fine-tuned from VPT (Baker et al. 2022), capable of solving various short-horizon tasks. (2) GROOT-1 (Cai et al. 2023): A video-conditioned policy designed for open-ended tasks, implemented with a VAE network. (3) ROCKET-1 (Cai et al. 2025a): A mask-conditioned policy capable of mastering 12 interaction tasks. Taking the *hunt right sheep* task as an example, we present the inference pipelines in Figure 5, where Molmo (Deitke

Figure 4: Case Study of Human-Agent Interaction. We show 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)** Human specifies a target tree in the presence of a tree distractor. ROCKET-2 accurately identifies the correct tree by reasoning about spatial relationships. The trajectories are visualized in bird's-eye view maps.

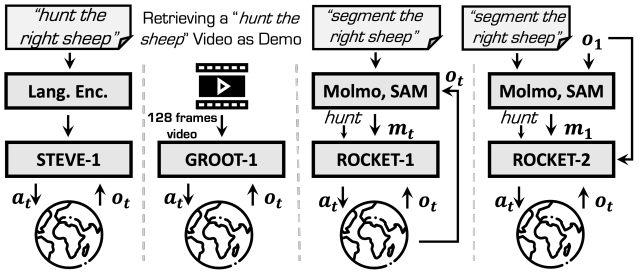

Figure 5: Inference Pipelines on Minecraft Interaction Benchmark. GROOT-1 requires retreving a short video as its condition. ROCKET-1's segmentation is coupled with the agent's real-time observations, whereas ROCKET-2's segmentation is solely tied to the third-person view.

et al. 2024) and SAM (Ravi et al. 2024) are combined to perform text-based instance segmentation. Given that ROCKET-2 relies on a third-person viewpoint for goal specification, in the Minecraft Interaction Benchmark, we set its third-person view to the observation acquired by the agent at the initial step. This view is then largely maintained, undergoing resets only at fixed periods (e.g., every 90 steps). This setup ensures that ROCKET-2 requires no additional privileged information during the inference.

4.2 Performance-Efficiency Analysis

Figure 2 presents our Minecraft Interaction Benchmark results, showcasing various agent configurations and their performance. Specifically, ROCKET-1 (10) performs object segmentation with Molmo + SAM every 10 steps, providing no masks for the intermediate 9 steps; its "+ track" variant leverages SAM's more computationally efficient object tracking to causally generate masks for these frames. In contrast, ROCKET-2 utilizes cross-view segmentation, only requiring a single third-person view segmentation, with ROCKET-2 (90) resetting this view every 90 steps. Our initial observations highlight that STEVE-1 and GROOT-1 exhibit success rates below 20% across most tasks, primarily due to their instructions' limited spatial sensitivity. While ROCKET-1 achieves over 80% success with high-frequency Molmo (every 3 frames), it suffers from slow inference; lowering Molmo's frequency or enabling SAM tracking (though still expensive) significantly degrades its performance. Conversely, our ROCKET-2 agent, by decoupling goal specification from the agent's current view, autonomously tracks targets without frequent mask modifications, achieving comparable or superior performance to ROCKET-1 with a remarkable $3\times$ to $6\times$ inference speedup.

4.3 Intuitive Human-Agent Interaction

In Figure 4, 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

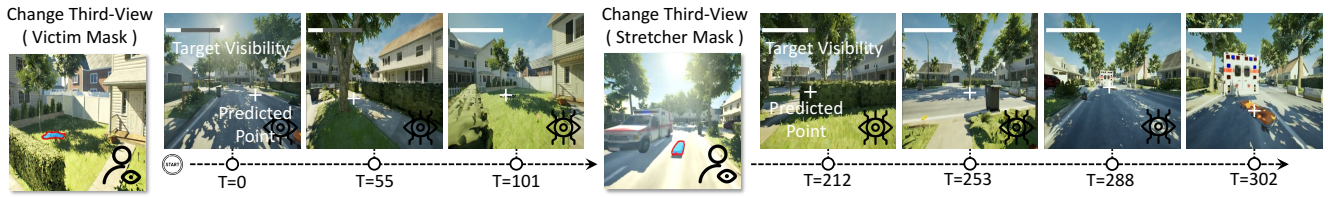

Figure 6: Zero-Shot 3D Environment Generalization. ROCKET-2 can find and transport the victim to the stretcher in the unseen Unreal Engine. We map the *left click*/*right click* actions of Minecraft to the *carry up*/*put down* actions of Unreal to interact with objects, respectively.

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. It highlights that agent's spatial reasoning extends beyond object appearance.

4.4 Zero-Shot 3D Worlds Generalization

We demonstrate that by mapping the action space of Minecraft to unseen 3D games (refer to Table 1), ROCKET-2 achieves zero-shot generalization, despite **being trained solely on the Minecraft dataset**. We attribute this to two key factors: (1) *the DINO-pretrained ViT backbone is frozen during training, preserving its general 3D view perception capability*; and (2) *the learned cross-view goal alignment skill is domain-agnostic*. We evaluated ROCKET-2 in the ATEC 2025 AI and Robotics Challenge [1], which is built in the Unreal Engine and assigns agents to *locate injured people and transport them to stretchers*. The benchmark provides both textual (a long text description) and visual cues (a third-person view) about the locations of the victims. Note that, ROCKET-2 takes in **solely visual cues** (Figure 6). As shown in Table 2, ROCKET-2 outperforms the **strong baseline** provided by the organizers, a Gemma-3 and YOLO-based two-tiered decision system using both textual and visual cues, by 12%, despite not being fine-tuned on the Unreal. Furthermore, the Unreal environment provides observations at a 640×480 resolution, a notable deviation from the 640×360 resolution of the training dataset. This robustly substantiates that cross-view goal alignment enables the acquisition of transferable 3D decision representations within one environment and their successful migration to another. Such capabilities underscore the significant feasibility of developing general-purpose multi-task agents for diverse 3D environments.

4.5 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 abilities.

Minecraft	**DMLab**	**Doom**	**Unreal**
forward $= 1$	$a[3] = 1$	discrete$[2] = 1$	velocity $= +100$
back $= 1$	$a[3] = -1$	discrete$[3] = 1$	velocity $= -100$
attack $= 1$	$a[4] = 1$	discrete$[1] = 1$	pick $= 1$
$\cdots$	$\cdots$	$\cdots$	$\cdots$
yaw $= x$	$a[0] = 4.8x$	continuous $= x$	angular $= x$
pitch $= x$	$a[1] = 2.8x$	/	viewport $= x$

Table 1: Bridging the Minecraft Action Space and Other 3D Games. "/" denotes the masked action.

Agent	ROCKET-2 (Zero-Shot)	YOLO-Gemma 3
Rank	**1/57**	20/57
Success Rate	**38%**	26%

Table 2: Results on ATEC 2025 AI & Robotics Challenge.

Model 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**

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 ones.

[1] https://github.com/atecup/atec2025_software_algorithm

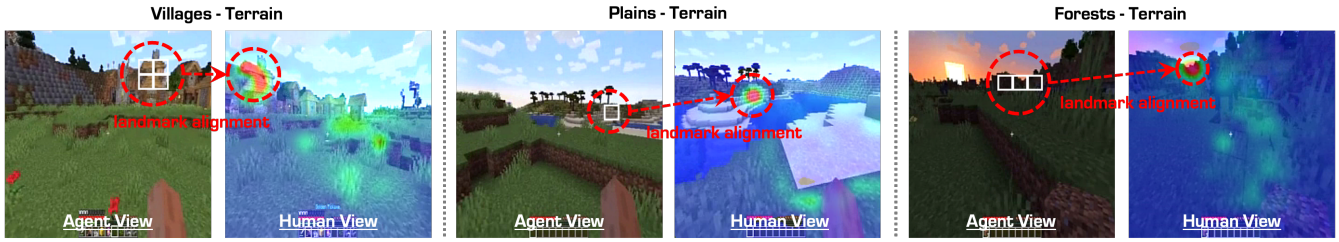

Figure 7: Visualization Analysis of Landmarks' Alignment. The vision patches (identified by white grid) represent a chosen background landmark in the agent's current view (instead of the goal object). We generate an attention map with the **spatial fusion transformer** using these patches as queries and the human view patches as keys and values. We find that ROCKET-2 well aligns selected landmarks across views.

4.6 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 $\{m_i\}_{i=1}^{196}$: $m_i = \frac{1}{|L|}\sum_{x\in L} a_{x,i+196}$, 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 7, we present three sets of data covering *villages*, *plains*, and *forest* terrains. In the left plot, the white grid indicates the selected landmark patches, while the right plot shows the human-view attention 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 the last data point, even subtle forest depressions are accurately matched.

4.7 Cross-Episode Cross-View Goal Alignment

We observe that ROCKET-2 exhibits cross-episode generalization capabilities. As shown in Figure 8, 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 forward the goal view, we place

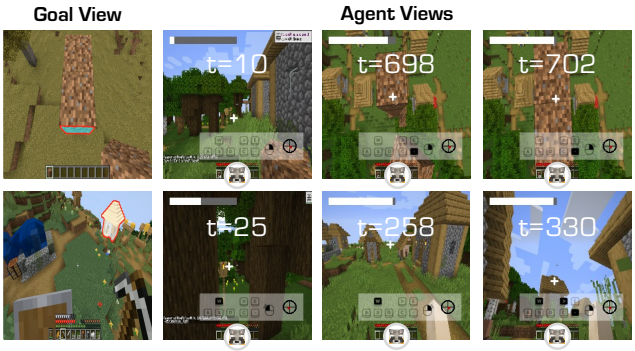

Figure 8: Cross-Episode Generalization. The goal view does not exist within the agent's world but originates from a different episode. ROCKET-2 attempts to infer the semantic intent underlying the goal specification (*building a bridge* and *approaching a house*).

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 shows agent's robustness to a variety of goal views.

5 Conclusion

We propose a cross-view goal specification method to improve human-agent interaction in embodied worlds and introduce cross-view consistency and visibility losses. Our agent sets a new benchmark on the *Minecraft Interaction Benchmark*, achieving 3×–6× higher efficiency. Our zero-shot generalization results underscore the promise of cross-view goal alignment as a core for building general-purpose decision-making agents in 3D worlds.

References

Baker, B.; Akkaya, I.; Zhokhov, P.; Huizinga, J.; Tang, J.; Ecoffet, A.; Houghton, B.; Sampedro, R.; and Clune, J. 2022. Video PreTraining (VPT): Learning to Act by Watching Unlabeled Online Videos. *ArXiv*, abs/2206.11795.

Beattie, C.; Leibo, J. Z.; Teplyashin, D.; Ward, T.; Wainwright, M.; Küttler, H.; Lefrancq, A.; Green, S.; Valdés, V.; Sadik, A.; Schrittwieser, J.; Anderson, K.; York, S.; Cant, M.; Cain, A.; Bolton, A.; Gaffney, S.; King, H.; Hassabis, D.; Legg, S.; and Petersen, S. 2016. DeepMind Lab. arXiv:1612.03801.

Borse, S.; Klingner, M.; Kumar, V. R.; Cai, H.; Almuzairee, A.; Yogamani, S.; and Porikli, F. 2023. 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*, 3287–3297.

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

Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Dabis, J.; Finn, C.; Gopalakrishnan, K.; Hausman, K.; Herzog, A.; Hsu, J.; Ibarz, J.; Ichter, B.; Irpan, A.; Jackson, T.; Jesmonth, S.; Joshi, N. J.; Julian, R. C.; Kalashnikov, D.; Kuang, Y.; Leal, I.; Lee, K.-H.; Levine, S.; Lu, Y.; Malla, U.; Manjunath, D.; Mordatch, I.; Nachum, O.; Parada, C.; Peralta, J.; Perez, E.; Pertsch, K.; Quiambao, J.; Rao, K.; Ryoo, M. S.; Salazar, G.; Sanketi, P. R.; Sayed, K.; Singh, J.; Sontakke, S. A.; Stone, A.; Tan, C.; Tran, H.; Vanhoucke, V.; Vega, S.; Vuong, Q. H.; Xia, F.; Xiao, T.; Xu, P.; Xu, S.; Yu, T.; and Zitkovich, B. 2022. RT-1: Robotics Transformer for Real-World Control at Scale. *ArXiv*, abs/2212.06817.

Cai, S.; Mu, Z.; He, K.; Zhang, B.; Zheng, X.; Liu, A.; and Liang, Y. 2024. MineStudio: A Streamlined Package for Minecraft AI Agent Development.

Cai, S.; Wang, Z.; Lian, K.; Mu, Z.; Ma, X.; Liu, A.; and Liang, Y. 2025a. ROCKET-1: Mastering Open-World Interaction with Visual-Temporal Context Prompting. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, 12122–12131.

Cai, S.; Zhang, B.; Wang, Z.; Lin, H.; Ma, X.; Liu, A.; and Liang, Y. 2025b. GROOT-2: Weakly Supervised Multimodal Instruction Following Agents. In *The Thirteenth International Conference on Learning Representations*.

Cai, S.; Zhang, B.; Wang, Z.; Ma, X.; Liu, A.; and Liang, Y. 2023. GROOT: Learning to Follow Instructions by Watching Gameplay Videos. In *The Twelfth International Conference on Learning Representations*.

Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging Properties in Self-Supervised Vision Transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q.; and Salakhutdinov, R. 2019. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Deitke, M.; Clark, C.; Lee, S.; Tripathi, R.; Yang, Y.; Park, J. S.; et al. 2024. Molmo and PixMo: Open Weights and Open Data for State-of-the-Art Multimodal Models. arXiv:2409.17146.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ArXiv*, abs/2010.11929.

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

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

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

Guss, W. H.; Houghton, B.; Topin, N.; Wang, P.; Codel, C.; Veloso, M. M.; and Salakhutdinov, R. 2019. MineRL: A Large-Scale Dataset of Minecraft Demonstrations. In *International Joint Conference on Artificial Intelligence*.

Huang, J.; Yong, S.; Ma, X.; Linghu, X.; Li, P.; Wang, Y.; Li, Q.; Zhu, S.-C.; Jia, B.; and Huang, S. 2023. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*.

Jang, E.; Irpan, A.; Khansari, M.; Kappler, D.; Ebert, F.; Lynch, C.; Levine, S.; and Finn, C. 2022. BC-Z: Zero-Shot Task Generalization with Robotic Imitation Learning. *ArXiv*, abs/2202.02005.

Jiang, J.; Yang, Y.; Deng, Y.; Ma, C.; and Zhang, J. 2024. BEVNav: Robot Autonomous Navigation Via Spatial-Temporal Contrastive Learning in Bird's-Eye View. *IEEE Robotics and Automation Letters*.

Kempka, M.; Wydmuch, M.; Runc, G.; Toczek, J.; and Jaśkowski, W. 2016. ViZDoom: A Doom-based AI research platform for visual reinforcement learning. In *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, 1–8. IEEE Press.

Krantz, J.; Gervet, T.; Yadav, K.; Wang, A.; Paxton, C.; Mottaghi, R.; Batra, D.; Malik, J.; Lee, S.; and Chaplot, D. S. 2023. Navigating to objects specified by images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10916–10925.

Krantz, J.; Lee, S.; Malik, J.; Batra, D.; and Chaplot, D. S. 2022. Instance-specific image goal navigation: Training embodied agents to find object instances. *arXiv preprint arXiv:2211.15876*.

Lifshitz, S.; Paster, K.; Chan, H.; Ba, J.; and McIlraith, S. A. 2023. STEVE-1: A Generative Model for Text-to-Behavior in Minecraft. *ArXiv*, abs/2306.00937.

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

Majumdar, A.; Aggarwal, G.; Devnani, B.; Hoffman, J.; and Batra, D. 2022. ZSON: Zero-Shot Object-Goal Navigation using Multimodal Goal Embeddings. *ArXiv*, abs/2206.12403.

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

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

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

Ravi, N.; Gabeur, V.; Hu, Y.-T.; Hu, R.; Ryali, C.; Ma, T.; Khedr, H.; Rädle, R.; Rolland, C.; Gustafson, L.; Mintun, E.; Pan, J.; Alwala, K. V.; Carion, N.; Wu, C.-Y.; Girshick, R.; Dollár, P.; and Feichtenhofer, C. 2024. SAM 2: Segment Anything in Images and Videos. *arXiv preprint arXiv:2408.00714*.

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

Sundaresan, P.; Vuong, Q.; Gu, J.; Xu, P.; Xiao, T.; Kirmani, S.; Yu, T.; Stark, M.; Jain, A.; Hausman, K.; et al. 2024. Rt-sketch: Goal-conditioned imitation learning from hand-drawn sketches. In *8th Annual Conference on Robot Learning*.

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

Wang, C.; Fan, L.; Sun, J.; Zhang, R.; Fei-Fei, L.; Xu, D.; Zhu, Y.; and Anandkumar, A. 2023a. MimicPlay: Long-Horizon Imitation Learning by Watching Human Play. In *Conference on Robot Learning*, 201–221. PMLR.

Wang, G.; Xie, Y.; Jiang, Y.; Mandlekar, A.; Xiao, C.; Zhu, Y.; Fan, L. J.; and Anandkumar, A. 2023b. Voyager: An Open-Ended Embodied Agent with Large Language Models. *ArXiv*, abs/2305.16291.

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

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

Xu, D.; Chen, J.; Liang, C.; Wang, Z.; and Hu, R. 2019. Cross-view Identical Part Area Alignment for Person Re-identification. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2462–2466.

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

Zhong, F.; Wu, K.; Wang, C.; Chen, H.; Ci, H.; Li, Z.; and Wang, Y. 2024. UnrealZoo: Enriching Photo-realistic Virtual Worlds for Embodied AI. arXiv:2412.20977.

Zhong, Y.; Huang, X.; Li, R.; Zhang, C.; Liang, Y.; Yang, Y.; and Chen, Y. 2025. DexGraspVLA: A Vision-Language-Action Framework Towards General Dexterous Grasping. arXiv:2502.20900.