# ROBOACT-CLIP: VIDEO-DRIVEN ATOMIC ACTION UNDERSTANDING FOR ROBOTICS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Vision–Language–Action (VLA) models have become a key framework for robotics, coupling multimodal perception with language-grounded decision making to enable cross-task generalization, dynamic interaction, and long-horizon planning. However, despite training on large-scale video and trajectory data, prevailing VLAs are predominantly imitation-driven and lack an intrinsic, spatiotemporal understanding of physical actions; as a result, their generalization degrades in unseen embodiments and contexts. In parallel, existing action-understanding approaches still fail to model temporally correlated action semantics and suffer from visual feature entanglement among the robot, manipulated objects, and background, hindering clean atomic-action semantics and reliable transfer.

We present **RoboAct-CLIP**, which addresses both issues with two components: (1) a *curated single-action training set* distilled from open-source robot videos via semantics-constrained action-unit segmentation and re-annotation, yielding purified clips each containing one atomic action (e.g., "grasp"); and (2) a *temporal-decoupling* architecture on a CLIP backbone. Concretely, a frozen CLIP visual encoder processes uniformly sampled frames; a *Temporal Diff-Transformer* operates on consecutive feature differences together with a start–end delta (the former emphasizes spatiotemporal dynamics, the latter summarizes action outcome); the fused representation is routed into *subject/object/action* branches with orthogonality constraints; and a *compositional contrastive objective* aligns branch-wise visual features with templated texts, with an additional *recombination alignment* loss between remixed branch features and their corresponding texts to *further strengthen disentanglement*. Used as a frozen backbone, RoboAct-CLIP supports lightweight policy heads and reduces per-task tuning.

In LIBERO and Franka Kitchen simulation, RoboAct-CLIP improves success rate by **12%** and **5.7%** over strong VLA baselines and exhibits better generalization in multi-object and unseen tasks; real-world evaluations on a *single* physical robot arm confirm stable atomic-action execution, with *RoboAct-CLIP kept frozen* and *only the downstream policy* adapted using task-specific data collected on the same platform. These results indicate that explicit temporal modeling plus factorized action/object/agent representations offers a simple, scalable path to more reliable VLA-based manipulation.

## 1 INTRODUCTION

Vision–Language–Action (VLA) models have enabled new paradigms in robotic perception, instruction following, and policy learning by aligning visual representations with natural-language semantics Li et al. (2025); Jeong et al. (2024); Ma et al. (2024). Representative systems such as RT-2 Brohan et al. (2023), RoboFlamingo Li et al. (2023), and SpatialVLM Chen et al. (2024) demonstrate strong cross-task generalization and zero-shot manipulation. However, most are trained on static image–text pairs or sparsely sampled frames, under-capturing the fine-grained temporal dynamics that distinguish sequential *atomic actions*, leading to error accumulation in long-horizon tasks and confusion between transient states (e.g., *lifting* vs. *tilting*).

Imitation learning paradigms incorporate temporal information by training on human or robot video demonstrations Zare et al. (2024); Lázaro-Gredilla et al. (2019); Miao et al. (2025). While this

provides sequential experience, policies often replicate low-level motion patterns and demo-specific idiosyncrasies rather than action intent, yielding brittle generalization. Overfitting to embodiment or environment further limits versatility.

A second bottleneck is *feature entanglement*: manipulation videos mix the robot, manipulated objects, and background, making it difficult to isolate the action-centric signal. Without explicit disentanglement, VLM representations are prone to contextual confusion and multimodal hallucination, undermining reliability and transfer Liu et al. (2024); Chakraborty et al. (2025).

We address these challenges with **RoboAct-CLIP**, a video-driven approach featuring two contributions. First, a *curated single-action training set* is distilled from open-source robot manipulation videos via semantics-constrained action-unit segmentation and re-annotation, producing purified clips each containing one atomic action (e.g., an isolated "grasp"). Second, a *temporal-decoupling* architecture on a CLIP backbone learns hierarchical action representations: a Temporal Diff-Transformer operates on consecutive feature differences together with a start–end delta (the former emphasizing spatiotemporal dynamics, the latter summarizing action outcome); the fused code is routed into subject/object/action branches with orthogonality constraints; and a compositional contrastive objective aligns branch-wise visual features with templated texts, with an additional recombination alignment term between remixed branch features and their corresponding texts to further strengthen disentanglement. In simulation, RoboAct-CLIP yields a 12% higher success rate than strong VLA baselines on long-horizon manipulation and generalizes to novel object configurations. Real-world evaluations on a *single* physical robot arm confirm stable atomic-action execution, with *RoboAct-CLIP kept frozen* and *only the downstream policy network* adapted using task-specific data collected on the same platform.

Our contributions can be summarized as follows:

- **Curated single-action training set.** We distill a single-action training set from open-source robot manipulation videos via semantics-constrained action-unit segmentation and re-annotation, producing purified clips each containing one atomic action (e.g., *grasp*) for action-centric supervision.
- **Temporal–decoupling CLIP.** A CLIP-based fine-tuning with a *Temporal Diff-Transformer* over consecutive feature differences and a start–end delta; the fused code is routed into orthogonal *subject/object/action* subspaces. A *compositional contrastive* objective, augmented with a *recombination alignment* term, strengthens disentanglement between action dynamics and object/agent appearance.
- **Policy-ready & downstream gains.** Used as a *frozen backbone*, RoboAct-CLIP supports lightweight policy heads (e.g., behavior cloning) and reduces per-task tuning. In simulation it improves success rate by **+12%** over strong VLA/VLM baselines and generalizes better to multi-/unseen-object settings. Real-world experiments on a *single* physical robot arm further validate the approach, with *RoboAct-CLIP kept frozen* and *only the downstream policy* adapted using task-specific data from the same platform.

## 2 Related Work

### 2.1 Vision–Language–Action Models

Vision–language–action (VLA) models jointly couple perception, language grounding, and action interfaces for robotic control. Early foundations such as CLIP Radford et al. (2021) and Flamingo Alayrac et al. (2022) have been adapted to robotics by co-training on web-scale image–text data and robot demonstrations, and then attaching policy layers. RT-2 Brohan et al. (2023) treats discrete robot actions as textual tokens and trains on internet and robot data to enable instruction following and object-conditioned skills. RoboFlamingo Li et al. (2023) fine-tunes OpenFlamingo Awadalla et al. (2023) on manipulation datasets and appends a lightweight control head for zero-shot manipulation. SpatialVLM Chen et al. (2024) augments the architecture with 3D positional cues to strengthen geometric reasoning in manipulation and navigation. Despite these advances, many VLAs are trained primarily on static images or sparsely sampled frames and tend to inherit demonstration-specific motion patterns, offering limited *temporal* resolution of atomic actions and often entangling embodiment and background context with task-relevant cues. Our work

is VLA-first: it introduces an explicitly temporal video pathway and a factorized representation that can be used as a *frozen* backbone for lightweight downstream policies.

### 2.2 Atomic Action Understanding

Fine-grained or *atomic* action understanding seeks to model the spatiotemporal micro-structure of manipulation. Video models that exploit motion, e.g., frame-difference networks and MSTDT Wang et al. (2024a) improve segmentation of long videos by encoding temporal change signals. In robotics, decomposition-based approaches like DART Wang et al. (2024b) translate language into sequences of atomic skills for stepwise execution, while outcome-aware embedding methods such as Robotic-CLIP compare start and end frames to capture action results Nguyen et al. (2025). However, boundary-only supervision may overlook intermediate transitions, and handcrafted primitive libraries can constrain flexibility. RoboAct-CLIP complements these ideas by (i) operating on *first-order feature differences* with a Temporal Diff-Transformer while also using a start–end delta, thereby capturing both transient dynamics and outcomes, and (ii) aligning action semantics at the *branch level* (subject/action/object) with a compositional contrastive objective. A recombination alignment term further encourages generalization to novel subject–action–object triplets without relying on manual primitive engineering.

### 2.3 Feature Disentanglement in Robot Learning

Disentangling agent, object, and scene factors is critical for transfer across embodiments and environments. Prior work has used attention/masking to isolate action-centric slots from context (e.g., DEVIAS) Bae et al. (2024), object-centered 3D interaction primitives with planner–executor loops to mitigate hallucinations (OmniManip) Pan et al. (2025), and adaptive prediction horizons with long-term memory to detect failures and solicit assistance (ACP) Misic (2024). These methods underscore the importance of separating what moves (agent), what is manipulated (object), and where/when it happens (context). Distinct from masking- or asset-heavy pipelines, RoboAct-CLIP performs *branch-wise factorization* directly within a CLIP-based VLA backbone: orthogonality constraints promote independence across subject/object/action subspaces; feature banks plus recombination alignment supply compositional supervision; and all components integrate with a contrastive training recipe that preserves language grounding. This yields action-focused, transferable embeddings without requiring explicit 3D assets or segmentation labels.

**Summary.** Existing VLAs excel at instruction grounding but under-capture fine-grained temporal dynamics and often entangle embodiment with task semantics. Work on atomic action understanding and disentanglement tackles parts of the problem—either dynamics at boundaries or factor separation via masking/3D priors—but rarely unifies them within a language-grounded VLA. RoboAct-CLIP brings these threads together via (i) temporal difference modeling that complements start–end outcomes and (ii) explicit subject/object/action factorization with compositional alignment, providing a simple, frozen backbone for downstream policy learning.

## 3 Methodology

Our framework starts with a curation and annotation pipeline over open-source robot video datasets, then integrates two components: a Temporal Difference Transformer for temporal reasoning and a Feature Disentanglement module for representation decoupling. We finally describe how the architecture is used for policy learning in sequential decision-making tasks.

### 3.1 Dataset Preparation

We select RH20T Fang et al. (2024), an open-source collection of $>110\,\mathrm{k}$ contact-rich manipulation sequences spanning actions, environments, robots, and viewpoints, with paired videos and language descriptions. Algorithm 1 sketches our preparation:

After processing, the dataset comprises: **199,797** videos · **143** tasks · **52** atomic actions · **63,922,209** frames.

**Algorithm 1** Dataset Preparation

**procedure** PREPARE(RH20T)
  Unzip videos and corresponding textual annotations.
  **for** each video $V$ with annotation $T$ **do**
    Query the DeepSeek R1 Guo et al. (2025) API with prompt: "Identify the number of actions, with verbs and objects, in: $T$."
    **if** response indicates multiple actions **then**
      Discard $V$ (retain only single-action clips).
    **else**
      Extract subject ($S$), action ($A$), object ($O$) from the response.
      Compose description: "Robot (or Human) [$A$] [$O$]. Action is $A$, Object is $O$."
    **end if**
  **end for**
**end procedure**

### 3.2 ROBOACT-CLIP

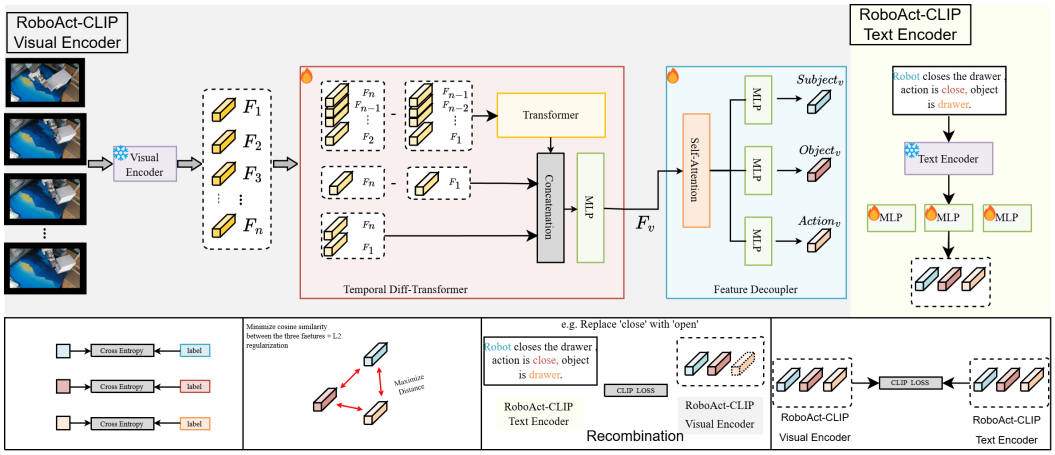


Figure 1: Overall framework of RoboAct-CLIP.

Our model extends CLIP with a temporal difference Transformer and feature-disentanglement modules to achieve fine-grained understanding of manipulation actions (fig. 1).

#### 3.2.1 CLIP ENCODERS (TEXT & VISUAL)

We use off-the-shelf CLIP text and visual encoders strictly as *frozen* feature extractors. Given a language instruction $I_{\text{text}}$ and a video sequence $[\text{Frame}_1, \ldots, \text{Frame}_n]$ (we use $n$=16 uniformly sampled frames), we compute:

$$F_{\text{t}} = \text{CLIP}_{\text{text}}(\text{tokenize}(I_{\text{text}})), \tag{1}$$

$$F_{\text{t\_sub}} = \text{MLP}_{\text{text\_subject}}(F_{\text{t}}), \tag{2}$$

$$F_{\text{t\_act}} = \text{MLP}_{\text{text\_action}}(F_{\text{t}}), \tag{3}$$

$$F_{\text{t\_obj}} = \text{MLP}_{\text{text\_object}}(F_{\text{t}}). \tag{4}$$

We then obtain per-frame visual features with the frozen visual encoder:

$$F_{v\_i} = \text{CLIP}_{\text{visual}}(\text{Frame}_i), \quad i{=}1, \ldots, n, \tag{5}$$

yielding $[F_{v\_1}, \ldots, F_{v\_n}]$ as inputs to the Temporal Diff-Transformer.

### 3.2.2 TEMPORAL DYNAMICS LEARNING

To explicitly model the spatiotemporal dynamics of actions, we propose the **Temporal Diff-Transformer**,

$$\Delta F_{v\_i} = F_{v\_i} - F_{v\_i-1}, \quad i = 2, \ldots, n, \tag{6}$$

which suppress static background and highlight motion. Differences are fed to a Transformer encoder (with positional encoding Vaswani et al. (2017)):

$$\{\mathrm{Tem}_{v\_i}\}_{i=2}^{n} = \mathrm{Transformer}(\{\Delta F_{v\_i}\}_{i=2}^{n}). \tag{7}$$

We take the last output as a summary, $\mathrm{Tem}_{\mathrm{visual}}{=}\mathrm{Tem}_{v\_n}$, and also compute the start–end change $\Delta F_v{=}F_{v\_n}{-}F_{v\_1}$. We then form the visual representation

$$F_{\mathrm{v}} = \mathrm{MLP}(\mathrm{Concat}[\mathrm{Tem}_{\mathrm{v}};\, \Delta F_v;\, F_{v\_1};\, F_{v\_n}]). \tag{8}$$

### 3.2.3 FEATURE DISENTANGLEMENT LEARNING

Self-attention augments context:

$$F_{\mathrm{v\_attn}} = \mathrm{MultiHeadAttention}(Q{=}F_{\mathrm{v}}, K{=}F_{\mathrm{v}}, V{=}F_{\mathrm{v}}). \tag{9}$$

We project to three branches:

$$F_{\mathrm{v\_sub}} = \mathrm{MLP}_{\mathrm{visual\_subject}}(F_{\mathrm{v\_attn}}), \quad F_{\mathrm{v\_obj}} = \mathrm{MLP}_{\mathrm{visual\_object}}(F_{\mathrm{v\_attn}}), \quad F_{\mathrm{v\_act}} = \mathrm{MLP}_{\mathrm{visual\_action}}(F_{\mathrm{v\_attn}}). \tag{10}$$

Orthogonality encourages independence:

$$\mathcal{L}_{\mathrm{sim}} = -\tfrac{1}{3N}\sum_{i=1}^{N}\Big(\mathrm{CosSim}(F^i_{\mathrm{v\_sub}}, F^i_{\mathrm{v\_act}}) + \mathrm{CosSim}(F^i_{\mathrm{v\_sub}}, F^i_{\mathrm{v\_obj}}) + \mathrm{CosSim}(F^i_{\mathrm{v\_act}}, F^i_{\mathrm{v\_obj}})\Big), \tag{11}$$

with small L2 regularization,

$$\mathcal{L}_{\mathrm{L2}} = 0.01\,(\|F_{\mathrm{v\_sub}}\|_2 + \|F_{\mathrm{v\_act}}\|_2 + \|F_{\mathrm{v\_obj}}\|_2). \tag{12}$$

**Feature banks and recombination.** We maintain visual feature banks $\mathcal{B}_{\mathrm{v\_sub}}, \mathcal{B}_{\mathrm{v\_act}}, \mathcal{B}_{\mathrm{v\_obj}}$ with one representative per class, updated every *Setting_Step*:

$$\mathcal{B}_{\mathrm{v\_sub}} = \{F^1_{\mathrm{v\_sub}},\ F^2_{\mathrm{v\_sub}},\ \ldots,\ F^{K_s}_{\mathrm{v\_sub}}\}\,, \tag{13}$$

$$\mathcal{B}_{\mathrm{v\_act}} = \{F^1_{\mathrm{v\_act}},\ F^2_{\mathrm{v\_act}},\ \ldots,\ F^{K_a}_{\mathrm{v\_act}}\}\,, \tag{14}$$

$$\mathcal{B}_{\mathrm{v\_obj}} = \{F^1_{\mathrm{v\_obj}},\ F^2_{\mathrm{v\_obj}},\ \ldots,\ F^{K_o}_{\mathrm{v\_obj}}\}\,, \tag{15}$$

We synthesize recombined visual features:

$$F^{s,a,o}_{\mathrm{recomb}} = \mathrm{Combiner}(\mathcal{B}_{\mathrm{v\_sub}}[sub\ s],\ \mathcal{B}_{\mathrm{\_act}}[act\ a],\ \mathcal{B}_{\mathrm{\_obj}}[obj\ o]) \tag{16}$$

with matching text embeddings:

$$T^{s,a,o}_{\mathrm{recomb}} = \mathrm{TextEnc}(\text{“}s\ a \text{ the } o\text{, action is } a\text{”}), \tag{17}$$

then apply a contrastive objective

$$\mathcal{L}_{\mathrm{recomb}} = -\tfrac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\mathrm{sim}(F^i_{\mathrm{recomb}}, T^i_{\mathrm{recomb}})/\tau)}{\sum_{b\in\mathcal{M}}\exp(\mathrm{sim}(F^i_{\mathrm{recomb}}, T^b_{\mathrm{recomb}})/\tau)}. \tag{18}$$

The enhanced disentanglement loss is

$$\mathcal{L}_{\mathrm{disent}} = \lambda_{\mathrm{ortho}}\,(\mathcal{L}_{\mathrm{sim}} + \mathcal{L}_{\mathrm{L2}}) + \lambda_{\mathrm{recomb}}\,\mathcal{L}_{\mathrm{recomb}}. \tag{19}$$

**Auxiliary classification.** To further guide the learning process, we incorporate auxiliary classification tasks for each branch of the disentanglement module Zhang et al. (2024); Wang et al. (2022). Specifically, we attach three classifiers (for subject, action, and object prediction) on $F_{\text{attn}}$:

$$\begin{aligned} P_{\text{v\_sub}} &= \text{Softmax}(\text{MLP}_{\text{classify-subject}}(F_{\text{v\_attn}})) \\ P_{\text{v\_act}} &= \text{Softmax}(\text{MLP}_{\text{classify-action}}(F_{\text{v\_attn}})) \\ P_{\text{v\_obj}} &= \text{Softmax}(\text{MLP}_{\text{classify-object}}(F_{\text{v\_attn}})) \end{aligned} \tag{20}$$

On $F_{\text{v\_attn}}$ we attach three heads with cross-entropy losses to predict subject/action/object classes (weights $\alpha_s, \alpha_a, \alpha_o$), forming

$$\mathcal{L}_{\text{aux}} = -\tfrac{1}{N}\sum_{i=1}^{N}\big(\alpha_s \,\text{CE}(P_{\text{sub}}, y_{\text{sub}}) + \alpha_a \,\text{CE}(P_{\text{act}}, y_{\text{act}}) + \alpha_o \,\text{CE}(P_{\text{obj}}, y_{\text{obj}})\big). \tag{21}$$

where CE denotes the cross-entropy loss, $\alpha_s, \alpha_a, \alpha_o$ are task-specific weights, and $y_{\text{sub}}, y_{\text{act}}, y_{\text{obj}}$ are the ground-truth labels. By training with these auxiliary tasks, each feature branch is encouraged to focus on its designated semantic aspect, thereby improving overall representation quality and downstream task performance.

### 3.2.4 Training Objective

To ensure cross-modal alignment between the visual and textual representations, we employ a CLIP-style contrastive loss. We first form the video-level feature $F_{\text{v}}^i$ by concatenating the three visual branch outputs, and likewise form the text-level feature $F_{\text{t}}^i$ by concatenating the text branch outputs:

$$F_{\text{v}}^i = \text{Concat}(F_{\text{v\_sub}}^i, F_{\text{v\_obj}}^i, F_{\text{v\_act}}^i), \quad F_{\text{t}}^i = \text{Concat}(F_{\text{t\_sub}}^i, F_{\text{t\_obj}}^i, F_{\text{t\_act}}^i), \tag{22}$$

$$\mathcal{L}_{\text{CLIP}} = -\tfrac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\text{sim}(F_{\text{v}}^i, F_{\text{t}}^i)/\tau)}{\sum_{j=1}^{N}\exp(\text{sim}(F_{\text{v}}^i, F_{\text{t}}^j)/\tau)}. \tag{23}$$

The total loss is

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{CLIP}} + \lambda_{\text{disent}}\,\mathcal{L}_{\text{disent}} + \lambda_{\text{aux}}\,\mathcal{L}_{\text{aux}}. \tag{24}$$

## 3.3 Application

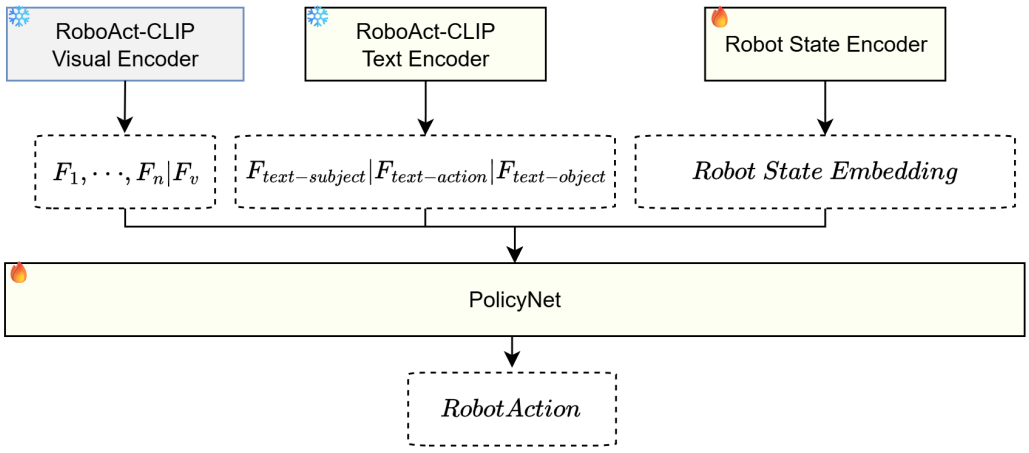


Figure 2: Applying RoboAct-CLIP in policy training.

During policy training, the pre-trained visual and textual encoders remain frozen. Robot state (joint configurations, gripper state) is concatenated with encoded features as input to the downstream policy.

Table 1: Success rates (%) in Libero dataset.

| Method | T1 | T2 | T3 | T4 | Overall |
|---|---|---|---|---|---|
| R3M | 46.0 | 80.0 | 58.0 | 52.0 | 59.0 |
| MPI (Small) | 50.0 | 74.0 | 38.0 | 66.0 | 57.0 |
| MPI (Base) | 62.0 | 70.0 | 44.0 | 82.0 | 64.5 |
| CLIP | 86.0 | 76.0 | 22.0 | 20.0 | 51.0 |
| Robotic-CLIP | 88.0 | 82.0 | 38.0 | 46.0 | 63.5 |
| **RoboAct-CLIP (Ours)** | **90.0** | **84.0** | **56.0** | **76.0** | **76.5** |
| - w/o Temporal Diff-Transformer | 78.0 | 72.0 | 70.0 | 44.0 | 66.0 |
| - w/o Feature Disentanglement | 88.0 | 76.0 | 56.0 | 60.0 | 70.0 |
| - w/o Reconstruction Contrastive Loss | 85.0 | 78.0 | 63.0 | 72.0 | 74.5 |
| - w/o Auxiliary Classification Loss | 83.0 | 73.0 | 54.0 | 70.0 | 70.0 |
| - Shorter sequence ($n$=2) | 74.0 | 69.0 | 50.0 | 62.0 | 63.8 |
| - Only original CLIP frame features | 88.0 | 79.0 | 28.0 | 32.0 | 56.7 |

# 4 EXPERIMENTS

We evaluate RoboAct-CLIP in simulated environments and on physical manipulators. RoboAct-CLIP is used as a frozen encoder for downstream policies. We perform ablations to isolate the contributions of the Temporal Diff-Transformer and Feature Disentanglement.

## 4.1 SIMULATION EXPERIMENTS

We use the LIBERO Liu et al. (2023) simulation benchmark for household manipulation. A robotic arm performs four tasks: (T1) open the middle drawer; (T2) push a plate to the stove front; (T3) place cream cheese in a bowl; (T4) turn on the stove. We report success rate: the fraction of episodes completing within 200 steps.

Baselines: R3M Nair et al. (2022); MPI (ViT-S/B) Zeng et al. (2024); CLIP Radford et al. (2021); Robotic-CLIP Nguyen et al. (2025). Our method: RoboAct-CLIP. All encoders are frozen within the same policy pipeline.

RoboAct-CLIP attains the highest average success rates on LIBERO, improving the strongest baseline by average (MPI-Base, 64.5%) to 76.5% (+12.0 pp) under a strictly frozen-encoder and identical policy setup. *Per-task trends.* Relative to MPI-Base, RoboAct-CLIP yields +28.0 pp on T1 (90.0 vs. 62.0), +14.0 pp on T2 (84.0 vs. 70.0), +12.0 pp on T3 (56.0 vs. 44.0), and $-6.0$ pp on T4 (76.0 vs. 82.0). Gains are largest on tasks with pronounced motion and state transitions (T1–T3), while T4 shows a smaller margin where precise goal completion dominates. Notably, against the best *per-task* baselines, RoboAct-CLIP remains competitive (e.g., +2.0 pp over Robotic-CLIP on T1 and T2) despite not leading on T3 (R3M: 58.0) and T4 (MPI-Base: 82.0).

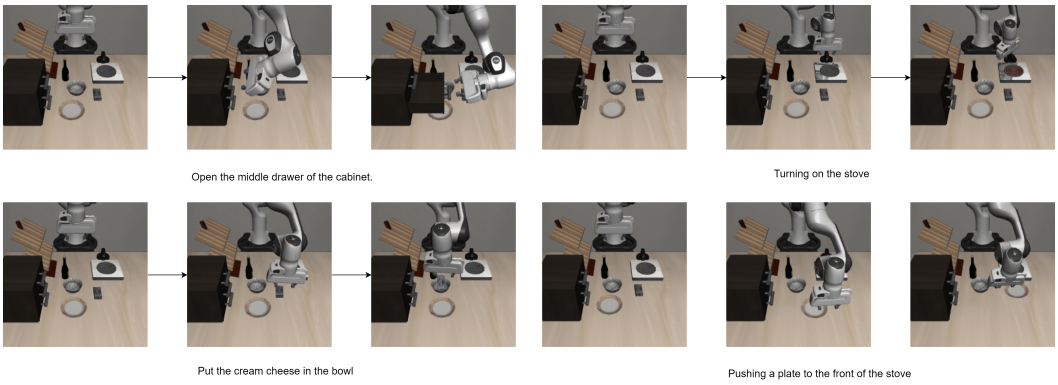


Figure 3: RoboAct-CLIP performing four manipulation tasks in LIBERO. Each row is a task; the model maintains precise control across sequences.

Table 2: Unseen generalization success rates(%) on LIBERO Simulation Environment.

| Method | T5 | T6 |
|---|---|---|
| R3M | 0 | 0 |
| MPI (Small) | 0 | 0 |
| MPI (Base) | 0 | 0 |
| CLIP | 0 | 0 |
| Robotic-CLIP | 0 | 0 |
| **RoboAct-CLIP (Ours)** | **4** | **16** |

Table 3: Success rates (%) in Franka Kitchen Simulation Environment.

| Method | Knob | Door | Switch | Micro | Slide | Overall |
|---|---|---|---|---|---|---|
| R3M | 53.6 | 49.8 | 85.9 | 58.7 | 98.2 | 69.2 |
| MPI (Small) | 84.1 | 49.5 | 88.8 | 58.8 | 99.4 | 76.1 |
| MPI (Base) | 88.1 | 57.1 | 94.0 | 54.1 | 99.4 | 78.5 |
| CLIP | 26.9 | 12.0 | 42.3 | 25.1 | 86.0 | 38.5 |
| Robotic-CLIP | 76.0 | 42.0 | 86.0 | 52.0 | 94.0 | 70.0 |
| **RoboAct-CLIP (Ours)** | **92.0** | **68.0** | **96.0** | **66.0** | **99.2** | **84.2** |

**Generalization to unseen tasks.** We further evaluate on two held-out tasks: (T5) put *alphabet soup* and *cream cheese* into the basket; (T6) pick the *black bowl* from the top drawer and place it on the plate. As reported in Table 2, all baselines achieve 0% on both tasks, whereas **RoboAct-CLIP** attains **4%** (T5) and **16%** (T6). Although the absolute rates are modest, they indicate better zero-shot generalization under the *same policy pipeline*—encoders frozen, no task-specific tuning on T5/T6, and only the downstream policy trained on in-distribution tasks.

We hypothesize two contributing factors. First, the *Temporal Diff-Transformer* plus start–end delta provides outcome-aware cues that help stitch primitives into coherent action chains (e.g., *open* $\rightarrow$ *pick* $\rightarrow$ *place*) required by T6. Second, factorizing *subject/object/action* features and optimizing with the *recombination alignment* term improves compositionality, enabling the policy to re-use atomic-action embeddings in new permutations. The higher success on T6 relative to T5 is consistent with this view: T6 primarily recombines primitives frequent in training (*open drawer*, *pick*, *place*), whereas T5 adds multi-object sequencing and state tracking (two distinct pickups and placements into a target receptacle), increasing long-horizon credit assignment difficulty.

**Franka Kitchen benchmark.** To test transfer beyond LIBERO, we evaluate on *Franka Kitchen* with five primitives (turn knob, open door, flip switch, open microwave, slide door) under the same protocol. Table 3 summarizes the results. Consistent with our motivation, models like Robotic-CLIP and RoboAct-CLIP that incorporate stronger temporal/action cues outperform static CLIP features on articulated-object manipulation (*open door*, *open microwave*).

### 4.2 Ablations

To further clarify the contribution of each component in RoboAct-CLIP, we performed the following ablation experiments:

1. **A1 – Without Temporal Diff-Transformer.** We fed only the first and last frame features $\big(F_1, F_n\big)$ to remove temporal modeling and assess its impact (cf. Eq. 8).
2. **A2 – Without Feature Disentanglement.** The disentanglement block was removed; the visual encoder output was directly aligned with text, isolating its effect.
3. **A3 – Without Reconstruction Contrastive Loss.** We disabled the contrastive term used to align reconstructed features, retaining all other losses, to test its influence on cross-modal representation quality.
4. **A4 – Without Auxiliary Classification Loss.** The three-way action/agent/object classification loss $\mathcal{L}_{\text{cls}}$ was dropped, leaving only contrastive and reconstruction losses, to quantify the value of categorical supervision for disentanglement.
5. **A5 – Shorter Video Sequence ($n$=2 frames).** Like Robotic-CLIP, we reduced each clip from 16 to 2 frames to gauge how sequence length affects performance and temporal reasoning.
6. **A6 – Only original CLIP frame features.** Keeping the full RoboAct-CLIP architecture intact, we supplied only the frame-level features $F_1, \ldots, F_n$ to the downstream policy, measuring how much the temporal and disentanglement modules contribute beyond raw frame features.

The ablation results in Table1 clearly demonstrate the importance of both proposed components. Removing the Temporal Diff-Transformer (Ablation 1) led to a significant performance drop of

Table 4: Real-world manipulation success rates (%).

| Method | Subtask success | | | | Avg. Success |
|---|---|---|---|---|---|
| | Open (middle) | Pick (tape) | Place (on table) | Close (drawer) | |
| MPI | 50.0 | 46.7 | 42.9 | 33.3 | 43.2 |
| CLIP | 43.3 | 30.8 | 50.0 | 0.0 | 31.3 |
| Robotic-CLIP | 53.3 | 37.5 | 50.0 | 33.3 | 43.5 |
| **RoboAct-CLIP (Ours)** | **66.7** | **60.0** | **58.3** | **71.4** | **64.6** |

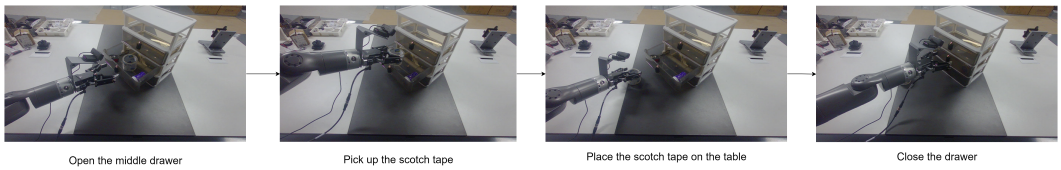


Figure 4: Execution sequence of the real-world task using RoboAct-CLIP.

10.5 percentage points in overall success rate, with particularly pronounced effects on Tasks 2 and 4. This confirms the critical role of temporal modeling in capturing action dynamics. Similarly, the absence of the Feature Disentanglement module (Ablation 2) resulted in a 6.5 percentage point decrease in overall performance, highlighting its effectiveness in separating task-relevant features from embodiment-specific information. These findings validate our architectural design choices and underscore the complementary nature of the proposed components.

### 4.3 Real-World Robot Experiments

We train a policy on teleoperated trajectories to execute: (1) open middle drawer, (2) pick up scotch tape, (3) place it on the table, (4) close the drawer. RoboAct-CLIP transfers from simulation to real settings, with temporal modeling providing robustness to lighting, appearance, and dynamics variations.

**Analysis.** Table 4 summarizes per-subtask and average success rates. RoboAct-CLIP attains an average of **64.6%**, outperforming Robotic-CLIP (43.5%), MPI (43.2%), and CLIP (31.3%) by **+21.1**, **+21.4**, and **+33.3** percentage points, respectively. Per-task gains over the best baseline are: *Open* $+13.4$ pts (66.7 vs. 53.3), *Pick* $+13.3$ pts (60.0 vs. 46.7), *Place* $+8.3$ pts (58.3 vs. 50.0), and *Close* $+38.1$ pts (71.4 vs. 33.3). The largest improvement appears on the terminal *Close* step, where success depends on accumulating temporal evidence and the net outcome of actuation; our Temporal Diff-Transformer, together with the start–end delta, explicitly encodes such outcome-aware dynamics. Meanwhile, factorizing subject/object/action features reduces embodiment- and appearance-induced interference, which benefits dexterous acquisition and placement (e.g., *Pick* $+13.3$ pts). All methods share the same policy pipeline; encoders are frozen and only the state encoder and policy head are trained on teleoperated data collected on the same platform.

## 5 Conclusions

We introduced **RoboAct-CLIP**, a VLA-oriented approach for atomic action understanding in robotics. The method combines two ingredients: (i) a *curated single-action training set* distilled from open-source manipulation videos via semantics-constrained action-unit segmentation and re-annotation; and (ii) a *temporal–decoupling* fine-tuning atop a frozen CLIP backbone, where a Temporal Diff-Transformer operates on frame-difference features together with a start–end delta, and the fused code is routed into orthogonality-constrained *subject/object/action* branches. A compositional contrastive objective, augmented with a recombination alignment term, aligns branch-wise visual features to templated texts and *further strengthens disentanglement* between action dynamics and object/agent appearance.

Used as a frozen backbone, RoboAct-CLIP supports lightweight policy heads and reduces per-task tuning. In LIBERO and Franka Kitchen simulation, it improves success rate by **12%** and **5.7%**

over strong VLA baselines and generalizes better to multi-/unseen-object settings. On hardware, real-world experiments on a *single* robot arm confirm stable atomic-action execution with *RoboAct-CLIP kept frozen* and *only the downstream policy/state encoder* adapted using platform-specific data. Ablations verify the complementary roles of the Temporal Diff-Transformer, the factorized branches, and the recombination alignment loss.

Overall, explicitly modeling temporal change while factorizing action, object, and agent cues provides a simple and scalable recipe for more reliable VLA-based manipulation.

## REPRODUCIBILITY STATEMENT

We aim to make our results reasonably verifiable under the double-blind review setting.

**Code (minimal release).** An anonymized repository is available at `https://anonymous.4open.science/r/RoboAct-CLIP-187C/`. It provides the core training and evaluation scripts and configuration templates for RoboAct-CLIP, along with a README describing basic setup and how to launch representative experiments.

**Datasets.** All benchmarks used in the paper are public. For our curated single-action clips derived from open-source videos, we share curation instructions and metadata/manifests that allow reviewers to reproduce the curation without redistributing third-party media.

**Evaluation.** Success metrics and task horizons follow the definitions stated in the paper; runs use fixed random seeds as specified in the provided configs. The released scripts compute the reported aggregate metrics from per-episode logs.

**Environment.** The repository includes basic dependency specifications sufficient to execute the main experiments on a single modern GPU.

**Accountability.** Any limitations or nondeterminism (e.g., hardware variance in real-robot trials) are documented in the README. All claims in the paper are supported by the artifacts described above.

## REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023.

Kyungho Bae, Geo Ahn, Youngrae Kim, and Jinwoo Choi. Devias: Learning disentangled video representations of action and scene. In *European Conference on Computer Vision*, pp. 431–448. Springer, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Neeloy Chakraborty, Melkior Ornik, and Katherine Driggs-Campbell. Hallucination detection in foundation models for decision-making: A flexible definition and review of the state of the art. *ACM Computing Surveys*, 2025.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities, 2024. URL `https://arxiv.org/abs/2401.12168`.

Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. In

*2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 653–660. IEEE, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Hyeongyo Jeong, Haechan Lee, Changwon Kim, and Sungtae Shin. A survey of robot intelligence with large language models. *Applied Sciences*, 14(19):8868, 2024.

Miguel Lázaro-Gredilla, Dianhuan Lin, J Swaroop Guntupalli, and Dileep George. Beyond imitation: Zero-shot task transfer on robots by learning concepts as cognitive programs. *Science Robotics*, 4(26):eaav3150, 2019.

Peihan Li, Zijian An, Shams Abrar, and Lifeng Zhou. Large language models for multi-robot systems: A survey, 2025. URL https://arxiv.org/abs/2502.03814.

Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, Hang Li, and Tao Kong. Vision-language foundation models as effective robot imitators. *arXiv preprint arXiv:2311.01378*, 2023.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Hanchao Liu, Wenyuan Xue, Yifei Chen, Dapeng Chen, Xiutian Zhao, Ke Wang, Liping Hou, Rongjun Li, and Wei Peng. A survey on hallucination in large vision-language models. *arXiv preprint arXiv:2402.00253*, 2024.

Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. A survey on vision-language-action models for embodied ai. *arXiv preprint arXiv:2405.14093*, 2024.

Zhuochen Miao, Jun Lv, Hongjie Fang, Yang Jin, and Cewu Lu. Knowledge-driven imitation learning: Enabling generalization across diverse conditions. *arXiv preprint arXiv:2506.21057*, 2025.

Aurora Misic. Robots that adaptively learn when to ask for help: Hallucination reduction in robotic task planning using large language models. Master's thesis, NTNU, 2024.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

Nghia Nguyen, Minh N Vu, Tung D Ta, Baoru Huang, Thieu Vo, Ngan Le, and Anh Nguyen. Robotic-clip: Fine-tuning clip on action data for robotic applications. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5930–5936. IEEE, 2025.

Mingjie Pan, Jiyao Zhang, Tianshu Wu, Yinghao Zhao, Wenlong Gao, and Hao Dong. Omnimanip: Towards general robotic manipulation via object-centric interaction primitives as spatial constraints. *arXiv preprint arXiv:2501.03841*, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Ni Wang, Dongliang Liao, and Xing Xu. Multi-scale temporal difference transformer for video-text retrieval. *arXiv preprint arXiv:2406.16111*, 2024a.

Yongdong Wang, Runze Xiao, Jun Younes Louhi Kasahara, Ryosuke Yajima, Keiji Nagatani, Atsushi Yamashita, and Hajime Asama. Dart-llm: Dependency-aware multi-robot task decomposition and execution using large language models. *arXiv preprint arXiv:2411.09022*, 2024b.

Ze Wang, Guogang Liao, Xiaowen Shi, Xiaoxu Wu, Chuheng Zhang, Yongkang Wang, Xingxing Wang, and Dong Wang. Learning list-wise representation in reinforcement learning for ads allocation with multiple auxiliary tasks. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, pp. 3555–3564, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392365. doi: 10.1145/3511808.3557094. URL https://doi.org/10.1145/3511808.3557094.

Maryam Zare, Parham M Kebria, Abbas Khosravi, and Saeid Nahavandi. A survey of imitation learning: Algorithms, recent developments, and challenges. *IEEE Transactions on Cybernetics*, 2024.

Jia Zeng, Qingwen Bu, Bangjun Wang, Wenke Xia, Li Chen, Hao Dong, Haoming Song, Dong Wang, Di Hu, Ping Luo, et al. Learning manipulation by predicting interaction. *arXiv preprint arXiv:2406.00439*, 2024.

Zhiyuan Zhang, Qichao Zhang, Xiaoxu Wu, Xiaowen Shi, Guogang Liao, Yongkang Wang, Xingxing Wang, and Dongbin Zhao. User response modeling in reinforcement learning for ads allocation. In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, pp. 131–140, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400701726. doi: 10.1145/3589335.3648310. URL https://doi.org/10.1145/3589335.3648310.

## A LLM USAGE STATEMENT

We used large language models (LLMs) in the following ways and take full responsibility for all content produced.

**Writing assistance.** We used ¡GPT-5¿ for wording/grammar suggestions on drafts of the abstract, introduction, and related work. All technical claims, references, and equations were authored and verified by the authors.

**Research assistance.** We used ¡Claude Sonnet 4¿ to brainstorm ablation variants and to generate boilerplate code snippets (e.g., logging/argument parsing). All experimental code and results were implemented, verified, and validated by the authors.

**Data and annotation.** During the single-action data curation, we used LLM prompts solely to assist with semantics-constrained filtering and slot extraction. The exact prompts are described in the paper (see Algorithm 1). Final labels and segment boundaries were checked by the authors.

No LLM was listed as an author. We verified all LLM outputs for factual accuracy and correctness, and we remain fully accountable for the paper's content.