RoboAct-CLIP: Video-Driven Atomic Action Understanding for Robotic Manipulation

Anonymous Authors

***Abstract*—Vision–Language–Action (VLA) models enable cross-task generalization in robotics, but many existing methods remain imitation-driven and lack intrinsic *action* understanding. Although training on human or robot demonstrations provides sequential experience, such models often mimic low-level motion patterns rather than learning the essential semantics of actions. Besides the inherent imitation behavior of VLAs, learning the physical concept of an action is challenging due to the feature entanglement: the acting subject, manipulated object, and background context are often tightly coupled in manipulation videos. As a result, the learned representations may remain tied to particular scenes, subjects, or manipulated objects, instead of capturing action concepts that are invariant across embodiments and environments, limiting the learning generalization. To address these challenges, we introduce RoboAct-CLIP, a video-driven backbone for atomic action understanding. Our key idea is to learn subject-invariant and action-focused representations from heterogeneous manipulation videos. To this end, we construct a heterogeneous single-action dataset in which different subjects (robot or human) perform the same atomic action, encouraging the model to focus on the essence of the action rather than subject-specific appearance. We further develop a spatiotemporal disentanglement architecture that separates subject, action, and object semantics while enforcing consistent compositional alignment across modalities. RoboAct-CLIP achieves 99.49% action recognition accuracy, demonstrating the compactness of the learned action representation and effectiveness of feature disentanglement. When integrated into a strong VLA baseline as a frozen visual backbone with lightweight policy heads, RoboAct-CLIP improves success rates by 12.0 and 5.7 percentage points on LIBERO and Franka Kitchen, respectively. Furthermore, RoboAct-CLIP enables real-world long-horizon task completion by recomposing atomic actions.**

***Index Terms*—Vision–Language–Action, Atomic Action Understanding, Representation Learning, Robotic Manipulation**

I. 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 [1]–[3]. Representative systems such as RT-2 [4], RoboFlamingo [5], and SpatialVLM [6] 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 *spatiotemporal* dynamics that distinguish sequential atomic actions, which can lead to error accumulation in long-horizon tasks and confusion between subtle motions and states (e.g., *lifting* vs. *tilting*).

Imitation-based VLAs incorporate temporal information by training on human or robot video demonstrations [7]–[9]. While this provides sequential experience, such policies often replicate low-level motion patterns and demonstration-specific idiosyncrasies rather than learning the essential semantics of an action itself. As a result, the learned representation may remain tied to particular scenes, subjects, or manipulated objects, instead of capturing an action concept that is invariant across embodiments and environments. This limits generalization when the same atomic action must be recognized or reused under different visual appearances, object instances, or executors.

However, enabling robots to understand the physical semantics of an action is challenging due to feature entanglement: manipulation videos often mix the acting subject, manipulated object, and background context, making it difficult to isolate action-relevant cues. Without explicit disentanglement, it is difficult for the model to achieve scene-independent action understanding because it over-relies on contextual appearance rather than the action itself. For example, a sequence of *lifting* can be misinterpreted as *tilting* when the object pose changes together with arm motion, or *opening* can be confused with *pulling* when the drawer, gripper, and surrounding scene are tightly coupled in the visual signal. Such confusion reduces the reliability of downstream policy transfer.

To address these challenges, we introduce RoboAct-CLIP, a video-driven approach for atomic action understanding. Our key insight is that representations of the same atomic action should be consistent across executors, manipulated objects, scenes, and modalities. In this regard, we construct a heterogeneous single-action training set from open-source manipulation videos through semantics-constrained filtering and re-annotation, in which different subjects (robot or human) perform the same atomic action. This design encourages the model to focus on action semantics rather than subject-specific appearance, thereby facilitating cross-subject, cross-object and cross-modality action alignment. Based on the dataset, we propose the **RoboAct-CLIP** architecture, which adopts a spatiotemporal disentanglement design to separate subject, action, and object factors from manipulation videos, and further performs feature recombination with cross-modal consistency constraints to learn more precise and subject-invariant atomic action representations. Offline evaluation shows that RoboAct-CLIP achieves **99.49%** action classification accuracy, indicating precise atomic action understanding. When integrated into a strong VLA baseline as a frozen action representation provider, RoboAct-CLIP improves the success rate by **12.0** percentage points on the LIBERO [10] simulation benchmark, generalizes better to unseen object configurations, and improves performance by **5.7** percentage points on Franka Kitchen. Furthermore, by decomposing complex manipulation tasks into atomic actions and recomposing them during execution, RoboAct-CLIP transfers to a real robot arm and enables

long-horizon task completion with only the downstream policy adapted.

Our contributions can be summarized as follows:

- **Heterogeneous single-action dataset for cross-subject action alignment.** We construct a heterogeneous single-action dataset from open-source manipulation videos via semantics-constrained filtering and re-annotation, in which different subjects (robot or human) perform the same atomic action. This design helps the model focus on the essence of the action rather than subject-specific appearance, and facilitates cross-subject alignment of atomic action semantics.
- **RoboAct-CLIP for inherent atomic action understanding.** We propose the RoboAct-CLIP architecture, which adopts a spatiotemporal disentanglement design to separate subject, action, and object factors from videos, and further performs feature recombination with cross-modal consistency constraints. This enables more inherent understanding of atomic actions and strengthens subject-invariant action representations.
- **Plug-in gains for downstream policy learning and real-world long-horizon execution.** RoboAct-CLIP can be integrated into a strong VLA baseline as a frozen visual backbone with lightweight policy heads. It achieves **99.49%** action classification accuracy offline, improves success rates by **12.0** and **5.7** percentage points on LIBERO and Franka Kitchen, respectively, and supports long-horizon real-world task execution by decomposing complex tasks into atomic actions and recomposing them during execution.

II. Related Work

A. 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 [11] and Flamingo [12] have been adapted to robotics by co-training on web-scale image–text data and robot demonstrations, and then attaching policy layers. RT-2 [4] treats discrete robot actions as textual tokens and trains on internet and robot data to enable instruction following and object-conditioned skills. RoboFlamingo [5] fine-tunes OpenFlamingo [13] on manipulation datasets and appends a lightweight control head for zero-shot manipulation. SpatialVLM [6] 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, providing limited *spatiotemporal* understanding of atomic actions and often entangling embodiment and background context with task-relevant cues. Our work addresses these limitations with a video-driven and factorized representation that can serve as a *frozen* backbone for downstream policies.

B. 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 [14], improve long-video understanding by encoding temporal change signals. In robotics, decomposition-based approaches like DART [15] 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 [16]. However, boundary-only supervision may overlook intermediate transitions, and handcrafted primitive libraries can constrain flexibility. In contrast, RoboAct-CLIP focuses on more precise atomic action understanding through spatiotemporal modeling together with subject–action–object factorization.

C. Feature Disentanglement in Robot Learning

Disentangling agent, object, and scene factors is critical for transfer across embodiments and environments. Prior work has used attention or masking to isolate action-centric slots from context (e.g., DEVIAS) [17], object-centered interaction primitives with planner–executor loops (OmniManip) [18], and adaptive prediction horizons with long-term memory to detect failures and solicit assistance (ACP) [19]. These methods highlight the importance of separating what moves (agent), what is manipulated (object), and where the interaction occurs (context). Different from masking- or asset-heavy pipelines, RoboAct-CLIP learns disentangled subject, action, and object representations directly within a CLIP-based VLA framework, enabling action-focused and transferable embeddings without requiring explicit 3D assets or segmentation labels.

Summary. Existing VLAs often under-model the fine-grained *spatiotemporal* structure of atomic actions and learn representations that remain sensitive to subject, object, and scene variations. Prior work on atomic action understanding and disentanglement addresses parts of this problem, but rarely unifies cross-subject action alignment, precise action modeling, and language grounding within a single VLA framework. RoboAct-CLIP fills this gap with a heterogeneous single-action dataset and a spatiotemporal disentanglement design, yielding a frozen visual backbone for more reliable downstream policy learning and long-horizon task execution.

III. Methodology

To facilitate action representation learning, our framework begins with the construction of a heterogeneous single-action dataset from open-source robot manipulation videos for subject-invariant atomic action learning across diverse subjects and visual conditions. We then introduce the RoboAct-CLIP architecture for precise atomic action understanding. The architecture consists of two parts: a spatiotemporal modeling module for capturing action dynamics and a feature disentanglement module for separating subject, action, and object semantics. Finally, we describe how the learned representation is used as a frozen visual backbone for downstream policy learning.

A. Dataset Preparation

To support spatiotemporal atomic action learning, we construct a heterogeneous single-action dataset from open-source

robot manipulation videos. Raw manipulation datasets often contain multi-step behaviors, diverse visual conditions, and strong coupling among the acting subject, manipulated object, and surrounding scene. If used directly, the model may overfit to subject-specific appearance or task context rather than the action itself.

We use two open-source robot manipulation datasets, RH20T [20] and BridgeData V2 [21], as the basis for dataset construction. We then perform semantics-constrained filtering and re-annotation to retain only clips corresponding to a single atomic action and convert them into a unified subject–action–object description. This process reduces semantic ambiguity and facilitates cross-subject alignment of atomic action semantics. Algorithm 1 sketches the preparation procedure. To improve annotation reliability, we further conduct manual spot checks on the filtered results.

Algorithm 1 Dataset Preparation

procedure PREPARE($\mathcal{D}$)
  Unzip videos and corresponding textual annotations.
  for each video V with annotation T **do**
    Query the DeepSeek R1 [22] 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

As shown in Table I, after processing, the dataset consists of 199,797 videos categorized into 143 unique tasks. These videos encompass 52 different atomic actions and contain a total of 63,922,209 frames.

TABLE I: Summary of the Processed Dataset

Item	Count
Total Videos	199,797
Unique Tasks	143
Distinct Atomic Actions	52
Total Frames	63,922,209

B. 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).

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} 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 Spatiotemporal Diff-Transformer.

2) Spatiotemporal Dynamics Learning: To explicitly model the spatiotemporal dynamics of actions, we propose the **Spatiotemporal 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 [23]):

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

Finally, we concatenate the Transformer output,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

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

3) Feature Disentanglement Learning: Self-attention augments context:

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

We project to three branches:

$$F_{\text{v\_sub}} = \text{MLP}_{\text{visual\_subject}}(F_{\text{v\_attn}}), \tag{10}$$

$$F_{\text{v\_obj}} = \text{MLP}_{\text{visual\_object}}(F_{\text{v\_attn}}), \tag{11}$$

$$F_{\text{v\_act}} = \text{MLP}_{\text{visual\_action}}(F_{\text{v\_attn}}). \tag{12}$$

Orthogonality encourages independence:

$$\begin{aligned}\mathcal{L}_{\text{sim}} = \frac{1}{3N}\sum_{i=1}^{N}\big(\ &\text{CosSim}(F^i_{\text{v\_sub}}, F^i_{\text{v\_act}}) \\ &+ \text{CosSim}(F^i_{\text{v\_sub}}, F^i_{\text{v\_obj}}) \\ &+ \text{CosSim}(F^i_{\text{v\_act}}, F^i_{\text{v\_obj}})\big).\end{aligned} \tag{13}$$

with small L2 regularization,

$$\mathcal{L}_{\text{L2}} = 0.01 * (\|F_{\text{v\_sub}}\|_2^2 + \|F_{\text{v\_act}}\|_2^2 + \|F_{\text{v\_obj}}\|_2^2). \tag{14}$$

a) Feature banks and recombination.: To further improve the disentanglement module, we introduce feature banks $\mathcal{B}_{\text{v\_sub}}, \mathcal{B}_{\text{v\_act}}, \mathcal{B}_{\text{v\_obj}}$ that store one representative feature vector per class for the subject, action, and object branches, respectively. These feature banks are updated at fixed intervals during training:

$$\mathcal{B}_{\text{v\_sub}} = \{F^1_{\text{v\_sub}},\ F^2_{\text{v\_sub}},\ \ldots,\ F^{K_s}_{\text{v\_sub}}\}\ , \tag{15}$$

$$\mathcal{B}_{\text{v\_act}} = \{F^1_{\text{v\_act}},\ F^2_{\text{v\_act}},\ \ldots,\ F^{K_a}_{\text{v\_act}}\}\ , \tag{16}$$

$$\mathcal{B}_{\text{v\_obj}} = \{F^1_{\text{v\_obj}},\ F^2_{\text{v\_obj}},\ \ldots,\ F^{K_o}_{\text{v\_obj}}\}\ , \tag{17}$$

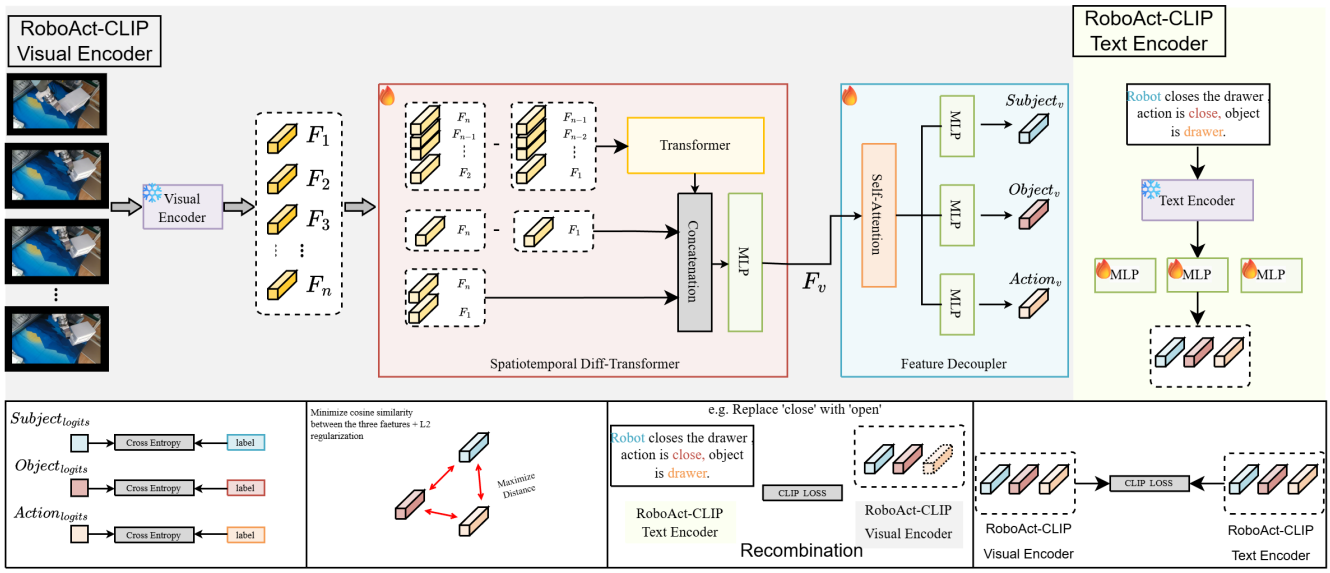

Fig. 1: Overall framework of RoboAct-CLIP.

TABLE II: Success rates (%) in the LIBERO simulation environment using different encoders with the same downstream policy.

Encoder + Same Policy	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 Spatiotemporal Diff-Transformer	78.0	72.0	47.0	53.0	62.5
- w/o Feature Disentanglement	88.0	76.0	56.0	60.0	70.0
- w/o Recombination Contrastive Loss	85.0	78.0	53.0	67.0	70.8
- w/o Auxiliary Classification Loss	84.0	80.0	54.0	74.0	73.0
- 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

TABLE III: Unseen generalization success rates (%) in the LIBERO simulation environment using different encoders with the same downstream policy.

Encoder	T5	T6
R3M	0.0	0.0
MPI (Small)	0.0	0.0
MPI (Base)	0.0	0.0
CLIP	0.0	0.0
Robotic-CLIP	0.0	0.0
RoboAct-CLIP (Ours)	**4.0**	**16.0**

where K_s, K_a, and K_o represent the number of unique subjects, actions, and objects in our dataset.

During the recombination phase, we leverage these stored features to compute an additional CLIP loss between recombined visual features and corresponding text instructions. For instance, the stored features for "robot" (subject), "open" (action), and "drawer" (object) can be recombined to create a synthetic visual representation that is then compared against the text encoding of "Robot opens the drawer, action is open." This approach allows us to evaluate whether the disentangled features can be effectively recombined to match novel combinations of subjects, actions, and objects described in text instructions:

$$F^{s,a,o}_{\text{recomb}} = \text{MLP}(\mathcal{B}_{\text{v\_sub}}[sub\ s], \mathcal{B}_{\text{v\_act}}[act\ a], \mathcal{B}_{\text{v\_obj}}[obj\ o]) \tag{18}$$

where $\mathcal{M}$ is a set of valid (subject, action, object) triplets sampled from the feature banks, and for each sample $a \in \mathcal{M}$ in the batch of size N, we randomly select one triplet from $\mathcal{M}$.

With matching text embeddings:

$$T^{s,a,o}_{\text{recomb}} = \text{TextEnc}(\text{“}s\ a \text{ the } o\text{, action is } a\text{”}), \tag{19}$$

then apply a contrastive objective:

$$\mathcal{L}_{\text{recomb}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(\text{sim}(F^i_{\text{recomb}}, T^i_{\text{recomb}})/\tau)}{\sum_{b\in\mathcal{M}} \exp(\text{sim}(F^i_{\text{recomb}}, T^b_{\text{recomb}})/\tau)}. \tag{20}$$

The enhanced disentanglement loss is:

$$\mathcal{L}_{\text{disent}} = \lambda_{\text{ortho}}\,(\mathcal{L}_{\text{sim}} + \mathcal{L}_{\text{L2}}) + \lambda_{\text{recomb}}\,\mathcal{L}_{\text{recomb}}. \tag{21}$$

b) Auxiliary classification.: To further guide the learning process, we incorporate auxiliary classification tasks for each branch of the disentanglement module [24], [25]. Specifically, we attach three classifiers (for subject, action, and object prediction) on $F_{\text{v\_attn}}$:

$$\begin{aligned} P_{\text{v\_sub}} &= \text{Softmax}(\text{MLP}_{\text{classify-subject}}(F_{\text{v\_attn}})) \\ P_{\text{v\_act}} &= \text{Softmax}(\text{MLP}_{\text{classify-action}}(F_{\text{v\_attn}})) \\ P_{\text{v\_obj}} &= \text{Softmax}(\text{MLP}_{\text{classify-object}}(F_{\text{v\_attn}})) \end{aligned} \tag{22}$$

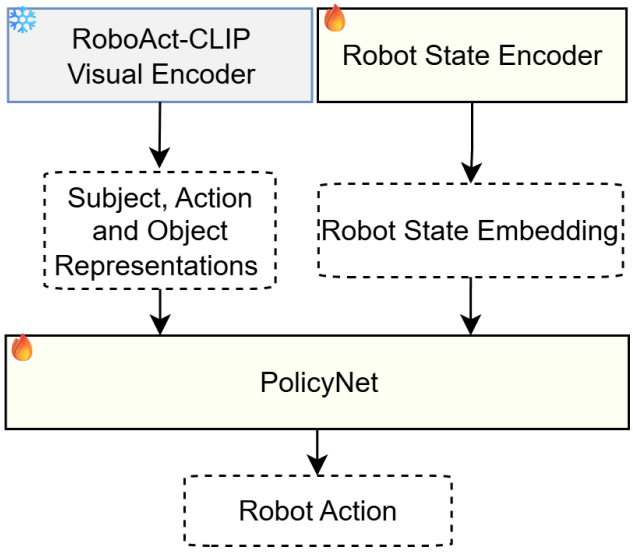

Fig. 2: Applying RoboAct-CLIP in policy training.

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:

$$\mathcal{L}_{\text{aux}} = -\frac{1}{N}\sum_{i=1}^{N}\big(\alpha_s \,\text{CE}(P^i_{\text{v\_sub}}, y^i_{\text{sub}}) + \alpha_a \,\text{CE}(P^i_{\text{v\_act}}, y^i_{\text{act}}) + \alpha_o \,\text{CE}(P^i_{\text{v\_obj}}, y^i_{\text{obj}})\big). \quad (23)$$

where CE denotes the cross-entropy loss, α_s, α_a, α_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.

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^i_{v} by concatenating the three visual branch outputs, and likewise form the text-level feature F^i_{t} by concatenating the text branch outputs:

$$F^i_{\text{v}} = \text{Concat}(F^i_{\text{v\_sub}}, F^i_{\text{v\_obj}}, F^i_{\text{v\_act}}), \quad (24)$$

$$F^i_{\text{t}} = \text{Concat}(F^i_{\text{t\_sub}}, F^i_{\text{t\_obj}}, F^i_{\text{t\_act}}), \quad (25)$$

$$\mathcal{L}_{\text{CLIP}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\text{sim}(F^i_{\text{v}}, F^i_{\text{t}})/\tau)}{\sum_{j=1}^{N}\exp(\text{sim}(F^i_{\text{v}}, F^j_{\text{t}})/\tau)}. \quad (26)$$

The total loss is

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{CLIP}} + \lambda_{\text{disent}}\,\mathcal{L}_{\text{disent}} + \lambda_{\text{aux}}\,\mathcal{L}_{\text{aux}}. \quad (27)$$

C. Application

As illustrated in Fig. 2, 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.

IV. Experiments

We evaluate RoboAct-CLIP through offline action classification, simulated manipulation, and real-world robot experiments. The offline results assess the quality of the learned atomic action representations, while the simulation and real-world results evaluate their effectiveness for downstream policy learning. In all downstream experiments, RoboAct-CLIP is used as a frozen visual encoder. We further conduct ablations to analyze the contributions of the spatiotemporal modeling and feature disentanglement modules.

A. Action Recognition and Frame-Sampling Analysis

We first evaluate RoboAct-CLIP on offline classification to assess the quality of the learned atomic action representations. With 16 uniformly sampled frames, the model achieves **99.49%** accuracy for Action, **99.88%** for Subject, and **97.43%** for Object, indicating precise semantic understanding of manipulation videos. Removing the feature disentanglement module reduces action accuracy to **73.74%**, showing its importance for isolating action semantics from contextual cues. We also study the effect of temporal resolution: reducing the input to 8 frames lowers action accuracy to **96.48%** (Subject 99.30%, Object 95.10%), while increasing to 32 frames yields only marginal gains (Action **99.51%**, Subject 99.91%, Object 97.60%) at higher computational cost. Therefore, 16 frames provide a good trade-off between performance and efficiency for downstream tasks.

B. Simulation Experiments

a) Experiments on LIBERO.: We use the LIBERO [10] 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.

We compared several representation encoders under the same downstream policy-learning setup. All methods use the same policy architecture, inputs, and training protocol; the only difference is the encoder used to extract representations. The compared encoders are:

- **R3M** [26]: A ResNet50-based model pre-trained on visual data for general robotic representations.
- **MPI** [27]: A model with a multimodal transformer encoder and transformer decoder for predicting image-goal interaction states and detecting interaction objects. We evaluate both ViT-small and ViT-Base versions.
- **CLIP** [11]: The original CLIP model without fine-tuning, used as a visual feature encoder.
- **Robotic-CLIP** [16]: A CLIP variant fine-tuned on large-scale action videos to align paired frames with action descriptions, capturing action-centric visual cues for robotic manipulation.
- **RoboAct-CLIP (Ours)**: Our proposed model with temporal modeling and feature disentanglement.

RoboAct-CLIP attains the highest average success rates on LIBERO, improving the strongest baseline by average

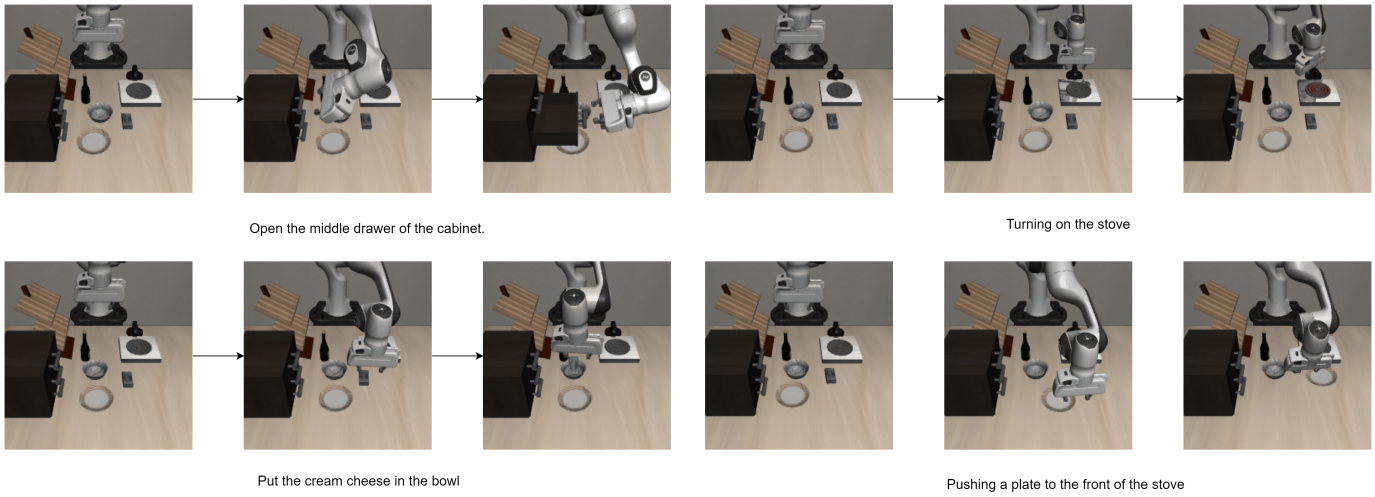

Fig. 3: RoboAct-CLIP performing four manipulation tasks in LIBERO. Each row is a task; the model maintains precise control across sequences.

(MPI(Base), 64.5%) to 76.5% (+12.0 percentage points (pps)) under a strictly frozen-encoder and identical policy setup. *Per-task trends.* Relative to MPI(Base), RoboAct-CLIP yields +28.0 pps on T1 (90.0% vs. 62.0%), +14.0 pps on T2 (84.0% vs. 70.0%), +12.0 pps on T3 (56.0% vs. 44.0%), and -6.0 pps 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 pps over Robotic-CLIP on T1 and T2) despite not leading on T3 (R3M: 58.0%) and T4 (MPI(Base): 82.0%).

b) 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 III, all baselines achieve 0% on both tasks, whereas **RoboAct-CLIP** attains **4.0%** (T5) and **16.0%** (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 *Spatiotemporal 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.

c) 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 IV 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*).

d) Limitations.: We note two limitations of RoboAct-CLIP. First, as an atomic action understanding model, downstream policies may fail on undecomposed multi-action sequences, since such clips no longer correspond to a single action primitive. In our following real-robot experiments, we address this by decomposing each task into a composition of atomic actions. Second, tool-use tasks can only remain a "single action" when their preconditions are already satisfied (e.g., the tool is already in hand); otherwise, they become multi-step sequences, such as *pick up* followed by *use it*.

C. Ablations

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

1) **A1 – Without Spatiotemporal Diff-Transformer.** We fed only the first and last frame features $\left(F_1, F_n\right)$ 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 Recombination 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 $\mathcal{L}_{\text{aux}}$ was dropped, leaving only contrastive and recombination losses, to quantify the value of categorical supervision for disentanglement.

TABLE IV: Success Rates (%) in Franka Kitchen Simulation Environment.

Encoder	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**

TABLE V: Real-world manipulation success rates (%).

Encoder	Subtask success				Avg. Success
	Open (middle drawer)	Pick (tape)	Place (on table)	Close (middle drawer)	
MPI(Base)	50.0	46.7	42.9	33.3	43.2
CLIP	43.3	30.8	50.0	0.0	31.0
Robotic-CLIP	53.3	37.5	50.0	33.3	43.5
RoboAct-CLIP (Ours)	**66.7**	**60.0**	**58.3**	**71.4**	**64.1**

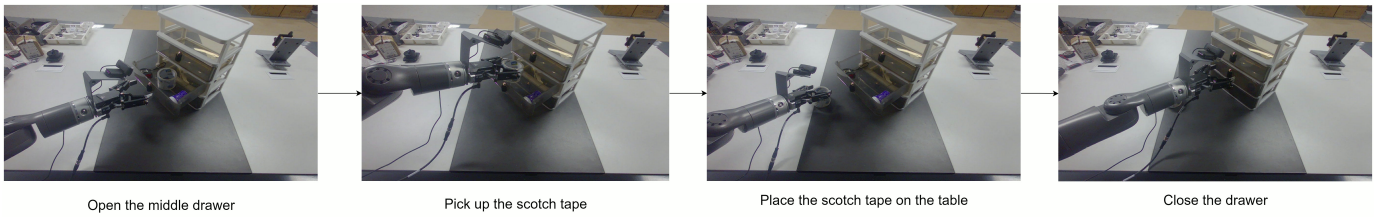

Fig. 4: Execution sequence of the real-world task using RoboAct-CLIP.

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 Table II clearly demonstrate the importance of both proposed components. Removing the Spatiotemporal Diff-Transformer (Ablation 1) led to a significant performance drop of 14.0 pps 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 pps 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.

D. Real-World Robot Experiments

a) Real-world setup and data collection.: We deploy RoboAct-CLIP on a real robotic platform built around a JAKA K-1 humanoid arm. The system consists of a 7-DoF arm (3 kg payload, 760 mm reach), a VI-H six-axis force–torque sensor, an AG-105-145 electric parallel gripper, and an Orbbec Gemini336L RGB-D camera that observes the drawer and tabletop workspace. Demonstration data are collected in the actual deployment environment via manual teleoperation: an expert operator teleoperates the arm and gripper to perform the full manipulation sequence while we record joint states, gripper commands, images, and force signals.

b) Policy training and task.: We then train a task-specific policy on these teleoperated trajectories while keeping all visual encoders frozen. The policy is trained to execute a four-step manipulation task: (1) open the middle drawer, (2) pick up the roll of tape, (3) place it on the table, and (4) close the drawer. RoboAct-CLIP serves as the visual backbone and transfers from simulation to the real setup, with temporal modeling improving robustness to variations in lighting, appearance, and dynamics.

c) Analysis.: Table V summarizes per-subtask and average success rates. RoboAct-CLIP attains an average of **64.1%**, outperforming Robotic-CLIP (43.5%), MPI(Base) (43.2%), and CLIP (31.0%) by **+20.6 pps**, **+20.9 pps**, and **+33.1 pps**, respectively. Per-task gains over the best baseline are: *Open* $+13.4pps$; *Pick* $+13.3pps$; *Place* $+8.3pps$, and *Close* $+38.1pps$. The largest improvement appears on the terminal *Close* step, where success depends on accumulating temporal evidence and the net outcome of actuation; our Spatiotemporal 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* $+13.3pps$). 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.

V. Conclusions

We presented **RoboAct-CLIP**, a VLA-oriented approach for atomic action understanding that combines a heterogeneous single-action training set with a spatiotemporal disentanglement architecture on top of a frozen CLIP backbone. By separating subject, action, and object factors from manipulation videos and enforcing feature recombination under cross-modal consistency constraints, RoboAct-CLIP learns subject-invariant atomic action representations and supports cross-subject action alignment. Empirically, RoboAct-CLIP achieves **99.49%** action classification accuracy, improves success rates by **12.0** and **5.7** percentage points on LIBERO and Franka Kitchen, respectively, and supports long-horizon real-world task execution through atomic-action decomposition and recomposition. Ablation studies further confirm the effectiveness of the proposed design. Overall, RoboAct-CLIP demonstrates that the learned action-focused video representations can improve both atomic action understanding and downstream policy learning in VLA systems. Our future research will be focused on a better physical motion generation framework with the learned intrinsic action representation.

References

[1] P. Li, Z. An, S. Abrar, and L. Zhou, "Large language models for multi-robot systems: A survey," 2025. [Online]. Available: https://arxiv.org/abs/2502.03814

[2] H. Jeong, H. Lee, C. Kim, and S. Shin, "A survey of robot intelligence with large language models," *Applied Sciences*, vol. 14, no. 19, p. 8868, 2024.

[3] Y. Ma, Z. Song, Y. Zhuang, J. Hao, and I. King, "A survey on vision-language-action models for embodied ai," *arXiv preprint arXiv:2405.14093*, 2024.

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

[5] X. Li, M. Liu, H. Zhang, C. Yu, J. Xu, H. Wu, C. Cheang, Y. Jing, W. Zhang, H. Liu, H. Li, and T. Kong, "Vision-language foundation models as effective robot imitators," *arXiv preprint arXiv:2311.01378*, 2023.

[6] B. Chen, Z. Xu, S. Kirmani, B. Ichter, D. Driess, P. Florence, D. Sadigh, L. Guibas, and F. Xia, "Spatialvlm: Endowing vision-language models with spatial reasoning capabilities," 2024. [Online]. Available: https://arxiv.org/abs/2401.12168

[7] M. Zare, P. M. Kebria, A. Khosravi, and S. Nahavandi, "A survey of imitation learning: Algorithms, recent developments, and challenges," *IEEE Transactions on Cybernetics*, 2024.

[8] M. Lázaro-Gredilla, D. Lin, J. S. Guntupalli, and D. George, "Beyond imitation: Zero-shot task transfer on robots by learning concepts as cognitive programs," *Science Robotics*, vol. 4, no. 26, p. eaav3150, 2019.

[9] Z. Miao, J. Lv, H. Fang, Y. Jin, and C. Lu, "Knowledge-driven imitation learning: Enabling generalization across diverse conditions," *arXiv preprint arXiv:2506.21057*, 2025.

[10] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, "Libero: Benchmarking knowledge transfer for lifelong robot learning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 44 776–44 791, 2023.

[11] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PmLR, 2021, pp. 8748–8763.

[12] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds *et al.*, "Flamingo: a visual language model for few-shot learning," *Advances in neural information processing systems*, vol. 35, pp. 23 716–23 736, 2022.

[13] A. Awadalla, I. Gao, J. Gardner, J. Hessel, Y. Hanafy, W. Zhu, K. Marathe, Y. Bitton, S. Gadre, S. Sagawa *et al.*, "Openflamingo: An open-source framework for training large autoregressive vision-language models," *arXiv preprint arXiv:2308.01390*, 2023.

[14] N. Wang, D. Liao, and X. Xu, "Multi-scale temporal difference transformer for video-text retrieval," *arXiv preprint arXiv:2406.16111*, 2024.

[15] Y. Wang, R. Xiao, J. Y. L. Kasahara, R. Yajima, K. Nagatani, A. Yamashita, and H. Asama, "Dart-llm: Dependency-aware multi-robot task decomposition and execution using large language models," *arXiv preprint arXiv:2411.09022*, 2024.

[16] N. Nguyen, M. N. Vu, T. D. Ta, B. Huang, T. Vo, N. Le, and A. Nguyen, "Robotic-clip: Fine-tuning clip on action data for robotic applications," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 5930–5936.

[17] K. Bae, G. Ahn, Y. Kim, and J. Choi, "Devias: Learning disentangled video representations of action and scene," in *European Conference on Computer Vision*. Springer, 2024, pp. 431–448.

[18] M. Pan, J. Zhang, T. Wu, Y. Zhao, W. Gao, and H. Dong, "Omnimanip: Towards general robotic manipulation via object-centric interaction primitives as spatial constraints," *arXiv preprint arXiv:2501.03841*, 2025.

[19] A. 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.

[20] H.-S. Fang, H. Fang, Z. Tang, J. Liu, C. Wang, J. Wang, H. Zhu, and C. Lu, "Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 653–660.

[21] H. R. Walke, K. Black, T. Z. Zhao, Q. Vuong, C. Zheng, P. Hansen-Estruch, A. W. He, V. Myers, M. J. Kim, M. Du *et al.*, "Bridgedata v2: A dataset for robot learning at scale," in *Conference on Robot Learning*. PMLR, 2023, pp. 1723–1736.

[22] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi *et al.*, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv preprint arXiv:2501.12948*, 2025.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[24] Z. Zhang, Q. Zhang, X. Wu, X. Shi, G. Liao, Y. Wang, X. Wang, and D. Zhao, "User response modeling in reinforcement learning for ads allocation," in *Companion Proceedings of the ACM Web Conference 2024*, ser. WWW '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 131–140. [Online]. Available: https://doi.org/10.1145/3589335.3648310

[25] Z. Wang, G. Liao, X. Shi, X. Wu, C. Zhang, Y. Wang, X. Wang, and D. 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*, ser. CIKM '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 3555–3564. [Online]. Available: https://doi.org/10.1145/3511808.3557094

[26] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," *arXiv preprint arXiv:2203.12601*, 2022.

[27] J. Zeng, Q. Bu, B. Wang, W. Xia, L. Chen, H. Dong, H. Song, D. Wang, D. Hu, P. Luo *et al.*, "Learning manipulation by predicting interaction," *arXiv preprint arXiv:2406.00439*, 2024.