ATTEND AS YOU ACT: PROGRESS-GUIDED ADAPTIVE CONTEXTUALIZATION FOR VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning from a single video demonstration through in-context learning (ICL) offers a promising way to adapt vision-language-action (VLA) models without task-specific data collection and fine-tuning. Effective adaptation, however, requires demonstration use to evolve with execution: the policy must identify which segments are currently relevant and determine how the demonstrated behavior should be instantiated in the present scene. We propose ***PROACT***, an execution-conditioned one-shot ICL framework that adapts both demonstration representation and behavior grounding. Progress-guided multi-resolution routing dynamically concentrates visual detail and attention on temporally relevant demonstration regions while compressing less relevant context. Predictive behavior grounding further learns a latent operator that links demonstrated behavior to its expected consequences under the current observation, making it actionable in the present scene. Together, these mechanisms transform a fixed video demonstration into a dynamic task context that evolves with execution. Experiments across diverse manipulation settings show that PROACT consistently outperforms approaches that treat the demonstration as static context.

1 INTRODUCTION

Vision-language-action (VLA) models offer a promising foundation for learning general-purpose robot policies from visual observations and language instructions (Kim et al., 2024; Kawaharazuka et al., 2025; Vuong et al., 2023; Black et al., 2025b; Yuan et al., 2026; Bjorck et al., 2025). Despite recent progress, adapting these models to new tasks often still requires collecting task-specific robot data and performing additional fine-tuning (Yang et al., 2026; Kim et al., 2025b), which limits scalable deployment. This makes in-context learning (ICL) (Brown et al., 2020; Dong et al., 2024) particularly appealing for robot learning: a video demonstration can directly provide task context at inference time, capturing how the task unfolds without requiring action annotations (Jang et al., 2022; Jain et al., 2024).

However, a demonstration video is not uniformly useful throughout execution. At any moment, only a small portion of the video may deserve a *"closer look."* Consider a simple pick-and-place task: while the robot is reaching for an object, the frames around the demonstrated grasp contain the fine-grained visual details that matter most; once the object is lifted, the frames depicting transport and placement become more relevant. The useful evidence therefore shifts through the demonstration as execution unfolds. Treating the entire video at a fixed resolution not only wastes computation on temporally irrelevant content, but may also dilute the visual details needed for the current decision.

Another challenge lies in grounding the demonstrated behavior. The demonstrated behavior is performed in a specific scenario, whereas the robot must carry it out in the current scene while dealing with different object configurations. What the robot needs is therefore not a literal replay of the retrieved frames, but a context-dependent reinterpretation of the behavior they convey. This leads to two coupled requirements for learning from video demonstrations: **temporal relevance**, which identifies where the useful evidence lies as execution progresses, and **behavior grounding**, which determines how that evidence should be instantiated in the current scene. We refer to their joint

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

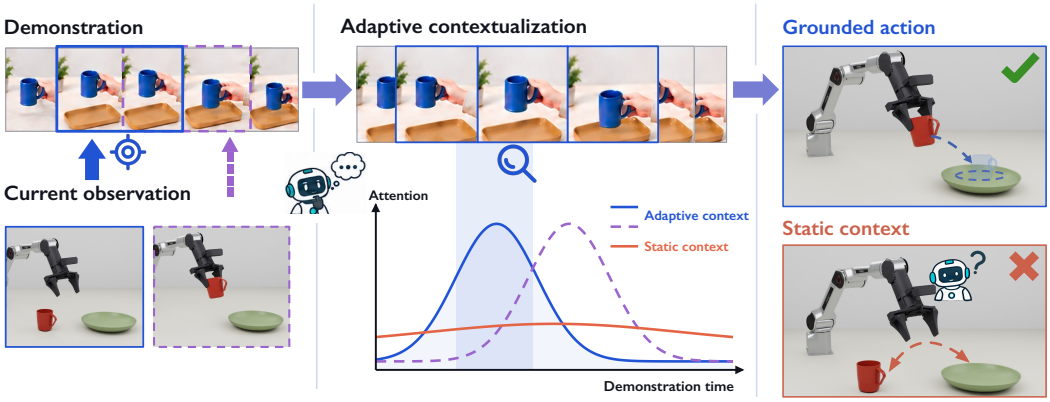


Figure 1: **Key insight.** PROACT dynamically focuses computation on relevant reference moments with action grounding, whereas static context can dilute attention across competing cues.

adaptation as **adaptive contextualization**: the demonstration should evolve from a static visual record into an execution-conditioned reference (Figure 1).

Motivated by this, we introduce ***PROACT*** (*Progress-Guided Adaptive ConTextualization*), a framework for one-shot VLA in-context learning that continuously reshapes how a demonstration is used as robot execution evolves. Rather than processing the demonstration in the same way at every step, PROACT first estimates where the robot currently lies relative to the demonstrated trajectory. This progress estimate provides a temporal prior for deciding which parts of the demonstration deserve a "closer look" and which can recede into a coarser representation.

As execution progresses, PROACT adapts the demonstration in two complementary ways. Through *progress-guided multi-resolution routing*, it redistributes representational capacity across the video, preserving fine-grained detail around temporally relevant regions while maintaining a coarse view of the remaining context. Through *predictive behavior grounding*, it learns how the retrieved evidence should translate into expected consequences under the robot's current observation. Together, these mechanisms let the policy adapt not only *where* it looks in the demonstration, but also *how* the demonstrated behavior is carried forward into the current scene.

The contributions of this paper can be summarized as follows:

- We introduce PROACT, a framework for adaptive demonstration contextualization in VLA in-context learning, enabling demonstration use to actively evolve with execution.
- We develop *progress-guided multi-resolution routing* to adaptively allocate visual detail across the demonstration, together with *predictive behavior grounding* to translate retrieved evidence into actionable behavior in the current scene.
- Extensive experiments demonstrate that adaptive contextualization consistently improves one-shot video-conditioned manipulation over existing approaches with static context.

2 Related Work

One-Shot Imitation Learning for Robotics. Vision-language-action (VLA) models combine semantic knowledge from vision-language pretraining with robot data to learn a broad range of manipulation skills (Zitkovich et al., 2023; Kim et al., 2024; Black et al., 2025b). However, extending this repertoire to new tasks often relies on additional robot data and task-specific fine-tuning. One-shot imitation addresses this adaptation problem by learning to infer the desired behavior from a single demonstration (Duan et al., 2017; James et al., 2018). Building on demonstration conditioning and multimodal prompting (Jiang et al., 2023), recent work develops in-context policies through autoregressive prediction (Fu et al., 2024), graph diffusion (Vosylius & Johns, 2025), behavior prompting (Patel et al., 2026), and retrieval-based adaptation (Sridhar et al., 2025). Video-conditioned approaches (Chen et al., 2026; Zhou et al., 2026), such as Vid2Robot (Jain et al., 2024) and MimicDroid (Shah et al., 2025), further enable human videos to serve as task prompts. Another line

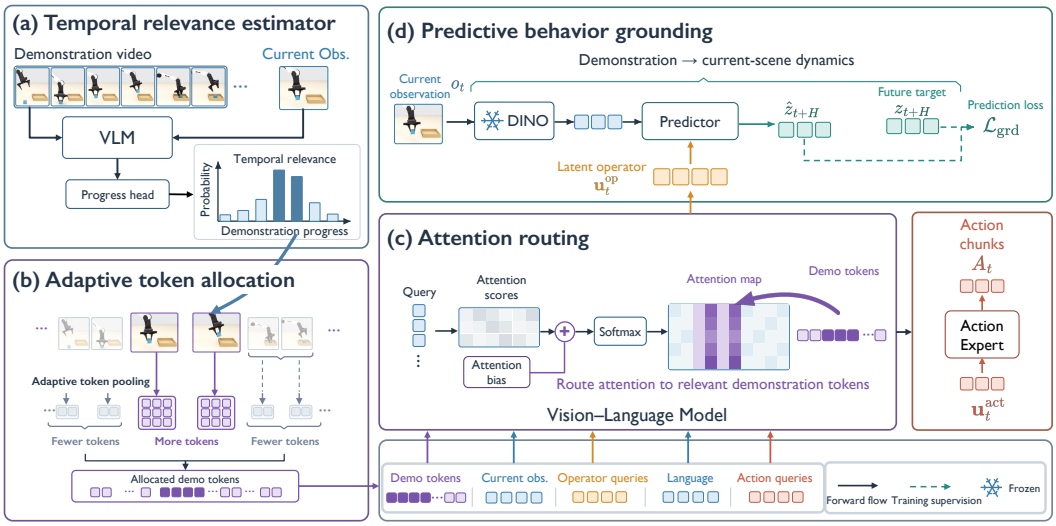


Figure 2: **Overview of PROACT.** (a) Temporal Relevance Estimator identifies demonstration regions relevant to the current execution state. (b) Adaptive Token Allocation preserves finer visual detail in relevant regions while compressing the remaining context. (c) Attention Routing steers attention using temporal relevance. (d) Predictive Behavior Grounding learns a latent behavior representation by predicting its visual consequence in the current scene.

of work incorporates demonstrations through test-time adaptation, updating fast-weight or auxiliary memory during inference (Feng et al., 2026; Jiang et al., 2026). Despite advances, these approaches largely treat the demonstration as fixed conditioning, without explicitly adapting its interpretation to the evolving dynamics of robot execution.

Latent Action and Predictive Representations for Robot Control. Latent action and motion representations support learning behavioral structure from videos without action annotations (Schmidt & Jiang, 2024; Bruce et al., 2024; Ye et al., 2025; Chen et al., 2025; Zhu et al., 2026). XSkill (Xu et al., 2023) and UniSkill (Kim et al., 2025a) learn cross-embodiment skills for conditioning robot policies on human videos. A complementary line connects visual prediction to control: GR-1 (Wu et al., 2023) and GR-2 (Cheang et al., 2024) combine video prediction with action learning, while Demo-JEPA (He et al., 2026) predicts target-compatible latent trajectories for planning. In contrast, PROACT uses predictive latent representations to ground demonstrated behavior in the current execution context.

3 Problem Formulation

We study one-shot in-context learning for robotic manipulation. At deployment, the robot is provided with a single video demonstration $\mathcal{D}$ that visually specifies the desired task, together with an optional language instruction ℓ. At execution step t, the policy receives the current observation o_t and robot state r_t, and predicts an action chunk $A_t = (a_t, \ldots, a_{t+H_{\text{act}}-1})$ of length H_{act}:

$$A_t \sim \pi_\theta(A_t \mid \mathcal{D}, o_t, \ell, r_t)\,. \tag{1}$$

The policy parameters remain fixed throughout deployment, and the demonstration provides visual evidence only: no demonstrator action annotations are required.

4 Method

We introduce ***PROACT*** (*Progress-Guided Adaptive ConTextualization*), a one-shot VLA in-context learning framework that addresses two challenges: identifying demonstration evidence that is temporally relevant to the current execution state, and grounding the demonstrated behavior in the

robot's current scene. As illustrated in Figure 2, PROACT first uses a **Temporal Relevance Estimator** (Sec. 4.1; Figure 2(a)) to infer which parts of the demonstration are relevant to the current execution state. The resulting relevance distribution then adapts the demonstration in two complementary ways (Sec. 4.2; Fig. 2(b,c)): **Adaptive Token Allocation** preserves finer visual detail in relevant regions while compressing less relevant context, and **Attention Routing** steers contextual reasoning toward the corresponding demonstration tokens. Finally, **Predictive Behavior Grounding** (Sec. 4.3) extracts a latent behavior representation and grounds it through future visual outcome prediction in the current scene (Sec. 4.3; Figure 2(d)). Together, these components determine both *which* demonstration evidence is emphasized and *how* the demonstrated behavior is instantiated during execution. Algorithm 1 summarizes the resulting routing and training procedure.

4.1 Temporal Relevance Estimation

A demonstration spans the entire task, while the evidence relevant to the current decision changes as execution progresses. We introduce a Temporal Relevance Estimator that predicts a soft relevance distribution over the demonstration conditioned on the current execution state.

Let the sampled demonstration $\mathcal{D}$ contain L temporal frames, indexed by $j \in \{1, \dots, L\}$. We partition these frames into K overlapping temporal windows $\{\mathcal{W}_k\}_{k=1}^{K}$, each spanning approximately $2/K$ of the normalized demonstration timeline, with adjacent windows overlapping by half. For each frame j, the vision encoder and multimodal projector in our VLA produce a sequence of projected visual tokens $V_j = \{v_{j,n}\}_{n=1}^{N_v}$. The visual tokens associated with window $\mathcal{W}_k$ are $V_{\mathcal{W}_k} = \bigcup_{j \in \mathcal{W}_k} V_j$. We summarize the window by mean-pooling: $\bar{v}_k = \frac{1}{|V_{\mathcal{W}_k}|} \sum_{v \in V_{\mathcal{W}_k}} v$. For relevance estimation, we jointly process the K window summaries $\{\bar{v}_k\}_{k=1}^{K}$, the projected visual tokens of the current observation, and the instruction ℓ with the Transformer blocks in our VLA. The final hidden state corresponding to $\bar{v}_k$ is used as the contextualized representation of the k-th demonstration window, denoted by $h_k^{\mathcal{D}}$. We mean-pool the final hidden states at the current-observation visual-token positions to obtain the contextualized observation representation h_t. Their compatibility defines a soft temporal relevance distribution:

$$q_t(k) = \mathrm{softmax}_k \left(\frac{(W_Q h_t)^\top (W_K h_k^{\mathcal{D}})}{\sqrt{d_p}} \right), \tag{2}$$

where d_p is the projected query/key dimension. Thus, $q_t(k)$ estimates the relevance of demonstration window $\mathcal{W}_k$ to the current execution state.

During training, we obtain a normalized demonstration-progress coordinate $\tau_t \in [0, 1]$ by aligning the execution and demonstration trajectories with dynamic time warping (DTW) (Jang et al., 2026; Memmel et al., 2025), using visual features from a pretrained embedding model (Li et al., 2026). Rather than supervising a single window, we convert τ_t into a soft target distribution over K ordered progress anchors $\xi_k = (k-1)/(K-1)$:

$$q_t^\star(k) = \frac{\left[1 - \frac{|\tau_t - \xi_k|}{1/(K-1)}\right]_+}{\sum_{r=1}^{K} \left[1 - \frac{|\tau_t - \xi_r|}{1/(K-1)}\right]_+}. \tag{3}$$

Here, $[x]_+ = \max(x, 0)$. We supervise the predicted relevance distribution using the normalized discrete Wasserstein-1 distance:

$$\mathcal{L}_{\mathrm{prog}} = \frac{1}{K-1} \sum_{k=1}^{K-1} \left| \sum_{r=1}^{k} q_t(r) - \sum_{r=1}^{k} q_t^\star(r) \right|. \tag{4}$$

This objective penalizes relevance mass by its displacement along the demonstration timeline.

4.2 Progress-Guided Multi-Resolution Routing

Processing the complete demonstration preserves all visual tokens, but its computational cost grows with video length and redundant context can dilute relevant cues (Bansal et al., 2026). Uniform compression reduces this cost, but allocates visual detail indiscriminately across the timeline. Conversely, selecting a single temporal window removes redundancy but introduces a hard,

non-differentiable boundary that may exclude useful evidence immediately outside the selected region. We therefore use the estimated temporal relevance as an execution-conditioned routing signal, adapting both visual-token allocation and attention over the demonstration.

Adaptive token allocation. Given the window-level relevance distribution q_t, we project it onto the demonstration timeline to obtain a frame-level relevance distribution:

$$p_t(j) = \sum_{k:\, j \in \mathcal{W}_k} \frac{q_t(k)}{|\mathcal{W}_k|}, \qquad \text{where } \sum_{j=1}^{L} p_t(j) = 1. \tag{5}$$

Given a total demonstration-token budget $B \geq mL$, we allocate a real-valued budget to each frame:

$$\widetilde{b}_t(j) = m + (B - mL)p_t(j), \tag{6}$$

where m is the minimum number of tokens retained from every frame. Because $\sum_j p_t(j) = 1$, the continuous allocations satisfy $\sum_j \widetilde{b}_t(j) = B$. We convert $\widetilde{b}_t(j)$ into feasible integer budgets $b_t(j)$ through capacity-aware rounding under $m \leq b_t(j) \leq N_v$, redistributing any allocation assigned to frames that have reached their token capacity. We then adaptively mean-pool the N_v projected tokens of frame j into $b_t(j)$ retained tokens, denoted by $\widetilde{V}_{t,j}$. Concatenating the retained features yields the state-dependent demonstration representation

$$\widetilde{V}_{\mathcal{D},t} = \left[\widetilde{V}_{t,1}; \ldots; \widetilde{V}_{t,L}\right]. \tag{7}$$

Consequently, temporally relevant frames retain finer visual detail, whereas less relevant frames remain available as compressed coarse context.

Relevance-guided attention routing. Beyond determining which visual tokens are retained, we use the same relevance distribution to steer attention toward tokens from temporally relevant frames. For a query token at position i, let $\mathcal{K}_i$ denote the set of keys visible to query i under the original VLA attention mask. For each retained demonstration token a originating from temporal frame $j(a)$, we augment its pre-softmax attention logit with routing bias as:

$$s_{i,a} = \frac{Q_i^\top K_a}{\sqrt{d_p}} + \Delta_t(a), \qquad \Delta_t(a) = \log\big(Lp_t(j(a))\big). \tag{8}$$

All non-demonstration logits remain unchanged. After softmax, this is equivalent to scaling the unnormalized attention weight of each demonstration token by its frame-level relevance $Lp_t(j(a))$. When $p_t(j) = 1/L$ for all j, the routing term vanishes and the original attention is recovered. Importantly, the routing operation is differentiable with respect to the relevance distribution, allowing downstream action supervision to further refine temporal relevance estimation. This routing update is summarized in Algorithm 1.

4.3 Predictive Behavior Grounding

Temporal relevance determines which demonstration evidence should be emphasized, but does not by itself specify how the demonstrated behavior should be instantiated in the robot's current scene. We therefore learn a demonstration-conditioned latent behavior representation and ground it through the visual consequence of the robot's own execution. During training, learned operator queries extract this representation from the routed context, and a lightweight predictor uses it together with the current observation to predict future latent features. Since the prediction target is obtained from the robot's trajectory rather than the demonstration, this objective encourages the latent representation to capture scene-specific behavioral consequences.

Behavior and action representations. Let V_t^{obs} denote the projected visual tokens of the current observation, Q^{op} a set of N_{op} learnable operator queries, and E^{act} the action-token embeddings. Using the routed demonstration representation $\widetilde{V}_{\mathcal{D},t}$ and the attention routing in Eq. 8, the VLA jointly processes

$$\left[\widetilde{V}_{\mathcal{D},t}; V_t^{\text{obs}}; Q^{\text{op}}; \ell; E^{\text{act}}\right] \xrightarrow{\text{VLA}_\theta} \left(\mathbf{u}_t^{\text{op}}, \mathbf{u}_t^{\text{act}}\right). \tag{9}$$

The final hidden states at the operator-query positions form $\mathbf{u}_t^{\text{op}} \in \mathbb{R}^{N_{\text{op}} \times d}$, which serves as the latent behavior representation. The hidden states at the action-token positions form $\mathbf{u}_t^{\text{act}}$. Conditioned on $\mathbf{u}_t^{\text{act}}$ and the robot state r_t, the action expert predicts an action chunk:

$$A_t \sim \pi_\theta\big(A_t \mid \mathbf{u}_t^{\text{act}}, r_t\big). \tag{10}$$

Table 1: **Success rate on LIBERO-Gen benchmark.** For each suite, tasks are categorized into two classes. *Overall* reports the success rate averaged over all tasks in the suite.

Method	*Combination*			*Chain*		
	Relative	*Support*	*Overall*	*Placement*	*Mixed*	*Overall*
UniSkill (2025a)	66.7	70.6	69.4	44.0	65.5	52.6
Goal-Image	22.0	59.1	48.0	40.0	75.0	54.0
Language	87.3	69.1	76.0	73.3	86.0	78.4
BPP (2026)	89.3	73.1	78.0	84.7	86.0	85.2
PROACT	**94.0**$_{+4.7}$	**81.1**$_{+8.0}$	**85.0**$_{+7.0}$	**88.7**$_{+4.0}$	**87.5**$_{+1.5}$	**88.2**$_{+3.0}$

Grounding through future visual outcomes. Let H denote the prediction horizon and ϕ be a frozen DINOv3 encoder (Siméoni et al., 2025). We extract dense visual features from the current and future robot observations:

$$z_t = \mathrm{sg}[\phi(o_t)]\,, \qquad z_{t+H} = \mathrm{sg}[\phi(o_{t+H})]\,, \tag{11}$$

where sg denotes stop-gradient. The future observation o_{t+H} is sampled from the robot's own execution trajectory. A lightweight predictor conditions on the current-scene features and latent behavior representation to predict a residual future feature:

$$\widehat{z}_{t+H} = z_t + f_\psi(z_t, \mathbf{u}_t^{\mathrm{op}})\,. \tag{12}$$

The predictor combines spatial self-attention over current visual features z_t with cross-attention to $\mathbf{u}_t^{\mathrm{op}}$, predicting the visual consequence of the inferred behavior in the current scene. This grounds $\mathbf{u}_t^{\mathrm{op}}$ in scene-specific behavioral outcomes. At inference time, the predictor and predicted future features are not required for action generation.

Training objectives. For valid dense feature locations Ω, we optimize

$$\mathcal{L}_{\mathrm{grd}} = \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \left[1 - \cos(\widehat{z}_{t+H,\omega}, z_{t+H,\omega})\right] \tag{13}$$

We jointly optimize action prediction, temporal relevance estimation, and predictive behavior grounding:

$$\mathcal{L} = \mathcal{L}_{\mathrm{act}} + \lambda_{\mathrm{prog}}\mathcal{L}_{\mathrm{prog}} + \lambda_{\mathrm{grd}}\mathcal{L}_{\mathrm{grd}}. \tag{14}$$

Implementation details are provided in Appendix A.

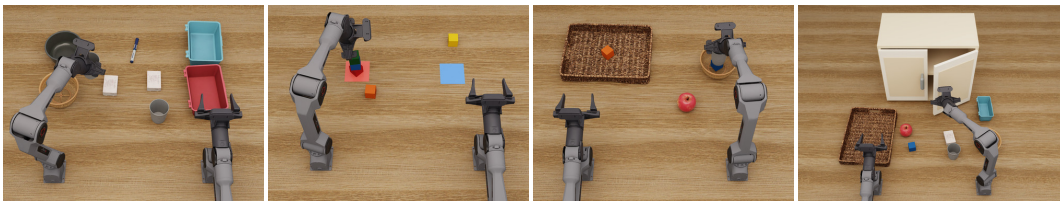
5 Experiments

5.1 Experimental Setup

Benchmarks. We evaluate PROACT on three challenging benchmarks: **LIBERO-Gen** (Patel et al., 2026), which tests compositional generalization to unseen task configurations and skill sequences; **IsaacSim-Gen**, our manipulation benchmark covering spatial grounding, temporal ordering, and long-horizon manipulation; and **RoboMME** (Dai et al., 2026), where we use *PatternLock* and *RouteStick* to evaluate complex demonstration-guided sequential execution. Benchmark details and evaluation protocols are provided in Appendix B.1.

Compared methods. We mainly compare PROACT against representative task-conditioning strategies. **UniSkill** (Kim et al., 2025a) extracts embodiment-agnostic skill representations from demonstration videos and uses them to condition robot execution. **BPP** (Patel et al., 2026) treats a demonstration as an in-context behavior prompt for action prediction. **Language** uses the same backbone as the proposed method but conditions the policy only on language instructions, without access to the demonstration. **Goal-Image** replaces the full demonstration with its final frame, providing the desired task outcome while removing the intermediate temporal information.

Implementation. We use Qwen3-VL-4B-Instruct as the vision-language backbone and condition a flow-matching action head on the hidden states of 16 action query tokens. The models are trained using AdamW optimizer for 60k steps, with learning rates of 10^{-5} for the backbone and 10^{-4} for the action head, following a cosine decay schedule with 6k warmup steps. At inference time, actions are generated using 20 Euler integration steps.



(a) Spatial Pick-and-Place (b) Stack Block (c) Sequential Table Clearing (d) Cabinet Manipulation

Figure 3: **Task Visualization of IsaacSim-Gen.** We construct four manipulation tasks to evaluate spatial grounding, temporal ordering, and long-horizon execution.

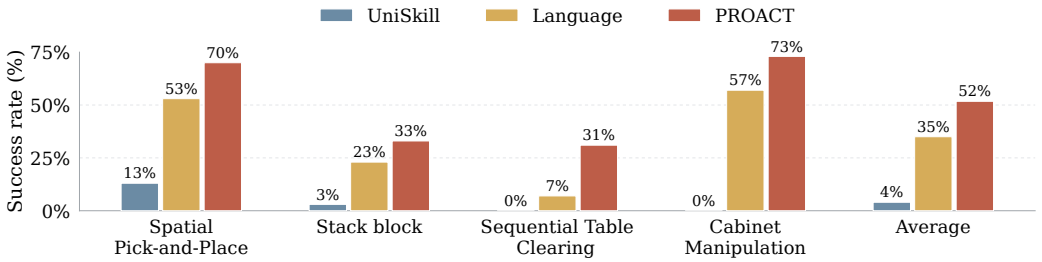


Figure 4: **Success rate on IsaacSim-Gen.** *Average* denotes the unweighted mean across the four tasks. PROACT consistently outperforms the baselines across all tasks.

5.2 Main Results

LIBERO-Gen. We evaluate on the *Combination* and *Chain* suites of LIBERO-Gen (Patel et al., 2026). *Combination* tests generalization to unseen recombinations of familiar task factors, while *Chain* composes familiar manipulation behaviors into longer task sequences. Unlike the original setup, PROACT requires only the demonstration video, without access to demonstration actions or proprioceptive states. As shown in Table 1, PROACT consistently outperforms the baselines across both suites, improving on both unseen configurations and behavior sequences. The gains are particularly clear on *Combination*, where the demonstrated behavior must be instantiated under new object–goal configurations. Interestingly, demonstration-conditioned baselines do not always outperform the language-only policy, despite receiving additional visual task information. This suggests that simply providing a demonstration is insufficient; effectively relating its content to the evolving execution state is important for exploiting it as in-context information.

IsaacSim-Gen. We further construct four manipulation tasks in IsaacSim-Gen covering spatial grounding, temporal ordering, and long-horizon skill composition, as visualized in Figure 3. Evaluation uses held-out combinations of object choices, destinations, and action sequences. Figure 4 shows that PROACT consistently outperforms both baselines across all four tasks. UniSkill performs substantially worse than the language-only policy, suggesting that representing a demonstration as a pre-extracted sequence of skills is insufficient for these settings. In contrast, PROACT continuously contextualizes the demonstration according to the current execution state, enabling more effective use of the same video context.

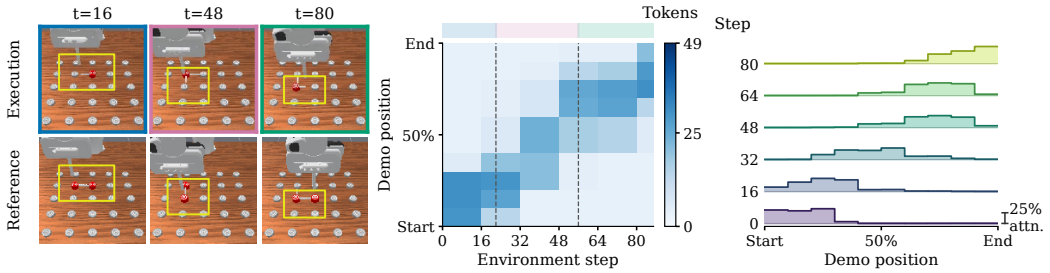
RoboMME. We evaluate *PatternLock* and *RouteStick* from RoboMME (Dai et al., 2026), where the robot must follow a demonstrated spatial sequence and interact with target locations in the prescribed order. These tasks provide an explicit test of maintaining correspondence between execution progress and a temporally structured reference. As reported in Table 2, PROACT substantially improves over the strongest baseline on both tasks, improving the overall success rate by 18%. These results highlight the benefit of dynamically adapting demonstration context as execution progresses in tasks that require sequential reference following.

5.3 Analysis of Adaptive Contextualization

To understand how PROACT adapts a fixed demonstration throughout execution, we visualize its internal behavior on *PatternLock*, whose ordered structure makes temporal relevance directly interpretable. Figure 5(a) shows the estimated correspondence between the current execution and the

Table 2: **Success rates on the *PatternLock* and *RouteStick*.** The evaluation follows Dai et al. (2026), where OpenVLA* is an OpenVLA-OFT variant that incorporates sampled past observations into the model's input. MME-VLA uses sampled past observations to modulate its internal features.

Method	*PatternLock*				*RouteStick*			
	Easy	*Medium*	*Hard*	*Overall*	*Easy*	*Medium*	*Hard*	*Overall*
$\pi_{0.5}$ (2025a)	7.7	0	0	4	7.7	0	0	4
OpenVLA* (2024)	26.9	0	0	14	46.2	16.7	8.3	30
MME-VLA (2026)	84.6	41.7	0	54	96.2	41.7	41.7	70
PROACT	**92.3**$_{+7.7}$	**75**$_{+33.3}$	**25**$_{+25.0}$	**72**$_{+18.0}$	**100**$_{+3.8}$	**91.6**$_{+49.9}$	**58.3**$_{+16.6}$	**88**$_{+18.0}$



(a) Execution–Reference Alignment (b) Adaptive Token Allocation (c) Progress-Aware Attention

Figure 5: **Visualization on *PatternLock*.** The robot follows a demonstrated sequence of target locations, causing the relevant reference segment to change throughout execution. (a) The estimated execution-reference correspondence shifts as the execution proceeds. (b) Adaptive token allocation assigns more representational capacity to the currently relevant demonstration region while compressing less relevant context. (c) Policy attention shifts accordingly over time, illustrating how the demonstration is dynamically interpreted as the task unfolds.

reference demonstration. As execution progresses, the corresponding reference stage shifts forward along the demonstration, indicating that the model tracks which demonstration region is currently relevant. This evolving relevance directly reshapes the demonstration representation: as shown in Figure 5(b), the region receiving more visual tokens progressively advances along the demonstration as execution unfolds, while less relevant regions are represented more compactly. Figure 5(c) further illustrates how this adaptation shapes the policy's use of the demonstration. The attention distribution shifts toward later regions of the demonstration with task progress, placing greater emphasis on temporally relevant regions. Together, these results show that execution progress consistently governs how the demonstration is represented and attended to, allowing a fixed demonstration to be dynamically contextualized throughout execution. Combined with the substantial gains reported in Table 2, these observations support the effectiveness of adaptive contextualization. Additional visualizations are provided in Appendix C.

5.4 Ablation Studies

We compare PROACT against four variants to disentangle the contributions of its core components: ***(i) Adaptive Allocation w/o Attention Routing***, which removes the relevance-dependent attention bias $\Delta_t(a)$ in Eq. 8; ***(ii) Hard Selection***, which retains only the most relevant demonstration window according to the predicted temporal relevance distribution; ***(iii) Uniform Allocation w/ Attention Routing***, which retains relevance-guided attention but distributes the same reduced token budget uniformly across the demonstration timeline; and ***(iv) w/o predictive behavior grounding***, which removes the predictive grounding objective $\mathcal{L}_{\text{grd}}$ while keeping all remaining components unchanged. Figure 6 presents the ablation results on LIBERO-Gen Combination. *Adaptive Allocation w/o Attention Routing* and *Uniform Allocation w/ Atten-*

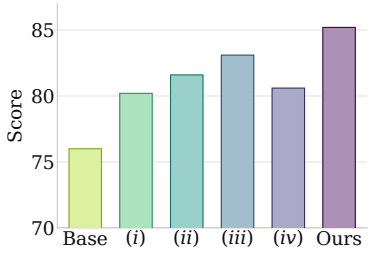


Figure 6: **Ablation studies.** We provide ablation of removing attention routing, adaptive token allocation, and behavior grounding.

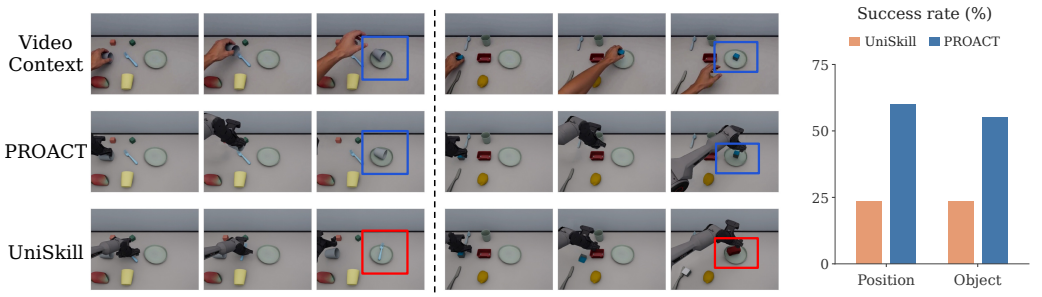


Figure 7: **Learning from human video demonstrations.** Despite the substantial embodiment gap, PROACT effectively uses human-hand videos as demonstration context to guide robot execution.

tion Routing both underperform PROACT, demonstrating the complementary benefits of relevance-guided attention and state-dependent token allocation: the former steers contextual reasoning toward relevant content, while the latter allocates visual capacity according to the execution state. *Hard Selection* also lags behind soft routing, as it discards coarse context outside the selected window and prevents action supervision from refining the relevance estimate through the differentiable routing pathway. Finally, *w/o Predictive Behavior Grounding* incurs a further performance drop, indicating that temporal relevance alone is insufficient and that grounding the retrieved behavior in the current scene provides additional benefit. Overall, these ablations support the design of PROACT: temporal relevance determines which demonstration evidence should be emphasized, while behavior grounding determines how that evidence should be instantiated in the current scene.

5.5 Cross-Embodiment Demonstration Conditioning

We further investigated whether PROACT could utilize human video demonstrations as an execution context, which requires the model to address the visual and embodied gaps between humans and robots. To this end, we construct $2,375$ paired human–robot demonstrations from two sources: collected data of human–robot pairs performing the same task and "human-hand" videos synthesized from robot videos using a video generation model (MiniMax, 2026). The human video serves as demonstration context, while the corresponding robot trajectory provides action supervision.

We evaluate closed-loop execution on 92 simulation cases, where both methods receive the same human video demonstration and initial scene, without language input. Figure 7 reports results on two held-out generalization subsets: *Position*, which evaluates unseen object locations and spatial layouts, and *Object*, which introduces held-out object instances and categories. PROACT achieves success rates of 60.0% and 55.3% on these subsets, respectively, compared with 23.3% and 23.7% for the baseline. The qualitative examples further illustrate this difference: UniSkill selects the wrong object or fails to retain the target after grasping, whereas PROACT successfully places the demonstrated target. These results show that PROACT can effectively exploit cross-embodiment video context to infer task-relevant behavior.

6 Conclusions and Limitations

In this paper, we introduce PROACT, an execution-conditioned framework for learning robot manipulation from a single video demonstration. Rather than treating the demonstration as static context, PROACT actively tracks execution progress and adapts both how the demonstration is represented and how the demonstrated behavior is grounded in the current scene. Across multiple manipulation benchmarks, PROACT consistently improves over prior demonstration-conditioned methods. Our results suggest that effective video-conditioned control depends not only on information contained in a demonstration, but also on continually reinterpreting that information as execution evolves.

One limitation of this work is that we do not explore large-scale pretraining. Scaling to larger and more diverse robot and video corpora may further strengthen the model's ability to interpret and adapt demonstrations in context. In addition, extending adaptive contextualization to multiple demonstrations remains an open direction. We leave both directions for future work.

AI USE STATEMENT

We used generative AI tools for drafting assistance and generating synthetic data. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed method, training and evaluation protocols, datasets, and experimental settings in the main paper and Appendix. Additional implementation details, hyperparameters, and experimental results are provided in the Appendix and supplementary materials. We will release the source code and necessary resources upon acceptance to facilitate reproduction of our results.

REFERENCES

Rachit Bansal, Aston Zhang, Rishabh Tiwari, Lovish Madaan, Sai Surya Duvvuri, Fnu Devvrit, David Brandfonbrener, David Alvarez-Melis, Prajjwal Bhargava, Mihir Kale, and Samy Jelassi. Let's (not) just put things in context: Test-time training for long-context LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=H0bcEdPCoc.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, brian ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 27–30 Sep 2025a. URL https://proceedings.mlr.press/v305/black25a.html.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of Robotics: Science and Systems*, 2025b. doi: 10.15607/RSS.2025.XXI.010. URL https://www.roboticsproceedings.org/rss21/p010.html.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Maria Elisabeth Bechtle, Feryal Behbahani, Stephanie C.Y. Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando De Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative Interactive Environments. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 4603–4623. PMLR, 2024. URL https://proceedings.mlr.press/v235/bruce24a.html.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, Hanbo Zhang, and Minzhao Zhu. GR-2: A Generative Video-Language-Action Model with Web-Scale Knowledge for Robot Manipulation. *arXiv preprint arXiv:2410.06158*, 2024. URL https://arxiv.org/abs/2410.06158.

Guangyan Chen, Meiling Wang, Te Cui, Zichen Zhou, Qi Shao, Shalfun Li, Hang Su, Roy Gan, Hao Wang, Mengyin Fu, Yi Yang, and Yufeng Yue. HOST: Robots Acquire Manipulation Skills in Seconds from a Single Human Video. *arXiv preprint arXiv:2607.20033*, 2026. URL https://arxiv.org/abs/2607.20033.

Yi Chen, Yuying Ge, Weiliang Tang, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent Motion Token as the Bridging Language for Learning Robot Manipulation from Videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19752–19763, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Chen_Moto_Latent_Motion_Token_as_the_Bridging_Language_for_Learning_ICCV_2025_paper.html.

Yinpei Dai, Hongze Fu, Jayjun Lee, Yuejiang Liu, Haoran Zhang, Jianing Yang, Chelsea Finn, Nima Fazeli, and Joyce Chai. Robomme: Benchmarking and understanding memory for robotic generalist policies. In *Forty-third International Conference on Machine Learning*, 2026.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, et al. A survey on in-context learning. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 1107–1128, 2024.

Yan Duan, Marcin Andrychowicz, Bradly C. Stadie, Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-Shot Imitation Learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/ba3866600c3540f67c1e9575e213be0a-Abstract.html.

Yusen Feng, Bingchen Han, Jiangran Lyu, Kai Liu, Yixin Zheng, Yuxuan Wan, Weiheng Liu, Sun Han, Ruiqin Li, Yulong Zhang, Fangfu Liu, Xuesong Shi, Libin Liu, Yizhou Wang, Zhizheng Zhang, and He Wang. WAM-TTT: Steering World-Action Models by Watching Human Play at Test Time. *arXiv preprint arXiv:2607.06988*, 2026. URL https://arxiv.org/abs/2607.06988.

Letian Fu, Huang Huang, Gaurav Datta, Lawrence Yunliang Chen, William Chung-Ho Panitch, Fangchen Liu, Hui Li, and Ken Goldberg. In-Context Imitation Learning via Next-Token Prediction. *arXiv preprint arXiv:2408.15980*, 2024. URL https://arxiv.org/abs/2408.15980.

Jingyang He, Guangrun Li, Jieyu Zhang, Chengkai Hou, Zhengping Che, and Shanghang Zhang. Demo-JEPA: Joint-Embedding Predictive Architecture for One-shot Cross-Embodiment Imitation. *arXiv preprint arXiv:2605.20811*, 2026. URL https://arxiv.org/abs/2605.20811.

Vidhi Jain, Maria Attarian, Nikhil J Joshi, Ayzaan Wahid, Danny Driess, Quan Vuong, Pannag R Sanketi, Pierre Sermanet, Stefan Welker, Christine Chan, Igor Gilitschenski, Yonatan Bisk, and Debidatta Dwibedi. Vid2Robot: End-to-end Video-conditioned Policy Learning with Cross-Attention Transformers. *arXiv preprint arXiv:2403.12943*, 2024. URL https://arxiv.org/abs/2403.12943.

Stephen James, Michael Bloesch, and Andrew J. Davison. Task-Embedded Control Networks for Few-Shot Imitation Learning. In *Proceedings of The 2nd Conference on Robot Learning*, volume 87, pp. 783–795. PMLR, 2018. URL https://proceedings.mlr.press/v87/james18a.html.

Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *conference on Robot Learning*, pp. 991–1002. PMLR, 2022.

Sanghwan Jang, Minjin Jeon, Minsoo Kim, Seong Jin Choi, Dongha Kim, and Hwanjo Yu. RA-VLA: Retrieval-augmented VLA for test-time adaptation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=ut6HebnnQe.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. VIMA: General Robot Manipulation with Multimodal Prompts. In *Fortieth International Conference on Machine Learning*, 2023. URL https://vimalabs.github.io/.

Yunfan Jiang, Yevgen Chebotar, Ruijie Zheng, Fengyuan Hu, Yunhao Ge, Jimmy Wu, Tianyuan Dai, Scott Reed, Li Fei-Fei, Yuke Zhu, and Linxi "Jim" Fan. RoboTTT: Context Scaling for Robot Policies. *arXiv preprint arXiv:2607.15275*, 2026. URL https://arxiv.org/abs/2607.15275.

Kento Kawaharazuka, Jihoon Oh, Jun Yamada, Ingmar Posner, and Yuke Zhu. Vision-language-action models for robotics: A review towards real-world applications. *IEEE Access*, 2025.

Hanjung Kim, Jaehyun Kang, Hyolim Kang, Meedeum Cho, Seon Joo Kim, and Youngwoon Lee. UniSkill: Imitating Human Videos via Cross-Embodiment Skill Representations. *arXiv preprint arXiv:2505.08787*, 2025a. URL https://arxiv.org/abs/2505.08787.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=ZMnD6QZAE6.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025b.

Mingxin Li, Yanzhao Zhang, Dingkun Long, Chen Keqin, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Qwen3-vl-embedding and qwen3-vl-reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv:2601.04720*, 2026.

Marius Memmel, Jacob Berg, Bingqing Chen, Abhishek Gupta, and Jonathan Francis. STRAP: Robot sub-trajectory retrieval for augmented policy learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4VHiptx7xe.

MiniMax. Minimax-h3: An open model breaking the boundaries between tasks and modalities. https://github.com/MiniMax-AI/MiniMax-H3, 2026.

Austin Patel, Ben Pekarek, Joel Enrique Castro Hernandez, and Shuran Song. Behavior prompting policy: Demonstrations as prompts for manipulation. *arXiv preprint arXiv:2606.30457*, 2026.

Dominik Schmidt and Minqi Jiang. Learning to Act without Actions. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/27985d21f0b751b933d675930aa25022-Abstract-Conference.html.

Rutav Shah, Shuijing Liu, Qi Wang, Zhenyu Jiang, Sateesh Kumar, Mingyo Seo, Roberto Martín-Martín, and Yuke Zhu. MimicDroid: In-Context Learning for Humanoid Robot Manipulation from Human Play Videos. *arXiv preprint arXiv:2509.09769*, 2025. URL https://arxiv.org/abs/2509.09769.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Kaustubh Sridhar, Souradeep Dutta, Dinesh Jayaraman, and Insup Lee. RICL: Adding In-Context Adaptability to Pre-Trained Vision-Language-Action Models. *arXiv preprint arXiv:2508.02062*, 2025. URL https://arxiv.org/abs/2508.02062.

Vitalis Vosylius and Edward Johns. Instant Policy: In-Context Imitation Learning via Graph Diffusion. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/5692c7dbc4abcaa50f9ce609819212e5-Paper-Conference.pdf.

Quan Vuong, Sergey Levine, Homer Rich Walke, Karl Pertsch, Anikait Singh, Ria Doshi, Charles Xu, Jianlan Luo, Liam Tan, Dhruv Shah, et al. Open x-embodiment: Robotic learning datasets and rt-x models. In *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@ CoRL2023*, 2023.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing Large-Scale Video Generative Pre-training for Visual Robot Manipulation. *arXiv preprint arXiv:2312.13139*, 2023. URL https://arxiv.org/abs/2312.13139.

Mengda Xu, Zhenjia Xu, Cheng Chi, Manuela Veloso, and Shuran Song. XSkill: Cross Embodiment Skill Discovery. In *Proceedings of The 7th Conference on Robot Learning*, volume 229, pp. 3536–3555. PMLR, 2023. URL https://proceedings.mlr.press/v229/xu23a.html.

Jiarui Yang, Wen Huang, Jiale Zhang, Maowei Hu, and Hang Guo. In-context vla: Endowing vision-language-action models with language via in-context post-training and agentic tool use. *arXiv preprint arXiv:2608.05738*, 2026.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent Action Pretraining from Videos. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=NCOP0KYb0u.

Haoqi Yuan, Zhixuan Liang, Anzhe Chen, Ye Wang, Haoyang Li, Pei Lin, Yiyang Huang, Zixing Lei, Tong Zhang, Jiazhao Zhang, et al. Qwen-robotmanip technical report: Alignment unlocks scale for robotic manipulation foundation models. *arXiv preprint arXiv:2606.17846*, 2026.

Jiaming Zhou, Qihang Zhang, Gangwei Xu, Cunxin Fan, Yujie Zhao, Ruilin Wang, Yiming Luo, Shuai Yang, Xing Zhu, Yujun Shen, Junwei Liang, and Yinghao Xu. Zero-WAM: In-Context World-Action Modeling from Human Videos for Open-Ended Task Generalization. *arXiv preprint arXiv:2608.26103*, 2026. URL https://arxiv.org/abs/2608.26103.

Xiangming Zhu, Huayu Deng, Haoran Zhao, Yiwei Hao, and Yunbo Wang. Learning transferable interaction primitives from game videos for humanoid locomotion. In *Forty-third International Conference on Machine Learning*, 2026.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *Proceedings of The 7th Conference on Robot Learning*, volume 229, pp. 2165–2183. PMLR, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

A Method Implementation Details

A.1 Temporal Summaries and Progress Supervision

We partition the normalized demonstration timeline into K overlapping temporal windows, where K controls the temporal resolution of the progress representation and is not fixed by the formulation. For zero-based window index k, a demonstration frame at normalized time τ_i is assigned to window $\mathcal{W}_k$ if

$$\max\left(0, \frac{k-0.5}{K}\right) \leq \tau_i \leq \min\left(1, \frac{k+1.5}{K}\right). \tag{15}$$

Adjacent windows therefore overlap by half of their temporal extent, providing smooth coverage across window boundaries. Each temporal summary is obtained by mean-pooling the original visual embeddings within its support, without introducing additional learned summary queries.

For progress estimation, we use a projection dimension of $d_p = 256$. Given the DTW-derived progress target $\tau_t \in [0, 1]$, we place K ordered anchors at

$$\xi_k = \frac{k}{K-1}, \qquad k = 0, \ldots, K-1, \tag{16}$$

with spacing $\Delta = 1/(K-1)$, and construct the soft target

$$q_t^*(k) = \frac{\max\left(0, 1 - \frac{|\tau_t - \xi_k|}{\Delta}\right)}{\sum_{r=0}^{K-1} \max\left(0, 1 - \frac{|\tau_t - \xi_r|}{\Delta}\right)}. \tag{17}$$

Unless otherwise specified, we use $K = 6$ in our experiments.

A.2 Token Allocation and Spatial Pooling

Given a target retention ratio ρ, we allocate a fraction of the original demonstration visual tokens across temporal frames according to their estimated relevance. For T frame with C visual tokens each, we first reserve a minimum of m tokens per frame and distribute the remaining continuous budget in proportion to temporal relevance. Allocations are capped at the capacity of each frame, with excess budget redistributed among frames that have remaining capacity. The resulting allocations are rounded to integer token counts.

Within each temporal frame, spatial compression operates on the original two-dimensional patch grid. We recursively split the largest rectangular region along its longer spatial dimension until the desired number of regions is obtained. Each region is represented by its mean-pooled feature, with a central original patch used as the positional anchor. The same spatial partition is applied to the corresponding intermediate visual features of the backbone.

Unless otherwise specified, we use $\rho = 0.25$ and $m = 4$ in our experiments.

A.3 Context Packing and Attention Structure

PROACT uses two VLA passes with different context layouts. In the first pass, temporal summaries, instruction tokens, and current visual tokens are jointly processed for progress estimation. Each temporal summary attends only to itself; instruction tokens attend to the summaries and their causal instruction prefix; current visual tokens attend to the summaries, instruction tokens, and their causal current prefix.

The second pass processes the routed demonstration together with the current observation, operator queries, instruction, and action queries, in that order. Its structural attention mask is summarized below:

Query Block	Visible Key Blocks
Demonstration	Causal demonstration prefix
Current	Demonstration; causal current prefix
Operator	Demonstration; current; causal operator prefix
Instruction	Demonstration; current; causal instruction prefix
Action	Demonstration; current; operator; instruction; causal action prefix

Operator queries do not directly attend to instruction tokens, encouraging the latent behavior representation to be inferred from visual context rather than language alone. The temporal routing bias is applied to all permitted query-to-demonstration visual-key connections and only reweights existing connections; it does not introduce attention paths forbidden by the structural mask.

A.4 Predictive Grounding and Action Generation

We use eight operator queries and a two-layer spatial predictor with eight attention heads per layer. Each layer consists of spatial self-attention, cross-attention to the projected operator states, and an MLP with residual connections. A final projection predicts a residual update to the current visual features. We use a frozen DINOv3 encoder to extract 300 patch features per view for both the current and future observations, with gradients stopped through the encoder.

The action head is an MLP-based flow model conditioned on the action-query hidden states and robot state. We use 16 action queries and predict a 50-step action chunk. The predictive grounding target is taken from a future observation at the corresponding prediction horizon. We set $\lambda_{\text{prog}} = \lambda_{\text{grd}} = 1$. The predictive grounding module is used only for auxiliary supervision during training and is not required for action generation at inference time.

Algorithm 1 summarizes the routing and training procedure.

B Experimental details

B.1 Benchmark and Evaluation Protocols

LIBERO-Gen. We evaluate on the *Combination* and *Chain* suites of LIBERO-Gen (Patel et al., 2026), which test compositional generalization over previously seen manipulation primitives. *Combination* extends the LIBERO environment with additional pick-and-place configurations involving two identical bowls and multiple target locations. The training set contains 164 tasks, while 10 pick–place combinations are held out for evaluation. For each held-out task, the corresponding pick location and place location have both appeared individually during training, but never jointly, requiring the policy to recombine familiar spatial factors at test time. *Chain* consists of two-step manipulation sequences that combine pick-and-place with primitives such as opening drawers, pushing a plate, and turning on the stove. The benchmark contains 21 first-step tasks, 146 second-step tasks, and 144 seen two-step chains, with 10 additional chains held out for evaluation. In our experiments, the reference demonstration is provided only as video context; we do not use demonstrator actions or proprioceptive states from the reference trajectory.

IsaacSim-Gen. We construct *IsaacSim-Gen* in NVIDIA Isaac Sim to evaluate demonstration-conditioned manipulation under spatial variation, temporal ordering, and multi-stage execution. The benchmark contains four task families: *Spatial Pick-and-Place*, *Stack Block*, *Sequential Table Clearing*, and *Cabinet Manipulation*. *Spatial Pick-and-Place* requires the robot to identify a designated object from multiple candidates and move it to a specified destination, testing spatial grounding under changes in object and target configurations. *Stack Block* requires arranging multiple blocks according to a specified order and spatial relation. *Sequential Table Clearing* requires collecting designated objects into target containers in a prescribed sequence. *Cabinet Manipulation* combines cabinet-door interaction with subsequent object placement, resulting in a longer-horizon manipulation sequence. The benchmark contains 80 training tasks and 33 held-out evaluation tasks across the four task families. For each task family, we collect 100 task-specific robot trajectories and train jointly with an additional $3,018$ general-purpose trajectories, including $2,442$ real-robot trajectories and 576 simulation trajectories. The held-out evaluation tasks introduce unseen combinations of

Algorithm 1 Training and deployment of PROACT

Require: Training tuple $(\mathcal{D}, o_t, \ell, r_t, A_t, o_{t+H}, \tau_t)$, where τ_t is the DTW-derived progress target
Ensure: Updated parameters θ, ψ

1: Encode demonstration frames and the current observation: $\{V_j\}_{j=1}^{L}, V_t^{\text{obs}} \leftarrow \text{Encode}(\mathcal{D}, o_t)$
2: Construct overlapping windows $\{\mathcal{W}_k\}_{k=1}^{K}$ and compute $\bar{v}_k \leftarrow \text{MeanPool}\Big(\bigcup_{j\in\mathcal{W}_k} V_j\Big)$
3: Obtain contextualized window and observation representations:

$$\{h_k^{\mathcal{D}}\}_{k=1}^{K}, h_t \leftarrow \text{VLA}_\theta\left(\{\bar{v}_k\}_{k=1}^{K}, V_t^{\text{obs}}, \ell\right)$$

4: Compute temporal relevance q_t using Eq. 2
5: Project relevance onto frames and allocate token budgets:

$$p_t \leftarrow \text{ProjectToFrames}(q_t, \{\mathcal{W}_k\}_{k=1}^{K}), \qquad \{b_t(j)\}_{j=1}^{L} \leftarrow \text{Allocate}(p_t, B, m)$$

6: Pool demonstration visual features according to the allocated budgets:

$$\widetilde{V}_{\mathcal{D},t}, \{n_a, j(a)\}_a \leftarrow \text{Pool}\Big(\{V_j\}_{j=1}^{L}, \{b_t(j)\}_{j=1}^{L}\Big)$$

7: For each retained demonstration token a, construct its attention bias:

$$\Delta_t(a) \leftarrow \log\big(Lp_t(j(a))\big)$$

8: Apply $\Delta_t(a)$ to every permitted query-to-a attention logit
9: Obtain behavior and action representations:

$$\mathbf{u}_t^{\text{op}}, \mathbf{u}_t^{\text{act}} \leftarrow \text{VLA}_\theta\left([\widetilde{V}_{\mathcal{D},t}; V_t^{\text{obs}}; Q^{\text{op}}; \ell; E^{\text{act}}]; \Delta_t\right)$$

10: Predict the action chunk: $\widehat{A}_t \leftarrow \pi_\theta(\mathbf{u}_t^{\text{act}}, r_t)$
11: Construct the progress target $q_t^\star \leftarrow \text{TriangularTarget}(\tau_t, K)$ and compute $\mathcal{L}_{\text{prog}} \leftarrow \text{W1}(q_t, q_t^\star)$
12: Extract frozen visual features and predict the future outcome:

$$z_t, z_{t+H} \leftarrow \text{sg}[\phi(o_t)], \text{sg}[\phi(o_{t+H})], \qquad \widehat{z}_{t+H} \leftarrow z_t + f_\psi(z_t, \mathbf{u}_t^{\text{op}})$$

13: Compute $\mathcal{L}_{\text{act}}$, $\mathcal{L}_{\text{grd}}$ from Eq. 13, and

$$\mathcal{L} \leftarrow \mathcal{L}_{\text{act}} + \lambda_{\text{prog}}\mathcal{L}_{\text{prog}} + \lambda_{\text{grd}}\mathcal{L}_{\text{grd}}$$

14: Update θ, ψ using $\nabla_{\theta,\psi}\mathcal{L}$

Deployment: Given $(\mathcal{D}, o_t, \ell, r_t)$, execute Lines 1–9 and return $\widehat{A}_t$. Future observations and grounding losses are not required.

object selections, placement targets, and action sequences, while preserving the underlying manipulation primitives observed during training.

RoboMME. We evaluate on *PatternLock* and *RouteStick* from RoboMME (Dai et al., 2026), two tasks that require the robot to follow a demonstrated spatial sequence and interact with target locations in the prescribed order. Their explicitly ordered structure makes them suitable for evaluating whether a policy can maintain correspondence between the current execution state and a temporally structured demonstration. We follow the RoboMME evaluation protocol and report success rates under the *Easy*, *Medium*, and *Hard* settings, together with the overall score. *Easy* uses paths of length 2–3 without backtracking, *Medium* uses paths of length 4–5 without backtracking, and *Hard* uses paths of length 4–7 with backtracking.

B.2 Cross-Embodiment Data Construction and Evaluation

Data construction. We construct a paired human–robot demonstration dataset containing $2,375$ episodes, including 674 collected human–robot pairs and $1,701$ additional pairs obtained by synthesizing human-hand videos from robot videos using MiniMax-H3. The synthesized human videos are paired with their source robot trajectories at the episode level. The human video is used as demonstration context, while action supervision is taken from the corresponding robot trajectory;

no frame-level alignment between human motion and robot actions is required. We use $2,247$ pairs for training and 128 for validation.

Human-video synthesis prompt. For synthesized demonstrations, MiniMax-H3 is prompted to replace the acting robot embodiment with a human hand while preserving the task semantics, scene configuration, event order, and manipulation outcome. The prompt used in our generation pipeline is shown below:

Robot-to-Human Video Translation Prompt

Transform Video into a realistic human-hand manipulation video of the same task.

Task instruction: *[source task instruction]*

Replace the acting robot arm and gripper with one anatomically natural *[LEFT/RIGHT]* human hand and forearm. Remove inactive robot hardware and do not introduce additional hands or limbs. Preserve the objects, table, containers, and background. Preserve the role of the corresponding source arm, including its approach, manipulated object, and interaction order.

The provided initial and final reference images specify the endpoint states of the source clip. Preserve object identities, positions, orientations, container states, and the final manipulation outcome while replacing robot hardware with the corresponding human anatomy.

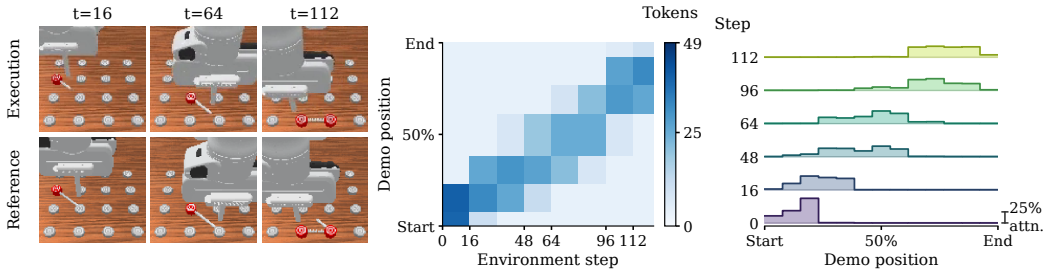
Preserve the manipulation sequence, object trajectory, contacted object part, grasp function, and final outcome shown in the source video. Adapt mechanical gripper interactions into physically plausible human grasps at the same functional contact region.

Natural differences in motion speed and intermediate contact timing are allowed; exact frame-by-frame synchronization is not required. Preserve event order. Do not introduce unrelated actions, extra objects, extra hands, scene cuts, camera motion, captions, or labels.

Evaluation protocol. We evaluate closed-loop execution on 92 fixed simulation cases using human demonstrations. No language instruction is provided at test time. Each model receives the same human video demonstration and the same initial scene, so the comparison isolates how the demonstration is represented and used for control. The evaluation set is divided into two subsets. *Position* varies object locations and spatial layouts while preserving the task identity, whereas *Object* changes object instances and categories.

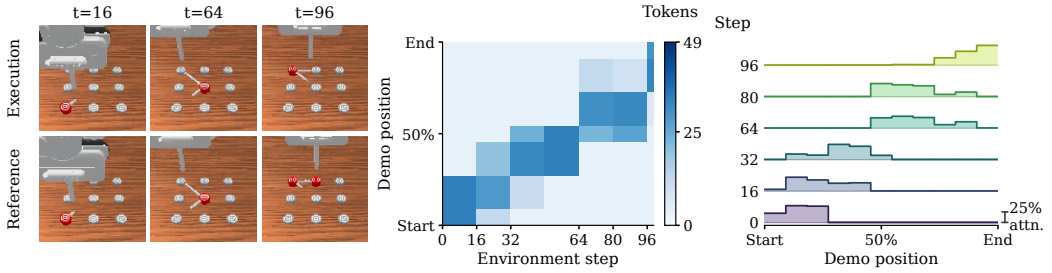
C Additional Analysis of Adaptive Contextualization

We provide additional visualizations complementing Figure 5 in the main paper. Figures 8 and 9 show two additional rollouts from *PatternLock*, while Figures 10 and 11 show two examples from *RouteStick*. Across these examples, the execution–reference correspondence, token allocation, and policy attention exhibit the same progressive shift along the demonstration timeline as execution unfolds. These additional examples further support that PROACT consistently adapts how a fixed demonstration is represented and used throughout execution.



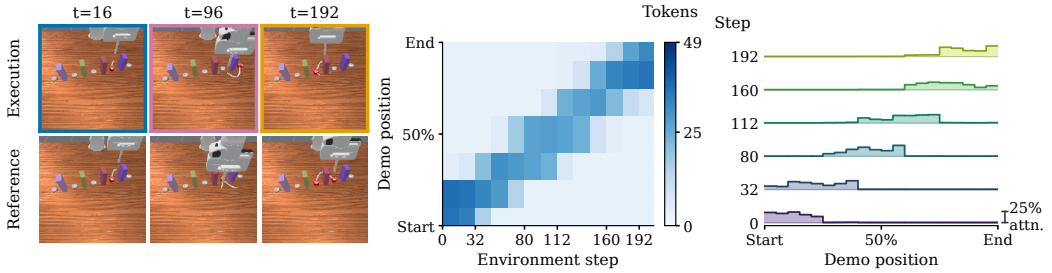
(a) Execution–Reference Alignment (b) Adaptive Token Allocation (c) Progress-Aware Attention

Figure 8: **Additional visualization on *PatternLock*.** Execution–reference correspondence, adaptive token allocation, and policy attention throughout execution.



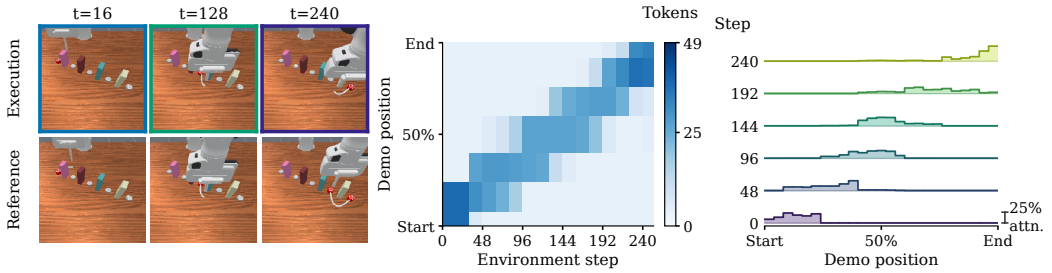
(a) Execution–Reference Alignment (b) Adaptive Token Allocation (c) Progress-Aware Attention

Figure 9: **Additional visualization on *PatternLock*.** Execution–reference correspondence, adaptive token allocation, and policy attention throughout execution.



(a) Execution–Reference Alignment (b) Adaptive Token Allocation (c) Progress-Aware Attention

Figure 10: **Additional visualization on *RouteStick*.** Execution–reference correspondence, adaptive token allocation, and policy attention throughout execution.



(a) Execution–Reference Alignment (b) Adaptive Token Allocation (c) Progress-Aware Attention

Figure 11: **Additional visualization on *RouteStick*.** Execution–reference correspondence, adaptive token allocation, and policy attention throughout execution.