SeeTraceAct: Visibility-Aware Latent Planning from Cross-Embodiment Demonstration Videos

Jaehyeon Son [1] **Junhyun Kim** [1] **Kyle Kam** [1] **Jinhoo Kim** [1] **Seok Joon Kim** [1] **Junghwan Yim** [1] **Zsolt Kira** [1]

Abstract

Vision-language-action models (VLAs) are promising general-purpose robot policies, but adapting them to new tasks typically requires costly task-specific teleoperation data. As an alternative, we study one-shot demo-conditioned VLAs, where a robot policy is conditioned on a single demonstration video of an unseen task. We find that existing end-to-end approaches often struggle when successful execution requires precisely localizing small target regions. To address this limitation, we propose SEETRACEACT, a demo-conditioned VLA framework that encourages precise spatial grounding through visibility-aware prediction of future end-effector traces. To enable reproducible evaluation with cross-embodiment demonstrations, we introduce and release ROBOCASA-DC, a demo-conditioned extension of RoboCasa with episode-paired humanoid videos. Experiments on ROBOCASA-DC and a real-world benchmark, where a Franka Panda arm is conditioned on human demonstrations, show that SEETRACEACT outperforms baselines, achieving the best success rate across all four ROBOCASA-DC settings and improving real-world average success by 12.5 percentage points.

1. Introduction

Vision-language-action models (VLAs) have recently emerged as a promising path toward general-purpose robotic agents by adapting pretrained vision-language models (VLMs) for robotic control (Black et al., 2024; Kim et al., 2024; NVIDIA et al., 2025). However, deploying VLAs on new tasks typically requires post-training on expert robot trajectories collected through teleoperation. Because collecting teleoperated robot data requires specialized hardware and substantial human effort, it remains a central bottleneck for scaling VLAs across diverse tasks, embodiments, and environments (Walke et al., 2023; Khazatsky et al., 2024; O'Neill et al., 2024).

A scalable alternative is to let an end user specify a new robot task by performing it once in front of a camera. This motivates one-shot demo-conditioned VLAs, where a policy is conditioned on a single human demonstration video of an unseen target task. However, this problem is non-trivial: the cross-embodiment gap makes directly translating human demonstrations into robot actions fundamentally challenging.

While methods that utilize explicit embodiment-specific retargeting have been developed (Park et al., 2026; Li et al., 2024; Heppert et al., 2024), they require embodiment- and task-specific expertise beyond what an end user can provide, so we instead focus on end-to-end approaches. Prior work in this direction has learned video representations and employed auxiliary objectives to ground demonstrations (Jain et al., 2024; Kim et al., 2025; Chen et al., 2025). While these methods have shown promise, we find that they often struggle with precision-sensitive tasks, such as pressing the target button on a coffee machine or turning on a sink faucet (see §5.3).

To address this limitation, we propose SEETRACEACT, a demo-conditioned VLA framework that encourages precise spatial grounding through an auxiliary future-trace prediction objective (Fig. 1). Given a demonstration video, SEETRACEACT learns a *visual latent plan* that summarizes the task-relevant motion the robot should execute. During training, this latent plan is decoded into future end-effector traces in the robot's camera views. However, trace supervision is not always well-defined, especially in multi-view setups: the end effector may leave some camera views, making its coordinates ill-posed targets. SEETRACEACT therefore uses a visibility-aware trace decoder that predicts both trace coordinates and their validity, preserving supervision under partial visibility. At inference time, the trace decoder is discarded, and the policy generates actions directly from the latent plan.

To support reproducible evaluation and future work on demo-conditioned VLAs, we introduce and release the ROBOCASA-DC benchmark, a demo-conditioned exten-

[1]Georgia Institute of Technology, Atlanta, GA, USA. Correspondence to: Zsolt Kira <zkira@gatech.edu>.

Published as a workshop paper at SCALE - ICML 2026.

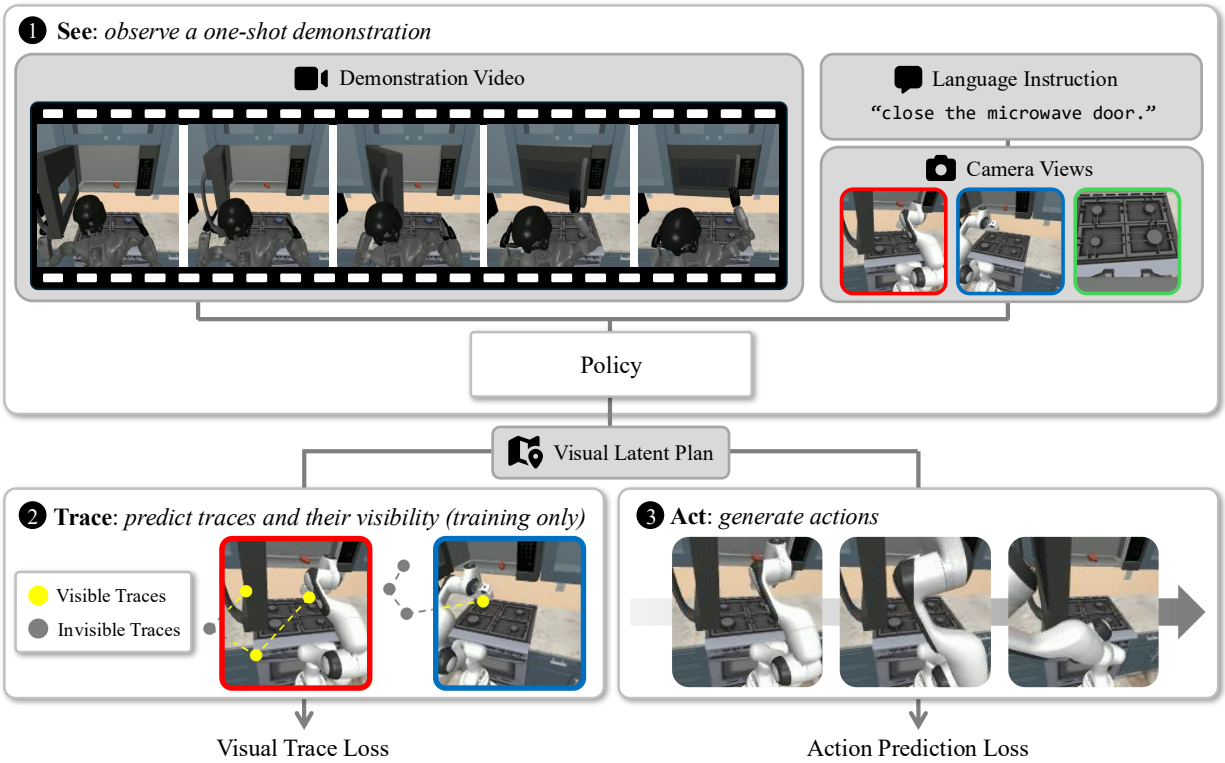

Figure 1. Overview of SEETRACEACT. Given a demonstration video, current camera views, and a language instruction, the policy encodes task-relevant information into a *visual latent plan* (SEE). During training, the policy predicts future visual traces and their visibility for each camera view (TRACE), while also predicting actions from the latent plan (ACT). At inference time, the trace prediction component is discarded, and the policy generates actions from the latent plan (ACT).

sion of the RoboCasa (Nasiriany et al., 2024) simulation environment (§4). ROBOCASA-DC supports both same-embodiment and cross-embodiment evaluation, allowing policies to be conditioned on either Panda-arm demonstrations or episode-paired humanoid videos as a proxy for human demonstrations. This complements prior evaluations that focus on real-world settings (Jain et al., 2024), use single-arm robot demonstrations in simulation (Chen et al., 2025), or do not release the human demonstrations used for simulated evaluation (Kim et al., 2025).

Experiments on ROBOCASA-DC and a real-world benchmark with a Franka Panda arm show that SEETRACEACT achieves the strongest results among competitive demo-conditioned VLA baselines (§5.2). On ROBOCASA-DC, SEETRACEACT achieves the best success rate across all four evaluation settings. On the real-world benchmark, where the robot is conditioned on human demonstrations, SEETRACEACT improves the average success rate by 12.5 percentage points. Our ablation study (§5.4) supports the importance of our key design choices, including visibility-aware trace supervision. Our main contributions are as follows:

1. We introduce SEETRACEACT, a demo-conditioned VLA framework that encourages precise spatial grounding of one-shot demonstration videos, including cross-embodiment ones, via visibility-aware prediction of future end-effector traces.
2. We introduce and release ROBOCASA-DC, a demo-conditioned extension of RoboCasa for benchmarking policies with both same- and cross-embodiment demonstration videos.
3. We show that SEETRACEACT achieves the strongest results among competitive baselines on ROBOCASA-DC and a real-world benchmark, with ablations supporting our key design choices.

2. Related Work

Vision-language-action models (VLAs). Early large-scale VLAs (Brohan et al., 2023b;a) introduce transformer-based policies trained on extensive robot trajectories paired with web-scale vision-language data. OpenVLA (Kim et al., 2024) open-sources a VLA pretrained on the Open X-Embodiment dataset (O'Neill et al., 2024). More recently, π_0 (Black et al., 2024) and GR00T N1 (NVIDIA et al., 2025) combine a pretrained vision-language backbone with a diffusion-based action expert for continuous, high-frequency control. However, these VLAs still require substantial task- and embodiment-specific teleoperation data for post-training, motivating our focus on one-shot demon-

stration videos.

One-shot demo-conditioned policies. One line of work uses kinematic retargeting between humans and robots to learn from one-shot demonstrations (Park et al., 2026; Li et al., 2024; Heppert et al., 2024). These approaches enable explicit cross-embodiment transfer from human demonstrations to robot actions, but often rely on substantial task- or embodiment-specific engineering. We instead focus on end-to-end approaches that condition robot policies directly on demonstrations without explicit retargeting. XSkill (Xu et al., 2023) and UniSkill (Kim et al., 2025) learn cross-embodiment skill representations from human and robot videos, while Vid2Robot (Jain et al., 2024) trains an end-to-end demo-conditioned policy with video- and language-alignment objectives. ViVLA (Chen et al., 2025) learns a unified latent action space across human and robot demonstrations and trains a VLA to predict both latent action tokens and robot actions. While these methods have shown promise, we find that they can still struggle when task success depends on precisely localizing small interaction regions (§5.3). SEETRACEACT addresses this limitation through an auxiliary visual trace objective that encourages precise spatial grounding.

Visual traces in VLAs. Several works have incorporated visual traces into VLAs: LLARVA (Niu et al., 2024) and MolmoAct (Lee et al., 2025) predict future visual traces and feed them autoregressively to the action model, while RT-Trajectory (Gu et al., 2024) uses trajectory sketches obtained from human drawings, videos, or model predictions. HAMSTER (Li et al., 2025) uses 2D traces as an intermediate representation in hierarchical policies, and TraceVLA (Zheng et al., 2025) conditions the policy on past traces extracted from previous observations. Closer to our approach, ThinkAct (Huang et al., 2025) and FastThinkAct (Huang et al., 2026) condition action generation on visual latent plans rather than decoded visual traces. SEETRACEACT differs by using visual traces to ground demonstration videos into spatial plans while explicitly handling partial visibility.

3. Method

3.1. Problem setting

We consider a setting in which a policy is evaluated on unseen tasks, conditioned on a single demonstration video that may come from a different embodiment. Let $\mathcal{T}$ denote a distribution over robotic tasks, and let $\mathcal{T}_{\text{seen}}$ and $\mathcal{T}_{\text{unseen}}$ denote disjoint sets of seen and held-out unseen tasks sampled from this distribution. During training, for each seen task $\kappa \in \mathcal{T}_{\text{seen}}$, the policy π_θ has access to multiple training tuples $(\xi^\kappa, D^\kappa, l^\kappa)$, where $\xi^\kappa = \{(o_t, q_t, a_t)\}_{t=1}^{T}$ is an expert trajectory from the policy embodiment, D^κ is a task-matched demonstration video that may come from a different embodiment, and l^κ is the corresponding language instruction. Here, o_t, q_t, and a_t denote the camera view, robot state, and robot action at timestep t, respectively. We omit the task superscript for brevity.

At evaluation, the policy is given a single demonstration video D and language instruction l for an unseen task $\kappa^* \in \mathcal{T}_{\text{unseen}}$. At each timestep t, it takes the current observation (o_t, q_t) and predicts an action chunk $A_t = \{a_t, \ldots, a_{t+H-1}\}$ toward completing κ^*, where H denotes the action horizon: $A_t \sim \pi_\theta(\cdot \mid o_t, q_t, l, D)$.

3.2. Architecture

We build SEETRACEACT on top of the GR00T N1.5 (NVIDIA et al., 2025) architecture, which consists of a vision-language model (VLM) and a flow-matching-based action expert. As shown in Fig. 2, our model augments this architecture with video encoding modules, learnable query tokens that form a visual latent plan, and a trace decoder. At each control timestep t, the VLM processes image and language tokens from the current camera views and language instruction, producing a representation ϕ_t.

Video tokens. The left side of Fig. 2 shows how we incorporate the demonstration video. We first encode the demonstration D using a video encoder (Assran et al., 2025) pretrained on large-scale video data, including action-centric datasets (Goyal et al., 2017; Kay et al., 2017). We add a Perceiver Resampler (Alayrac et al., 2022; Jaegle et al., 2021) on top of the video encoder to compress the extracted features into a compact set of video tokens (from 8,192 to 32 tokens; see App. B), and append them to the image and language tokens in the VLM input sequence. Under causal attention, this ordering lets the video tokens attend to the image and language tokens, making the demonstration representation context-dependent rather than fixed across the episode. We denote the final VLM hidden states corresponding to these video tokens by ψ_t.

Visual latent plan. The middle of Fig. 2 illustrates how we form the visual latent plan. Following recent visual latent planning work (Huang et al., 2025; 2026), we use this term to refer to a latent representation that is trained to encode future task progression and conditions action generation. Rather than assigning this planning role directly to the existing input tokens, we append a separate set of learnable query tokens after them. Under causal attention, these query tokens attend to the image, language, and video tokens, and their final hidden states constitute the visual latent plan z_t.

Visibility-aware trace decoder. As shown on the right side of Fig. 2, the trace decoder predicts future end-effector traces in each static camera view from the visual latent plan

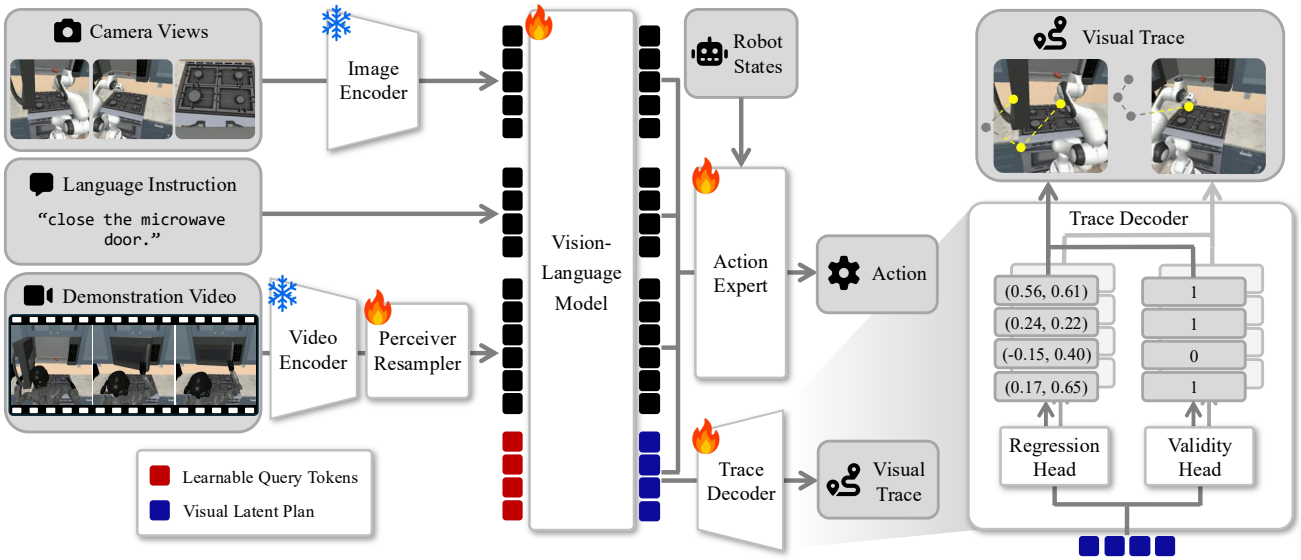

Figure 2. Architecture of SEETRACEACT. It receives camera views, a language instruction, a demonstration video, and robot states, and outputs an action chunk. We append learnable query tokens after the input tokens; their final hidden states form a *visual latent plan*, which is decoded into future end-effector traces during training. The trace decoder consists of a regression head that predicts the trace coordinates and a validity head that predicts whether each trace point lies within a camera view. The trace decoder is used only during training and discarded at inference time.

z_t. However, the end effector may leave some camera views, making its normalized 2D image coordinates ill-posed regression targets. To address this, we use a *visibility-aware* trace decoder with two heads. The regression head predicts normalized end-effector coordinates, while the validity head predicts whether or not each trace point lies within the image, allowing off-screen points to still provide a learning signal (see ablation in §5.4). The trace decoder is discarded after training.

3.3. Training and inference

We train SEETRACEACT with two objectives: the *action prediction loss* and the auxiliary *visual trace loss*. The former optimizes the policy to predict robot action chunks, while the latter shapes the visual latent plan z_t to encode future task progression in the robot's image space.

Action prediction loss. Following prior works (Black et al., 2024; NVIDIA et al., 2025), we train the action expert with an action prediction loss instantiated as a flow-matching objective (Lipman et al., 2023). At each control timestep t, we sample noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and draw a flow-matching timestep $\tau \in [0, 1]$ from a shifted beta distribution. Given the ground-truth action chunk A_t, we construct the noised action chunk as $A_t^\tau = \tau A_t + (1-\tau)\epsilon$. The action prediction loss trains the action expert V_θ to predict the target velocity field from the VLM outputs ϕ_t, ψ_t, and z_t, together with q_t and the noised action chunk A_t^τ:

$$\mathcal{L}_{\text{act}}(\theta) = \mathbb{E}_{\tau,\epsilon}\left[\left\|V_\theta(\phi_t, \psi_t, z_t, A_t^\tau, q_t) - (A_t - \epsilon)\right\|^2\right]. \tag{1}$$

Visual trace loss. We supervise the policy with visibility-aware 2D image-space traces, directly aligning the target with the robot's visual observation space. For each camera view, we predict N future trace points at a temporal stride Δ using N learnable query tokens. We denote the final hidden states of these tokens by $z_t = \{z_{t,1}, \ldots, z_{t,N}\}$. The corresponding trace target is $Y_t = \{y_{t+\Delta}, y_{t+2\Delta}, \ldots, y_{t+N\Delta}\}$, where each $z_{t,n}$ is supervised to predict $y_{t+n\Delta}$, the relative coordinate of the end effector. We also define the binary validity target $M_t = \{m_{t+\Delta}, m_{t+2\Delta}, \ldots, m_{t+N\Delta}\}$, where $m_{t+n\Delta}$ indicates whether the end effector lies within the camera view. The trace decoder takes z_t as input and applies two heads. The regression head outputs the predicted trace $\hat{Y}_t = \{\hat{y}_{t+\Delta}, \ldots, \hat{y}_{t+N\Delta}\}$, while the validity head outputs the predicted validity scores $\hat{M}_t = \{\hat{m}_{t+\Delta}, \ldots, \hat{m}_{t+N\Delta}\}$. We define the masked regression loss as:

$$\mathcal{L}_{\text{reg}}(\theta) = \frac{1}{\max(1, \sum_{n=1}^{N} m_{t+n\Delta})} \sum_{n=1}^{N} m_{t+n\Delta} \left\|\hat{y}_{t+n\Delta} - y_{t+n\Delta}\right\|_1. \tag{2}$$

We define the validity loss with binary cross-entropy:

$$\mathcal{L}_{\text{valid}}(\theta) = \frac{1}{N} \sum_{n=1}^{N} \text{BCE}\left(\hat{m}_{t+n\Delta}, m_{t+n\Delta}\right). \tag{3}$$

The auxiliary visual trace loss combines the regression and validity losses:

$$\mathcal{L}_{\text{trace}}(\theta) = \mathcal{L}_{\text{reg}}(\theta) + \lambda_{\text{valid}}\mathcal{L}_{\text{valid}}(\theta), \tag{4}$$

where λ_{valid} controls the relative weight of the validity loss. This loss is applied independently to each camera view and averaged across views.

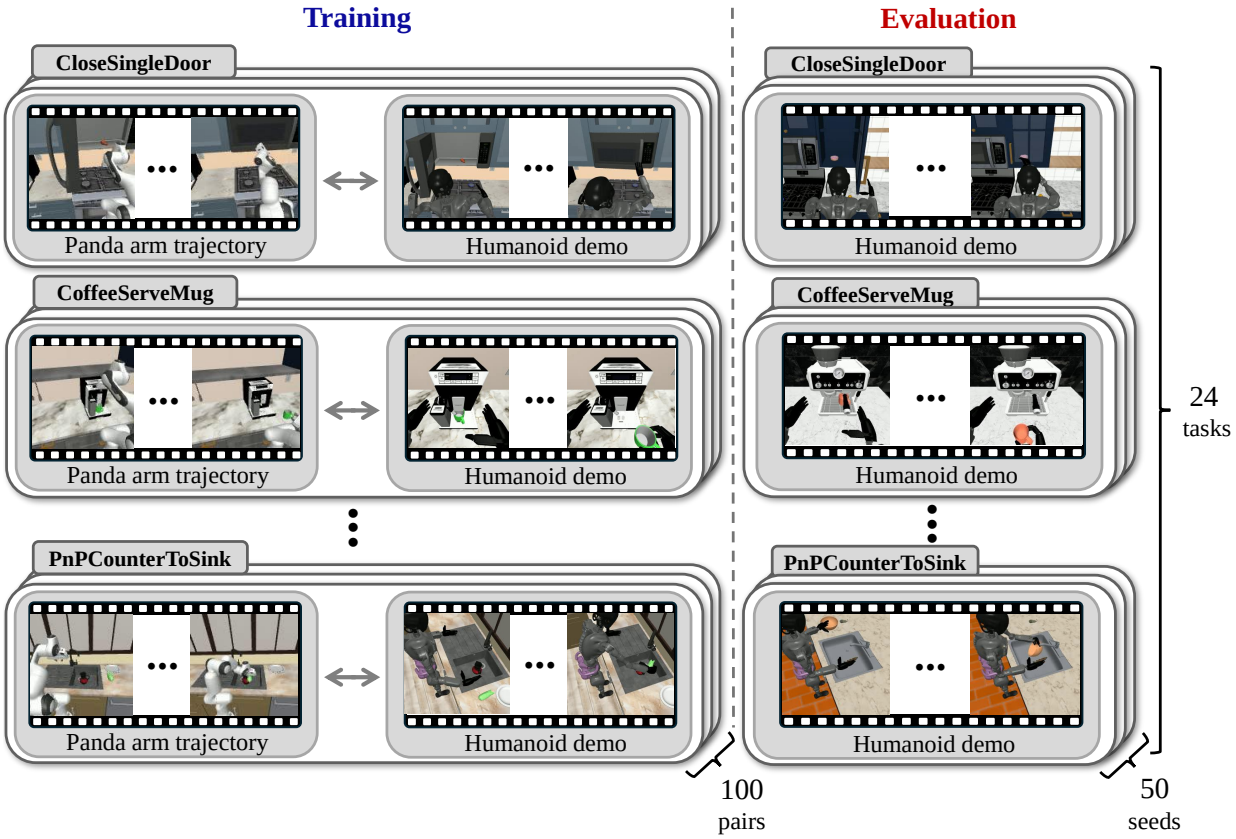

Figure 3. Cross-embodiment benchmark dataset in RoboCasa-DC. For each of the 24 tasks, we pair 100 original Panda-arm trajectories with collected GR-1 humanoid demonstrations for training. For evaluation, we collect humanoid demonstrations for 50 pre-defined seeds per task.

Overall training objective. Our final training objective combines the action prediction loss and the visual trace loss:

$$\mathcal{L}_{\text{total}}(\theta) = \mathcal{L}_{\text{act}}(\theta) + \lambda_{\text{trace}}\mathcal{L}_{\text{trace}}(\theta), \tag{5}$$

where λ_{trace} controls the strength of the visual trace supervision.

Inference. During inference, the VLM backbone processes the camera views, language instruction, and demonstration video to obtain the representations ϕ_t, ψ_t, and z_t at control timestep t. The action expert predicts actions from ϕ_t, ψ_t, z_t, q_t, and the current noisy action estimate. We initialize $A_t^0 = \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and update it with Euler integration: $A_t^{(k+1)/K} = A_t^{k/K} + \frac{1}{K}V_\theta(\phi_t, \psi_t, z_t, A_t^{k/K}, q_t)$ for $k = 0, \ldots, K-1$. After K denoising steps, we obtain A_t^1 as the final predicted action chunk.

4. RoboCasa-DC

Base environment. RoboCasa (Nasiriany et al., 2024) provides a suite of 24 kitchen manipulation tasks with a 7-DoF Panda arm. Each observation consists of proprioceptive states and three camera views, including two static cameras and one wrist-mounted camera. The original benchmark provides approximately 3,000 expert Panda-arm trajectories per task for training, and evaluates trained policies on unseen seeds that vary scene layouts, object categories, and object placements.

Benchmark construction. RoboCasa-DC extends RoboCasa with episode-paired demonstrations for demo-conditioned policy learning, allowing policies to condition on either cross- or same-embodiment demonstrations. Fig. 3 illustrates the cross-embodiment case, where Panda-arm trajectories are paired with corresponding GR-1 humanoid demonstrations. For this setting, we pair 100 Panda-arm trajectories per task with corresponding humanoid expert demonstrations across all 24 tasks. We collect each humanoid demonstration by restoring the corresponding Panda-arm trajectory's initial state and teleoperating the humanoid in the same scene with a Leap Motion controller, providing a simulated proxy for real-world human demonstrations. For evaluation, we pre-define 50 evaluation seeds per task and collect corresponding humanoid demonstrations for each seed. For each task, we manually select the demonstration camera view that best captures task execution. We also include a same-embodiment case, following prior work (Kim et al., 2025; Chen et al., 2025), where videos

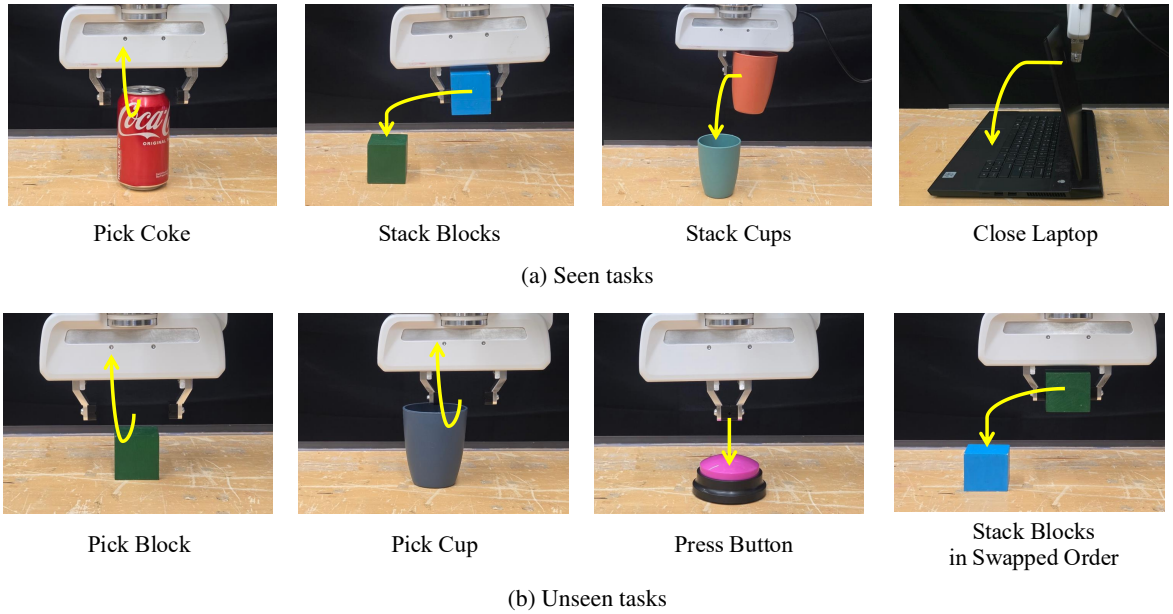

Figure 4. (a) Four seen tasks and (b) four unseen tasks in the real-world benchmark. The yellow arrow indicates the desired path of the end effector.

from Panda-arm trajectories are used as demonstrations. We publicly release RoboCasa-DC to support reproducible evaluation and future work on demo-conditioned policies.

Evaluation protocol. A RoboCasa-DC evaluation split partitions the 24 tasks into seen training tasks and unseen evaluation tasks. Demo-conditioned policies are trained on the paired dataset from the seen tasks. At test time, each policy is evaluated on evaluation seeds from unseen tasks while conditioning on a single demonstration video paired with each seed. This protocol tests whether a policy can extract task-relevant guidance from one-shot demonstrations, including cross-embodiment ones, rather than merely imitating expert trajectories from its own embodiment.

5. Experiments

5.1. Experimental setup

RoboCasa-DC. We evaluate methods on two task splits of RoboCasa-DC: a *category-balanced* split (similar to (Yu et al., 2019)) and a *precision-sensitive* split (motivated by (Goyal et al., 2024; Yin et al., 2025)). Each split holds out five tasks as unseen evaluation tasks, while the remaining 19 tasks are used for training. The category-balanced split samples unseen tasks from different action categories: *CloseDrawer*, *TurnOffSinkFaucet*, *OpenDoubleDoor*, *CoffeeServeMug*, and *PnPCounterToMicrowave*. The precision-sensitive split instead holds out five tasks with small target interaction ratio (TIR) (see Fig. 7), which stress precise spatial grounding: *TurnOffStove*, *CoffeePressButton*, *TurnOffSinkFaucet*, *PnPCounterToSink*, and *PnPStoveToCounter*. We use the dataset and evaluation protocol described in §4. For SeeTraceAct, the trace labels are generated automatically by projecting end-effector positions into each static camera view.

Real-world benchmark. We evaluate demo-conditioned policies on real-world tabletop manipulation tasks using a Franka Panda arm, as shown in Fig. 4. For each of four seen tasks—*Pick Coke*, *Stack Blocks*, *Stack Cups*, and *Close Laptop*—we collect 150 episode pairs, each consisting of a teleoperated Franka trajectory and a task-matched human demonstration video; policies are trained on these pairs. For SeeTraceAct, we obtain trace labels from the robot trajectories during collection by projecting end-effector positions into the static camera view, following (Gu et al., 2024; Li et al., 2025). For evaluation, we define four unseen tasks: *Pick Block*, *Pick Cup*, *Press Button*, and *Stack Blocks in Swapped Order*. These tasks are designed by recombining seen skills and objects, introducing a new contact interaction, and swapping the source and target objects. We test each unseen task across 10 different object positions and poses. Following prior evaluation setups (Kim et al., 2025; Chen et al., 2025; Xu et al., 2023), each policy is conditioned on a one-shot human demonstration video captured in a closely matched scene. We report per-task and average success rates over the unseen tasks. Hardware setup and task language instructions are provided in Fig. 6 and Table 3, respectively.

Baselines. We compare SeeTraceAct against three demo-conditioned baselines: Vid2Robot (Jain et al., 2024), UniSkill (Kim et al., 2025), and ViVLA (Chen et al., 2025).

Method	Category-balanced split		Precision-sensitive split	
	Same-emb demo	Cross-emb demo	Same-emb demo	Cross-emb demo
Vid2Robot (Jain et al., 2024)	21.5%	8.8%	12.6%	6.4%
UniSkill (Kim et al., 2025)	13.3%	11.2%	10.4%	6.0%
ViVLA (Chen et al., 2025)	14.4%	8.0%	8.9%	8.4%
SEETRACEACT (Ours)	**23.0%**	**11.6%**	**14.1%**	**12.8%**

Table 1. Experimental results on ROBOCASA-DC. We report success rates averaged over the five unseen tasks in each split, with 50 evaluation episodes per task. In *Same-emb demo*, policies are conditioned on demonstrations from the same Panda-arm embodiment; in *Cross-emb demo*, policies are conditioned on simulated GR-1 humanoid demonstrations as a proxy for human videos.

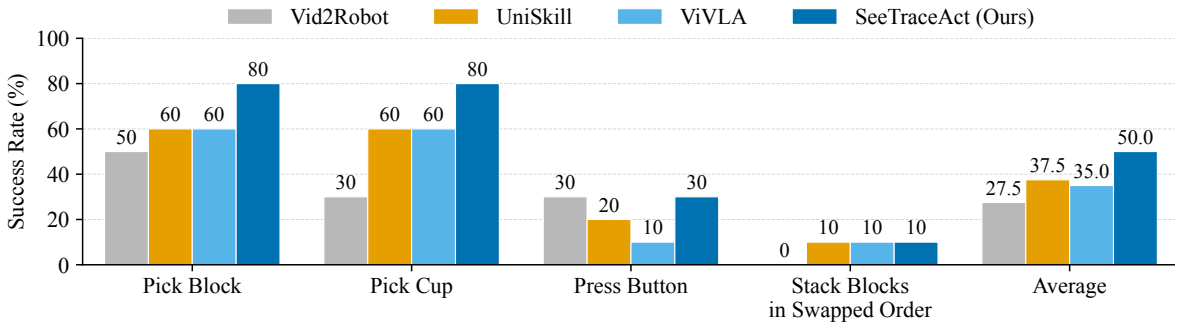

Figure 5. Experimental results on the real-world benchmark with a Franka Panda arm. We report success rates over 10 trials for each of the four unseen tasks, along with their average.

For fair comparison, all baselines are re-implemented on top of the same GR00T N1.5 backbone as SEETRACEACT and trained with their corresponding objectives. Further implementation details are provided in App. B.

5.2. Results

Table 1 summarizes the experimental results on ROBOCASA-DC. SEETRACEACT achieves the best performance across all four evaluation settings, covering both category-balanced and precision-sensitive splits under same-embodiment and cross-embodiment demonstrations. This consistency is important because the two axes stress different aspects of demo-conditioned control: the precision-sensitive split requires fine-grained spatial grounding, while the cross-embodiment setting requires extracting task-relevant motion from demonstrations whose embodiment differs from the robot. Although absolute performance generally decreases under these harder settings, SEETRACEACT maintains its advantage over the baselines, with the largest margin appearing in the precision-sensitive cross-embodiment setting.

Fig. 5 shows the experimental results on the real-world benchmark. SEETRACEACT again achieves the strongest performance across all four unseen tasks, improving average success from 37.5% to 50.0% over the strongest baseline. These results extend the gains observed in simulation to real-world cross-embodiment demo conditioning.

5.3. Larger gains on precision-sensitive tasks

Table 1 shows that SEETRACEACT's margin over the strongest baseline is larger on the precision-sensitive split than on the category-balanced split, increasing from 1.0 to 3.0 percentage points on average across the same- and cross-embodiment settings. This suggests that visibility-aware trace supervision is especially useful when successful execution depends on accurately localizing a small task-critical interaction region.

To examine this trend more directly, we train all models on all 24 ROBOCASA-DC tasks and evaluate them on 50 held-out seeds per task (Table 5), similar to (Goyal et al., 2024). We use the same-embodiment demonstrations to factor out the cross-embodiment gap. In this setting, SEETRACEACT again achieves the highest average success rate. To characterize where these gains arise, we measure precision sensitivity using the target interaction ratio (TIR), defined as the fraction of the image covered by the target interaction region (Fig. 7); a smaller TIR indicates a more precision-sensitive task. We compare TIR with the per-task improvement of SEETRACEACT over each baseline using a one-sided Spearman rank correlation test. Across the 24 tasks, TIR is negatively correlated with the gain over every baseline: Vid2Robot ($\rho = -0.80$, $p < 10^{-5}$), UniSkill ($\rho = -0.45$, $p < 0.05$), and ViVLA ($\rho = -0.52$, $p < 0.01$). The correlation remains significant even when we compare SEETRACEACT against the strongest baseline for each task ($\rho = -0.63$, $p < 10^{-3}$), consistent with the larger gains on

Method	Success rate
GR00T N1.5 backbone (no demonstrations)	9.0%
SEETRACEACT w/o trace supervision	9.4%
SEETRACEACT w/o validity head	8.2%
SEETRACEACT w/o action-aware video encoder	9.2%
SEETRACEACT w/ 3D trace supervision	10.4%
SEETRACEACT (Ours)	**12.2%**

Table 2. Ablation study in the cross-embodiment setting of ROBOCASA-DC. Success rates are averaged over the category-balanced and precision-sensitive splits.

precision-sensitive tasks in Table 1.

5.4. Ablation study

We ablate the main design choices of SEETRACEACT in the cross-embodiment setting of ROBOCASA-DC, averaging results across the category-balanced and precision-sensitive splits. We also report the *GR00T N1.5 backbone (no demonstrations)* as a reference point for a standard VLA policy. Table 2 shows that simply adding demonstration inputs is not sufficient. Removing trace supervision reduces performance to 9.4%, close to the no-demonstration backbone at 9.0%, suggesting that explicitly supervising how the model grounds demonstrations through visual traces provides a clear benefit. Among the ablations, removing the validity head causes the largest degradation, falling to 8.2%. This supports the importance of visibility-aware supervision; forcing the model to regress ill-defined off-screen coordinates can hurt representation learning. Replacing the V-JEPA 2 (Assran et al., 2025) video encoder with the backbone VLA's SigLIP (Zhai et al., 2023) encoder also lowers performance, suggesting that an action-aware video representation is useful for extracting task progression from demonstrations. Finally, 3D trace supervision outperforms the other ablations but still lags behind image-space trace supervision, suggesting that 2D traces offer a training signal better matched to the model's visual inputs.

6. Limitations

Our study has several limitations. First, our real-world evaluation is limited to a tabletop single-arm benchmark, leaving broader evaluation across embodiments, environments, and tasks for future work. Second, our auxiliary objective assumes future trace labels during training, which may not always be readily available; however, such labels could be obtained using motion trackers or foundation models, following prior work (Niu et al., 2024; Zheng et al., 2025; Huang et al., 2025; 2026; Lee et al., 2025; Gu et al., 2024). Finally, despite improving over baselines, the modest absolute success rates on ROBOCASA-DC show that demo-conditioned robot learning leaves substantial room for progress.

7. Conclusion

We presented SEETRACEACT, a demo-conditioned VLA framework that grounds one-shot demonstrations through visibility-aware latent planning. We also introduced ROBOCASA-DC, a benchmark for evaluating demo-conditioned policies with same- and cross-embodiment demonstrations. Experiments show that SEETRACEACT achieves the strongest performance among baselines, with especially large gains on precision-sensitive tasks. These results highlight precise spatial grounding as a promising direction for demo-conditioned robot learning.

LLM/Agent Usage

The authors used LLM-based tools to assist with manuscript writing, including improving clarity, concision, grammar, and organization of the draft. These tools were not used to formulate the research idea and design the method. All technical content, experimental design, analyses, claims, and final manuscript text were reviewed, revised, and verified by the authors. No autonomous research agent was used in conducting the research or preparing the submission.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Alayrac, J., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J. L., Borgeaud, S., Brock, A., Nematzadeh, A., Sharifzadeh, S., Binkowski, M., Barreira, R., Vinyals, O., Zisserman, A., and Simonyan, K. Flamingo: a visual language model for few-shot learning. In *Neural Information Processing Systems (NeurIPS)*, 2022.

Assran, M., Bardes, A., Fan, D., Garrido, Q., Howes, R., Komeili, M., Muckley, M., Rizvi, A., Roberts, C., Sinha, K., Zholus, A., Arnaud, S., Gejji, A., Martin, A., Robert Hogan, F., Dugas, D., Bojanowski, P., Khalidov, V., Labatut, P., Massa, F., Szafraniec, M., Krishnakumar, K., Li, Y., Ma, X., Chandar, S., Meier, F., LeCun, Y., Rabbat, M., and Ballas, N. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv*, abs/2506.09985, 2025.

Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., Jakubczak, S., Jones, T., Ke, L., Levine, S., Li-Bell, A., Mothukuri, M., Nair, S., Pertsch, K., Shi, L. X., Tanner, J., Vuong, Q., Walling, A., Wang, H., and Zhilinsky, U. $\pi 0$: A vision-language-action flow model for general robot control. *ArXiv*, abs/2410.24164, 2024.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Choromanski, K., Ding, T., Driess, D., Dubey, K. A., Finn, C., Florence, P. R., Fu, C., Arenas, M. G., Gopalakrishnan, K., Han, K., Hausman, K., Herzog, A., Hsu, J., Ichter, B., Irpan, A., Joshi, N. J., Julian, R. C., Kalashnikov, D., Kuang, Y., Leal, I., Levine, S., Michalewski, H., Mordatch, I., Pertsch, K., Rao, K., Reymann, K., Ryoo, M. S., Salazar, G., Sanketi, P. R., Sermanet, P., Singh, J., Singh, A., Soricut, R., Tran, H., Vanhoucke, V., Vuong, Q. H., Wahid, A., Welker, S., Wohlhart, P., Xiao, T., Yu, T., and Zitkovich, B. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning (CoRL)*, 2023a.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jackson, T., Jesmonth, S., Joshi, N. J., Julian, R. C., Kalashnikov, D., Kuang, Y., Leal, I., Lee, K.-H., Levine, S., Lu, Y., Malla, U., Manjunath, D., Mordatch, I., Nachum, O., Parada, C., Peralta, J., Perez, E., Pertsch, K., Quiambao, J., Rao, K., Ryoo, M. S., Salazar, G., Sanketi, P. R., Sayed, K., Singh, J., Sontakke, S. A., Stone, A., Tan, C., Tran, H., Vanhoucke, V., Vega, S., Vuong, Q. H., Xia, F., Xiao, T., Xu, P., Xu, S., Yu, T., and Zitkovich, B. Rt-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems (RSS)*, 2023b.

Chen, G., Wang, M., Shao, Q., Zhou, Z., Mao, W., Cui, T., Zhu, M., Deng, Y., Yang, L., Zhang, Z., Yang, Y., Chen, H., and Yue, Y. See once, then act: Vision-language-action model with task learning from one-shot video demonstrations. *ArXiv*, abs/2512.07582, 2025.

Goyal, A., Blukis, V., Xu, J., Guo, Y., Chao, Y.-W., and Fox, D. Rvt2: Learning precise manipulation from few demonstrations. In *Robotics: Science and Systems (RSS)*, 2024.

Goyal, R., Kahou, S. E., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fründ, I., Yianilos, P. N., Mueller-Freitag, M., Hoppe, F., Thurau, C., Bax, I., and Memisevic, R. The "something something" video database for learning and evaluating visual common sense. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

Gu, J., Kirmani, S., Wohlhart, P., Lu, Y., Gonzalez Arenas, M., Rao, K., Yu, W., Fu, C., Gopalakrishnan, K., Xu, Z., Sundaresan, P., Xu, P., Su, H., Hausman, K., Vuong, Q., and Xiao, T. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches. In *International Conference on Learning Representations (ICLR)*, 2024.

Heppert, N., Argus, M., Welschehold, T., Brox, T., and Valada, A. Ditto: Demonstration imitation by trajectory transformation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.

Huang, C.-P., Wu, Y.-H., Chen, M.-H., Wang, Y.-C. F., and Yang, F.-E. Thinkact: Vision-language-action reasoning via reinforced visual latent planning. In *Neural Information Processing Systems (NeurIPS)*, 2025.

Huang, C.-P., Man, Y., Yu, Z., Chen, M.-H., Kautz, J., Wang, Y.-C. F., and Yang, F.-E. Fast-thinkact: Efficient vision-language-action reasoning via verbalizable latent planning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Jaegle, A., Gimeno, F., Brock, A., Vinyals, O., Zisserman, A., and Carreira, J. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning (ICML)*, 2021.

Jain, V., Attarian, M., Joshi, N. J., Wahid, A., Driess, D., Vuong, Q., Sanketi, P. R., Sermanet, P., Welker, S., Chan, C., Gilitschenski, I., Bisk, Y., and Dwibedi, D. Vid2robot: End-to-end video-conditioned policy learning with cross-attention transformers. In *Robotics: Science and Systems (RSS)*, 2024.

Kay, W., Carreira, J., Simonyan, K., Zhang, B. H., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, A., Suleyman, M., and Zisserman, A. The kinetics human action video dataset. *ArXiv*, abs/1705.06950, 2017.

Khazatsky, A., Pertsch, K., Nair, S., Balakrishna, A., Dasari, S., Karamcheti, S., Nasiriany, S., Srirama, M. K., Chen, L. Y., Ellis, K., Fagan, P., Hejna, J., Itkina, M., Lepert, M., Ma, Y., Miller, P., Wu, J., Belkhale, S., Dass, S., Ha, H., Jain, A., Lee, A., Lee, Y., Memmel, M., Park, S., Radosavovic, I., Wang, K., Zhan, A., Black, K., Chi, C., Hatch, K. B., Lin, S., Lu, J., Mercat, J.-P., Rehman, A., Sanketi, P. R., Sharma, A., Simpson, C. B., Vuong, Q., Walke, H. R., Wulfe, B., Xiao, T., Yang, J. H., Yavary, A., Zhao, T., Agia, C., Baijal, R., Castro, M. G., Chen, D. L., Chen, Q., Chung, T., Drake, J., Foster, E. P., Gao, J., Herrera, D. A., Heo, M., Hsu, K., Hu, J., Irshad, M. Z., Jackson, D., Le, C., Li, Y., Lin, K., Lin, R., Ma, Z., Maddukuri, A., Mirchandani, S., Morton, D., Nguyen, T., O'Neill, A., Scalise, R. M., Seale, D., Son, V., Tian, S., Tran, E., Wang, A., Wu, Y., Xie, A., Yang, J., Yin, P., Zhang, Y., Bastani, O., Berseth, G., Bohg, J., Goldberg, K., Gupta, A., Gupta, A., Jayaraman, D., Lim, J. J., Malik, J., Mart'in-Mart'in, R., Ramamoorthy, S., Sadigh, D., Song, S., Wu, J., Yip, M. C., Zhu, Y., Kollar, T., Levine, S., and Finn, C. Droid: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems (RSS)*, 2024.

Kim, H., Kang, J., Kang, H., Cho, M., Kim, S. J., and Lee, Y. Uniskill: Imitating human videos via cross-embodiment skill representations. In *Conference on Robot Learning (CoRL)*, 2025.

Kim, M. J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E. P., Lam, G., Sanketi, P. R., Vuong, Q., Kollar, T., Burchfiel, B., Tedrake, R., Sadigh, D., Levine, S., Liang, P., and Finn, C. Openvla: An open-source vision-language-action model. *ArXiv*, abs/2406.09246, 2024.

Lee, J., Duan, J., Fang, H., Deng, Y., Liu, S., Li, B., Fang, B., Zhang, J., Wang, Y. R., Lee, S., Han, W., Pumacay, W., Wu, A., Hendrix, R., Farley, K., VanderBilt, E., Farhadi, A., Fox, D., and Krishna, R. Molmoact: Action reasoning models that can reason in space. *ArXiv*, abs/2508.07917, 2025.

Li, J., Zhu, Y., Xie, Y., Jiang, Z., Seo, M., Pavlakos, G., and Zhu, Y. Okami: Teaching humanoid robots manipulation skills through single video imitation. In *Conference on Robot Learning (CoRL)*, 2024.

Li, Y., Deng, Y., Zhang, J., Jang, J., Memmel, M., Yu, R., Garrett, C. R., Ramos, F., Fox, D., Li, A., Gupta, A., and Goyal, A. Hamster: Hierarchical action models for open-world robot manipulation. In *International Conference on Learning Representations (ICLR)*, 2025.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Nasiriany, S., Maddukuri, A., Zhang, L., Parikh, A., Lo, A., Joshi, A., Mandlekar, A., and Zhu, Y. Robocasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems (RSS)*, 2024.

Niu, D., Sharma, Y., Biamby, G., Quenum, J., Bai, Y., Shi, B., Darrell, T., and Herzig, R. Llarva: Vision-action instruction tuning enhances robot learning. In *Conference on Robot Learning (CoRL)*, 2024.

NVIDIA, Bjorck, J., Castañeda, F., Cherniadev, N., Da, X., Ding, R., Fan, L. J., Fang, Y., Fox, D., Hu, F., Huang, S., Jang, J., Jiang, Z., Kautz, J., Kundalia, K., Lao, L., Li, Z., Lin, Z., Lin, K., Liu, G., Llontop, E., Magne, L., Mandlekar, A., Narayan, A., Nasiriany, S., Reed, S., Tan, Y. L., Wang, G., Wang, Z., Wang, J., Wang, Q., Xiang, J., Xie, Y., Xu, Y., Xu, Z.-T., Ye, S., Yu, Z., Zhang, A., Zhang, H., Zhao, Y., Zheng, R., and Zhu, Y. Gr00t n1: An open foundation model for generalist humanoid robots. *ArXiv*, abs/2503.14734, 2025.

O'Neill, A., Rehman, A., Maddukuri, A., Gupta, A., Padalkar, A., Lee, A., Pooley, A., Gupta, A., Mandlekar, A., Jain, A., Tung, A., Bewley, A., Herzog, A., Irpan, A., Khazatsky, A., Rai, A., Gupta, A., Wang, A., Singh, A., Garg, A., Kembhavi, A., Xie, A., Brohan, A., Raffin, A., Sharma, A., Yavary, A., Jain, A., Balakrishna, A., Wahid, A., Burgess-Limerick, B., Kim, B., Schölkopf, B., Wulfe, B., Ichter, B., Lu, C., Xu, C., Le, C., Finn, C., Wang, C., Xu, C., Chi, C., Huang, C., Chan, C., Agia, C., Pan, C., Fu, C., Devin, C., Xu, D., Morton, D., Driess, D., Chen, D., Pathak, D., Shah, D., Büchler, D., Jayaraman, D., Kalashnikov, D., Sadigh, D., Johns, E., Foster, E. P., Liu, F., Ceola, F., Xia, F., Zhao, F., Stulp, F., Zhou, G., Sukhatme, G. S., Salhotra, G., Yan, G., Feng, G., Schiavi, G., Berseth, G., Kahn, G., Wang, G., Su, H., Fang, H., Shi, H., Bao, H., Amor, H. B., Christensen, H. I., Furuta, H., Walke, H. R., Fang, H., Ha, H., Mordatch, I., Radosavovic, I., Leal, I., Liang, J., Abou-Chakra, J., Kim, J., Drake, J., Peters, J., Schneider, J., Hsu, J., Bohg, J., Bingham, J., Wu, J., Gao, J., Hu, J., Wu, J., Wu, J., Sun, J., Luo, J., Gu, J., Tan, J., Oh, J., Wu, J., Lu, J., Yang, J., Malik, J., Silvério, J., Hejna, J., Booher, J., Tompson, J., Yang, J., Salvador, J., Lim, J. J., Han, J., Wang, K., Rao, K., Pertsch, K., Hausman, K., Go, K., Gopalakrishnan, K., Goldberg, K., Byrne, K., Oslund, K., Kawaharazuka, K., Black, K., Lin, K., Zhang, K., Ehsani, K., Lekkala,

K., Ellis, K., Rana, K., Srinivasan, K. P., Fang, K., Singh, K. P., Zeng, K.-H., Hatch, K. B., Hsu, K., Itti, L., Chen, L. Y., Pinto, L., Li, F.-F., Tan, L., Fan, L. J., Ott, L., Lee, L., Weihs, L., Chen, M., Lepert, M., Memmel, M., Tomizuka, M., Itkina, M., Castro, M. G., Spero, M., Du, M., Ahn, M., Yip, M. C., Zhang, M., Ding, M., Heo, M., Srirama, M. K., Sharma, M., Kim, M. J., Irshad, M. Z., Kanazawa, N., Hansen, N., Heess, N. M. O., Joshi, N. J., Suenderhauf, N., Liu, N., Palo, N. D., Shafiullah, N. M., Mees, O., Kroemer, O., Bastani, O., Sanketi, P. R., Miller, P., Yin, P., Wohlhart, P., Xu, P., Fagan, P. D., Mitrano, P., Sermanet, P., Abbeel, P., Sundaresan, P., Chen, Q., Vuong, Q., Rafailov, R., Tian, R., Doshi, R., Martín-Martín, R., Baijal, R., Scalise, R., Hendrix, R., Lin, R., Qian, R., Zhang, R., Mendonca, R., Shah, R., Hoque, R., Julian, R. C., Bustamante, S., Kirmani, S., Levine, S., Lin, S., Moore, S., Bahl, S., Dass, S., Sonawani, S. D., Song, S., Xu, S., Haldar, S., Karamcheti, S., Adebola, S., Guist, S., Nasiriany, S., Schaal, S., Welker, S., Tian, S., Ramamoorthy, S., Dasari, S., Belkhale, S., Park, S., Nair, S., Mirchandani, S., Osa, T., Gupta, T., Harada, T., Matsushima, T., Xiao, T., Kollar, T., Yu, T., Ding, T., Davchev, T., Zhao, T., Armstrong, T., Darrell, T., Chung, T., Jain, V., Vanhoucke, V., Zhan, W., Zhou, W., Burgard, W., Chen, X., Wang, X., Zhu, X., Geng, X., Liu, X., Xu, L., Li, X., Lu, Y., Ma, Y. J., Kim, Y., Chebotar, Y., Zhou, Y., Zhu, Y., Wu, Y., Xu, Y., Wang, Y., Bisk, Y., Cho, Y., Lee, Y., Cui, Y., Cao, Y., Wu, Y.-H., Tang, Y., Zhu, Y., Zhang, Y., Jiang, Y., Li, Y., Li, Y., Iwasawa, Y., Matsuo, Y., Ma, Z., Xu, Z., Cui, Z. J., Zhang, Z., and Lin, Z. Open x-embodiment: Robotic learning datasets and rt-x models : Open x-embodiment collaboration. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Park, S., Bharadhwaj, H., and Tulsiani, S. Demodiffusion: One-shot human imitation using pre-trained diffusion policy. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

Walke, H. R., Black, K., Lee, A., Kim, M. J., Du, M., Zheng, C., Zhao, T., Hansen-Estruch, P., Vuong, Q. H., He, A. W., Myers, V., Fang, K., Finn, C., and Levine, S. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023.

Xu, M., Xu, Z., Chi, C., Veloso, M., and Song, S. Xskill: Cross embodiment skill discovery. In *Conference on Robot Learning (CoRL)*, 2023.

Ye, S., Jang, J., Jeon, B., Joo, S., Yang, J., Peng, B., Mandlekar, A., Tan, R., Chao, Y.-W., Lin, B. Y., et al. Latent action pretraining from videos. In *International Conference on Learning Representations (ICLR)*, 2025.

Yin, Y., Han, Z., Aarya, S., Xu, S., Wang, J., Peng, J., Wang, A., Yuille, A., and Shu, T. Partinstruct: Part-level instruction following for fine-grained robot manipulation. In *Robotics: Science and Systems (RSS)*, 2025.

Yu, T., Quillen, D., He, Z., Julian, R. C., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2019.

Zhai, X., Mustafa, B., Kolesnikov, A., and Beyer, L. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Zheng, R., Liang, Y., Huang, S., Gao, J., Daum'e, H., Kolobov, A., Huang, F., and Yang, J. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. In *International Conference on Learning Representations (ICLR)*, 2025.

A. Benchmark Details

A.1. RoboCasa-DC

To collect GR-1 humanoid demonstrations, we restore the initial simulation state of each pre-defined Panda-arm trajectory and teleoperate the humanoid in the same scene. Because the released RoboCasa trajectories do not include the generative fixture textures needed to exactly restore the original rendered observations, we randomly assign fixture textures, replay the Panda-arm trajectories, and re-render the corresponding videos. For the cross-embodiment setting, we use the paired Panda–GR-1 episodes described in Sec. 4 for training and the pre-defined evaluation seeds for testing. In a small number of cases, we could not obtain expert humanoid demonstrations for the corresponding evaluation seeds; we exclude these seeds from all reported evaluations. For the same-embodiment case, demonstrations are rendered from the Panda-arm trajectories themselves, allowing us to use the full RoboCasa training set, up to approximately 3,000 trajectories per task.

A.2. Real-world benchmark

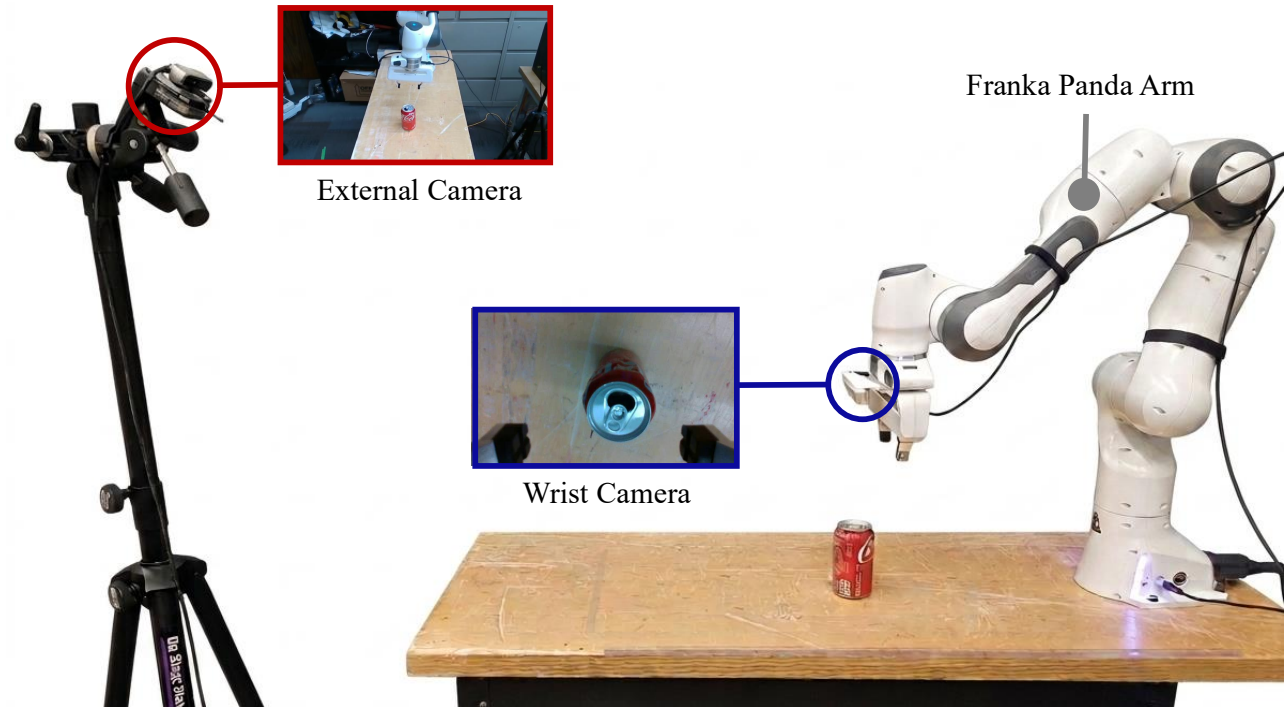

Figure 6. Hardware setup for the real-world experiments. The setup consists of a Franka Panda arm, an external third-person-view camera, and a wrist-mounted camera.

Hardware setup. Our real-world experiments use a 7-DoF Franka Panda arm mounted on a tabletop and equipped with a parallel-jaw gripper (Fig. 6). The visual observations are captured from two RGB cameras: a single external camera providing the third-person view and a wrist-mounted camera attached near the end effector. Both camera streams are captured at a resolution of 640×480.

Tasks. The language instructions for the real-world tasks shown in Fig. 4 are listed in Table 3.

B. Model Details

Common setup. For fair comparison, all methods are implemented on top of the same GR00T N1.5 architecture. Each method receives the same camera views, language instruction, demonstration video, and robot states as input, and predicts action chunks with the same horizon. Each baseline method is trained with the corresponding objectives proposed in the original work. For action prediction, we use an action horizon of $H = 16$ and perform $K = 4$ denoising steps at inference time. For the action prediction loss, we sample $u \sim \mathrm{Beta}(1.5, 1.0)$ and set $\tau = (s - u)/s$ with $s = 0.999$, following prior work (Black et al., 2024; NVIDIA et al., 2025). The hyperparameters used in our experiments are summarized in Table 4. Further details are provided in our code.

Task	Language instruction
Seen tasks	
Pick Coke	"Grab the coke can and lift it up"
Stack Blocks	"Stack the blue block on top of the green block"
Stack Cups	"Stack the orange cup onto the green cup"
Close Laptop	"Close the laptop lid fully"
Unseen tasks	
Pick Block	"Grab the green block and lift it up"
Pick Cup	"Grab the blue cup and lift it up"
Press Button	"Press the pink button"
Stack Blocks in Swapped Order	"Stack the green block on top of the blue block"

Table 3. Real-world benchmark tasks and corresponding language instructions.

SeeTraceAct. For demonstration video encoding, we follow the open-sourced V-JEPA 2 preprocessing and uniformly sample 64 frames from each demonstration video. The sampled frames are resized to the video encoder input resolution of 256×256. Directly feeding all video features into the VLM would substantially increase the input sequence length. With a 3D patch size of $2 \times 16 \times 16$, a 64-frame video produces $32 \times 16 \times 16 = 8192$ video tokens, which is much larger than the approximately 200 image and language tokens used by the backbone. We therefore employ a Perceiver Resampler to compress the video features into 32 video tokens before appending them to the VLM input sequence. As in the original GR00T N1.5, robot observation images are encoded with its SigLIP (Zhai et al., 2023) image encoder.

Vid2Robot (Jain et al., 2024). Since Vid2Robot does not provide an official implementation, we re-implement the method described in the paper within our common GR00T N1.5 backbone. The original method uses a shared visual encoder for robot observations and prompt videos, so we use the SigLIP encoder from GR00T N1.5 to encode demonstration videos as well. We then pass the resulting video features through a Perceiver Resampler to compress them into 32 video tokens. We train the model with the action prediction loss, the prompt–robot video contrastive loss (VVCL), and the video–text contrastive loss (VTCL). In the same-embodiment setting, we omit VVCL because it already attains its theoretical minimum.

UniSkill (Kim et al., 2025). We use the inverse skill dynamics (ISD) module from the official UniSkill implementation. At each control timestep t, UniSkill passes the demonstration frames at t and $t + k$ to the ISD module, where k is the skill interval, to obtain a skill representation for subsequent action prediction. Following the original implementation, we set $k = 20$ frames. We project the resulting skill representation into the VLM embedding space and append it as a single token to the input image, language, and video tokens.

ViVLA (Chen et al., 2025). Because ViVLA does not provide an official implementation, we re-implement the method described in the paper within our common GR00T N1.5 backbone. To isolate the effect of the latent action prediction objective rather than the video encoder, we use the same V-JEPA 2 video encoder and Perceiver Resampler as those used in SEETRACEACT to generate input video tokens. For the latent action encoder, we use the open-sourced LAPA (Ye et al., 2025). We remove the prediction of the number of LACT tokens and instead fix the number of tokens to 9, allowing ViVLA to generate each action chunk with a single forward pass. Specifically, we divide the demonstration video into 8 equal segments and train 8 LACT tokens to predict the LAPA representations for the corresponding segments, while the remaining LACT token predicts the representation of the robot's subsequent latent action. These 9 auxiliary tokens provide at least as much capacity as the 5 learnable query tokens used by SEETRACEACT.

Hyperparameter	Vid2Robot	UniSkill	ViVLA	SeeTraceAct
Training				
Training steps		50,000		
Batch size		128		
Optimizer		AdamW (β_1=0.95, β_2=0.999, ϵ=10^{-8})		
Learning rate		10^{-4}		
Weight decay		10^{-5}		
Warmup steps		2500		
Learning rate scheduler		cosine		
Gradient clip norm		1.0		
Demonstration video				
Encoder architecture	SigLIP + Perceiver	ISD	V-JEPA 2 + Perceiver	
Frame sampling method	uniform	timestep	uniform	
Encoder input resolution	224×224	224×224	256×256	
# frames sampled	64	—	64	
# video tokens	32	1	32	
Action prediction				
Action horizon (H)		16		
Denoising steps (K)		4		
Trace supervision				
Validity loss weight (λ_{valid})	—	—	—	1.0
Visual trace loss weight (λ_{trace})	—	—	—	0.2
# future trace points (N)	—	—	—	5
Temporal stride (Δ)	—	—	—	5

Table 4. Hyperparameters used for the compared methods on RoboCasa-DC. *Encoder input resolution* follows each video encoder's default. Following (Kim et al., 2025), UniSkill extracts demonstration frames at timesteps t and $t + k$ at each control timestep, where k is the skill interval, while the other methods use uniformly sampled frames from the episode.

Task	Vid2Robot	UniSkill	ViVLA	SeeTraceAct	TIR
CloseDoubleDoor	83.3	51.9	85.2	72.2	6.403
CloseDrawer	98.1	100.0	94.4	94.4	4.736
CloseSingleDoor	94.4	96.3	94.4	92.6	8.801
CoffeePressButton	64.8	37.0	70.4	83.3	0.092
CoffeeServeMug	53.7	57.4	74.1	53.7	0.507
CoffeeSetupMug	25.9	18.5	27.8	24.1	0.708
OpenDoubleDoor	20.8	13.2	41.5	17.0	1.349
OpenDrawer	33.3	48.1	53.7	48.1	0.281
OpenSingleDoor	57.4	29.6	55.6	61.1	0.439
PnPCabToCounter	42.6	51.9	44.4	50.0	0.427
PnPCounterToCab	33.3	11.1	22.2	33.3	0.671
PnPCounterToMicrowave	20.4	14.8	22.2	18.5	0.452
PnPCounterToSink	37.0	14.8	31.5	38.9	0.153
PnPCounterToStove	30.2	7.5	32.1	26.4	0.366
PnPMicrowaveToCounter	27.8	9.3	51.9	33.3	0.275
PnPSinkToCounter	7.4	7.4	18.5	22.2	0.165
PnPStoveToCounter	40.7	18.5	51.9	48.1	0.165
TurnOffMicrowave	79.6	48.1	74.1	83.3	0.262
TurnOffSinkFaucet	75.9	72.2	75.9	85.2	0.116
TurnOffStove	37.0	42.6	46.3	50.0	0.049
TurnOnMicrowave	64.8	63.0	68.5	66.7	0.470
TurnOnSinkFaucet	20.4	37.0	31.5	46.3	0.189
TurnOnStove	60.4	66.0	54.7	79.2	0.067
TurnSinkSpout	87.0	94.4	88.9	87.0	0.336
Average	49.9	42.1	54.7	54.8	1.139

Table 5. Per-task success rates when training all methods on all 24 RoboCasa-DC tasks. Results are reported in the same-embodiment setting, with evaluation on 50 held-out seeds per task. We also report the target interaction ratio (TIR), defined as the area ratio of the target interaction region to the full camera view, as visualized in Fig. 7. Lower TIR indicates higher precision sensitivity. Success rates and TIR are reported as percentages.

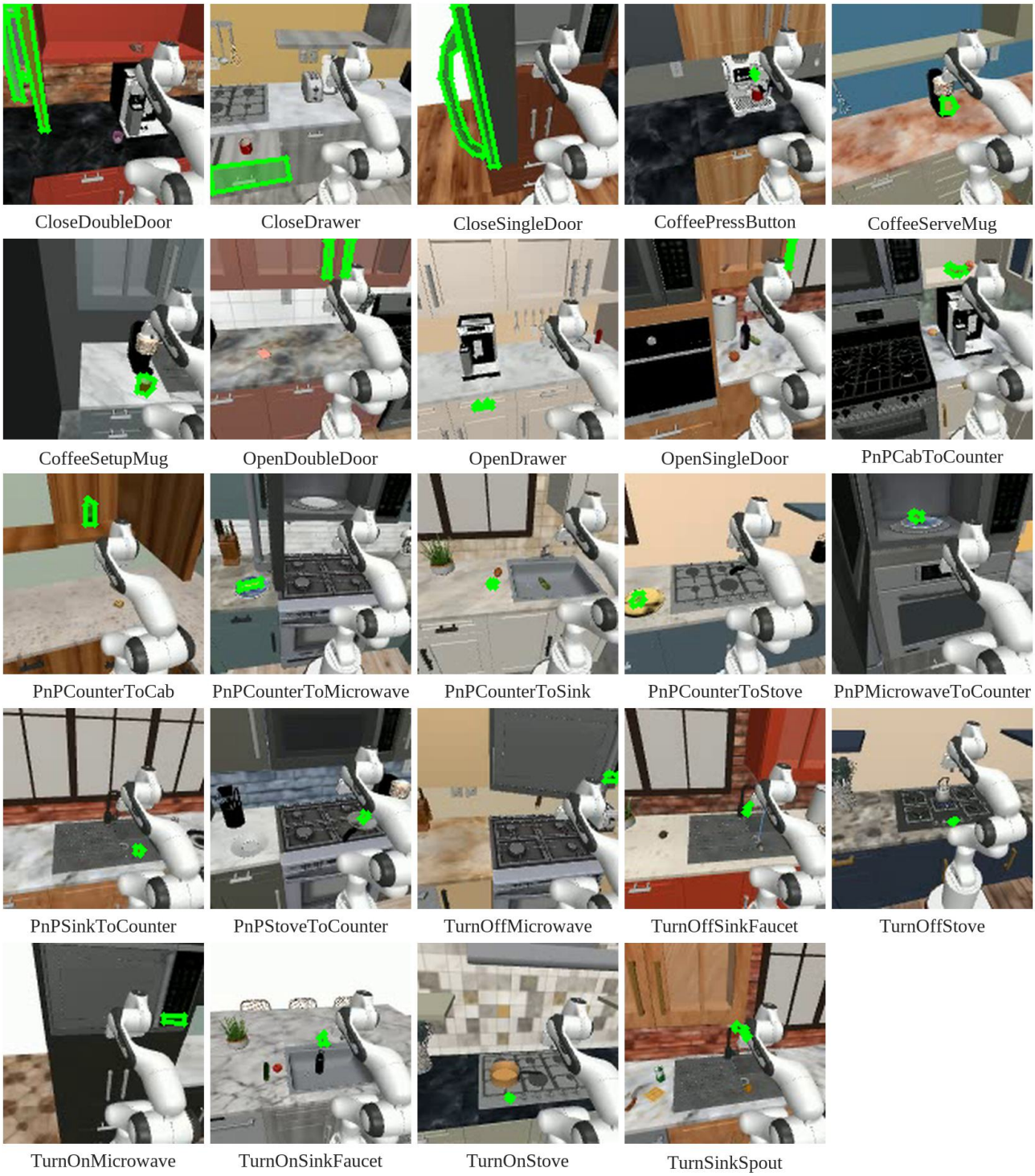

Figure 7. Target interaction regions in RoboCasa-DC tasks. The highlighted region indicates the area in a static camera view where the robot must interact to complete the task. We use the area ratio of this region to the full camera view as the *target interaction ratio* (TIR) in §5 and Table 5; lower TIR indicates a more precision-sensitive task.