# DELE-W0.5: INFERRING ACTION FROM FUTURE LATENT STATE FOR ROBOTIC MANIPULATION

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

## ABSTRACT

Video-generation-based World-Action Models (WAMs) jointly predict dense future visual trajectories and robot actions. For robotic manipulation, we investigate whether action learning can benefit from predicting physical outcomes without reconstructing every intermediate frame. In this paper, we propose `DELE-w0.5`, which learns robot actions jointly with future latent-state prediction, without video generation. The future latent state models action-relevant physical outcomes and connects world modeling with action learning through shared representations. The principle of `DELE-w0.5` is to model how the real world changes under robot actions, rather than how its visual appearance evolves frame by frame. Across 640 real-robot trials on four long-horizon manipulation tasks, `DELE-w0.5` achieves the best performance among all compared policies, attaining 62.5% overall full-task success and 81.3% macro ordered-stage progress. It outperforms the strongest baseline by 32.5 percentage points in full-task success and 20.1 percentage points in macro progress.

## 1 INTRODUCTION

Embodied intelligence has made substantial progress in robotic manipulation (Physical Intelligence et al., 2025; Li et al., 2025; 2026b). Vision-Language-Action (VLA) policies and World-Action Models (WAMs) are two prominent approaches. VLA policies predict actions from visual observations and language instructions. In a common design, a pretrained multimodal backbone provides representations for an action decoder, as illustrated in Figure 1(a). VLA models have demonstrated strong generalization, but robust transfer to unseen physical motions and environments remains challenging (Ye et al., 2026b; Kim et al., 2026). WAMs connect predictive world modeling with action generation. Video-based approaches learn from future visual trajectories using video-generation models (Du et al., 2023; Ye et al., 2026b; Kim et al., 2026), as illustrated in Figure 1(b).

Video-based WAMs learn useful spatiotemporal priors through future visual prediction, but representing multiple future frames introduces additional visual tokens and computation. Recent methods reduce this overhead through compact backbones and latent downsampling (Li et al., 2026c), or decouple action decoding from future-video generation (Ye et al., 2026a; GigaWorld Team et al., 2026). For models that jointly denoise video-action latents at deployment, future-video generation also contributes to inference cost. These advances motivate a complementary question: can compact future-state prediction support action learning without reconstructing a visual trajectory?

In this paper, we propose `DELE-w0.5`, which couples action learning with compact future latent-state prediction.

**Rethinking World Model for Robotic Manipulation**. We study future-state prediction as a compact training objective for robotic manipulation. Rather than reconstructing intermediate frames, we predict the latent state reached after an action chunk. This formulation focuses supervision on the corresponding physical outcome while retaining action generation as a joint learning objective.

**Inferring Action from Future-State**. `DELE-w0.5` learns robot actions and their corresponding future latent states through shared representations. Future-state supervision shapes action learning during joint training, while a block-structured attention mask permits action-only inference at deployment. This design avoids intermediate-frame reconstruction and achieves a median core-model inference



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.

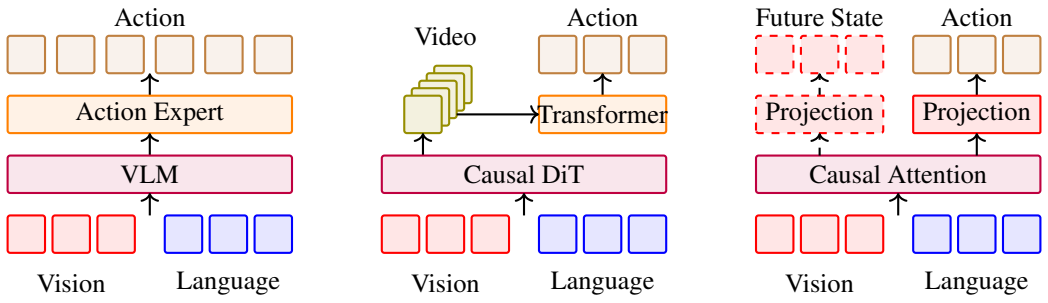



Figure 1: Representative models for robotic manipulation. DELE-w0.5 jointly learns actions and future latent states; dashed components are used only during training.

latency of 87.5 ms per action chunk in our evaluation setup. We compare the modeled prediction targets of three representative formulations:

- VLA models $p_\theta(\mathbf{A}_t|\mathbf{O}_t, \mathbf{q}_t, \mathbf{L})$;
- Video-based WAM models $p_\theta(\mathbf{A}_t, \mathbf{O}_{t:t+\Delta t}|\mathbf{O}_t, \mathbf{q}_t, \mathbf{L})$;
- DELE-w0.5 models $p_\theta(\mathbf{A}_t, \mathbf{z}^\star_{t+\Delta t}|\mathbf{O}_t, \mathbf{q}_t, \mathbf{L})$,

where $\mathbf{O}_{t:t+\Delta t}$ is a future visual trajectory and $\mathbf{z}^\star_{t+\Delta t}$ is the future state encoded by the frozen vision encoder.

## 2 Related Work

**Vision-Language-Action Models.** VLA models connect pretrained vision-language knowledge with robot control. PaLM-E (Driess et al., 2023) and RT-2 (Zitkovich et al., 2023) demonstrate that large-scale multimodal models can transfer semantic knowledge to embodied tasks, while Open X-Embodiment (O'Neill et al., 2024) shows that heterogeneous robot data enables cross-embodiment generalization. With these foundations, Octo (Mees et al., 2024) and OpenVLA (Kim et al., 2024) develop generalist robot policies with open-source large-scale training pipelines. Recent studies improve VLA efficiency and action generation through compact architectures and expressive action representations, including TinyVLA and SmolVLA for efficient deployment (Wen et al., 2025; Shukor et al., 2025), RDT-1B and CogACT for diffusion-based action (Liu et al., 2025; Li et al., 2024), and $\pi_0$ for flow-matching-based continuous action (Black et al., 2026). Other works explore scalable action tokenization, embodiment transfer, and long-horizon reasoning, such as FAST (Pertsch et al., 2025), UniVLA (Wang et al., 2025), $\pi_{0.5}$ (Physical Intelligence et al., 2025), HAMLET (Koo et al., 2026), ProgressVLA (Yan et al., 2026), and Qwen-VLA (Wang et al., 2026).

Direct action prediction maps observations and instructions to actions, but does not by itself require prediction of their physical consequences. `DELE-w0.5` couples action learning with future latent-state prediction through shared representations, learning actions and their corresponding physical outcomes jointly.

**World-Action Models.** World models learn predictive representations of the environment to support control and planning. UniPi (Du et al., 2023) formulates decision making as text-conditioned video generation, and Genie (Bruce et al., 2024) learns action-controllable environments from unlabeled videos. GR-1 (Wu et al., 2023) and GR-2 (Cheang et al., 2024) further connect future visual prediction with robot action generation through large-scale video pretraining. Other approaches incorporate additional structure, including 3D reasoning in 3D-VLA (Zhen et al., 2024), IRASim (Zhu et al., 2025), predictive inverse dynamics in Seer (Tian et al., 2025), and future-state planning in V-JEPA 2-AC (Assran et al., 2025). More recent World-Action Models explicitly combine video prediction and action generation. DreamZero (Ye et al., 2026b) builds WAM on a pretrained video diffusion backbone, while $\tau_0$-WM (Zhou et al., 2026) integrates video-action prediction with action-conditioned simulation. GigaWorld-Policy (Ye et al., 2026a) explores action-centered world modeling by reducing

the dependency on future-video generation. To improve the efficiency of video-based WAMs, Flash-WAM (Akbari et al., 2026) distills iterative video-action diffusion, and Efficient-WAM (Li et al., 2026a) reduces computational cost through compact video experts and sparse visual tokens.

Video-generation-based WAMs model dense visual transitions, introducing additional visual tokens and computation. In contrast, `DELE-w0.5` predicts a compact future latent state jointly with robot actions, without intermediate-frame reconstruction. At deployment, the model generates actions without future-state tokens.

## 3 PRELIMINARY

**Robotic Manipulation.** We model robotic manipulation as a probabilistic inference model. At each time $t$, the robot receives multi-view RGB observations $\mathbf{O}_t$, proprioception $\mathbf{q}_t$, and a language instruction $\mathbf{L}$. We predict actions according to the policy $\mathbf{A}_t \sim \pi(\cdot|\mathbf{O}_t, \mathbf{q}_t, \mathbf{L})$, where

$$\mathbf{A}_t = [\mathbf{a}_{t+1}, \mathbf{a}_{t+2}, \cdots, \mathbf{a}_{t+H}]$$

corresponds to a chunk of future actions to be executed by the robot. For single-arm manipulation, each $\mathbf{a} \in \mathbb{R}^7$ represents a 7-DoF action: a 3-DoF relative positional displacement $(x, y, z)$, a 3-DoF Euler angle rotation (`roll`, `pitch`, `yaw`), and a 1-DoF binary gripper state $g \in \{0, 1\}$. The dual-arm manipulation extends the setting to a 14-DoF space. The proprioceptive input $\mathbf{q}_t$ consists of the current end-effector pose and gripper state. After normalization and linear projection, the resulting EEF-anchor tokens are prepended to the action tokens and processed by the action refiner and subsequent joint attention layers.

**Language and Vision Tokenization.** The language instruction $\mathbf{L}$ defines the task to be performed by the robot (e.g. " `put the flowers into the vase`"). We encode $\mathbf{L}$ using Qwen3 (Yang et al., 2025) as follows:

$$\hat{\mathbf{l}} = \texttt{Text Encoder}(\mathbf{L}). \tag{1}$$

At each time $t$, the current observation $\mathbf{O}_t \in \mathbb{R}^{v\times 3\times h\times w}$ contains multi-view images. Our robot uses three views ($v = 3$): one head-mounted camera and one wrist-mounted camera on each arm. The vision tokenization encoder compresses the input images $\mathbf{O}_t$ into a latent representation, denoted as

$$\hat{\mathbf{z}}_t = \texttt{Vision Encoder}(\mathbf{O}_t), \tag{2}$$

where $\hat{\mathbf{z}}_t \in \mathbb{R}^{c'\times h'\times w'}$ has $c'$ channels and a spatial size of $h' \times w'$. In this paper, we apply DINO-v3 (Siméoni et al., 2025) as the vision encoder to encode the input images. The DINO-v3 encoder provides comprehensive representations that capture spatial and geometric relationships across observations.

**Timestep Embedding.** We encode the scalar timestep $\tau$ using sinusoidal features followed by a two-layer MLP. Let $d$ be the embedding dimension, $k = \lfloor d/2 \rfloor$, $\tilde{\mathbf{t}} \in \mathbb{R}^d$, for each $j \in \{0, 1, \cdots, d-1\}$:

$$\tilde{\mathbf{t}}[j] = \begin{cases} \cos\left(\tau \cdot 10000^{-j/k}\right), & \text{if } 0 \le j < k, \\ \sin\left(\tau \cdot 10000^{-(j-k)/k}\right), & \text{if } k \le j < 2k, \\ 0, & \text{if } j = d-1 \ \& \ d \text{ is an odd number.} \end{cases} \tag{3}$$

We apply a two-layer MLP to map the vector $\tilde{\mathbf{t}}$ to a $d$-dimensional embedding: $\mathbf{t} = \texttt{FNN}(\tilde{\mathbf{t}})$, where `FNN` first expands the dimensionality of $\tilde{\mathbf{t}}$ and then projects it back to $d$ dimensions. We summarize this embedding as follows:

$$\mathbf{t} = \texttt{Timestep Embedding}(\tau). \tag{4}$$

## 4 FRAMEWORK OF DELE-W0.5

The framework of `DELE-w0.5` adopts a dual-stream attention architecture consisting of a condition stream and a noise stream. The condition stream provides a shared representation of the task and the current state. During training, the noise stream processes noisy action chunks and future-state latents. Guided by masked attention, it predicts clean action and future-state targets, which are converted to velocities for training and sampling. This design separates the known conditioning information from the prediction targets, while unifying the learning of robot actions and their corresponding future states.

### 4.1 CONDITION STREAM BLOCK

**Input and Tokenization**. For each time $t$, we encode the language instruction input $\mathbf{L}$ and current state input $\mathbf{O}_t$ according to the methods (1) and (2) correspondingly, and obtain the language token $\hat{\mathbf{l}}$, latent space of vision $\hat{\mathbf{z}}_t$. Then we apply the projection operator to ensure that feature information $\mathbf{l}$ and $\mathbf{z}_t$ have the same hidden dimension,

$$\mathbf{l} = \texttt{Projection}(\hat{\mathbf{l}}), \;\; \mathbf{z}_t = \texttt{Projection}(\hat{\mathbf{z}}_t), \tag{5}$$

where each $\texttt{Projection}(\cdot)$ operator is a linear layer.

**Single-Stream Attention Block**. To integrate the information of language instruction with current observations, we employ a single-stream attention block (Li et al., 2019) that concatenates the multimodal tokens (i.e., $\mathbf{l}$ and $\mathbf{z}_t$), and feeds them into a shared Transformer encoder. Concretely, firstly, we concatenate the language and vision tokens as follows,

$$\mathbf{Z}_t^{\mathrm{c}} = \texttt{cat}(\mathbf{l}, \mathbf{z}_t) =: [\mathbf{l}, \mathbf{z}_t]. \tag{6}$$

Then we apply attention (Vaswani et al., 2017) to refine the concatenated information as follows,

$$\hat{\mathbf{Z}}_t^{\mathrm{c}} = \texttt{Attention}(\mathbf{Z}_t^{\mathrm{c}}), \tag{7}$$

which integrates complementary language and visual information.

**AdaLN and Output of Condition Stream**. `DELE-w0.5` uses AdaLN to integrate conditioning information into the feature representation. After the single-stream attention block, we obtain the queries, keys, and values as outputs of the condition stream:

$$\mathbf{Q}_{\mathrm{c}} = \mathrm{QKNorm}\left\{\mathbf{W}_q(\mathrm{RMSNorm}(\hat{\mathbf{Z}}_t^{\mathrm{c}}) \odot \boldsymbol{\gamma}_{\mathrm{c}})\right\}, \tag{8}$$

$$\mathbf{K}_{\mathrm{c}} = \mathrm{QKNorm}\left\{\mathbf{W}_k(\mathrm{RMSNorm}(\hat{\mathbf{Z}}_t^{\mathrm{c}}) \odot \boldsymbol{\gamma}_{\mathrm{c}})\right\}, \tag{9}$$

$$\mathbf{V}_{\mathrm{c}} = \mathbf{W}_v(\mathrm{RMSNorm}(\hat{\mathbf{Z}}_t^{\mathrm{c}}) \odot \boldsymbol{\gamma}_{\mathrm{c}}), \tag{10}$$

where $\mathrm{RMSNorm}(\cdot)$ is root mean square normalization (Zhang & Sennrich, 2019); $\boldsymbol{\gamma}_{\mathrm{c}}$ is the scaling factor for the condition stream with following linear structure,

$$\boldsymbol{\gamma}_{\mathrm{c}} = 1 + \mathbf{W}_{\mathrm{Adaln}}\mathbf{t}_{\mathrm{c}} + \mathbf{b}_{\mathrm{Adaln}}, \tag{11}$$

$\mathbf{W}_{\mathrm{Adaln}}$ and $\mathbf{b}_{\mathrm{Adaln}}$ are the learnable weights, $\mathbf{t}_{\mathrm{c}}$ is the timestep $\tau = 1$ for condition stream, i.e.,

$$\mathbf{t}_{\mathrm{c}} \overset{(4)}{=} \texttt{Timestep Embedding}(1);$$

$\mathrm{QKNorm}(\cdot)$ is a normalization operator: $\mathbf{x}' = \mathrm{QKNorm}(\mathbf{x}) = \mathbf{x}/\|\mathbf{x}\|_2$.

### 4.2 NOISE STREAM BLOCK

At each time $t$, we corrupt the clean action chunk $\mathbf{A}_t$ and the encoded future state $\mathbf{z}^{\star}_{t+\Delta t} = \texttt{Vision Encoder}(\mathbf{O}_{t+\Delta t})$. We use $\tau = 0$ for noise and $\tau = 1$ for clean targets, equivalent to a noise level $\sigma = 1 - \tau$. For $\tau \sim \mathcal{U}_{[0,1]}$, let $\boldsymbol{\epsilon}_{\mathrm{a}}$ and $\boldsymbol{\epsilon}_{\mathrm{o}}$ be standard Gaussian noise tensors with the respective target shapes. The noisy action and future latent are

$$\mathbf{A}_t^{\tau} = \tau\mathbf{A}_t + (1-\tau)\boldsymbol{\epsilon}_{\mathrm{a}}, \tag{12}$$

$$\mathbf{z}_{t+\Delta t}^{\tau} = \tau\mathbf{z}^{\star}_{t+\Delta t} + (1-\tau)\boldsymbol{\epsilon}_{\mathrm{o}}. \tag{13}$$

Following the same steps in (6), (7), (8), (9), and (10), we obtain the queries, keys, and values as outputs of the noise stream, denoted as $\mathbf{Q}_{\mathrm{n}}$, $\mathbf{K}_{\mathrm{n}}$ and $\mathbf{V}_{\mathrm{n}}$.

### 4.3 FLOW MATCHING OBJECTIVE

We concatenate the information from the condition stream and the noise stream as follows:

$$\mathbf{Q} = \texttt{cat}(\mathbf{Q}_{\mathrm{c}}, \mathbf{Q}_{\mathrm{n}}), \quad \mathbf{K} = \texttt{cat}(\mathbf{K}_{\mathrm{c}}, \mathbf{K}_{\mathrm{n}}), \quad \mathbf{V} = \texttt{cat}(\mathbf{V}_{\mathrm{c}}, \mathbf{V}_{\mathrm{n}}). \tag{14}$$

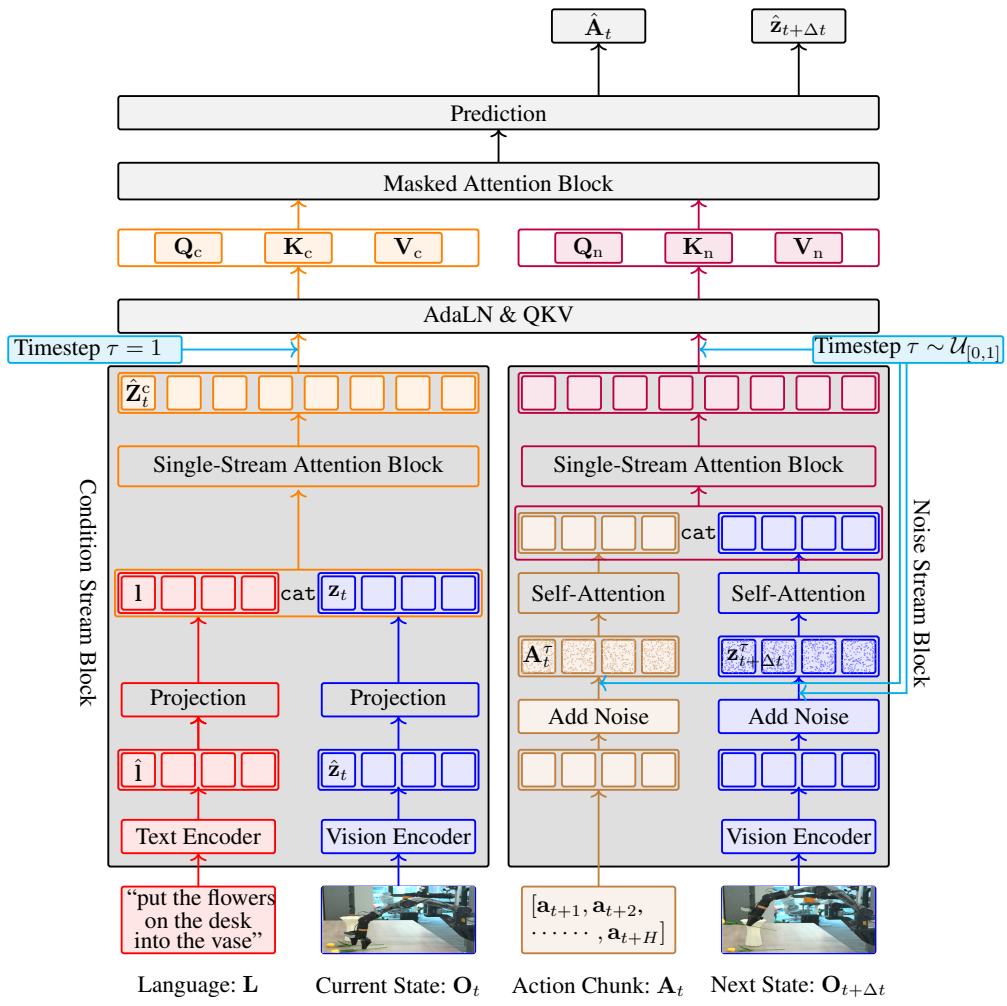



Figure 2: **Overview of DELE-w0.5 Architecture.** The model consists of a condition stream and a noise stream. The condition stream encodes the language instruction $\mathbf{L}$ and current observation $\mathbf{O}_t$, while the noise stream processes the noisy action chunk and future-state latent. The two streams interact through AdaLN-modulated QKV projections and masked attention. The output heads predict clean action and latent targets, which are converted to velocities for the training objective.

The masked attention block and output heads predict the clean action chunk and future latent at noise level $1 - \tau$:

$$\hat{\mathbf{A}}_t^\tau,\ \hat{\mathbf{z}}_{t+\Delta t}^\tau = \texttt{Projection}(\texttt{Masked Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})). \tag{15}$$

Here $\tau$ indexes the noisy input. We convert the clean-sample predictions to velocities using

$$\mathbf{u}_{\mathrm{a}}^\tau = \frac{\hat{\mathbf{A}}_t^\tau - \mathbf{A}_t^\tau}{\max(1-\tau, \delta)}, \qquad \mathbf{u}_{\mathrm{o}}^\tau = \frac{\hat{\mathbf{z}}_{t+\Delta t}^\tau - \mathbf{z}_{t+\Delta t}^\tau}{\max(1-\tau, \delta)}, \qquad \delta = 0.05.$$

The denominator floor stabilizes the conversion near clean inputs. We train the converted velocities against the target fields:

$$\mathcal{L}_{\mathrm{a}} = \mathbb{E}\left[\left\|\mathbf{u}_{\mathrm{a}}^\tau - (\mathbf{A}_t - \boldsymbol{\epsilon}_{\mathrm{a}})\right\|_2^2\right], \tag{16}$$

$$\mathcal{L}_{\mathrm{o}} = \mathbb{E}\left[\left\|\mathbf{u}_{\mathrm{o}}^\tau - (\mathbf{z}_{t+\Delta t}^\star - \boldsymbol{\epsilon}_{\mathrm{o}})\right\|_2^2\right]. \tag{17}$$

where expectations are over training samples, $\tau$, and Gaussian noise. The overall training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{a}} + \lambda \mathcal{L}_{\mathrm{o}}, \tag{18}$$

where $\lambda$ balances action learning and future-state prediction.

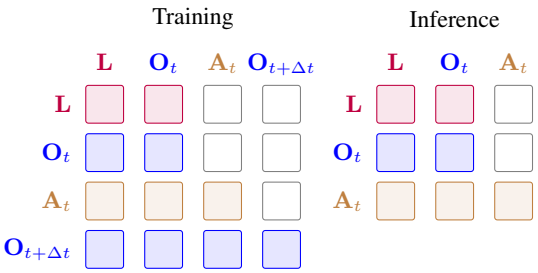



Figure 3: **Attention masks for training and inference.** Rows represent query-token groups; columns represent key/value groups. $\mathbf{L}$, $\mathbf{O}_t$, $\mathbf{A}_t$, and $\mathbf{O}_{t+\Delta t}$ denote language, current state, action, and predicted future state, respectively. Colored cells permit attention; white cells mask it. Inference omits the predicted future observation group. Multi-view tokens are merged into each observation group; padded language positions are masked along both query and key/value dimensions.

### 4.4 Training and Efficient Inference

We apply a block-structured attention mask to language, current observation, action, and future-state tokens at every attention layer that mixes these groups, as shown in Figure 3. Language and current observation tokens attend bidirectionally only within their conditioning group. Action tokens attend to the conditioning and action groups, but not to future-state tokens. Future-state tokens attend to all four groups. At inference time, future-state tokens are omitted without changing the remaining visibility rules, shortening the sequence and avoiding future visual generation. The action and future-state objectives optimize shared parameters during training. Starting from a Gaussian action sample $\mathbf{A}^{(0)}$ at $\tau_0 = 0$, we convert each clean action prediction to $\mathbf{u}_{\mathrm{a}}^{\tau_k}$ and apply Euler updates along an increasing schedule ending at $\tau_K = 1$:

$$\mathbf{A}^{(k+1)} = \mathbf{A}^{(k)} + (\tau_{k+1} - \tau_k)\mathbf{u}_{\mathrm{a}}^{\tau_k}.$$

## 5 Experiments

In this section, we study the following three questions: How reliably does `DELE-w0.5` complete real-world manipulation tasks with different horizons? How does it compare with representative policies under the same fine-tuning data and evaluation protocol? At which stages do the methods fail as task horizon and coordination difficulty increase? We answer these questions through full-task success, normalized stage progress, and task-wise stage-reach analysis.

### 5.1 Experimental Setup

**Robot and Tasks.** We conduct all experiments on an Astribot S1 dual-arm robot in fixed real-world scenes. We consider four tasks with different manipulation horizons: opening a hinged door, retrieving a canned Pepsi from a refrigerator, adding ice to a cup, and loading a popcorn package into a microwave and starting the heating cycle. Figure 4 illustrates the task sequences, and Table 1 gives the ordered stages and conditions for full success.

**Training and Baselines.** `DELE-w0.5` is pretrained on approximately 20,000 hours of multimodal robot-interaction data. Training hyperparameters are summarized in Appendix A. We compare `DELE-w0.5` with seven policies: GWP0.5 (GigaWorld Team et al., 2026), XR0 (Cai et al., 2026), $\pi_{0.5}$ (Physical Intelligence et al., 2025), LingBot-VLA2 (Wu et al., 2026), HY-VLA (Zhang et al., 2026), Spirit-v1.5 (Spirit AI, 2026), and GR00T-N1.7 (NVIDIA et al., 2025). For each real-robot evaluation setting, all methods are trained on the same task-specific sample pool with matched sample exposure. Baselines follow their official initialization and training recipes. At evaluation time, all methods use the same robot embodiment, task instructions, scene-reset configurations, 180-second execution budget, ordered-stage definitions, and complete-success criteria.

**Evaluation.** For each method-task pair, we run twenty formal trials, giving $8 \times 4 \times 20 = 640$ real-robot trials in total. Before each trial, we reset the robot and all task-relevant objects to predefined poses. Each policy has at most 180 seconds to complete the task. A rollout terminated for unsafe execution is counted as a failure. A trial is counted as a full success only when the task-specific complete condition in Table 1 is satisfied without human assistance.

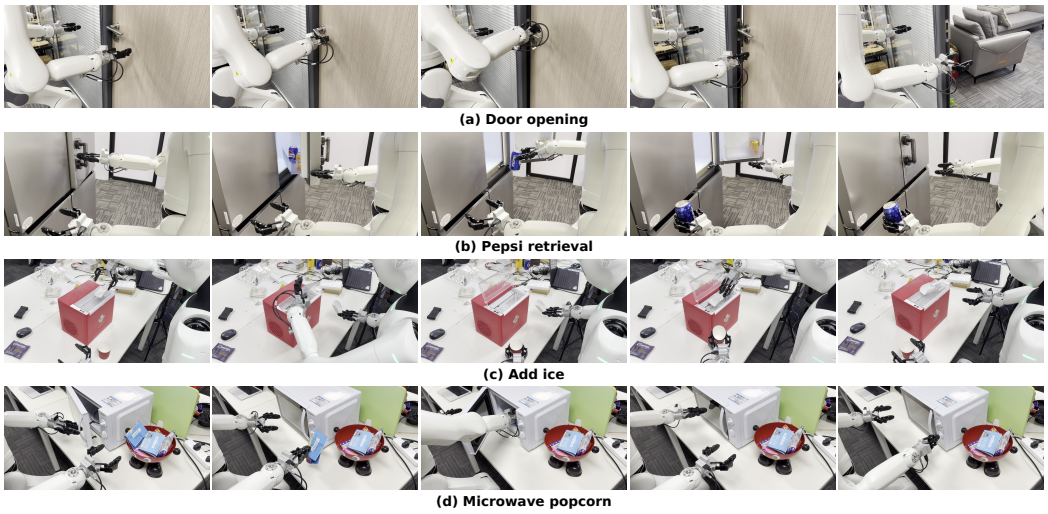



Figure 4: Representative stage sequences for the real tasks.

Table 1: Ordered task stages and complete task conditions. Stage zero denotes no effective progress, and reaching the final stage denotes full task completion.

| **Task** | $K$ | **Ordered stages** | **Complete physical state** |
|---|---|---|---|
| **Door opening** | 4 | Handle contact → stable grasp → coordinated handle rotation and push → sustained opening | Door remains beyond $45^\circ$ for at least 3 seconds. |
| **Pepsi retrieval** | 5 | Handle interaction → door open → can extracted → left-hand handoff → door closed | Left gripper retains the target can while the refrigerator is closed. |
| **Add ice** | 6 | Scoop pickup → lid open → cup pickup → ice acquired → ice poured → scene restored | Ice remains in the upright cup and the manipulated scene is restored. |
| **Microwave popcorn** | 5 | Door open → package pickup → insertion → door closed → heating started | Popcorn is inside the closed microwave and heating visibly starts. |

**Metrics.** We report *full-task success rate*, requiring satisfaction of the complete physical task condition, and *normalized ordered-stage progress* $s/K$, where $s$ counts consecutively completed stages out of $K$. Progress is averaged over twenty trials per task and equally across four tasks. For Add Ice, eight raw stages are consolidated into six semantic stages: raw stages 5–7 map to semantic stage 5, and raw stage 8 to stage 6. Raw trial annotations remain unchanged.

### 5.2 Main Results

Table 2 and Figure 5 compare normalized progress and full-task success. `DELE-w0.5` performs best on all four tasks. It obtains 81.3% macro progress and 62.5% overall success (50/80). GWP0.5 is the strongest baseline in macro progress at 61.3%, whereas XR0 has the highest baseline full-task success at 30.0% (24/80). Using unrounded values and the strongest baseline for each metric, `DELE-w0.5` improves macro progress by 20.1 percentage points and full-task success by 32.5 percentage points. Figure 5(a) shows that `DELE-w0.5` reaches later stages and converts progress into full success more reliably. Its inference is faster than GWP0.5, $\pi_{0.5}$, LingBot-VLA2, and HY-VLA, and comparable to Spirit-v1.5 with substantially higher success. Core-model latency excludes network transfer, image decoding, action projection, IK, trajectory processing, and robot execution. Figures 5(c)–(f) compare success rates and completion times conditioned on success. For Door, Pepsi, Add Ice, and Microwave, `DELE-w0.5` records 16, 13, 9, and 12 successful trials, with mean completion times

Table 2: Real-robot results over twenty trials per task. Each task entry is reported as normalized progress (Prog.) and full-task success (Succ.), in percent.

| | Door | | Pepsi | | Add ice | | Microwave | | Macro | Overall |
|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | Prog. | Succ. | Prog. | Succ. | Prog. | Succ. | Prog. | Succ. | Prog. | Succ. |
| DELE-w0.5 **(ours)** | **95.0** | **80.0** | **82.0** | **65.0** | **63.3** | **45.0** | **85.0** | **60.0** | **81.3** | **62.5** |
| GWP0.5 | 80.0 | 45.0 | 60.0 | 25.0 | 50.0 | 20.0 | 55.0 | 15.0 | 61.3 | 26.3 |
| XR0 | 77.5 | 45.0 | 67.0 | 30.0 | 41.7 | 25.0 | 52.0 | 20.0 | 59.5 | 30.0 |
| $\pi_{0.5}$ | 80.0 | 45.0 | 42.0 | 5.0 | 32.5 | 0.0 | 48.0 | 10.0 | 50.6 | 15.0 |
| LingBot-VLA2 | 60.0 | 20.0 | 50.0 | 0.0 | 25.8 | 0.0 | 48.0 | 0.0 | 46.0 | 5.0 |
| HY-VLA | 57.5 | 0.0 | 35.0 | 0.0 | 37.5 | 10.0 | 32.0 | 15.0 | 40.5 | 6.3 |
| Spirit-v1.5 | 57.5 | 15.0 | 35.0 | 0.0 | 14.2 | 0.0 | 29.0 | 0.0 | 33.9 | 3.8 |
| GR00T-N1.7 | 40.0 | 0.0 | 25.0 | 0.0 | 9.2 | 0.0 | 14.0 | 0.0 | 22.0 | 0.0 |

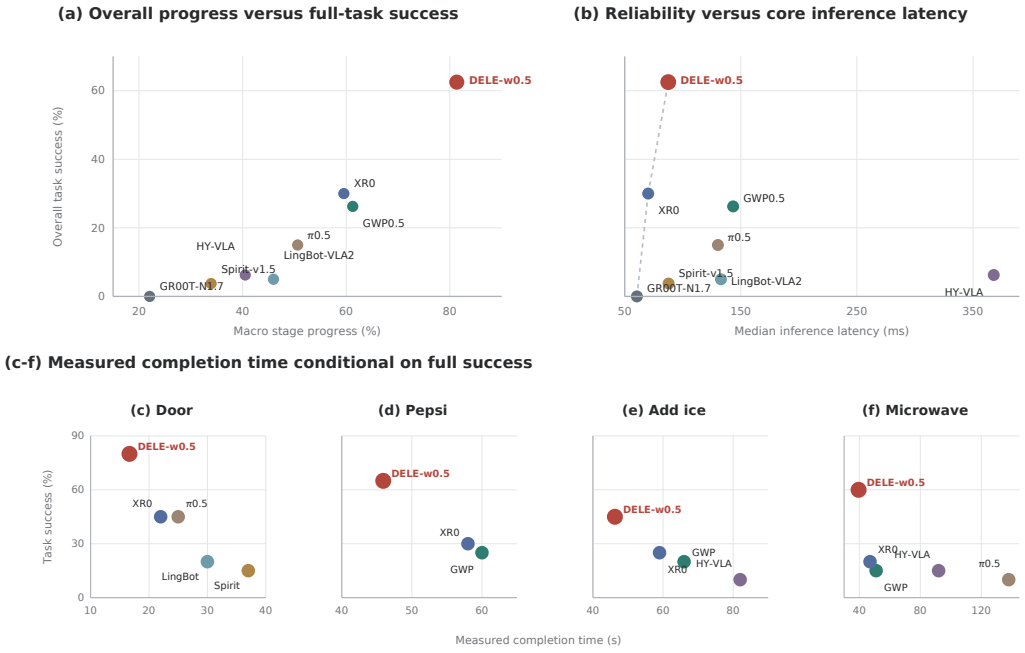



Figure 5: (a) Macro-average ordered-stage progress versus overall full-task success. (b) Overall success versus median core-model inference latency on an NVIDIA RTX 4090. (c–f) Task-level success versus measured completion-time centers conditional on successful completion.

of 16.66 s, 45.92 s, 46.27 s, and 39.50 s, respectively. All methods share the same timing protocol, distinguishing inference latency, successful execution time, and success probability.

### 5.3 Task-wise and Stage-wise Analysis

Figure 6 shows the empirical stage-reach probability $P(s \geq k)$ for each task. Figure 6(a) shows that DELE-w0.5 achieves 16/20 successes and 95.0% progress. GWP0.5 and $\pi_{0.5}$ each reach 80.0% progress; GWP0.5, XR0, and $\pi_{0.5}$ each complete 9/20 trials. Our model reaches coordinated rotation and pushing in all 20 trials, with four failures only at sustained opening. In Figure 6(b), DELE-w0.5 opens the refrigerator and extracts the can in 18/20 trials, achieving 13/20 full successes. XR0 leads the baselines with 67.0% progress and 6/20 successes, followed by GWP0.5 with 60.0% and 5/20. LingBot-VLA2 completes six handoffs but never closes the refrigerator while retaining the can; $\pi_{0.5}$ succeeds once. Figure 6(c) shows that DELE-w0.5 completes the six-stage task in 9/20

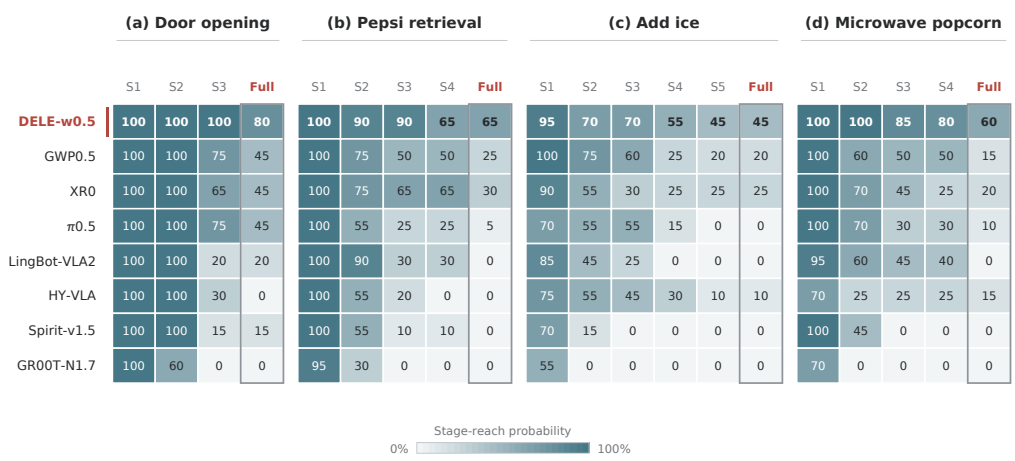



Figure 6: Stage-wise long-horizon analysis. Each cell reports the percentage of twenty trials whose ordered-prefix score reaches or exceeds the indicated stage. Task blocks share a common model axis and probability scale; *Full* represents complete physical task success.

trials. It reaches scoop pickup, cup pickup, ice acquisition, and pouring in 19, 14, 11, and 9 trials, respectively. GWP0.5 leads baseline progress (50.0%), while XR0 leads baseline success (5/20). HY-VLA achieves 37.5% progress and two successes; $\pi_{0.5}$ reaches 32.5% but never places ice into the cup. In Figure 6(d), DELE-w0.5 achieves 12/20 successes and 85.0% progress. It opens the microwave and picks up the package in all 20 trials, closes the door in 16, and starts heating in 12. GWP0.5 and XR0 achieve 55.0% and 52.0% progress, with 3/20 and 4/20 successes, respectively. $\pi_{0.5}$, Spirit-v1.5, and LingBot-VLA2 open the microwave in 20, 20, and 19 trials, but complete only 2/20, 0/20, and 0/20, respectively.

### 5.4 Early signs of emergence.

In additional intervention rollouts, we observe adaptive behaviors at intermediate states absent from the task demonstrations. When the microwave door is closed during execution, DELE-w0.5 retains the popcorn package, reopens the door, and continues toward the original task objective. When the available opening becomes narrow, the policy instead uses the package already in hand to push the door open, rather than returning to the demonstrated handle-opening routine. These behaviors exhibit goal-conditioned behavioral recomposition: the policy preserves the intended outcome while adapting its interaction strategy to an unseen task state. We view these observations as early evidence rather than a quantitative claim of emergence.

## 6 Conclusion

We show that effective robotic manipulation can be learned without video generation. We propose DELE-w0.5, a dual-stream architecture that unifies action generation and compact future-state prediction. This formulation focuses on action-relevant physical outcomes rather than dense visual transitions, enabling efficient inference. Finally, an evaluation totaling 640 real-robot trials on four tasks shows that DELE-w0.5 achieves 62.5% full-task success and 81.3% macro ordered-stage progress. These results exceed the strongest baseline for each metric by 32.5 and 20.1 percentage points, respectively. Our findings support compact future-state prediction as a practical alternative to video generation for world-action modeling.

### AI USE STATEMENT

In this work, we used generative AI tools to assist with literature search and summarization, manuscript organization and language editing, figure and table design and formatting, and discussions of experimental methodology and result interpretation. We did not use generative AI tools to generate experimental data or conduct robot experiments. All AI-assisted material was reviewed and verified by the authors, who take responsibility for the final manuscript, reported results, and conclusions.

## REFERENCES

Arman Akbari, Ci Zhang, Arash Akbari, Lin Zhao, Yixiao Chen, Weiwei Chen, Xuan Zhang, Geng Yuan, and Yanzhi Wang. Flash-wam: Modality-aware distillation for world action models, 2026. URL https://arxiv.org/abs/2606.05254.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-jepa 2: Self-supervised video models enable understanding, prediction and planning, 2025. URL https://arxiv.org/abs/2506.09985.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael 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, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. $\pi_0$: A vision-language-action flow model for general robot control, 2026. URL https://arxiv.org/abs/2410.24164.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie 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, 2024. URL https://arxiv.org/abs/2402.15391.

Rui Cai, Jun Guo, Xinze He, Piaopiao Jin, Jie Li, Bingxuan Lin, Futeng Liu, Wei Liu, Fei Ma, Kun Ma, Feng Qiu, Heng Qu, Yifei Su, Qiao Sun, Dong Wang, Donghao Wang, Yunhong Wang, Rujie Wu, Diyun Xiang, Yu Yang, Hangjun Ye, Yuan Zhang, and Quanyun Zhou. Xiaomi-robotics-0: An open-sourced vision-language-action model with real-time execution, 2026. URL https://arxiv.org/abs/2602.12684.

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, 2024. URL https://arxiv.org/abs/2410.06158.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. PaLM-e: An embodied multimodal language model. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 8469–8488. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/driess23a.html.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 9156–9172. Curran Associates, Inc., 2023. doi: 10.52202/075280-0403. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1d5b9233ad716a43be5c0d3023cb82d0-Paper-Conference.pdf.

GigaWorld Team, Angen Ye, Angyuan Ma, Boyuan Wang, Chaojun Ni, Fangzheng Ye, Guan Huang, Guo Li, Guosheng Zhao, Haodong Yan, Hengtao Li, Jiwen Lu, Kai Wang, Mingming Yu, Qitang Hu, Qiuping Deng, Songling Liu, Xiaoyu Tian, Xiaofeng Wang, Xinyu Zhou, Xiuwei Xu, Xinze Chen, Yang Wang, Yejun Zeng, Yifan Chang, Yun Ye, Zhenyu Wu, Zhanqian Wu, and Zheng Zhu. Gigaworld-policy-0.5: A faster and stronger wam empowered by autoresearch, 2026. URL https://arxiv.org/abs/2607.13960.

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, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning, 2026. URL https://arxiv.org/abs/2601.16163.

Myungkyu Koo, Daewon Choi, Taeyoung Kim, Kyungmin Lee, Changyeon Kim, Younggyo Seo, and Jinwoo Shin. HAMLET: Switch your vision-language-action model into a history-aware policy. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=KcJ9U0x6kO.

Gaofeng Li, Ruize Wang, Peisen Xu, Qi Ye, and Jiming Chen. The developments and challenges toward dexterous and embodied robotic manipulation: A survey. *IEEE Robotics & Automation Magazine*, 2025. URL https://ieeexplore.ieee.org/document/11317793/.

Jiajun Li, Tiecheng Guo, Yifan Ye, Rongyu Zhang, Xiaowei Chi, Qianpu Sun, Ying Li, Yunfan Lou, Yan Huang, Zhihe Lu, Meng Guo, and Shanghang Zhang. Efficient-wam: A 1b-parameter world-action model with low-cost future imagination, 2026a. URL https://arxiv.org/abs/2606.10040.

Kehan Li, Bohan Hou, Minghao Zhu, Tianyi Zhang, Zesen Cheng, Zhikai Wang, Sicong Leng, Xin Li, Xiao Lin, Biying Yao, et al. Rynnbrain 1.1: Towards more capable and generalizable embodied foundation model. *arXiv preprint arXiv:2607.17977*, 2026b. URL https://arxiv.org/pdf/2607.17977.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language, 2019. URL https://arxiv.org/abs/1908.03557.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, Xiaofan Wang, Bei Liu, Jianlong Fu, Jianmin Bao, Dong Chen, Yuanchun Shi, Jiaolong Yang, and Baining Guo. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation, 2024. URL https://arxiv.org/abs/2411.19650.

Ziang Li, Dongzhou Cheng, Yibin Wang, Shiyue Wang, Xiaoyang Xu, Lingxuan Weng, Juan Wang, and Jiaqi Wang. Light-wam: Efficient world action models with state-fusion action decoding, 2026c. URL https://arxiv.org/abs/2606.08242.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1b: a diffusion foundation model for bimanual manipulation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=yAzN4tz7oI.

Oier Mees, Dibya Ghosh, Karl Pertsch, Kevin Black, Homer Rich Walke, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*, 2024. URL https://openreview.net/forum?id=jGrtIvJBpS.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang

Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. Gr00t n1: An open foundation model for generalist humanoid robots, 2025. URL https://arxiv.org/abs/2503.14734.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, et al. Open X-Embodiment: Robotic learning datasets and RT-X models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903, 2024. URL https://doi.org/10.1109/ICRA57147.2024.10611477.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models, 2025. URL https://arxiv.org/abs/2501.09747.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael 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, 2025. URL https://arxiv.org/abs/2504.16054.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. Smolvla: A vision-language-action model for affordable and efficient robotics, 2025. URL https://arxiv.org/abs/2506.01844.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3, 2025. URL https://arxiv.org/abs/2508.10104.

Spirit AI. Spirit-v1.5: Clean data is the enemy of great robot foundation models. Spirit AI Blog, 2026. URL https://www.spirit-ai.com/en/blog/spirit-v1-5.

Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=meRCKuUpmc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Qiuyue Wang, Mingsheng Li, Jian Guan, Jinhui Ye, Sicheng Xie, Yitao Liu, Junhao Chen, Zhixuan Liang, Jie Zhang, Xintong Hu, Xuhong Huang, Pei Lin, Junyang Lin, Dayiheng Liu, Shuai Bai, Jingren Zhou, Jiazhao Zhang, Haoqi Yuan, Gengze Zhou, Hang Yin, Ye Wang, Yiyang Huang, Zixing Lei, Wujian Peng, Delin Chen, Yingming Zheng, Jingyang Fan, Xianwei Zhuang, Xin Zhou, Haoyang Li, Anzhe Chen, Tong Zhang, Xuejing Liu, Yuchong Sun, Ruizhe Chen, Zhaohai Li, Chenxu Lü, Zhibo Yang, Tao Yu, and Xionghui Chen. Qwen-vla: Unifying vision-language-action modeling across tasks, environments, and robot embodiments, 2026. URL https://arxiv.org/abs/2605.30280.

Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li, Yuntao Chen, Xinlong Wang, and Zhaoxiang Zhang. Unified vision-language-action model, 2025. URL https://arxiv.org/abs/2506.19850.

Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Zhibin Tang, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, et al. Tinyvla: toward fast, data-efficient vision-language-action models for robotic manipulation. *IEEE Robotics and Automation Letters*, 10(4):3988–3995, 2025. URL https://ieeexplore.ieee.org/abstract/document/10900471.

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, 2023. URL https://arxiv.org/abs/2312.13139.

Wei Wu, Fangjing Wang, Fan Lu, He Sun, Shi Liu, Yunnan Wang, Yibin Yan, Yong Wang, Shuailei Ma, Xinyang Wang, Yibin Liu, Shuai Yang, Tianxiang Zhou, Kejia Zhang, Lei Zhou, Cheng Su, Nan Xue, Bin Tan, Han Zhang, Youchao Zhang, Fei Liao, Xing Zhu, Yujun Shen, and Kecheng Zheng. From foundation to application: Improving vla models in practice, 2026. URL https://arxiv.org/abs/2607.06403.

Hongyu Yan, Qiwei Li, Jiaolong Yang, and Yadong Mu. Progressvla: Progress-guided diffusion policy for vision-language robotic manipulation, 2026. URL https://arxiv.org/abs/2603.27670.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, Min Cao, Peng Li, Qiuping Deng, Wenjun Mei, Xiaofeng Wang, Xinze Chen, Xinyu Zhou, Yang Wang, Yifan Chang, Yifan Li, Yukun Zhou, Yun Ye, Zhichao Liu, and Zheng Zhu. Gigaworld-policy: An efficient action-centered world–action model, 2026a. URL https://arxiv.org/abs/2603.17240.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World action models are zero-shot policies, 2026b. URL https://arxiv.org/abs/2602.15922.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/1e8a19426224ca89e83cef47f1e7f53b-Paper.pdf.

He Zhang, Lingzhu Xiang, Haitao Lin, Zeyu Huang, Minghui Wang, Dingyan Zhong, Yubo Dong, Yihao Wu, Yongming Rao, Dongsheng Zhang, Wanjia He, Ling Chen, Kai Huang, Jiahao Chen, Sichang Su, Xumin Yu, Ziyi Wang, Chengwei Zhu, Xiao Teng, Yuchun Guo, Yufeng Zhang, Yuandong Liu, Rui Wang, Zisheng Lu, Han Hu, and Zhengyou Zhang. Hy-embodied-0.5-vla: From vision-language-action models to a real-world robot learning stack, 2026. URL https://arxiv.org/abs/2606.14409.

Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model, 2024. URL https://arxiv.org/abs/2403.09631.

Pengfei Zhou, Shengcong Chen, Di Chen, Jiaxu Wang, Rongjun Jin, Bingwen Zhu, Yike Pan, Songen Gu, Kuanning Wang, Shufeng Nan, Xingyu Qiu, Chenhao Qiu, Pu Yang, Yunuo Cai, Jianxiong Gao, Yifan Li, Yanwei Fu, Xiangyu Yue, Zhi Chen, and Jianlan Luo. $\tau_0$-wm: A unified video-action world model for robotic manipulation, 2026. URL https://arxiv.org/abs/2606.01027.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: A fine-grained world model for robot manipulation, 2025. URL https://arxiv.org/abs/2406.14540.

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 Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229, pp. 2165–2183, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

# A Model and Training Details

**Model configuration.** Table 3 reports the parameter counts of the trainable policy and frozen feature encoders. The policy count includes input projections and prediction heads. The encoder checkpoints are Qwen/Qwen3-0.6B and facebook/dinov3-vitl16-pretrain-lvd1689m.

Table 3: Model components and parameter counts of `DELE-w0.5`.

| Component | Configuration | Trainable | Parameters |
|---|---|---|---|
| Policy network | DELE-w0.5 | Yes | 2.17B |
| Text encoder | Qwen3-0.6B | No | 596.0M |
| Vision encoder | DINOv3 ViT-L/16 | No | 303.1M |
| **Full model** | – | – | **3.07B** |

Parameter counts are rounded for readability. The total includes both frozen encoders, each counted once.

**Training configuration.** We randomly initialize the trainable backbone for pretraining and initialize fine-tuning from the pretrained checkpoint. The text and vision encoders remain frozen during both stages. Table 4 summarizes the shared training configuration. Learning-rate warmup spans the first 10% of optimization steps in each stage and is followed by cosine decay.

Table 4: Shared pretraining and fine-tuning hyperparameters of `DELE-w0.5`.

| Parameter | Value |
|---|---|
| Optimizer | Muon |
| Global batch size | 1,024 |
| Peak learning rate | $2 \times 10^{-4}$ |
| Learning-rate schedule | Cosine decay |
| Warmup ratio | 0.1 |
| Training precision | BF16 |

# B Task Definitions and Evaluation Protocol

We evaluate complete physical outcomes on four fixed-scene tasks. Each of the eight policies is evaluated for twenty trials per task. Before each trial, the robot returns to the shared ready pose with both grippers open, and all task-relevant objects return to their predefined positions. The robot base, camera poses, lighting, and background remain fixed. Each rollout has a 180-second execution budget. Manual assistance, an emergency stop, or termination for unsafe execution precludes full-task success. Setup checks and pilot runs are excluded from the 640 formal trials.

**Ordered progress and full success.** A stage is credited only after all preceding stages have been completed. Stage zero denotes failure to complete the first stage; incidental contact or a later action does not substitute for a missing prerequisite. Let $s_{m,t,i}$ denote the ordered-prefix score and $y_{m,t,i}$ the full-success indicator for method $m$, task $t$, and trial $i$. With $N = 20$ trials and task maximum $K_t$, the reported task metrics are

$$\mathrm{Prog}_{m,t} = \frac{100}{N} \sum_{i=1}^{N} \frac{s_{m,t,i}}{K_t}, \qquad \mathrm{Succ}_{m,t} = \frac{100}{N} \sum_{i=1}^{N} y_{m,t,i}. \tag{19}$$

Full success requires the complete physical condition specified below, without assistance or unsafe termination. The two metrics are averaged equally across tasks. Stage-reach probability is $\widehat{P}_{m,t}(s \geq k) = N^{-1} \sum_i \mathbf{1}[s_{m,t,i} \geq k]$. The task instructions below are given in English translation.

### B.1 DOOR OPENING

**Instruction and reset.** *Open the door.* The ordinary hinged door begins fully closed and latched. The handle and the marked opening threshold are unchanged across trials, and the door sweep is unobstructed.

| Stage | Observable completion condition |
|---|---|
| 1 | Intentional contact with the correct door handle. |
| 2 | A stable grasp of the handle. |
| 3 | Coordinated handle rotation and pushing release the latch and open the door. |
| 4 | The door crosses the marked $45^\circ$ threshold and remains beyond it for at least three seconds. |

**Interaction requirement.** Grasping the handle is insufficient: the robot must rotate it far enough to release the latch while applying a suitably timed push. A push before latch release does not open the door, while releasing the handle too early can interrupt the transition. Stage 3 therefore captures coordinated rotation and pushing; stage 4 additionally requires sustained opening. Incidental contact or opening through an unsafe collision does not satisfy the rubric.

### B.2 PEPSI RETRIEVAL

**Instruction and reset.** *Take the canned Pepsi out of the refrigerator.* The refrigerator starts fully closed, and the target can is placed at the same marked shelf pose. Shelf contents and distractors remain unchanged.

| Stage | Observable completion condition |
|---|---|
| 1 | Intentional interaction with the correct refrigerator handle. |
| 2 | The refrigerator door is opened. |
| 3 | The target can is fully extracted and stably held outside the refrigerator. |
| 4 | The left gripper receives and stably holds the target can. |
| 5 | The refrigerator is fully closed while the left gripper retains the can; both conditions hold for at least three seconds. |

**Interaction requirement.** The evaluation requires extraction, a left-hand handoff, and refrigerator closure as one continuous sequence. Retrieving a different can, knocking the target out, or dropping it outside the refrigerator does not count as successful extraction. Handoff alone is partial progress: full success also requires closing the refrigerator without losing the transferred can.

### B.3 ADDING ICE

**Instruction and reset.** *Add ice to the cup.* The ice-maker lid starts closed, the scoop rests at its marked pose, and the empty cup stands upright at its marked table position. The usable ice supply is restored, and spilled ice and water are removed before each trial.

| Stage | Observable completion condition |
|---|---|
| 1 | The right gripper grasps and lifts the scoop from its rest pose. |
| 2 | The left arm opens the lid sufficiently for the scoop to enter. |
| 3 | The left gripper lifts and holds the target cup upright. |
| 4 | The right arm scoops and lifts at least one complete ice piece. |
| 5 | Ice is poured into the left-held cup, with at least one complete piece remaining inside for at least three seconds. |
| 6 | The cup is returned upright with ice retained, the lid is fully closed, and the scoop is returned and released at its rest pose. |

**Stage consolidation.** The raw annotation uses eight ordered stages, separating cup return, lid closure, and scoop return. For reporting, raw scores $0, 1, 2, 3, 4$ retain their values, raw scores $5, 6, 7$ map to semantic score 5, and raw score 8 maps to semantic score 6. Thus, intermediate restoration actions do not earn extra semantic stages, and full completion still requires all three restoration actions. This mapping leaves the raw annotations and full-success outcomes unchanged.

**Interaction requirement.** Ice must be acquired with the scoop and poured into a cup held by the left gripper. Directly moving ice by hand or pouring into a cup left on the table does not satisfy the sequence. A rim hit, bounce-out, or spill without a retained ice piece does not satisfy stage 5. The final cup must remain upright with ice inside for at least three seconds after placement.

### B.4 MICROWAVE POPCORN

**Instruction and reset.** *First open the microwave door, then put the popcorn inside and start heating.* The door begins fully closed with heating stopped. The popcorn package returns to the same marked position and orientation before each trial.

| Stage | Observable completion condition |
|---|---|
| 1 | The microwave door is fully opened. |
| 2 | The popcorn package is lifted and retained after door opening. |
| 3 | The package is inserted and released inside the microwave. |
| 4 | The microwave door is fully closed after insertion. |
| 5 | Heating visibly starts with the package inside and the door closed. |

**Interaction requirement.** The task combines articulated opening, object placement through the opening, closure, and heating activation. Placing the package inside is not full success. The terminal condition is the visible start of heating; the rollout does not need to wait for the cooking cycle to finish.

## C Inference Latency and Task Completion Time

Figure 5 separates policy prediction cost from physical task duration. These quantities answer different questions and are evaluated alongside full-task success.

**Core-model inference latency.** Latency is reported in milliseconds as the median of repeated core-model prediction measurements on an NVIDIA RTX 4090. It excludes network transfer, image decoding, action projection, inverse kinematics, trajectory processing, and robot execution. A prediction produces an action chunk rather than a single physical control step, so model latency is not an end-to-end task duration or a robot control period. LingBot-VLA2 uses its official compiled inference path; the reported 132.9 ms is a median over repeated measurements.

**Successful-task completion time.** We measure completion time in seconds from the first executed robot action until the full task-specific success criteria are satisfied, including any required terminal-state hold. For each method-task pair, we report the mean over successful trials with recorded completion times. Failed trials are excluded from this average but remain included in the success-rate calculation. For `DELE-w0.5`, the means use 16, 13, 9, and 12 successful trials for Door, Pepsi, Add Ice, and Microwave, respectively. Figure 5(c)–(f) includes only method-task pairs with available timing measurements; an omitted point does not necessarily indicate zero task success.

**Reading the efficiency comparison.** The inference-latency panel describes the cost of one policy prediction, while the task-runtime panels describe successful physical execution. Success probability provides the complementary measure of how often the complete objective is achieved. Keeping these quantities separate avoids equating fast prediction, fast successful execution, and reliable task completion.

## D Adaptive Execution under Task Perturbations

The additional intervention rollouts described in Section 5.4 examine how the policy responds when access to the microwave changes during execution. They are separate from the 640 formal evaluation trials. In these rollouts, the external intervention changes the door state; the subsequent manipulation is performed by the policy.

**Reopening while retaining the package.** When the microwave door is closed during an ongoing interaction, the robot retains the package, reopens the door, and continues toward the original task objective. The relevant observation is the combination of object retention and renewed access: the robot adjusts its interaction with the articulated door without first returning to the initial empty-handed state.

**Using the held package to restore access.** When the opening becomes narrow, the robot uses the package already in hand to push the door farther open. This differs from returning to the demonstrated handle-opening routine. The held object participates in changing the scene so that the ongoing manipulation can continue.

**Behavioral interpretation.** Both observations preserve the task objective while changing the immediate interaction strategy. They illustrate state-dependent recomposition of object holding and door manipulation, rather than an invariant replay of the initial action sequence. These are qualitative observations of adaptive execution, not an additional success-rate benchmark or a quantitative demonstration of emergence.