Inferring Action from Future Latent State for Robotic Manipulation

Fenghao Lei Zhixiong Huang Long Yang Jiabao Chen Jie Cheng Peilin Huang Han Fu, Zhuo Li Xiaoxue Ren

[1]DeepLeap Research, Shenzhen, China
Email: yanglong0908@gmail.com

ABSTRACT

World-Action Models (WAMs) build robot control on video-generation backbones, which jointly predict dense future visual trajectories and robot actions. We argue that video generation is an unnecessary intermediate objective for world-action modeling. For robotic manipulation, the goal of a world model is not to reproduce how the world looks at every intermediate moment, but to predict the state that the world will reach after an action is executed. The intermediate frames only describe the visual transition between physical states, which consumes substantial model capacity and computation, but do not directly specify the physical outcome that the robot action is intended to produce. In this paper, we propose `DELE-w0.5`, which infers robot actions from predicted future states without relying on video generation. Concretely, `DELE-w0.5` infers the action sequence from its corresponding compact future latent state. The future latent state captures the action-relevant physical outcome of robot interaction and serves as an explicit bridge between world modeling and action generation. The core design principle of `DELE-w0.5` is to model how the physical world changes under robot actions, rather than how its visual appearance evolves frame by frame. This formulation removes the high-dimensional visual redundancy introduced by dense video representations, and it therefore enables cheaper training and low-latency inference. Across 480 real-robot trials on four long-horizon manipulation tasks, our `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, outperforming the strongest baseline by 47.5 and 30.7 percentage points, respectively.

Keywords embodied intelligence, vision-language-action models, world model **Project** Research page

Contents

1 Introduction

Embodied intelligence has achieved rapid development and achieved strong performance for robotic manipulation [16, 18]. Vision-Language-Action (VLA) and World Action Model (WAM) are two popular model architectures. VLA is a three-step framework, see Figure 1 (a): First, the input comprises multimodal data, including human task instructions and the robot's current observations; Then, a vision-language model (VLM) backbone performs the task planning according to the input data; Finally, an action expert decodes the task planning to the commands that are executable by the robot. For some classic work of VLA, refer to [4, 11, 12, 53]. Since the information density of language is lower than that of vision, VLA models require a large amount of data to adapt to the downstream tasks. As a result, generalization of VLA is poor, when light, color, or the task changes, the success rate decays rapidly, extensive empirical results have shown this view, see [14, 44]. WAM [9, 14, 44] fundamentally discards the framework of VLA, WAM's core lying in the construction of a unified end-to-end learning objective, which integrates the prediction of world state evolution and action generation, see Figure 1 (b). The critical technology of WAM relies on self-supervised future state prediction on large-scale video data, then WAM develops the action generation as a conditional denoising process according to the future state prediction.

1.1 Limitation of WAM

A dominant line of recent World-Action Models (WAMs) builds robot control on top of video generation and jointly predicts future visual trajectories and actions [44, 45], which brings some useful spatiotemporal priors into the policy. However, it also creates a structural mismatch between the objective of visual generation and the objective of robot control. Future videos are high-dimensional and appearance-sensitive. Most intermediate frames describe how the scene looks during a transition, rather than the physical consequence that determines the subsequent action. Therefore, the model is required to solve a substantially harder problem than control itself. It must first reconstruct dense visual evolution and then extract a compact action signal from it. Additionally, multi-frame video latents are much larger than the corresponding robot action sequence [48]. They dominate the sequence length, memory consumption, and computational budget of joint world-action learning. Consequently, WAM training inherits the heavy cost of large video generators. Recent methods introduce compact backbones, latent downsampling, or lightweight adaptation to make video-action co-training more practical [21]. These techniques reduce the cost of the existing pipeline, but do not remove its underlying redundancy. The same limitation becomes more critical during inference. Before an action can be executed, the model must perform iterative denoising over high-dimensional future video-action latents [1]. The control loop is therefore bottlenecked by synthesizing a visual trajectory that is not itself executed by the robot [17].

Therefore, the fundamental limitation of video-generation-based WAMs is not merely their computational cost. The deeper problem is that they optimize visual trajectory reconstruction as a surrogate for action-relevant world prediction. For robot control, a world model should predict how the world will become after an interaction, rather than reproduce how the world looks at every intermediate moment. This motivates a more direct formulation that models action-relevant future states and infers robot actions without relying on video generation in the control loop.

1.2 Our Work

We reformulate the current WAM of visual world modeling for robotic manipulation. It is different from recent video generation techinique based WAMs that model the future world through synthesizing dense visual trajectories, the proposed `DELE-w0.5` argues that robot control does not require reconstructing the entire visual evolution of the real world, but only requires to predict the future latent state with downstream actions. With this observation, we formulate WAM as prediction problem of a future latent state and action, where the `DELE-w0.5` directly infers the future visual state and corresponding robot behaviors, without any generating intermediate video frames.

Rethinking World Model for Robotic Manipulation. We argue that existing video-generation-based WAMs formulate unnecessary intermediate objectives for robotic manipulation. Although predicting future videos provides rich spatiotemporal priors, it implicitly treats visual evolution as the primary target of

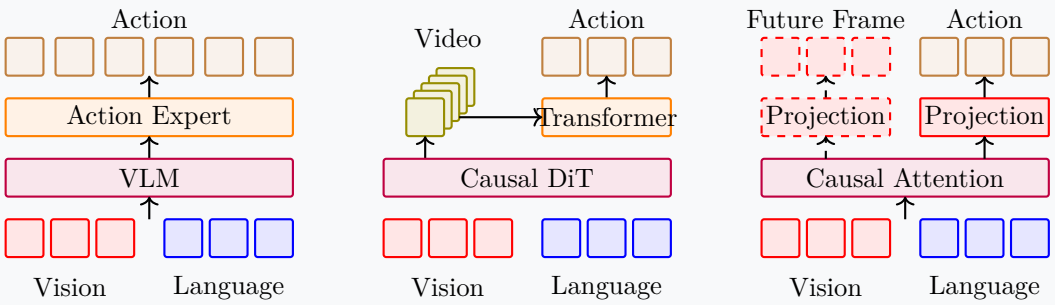

Figure 1: Different Models for Manipulation.

world modeling. However, for robot robotic manipulation, a world model should not reproduce how the world looks at every intermediate moment. Instead, it should predict how the world will become after an action is executed. Most intermediate frames in generated videos only describe the visual transition between states, which provides limited information about the final physical consequence that determines subsequent actions. Therefore, video-generation-based WAMs force the model to learn the distribution of visual evolution rather than directly modeling action-relevant future states. This design introduces substantial computational overhead without necessarily improving control capability. We argue that future-state prediction with causal relevance to robot actions is a more fundamental objective for embodied world models than reconstructing complete visual trajectories.

Inferring Action from Future-State. With this observation, we propose `DELE-w0.5` that infers robot actions from predicted future states rather than generating future visual trajectories. Different from existing video-generation-based approaches that model the continuous evolution of the visual world, `DELE-w0.5` treats future prediction as a state transition problem and focuses on the physical states that are causally relevant to subsequent actions. The key insight is that a world model for robotic manipulation does not need to reproduce how the world visually evolves at every intermediate moment, but should capture how the world will become after an action is executed. By directly modeling action-relevant future states, `DELE-w0.5` avoids the high-dimensional visual redundancy introduced by video representations, and eliminates the costly iterative generation process required by video-based WAMs. This formulation provides a more compact and efficient paradigm for world-action modeling, enabling significantly cheaper training and low-latency inference for real-world robotic deployment. Finally, we comparse the three formulation from probabilistic inference as follows,

- VLA infers $p_\theta(\mathbf{A}_t|\mathbf{O}_t, \mathbf{q}_t, \mathbf{L})$;
- WAM infers $p_\theta(\mathbf{A}_t, \mathbf{O}_{t:t+\Delta t}|\mathbf{O}_t, \mathbf{q}_t, \mathbf{L})$;
- `DELE-w0.5` infers $p_\theta(\mathbf{A}_t, \mathbf{O}_{t+\Delta t}|\mathbf{O}_t, \mathbf{q}_t, \mathbf{L})$,

where $\mathbf{O}_{t:t+\Delta t}$ is the video from current observation $\mathbf{O}_t$ to the predicted future state $\mathbf{O}_{t+\Delta t}$.

2 Related Work

2.1 Vision-Language-Action Models

VLA models connect the semantic knowledge of pretrained vision-language models with low-level robot control. PaLM-E [8] injects continuous visual and state observations into a large language model for embodied reasoning. RT-2 [53] further casts robot actions as language-like tokens and co-trains web-scale vision-language tasks with robot trajectories. Open X-Embodiment and RT-X [26] show that heterogeneous trajectories from many robot platforms can support positive cross-embodiment transfer . Octo [24] provides an open generalist policy that supports different observations, action spaces, and robot embodiments. OpenVLA [12] adapts a pretrained VLM to autoregressive action prediction and makes large-scale VLA training accessible. TinyVLA reduces model size and data requirements, while SmolVLA further targets training and deployment on affordable hardware [30, 38]. RDT-1B [23] uses a diffusion transformer for multi-modal bimanual actions. CogACT [20]

couples a VLM with a diffusion action module. π_0 [3] instead uses flow matching to generate continuous action chunks. DexVLA [37] adopt dual-system designs in which a vision-language component provides semantic features and a diffusion expert produces motor commands. FAST [27] compresses high-frequency action sequences into a shorter autoregressive vocabulary. OpenVLA-OFT [13] uses parallel action decoding, continuous actions, and action chunking to improve both control rate and downstream success. UniVLA [36] learns task-centric latent actions from heterogeneous videos and decodes them for different embodiments. $\pi_{0.5}$ [11] combines robot data, web data, semantic prediction, and heterogeneous supervision for open-world manipulation. HAMLET introduces compact historical memory, while ProgressVLA explicitly estimates task progress for long-horizon control [15, 41]. Qwen-VLA further unifies manipulation, navigation, and trajectory prediction across tasks and robot embodiments [35].

VLA still learns a direct conditional mapping from the current context to an action sequence, which is effective for behavior cloning, but it does not require the policy to represent the physical consequence of its action. The cost of large VLM backbones and iterative generative action heads also complicates real-time deployment. In contrast, `DELE-w0.5` proposes a predicted future state as an explicit interface between perception and control. It first estimates the action-relevant physical outcome and then infers the robot action from this state.

2.2 World-Action Models

World models learn predictive structure that can support control, planning, or policy learning, and recent studies show that future prediction can provide useful structure beyond a purely reactive policy. Latent world models such as DreamerV3 [10] improve a policy through imagined state trajectories. Multi-view Masked World Models [29] learn predictive representations from multiple cameras for visual manipulation. UniPi [9] formulates sequential decision making as text-conditioned video generation and recovers actions from generated plans. Genie [5] learns action-controllable interactive environments from unlabeled videos through latent actions. Recent robotic world models connect future prediction more tightly with action generation. GR-1 [39] jointly predicts future images and robot actions after large-scale video pretraining. GR-2 [7] scales this paradigm with web videos and robot trajectories. 3D-VLA [50] predicts goal images and point clouds to connect 3D reasoning with action planning. IRASim generates action-conditioned videos with frame-level action alignment for fine-grained robot interactions [52]. Seer closes the loop through a predictive inverse-dynamics formulation, in which actions are inferred from forecast visual states [33]. CoT-VLA [49] predicts future visual goals as an explicit visual chain of thought before producing actions. WorldVLA unifies image and action generation in one autoregressive model [6]. V-JEPA 2-AC [2] instead predicts future states in a learned representation space and uses them for planning. Genie Envisioner builds policy learning and neural simulation on a shared video diffusion foundation [22]. DreamZero [44] builds a large WAM on a pretrained video diffusion backbone and demonstrates strong physical generalization. τ_0-WM [51] combines video-action prediction, action-conditioned simulation, and candidate evaluation in one framework. GigaWorld-Policy [43] makes future-video generation optional at deployment and places the action stream before the video stream. Flash-WAM [1] distills iterative video-action diffusion into very few sampling steps. Efficient-WAM [17] uses a compact video expert, sparse video tokens, and asymmetric denoising. AGRA [28] shows that visually plausible futures do not always yield accurate actions and aligns video features with action-relevant regions.

Video-generation-based WAMs couple robot control to a substantially harder surrogate problem. A robot ultimately executes a low-dimensional action sequence, but the model must first process or synthesize dense multi-frame latents. Most intermediate frames describe visual transition. They do not directly specify the final physical state that determines the next action. This mismatch increases sequence length, memory use, and training cost. During inference, iterative denoising over future video and action latents introduces latency into the control loop. More importantly, features optimized for visual reconstruction are not necessarily organized for low-level control, as recent representation analyses have shown [28]. Distillation, token sparsification, and smaller video backbones reduce this cost, but they retain video synthesis as the underlying objective. `DELE-w0.5` instead formulates WAM as future-state and action prediction. It predicts a compact future state that is causally relevant to manipulation and infers the corresponding action from this state. The model does not rely on any video generation technique and does not reconstruct intermediate frames. It therefore

avoids the high-dimensional visual redundancy of video WAMs and provides a more direct, cheaper, and lower-latency route from world prediction to robot control.

3 Preliminary

3.1 Robotic Manipulation

We model robotic manipulation as a probabilistic inference model. For each time t, the robot obtains the multi-view RGB images $\mathbf{O}_t$ (also known as observation), the proprioception $\mathbf{q}_t$, and language instruction $\mathbf{L}$, we aim to predict the action 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 an action chunk of furture actions (we use chunk length $H = 60$ for our tasks) for the robot will play; where for the single-arm manipulation, each $\mathbf{a} \in \mathbb{R}^7$ denoted as 7-DoF action space: 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 $g \in \{0, 1\}$. For the dual-arm manipulation, it extends the setting to a 14-DoF space.

3.2 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`"), where in this paper, $\mathbf{L}$ is tokenized by Qwen3 [42], we denote it as follows,

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

For each time t, current observation $\mathbf{O}_t \in \mathbb{R}^{v\times 3\times h\times w}$ contains multi-view images, where our robot is with three views (i.e., $v = 3$): a head-mounted view, along with a wrist-mounted view for each arm. Furthermore, a vision tokenization encoder compress the raw image $\mathbf{O}_t$ into a latent space, denoted it as

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

where $\hat{\mathbf{z}}_t \in \mathbb{R}^{c'\times h'\times w'}$ is with the channel number of c', and the latent size is with $h' \times w'$. In this paper, we apply DINO-v3 [31] as the vision encoder to encode the input images. The DINO-v3 encoder provides more comprehensive representations with concurrently discern spatial and geometric relationships across observations. Such representations serve as high-quality inputs for downstream policy networks (e.g., diffusion policies or action models), enhancing the accurate perception of the relative pose between the manipulator's end-effector and the target object, and consequently enabling the generation of more refined motion trajectories.

3.3 Timestep Embedding

This section present the method of timestep embedding. For a given timestep τ, first, the scalar τ is embedded into a high-dimensional, multi-scale space that exhibits favorable distance properties; then, a two-layer MLP adapts the frequency encoding for the downstream task. Let d be the embedding dimension, $k = \lfloor d/2 \rfloor$, $\tilde{\mathbf{t}} \in \mathbb{R}^d$, for each $j \in \{0, 1, \cdots, d-1\}$:

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

Furthermore, we apply a two-layer MLP to map the vector $\tilde{\mathbf{t}}$ from d-dimension to d-dimension as follows,

$$\mathbf{t} = \texttt{FNN}(\tilde{\mathbf{t}}), \tag{4}$$

where `FNN` is a neural network that first increases the dimensiona of $\tilde{\mathbf{t}}$, and then reduces it to be the d-dimension. Finally, we present above embedding as the following formulation,

$$\mathbf{t} = \texttt{Timestep Embedding}(\tau). \tag{5}$$

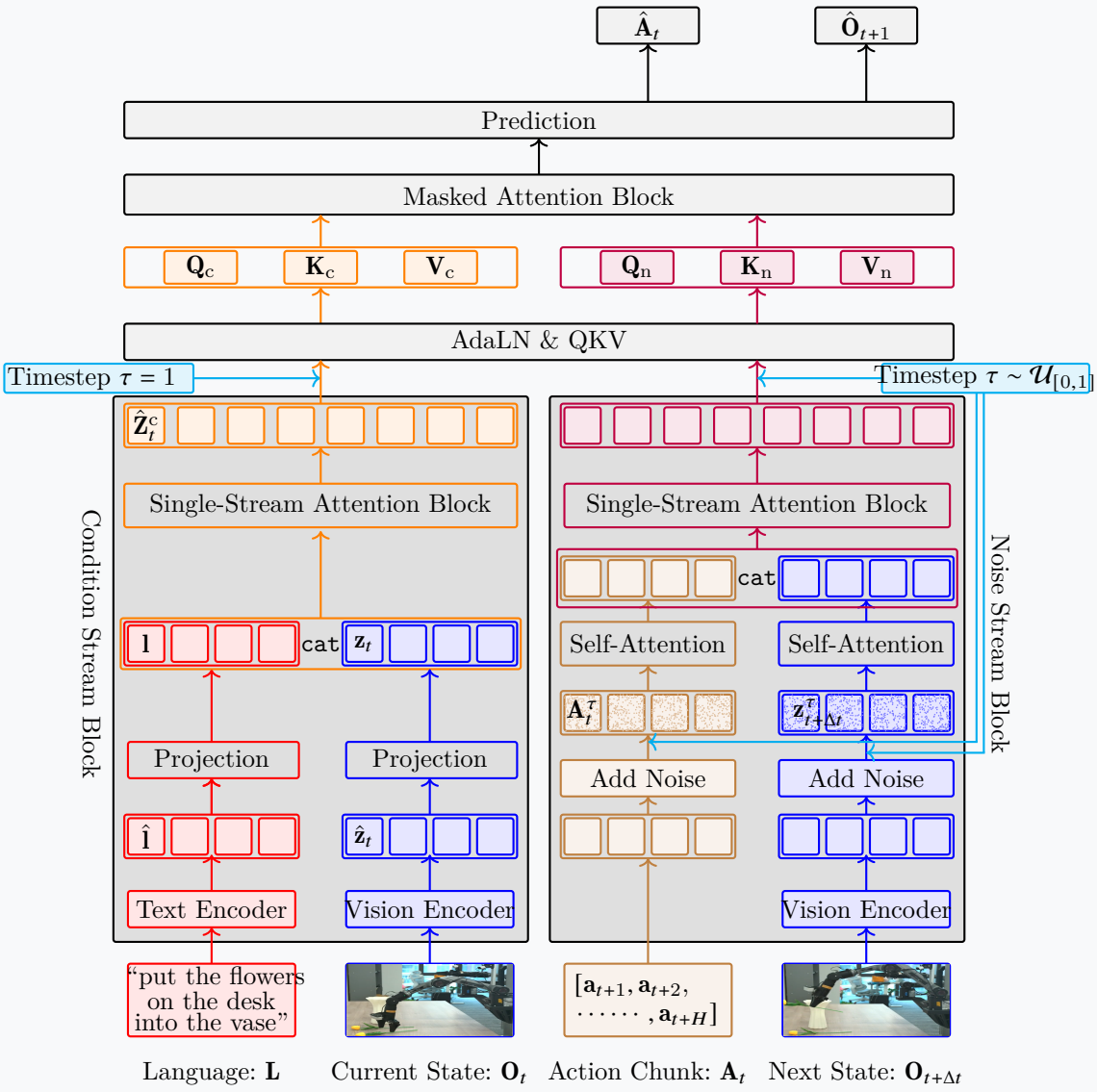

Figure 2: Overview of `HeTu-1.0` architecture. The model consists of a condition stream and a noise stream. The condition stream encodes the language instruction $\mathbf{L}$ and current observation $\mathbf{O}_t$, while the noise stream jointly denoises the action chunk $\mathbf{A}_t$ and future state $\mathbf{O}_{t+\Delta t}$. The two streams are coupled through AdaLN-modulated QKV projections and masked attention, enabling unified prediction of the robot action and its corresponding latent future state.

4 Framwork of HeTu-1.0

The framwork of `DELE-w0.5` adopts a dual-stream attention mechanism architecture, which contains two blocks: condition stream and noise stream.

4.1 Condition Stream Block

4.1.1 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 $\hat{\mathbf{z}}_t$. Since the output dimensions of the vision encoder and the language encoder are not necessarily the same, then we apply the projection operator to ensure that feature information $\mathbf{l}$ and $\mathbf{z}_t$ have the same hidden dimension,

$$\mathbf{l} = \texttt{Projection}(\hat{\mathbf{l}}), \quad \mathbf{z}_t = \texttt{Projection}(\hat{\mathbf{z}}_t), \tag{6}$$

where in this paper, we set hidden dimension with the value of 1024, the $\texttt{Projection}(\cdot)$ operator we used is the linear neural networks.

4.1.2 Single-Stream Attention Block

To integrate the information of language instruction with current observations, we employ a single-stream attention block [19] that concatenates the multimodal tokens (i.e., $\mathbf{l}$ and $\mathbf{z}_t$), and feeds them into a shared Transformer encoder, where the attention block enables unified, source-agnostic cross-modal interactions. Concretely, firstly, we concatenate the language and vision tokens as follows,

$$\mathbf{Z}_t^c = \texttt{cat}(\mathbf{l}, \mathbf{z}_t) =: [\mathbf{l}, \mathbf{z}_t]. \tag{7}$$

Then we apply the standard self-attention block [34] to refine the concatenated information as follows,

$$\hat{\mathbf{Z}}_t^c = \texttt{Attention}(\mathbf{Z}_t^c), \tag{8}$$

which exploites the complementary information gain from language and vision, and shows the generalization robustness in embodied artificial intelligence.

4.1.3 AdaLN and Output of Condition Stream

Furthermore, `DELE-w0.5` applies the AdaLN facilitates the integration of conditioning information into the feature representation, which resides the dynamic learning of intermediate representations in response to the incoming conditioning signal, and which ia a critical prerequisite for robots to attain fine-grained control. After the single-stream attention block, with the fused information $\hat{\mathbf{Z}}_t^c$, we obtain the query, keys, values as the output of condition stream as follows,

$$\mathbf{Q}_c = \text{QKNorm}\left\{\mathbf{W}_q(\text{RMSNorm}(\hat{\mathbf{Z}}_t^c) \odot \boldsymbol{\gamma}_c)\right\}, \tag{9}$$

$$\mathbf{K}_c = \text{QKNorm}\left\{\mathbf{W}_k(\text{RMSNorm}(\hat{\mathbf{Z}}_t^c) \odot \boldsymbol{\gamma}_c)\right\}, \tag{10}$$

$$\mathbf{V}_c = \mathbf{W}_v(\text{RMSNorm}(\hat{\mathbf{Z}}_t^c) \odot \boldsymbol{\gamma}_c), \tag{11}$$

where $\text{RMSNorm}(\cdot)$ is root mean square normalization [46]; $\boldsymbol{\gamma}_c$ is the scaling factor for the condition stream with following linear structure,

$$\boldsymbol{\gamma}_c = 1 + \mathbf{W}_{\text{Adaln}}\mathbf{t}_c + \mathbf{b}_{\text{Adaln}}, \tag{12}$$

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

$$\mathbf{t}_c \overset{(5)}{=} \texttt{Timestep Embedding}(1);$$

$\text{QKNorm}(\cdot)$ is a normalization operator maps any vector $\mathbf{x}$ as follows,

$$\mathbf{x}' = \text{QKNorm}(\mathbf{x}) = \mathbf{x}/\|\mathbf{x}\|_2; \tag{13}$$

and $\text{QKNorm}(\cdot)$ applies ℓ_2 normalization along the head dimension of each query and key matrix prior to multiplying them.

4.2 Noise Stream Block

For each time t, we need to add noise to the clear data $\mathbf{A}_t$ and next state $\mathbf{O}_{t+\Delta t}$. Let $\tau \sim \mathcal{U}_{[0,1]}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, where $\mathcal{U}_{[0,1]}$ is the uniform distribution on $[0, 1]$, we obtain noise action as follows,

$$\mathbf{A}_t^\tau = \tau\mathbf{A}_t + (1 - \tau)\boldsymbol{\epsilon}. \tag{14}$$

The noise addition of image is on the latent space:

$$\hat{\mathbf{z}}_{t+\Delta t} = \texttt{Vision Encoder}(\mathbf{O}_{t+\Delta t}), \tag{15}$$

then we obtain the noise image as follows,

$$\mathbf{z}_{t+\Delta t}^\tau = \tau\hat{\mathbf{z}}_{t+\Delta t} + (1 - \tau)\boldsymbol{\epsilon}. \tag{16}$$

Furthermore, following the same steps from (7), (8), (9), (10) and (11), we obtain the query, keys, values as the output of noise stream, denoted them as $\mathbf{Q}_n$, $\mathbf{K}_n$ and $\mathbf{V}_n$.

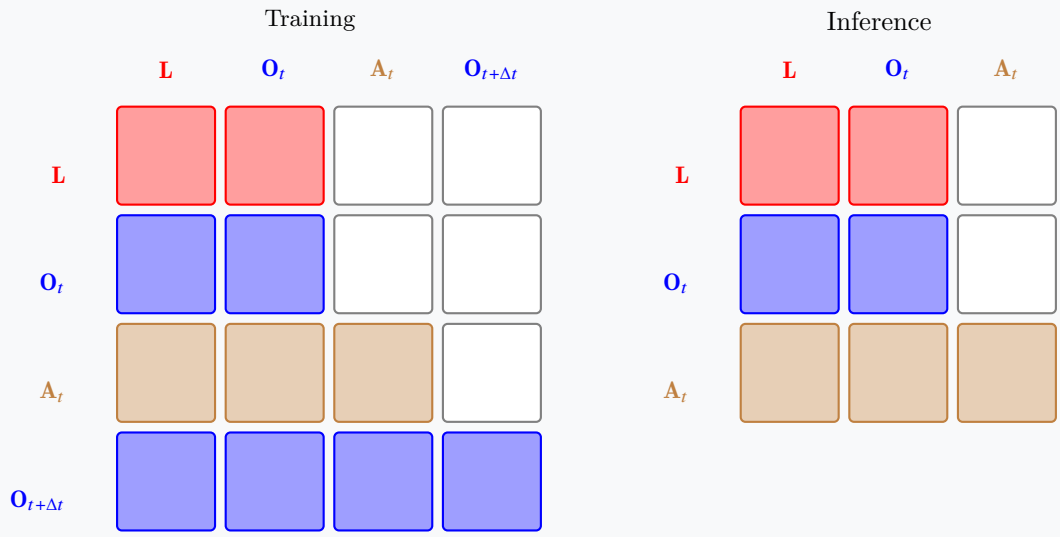

Figure 3: Attention masks for training and inference. Rows are query-token groups and columns are key/value-token groups. $\mathbf{L}$, $\mathbf{O}_t$, $\mathbf{A}_t$, and $\mathbf{O}_{t+\Delta t}$ denote language, current observation, action, and predicted future observation, respectively. During training (left), $\mathbf{L}$ and $\mathbf{O}_t$ attend bidirectionally; $\mathbf{A}_t$ attends to $\{\mathbf{L}, \mathbf{O}_t, \mathbf{A}_t\}$; and $\mathbf{O}_{t+\Delta t}$ attends to all groups. During inference (right), $\mathbf{O}_{t+\Delta t}$ is omitted. Colored and white cells indicate permitted and masked attention, respectively. Left- and right-view tokens are merged into each observation group, and padded language positions are masked along both dimensions.

4.3 Flow Matching Objective

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

$$\mathbf{Q} = \texttt{cat}(\mathbf{Q}_\text{c}, \mathbf{Q}_\text{n}), \quad \mathbf{K} = \texttt{cat}(\mathbf{K}_\text{c}, \mathbf{K}_\text{n}), \quad \mathbf{V} = \texttt{cat}(\mathbf{V}_\text{c}, \mathbf{V}_\text{n}). \tag{17}$$

With the joint attention block and projection layer, we obtain the predicted action chunk $\hat{\mathbf{A}}_t$, and predicted next latent observations $\hat{\mathbf{z}}_{t+\Delta t}$ as follows,

$$\hat{\mathbf{A}}_t^\tau,\ \hat{\mathbf{z}}_{t+\Delta t}^\tau = \texttt{Projection}(\texttt{Masked Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})). \tag{18}$$

Finally, we regress the velocity field according the following way,

$$\mathcal{L}_\text{a} = \mathbb{E}_{\tau\sim\mathcal{U}_{[0,1]},\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\left\|\mathbf{u}_\text{a}(\mathbf{A}_t^\tau, \mathbf{z}_{t+\Delta t}^\tau, \mathbf{q}_t, \mathbf{l}) - (\mathbf{A}_t - \boldsymbol{\epsilon})\right\|_2^2\right], \tag{19}$$

$$\mathcal{L}_\text{o} = \mathbb{E}_{\tau\sim\mathcal{U}_{[0,1]},\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\left\|\mathbf{o}_\text{a}(\mathbf{A}_t^\tau, \mathbf{z}_{t+\Delta t}^\tau, \mathbf{q}_t, \mathbf{l}) - (\hat{\mathbf{z}}_{t+\Delta t} - \boldsymbol{\epsilon})\right\|_2^2\right]. \tag{20}$$

The overall training objective is

$$\mathcal{L} = \mathcal{L}_\text{a} + \lambda\mathcal{L}_\text{o}, \tag{21}$$

where λ balances the action learning the next state prediction.

4.4 Training and Inference

During training, the input tokens are arranged as language, left-view observation, right-view observation, action, observation, and future observation. The language and current-observation tokens form the conditioning group and attend bidirectionally to one another. They cannot attend to either the action tokens or the predicted future-observation tokens, which prevents information from the prediction targets from leaking into the conditioning representation. The action tokens attend to the entire conditioning group and to the action group itself, but they cannot access the predicted future observations. Therefore, robot actions are learned only from the language instruction, the current multi-view observations, and the dependencies within the action sequence. In contrast, the predicted future-observation tokens attend to all token groups, allowing the

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° 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.

model to learn the physical state transition conditioned on the instruction, current observations, and robot actions. During inference, the two predicted future-observation groups are removed, while the visibility rules for the conditioning and action groups remain unchanged. The model therefore predicts the action sequence without generating any future visual tokens, which shortens the inference sequence, removes unnecessary visual generation, and enables efficient low-latency robot control.

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 vision-language-action (VLA) 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 the same 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. A stage is counted only when all preceding stages have been completed in order. Thus, a later action cannot compensate for a skipped prerequisite.

Task-suite design. The task suite covers six sources of difficulty in long-horizon manipulation: contact-rich articulated interaction, target-object retrieval, bimanual transfer, tool use, constrained insertion, and terminal-state verification. The door task requires a stable handle grasp followed by timely handle rotation and pushing to release the latch. The Pepsi task adds target extraction, a right-to-left handoff, and refrigerator closure while retaining the can. The ice task is the longest sequence and requires coordinated use of the scoop, ice-maker lid, and cup, followed by pouring and scene restoration. The microwave task combines appliance opening, package pickup, constrained insertion, door closure, and heating activation. These tasks therefore test not only individual manipulation skills, but also whether a policy can connect the skills while preserving previously achieved task state.

Baselines and training protocol. We compare `DELE-w0.5` with five representative VLA policies: $\pi_{0.5}$ [11], LingBot-VLA2 [40], Hy-Embodied-0.5-VLA (HY-VLA) [47], Spirit-v1.5 [32], and GR00T-N1.7 [25]. All methods are compared under identical fine-tuning data and evaluation protocols. Each baseline follows its official training configuration and is trained on the same task dataset for an equivalent of three epochs. At

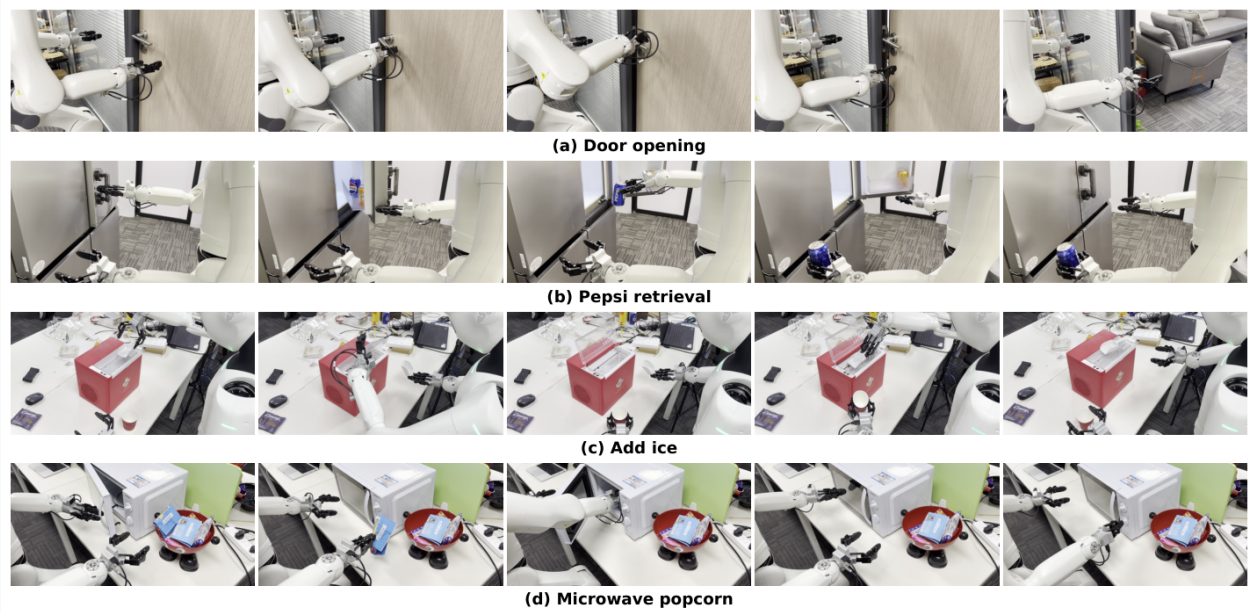

Figure 4: Representative stage sequences for the four real-robot tasks. Each row shows five temporally ordered key states from left to right for door opening, Pepsi retrieval, adding ice, and microwave popcorn.

evaluation time, all methods use the same robot embodiment, task instructions, scene-reset configurations, 180-second execution budget, ordered-stage definitions, and full-task success criteria.

Evaluation protocol. For each method-task pair, we run twenty formal trials, giving $6 \times 4 \times 20 = 480$ 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. The operator provides no physical assistance after execution begins and stops a rollout only after observable success or terminal task failure, or when continued execution is unsafe. 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.

Metrics. We report two metrics. The first is the *full-task success rate*, which measures whether the complete physical task condition is satisfied. The second is normalized ordered-stage progress. For a task with K stages and a trial whose highest consecutively completed stage is s, the progress score is s/K. We average this score over twenty trials for each task and then macro-average over the four tasks. Therefore, every task has equal weight even though the tasks contain different numbers of stages.

5.2 Main Results

Table 2 and Figure 5 compare normalized progress and full-task success. Results show that `DELE-w0.5` performs best on all four tasks. It obtains 81.3% macro progress and 62.5% overall success (50/80). The strongest baseline, $\pi_{0.5}$, obtains 50.6% macro progress and 15.0% success (12/80). Compared with this baseline, `DELE-w0.5` improves macro progress by 30.7 percentage points and full-task success by 47.5 percentage points.

Figure 5 also shows a consistent gap between stage progress and full success. Several baselines complete early interactions, such as contacting a handle or opening an appliance, but rarely preserve this progress through object transfer, bimanual coordination, and terminal-state completion. `DELE-w0.5` converts early progress into full completion much more often. This result indicates that its advantage mainly appears when errors can accumulate over a long action sequence.

5.3 Task-wise and Stage-wise Analysis

Figure 6 shows the empirical stage-reach probability $P(s \geq k)$ for each task. A single progress value measures how far a method moves on average, while the heatmaps show the exact stage at which performance drops. We make four observations from the results.

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. Macro progress weights the four tasks equally; overall success is computed over all 80 trials per method.

	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**
$\pi_{0.5}$	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

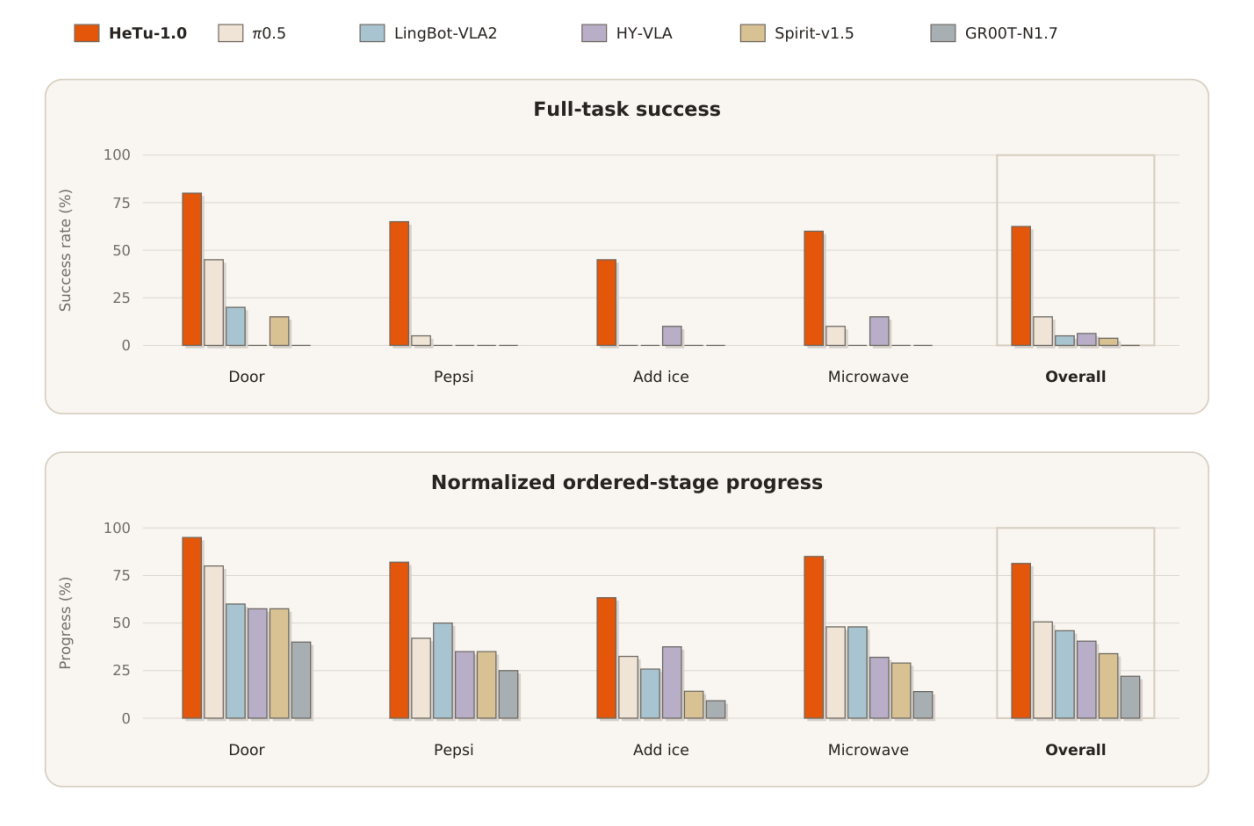

Figure 5: Task-wise real-robot performance. The top panel reports full-task success, and the bottom panel reports normalized ordered-stage progress. Overall denotes aggregate success over 80 trials in the top panel and macro-average progress across the four tasks in the bottom panel.

Door opening. From Figure 6(a), `DELE-w0.5` achieves 16/20 full successes and 95.0% normalized progress, followed by $\pi_{0.5}$ with 9/20 successes and 80.0% progress. After grasping the handle, the robot must maintain the grasp, rotate the handle to a sufficient angle, and push at the correct time. `DELE-w0.5` reaches this coordinated rotation-and-push stage in all twenty trials, and its four failures occur only at the final sustained-opening condition. In comparison, most baselines reach the handle but fail to synchronize the rotation and push or fail to maintain the required opening angle. This result shows that handle perception alone is insufficient; the task depends on timely contact-rich coordination.

Pepsi retrieval. Figure 6(b) shows that `DELE-w0.5` completes 13/20 Pepsi retrieval trials. It opens the refrigerator and extracts the target can in 18/20 trials. The remaining failures mainly occur at the right-to-left handoff and final door closure. LingBot-VLA2 reaches a stable left-hand handoff in six trials but never closes the refrigerator while retaining the can, whereas $\pi_{0.5}$ completes the full task once. Most other baselines fail

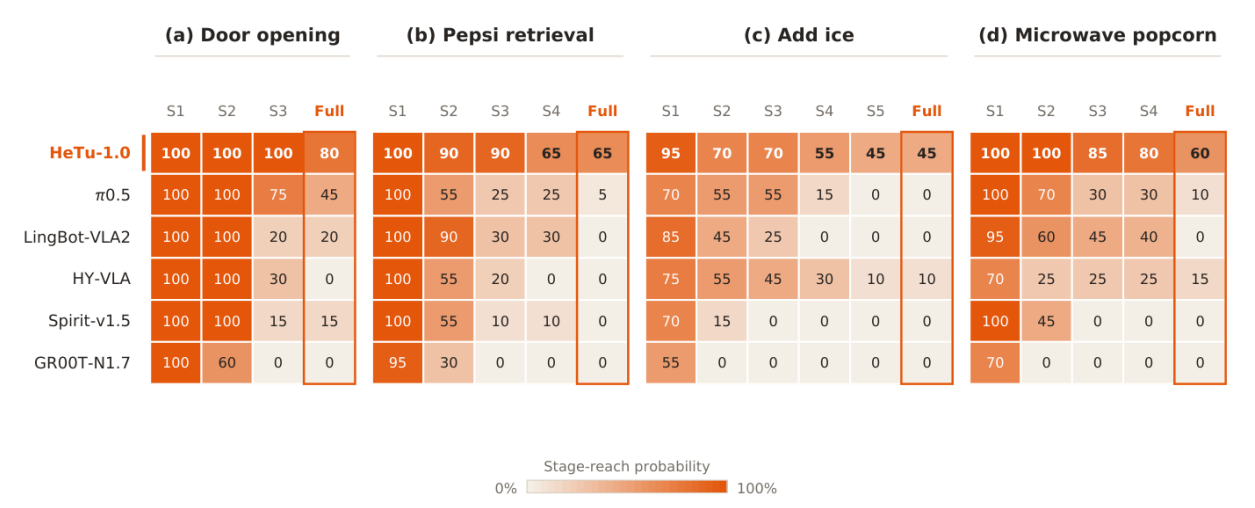

Figure 6: Stage-wise long-horizon analysis. Each cell is the percentage of twenty trials whose ordered-prefix score reaches the indicated stage. Task blocks share a common model axis, and the outlined *Full* column in each block represents complete physical task success.

during refrigerator opening. Thus, articulated interaction is the first bottleneck, while handoff and closure determine whether early progress becomes a complete retrieval.

Adding ice. Figure 6(c) reports the longest task sequence. `DELE-w0.5` completes the six-stage task in 9/20 trials. It picks up the scoop in 19 trials, reaches cup pickup in 14, acquires ice in 11, and pours ice into the cup in nine. HY-VLA is the strongest competing method by progress (37.5%) and completes the task twice. $\pi_{0.5}$ obtains 32.5% progress but never places ice into the cup, while the remaining methods generally stop at scoop pickup or lid opening. These results show that a successful initial grasp is not enough. Repeated bimanual coordination and precise release determine the final performance.

Microwave popcorn. From Figure 6(d), `DELE-w0.5` obtains 12/20 full successes and 85.0% normalized progress. It opens the microwave and picks up the package in every trial, closes the microwave in 16 trials, and starts heating in 12. Although $\pi_{0.5}$ and Spirit-v1.5 open the microwave in all twenty trials and LingBot-VLA2 does so in 19, their final success rates are only 2/20, 0/20, and 0/20. This sharp stage-wise drop explains why appliance opening alone is a weak measure of long-horizon performance. Package retention, insertion, door closure, and heating activation introduce successive opportunities for failure.

5.4 Failure Analysis

For each unsuccessful trial, we record the first uncompleted ordered stage. This view identifies the main bottleneck without relying on model-specific runtime logs. The dominant failure stage differs across tasks. For door opening, most baselines grasp the handle but fail during coordinated rotation and pushing. For Pepsi retrieval, weaker methods fail before full extraction, while stronger methods lose progress during handoff or final refrigerator closure. For adding ice, the main drop occurs before ice acquisition and pouring. For microwave popcorn, many rollouts open the appliance but fail during insertion, door closure, or heating activation.

All four bottlenecks require a policy to preserve previously achieved task state while changing the active arm, object, or interaction mode. `DELE-w0.5` reaches the late stages substantially more often, although its remaining failures still occur during object transfer, precise release, or the final task condition. The results therefore show that long-horizon performance depends both on individual skills and on the reliability of the transitions between them.

6 Conclusion

In this work, we revisit the objective of World-Action Models for robotic manipulation. We show that video generation is not a necessary component of world modeling for robot control. Instead of reconstructing how the world visually evolves over time, a practical world model should predict how the physical world will change after an action is executed. Based on this observation, we introduce HeTu-1.0, a new future-state-driven world-action framework that infers robot actions from compact action-relevant future states without relying on video generation. By treating future-state prediction as the intermediate representation between world understanding and action generation, DELE removes unnecessary visual redundancy and enables efficient training and low-latency inference. Extensive real-world experiments demonstrate the effectiveness of this formulation on challenging long-horizon manipulation tasks. Across 480 robot trials on four diverse tasks, DELE consistently outperforms representative VLA baselines, achieving 62.5% full-task success and 81.3% normalized ordered-stage progress. The results show that the advantage of DELE becomes more significant as task horizons increase and errors accumulate across multiple interaction stages. These findings suggest that action-relevant future-state prediction provides a more direct and practical foundation for embodied world models, offering a promising direction toward scalable and deployable robot intelligence.

References

[1] 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.

[2] 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.

[3] 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. π_0: A vision-language-action flow model for general robot control, 2026. URL https://arxiv.org/abs/2410.24164.

[4] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. URL https://arxiv.org/pdf/2212.06817.

[5] 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.

[6] Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. Worldvla: Towards autoregressive action world model, 2025. URL https://arxiv.org/abs/2506.21539.

[7] 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.

[8] 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*, pages 8469–8488. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/driess23a.html.

[9] 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, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 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.

[10] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025. URL https://www.nature.com/articles/s41586-025-08744-2.pdf.

[11] 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.

[12] 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.

[13] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success, 2025. URL https://arxiv.org/abs/2502.19645.

[14] 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.

[15] 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.

[16] 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/.

[17] 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, 2026. URL https://arxiv.org/abs/2606.10040.

[18] 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*, 2026. URL https://arxiv.org/pdf/2607.17977.

[19] 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.

[20] 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.

[21] 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, 2026. URL https://arxiv.org/abs/2606.08242.

[22] Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Jingbin Cai, Si Liu, Jianlan Luo, Liliang Chen, Shuicheng Yan, Maoqing Yao, and Guanghui Ren. Genie envisioner: A unified world foundation platform for robotic manipulation, 2025. URL https://arxiv.org/abs/2508.05635.

[23] 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.

[24] 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.

[25] 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.

[26] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Gupta, and Zipeng Lin. Open x-embodiment: Robotic learning datasets and rt-x models : Open x-embodiment collaboration0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903, 2024. URL 10.1109/ICRA57147.2024.10611477.

[27] 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.

[28] Lu Qiu, Yizhuo Li, Yi Chen, Yuying Ge, Yixiao Ge, and Xihui Liu. Making foresight actionable: Repurposing representation alignment in world action models, 2026. URL https://arxiv.org/abs/2606.12217.

[29] Younggyo Seo, Junsu Kim, Stephen James, Kimin Lee, Jinwoo Shin, and Pieter Abbeel. Multi-view masked world models for visual robotic manipulation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 30613–30632. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/seo23a.html.

[30] 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.

[31] 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.

[32] Spirit AI. Spirit-v1.5: Clean data is the enemy of great robot foundation models. Spirit AI Blog, January 2026. URL https://www.spirit-ai.com/en/blog/spirit-v1-5. Published January 11, 2026.

[33] 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.

[34] 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, editors, *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.

[35] 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.

[36] 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.

[37] Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. Dexvla: Vision-language model with plug-in diffusion expert for general robot control, 2025. URL https://arxiv.org/abs/2502.05855.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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, 2026. URL https://arxiv.org/abs/2603.17240.

[44] 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, 2026. URL https://arxiv.org/abs/2602.15922.

[45] Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination?, 2026. URL https://arxiv.org/abs/2603.16666.

[46] 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, editors, *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.

[47] 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.

[48] Yuyang Zhang, Wenyao Zhang, Zekun Qi, He Zhang, Haitao Lin, Jingbo Zhang, Yao Mu, Xiaokang Yang, Wenjun Zeng, and Xin Jin. Imagewam: Do world action models really need video generation, or just image editing?, 2026. URL https://arxiv.org/abs/2606.19531.

[49] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Tsung-Yi Lin, Gordon Wetzstein, Ming-Yu Liu, and Donglai Xiang. CoT-VLA: Visual Chain-of-Thought Reasoning for Vision-Language-Action Models . In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1702–1713, 2025. URL https://cvpr.thecvf.com/virtual/2025/poster/33233.

[50] 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.

[51] 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. τ_0-wm: A unified video-action world model for robotic manipulation, 2026. URL https://arxiv.org/abs/2606.01027.

[52] 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.

[53] 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, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229, pages 2165–2183, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.