DELTAWAM: CHANGE-CENTRIC VISUAL FORESIGHT VIA DELTA TOKENS FOR EFFICIENT WORLD-ACTION MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

World-Action Models (WAMs) offer promising foresight for robotic manipulation, yet efficient and dynamics-aware future representations haven't been fully established. Pixel-space WAMs use representations that serve for repainting entire scenes, which brings up computational overhead, spatial-temporal redundancy and waste of model capacities on task-irrelevant information. To resolve these inefficiencies, we start from an overlooked observation: consecutive frames in physical manipulation usually share substantial contexts and differ only in structured and low-dimensional ways (e.g., object displacements and end-effector motion). In essence, it is these dynamic changes, rather than sequences of entire scenes, that an action policy needs to anticipate. Grounded in this principle, we build **DeltaWAM**, a World-Action model that shifts the fundamental unit of future predictions to a delta token. Each delta token is a single 1D vector that encodes the changes of dense DINO features between consecutive frames, inherently capturing dynamic transitions while reducing redundancy from shared static context. Our DeltaWAM builds on DeltaWorld, a latent world model that is pretrained on large-scale videos and forecasts change-centric futures: given a history of observed delta tokens, it autoregressively predicts the future delta sequence, one token per frame. These predicted transitions, alongside the DINO features of current observations acting as the spatial anchor, are then passed to a flow-matching action expert that generates the action chunk. Our experiments show that by focusing merely on what will change in the future, delta tokens prove to be more dynamics-aware and action-forcing, providing a natural bridge from high-level semantics to low-level control. With just 256 GPU hours on 2 H100 GPUs and 0.725B total parameters, DeltaWAM achieves high success rates(92.8%) on LIBERO while also demonstrating robust generalization under procedural perturbations on LIBERO-Pro. At inference the latency is 142.1 ms latency per action chunk and peak memory is only 3.86 GB.

1 INTRODUCTION

World-Action Models (WAMs), which couple a predictive world model with action generation, have recently made rapid progress in robotic manipulation [35, 52, 27, 30, 23, 29]. However, most existing WAMs synthesize full scenes at every future timestep rather than directly modeling dynamic changes. As a prime example, pixel-space WAMs typically adopt video generative backbones to imagine the future frame by frame [22, 58, 39, 35, 52, 27, 30]. These models allocate substantial capacity to full pixel-level synthesis rather than distilling compact, action-relevant dynamics. Consequently, they spend much of their computation reconstructing task-irrelevant visual details that are largely repeated across frames, resulting in severe spatio-temporal redundancy. While latent WAMs circumvent pixel-level synthesis and operate in compressed feature spaces [36, 32, 10, 29, 34], they inherit the same limitation. By forecasting absolute latent states frame by frame, they continue to model full scenes in the latent space rather than directly compressing dynamic transitions. This not only wastes models' representational capacity in maintaining static shared contexts, but also overshadows action-relevant dynamics that is important for low-level control.

In the physical world, **motion manifests as changes in state over time**. During action planning, the human visual and sensorimotor systems prioritize task-relevant visual dynamics such as changes in

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.

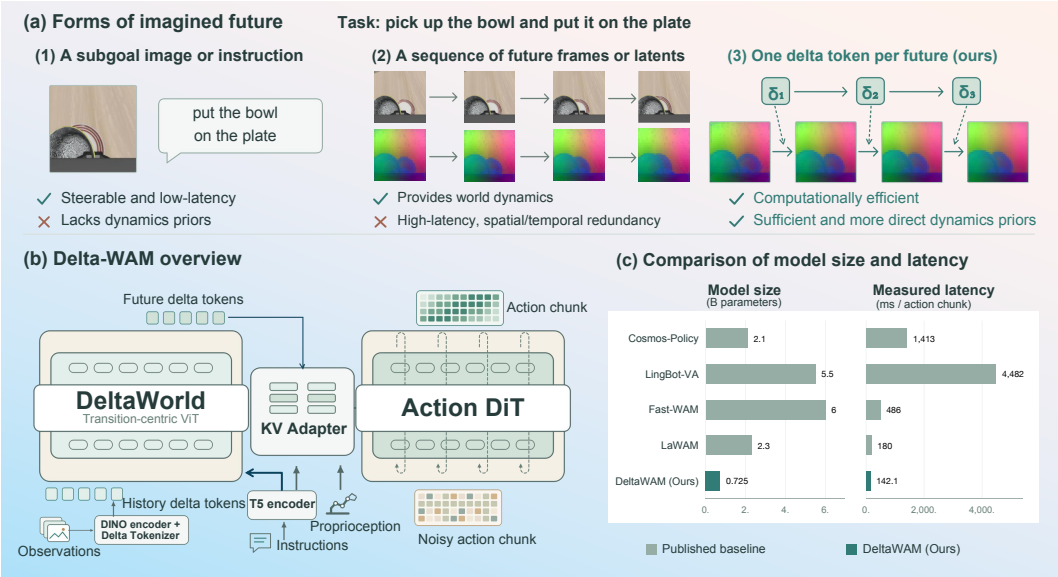


Figure 1: **DeltaWAM overview.** (a) Existing policies represent an imagined future as a single subgoal or a sequence of dense frames or latent states, whereas DeltaWAM predicts one compact transition token per future step. (b) DeltaWAM couples a latent transition-centric world model DeltaWorld with an Action DiT, preserving efficient test-time visual foresight. (c) DeltaWAM is lightweight and has low latency compared with representative World-Action models.

object positions while attenuating static visual context such as the background details [17, 13, 48, 12, 49, 44]. In the same vein, the predictive focus of a World-Action model must center on dynamic changes that the robot needs to realize, stripping away the redundant reconstruction of visual contexts shared across consecutive frames.

Following this principle, we propose to represent the imagined future as a sequence of transitions rather than a sequence of scenes. We instantiate this idea with delta tokens, originally introduced in DeltaWorld [25] for efficient video world modeling. In DeltaWorld, a delta tokenizer compresses frame-to-frame DINOv3 [45] feature differences into a single compact delta token. We observe that this representation, though designed for video prediction, is particularly well suited to control. It discards the static context shared across frames and retains only what changes. Consequently, temporal dynamics are directly exposed as a compact control prior, allowing the policy to plan actions conditioned on explicit state transitions.

Turning DeltaWorld into a change-centric World-Action model raises two critical questions: how can the prediction itself be made more task-relevant, and how to feed the predicted transitions into the policy? Our DeltaWAM introduces a unified architecture to address both. The first question matters because we recognize that original DeltaWorld is inherently unconditional: it extrapolates futures based solely on observation history. In robotic manipulation, however, a single scene can unfold into multiple valid futures depending on the instruction. An unconditional rollout might therefore hallucinate a physically plausible future that misaligns with the task, ultimately misleading the downstream policy. To resolve this, we inject language instructions directly into the DeltaWorld Predictor via Gated AdaLN [40, 41]. An instruction embedding from a frozen text encoder modulates the scale, shift, and residual gates of every DeltaWorld block. For the second question, we pair DeltaWorld with a flow-matching Action DiT and condition it on two complementary signals. The first signal is the dense, patch-level DINO features of the current observation, which serve as a **spatial anchor** encoding precise static context (i.e., *where* things are). The second signal is the sequence of predicted future delta tokens, which form a **dynamic subgoal** encapsulating motion intent (i.e., *how* things should get moved).

A controlled ablation validates our theory: providing the policy with full future DINO latents reduces the success rate on the standard LIBERO benchmark from 92.8% to 79.0% relative to conditioning on compressed delta tokens. On LIBERO-Pro, DeltaWAM achieves a 17.85% average success rate under task perturbations, while the success rates of representative baselines are close to 0%. This

comparison further shows that instead of simply memorizing action trajectories for a given scene, DeltaWAM relies more on motion dynamics. To sum up, our contributions are three-fold:

- We identify the spatio-temporal redundancy in existing WAMs and propose factorizing visual foresight into static visual anchors and dynamic transition tokens, fundamentally shifting world modeling from scene reconstruction to change prediction.
- We present DeltaWAM, a lightweight (0.725B parameters, 3.86GB peak memory at inference) and computationally efficient World-Action model.
- Our experiments substantiate that compressed delta tokens provide a superior, action-forcing signal compared to full state predictions.

2 Related Work

2.1 World-Action Models

Vision-Language-Action (VLA) models build on large-scale vision-language pretraining, which connects language instructions with target objects and task structure [59, 26, 6, 5]. This semantic prior tells the robot what to do, but lacks dynamics priors of how the environment will evolve during interaction [39, 52, 10]. World-Action Models (WAMs) address this gap by coupling action generation with future prediction[22, 58, 39, 35, 52, 27]. Video-based WAMs leverage rich dynamics priors inherited from web-scale video pretraining to generate future frame sequences [22, 58, 39, 52, 27, 30]. However, their reliance on iterative pixel-level video diffusion introduces severe computational latency, creating a fundamental deployment bottleneck for real-time closed-loop control [53, 51, 10, 31]. To mitigate this overhead, efficiency-oriented architectures like Fast-WAM and Giga-World Policy either relegate world prediction to an auxiliary loss or formulate action-centered generation [53, 51]. While faster, these compromises weaken the functional role of predicted dynamics as explicit visual subgoals at test time. To preserve explicit test-time foresight without the burden of pixel rendering, recent Latent WAMs shift prediction to representation spaces not optimized for pixel reconstruction [36, 32, 10, 47, 34, 50, 23, 29]. For instance, LDA-1B models future states in a DINO feature space and jointly denoises future representations and action chunks [36]. JEPA-WAM predicts future V-JEPA latents via a language backbone, yet utilizes latent predictions purely as auxiliary conditioning during training rather than execution [32]. LaWAM predicts chunk-level DINO visual subgoals, but requires additional latent action distillation to guide world prediction [10]. Fundamentally, these models still predict latent for full scenes at each future timestep without extracting dynamic changes.

By contrast, our DeltaWAM also operates in DINO latent space but compresses the DINO differences between consecutive frames into a single compact delta token. Rather than iterative denoising, each delta token's generation requires only a single forward pass, which is more computational efficient. By decoupling future dynamic prediction from raw pixel synthesis and exposing the dynamic transitions directly to the policy, DeltaWAM provides stronger visual foresight for low-level control.

2.2 Latent Representations for World Modeling

Effective physical foresight in embodied decision-making relies heavily on the choice of world model state space [18, 19, 21, 56, 2]. Early visual world models typically operated in pixel space or compressed variational autoencoder (VAE) latents designed primarily for image reconstruction [16, 18, 19, 20]. However, raw visual representations exhibit severe spatial and temporal redundancy, and generative capacity is wasted on reconstructing task-irrelevant background details [22, 25, 10, 34].

To overcome these representation limits, self-supervised visual foundation models (VFMs) such as DINO capture high-level semantic layout and spatial structure without serving for pixel reconstruction [38, 45]. Building upon DINOv3 features, DeltaWorld compresses frame-to-frame feature differences into compact, one-dimensional delta tokens [25] for more effficient world modeling. Our work leverages this transition-centric latent paradigm, demonstrating that autoregressive prediction of low-dimensional delta tokens provides a highly efficient, action-forcing dynamic prior for downstream low-level control.

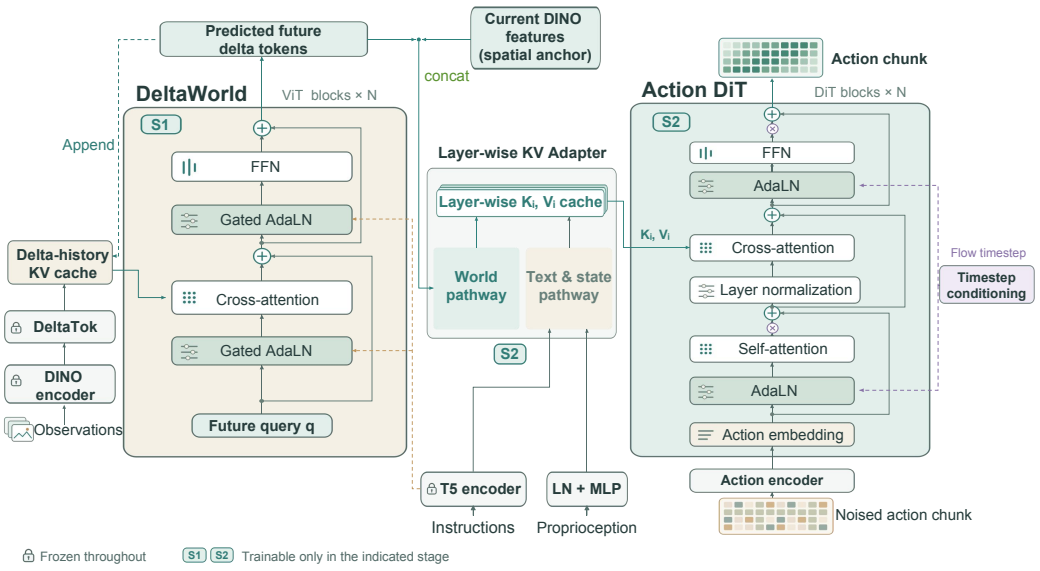


Figure 2: **Detailed DeltaWAM architecture.** DeltaWAM couples the change-centric DeltaWorld with a flow-matching Action DiT. The pretrained DeltaWorld is first finetuned on demonstrations; then Action DiT is trained on robot data with the finetuned DeltaWorld frozen. DeltaWorld first rolls out predicted delta tokens autoregressively. Afterwards, both current DINO latents (the spatial anchor) and predicted delta tokens (the dynamic subgoal) are fed to Action DiT.

3 METHOD

In this section, we present DeltaWAM, a framework that integrates transition-centric delta token prediction with low-level action generation. We first introduce the concept of delta tokens from DeltaWorld in Section 3.1. We then present task-conditioned world modeling in Section 3.2. Finally, in Section 3.3, we describe the training paradigms for the lightweight and computationally efficient DeltaWAM.

3.1 PRELIMINARY

Delta Tokenization Given a multi-view visual sequence $(O_1, \ldots, O_t)$, a frozen visual foundation backbone DINOv3 [45] extracts dense patch-level spatial feature maps $X_\tau = \phi(O_\tau) \in \mathbb{R}^{V \times N \times D_v}$ for each frame τ, where N is the number of spatial patches per view and D_v is the feature dimension. In DeltaWorld [25], the frame-level delta tokenizer is constructed based on a continuous autoencoder design. To initialize absolute visual grounding, a synthetic reference frame $O_\emptyset = \mathbf{0}$ with features $X_\emptyset = \phi(O_\emptyset)$ is prepended. The tokenizer encoder g compresses each pair of consecutive spatial feature maps into a 1D sequence of delta tokens $Z_{1:t} = (z_1, \ldots, z_t)$, where $z_\tau \in \mathbb{R}^{V \times 1 \times D_v}$, $z_1 = g(X_\emptyset, X_1, z_{\text{init}})$, and $z_\tau = g(X_{\tau-1}, X_\tau, z_{\text{init}})$ for $\tau > 1$. Here, $z_{\text{init}} \in \mathbb{R}^{1 \times D_v}$ is a learnable aggregation token prepended to the paired spatial feature maps before being fed into g. The tokenizer decoder h reconstructs the current feature map by transforming the previous features according to the delta token. The delta tokenizer is then trained using a reconstruction loss between the original and reconstructed feature maps. This objective encourages the delta token z_τ to serve as a compact transition representation between consecutive frames. Note that a timestamp sampling procedure can be employed for training pairs of $(X_{\tau-1}, X_\tau)$, which means that a single delta token can represent transitions ranging from nearly static scenes to substantial scene changes.

3.2 TASK-CONDITIONED WORLD MODELING WITH DELTA TOKENS

Let $Z_{\text{obs}} = Z_{t-T_c+1:t}$ denote the observed history context of delta tokens, with one delta token per camera view at each timestep. DeltaWorld generates each timestep's delta tokens autoregressively. At

rollout step j, it predicts the next tokens for all views in parallel, conditioned on the history context and its previous predictions:

$$\hat{z}_{t+j}^{1:V} = P_\psi\Big(q_{t+j}, Z_{\text{obs}}, \hat{Z}_{t+1:t+j-1}, c\Big), \qquad j = 1, \ldots, L_w, \tag{1}$$

where q_{t+j} is a stochastic query sampled from a Gaussian distribution and shared across views, and c is the task embedding. We start from DeltaWorld pretrained on Kinetics-700, a large-scale video dataset containing diverse human activities and object interactions [8]. To inject high-level semantics, we encode the task instruction with a frozen T5 encoder [43], project its token features to the predictor dimension, and obtain the task embedding c. The resulting embedding modulates each attention and MLP branch through Gated Adaptive Layer Normalization (Gated AdaLN) [41, 40]. For a branch input h and transformation F,

$$\tilde{h} = (1 + \gamma(c)) \odot \text{LN}(h) + \beta(c), \qquad h_{\text{out}} = h + (1 + \eta(c)) \odot F(\tilde{h}), \tag{2}$$

where γ, β, and η are language-dependent scale, shift, and residual-gate parameters. Crucially, their output projections are zero-initialized—this ensures that fine-tuning smoothly leverages the pretrained dynamics prior, allowing language to *steer* the learned physics rather than *overwrite* it.

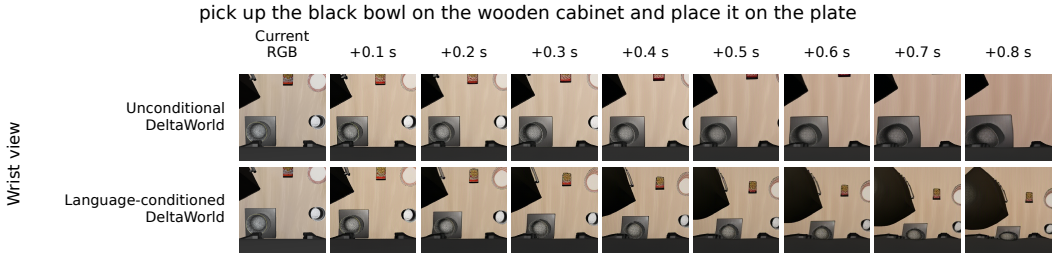


Figure 3: **Language steers predicted futures toward the task goal.** Wrist-view rollouts for LIBERO-Spatial task 9, which asks the robot to pick up the black bowl on the wooden cabinet and place it on the plate. In DeltaWorld's unconditional predictions, the robot approaches the stove instead of the cabinet. This is misleading since the bowl is on the cabinet.

3.3 Training Paradigm

3.3.1 Stage 1: BoM Training for Task-Conditioned DeltaWorld

Following the original DeltaWorld training recipe [25], we adopt a Best-of-Many (BoM) training paradigm [3]. At each supervised position τ, K Gaussian queries are sampled and go through the forward pass in parallel. For each candidate k, the same query is shared across different camera viewpoints, allowing the DeltaWorld predictor P_ψ to predict a coherent multi-view hypothesis:

$$\hat{z}_\tau^{(k),1:V} = P_\psi\left(q_\tau^{(k)}, Z_{<\tau}, c\right), \qquad k = 1, \ldots, K. \tag{3}$$

Here causal teacher forcing on the clean target sequence $Z_{1:T}$ is applied. For every supervised position $\tau \in \{2, \ldots, T\}$, the predictor attends only to the ground-truth prefix $Z_{<\tau}$ through a causal mask, which prevents access to the current and future targets. Let ρ_{SL1} denote the Smooth-L1 prediction loss of predicted delta tokens averaged over all the camera viewpoints. Using this loss, we can get the candidate that is closest to the ground-truth future:

$$\mathcal{E}_\tau^{(k)} = \frac{1}{V}\sum_{v=1}^{V} \rho_{\text{SL1}}\left(z_\tau^v, \hat{z}_\tau^{(k),v}\right), \qquad k_\tau^* = \arg\min_{k \in \{1,\ldots,K\}} \mathcal{E}_\tau^{(k)}. \tag{4}$$

Then at each position only the selected candidate gets supervised by minimizing the training objective $\mathcal{L}_{\text{world}} = \frac{1}{T-1}\sum_{\tau=2}^{T} \mathcal{E}_\tau^{(k_\tau^*)}$. In other words, gradients are propagated only through the selected candidate. Because each position is trained under teacher forcing with its own Gaussian queries, the selected candidate index is not required to remain fixed across the sequence.

The BoM training paradigm brings several advantages. Firstly, it allows DeltaWorld to be generative rather than discriminative. When multiple future outcomes are plausible, a regression objective tends to drive the model toward an averaged prediction that may not correspond to any realistic outcomes. Also, compared with diffusion-based generative models that involve iterative denoising, this is more computationally efficient. Another main point is that the downstream policies can be multimodal—there are multiple ways to finish a single task. The BoM training objective leaves space for imagining diverse visual subgoals with a unified motion intent.

3.3.2 Stage 2: Delta-Conditioned Policy Training

We freeze the task-finetuned DeltaWorld predictor and train Action DiT together with its KV adapter. DeltaWorld first autoregressively samples K candidate future trajectories. An "oracle" candidate branch $k^\dagger$ is identified by the minimal trajectory error:

$$k^\dagger = \arg\min_{k\in\{1,\dots,K\}} \sum_{j=1}^{L_w} \sum_{v=1}^{V} \|z_{t+j}^{v} - \hat{z}_{t+j}^{(k),v}\|_2^2. \tag{5}$$

During training, Action DiT conditions on the "oracle" branch $k^\dagger$ with probability p_{oracle} and on a randomly selected branch from $\{1,\dots,K\}$ with probability $1-p_{\text{oracle}}$. This design makes sure that Action DiT is more likely to condition on the "oracle" branch by setting a reasonable p_{oracle}. The prediction quality of non-oracle branches is unstable; therefore, we cannot sample all branches uniformly. For a ground-truth action chunk a, Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, and flow time σ, let $a_\sigma = (1-\sigma)a + \sigma\epsilon$. Given the selected conditioning context $\mathcal{C}$, we minimize:

$$\mathcal{L}_{\text{action}} = \mathbb{E}_{a,\epsilon,\sigma}\left[w(\sigma)\,\|v_\theta(a_\sigma, \sigma; \mathcal{C}) - (\epsilon - a)\|_F^2\right] \tag{6}$$

where v_θ is Action DiT's predicted velocity and $w(\sigma)$ is the flow-time weight. At inference, a future trajectory is sampled from the DeltaWorld predictor for generation of one action chunk.

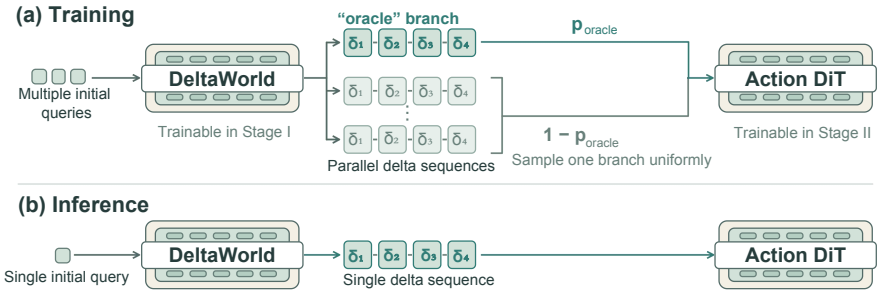


Figure 4: **DeltaWAM training and inference.** During the second training stage, Action DiT learns from oracle-selected or randomly sampled future trajectories produced by frozen DeltaWorld. At inference, one sampled future trajectory conditions action generation.

4 Experiments

We aim to evaluate whether compact delta-token foresight enables effective manipulation at a low inference cost, and whether it exhibits representational superiority in providing dynamics priors for low-level control. Our experiments examine closed-loop task success and latency on LIBERO, robustness to visual and procedural distribution shifts, and the contribution of predicted delta tokens for task execution. We also conduct the real-world evaluation on a PiPER robotic arm.

4.1 Experimental Setup

Benchmarks. LIBERO is a simulated benchmark for various manipulation tasks, and we evaluate across its four suites: Spatial, Object, Goal, and LIBERO-10 (Long) [33]. LIBERO-Pro tests procedural robustness beyond the standard tasks, and we use its task-perturbation dimension, which redefines task logic and target states[57].

Table 1: Standard LIBERO success rate (%), model size, and model-only latency per action-chunk query. Emb. PT denotes embodied action pretraining. Baseline sizes and latency values follow their source papers and the comparison protocol of LaWAM [10].

Method	Size	Emb. PT	Latency (ms)	Spatial	Object	Goal	Long	Average
Reactive Policies								
Diffusion Policy [11]	263M	×	–	78.3	92.5	68.3	50.5	72.4
OpenVLA [26]	7B	✓	–	84.7	88.4	79.2	53.7	76.5
π_0 [6]	3.5B	✓	220	98.0	96.8	94.4	88.4	94.4
$\pi_{0.5}$ [5]	3.5B	✓	220	98.8	98.2	98.0	92.4	96.9
GR00T-N1.6 [37]	3.3B	✓	259	97.7	98.5	97.5	94.4	97.0
World-action models								
F1 [35]	4B	✓	399	98.2	97.8	95.4	91.3	95.7
Fast-WAM [53]	6B	×	486	98.2	**100.0**	97.0	95.2	97.6
Cosmos-Policy [27]	2.1B	×	1413	98.1	**100.0**	98.2	97.6	98.5
Motus [4]	8B	✓	3231	96.8	99.8	96.6	97.6	97.7
LingBot-VA [30]	5.5B	✓	4482	98.5	99.6	97.2	**98.5**	98.5
LaWAM [10]	2.3B	✓	187	**99.4**	99.6	**98.4**	97.0	**98.6**
DeltaWAM (Ours)	**0.725B**	×	**142.1**	92.6	97.0	95.8	85.8	92.8

Training and Evaluation Protocol For simulation, we first fine-tune the pretrained DeltaWorld predictor on LIBERO demonstrations, and then freeze it to train suite-specialized policies. The entire training pipeline consumes merely 256 GPU hours on two H100 GPUs, dramatically lowering the training barrier compared to prior WAMs. At test time, DeltaWorld samples a future delta sequence and the action expert generates a 16-step action chunk with ten Euler steps, executing eight actions before replanning. Following the standard protocol, on LIBERO and LIBERO-Pro we evaluate each task 50 times and run the test with 3 different random seeds. Architecture, optimization, and evaluation details appear in Appendix A.

Real-World Manipulation Setup For the real-world evaluation, we deploy DeltaWAM on an AgileX PiPER six-DoF robotic arm. We use a 150-episode pick-and-place manipulation dataset collected at 30 Hz, with RGB observations from front and wrist cameras together with 14-dimensional robot states. The policy predicts a seven-dimensional action consisting of six absolute joint-position targets and a gripper command.

4.2 Main Performance and Efficiency

Table 1 jointly compares success rates, model size and model-only query latency across the four standard LIBERO suites. Latency results mainly come from LaWAM [10] and our measurement strictly follows its evaluation protocol. DeltaWAM achieves an average success rate of 92.8%. In both LIBERO-Spatial and LIBERO-Long, the failure cases mainly stem from instability during precision contact—the robot successfully puts the item on the plate, yet the placement is off-center. This is an expected consequence of the architecture, since highly compressed delta tokens may filter out the fine-grained spatial nuances necessary for precise alignment.

DeltaWAM improves over small reactive policies but remains below the strongest pretrained VLA and WAM baselines. Its principal advantage in this comparison is the combination of manipulation performance and a much smaller computational footprint, rather than the highest success rate. Note that DeltaWAM is trained on LIBERO from scratch, which puts it at a disadvantage compared with policies pretrained on enormous robot datasets. Compared with other baseline models, our model is extremely lightweight with low inference costs. DeltaWAM uses 0.725B parameters in total, including 120.3M trainable parameters, and achieves an average latency of 142.1 ms per chunk after querying 100 times on an A100. The peak GPU memory at inference is about 3.86 GB and the estimated cost is 0.584 TFLOPs per action.

4.3 Procedural Robustness of DeltaWAM

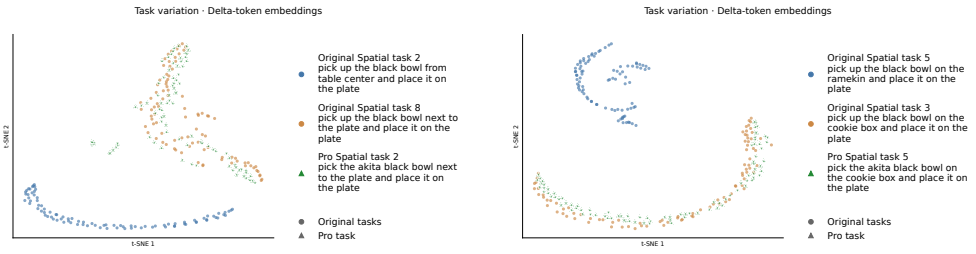
Task perturbations in the LIBERO-PRO benchmark alter task-relevant factors such as the target object, target receptacle, spatial relation, or instruction semantics, requiring the policy to correctly interpret the changed goal rather than simply reproduce behaviors associated with the original task.

They therefore evaluate whether a model can generalize its learned visuomotor skills to novel task configurations and remain responsive to changes in task intent. As shown in Table 2, many baselines' success rates under task perturbations are close to 0%, which stands in sharp contrast with their high success rates on the original LIBERO tasks. This indicates that very likely these models' success stems from mechanical memorization of training scenarios rather than genuine acquisition of transferable task-solving strategies.

Table 2: Success rate (%) under the LIBERO-Pro task perturbation, broken down by the four standard task suites. Official benchmark baselines follow Zhou et al. [57]. We evaluated the results for Fast-WAM and Lingbot-VA ourselves.

Method	Spatial	Object	Goal	Long	Average
OpenVLA [26]	0.0	0.0	0.0	0.0	0.0
π_0 [6]	0.0	0.0	0.0	0.0	0.0
$\pi_{0.5}$ [5]	1.0	1.0	0.0	1.0	0.75
MolmoAct [28]	0.0	0.0	0.0	6.0	1.5
NORA [24]	0.0	0.0	0.0	0.0	0.0
x-VLA [55]	0.0	8.0	9.0	**10.0**	6.75
Fast-WAM [53]	8.0	6.6	6.2	7.6	7.1
LingBot-VA [30]	2.0	1.0	1.0	0.0	1.0
DeltaWAM (Ours)	**42.2**	**10.2**	**9.2**	9.8	**17.85**

In comparison, our DeltaWAM is more robust to task perturbations, especially on LIBERO-Spatial. Figure 5(a,b) provides a qualitative view of how the predicted representation responds to task variations. In both examples, the delta tokens of the perturbed task align more closely with those of a training task that shares the same underlying manipulation requirement, despite differences in task configuration and resulting motion trajectories, than with those of its nominal source task. This pattern demonstrates DeltaWAM's procedural robustness: it sticks to the task-relevant dynamics, rather than trajectories tied with a specific scene layout. This robustness arises from the representation itself: by filtering out shared context and merely focusing on dynamic changes, delta tokens remain more invariant across different task configurations that share the same underlying manipulation requirement. The evaluation results on LIBERO-PLUS in Appendix B further prove this.



(a) LIBERO-Pro: Spatial task 2 (b) LIBERO-Pro: Spatial task 5

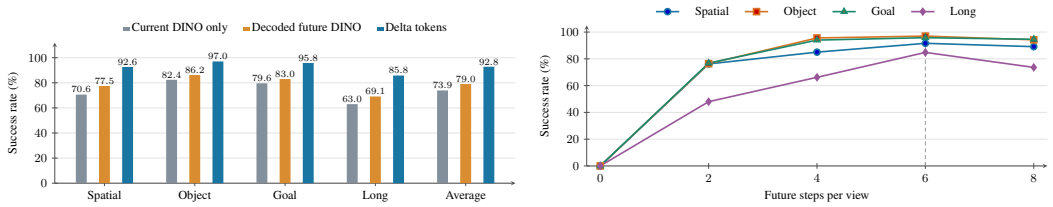
Figure 5: **Delta-token structure under LIBERO-Pro task variations.** The two t-SNE projections compare predicted delta tokens from a perturbed task (green triangles) with tokens from its nominal original task (blue) and an original task whose task requirements match the perturbation (orange).

4.4 Representation Benefits: Ablate The Role of Delta Tokens

To fairly evaluate the importance of predicted delta tokens for task performance, we construct two model variants. The *current-DINO-only* variant bypasses the DeltaWorld rollout and conditions the policy only on DINO patches of the current observation, while the *decoded-future-DINO* variant retains DeltaWorld's predictions but decodes every future delta token back into the DINO latent. The two model variants receive the same training and inference configuration as the original DeltaWAM, and the only difference is what the action expert conditions on. As shown in Figure 6(a), performance drops by 18.9 and 13.8 percentage points, respectively. The *current-DINO-only* variant has the lowest average success rate of 73.9%, suggesting the necessity of future predictions. The comparison between the *decoded-future-DINO* variant and the original DeltaWAM indicates that delta tokens

are more action-forcing by exposing dynamic changes. Compared with the *decoded-future-DINO* variant, DeltaWAM reduces future tokens by roughly $1{,}024\times$. Therefore we do not need to model the full state at each future timestep, and a change-centric representation is actually more compact and more beneficial for policy learning.

To assess how reliant DeltaWAM is on predicted delta tokens, we further conduct a test-time ablation. During training, the future step is set to six, which corresponds to twelve future action steps owing to temporal subsampling. Figure 6(b) compares success rates on the standard LIBERO benchmark by varying the number of predicted delta tokens. Removing all future tokens reduces success rates to 0% on all four suites, showing that the trained policies rely strongly on predicted futures. We observe that in the absence of predictive guidance, the policy degenerates into aimless reaching and grasping behaviors. With only two future steps per view, success recovers to 76.2%, 76.4%, 76.8%, and 48.0% on Spatial, Object, Goal, and Long. Longer foresight is not uniformly beneficial—there is a trade-off between additional future context and prediction error. The analysis of accumulating errors in DeltaWorld's predictions and the ablation study of training policies with the full-horizon future are provided in the appendix.



(a) Performance Comparison (b) Test-time Ablation Results

Figure 6: **Training and Test Ablations.** (a) Success rates of three models across the four task suites on LIBERO. Current DINO only, decoded future DINO, and delta tokens use 2,048, 14,336, and 2,060 visual tokens in total for one action chunk, respectively. (b) Comparison of success rates on LIBERO as the number of predicted future steps per camera view varies. Each step contributes one delta token per view, and the dashed line marks the default six-step horizon.

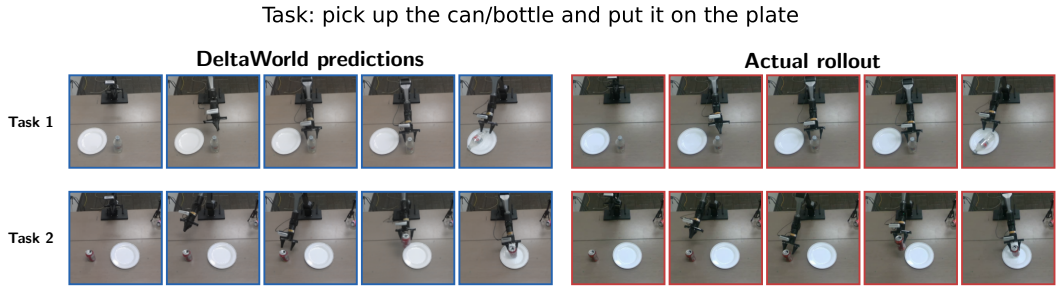


Figure 7: **Real-world deployment** We conduct real-world experiments on a Piper robotic arm. The task is to pick up the bottle/can and place it on the plate. DeltaWorld first predicts the delta tokens, then the Action DiT predicts the action chunk for execution. The frozen delta tokenizer decoder and DINO-to-RGB decoder is applied for DeltaWorld's predictions.

5 Conclusion

In this work, we propose DeltaWAM, a fast and lightweight World-Action model that shifts the prediction unit to a single compact delta token. Experiments on LIBERO show that this change-centric representation supports strong manipulation performance with low latency and provides more effective dynamics priors than full-scene future representations. Its robustness under procedural task perturbations further indicates that delta tokens capture transferable task-relevant dynamics. These results suggest that efficient World-Action modeling need not reconstruct the future in full: predicting merely on what changes not only significantly reduces spatio-temporal redundancy and computational overhead, but also facilitates stronger policy learning.

AI USE STATEMENT

In accordance with the ICLR policy on generative AI tools, we report that large language models (specifically OpenAI ChatGPT and Gemini) were utilized to assist with the following aspects of this research: (1) **Writing Assistance and Polishing:** Generative AI was employed for grammatical refinement, stylistic polishing, and LaTeX typesetting. (2) **Literature Retrieval and Discovery:** AI tools assisted in exploratory bibliographic searches and summarizing related works during the initial literature survey. (3) **Research Ideation and Execution:** LLMs were consulted as conversational sounding boards during preliminary research brainstorming, and provided coding assistance for experimental setup and evaluation scripts. (4) **Drafting Paper Sections:** Initial drafts of descriptive sections were outlined with AI assistance and subsequently heavily rewritten, expanded, and synthesized by the authors.

Crucially, all fundamental conceptual formulations (such as change-centric delta-token foresight), algorithmic architectures, mathematical derivations, codebase implementations, and empirical benchmark evaluations were conceived, verified, and executed under the rigorous supervision and intellectual control of the authors. All generated text, bibliographic references, and empirical claims were independently checked and validated against source literature. The authors take full and sole responsibility for the final contents, integrity, and scientific claims presented in this work.

ETHICS STATEMENT

Physical experiments in this paper adhered strictly to standard laboratory safety protocols, including emergency stop mechanisms and restricted robot joint limits. The public datasets utilized in this work (Kinetics-700 and LIBERO) are open-source and do not contain sensitive personal or proprietary information.

REPRODUCIBILITY STATEMENT

To ensure the thorough reproducibility of our empirical results, the algorithmic formulations of DeltaWorld adaptation and DeltaWAM policy training are presented in Section 3, with experimental protocols and evaluation metrics detailed in Section 4. Comprehensive implementation specifications—including detailed neural network architectures, hyperparameter selections, dual-view observation and action spaces, temporal subsampling strategies, and multi-stage training schedules—are provided in Appendix A. Complete, fully anonymized source code are included directly in the accompanying Supplementary Material zip archive.

REFERENCES

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: A visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, 2022.

[2] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

[3] Apratim Bhattacharyya, Bernt Schiele, and Mario Fritz. Accurate and diverse sampling of sequences based on a "best of many" sample objective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[4] Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. *arXiv preprint arXiv:2512.13030*, 2025.

[5] Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Proceedings of the 9th Conference on Robot Learning*, pp. 17–40, 2025.

[6] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.010.

[7] Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. UniVLA: Learning to act anywhere with task-centric latent actions. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.014.

[8] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the Kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019.

[9] 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. *arXiv preprint arXiv:2506.21539*, 2025.

[10] Jialei Chen, Kai Wang, Kang Chen, Shuaihang Chen, Feng Gao, Wenhao Tang, Zhiyuan Li, Weilin Liu, Zhuyu Yao, Boxun Li, Yuanbo Xu, and Chao Yu. LaWAM: Latent world action models for efficient dynamics-aware robot policies. *arXiv preprint arXiv:2606.15768*, 2026.

[11] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023.

[12] Maurizio Corbetta and Gordon L. Shulman. Control of goal-directed and stimulus-driven attention in the brain. *Nature Reviews Neuroscience*, 3(3):201–215, 2002. doi: 10.1038/nrn755.

[13] Robert Desimone and John Duncan. Neural mechanisms of selective visual attention. *Annual Review of Neuroscience*, 18:193–222, 1995. doi: 10.1146/annurev.ne.18.030195.001205.

[14] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. In *International Conference on Learning Representations*, 2017.

[15] Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: A progressive robustness benchmark for visual-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38574–38583, 2026.

[16] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems*, 2016.

[17] Melvyn A. Goodale and A. David Milner. Separate visual pathways for perception and action. *Trends in Neurosciences*, 15(1):20–25, 1992. doi: 10.1016/0166-2236(92)90344-8.

[18] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, 2018.

[19] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, 2019.

[20] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

[21] Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024.

[22] Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *International Conference on Machine Learning*, 2025.

[23] Jiakai Huang, Zhongbo Wu, Zheng Zhang, Zihan Wang, Shan You, and Tao Huang. Foresight without seeing: Latent futures for world action models. *arXiv preprint arXiv:2608.11605*, 2026.

[24] Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U-Xuan Tan, Navonil Majumder, and Soujanya Poria. NORA: A small open-sourced generalist vision language action model for embodied tasks. *arXiv preprint arXiv:2504.19854*, 2025.

[25] Tommie Kerssies, Gabriele Berton, Ju He, Qihang Yu, Wufei Ma, Daan de Geus, Gijs Dubbelman, and Liang-Chieh Chen. A frame is worth one token: Efficient generative world modeling with delta tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[26] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, et al. OpenVLA: An open-source vision-language-action model. In *Proceedings of the 8th Conference on Robot Learning*, pp. 2679–2713, 2025.

[27] 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. *arXiv preprint arXiv:2601.16163*, 2026.

[28] Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, Winson Han, Wilbert Pumacay, Angelica Wu, Rose Hendrix, Karen Farley, Eli VanderBilt, Ali Farhadi, Dieter Fox, and Ranjay Krishna. MolmoAct: Action reasoning models that can reason in space. *arXiv preprint arXiv:2508.07917*, 2025.

[29] Fenghao Lei, Zhixiong Huang, Long Yang, Jiabao Chen, Peilin Huang, Han Fu, Zhuo Li, and Xiaoxue Ren. DELE-w0.5: Inferring action from future latent state for robotic manipulation. *arXiv preprint arXiv:2608.22067*, 2026.

[30] Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

[31] 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. *arXiv preprint arXiv:2606.08242*, 2026.

[32] Yihan Lin, Jiawei He, Shifeng Bao, Chen Zhao, Yang Li, Xiaobo Wang, Yan Wang, Cheng Chi, and Jing Zhang. JEPA-WAM: Learning vision-language-action policies with joint-embedding world modeling. *arXiv preprint arXiv:2608.09381*, 2026.

[33] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

[34] Hao Luo, Wanpeng Zhang, Yicheng Feng, Sipeng Zheng, Haiweng Xu, Chaoyi Xu, Ziheng Xi, Yuhui Fu, and Zongqing Lu. Being-H0.7: A latent world-action model from egocentric videos. *arXiv preprint arXiv:2605.00078*, 2026.

[35] Qi Lv, Weijie Kong, Hao Li, Jia Zeng, Zherui Qiu, Delin Qu, Haoming Song, Qizhi Chen, Xiang Deng, and Jiangmiao Pang. F1: A vision-language-action model bridging understanding and generation to actions. *arXiv preprint arXiv:2509.06951*, 2025.

[36] Jiangran Lyu, Kai Liu, Xuheng Zhang, Haoran Liao, Yusen Feng, Wenxuan Zhu, Tingrui Shen, Jiayi Chen, Jiazhao Zhang, Yifei Dong, et al. LDA-1B: Scaling latent dynamics action model via universal embodied data ingestion. *arXiv preprint arXiv:2602.12215*, 2026.

[37] NVIDIA GEAR Team. GR00T N1.6: An improved open foundation model for generalist humanoid robots, 2025. URL https://research.nvidia.com/labs/gear/gr00t-n1_6/.

[38] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[39] Jonas Pai, Liam Achenbach, Victoriano Montesinos, Benedek Forrai, Oier Mees, and Elvis Nava. mimic-video: Video-action models for generalizable robot control beyond VLAs. *arXiv preprint arXiv:2512.15692*, 2025.

[40] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

[41] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *AAAI Conference on Artificial Intelligence*, 2018.

[42] 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. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.012.

[43] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[44] John H. Reynolds and Leonardo Chelazzi. Attentional modulation of visual processing. *Annual Review of Neuroscience*, 27:611–647, 2004. doi: 10.1146/annurev.neuro.26.041002.131039.

[45] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

[46] Florian Strub, Mathieu Seurin, Ethan Perez, Harm de Vries, Jeremie Mary, Philippe Preux, Aaron Courville, and Olivier Pietquin. Visual reasoning with multi-hop feature modulation. In *European Conference on Computer Vision*, 2018.

[47] Yue Su, Sijin Chen, Haixin Shi, Mingyu Liu, Zhengshen Zhang, Ningyuan Huang, Weiheng Zhong, Zhengbang Zhu, Yuxiao Liu, and Xihui Liu. World guidance: World modeling in condition space for action generation. *arXiv preprint arXiv:2602.22010*, 2026.

[48] Stefan Treue and John H. R. Maunsell. Attentional modulation of visual motion processing in cortical areas MT and MST. *Nature*, 382(6591):539–541, 1996.

[49] David Whitney, David A. Westwood, and Melvyn A. Goodale. The influence of visual motion on fast reaching movements to a stationary object. *Nature*, 423(6942):869–873, 2003. doi: 10.1038/nature01693.

[50] Fan Yang, Yuting Su, Xiaobo Wang, Yuncheng You, Fugui Fan, Yuting Wu, Minghui Wu, Chenxu Zhao, Jiahong Ning, and Peiguang Jing. LiLa-WAM: Lightweight latent reasoning world-action model for robotic manipulation. *arXiv preprint arXiv:2608.03701*, 2026.

[51] Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, et al. GigaWorld-Policy: An efficient action-centered world–action model. *arXiv preprint arXiv:2603.17240*, 2026.

[52] Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

[53] Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.

[54] Zhanguang Zhang, Zhiyuan Li, Behnam Rahmati, Rui Heng Yang, Yintao Ma, Amir Rasouli, et al. Do world action models generalize better than VLAs? a robustness study. *arXiv preprint arXiv:2603.22078*, 2026.

[55] Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025.

[56] Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 79115–79135, 2025.

[57] Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. LIBERO-PRO: Towards robust and fair evaluation of vision-language-action models beyond memorization. *arXiv preprint arXiv:2510.03827*, 2025.

[58] Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. In *Robotics: Science and Systems*, 2025.

[59] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the 7th Conference on Robot Learning*, pp. 2165–2183, 2023.

APPENDIX

A IMPLEMENTATION DETAILS

Data and inputs. We use the two LIBERO RGB views (`agentview` and `eye_in_hand`) at 512×512 resolution. Four context frames per view are sampled at a two-frame stride from 20 Hz demonstrations. The policy receives a nine-dimensional robot state and eight executed actions, and predicts 16 seven-dimensional actions. DeltaWorld is adapted on the 6,500 demonstrations from LIBERO-Spatial, Object, Goal, LIBERO-90, and LIBERO-10 with suite-balanced sampling.

Architecture. Frozen DINOv3 ViT-B/16 produces a 32×32 patch grid per view, and DeltaTok compresses each transition to one 768-dimensional token per view. DeltaWorld and Action DiT each use 12 transformer blocks, width 768, and 12 attention heads. Current DINO patch anchors and predicted delta tokens receive camera, temporal, and token-type embeddings after projection into the action hidden space. Layer-wise KV caches for the Action DiT are reused across flow steps in the generation of one action chunk.

Two-stage training. We adapt the Kinetics-pretrained DeltaWorld predictor while keeping DINOv3, DeltaTok, and T5-Base frozen. Camera embeddings and zero-initialized gated language modulation are trainable. Subsampling is applied—each delta step corresponds to two action steps. In the policy-training stage, the adapted DeltaWorld is frozen. During this stage, DeltaWorld first samples eight future branches, then the Action DiT conditions on the minimum-error branch with probability 0.5 and on a uniformly-sampled branch otherwise.

Table 3: Training Hyperparameters.

Setting	**DeltaWorld adaptation**	**Action DiT training**
Optimizer	AdamW	AdamW
Learning rate	10^{-4}	5×10^{-5}
Weight decay	0.4	0.01
Warmup steps	1,000	500
Batch size / GPU	32	2
Gradient accumulation	32	2
Gradient clip norm	0.5	1.0
Future candidates	256 (Best-of-Many)	8 rollouts
Future horizon / view	6 delta steps	6 delta steps

Inference and evaluation. At test time, DeltaWorld samples one trajectory without oracle selection. Action DiT uses ten Euler steps, executes eight actions per query, and temporally ensembles overlapping chunks with age-decay 0.1. Continuous end-effector actions are clipped to $[-1, 1]$ and the gripper is thresholded to $\{-1, +1\}$. Both in standard LIBERO and LIBERO-PRO we conduct 50 trials per task (500 per suite) and reports terminal success under three different random seeds.

B LIBERO-PLUS EVALUATION

LIBERO-Plus evaluates robustness to controlled visual and embodiment distribution shifts [15]. We report success rates under camera, robot, lighting, and layout perturbations, performing one trial per perturbation instance across all test suites.

Table 4: Success rates (%) across selected LIBERO-Plus perturbation suites. Published baseline results are taken from the official benchmark leaderboard and the WAM robustness study [15, 54].

Scope / Method	Camera	Robot	Light	Layout
OpenVLA [26]	0.8	3.5	8.1	28.5
NORA [24]	2.2	37.0	45.7	62.1
WorldVLA [9]	0.1	27.9	43.7	38.0
UniVLA [7]	1.8	46.2	69.0	31.9
π_0 [6]	13.8	6.0	**85.0**	**68.9**
π_0-FAST [42]	65.1	21.6	73.2	68.8
Fast-WAM [53]	16.4	44.5	78.2	60.7
DeltaWAM (Average)	**66.9**	**62.5**	84.2	64.1

Across suites, DeltaWAM achieves 66.9%, 62.5%, 84.2%, and 64.1% success under camera, robot, lighting, and layout perturbations, respectively. Figure 8 shows that under lighting perturbations, the delta representations remain close to those of the original tasks. This provides a new perspective on improving robustness besides collecting more diverse data—we can make future representations more invariant to environmental perturbations. Delta tokens already partially show this property by focusing on dynamic changes.

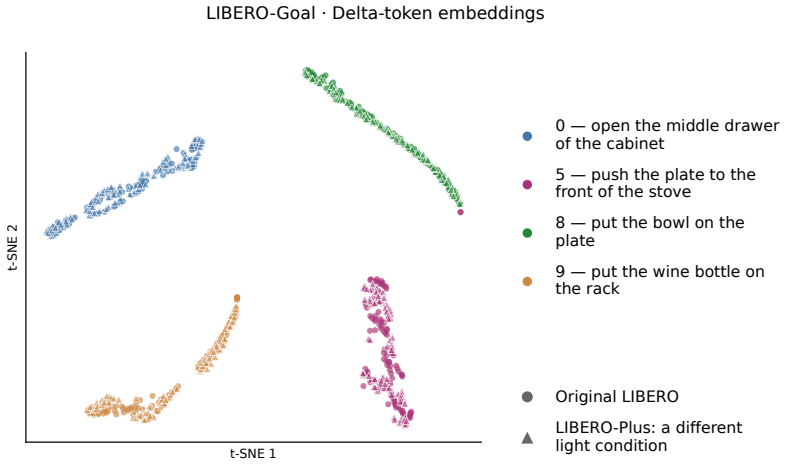


Figure 8: **Delta-token structure under LIBERO-Plus lighting variation.** LIBERO-Goal task embeddings under original light (circles) and a changed light condition (triangles).

C Additional Representation Analysis

C.1 Analysis of DeltaWorld's Future Predictions

We evaluate decoded predictions against future DINO features, using six future steps across all four suites. Figure 9 shows that prediction–future similarity remains high as the horizon grows, while both predicted and observed futures become less similar to the current frame. Thus, the predicted representations capture temporal change rather than simply remaining near the initial observation.

Figure 10 provides a complementary comparison with a static baseline: predicted features have lower MSE than copying the current frame at every displayed horizon in all four suites. Unavoidably, the MSE of DeltaWorld's predictions gradually increases as the future step becomes larger, which is a natural error accumulation for autoregressive predictions. Because of this accumulating error, training the policy with 6 future steps rather than 8 future steps(which correspond to the full horizon of one action chunk) actually gets better performance, as shown in table5.

Table 5: LIBERO success rates (%) with six versus eight future steps.

Future steps	Spatial	Object	Goal	Long	Average
Default (6)	92.6	97.0	95.8	85.8	92.8
Full horizon (8)	88.8	95.4	94.6	81.2	90.0

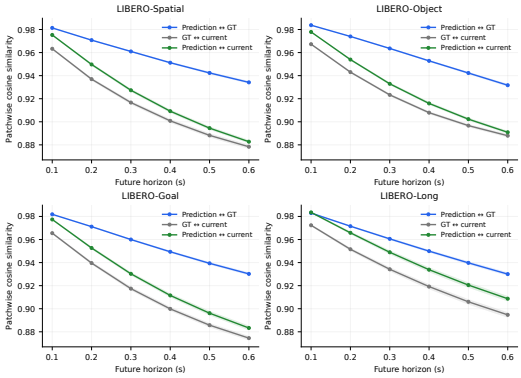


Figure 9: **Future-feature alignment across prediction horizons.** Patchwise DINO cosine similarity on training demonstrations compares predicted futures with ground truth (blue), ground truth with the current frame (gray), and predictions with the current frame (green). Six steps span 0.1–0.6 s.

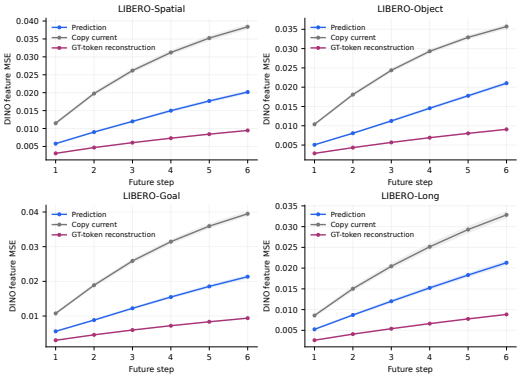


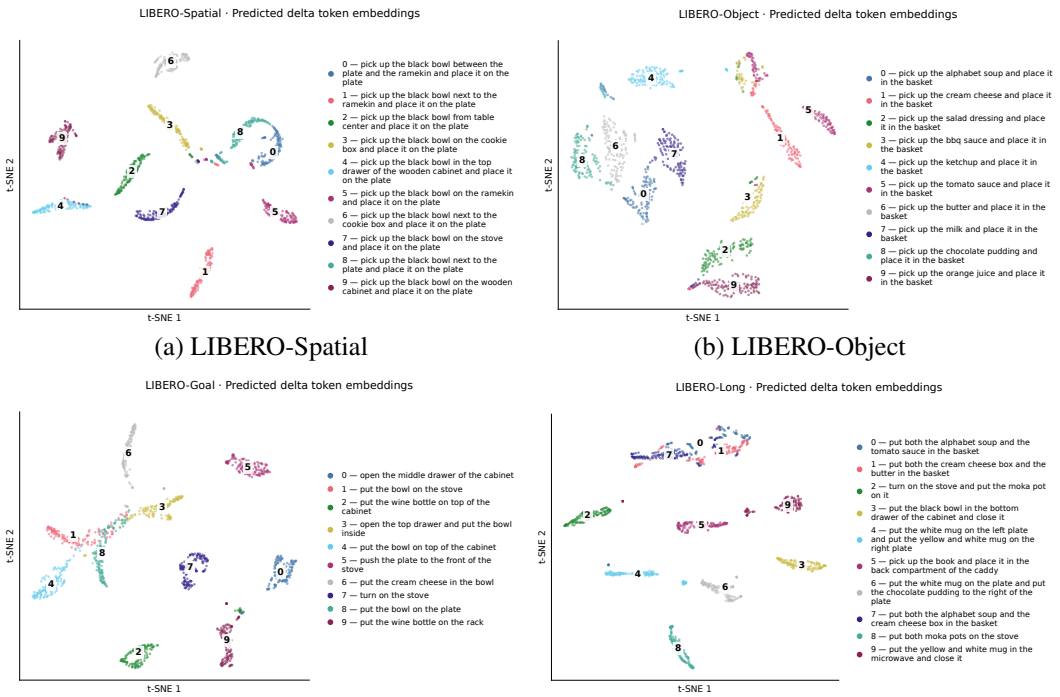
Figure 10: **Future prediction improves over copying the current state.** DINO feature MSE against actual future features for predicted delta-token rollouts (blue), a copy-current baseline (gray), and decoding delta tokens encoded from ground-truth futures (magenta). The latter is a reconstruction reference with access to the actual future.

C.2 Task Structure in Delta-Token Embeddings

Figure 11 visualizes predicted delta tokens across the four standard LIBERO suites. The delta token embeddings of tasks that share the similar underlying semantics are generally closer. These qualitative projections are consistent with task-dependent future representations.

C.3 Analysis of Action Attention to Predicted Future Tokens

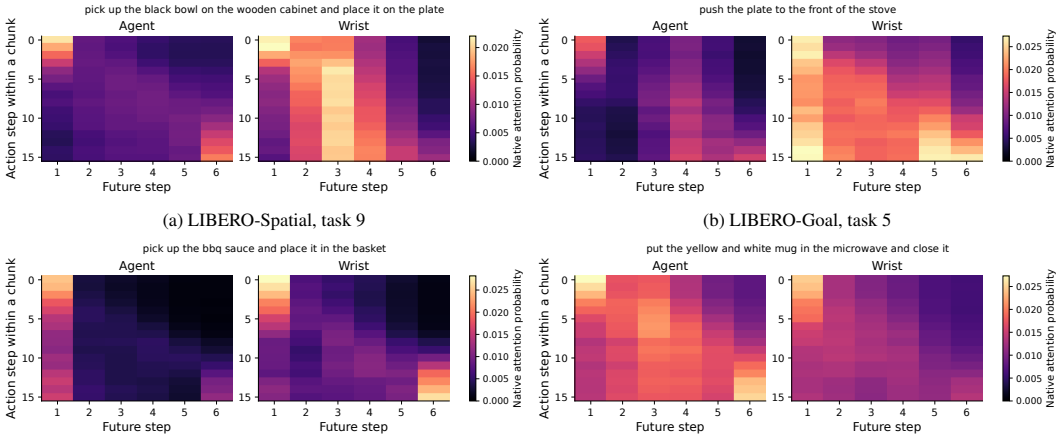
To examine how the action expert uses the predicted sequence, we visualize its native cross-attention probabilities from each action query to the future delta tokens. Figure 12 shows representative examples from all four standard LIBERO suites. Attention is non-uniform across both future steps and action positions, and the patterns differ between agent and wrist views. Early action queries often emphasize near-term tokens, while later queries can shift weight toward more distant future steps.



(a) LIBERO-Spatial (b) LIBERO-Object

(c) LIBERO-Goal (d) LIBERO-10

Figure 11: **Task structure in predicted delta tokens across standard LIBERO suites.** Each point concatenates synchronized agent-view and wrist-view tokens at one future step, yielding one joint embedding per step. Samples come from middle-phase windows of collected rollouts.



(a) LIBERO-Spatial, task 9 (b) LIBERO-Goal, task 5

(c) LIBERO-Object, task 3 (d) LIBERO-10, task 9

Figure 12: **Action DiT attention to predicted future delta tokens.** Native cross-attention probabilities are averaged across Action DiT layers. Rows denote action positions within a generated chunk, columns denote predicted future steps, and agent and wrist tokens are shown separately. Probabilities remain normalized over the complete conditioning context, not only the displayed future tokens.