ForeTime-VLA: Causal Future-Token Distillation from a World Action Model for Conveyor-Belt Manipulation

Siyuan Ma[*,1], Yutian Zhang[*,2], Boshi Zhang[*,1], Qinglian Wu[3],
Jiaqi Zhai[4], Dong Wei[†,4], Xiaojin Huang[†,1]
[1]Tsinghua University, Beijing, China
[2]Shanghai Artificial Intelligence Laboratory, Shanghai, China
[3]Harbin Institute of Technology, Harbin, China
[4]Hangzhou Yunshenchu Technology Co., Ltd. (DEEP Robotics), Hangzhou, China
[*]Equal contribution. [†]Corresponding authors.

***Abstract*— Manipulating moving objects requires a policy to anticipate contact events, yet vision-language-action (VLA) policies are commonly fine-tuned from the current observation alone. World action models (WAMs) learn predictive dynamics, but running a video-scale teacher or explicitly imagining future frames at deployment is costly. We introduce ForeTime-VLA, a dense $\pi_{0.5}$ policy that distills a future-aware, action-equivalent representation from a frozen Fast-WAM-derived teacher while remaining causal at inference. Offline, current and future video latents are compressed into a whitened 64-D target. Online, an eight-frame history encoder predicts this target together with manipulation phase and normalized time-to-transition. Four future tokens and one phase token condition the VLM prefix, while the predicted future and transition horizon condition the action expert. Training retains the original flow-matching action target and adds cosine, relational geometry, phase, time-to-transition, and action-equivalence objectives. On a deduplicated conveyor-belt dataset, we compare models on 768 matched windows per split. Test MAE decreases from 0.134119 to 0.130593 (2.63%; paired-bootstrap 95% CI: 0.82–4.48% improvement), and test L2 decreases by 3.02%, at a 2.46–2.93% latency cost. In quantitative real-robot evaluation, ForeTime-VLAachieves 81.1% stationary and 58.9% slow-moving grasp success, exceeding the next-best reference by 12.2 and 22.2 percentage points, respectively. Across three belt speeds, it completes 44/90 grasps versus 23/90 for $\pi_{0.5}$, including 11/30 versus 2/30 at fast speed. The agreement between offline orientation gains and reduced real-robot contact-pose failures supports causal future-token distillation as an effective way to improve dynamic manipulation without deploying the world-model teacher.**

I. Introduction

Vision-language-action (VLA) models transfer semantic priors from large vision-language backbones to robot control. RT-2 [1], OpenVLA [2], Octo [3], and the π_0/$\pi_{0.5}$ family [4], [5] show that a common policy can be adapted across tasks, scenes, and embodiments. Expressive action decoders—including diffusion [6] and flow matching [7]—further support multimodal, high-dimensional action chunks.

Dynamic conveyor-belt pick-and-place exposes a complementary weakness. The correct motion depends not only on what is visible now, but also on when the object will enter the reachable set, when the gripper should close, and how the end effector should be oriented at contact. A policy trained only to reconstruct demonstrated actions may learn these regularities indirectly, but receives no explicit future-structure supervision.

World models provide such supervision through video prediction. Text-guided video policies [8], generative video-language-action models [9], and recent world action models (WAMs) use predicted future observations to support control. Explicit imagine-then-act pipelines, however, add video generation to the control loop. Fast-WAM [10] instead reports that video modeling is especially useful as a training signal and can be removed from test-time inference. This raises a practical question: *can a pretrained VLA acquire a compact version of that predictive structure without running the WAM at deployment?*

We answer this question with ForeTime-VLA, whose teacher–student pipeline is summarized in Fig. 1. During offline preprocessing, a Fast-WAM/Wan video latent pipeline sees the current observation and eight future offsets. A non-collapsed adapter compresses these features into an action-equivalent teacher code. During policy training, a small causal encoder must predict that code from the preceding eight observations. The predicted code conditions both the slow VLM path and the fast action-expert path of $\pi_{0.5}$, while the original flow-matching construction continues to use the recorded action chunk. Ground-truth actions remain unchanged, avoiding the target-alignment confound of pseudo-label replacement.

Our contributions are:

- a deployable teacher–student formulation that transfers future-conditioned WAM structure into an eight-frame causal history encoder, with no future frame or teacher forward pass at inference;
- dual-path conditioning of $\pi_{0.5}$ with future, phase, and time-to-transition variables, supervised by pointwise, relational, temporal, and action-equivalence objectives; and
- a paired offline evaluation and two quantitative real-robot studies spanning stationary, moving, and speed-stress conditions, with outcome-level analysis connecting future-aware supervision to fewer late and contact-pose failures.

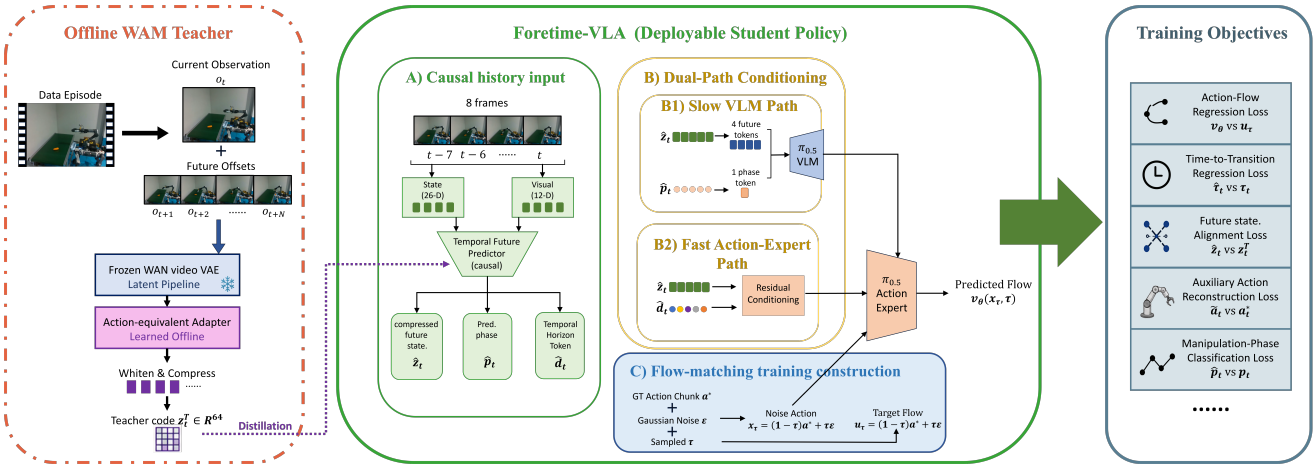

Fig. 1. **ForeTime-VLA overview.** *Left:* the offline teacher encodes the current observation and sampled future frames with a frozen video VAE, then maps them to a compact action-equivalent target. *Center:* the deployable student uses eight causal observations to predict future, phase, and transition horizon variables. These predictions condition both the slow $\pi_{0.5}$ VLM prefix and the fast action-expert suffix, while the original noisy-action and target-flow construction is retained. *Right:* training combines action flow, future alignment, relational geometry, transition, phase, and auxiliary action-reconstruction losses. Future frames and the WAM are absent at test time.

II. Related Work

A. Vision-Language-Action Policies

VLA policies connect visual-language representations to robot actions. RT-1 demonstrated large-scale transformer control from image and language inputs [11], while Gato and PaLM-E placed robot control within broader generalist and embodied multimodal models [12], [13]. The Open X-Embodiment effort subsequently assembled cross-embodiment data and RT-X policies to study transfer across robot platforms [14]. RT-2 casts actions as language tokens [1]; OpenVLA provides an open 7B policy and efficient adaptation recipe [2]; and Octo studies flexible observation and action interfaces [3]. π_0 introduces a VLM-conditioned action expert trained with flow matching [4]. $\pi_{0.5}$ extends this family with heterogeneous co-training and high-level semantic prediction [5]. GR00T N1 couples a vision-language backbone to a diffusion-transformer action head and provides an open recipe for adapting the policy to new embodiments [15]. FAST explores efficient action tokenization for high-frequency control [16]. Complementary manipulation systems study multimodal prompting, multi-task 3-D action prediction, and long-horizon language-conditioned evaluation [17]–[19]. Our work keeps the continuous flow-matching action interface of $\pi_{0.5}$ and adds predictive temporal supervision during full-parameter fine-tuning.

B. Predictive Representations for Control

Latent world models learn compact predictive state and support behavior through imagined rollouts [20]–[22]. In robotics, visual-foresight methods predict future observations for model-based planning [23], [24], while UniSim and Genie explore generative interactive environments at broader visual scales [25], [26]. Video-conditioned policies can use generated futures as plans or predictive features [8], [9], [27]. Fast-WAM separates video co-training from future synthesis and argues that predictive representation learning can retain much of the WAM benefit without test-time imagination [10]. DECOWAM extends this direction to legged mobile manipulation by decoupling camera ego-motion, base actions, and arm actions while distilling an action-equivalent future bottleneck from privileged observations [28]. ForeTime-VLA is complementary: rather than deploying the video-scale WAM, it distills a 64-D future-aware code into a pretrained VLA and predicts the code from compact causal histories.

C. Knowledge Distillation

Classical knowledge distillation transfers the behavior of a costly teacher to a deployable student [29]. Intermediate-feature and relational distillation preserve more than output logits; relational knowledge distillation, for example, matches structure between examples in representation space [30]–[33]. Because intermediate targets can become low-rank or collapsed, variance and covariance regularization provide useful complementary constraints [34], [35]. Our future cosine loss transfers instance-level direction, while a Gram-matrix loss transfers batch geometry. Phase, time-to-transition, and action reconstruction constrain the code to remain useful for manipulation rather than merely matching an arbitrary latent basis.

III. Method

A. Problem Formulation

At time t, the policy receives current RGB observations o_t, a language instruction ℓ, robot state s_t, and an eight-step causal history $\mathcal{H}_t$. It predicts an action chunk $a_{t:t+H-1} \in \mathbb{R}^{H\times D}$, with $H = 20$ and $D = 16$. The base $\pi_{0.5}$ action expert uses conditional flow matching. For actions a, noise $\epsilon \sim \mathcal{N}(0, I)$, and $\tau \in (0, 1)$,

$$x_\tau = (1-\tau)a + \tau\epsilon, \qquad u_\tau = \epsilon - a, \tag{1}$$

and the base loss is

$$\mathcal{L}_{\text{flow}} = \mathbb{E}\left[\|v_\theta(x_\tau, \tau, o_t, \ell) - u_\tau\|_2^2\right]. \tag{2}$$

This noisy-action/target-flow construction is shown in Fig. 1(C) and is preserved when the temporal conditioning is added. Our goal is to add future-sensitive structure without changing a and without using future observations at inference.

B. Offline Action-Equivalent Future Teacher

The left side of Fig. 1 depicts the privileged teacher branch. For every valid training window, we sample the present and eight future frames at offsets $\{2, 4, 7, 9, 12, 14, 17, 19\}$. A frozen Wan2.2 video VAE [36]—the visual latent pipeline used by Fast-WAM [10]—encodes the sequence. Spatial-temporal means and standard deviations produce 96-D current and 96-D future features.

The initially exported quotient codes exhibited insufficient variation. We therefore train a non-collapsed action-equivalent adapter while keeping the video features frozen. Its teacher encoder consumes current feature, future feature, and the 26-D state; a causal auxiliary encoder consumes only current feature and state. Both emit 64-D codes. A shared action decoder reconstructs the normalized 20×16 action chunk, and a future decoder reconstructs the future-minus-current feature. Training combines teacher and causal action MSE, future smooth-L1, teacher–causal cosine, per-dimension variance, off-diagonal covariance, and pairwise action-geometry losses with weights 1.0, 0.5, 0.25, 0.10, 0.20, 0.01, and 0.05, respectively. The resulting teacher codes are whitened per dimension and cached. Thus the privileged future is used to define z_t^T, but never enters the deployed policy.

Figure 2 summarizes the adapter from top to bottom. The frozen latent pipeline extracts c_t from the current observation and f_t from the sampled future frames, while the 26-D state s_t is supplied to the adapter. The privileged teacher encoder uses all three inputs, $\hat{z}_t^T = g_T(c_t, f_t, s_t)$, whereas the causal auxiliary encoder uses only (c_t, s_t) and produces $\hat{z}_t^C = g_C(c_t, s_t)$. Both 64-D codes are sent through a shared action decoder, and the teacher code is additionally sent to a future decoder that reconstructs $f_t - c_t$. Teacher–causal alignment transfers future information to the causal branch; variance and covariance regularization keep every latent dimension active; and pairwise action geometry preserves relative distances between examples. After training, only the per-dimension whitened teacher output z_t^T is cached for causal-student supervision.

C. Causal Temporal Student

Figure 1(A) shows the causal input and its three prediction heads. The deployment history contains eight 26-D normalized states and 12-D visual statistics (per-channel means and standard deviations from two auxiliary cameras). The current full-resolution images still enter the standard $\pi_{0.5}$ vision backbone. For history index i, we compute

$$e_i = \mathrm{swish}(W_s\,\mathrm{LN}(s_i)) + \mathrm{swish}(W_v\,\mathrm{LN}(r_i)) + p_i, \quad (3)$$

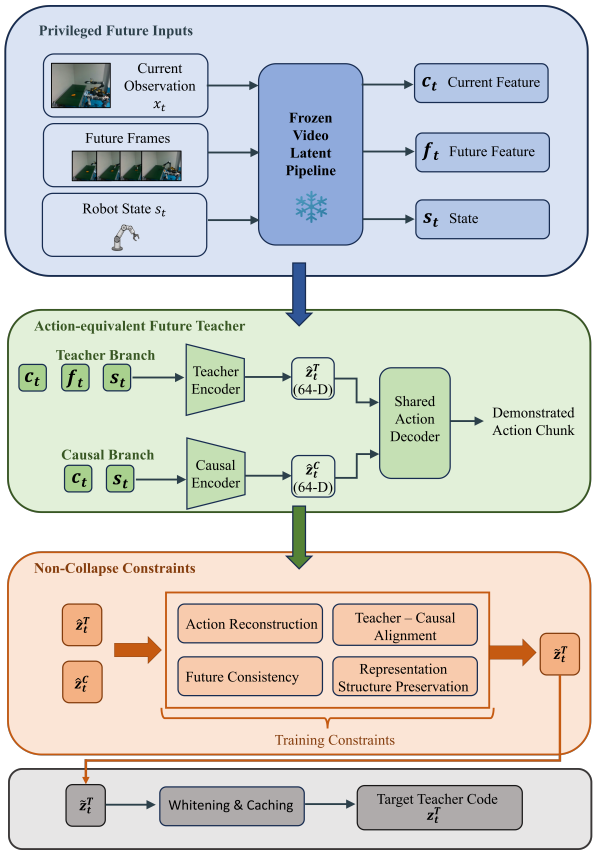

Fig. 2. **Offline action-equivalent future teacher.** The frozen latent pipeline extracts current and future features (c_t, f_t), and the adapter also receives robot state s_t. The privileged encoder uses (c_t, f_t, s_t), whereas the causal branch uses (c_t, s_t). Both codes share an action decoder; the teacher additionally reconstructs the future feature difference. Teacher–causal alignment, variance/covariance regularization, and pairwise action geometry prevent collapse and retain action relevance. The selected 64-D teacher code is whitened and cached as z_t^T.

where p_i is a learned temporal embedding and $e_i \in \mathbb{R}^{256}$. After flattening, a two-layer residual MLP produces

$$q = \mathrm{swish}(W_p\,\mathrm{vec}(e_{1:8})), \quad h_t = \mathrm{LN}(\mathrm{swish}(W_o q) + q). \quad (4)$$

Linear heads predict the compressed future state $\hat{z}_t \in \mathbb{R}^{64}$, four manipulation-phase logits $\hat{p}_t$, and a sigmoid normalized time-to-transition $\hat{d}_t \in [0, 1]$; the latter is the temporal-horizon signal shown in the figure.

The phase target is obtained from gripper-state transitions: approach, grasp/release transition, transport while holding, and place/retract. The time-to-transition target is the clipped, horizon-normalized distance to the next gripper-state transition. These low-cost labels expose event structure that is otherwise implicit in the action chunk.

D. Dual-Path Conditioning

As illustrated in Fig. 1(B), the student conditions the policy through two complementary routes. The slow path maps $\hat{z}_t$ to four 2048-D future tokens. The expected embedding under the predicted phase distribution supplies a fifth token. The five tokens are appended to the $\pi_{0.5}$ prefix. The fast path maps $\hat{z}_t$ and $\hat{d}_t$ to a 1024-D residual that is

added to every action-expert suffix token and its adaptive normalization condition. Learned scalar gates initialize both injections at 0.05, limiting disruption of the pretrained policy early in training.

The added modules contain 1.251M parameters, small relative to the dense Gemma-2B VLM and 311M-parameter action expert. The Fast-WAM teacher, video VAE, and offline adapter are absent from deployment.

E. Training Objective

The right side of Fig. 1 groups the supervision into five functional objectives: action flow, time to transition, future-state alignment, auxiliary action reconstruction, and manipulation-phase classification. The future-state term contains both pointwise and relational components. Let row-normalized predicted and teacher codes in a batch be $\bar{Z}$ and $\bar{Z}^T$. We use pointwise cosine and relational geometry losses,

$$\mathcal{L}_{\cos} = \frac{1}{B}\sum_i (1 - \bar{z}_i^\top \bar{z}_i^T), \tag{5}$$

$$\mathcal{L}_{\text{geo}} = \frac{1}{B^2}\left\|\bar{Z}\bar{Z}^\top - \bar{Z}^T(\bar{Z}^T)^\top\right\|_F^2. \tag{6}$$

Cross-entropy supervises phase, Huber loss with $\delta = 0.1$ supervises time to transition, and a linear decoder from $\hat{z}_t$ reconstructs the normalized action chunk for $\mathcal{L}_{\text{ae}}$. The total objective is

$$\begin{aligned}\mathcal{L} = \mathcal{L}_{\text{flow}} + 0.20\mathcal{L}_{\cos} + 0.02\mathcal{L}_{\text{geo}} + 0.04\mathcal{L}_{\text{phase}} \\ + 0.04\mathcal{L}_{\text{transition}} + 0.05\mathcal{L}_{\text{ae}}.\end{aligned} \tag{7}$$

All $\pi_{0.5}$ parameters and ForeTime-VLA modules are optimized jointly; the recorded action target is never replaced by a teacher action.

F. Design Rationale and Distinction

ForeTime-VLA transfers predictive structure rather than generated pixels. Its privileged branch compresses future video into a code constrained by the demonstrated action, filtering appearance changes that are irrelevant to control. The causal student recovers this code from history alone, retaining the temporal bias of a video model without future synthesis in the control loop. Future and phase tokens expose the forecast to the semantic VLM path, while the fast residual gives the action expert direct access to the predicted code and transition horizon. This action-equivalent, event-aware interface—together with an unchanged action target—distinguishes ForeTime-VLA from attaching a video generator to the policy and makes it complementary to flow matching.

IV. Experimental Setup

A. Dataset

The conveyor-belt corpus contains mobile-manipulator demonstrations of picking boxes, corn, and watermelons under variations in belt speed, viewpoint, and scene layout. Auditing found 487 HDF5 files, of which one was unreadable and 28 were exact duplicates. The remaining 458 episodes contain 96,512 frames (1.79 hours at 15 Hz). We use a deterministic, duplicate-free split of 402/31/25 episodes for train/validation/test, yielding 77,125/6,075/4,610 valid action windows. All blank instructions are replaced with "Pick up and place the moving object from the conveyor belt." Figure 3 summarizes the realized collection rather than the larger acquisition target: the outer taxonomy retains every on-disk group, while all aggregate counts use only canonical episodes.

B. Models and Protocol

The baseline and ForeTime-VLA use the same $\pi_{0.5}$ backbone and base initialization. The baseline retains the original current-observation interface, whereas ForeTime-VLA adds the temporal module and cached auxiliary supervision described in Sec. III.

The cross-family comparison includes GR00T N1.6-3B [15], StarVLA QwenGR00T, and SmolVLA-450M [37]. All references use the same duplicate-free data split and preserve the three camera streams, 26-D state, 16-D absolute action, 20-step horizon, and instruction, providing a consistent data and action contract across model families.

The shared backbone, initialization, action targets, and matched windows make the direct $\pi_{0.5}$ comparison a controlled test of temporal conditioning. The additional families are not parameter-matched ablations, but test whether the benefit remains competitive beyond a single baseline family.

For each split, the evaluator draws 768 windows. All five models receive exactly matched episode/start indices and raw target action chunks. We report mean squared error (MSE), mean absolute error (MAE), and mean per-timestep action-vector L2 error in the unnormalized 16-D action space; lower is better. Confidence intervals for the controlled $\pi_{0.5}$ comparison use 20,000 paired bootstrap resamples; relative improvement is $100(m_{\text{base}} - m_{\text{ours}})/m_{\text{base}}$. Latency is measured on the same A100.

C. Real-Robot Protocol

The reported spreadsheet contains two closed-loop real-robot evaluations. The first compares all five policies on a stationary task and a slow-speed moving task, with 90 trials per method in each task. The stationary task reports grasp success. For the moving task, every trial is assigned to exactly one of four outcomes: grasp success, early failure, late failure, or contact-pose error; the four counts therefore sum to 90 for each method.

The second evaluation isolates sensitivity to belt speed. It compares the $\pi_{0.5}$ baseline and ForeTime-VLA over slow, medium, and fast settings, with 30 trials per method at each speed. We report the exact workbook counts and their corresponding rates.

V. Results

A. Aggregate Reconstruction

Table I reports validation and test MAE/L2, the metrics for which the paired evidence is strongest. Validation and test MAE improve by 2.96% and 2.63%, respectively, with 95% intervals above zero, while test L2 improves by 3.02%.

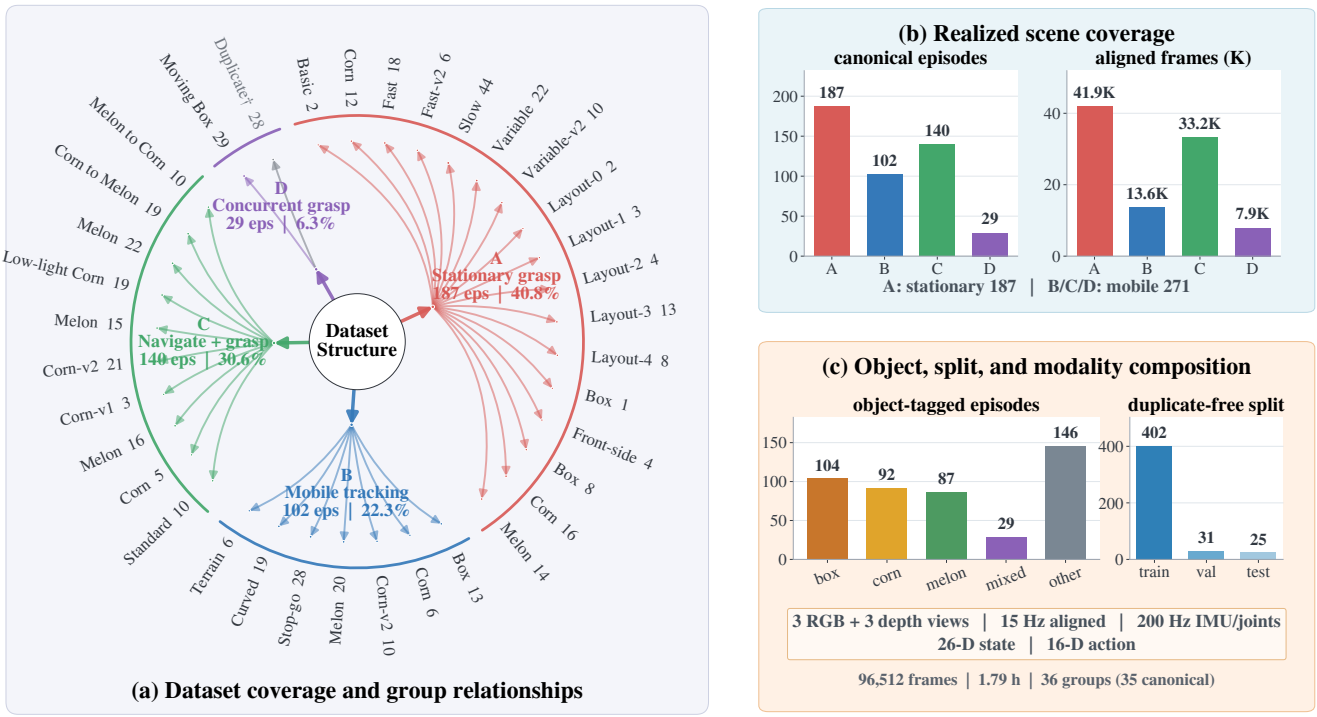

Fig. 3. **Dataset structure and realized coverage.** *Left:* the collection taxonomy spans stationary grasp, mobile tracking, navigate-then-grasp, and concurrent-grasp settings; outer labels give canonical episode counts. *Upper right:* episode and aligned-frame coverage by scene. *Lower right:* object composition, the duplicate-free split, and the recorded multimodal contract. Aggregate statistics exclude 28 exact duplicate files; the remaining 458 episodes contain 96,512 aligned frames and are split 402/31/25 for train/validation/test.

The consistent validation/test gains show that the distilled future representation improves action reconstruction beyond the training distribution, with especially clear benefits for coordinate-wise and action-vector accuracy.

The confidence intervals bound this claim: evidence is positive for MAE on both splits and for test L2, whereas validation L2 and the separately analyzed MSE intervals overlap zero. More importantly, the offline gains agree with independent downstream evidence: orientation improves most, component masking harms transition accuracy, and the closed-loop advantage grows as motion shortens the interception horizon. Agreement across reconstruction, intervention, and execution is stronger evidence for useful temporal foresight than any single aggregate score.

MSE is retained as a complementary view of error concentration. ForeTime-VLA has the best validation MSE (0.249276), while StarVLA has the lowest test MSE (0.237490); the latter does not overturn the broader result because ForeTime-VLA remains best on both test MAE and test L2. In other words, the proposed policy reduces typical coordinate and action-vector error consistently across splits, whereas the MSE ranking is more sensitive to a small number of large residuals. The paired MSE intervals also cross zero on validation and test, so we do not claim a statistically significant MSE advantage. Reporting all three metrics therefore makes the comparison more informative: MSE exposes tail sensitivity, while MAE and L2 capture the stable accuracy gains that align with the robot results.

TABLE I

OFFLINE MSE, MAE, AND L2 OVER 768 MATCHED WINDOWS PER SPLIT. LOWER IS BETTER. THE LOWER PANEL REPORTS PAIRED RELATIVE IMPROVEMENT AND 95% CIS FOR FORETIME-VLA VERSUS THE $\pi_{0.5}$ BASELINE.

Split	Model	MSE	MAE	L2
Val.	$\pi_{0.5}$ baseline	0.256338	0.138628	1.153725
	ForeTime-VLA	**0.249276**	**0.134522**	**1.128938**
	GR00T N1.6	0.278784	0.162035	1.285486
	StarVLA	0.258356	0.163372	1.262874
	SmolVLA	0.257650	0.147754	1.182720
Test	$\pi_{0.5}$ baseline	0.248680	0.134119	1.104541
	ForeTime-VLA	0.244223	**0.130593**	**1.071214**
	GR00T N1.6	0.251833	0.152173	1.183549
	StarVLA	**0.237490**	0.149984	1.157864
	SmolVLA	0.252837	0.141363	1.133300

Metric	Validation gain [95% CI]	Test gain [95% CI]
MSE	+2.76 [−1.92, +7.23]	+1.79 [−1.60, +5.33]
MAE	+2.96 [+0.74, +5.15]	+2.63 [+0.82, +4.48]
L2	+2.15 [−0.65, +4.89]	+3.02 [+0.94, +5.07]

B. Cross-Family References

The GR00T reference has higher MAE and L2 than both $\pi_{0.5}$ models on both splits. Relative to the $\pi_{0.5}$ baseline, its validation and test MAE are 16.89% and 13.46% higher; paired-window bootstrap intervals for the increase are [12.50, 21.62]% and [9.97, 17.14]%. StarVLA's test MAE and L2 are 11.83% and 4.83% higher than the $\pi_{0.5}$ baseline. SmolVLA obtains validation/test MAE of 0.147754/0.141363 and L2 error of 1.182720/1.133300. ForeTime-VLA achieves the lowest MAE and L2 on both splits across all five policies,

TABLE II

POOLED ACTION-GROUP MAE. PARENTHESES SHOW RELATIVE IMPROVEMENT.

Split	Group	Baseline	ForeTime-VLA
Val	base XY	0.016555	0.016176 (+2.30%)
	base yaw	0.008367	0.008367 (+0.00%)
	EE position	0.023643	0.023479 (+0.69%)
	EE RPY	0.478830	**0.459112 (+4.12%)**
	arm	0.095593	0.094838 (+0.79%)
Test	base XY	0.015229	0.015642 (−2.71%)
	base yaw	0.003279	0.003279 (−0.01%)
	EE position	0.023745	0.023499 (+1.03%)
	EE RPY	0.475728	**0.457734 (+3.78%)**
	arm	0.087678	0.087318 (+0.41%)

demonstrating that its predictive temporal structure provides a strong advantage over multiple VLA families rather than only over its direct $\pi_{0.5}$ baseline.

C. Action-Group Analysis

The strongest group-level effect is end-effector roll/pitch/yaw (RPY): validation MAE improves by 4.12% and test MAE by 3.78% (Table II). End-effector position improves by 1.03% on test, and arm error also improves on both splits. The concentration of gains in end-effector orientation shows that future/event supervision is particularly effective at preparing the gripper pose for moving-object contact.

D. Inference-Time Component Ablation

We additionally perform a controlled deployment-path intervention on the ForeTime-VLA checkpoint (Table III). Each row uses the same checkpoint weights; the indicated condition is zeroed or injection path is removed only at inference. We evaluate 96 matched test windows and a critical subset of 48 matched grasp/release-transition windows (phase 1), with identical initial action noise across rows. This smaller study isolates whether the trained policy uses each condition.

TABLE III

OFFLINE INFERENCE-TIME COMPONENT ABLATION. OVERALL METRICS USE 96 MATCHED TEST WINDOWS; TRANSITION METRICS USE 48 MATCHED PHASE-1 WINDOWS. LOWER IS BETTER. ALL FORETIME-VLA VARIANTS SHARE WEIGHTS AND DIFFER ONLY BY COMPONENT MASKING AT INFERENCE.

Method	Overall MAE ↓	Transition MAE ↓	Transition EE-RPY MAE ↓
$\pi_{0.5}$ baseline	0.122124	0.150140	0.501144
ForeTime-VLA w/o future condition	0.128787	0.159982	0.498349
ForeTime-VLA w/o phase/time condition	0.134846	0.160672	0.497727
ForeTime-VLA slow path only	0.136006	0.156587	0.494265
ForeTime-VLA fast path only	0.127140	0.161749	**0.479751**
ForeTime-VLA	**0.118354**	**0.145320**	0.482656

The full model has the lowest overall and transition MAE. Removing the future condition increases transition MAE by 10.1%, while removing phase/time conditioning increases it by 10.6%. Restricting conditioning to either path also degrades transition MAE, supporting complementary slow- and fast-path use. Together, these interventions show that predictive future features, explicit event timing, and dual-path conditioning are all actively used by the trained policy, validating the central architectural choices of ForeTime-VLA.

E. Deployment Efficiency

Pooled validation latency increases from 66.69 to 68.65 ms (+2.93%), and test latency from 66.97 to 68.62 ms (+2.46%). This small overhead is consistent with the 1.251M added parameters and five extra slow-prefix tokens. GR00T takes 133.86 and 132.12 ms on validation and test, respectively, while StarVLA takes 101.72 and 97.24 ms; both use four denoising steps. Thus, ForeTime-VLA combines the strongest reported action accuracy and real-robot performance with substantially lower latency than the cross-family references. This efficiency follows directly from the teacher–student design: the video-scale WAM supplies predictive supervision offline, while deployment requires only a lightweight 1.251M-parameter temporal module and five additional prefix tokens.

F. Quantitative Real-Robot Evaluation

Table IV reports the first closed-loop campaign. ForeTime-VLA attains the highest grasp success in both tasks. On the stationary task, it succeeds in 73/90 trials (81.1%), 11 successes and 12.2 percentage points above the next-best StarVLA result. On the slow-speed moving task, it succeeds in 53/90 trials (58.9%), 20 successes and 22.2 points above the next-best $\pi_{0.5}$ baseline. Relative to $\pi_{0.5}$, the absolute advantage grows from 15.6 points when the target is stationary to 22.2 points when target motion introduces an interception horizon. This widening gap is consistent with the intended role of the distilled future code: its benefit is largest when a current-observation policy must extrapolate where and when contact will occur.

TABLE IV

REAL-ROBOT OUTCOMES FOR STATIONARY AND SLOW-SPEED MOVING GRASP TASKS. ALL ENTRIES ARE PERCENTAGES; HIGHER SUCCESS AND LOWER FAILURE ARE BETTER.

Stationary grasp task

Method	Grasp success
$\pi_{0.5}$	65.56%
GR00T N1.6	46.67%
StarVLA	68.89%
SmolVLA	63.33%
ForeTime-VLA	**81.11%**

Moving grasp task

Method	Success	Early failure	Late failure	Contact-pose error
$\pi_{0.5}$	36.67%	21.11%	26.67%	15.56%
GR00T N1.6	12.22%	27.78%	35.56%	24.44%
StarVLA	28.89%	32.22%	18.89%	20.00%
SmolVLA	32.22%	25.56%	27.78%	14.44%
ForeTime-VLA	**58.89%**	**17.78%**	**16.67%**	**6.67%**

ForeTime-VLA is the only policy with the best rate in all three moving-task failure categories. Relative to $\pi_{0.5}$, early,

late, and contact-pose failures decrease by 15.8%, 37.5%, and 57.1%, respectively. The larger reductions in late and contact-pose errors are especially diagnostic: they occur after approach has begun, when success depends on predicting contact time and preparing gripper orientation. Together with the strongest offline gain in end-effector RPY, this localizes the benefit to contact timing and geometry rather than generic action smoothing.

G. Robustness to Belt Speed

The speed stress test in Fig. 4 strengthens the dynamic-manipulation result. ForeTime-VLA succeeds in 19 versus 13 trials at slow speed, 14 versus 8 at medium speed, and 11 versus 2 at fast speed. Across all three settings, it completes 44/90 grasps compared with 23/90 for $\pi_{0.5}$, adding 21 successful executions. At fast speed, it completes more than five times as many grasps as the baseline.

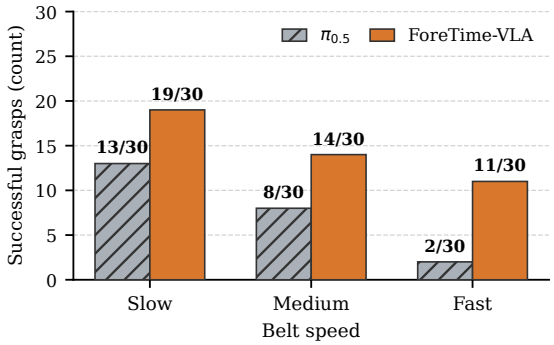

Fig. 4. **Real-robot grasp completions across belt speeds.** Successful grasps out of 30 trials; labels give completed/total trials.

From slow to fast speed, ForeTime-VLA drops from 19 to 11 successes, whereas $\pi_{0.5}$ drops from 13 to 2. Thus the absolute margin widens from six to nine successful trials as the interception window contracts. The deployed policy never observes future frames or runs the WAM, so this robustness must be carried by the distilled causal history and event predictions.

H. Temporal-Task Contribution

An interception error changes both *when* to close and *how* to orient the end effector. ForeTime-VLA represents this coupling through a future code, phase, and time-to-transition predicted from causal history, while all privileged video remains offline. Its advantage over $\pi_{0.5}$ grows from 15.6 percentage points for stationary grasping to 22.2 points for slow motion and widens again at fast speed. Together with the component ablations and orientation gains, this trend supports predictive temporal conditioning, rather than generic action smoothing, as the source of improvement.

I. Qualitative Real-Robot Rollout

Figure 5 shows one successful rollout. The phase trace remains in approach while tracking the moving object. At $t = 9.8$ s, the gripper closes and the phase head assigns 87.3% probability to grasp; the subsequent rise in transport probability yields a coherent approach–grasp–transport sequence rather than a static scene cue.

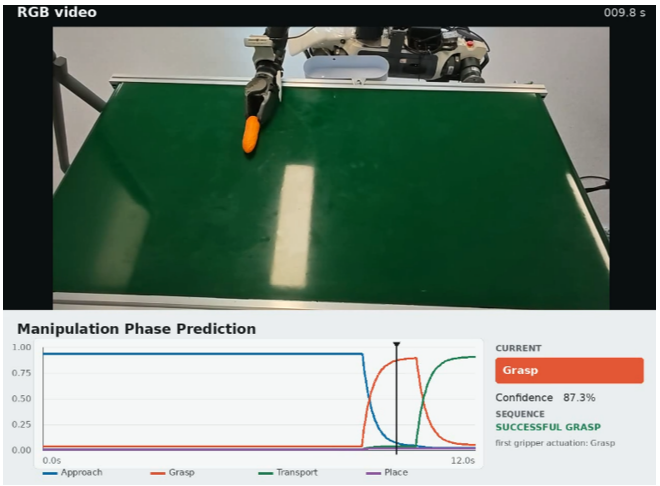

Fig. 5. **Qualitative real-robot grasp and phase prediction.** *Top:* grasp of the moving target. *Bottom:* causal phase probabilities; at 9.8 s, grasp has 87.3% confidence before transport rises after acquisition.

J. Evidence Synthesis

The evidence supports a consistent mechanism. ForeTime-VLA lowers MAE/L2 on both splits; masking the future code, phase/time signal, or either conditioning path worsens transition error; and its robot advantage grows as belt speed shortens the interception horizon. Together, these results connect the gain to temporal conditioning rather than arbitrary capacity. The effect is obtained with less than 3% latency overhead and no video generation at inference. Our claim is therefore narrower than domination of every scalar metric: test MSE favors StarVLA, whereas stable MAE/L2 accuracy, causal deployment, and contact-relevant robustness define the practical advantage of ForeTime-VLA.

VI. Conclusion

We presented ForeTime-VLA, which transfers future-conditioned WAM structure into a causal flow-matching VLA through an action-equivalent code, phase, and transition horizon. It reduces test MAE/L2 by 2.63%/3.02% with under 3% latency overhead, reaches 81.1% stationary and 58.9% slow-moving grasp success, and completes 44/90 speed-stress trials versus 23/90 for $\pi_{0.5}$. The 11/30 versus 2/30 fast speed result shows that compact future-token distillation is most useful when reaction time is limited, without deploying the video teacher. Evidence remains limited to one conveyor-belt embodiment and a fixed offline video backbone; longer-horizon tasks, changing viewpoints, and broader action spaces are needed to test generality. Overall, separating predictive supervision from test-time generation provides the central practical benefit of the design.

REFERENCES

[1] B. Zitkovich, T. Yu, S. Xu, *et al.*, "RT-2: Vision-language-action models transfer web knowledge to robotic control," in *Proceedings of the 7th Conference on Robot Learning*, 2023, pp. 2165–2183.

[2] M. J. Kim, K. Pertsch, S. Karamcheti, *et al.*, "OpenVLA: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[3] Octo Model Team, D. Ghosh, H. Walke, *et al.*, "Octo: An open-source generalist robot policy," *arXiv preprint arXiv:2405.12213*, 2024.

[4] K. Black, N. Brown, D. Driess, *et al.*, "π_0: A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024.

[5] Physical Intelligence, K. Black, N. Brown, *et al.*, "$\pi_{0.5}$: A vision-language-action model with open-world generalization," *arXiv preprint arXiv:2504.16054*, 2025.

[6] C. Chi, Z. Xu, S. Feng, *et al.*, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Robotics: Science and Systems*, 2023.

[7] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," in *International Conference on Learning Representations*, 2023.

[8] Y. Du, M. Yang, B. Dai, *et al.*, "Learning universal policies via text-guided video generation," *arXiv preprint arXiv:2302.00111*, 2023.

[9] H. Wu, Y. Jing, C. Cheang, *et al.*, "Unleashing large-scale video generative pre-training for visual robot manipulation," *arXiv preprint arXiv:2312.13139*, 2023.

[10] T. Yuan, Z. Dong, Y. Liu, and H. Zhao, "Fast-WAM: Do world action models need test-time future imagination?" *arXiv preprint arXiv:2603.16666*, 2026.

[11] A. Brohan, N. Brown, J. Carbajal, *et al.*, "RT-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[12] S. Reed, K. Zolna, E. Parisotto, *et al.*, "A generalist agent," *Transactions on Machine Learning Research*, 2022.

[13] D. Driess, F. Xia, M. S. M. Sajjadi, *et al.*, "PaLM-E: An embodied multimodal language model," in *International Conference on Machine Learning*, 2023, pp. 8469–8488.

[14] Open X-Embodiment Collaboration, "Open X-embodiment: Robotic learning datasets and RT-X models," *arXiv preprint arXiv:2310.08864*, 2023.

[15] J. Bjorck, F. Castañeda, N. Cherniadev, *et al.*, "GR00T N1: An open foundation model for generalist humanoid robots," *arXiv preprint arXiv:2503.14734*, 2025.

[16] K. Pertsch, K. Stachowicz, B. Ichter, *et al.*, "FAST: Efficient action tokenization for vision-language-action models," *arXiv preprint arXiv:2501.09747*, 2025.

[17] Y. Jiang, A. Gupta, Z. Zhang, *et al.*, "VIMA: General robot manipulation with multimodal prompts," in *International Conference on Machine Learning*, 2023, pp. 15 694–15 715.

[18] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Proceedings of the 6th Conference on Robot Learning*, 2023, pp. 785–799.

[19] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, "CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7327–7334, 2022.

[20] D. Ha and J. Schmidhuber, "World models," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[21] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," in *International Conference on Learning Representations*, 2020.

[22] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering diverse domains through world models," *arXiv preprint arXiv:2301.04104*, 2023.

[23] C. Finn and S. Levine, "Deep visual foresight for planning robot motion," in *IEEE International Conference on Robotics and Automation*, 2017, pp. 2786–2793.

[24] F. Ebert, C. Finn, S. Dasari, A. Xie, A. Lee, and S. Levine, "Visual foresight: Model-based deep reinforcement learning for vision-based robotic control," *arXiv preprint arXiv:1812.00568*, 2018.

[25] M. Yang, Y. Du, K. Ghasemipour, J. Tompson, D. Schuurmans, and P. Abbeel, "UniSim: Learning interactive real-world simulators," *arXiv preprint arXiv:2310.06114*, 2023.

[26] J. Bruce, M. Dennis, A. Edwards, *et al.*, "Genie: Generative interactive environments," in *International Conference on Machine Learning*, 2024.

[27] Y. Hu, Y. Guo, P. Wang, *et al.*, "Video prediction policy: A generalist robot policy with predictive visual representations," *arXiv preprint arXiv:2412.14803*, 2024.

[28] S. Ma, B. Zhang, Y. Zhang, *et al.*, "DECOWAM: Decoupled whole-body world-action model for legged mobile manipulation," *arXiv preprint arXiv:2608.20114*, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2608.20114

[29] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[30] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for thin deep nets," in *International Conference on Learning Representations*, 2015.

[31] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *International Conference on Learning Representations*, 2017.

[32] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3967–3976.

[33] Y. Tian, D. Krishnan, and P. Isola, "Contrastive representation distillation," in *International Conference on Learning Representations*, 2020.

[34] A. Bardes, J. Ponce, and Y. LeCun, "VICReg: Variance-invariance-covariance regularization for self-supervised learning," in *International Conference on Learning Representations*, 2022.

[35] L. Jing, P. Vincent, Y. LeCun, and Y. Tian, "Understanding dimensional collapse in contrastive self-supervised learning," in *International Conference on Learning Representations*, 2022.

[36] T. Wan, A. Wang, B. Ai, *et al.*, "Wan: Open and advanced large-scale video generative models," *arXiv preprint arXiv:2503.20314*, 2025.

[37] M. Shukor, D. Aubakirova, F. Capuano, *et al.*, "Smolvla: A vision-language-action model for affordable and efficient robotics," *arXiv preprint arXiv:2506.01844*, 2025.