DEXFORESIGHT: SELF-DISTILLED FORESIGHT FOR DEXTEROUS VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Task-oriented dexterous manipulation requires coordinated arm–hand actions whose quality is determined by the physical and visual consequences that follow them. Demonstration trajectories therefore contain more supervision than the demonstrated actions alone: they also record how objects move, articulate, and appear after each interaction. We introduce DexForesight, a training framework built on a single observation: a demonstrated future can supervise a causal policy in two distinct ways—a predicted future teaches the representation what interaction state to anticipate, while the realized future teaches the action generator how its current prediction should change. For representation learning, we adapt V-JEPA 2 to demonstration trajectories and design an action-conditioned latent predictor with structured arm–hand controls; its predicted future latent supervises learnable query tokens in the vision-language model (VLM) representation. For action generation, we execute the same policy under causal and privileged future context; their velocity difference defines a future-conditioned flow correction learned by a lightweight residual adapter. Future observations and simulator object states are used only during training and are removed at deployment. Across the 11 single-arm and bimanual tasks of DexJoCo, DexForesight improves the average success rate of the DexJoCo $\pi_{0.5}$ baseline by 10.1 percentage points (62.6% vs. 52.5%). Both supervision paths improve the policy in isolation, and their combination performs best; combining physical and visual privileged future context also achieves the strongest average performance.

1 INTRODUCTION

Task-oriented dexterous manipulation requires more than predicting a plausible instantaneous motion: the interaction must evolve toward the intended object state, whether through sustained tool alignment, a preserved grasp, or two hands cooperating toward a shared configuration (Wang et al., 2026; Xu et al., 2026c; Yuan et al., 2026; Fu et al., 2026; Fan et al., 2023; Zhan et al., 2024; Bao et al., 2023).

Vision-language-action (VLA) models provide a strong foundation for learning such behaviors from robot demonstrations (Zitkovich et al., 2023; Black et al., 2024a; Physical Intelligence et al., 2025; Wen et al., 2025; Kim et al., 2024; Cheang et al., 2024; NVIDIA et al., 2025; Octo Model Team et al., 2024; Liu et al., 2025). Their standard training objective, however, mainly asks the policy to reproduce the demonstrated action from the current observation or history. The trajectory contains an additional source of supervision: after each action, it records the visual scene and physical object state that the action actually produced. For dexterous manipulation, these future states directly characterize whether the demonstrated interaction was moving the task in the desired direction. Can this future information be distilled into a policy that still acts only from causal observations at deployment?

Figure 1 gives an overview of DexForesight.

We distinguish two ways in which the future of a demonstration can supervise learning. A **predicted future**—a future latent inferred from causal history and demonstrated actions—can teach the policy representation what interaction state to anticipate. A **realized future**—the future observations and object states actually recorded in the demonstration—can teach the action generator how its current prediction should change.

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.

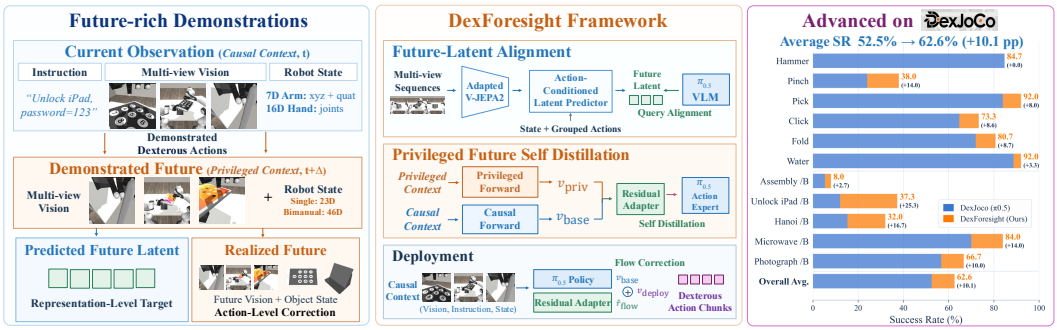

Figure 1: Overview of DexForesight. Future-latent alignment supervises the VLM representation; privileged flow correction is distilled into a causal residual adapter. All privileged inputs are removed at deployment. DexJoCo rand-obj: 52.5% → 62.6% (+10.1 pp).

Prior work gives strong evidence that future information improves robot learning. Predictive policies and world models learn from predicted futures to capture action-conditioned dynamics (Black et al., 2024b; Wu et al., 2024; Tian et al., 2025; Hu et al., 2025; Assran et al., 2025; Zheng et al., 2025; Xu et al., 2026b; Ma et al., 2026; Xu et al., 2026a), while privileged-learning methods use realized futures as training-only context (Fang et al., 2026; Kim et al., 2026). These roles are typically studied separately rather than jointly exploiting the same demonstrated future within a causal VLA. In this work, we show that the two roles can be combined within a causal VLA and provide complementary supervision.

Our method, dubbed as DexForesight, instantiates both roles inside a single flow-based VLA: the predicted future supervises the representation, and the realized future supervises the action correction. First, a V-JEPA 2 (Assran et al., 2025) encoder is adapted to dexterous trajectories and paired with an action-conditioned latent predictor using structured arm–hand actions; its predicted future latent supervises learnable query tokens in the vision-language model (VLM) representation. Second, the same $\pi_{0.5}$ (Physical Intelligence et al., 2025) policy is evaluated with and without future simulator object state and future visual features under an identical noisy action and flow time; the resulting velocity difference defines a future-conditioned flow correction distilled into a residual adapter. The two paths act on complementary parts of the model: visual-language representation and action generation.

We evaluate DexForesight on DexJoCo (Wang et al., 2026), which provides 11 functionally grounded single-arm and bimanual manipulation tasks under a common $\pi_{0.5}$ protocol. Our contributions are threefold. First, we formulate future trajectory supervision as a unified training principle for dexterous VLA: the same demonstration future that records the consequences of the demonstrated action provides additional trajectory-derived supervision at both the action and representation levels. Second, we show that both roles can be instantiated without any future or privileged observation at deployment: a paired-forward protocol isolates a future-conditioned flow correction for distillation, and a dexterity-adapted latent predictor supplies representation-level targets through masked query tokens. Third, we show on DexJoCo that future supervision improves closed-loop success across both single-arm and bimanual embodiments: controlled ablations show that representation-level alignment and action-level flow distillation each improve the base policy independently (55.6% and 57.4% vs. 52.5%), while their combination reaches 62.6%, and combining physical and visual privileged future context achieves the strongest average performance.

2 Related Work

2.1 Dexterous Vision-Language-Action Learning

Generalist robot policies have increasingly been extended from gripper manipulation to dexterous hands and high-dimensional arm–hand control. DexVLA (Wen et al., 2025) scales a diffusion action expert across heterogeneous embodiments, XL-VLA (Jiang et al., 2026) learns cross-hand latent actions for transfer across dexterous hands. A complementary line models interaction dynamics more

explicitly: Video Prediction Policy (Hu et al., 2025) learns from predictive visual representations, Contact-Grounded Policy (Xu et al., 2026c) predicts future robot and tactile states, and recent dexterous world models (Yuan et al., 2026; Fu et al., 2026) study action-conditioned dynamics for high-DoF control. On the policy architecture side, diffusion- and flow-based action generation (Chi et al., 2023; Lipman et al., 2023) has become a standard interface for continuous, high-dimensional control, and chunked/transformer-based policies (Zhao et al., 2023) further improve temporal coherence. Large-scale robot datasets and benchmarks (O'Neill et al., 2024) have enabled training generalist policies across multiple embodiments. Yet most dexterous VLAs are trained primarily to predict demonstrated action chunks from current observations, without directly supervising the future visual and physical consequences of those actions. This gap between what demonstrations record and what the objective optimizes is addressed by DexForesight.

2.2 Future Prediction for Robot Policies

Future prediction has been incorporated into robot learning at several levels. SuSIE (Black et al., 2024b) generates visual subgoals for goal-conditioned control, GR-1 (Wu et al., 2024) jointly predicts future images and actions, and Seer (Tian et al., 2025) uses forecast visual states to support inverse dynamics. More recent approaches move prediction into representation space: V-JEPA 2 action-conditioned (AC) (Assran et al., 2025) learns action-conditioned transitions in a pretrained video latent space; FLARE and FutureVLA (Zheng et al., 2025; Xu et al., 2026b) align policy features with future-aware embeddings; InternVLA-A1.5 and JEPA Policy (Ma et al., 2026; Xu et al., 2026a) couple action learning with latent foresight. In contrast, DexForesight uses a predictive latent model as training supervision for the VLM representation, with the action interface adapted to dexterous control—instantiating the predicted-future role.

2.3 Privileged Supervision and Distillation

Learning with privileged information uses signals that are available during training but unnecessary at deployment. PFD (Fang et al., 2026) compares a shared policy with and without true future observations and distills the resulting action-denoising difference into a causal residual. Pri4R (Kim et al., 2026) uses privileged 4D scene information to supervise future geometry for VLA learning. PFD and Pri4R exemplify this realized-future role, while related dexterous work also uses other privileged signals for deployable policy learning (Chen et al., 2026). In contrast, DexForesight follows the paired privileged/causal principle for action-level flow distillation and connects it with the predicted-future role of the same demonstration inside one causal VLA.

3 Method

A demonstration trajectory records the action to imitate as well as the future physical and visual states that the action produces. DexForesight converts this future trajectory information into two training signals for a causal $\pi_{0.5}$ policy. First, a V-JEPA 2 encoder adapted to dexterous trajectories, paired with an action-conditioned latent predictor, provides a representation-level future target for learnable query tokens in the VLM prefix (Sec. 3.2–3.4). Second, a privileged forward pass with future simulator object state and future visual features encoded by the adapted encoder defines an action-level correction in the flow velocity field, which is distilled into a lightweight residual adapter (Sec. 3.3). Figure 2 shows the training architecture.

3.1 Problem Formulation

At control step t, the policy receives a multimodal observation $\mathbf{o}_t = (\mathcal{I}_t, \ell, \mathbf{s}_t^{\mathrm{rob}})$ consisting of synchronized RGB observations $\mathcal{I}_t$, a language instruction ℓ, and deployment-available robot state $\mathbf{s}_t^{\mathrm{rob}}$. It predicts an action chunk $\mathbf{A}_t = [\mathbf{a}_t, \ldots, \mathbf{a}_{t+H-1}]$. We build on OpenPI $\pi_{0.5}$ (Physical Intelligence et al., 2025), which generates continuous action chunks through flow matching (Lipman et al., 2023): for flow time $\tau \in [0, 1]$, the noisy action is $\mathbf{x}_\tau = (1-\tau)\mathbf{A}_t + \tau\boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$, and the causal policy predicts the velocity $\mathbf{v}_\theta(\mathbf{x}_\tau, \tau \mid \mathbf{o}_t)$.

During training, the same demonstration also provides future visual observations and, in simulation, task-relevant object state $\mathbf{s}_{t+\Delta}^{\mathrm{obj}}$ at a later control time $t + \Delta$. We distinguish three future quantities

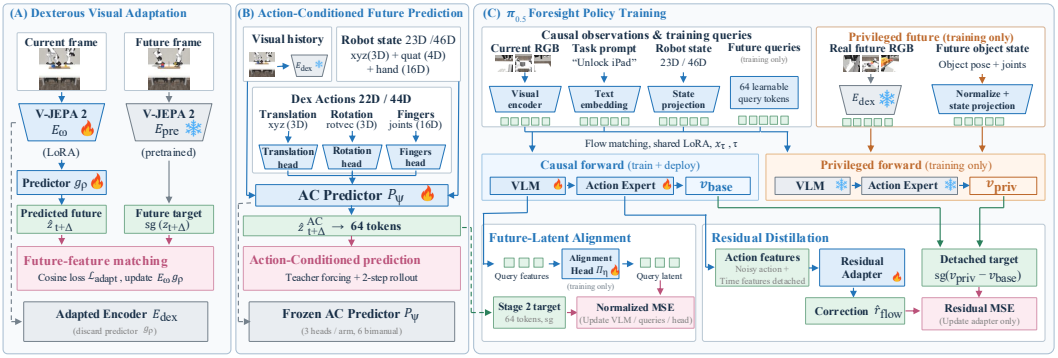

Figure 2: DexForesight training architecture. Left: a V-JEPA 2 encoder is adapted to dexterous trajectories, and an action-conditioned predictor produces a 64-token future-latent target. Right: future-latent alignment supervises training-only query tokens; paired causal and privileged forwards define the flow correction distilled into the residual adapter. All privileged inputs and future-query tokens are removed at deployment.

used during training: the recorded future visual observation $\mathcal{I}_{t+\Delta}$ (the synchronized multi-view frames), the simulator-provided future object state $\mathbf{s}^{\text{obj}}_{t+\Delta}$, and the predicted future latent $\widehat{\mathbf{z}}^{\text{AC}}_{t+\Delta}$. The first two are privileged observations of the realized demonstration future, whereas the third is a prediction conditioned on causal history and demonstrated actions alone. These signals are never part of the deployment observation. The future object state and future visual observation are encoded as privileged physical and visual context tokens, respectively, and together form the privileged future context used by the action-level path. The velocity difference induced by this context defines a future-conditioned flow correction, which is distilled into a residual adapter. We refer to this paired causal/privileged training mechanism as privileged future flow distillation. In DexJoCo, the privileged object-state labels are read directly from the simulator; analogous labels for real-world training data could be obtained offline from pose and articulation tracking while remaining unnecessary at deployment. The deployed policy therefore requires no additional sensing beyond the standard $\pi_{0.5}$ observation interface: no future frames, no object-state estimators, and no extra cameras.

3.2 Dexterous Action-Conditioned Future Prediction

The first training path models how the interaction should evolve under the demonstrated arm–hand action sequence. A pretrained video encoder such as V-JEPA 2 captures general visual dynamics, but dexterous manipulation involves fine-grained contact patterns and hand-object configurations that differ from the natural videos used in pretraining. Generic video latents also cannot be conditioned on high-dimensional arm–hand actions without an adapted action interface. We therefore adapt the encoder to dexterous trajectories and train an action-conditioned predictor in two stages, then use the resulting latent space both as the alignment target for the policy representation and as the visual component of the privileged context.

Stage 1: visual adaptation. We start from a pretrained V-JEPA 2 encoder E_{pre} (Assran et al., 2025) and adapt it to dexterous trajectories via LoRA (Hu et al., 2022), obtaining E_ω. For clarity, we describe the operation for one camera view and omit the view superscript; the same adaptation is applied independently to each view, as detailed in Appendix A. The adaptation task is future-latent prediction: given the current frame, a lightweight predictor g_ρ maps its adapted representation toward the frozen pretrained representation of the future frame at $t + \Delta$ (sg denotes stop-gradient):

$$\mathcal{L}_{\text{adapt}} = 1 - \cos(g_\rho(E_\omega(I_t)), \text{sg}(E_{\text{pre}}(I_{t+\Delta}))) \,. \tag{1}$$

The adapted encoder and g_ρ are optimized for this objective. After adaptation, g_ρ is discarded and the resulting encoder is frozen as E_{dex}.

Stage 2: action-conditioned latent prediction. Conditioned on the frozen E_{dex}, the predictor receives a history of visual latents together with robot state and demonstrated actions and predicts

future latents in the same E_{dex} space. Dexterous actions contain heterogeneous control components: arm transport, wrist rotation, and finger articulation operate at different spatial scales and affect different aspects of the interaction. To preserve this structure in the conditioning interface, we encode arm translation, arm rotation, and finger articulation as separate conditioning groups before combining them into the action condition; the same grouping is applied independently to the two sides for bimanual tasks. Training combines teacher-forced prediction with a short autoregressive rollout:

$$\mathcal{L}_{\text{AC}} = \mathcal{L}_{\text{teacher-forcing}} + \mathcal{L}_{\text{rollout}}. \tag{2}$$

The final predictor output, denoted $\widehat{\mathbf{z}}^{\text{AC}}_{t+\Delta}$, is the future representation used by the policy-alignment objective below. Because the predictor operates on the frozen E_{dex} space, it provides a target that is already adapted to dexterous visual statistics, rather than a generic video representation. During policy training, these two components serve distinct roles. The frozen E_{dex} encodes the recorded future observation to construct the visual privileged context for privileged future flow distillation (Sec. 3.3), whereas $\widehat{\mathbf{z}}^{\text{AC}}_{t+\Delta}$ provides the predictive target for future-latent alignment (Sec. 3.4).

3.3 Privileged Future Flow Distillation

Privileged future context. The second training path measures how the velocity field of the policy would change if the consequences of the current action were observable before committing to it. We therefore evaluate the same flow policy under two information conditions that share all policy parameters, the noisy action $\mathbf{x}_\tau$, and the flow time τ. The base prediction receives only the deployment-available observation. The privileged prediction additionally receives a physical and visual summary of the future trajectory at $t + \Delta$. The physical context is the task-relevant simulator object state, including object pose and articulated joint state. The visual context is obtained by encoding the recorded future observation with the dexterity-adapted visual encoder introduced in Sec. 3.2. These two feature sets are represented as additional context tokens, together forming the privileged context $\mathbf{c}^{\text{priv}}_t$. The camera views are identical to those of the standard policy input, so privileged training changes what the policy sees in time, not how it sees it. The physical and visual components are complementary: object state captures configuration and articulation outcomes that are difficult to infer from images, while visual features preserve appearance-level information that object state does not contain.

Both evaluations use the same policy parameters, noisy action, and flow time:

$$\mathbf{v}_{\text{base}} = \mathbf{v}_\theta(\mathbf{x}_\tau, \tau \mid \mathbf{o}_t), \qquad \mathbf{v}_{\text{priv}} = \mathbf{v}_\theta(\mathbf{x}_\tau, \tau \mid \mathbf{o}_t, \mathbf{c}^{\text{priv}}_t). \tag{3}$$

Their difference isolates the change in the velocity field induced by the observed future:

$$\mathbf{r}_{\text{flow}} = \text{sg}(\mathbf{v}_{\text{priv}} - \mathbf{v}_{\text{base}}). \tag{4}$$

Because $\mathbf{x}_\tau$, τ, and the policy weights are shared, the target is defined in the same local denoising problem as the causal prediction. This paired-forward design isolates the local change in the velocity prediction induced by adding future context, without confounding differences in policy weights, noise realization, or flow time.

Residual adapter. A lightweight residual adapter f^{res}_ϕ learns to recover $\mathbf{r}_{\text{flow}}$ from causal policy features, so that the correction remains available at deployment without any privileged input. The adapter receives the base-path action representation $\mathbf{h}_{\text{act}}$, noisy action $\mathbf{x}_\tau$, and flow-time embedding $\mathbf{e}_\tau$, and predicts $\widehat{\mathbf{r}}_{\text{flow}}$. Its output projection is zero-initialized, so training starts from the exact base-policy behavior and gradually learns the future-conditioned flow correction. We optimize

$$\mathcal{L}_{\text{res}} = \left\| f^{\text{res}}_\phi(\text{sg}(\mathbf{h}_{\text{act}}), \mathbf{x}_\tau, \text{sg}(\mathbf{e}_\tau)) - \mathbf{r}_{\text{flow}} \right\|^2_2. \tag{5}$$

The stop-gradient operations restrict this loss to the adapter. During sampling, the predicted correction is added to the base velocity at every flow-integration step.

3.4 Future-Latent Alignment

The action-conditioned predictor supplies a future target for the VLM representation. To interface this target with the policy, we append a set of N_q learnable future-query tokens to the VLM prefix. The

future-query tokens serve only as training-time representation slots. They do not provide additional observations and do not enter the Action Expert input. Future-query tokens may attend to the causal VLM context to read out a future-predictive representation, while causal observation tokens and the Action Expert are prevented from attending to the query positions. The queries therefore act as readout slots for future-predictive supervision and can be removed at deployment without changing the causal token inputs consumed by the action pathway. This design allows the policy to learn future-aware representations during training while maintaining a causal interface at deployment. After processing by the shared VLM, the query hidden features $\mathbf{h}_Q$ are mapped into the V-JEPA latent space by an alignment head Π_η. The action-conditioned prediction is mapped to the same token layout, producing $\widetilde{\mathbf{z}}^{\mathrm{AC}}_{t+\Delta}$. Because different latent channels can have different dynamic ranges, we normalize each token vector before computing the regression loss, so that the objective measures directional agreement rather than absolute magnitude:

$$\mathcal{L}_{\mathrm{align}} = \frac{1}{N_q}\sum_{i=1}^{N_q} \left\| \nu(\Pi_\eta(\mathbf{h}_{Q,i})) - \mathrm{sg}\big(\nu(\widetilde{\mathbf{z}}^{\mathrm{AC}}_{t+\Delta,i})\big) \right\|_2^2, \qquad \nu(\mathbf{z}) = \frac{\mathbf{z}}{\|\mathbf{z}\|_2}. \tag{6}$$

This masking prevents the Action Expert from directly consuming the query representations. Consequently, $\mathcal{L}_{\mathrm{align}}$ updates the future-query pathway and shared VLM representation, but does not directly supervise the Action Expert parameters. The alignment objective thus provides a second form of foresight: the residual adapter learns what the future implies for immediate action correction, while the query tokens learn what the future implies for the visual-language representation that conditions all future actions. This design gives the policy backbone a degree of temporal sensitivity without requiring any future or privileged observation at deployment.

3.5 Training Objective and Inference

The visual adaptation and action-conditioned predictor are trained before VLA optimization and remain frozen during policy training. The final policy objective separates into an action-level component and a representation-level component:

$$\mathcal{L}_{\mathrm{total}} = \underbrace{\mathcal{L}^{\mathrm{base}}_{\mathrm{FM}} + \lambda_{\mathrm{res}}\mathcal{L}_{\mathrm{res}} + \lambda_{\mathrm{priv}}\mathcal{L}^{\mathrm{priv}}_{\mathrm{FM}}}_{\text{action level}} + \underbrace{\lambda_{\mathrm{align}}\mathcal{L}_{\mathrm{align}}}_{\text{representation level}}, \tag{7}$$

where $\mathcal{L}^{\mathrm{base}}_{\mathrm{FM}}$ is the standard causal flow-matching loss and $\mathcal{L}^{\mathrm{priv}}_{\mathrm{FM}}$ is a lightly weighted flow-matching regularizer for the privileged prediction, preventing the privileged forward pass from drifting away from a valid velocity field. The three foresight-related losses have distinct gradient routes: $\mathcal{L}_{\mathrm{res}}$ updates only the residual adapter; $\mathcal{L}_{\mathrm{align}}$ updates the future-query embeddings, alignment head, and shared VLM representation; and $\mathcal{L}^{\mathrm{priv}}_{\mathrm{FM}}$ updates the shared policy through the privileged branch with a small weight. The weights λ_{res}, λ_{align}, and λ_{priv} are set so that future supervision contributes meaningful gradients without overwhelming the base action objective; exact values are reported in Appendix A. At deployment, the privileged context, future observations, predictive visual modules, and future-query tokens are removed. The policy acts from the standard causal observation using the trained $\pi_{0.5}$ backbone and lightweight residual adapter, requiring no privileged sensing or predictive model at deployment.

4 Experiments

Our experiments are designed to examine whether future trajectory supervision improves closed-loop dexterous VLA performance, how the representation-level and action-level foresight paths each contribute to the final policy, and how the two privileged future modalities shape the learned correction.

4.1 Experimental Setup

Benchmark and evaluation. We evaluate on DexJoCo (Wang et al., 2026). The benchmark contains 11 functionally grounded dexterous tasks: six single-arm tasks (Water Plant, Hammer Nail, Click Mouse, Pick Bucket, Pinch Tongs, and Fold Glasses) and five bimanual tasks (Microwave /B, Unlock iPad /B, Hanoi /B, Assembly /B, and Photograph /B). We use the benchmark's `rand-obj`

Table 1: Closed-loop success rate (%) on the 11 DexJoCo tasks. Mean ± std over 3 seeds × 50 episodes. Single-arm / Bimanual / Overall Avg. over 6, 5, 11 tasks. **Bold** marks the best per task-setting.

Setting	Method	Single-arm						Single-arm Avg.
		Hammer	Pinch	Pick	Click	Fold	Water	
rand-obj	DexJoCo ($\pi_{0.5}$)	84.7±5.0	24.0±6.9	84.0±7.2	64.7±8.1	72.0±3.5	88.7±3.1	69.7±9.8
rand-obj	DexForesight (Ours)	**84.7**±4.2	**38.0**±5.3	**92.0**±5.0	**73.3**±1.2	**80.7**±10.3	**92.0**±2.0	**76.8**±8.3
rand-full	DexJoCo ($\pi_{0.5}$)	17.3±5.0	18.7±1.2	**78.7**±6.1	54.7±7.0	39.3±3.1	75.3±6.4	47.3±11.0
rand-full	DexForesight (Ours)	**28.0**±4.0	**23.3**±6.1	72.7±4.2	**68.7**±7.6	**46.0**±6.0	**80.0**±5.3	**53.1**±9.9
multi-task	DexJoCo ($\pi_{0.5}$)	**86.7**±3.1	**45.3**±6.1	83.3±8.1	**80.7**±3.1	42.0±6.0	84.0±4.0	70.3±8.5
multi-task	DexForesight (Ours)	80.7±1.2	43.3±9.0	**89.3**±5.0	58.0±4.0	**64.7**±3.1	**97.3**±1.2	**72.2**±8.3

Setting	Method	Bimanual					Bimanual Avg.	Overall Avg.
		Assembly /B	Unlock iPad /B	Hanoi /B	Microwave /B	Photograph /B		
rand-obj	DexJoCo ($\pi_{0.5}$)	5.3±1.2	12.0±3.5	15.3±3.1	70.0±3.5	56.7±5.0	31.9±13.1	52.5±1.5
rand-obj	DexForesight (Ours)	**8.0**±2.0	**37.3**±3.1	**32.0**±4.0	**84.0**±5.3	**66.7**±5.0	**45.6**±13.4	**62.6**±1.5
rand-full	DexJoCo ($\pi_{0.5}$)	0.0±0.0	0.0±0.0	**15.3**±2.3	54.7±6.1	**21.3**±2.3	18.3±10.0	34.1±1.3
rand-full	DexForesight (Ours)	**2.0**±0.0	**21.3**±5.0	**11.3**±5.8	**56.0**±5.3	**24.0**±5.3	**22.9**±9.1	**39.4**±1.6
multi-task	DexJoCo ($\pi_{0.5}$)	3.3±2.3	0.7±1.2	6.0±0.0	39.3±13.0	29.3±1.2	15.7±7.8	45.5±1.7
multi-task	DexForesight (Ours)	**6.0**±3.5	**16.0**±7.2	**23.3**±5.8	**56.0**±4.0	**46.0**±4.0	**29.5**±9.3	**52.8**±1.5

setting as the primary evaluation protocol, which randomizes object placement and table height. Table 1 also reports results under the more challenging `rand-full` setting and the `multi-task` setting, following the same per-task protocol. We reproduce the DexJoCo $\pi_{0.5}$ baseline for the primary `rand-obj` setting; the `rand-full` and `multi-task` baseline results are taken from the original DexJoCo evaluation. For DexForesight and the official rand-obj baseline, we evaluate each method for 50 episodes under each of three random seeds (0, 1, and 2), and report the mean success rate with standard deviation across seeds. Single-arm tasks use a third-person camera and a wrist-mounted camera; bimanual tasks use an egocentric camera with two wrist-mounted cameras.

Policy training. We instantiate DexForesight on OpenPI $\pi_{0.5}$ and follow the official DexJoCo fine-tuning recipe. All variants use the same initialization and training budget; exact training durations, optimization, and loss-weight details are reported in Appendix A.

Predictive visual training. Both future-supervision paths use a horizon of approximately 0.5 s. Stage 1 adapts V-JEPA 2 with Eq. (1); the resulting encoder is then frozen while Stage 2 learns the action-conditioned predictor in the adapted latent space. Its final future prediction provides the target for Eq. (6). Complete optimization, sampling, and implementation details are given in Appendix A.

4.2 Main Results

Table 1 compares DexForesight with the DexJoCo $\pi_{0.5}$ baseline across the 11 DexJoCo tasks. On the primary rand-obj setting, DexForesight improves the average success rate from 52.5% to 62.6%, an absolute gain of 10.1 percentage points, improving 10 of the 11 tasks and matching the baseline on Hammer Nail. The largest gains concentrate on tasks requiring precise finger-mediated interaction or coordinated bimanual motion (Pinch Tongs +14.0, Unlock iPad /B +25.3, Hanoi /B +16.7, Microwave /B +14.0). The improvement holds under the harder rand-full setting (34.1% to 39.4%) and under multi-task training (45.5% to 52.8%), where the gains concentrate on bimanual tasks (Hanoi /B +17.3, Photograph /B +16.7) while several single-arm tasks degrade under joint training. Sec. 4.3 analyzes how these gains decompose across the two supervision paths and privileged modalities.

4.3 Component Analysis

We use two controlled ablations to separate the two roles of future supervision. Table 2 compares the base policy against Alignment Only (only future-latent alignment enabled), Flow Path Only (the action-level path enabled with combined physical–visual privileged future context and alignment disabled), and the full model that enables both—identical to the DexForesight configuration reported in Table 1. Table 3 then looks inside the action-level path, varying the privileged future-context modality while keeping alignment disabled.

Table 2: Ablation of the two foresight supervision paths under rand-obj. Mean ± std over 3 seeds × 50 episodes. **Bold** marks results exceeding the baseline.

Task	DexJoCo ($\pi_{0.5}$)	Alignment Only	Flow Path Only	Full DexForesight
Hammer Nail	84.7±5.0	76.0±3.5	82.0±3.5	84.7±4.2
Pinch Tongs	24.0±6.9	**29.3**±9.0	**35.3**±3.1	**38.0**±5.3
Pick Bucket	84.0±7.2	**85.3**±1.2	**87.3**±4.2	**92.0**±5.0
Click Mouse	64.7±8.1	**74.7**±12.7	**73.3**±3.1	**73.3**±1.2
Fold Glasses	72.0±3.5	**79.3**±4.6	**78.0**±11.1	**80.7**±10.3
Water Plant	88.7±3.1	86.0±8.0	**91.3**±4.2	**92.0**±2.0
Assembly /B	5.3±1.2	**7.3**±3.1	4.7±1.2	**8.0**±2.0
Unlock iPad /B	12.0±3.5	**31.3**±11.0	**29.3**±2.3	**37.3**±3.1
Hanoi /B	15.3±3.1	**24.7**±10.3	**22.0**±4.0	**32.0**±4.0
Microwave /B	70.0±3.5	56.7±4.2	**80.7**±6.4	**84.0**±5.3
Photograph /B	56.7±5.0	**60.7**±1.2	48.0±2.0	**66.7**±5.0
Average	52.5	55.6	57.4	**62.6**
Δ vs. Base	–	+3.1	+5.0	**+10.1**

Table 3: Ablation of privileged future context under rand-obj (alignment disabled). Physical: simulator object state; Visual: E_{dex} features of future observations. Mean ± std over 3 seeds × 50 episodes. **Bold** marks the best per task.

Task	DexJoCo ($\pi_{0.5}$)	Physical Future	Visual Future	Physical + Visual
Hammer Nail	**84.7**±5.0	80.0±5.3	70.0±6.9	82.0±3.5
Pinch Tongs	24.0±6.9	31.7±10.4	22.5±6.4	**35.3**±3.1
Pick Bucket	84.0±7.2	78.7±7.0	**87.3**±3.1	**87.3**±4.2
Click Mouse	64.7±8.1	66.0±4.0	69.3±3.1	**73.3**±3.1
Fold Glasses	72.0±3.5	72.7±8.1	75.3±11.5	**78.0**±11.1
Water Plant	88.7±3.1	90.0±2.0	90.0±5.7	**91.3**±4.2
Assembly /B	**5.3**±1.2	4.7±1.2	**5.3**±1.2	4.7±1.2
Unlock iPad /B	12.0±3.5	23.3±3.1	28.0±2.0	**29.3**±2.3
Hanoi /B	15.3±3.1	20.7±4.2	18.7±4.2	**22.0**±4.0
Microwave /B	70.0±3.5	72.0±3.5	64.0±8.0	**80.7**±6.4
Photograph /B	**56.7**±5.0	27.3±9.2	52.0±9.2	48.0±2.0
Average	52.5	51.6	52.9	**57.4**
Δ vs. Base	–	−0.9	+0.5	**+5.0**

Complementary foresight paths. Both paths independently improve over the base policy on average (Alignment Only 55.6%, Flow Path Only 57.4%, vs. 52.5%), and the full model reaches 62.6%, exceeding both isolated paths on 10 of 11 tasks; on Click Mouse, it matches Flow Path Only but remains below Alignment Only. The two paths are therefore complementary at the aggregate level, while their task-level contributions differ. The largest alignment gains are observed on Unlock iPad /B (+19.3), Click Mouse (+10.0), and Hanoi /B (+9.4). The largest flow-distillation gains are observed on Unlock iPad /B (+17.3), Pinch Tongs (+11.3 over base), and Microwave /B (+10.7). Isolated path–task combinations also degrade: Alignment Only reduces performance on Hammer Nail (−8.7) and Microwave /B (−13.3), indicating that representation-level foresight is not uniformly beneficial across interaction regimes, and Flow Path Only reduces performance on Photograph /B (−8.7), where Visual Future substantially outperforms Physical Future (52.0% vs. 27.3%), suggesting that object-state context alone is insufficient for this task. Enabling both paths recovers or exceeds the better of the two on nearly all tasks, indicating that the two paths provide non-redundant supervision.

Complementary privileged future modalities. Inside the flow-distillation path, neither single privileged modality provides a consistent aggregate improvement (Physical Future 51.6%, Visual Future 52.9%, vs. the 52.5% base), whereas combining them raises the average to 57.4% and outperforms both single-modality variants on 8 of 11 tasks (matching the Visual Future variant on Pick Bucket). The preferred privileged modality is task-dependent: visual future features are more effective on tasks such as Pick Bucket, whereas physical future state is more useful on tasks such as Pinch Tongs and Hanoi /B. Photograph /B degrades under the Physical Future variant—whose 32-D state carries little appearance information—but is partially recovered once visual future context is added. The two modalities thus supply complementary privileged signals for constructing the future-conditioned flow correction.

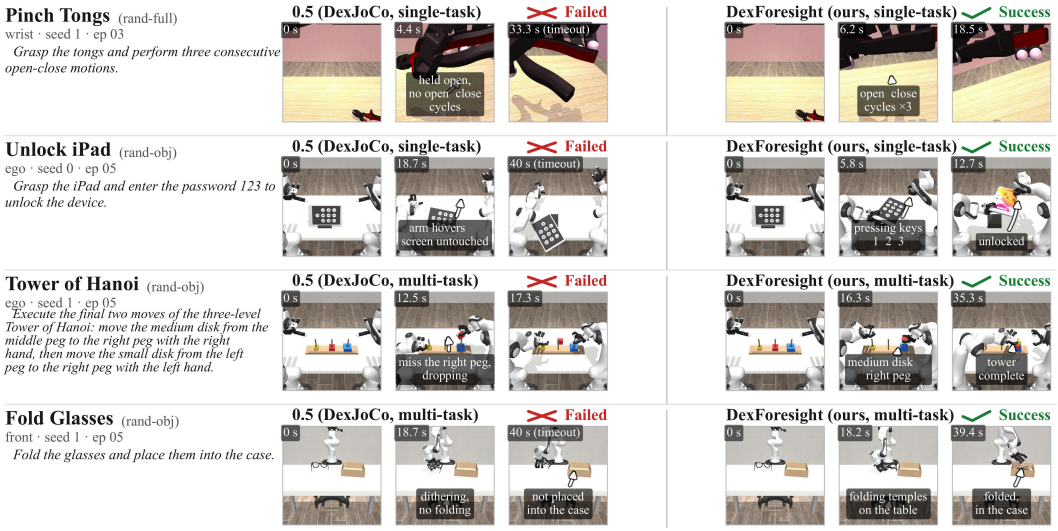

Figure 3: Matched-initialization rollouts in which the official baseline fails while DexForesight succeeds. Frame annotations highlight arm–hand trajectory differences before the final task outcome. Examples span finger-mediated interaction, articulated manipulation, and bimanual coordination.

4.4 Qualitative Analysis

Figure 3 compares matched closed-loop rollouts of the official base policy and DexForesight across representative interaction types. DexForesight reaches a more favorable arm–hand configuration before the final task event, followed by successful downstream execution.

5 Conclusion

DexForesight converts the future trajectory already recorded in dexterous demonstrations into two complementary supervision signals for a causal VLA policy: a representation-level target for the interaction state to anticipate, and an action-level correction distilled from causal–privileged velocity differences. On the 11-task DexJoCo benchmark, it improves the official DexJoCo $\pi_{0.5}$ baseline by 10.1 percentage points on average (62.6% vs. 52.5%), with higher average success across rand-obj, rand-full, and multi-task settings. Ablations show that the two paths are complementary, and that combining physical and visual privileged future context achieves the strongest average performance. Our evaluation is confined to simulation; real-world transfer would require offline object-state estimation.

Reproducibility Statement

Training configuration, gradient routing, deployment details, software, hardware, and evaluation seeds are reported in Sec. 4.1 and Appendix A. Per-seed results and code will be released with the checkpoints.

AI Use Statement

Generative AI tools (DeepSeek V4 Flash) assisted with analysis code; all technical claims and the manuscript are reviewed by the authors.

Ethics Statement

Our experiments are conducted in simulation without human subjects or personal data; transfer to physical robots should follow standard hardware-safety practices.

REFERENCES

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

Chen Bao, Helin Xu, Yuzhe Qin, and Xiaolong Wang. DexArt: Benchmarking generalizable dexterous manipulation with articulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21190–21200, 2023.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, Ury Zhilinsky, et al. π_0: A vision-language-action flow model for general robot control, 2024a.

Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Rich Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pre-trained image-editing diffusion models. In *International Conference on Learning Representations*, 2024b.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. GR-2: A generative video-language-action model with web-scale knowledge for robot manipulation, 2024.

Rosy Chen, Mustafa Mukadam, Michael Kaess, Tingfan Wu, Francois R. Hogan, Jitendra Malik, and Akash Sharma. PTLD: Sim-to-real privileged tactile latent distillation for dexterous manipulation, 2026.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023.

Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J. Black, and Otmar Hilliges. ARCTIC: A dataset for dexterous bimanual hand-object manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12943–12954, 2023. doi: 10.1109/CVPR52729.2023.01244.

Pengcheng Fang, Hongli Chen, and Xiaohao Cai. Privileged foresight distillation: Zero-cost future correction for world action models, 2026.

Yankai Fu, Ning Chen, Junkai Zhao, Heng Zhang, Guocai Yao, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. DeCAL: Towards physically-grounded dexterous vision-language-action models via contact-aware latent co-imagination, 2026.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

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 *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 24328–24346. PMLR, 2025.

Guangqi Jiang, Yutong Liang, Jianglong Ye, Jia-Yang Huang, Changwei Jing, Rocky Duan, Pieter Abbeel, Xiaolong Wang, and Xueyan Zou. Cross-hand latent representation for vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13496–13507, 2026.

Jisoo Kim, Jungbin Cho, Sanghyeok Chu, Ananya Bal, Jinhyung Kim, Gunhee Lee, Sihaeng Lee, Seung Hwan Kim, Bohyung Han, Hyunmin Lee, Laszlo A. Jeni, and Seungryong Kim. Pri4R: Learning world dynamics for vision-language-action models with privileged 4d representation, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model, 2024.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: A diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations*, 2025.

Haoxiang Ma, Junhao Cai, Xiaoxu Xu, Hao Li, Yuyin Yang, Yang Tian, Jiafei Cao, Hongrui Zhu, Zherui Qiu, Zhaxizhuoma, Yuqiang Yang, Jiaqi Peng, Xueyuan Wei, Yangkun Zhu, Jiahao Jiang, Xing Gao, Hanqing Wang, Feng Yuan, Kailin Li, Xueyue Zhu, Tai Wang, Yan Ding, Jiangmiao Pang, Jia Zeng, Jingjing Zhang, Bowen Zhou, Yao Mu, Chunhua Shen, and Weinan Zhang. InternVLA-A1.5: Unifying understanding, latent foresight, and action for compositional generalization, 2026.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. GR00T N1: An open foundation model for generalist humanoid robots, 2025.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy, 2024.

Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, et al. Open X-Embodiment: Robotic learning datasets and RT-X models. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, Ury Zhilinsky, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025.

Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. In *International Conference on Learning Representations*, 2025.

Hanwen Wang, Weizhi Zhao, Xiangyu Wang, Siyuan Huang, He Lin, Boyuan Zheng, Rongtao Xu, Gang Wang, Yao Mu, He Wang, Lue Fan, Hongsheng Li, Zhaoxiang Zhang, and Tieniu Tan. DexJoCo: A benchmark and toolkit for task-oriented dexterous manipulation on MuJoCo, 2026. URL `https://arxiv.org/abs/2605.16257`.

Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. DexVLA: Vision-language model with plug-in diffusion expert for general robot control, 2025.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations*, 2024.

Jie Xu, Kangjin Yu, Ziyi Jin, Junjie Gao, Liqing Chen, Yixian Li, Shuai Tian, and Zhongpu Xia. JEPA policy: Diffusion-free imitation learning via paired action and future representation prediction, 2026a.

Xiaoxu Xu, Hao Li, Jinhui Ye, Yilun Chen, Jia Zeng, Xinyi Chen, Linning Xu, Dahua Lin, Weixin Li, and Jiangmiao Pang. FutureVLA: Joint visuomotor prediction for vision-language-action model, 2026b.

Zhengtong Xu, Yeping Wang, Ben Abbatematteo, Jom Preechayasomboon, Sonny Chan, Nicholas Colonnese, and Amirhossein H. Memar. Contact-grounded policy: Dexterous visuotactile policy with generative contact grounding, 2026c.

Zizhao Yuan, Zhengtu Liang, Taowen Wang, Qiwei Liang, Yichi Wang, Yunheng Wang, Yuetong Fang, Lusong Li, Zecui Zeng, and Renjing Xu. Not all actions are equal: Rethinking conditioning for dexterous world model, 2026.

Xinyu Zhan, Lixin Yang, Yifei Zhao, Kangrui Mao, Hanlin Xu, Zenan Lin, Kailin Li, and Cewu Lu. OAKINK2: A dataset of bimanual hands-object manipulation in complex task completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 445–456, 2024. doi: 10.1109/CVPR52733.2024.00050.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems (RSS)*, 2023.

Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, Avnish Narayan, You Liang Tan, Guanzhi Wang, Qi Wang, Jiannan Xiang, Yinzhen Xu, Seonghyeon Ye, Jan Kautz, Furong Huang, Yuke Zhu, and Linxi Fan. FLARE: Robot learning with implicit world modeling. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 3952–3971, 2025.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of The 2023 Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 2023.

A Implementation and Reproducibility Details

A.1 Compute and Software

Policy experiments are trained on NVIDIA A100-SXM4-80GB GPUs. The final policy configuration uses four GPUs per task with FSDP; Stage-1 visual adaptation and Stage-2 action-conditioned prediction each use a single A100 GPU.

Table A.1: Software environment used for policy training and DexJoCo evaluation.

Environment	Package	Version
Policy	Python	3.11.15
Policy	JAX / jaxlib	0.5.3
Policy	Flax (NNX)	0.10.2
Policy	PyTorch	2.7.1+cu126
Policy	Optax	0.2.4
Policy	LeRobot	0.4.0
Policy	NumPy	2.4.6
Policy	einops	0.8.1
Policy	video decoding	PyAV
Evaluation	Python	3.11.15
Evaluation	MuJoCo	3.4.0 (EGL rendering)
Evaluation	openpi-client	0.1.0 (WebSocket policy client)
Evaluation	websockets	17.1
Evaluation	NumPy	1.26.4

Measured wall-clock times: a 30K-step single-arm run takes approximately 9 hours on four A100 GPUs (approximately 26 hours on a single GPU); a 30K-step bimanual run takes approximately 10 hours on four GPUs; the final 60K-step bimanual run takes approximately 22.3 hours on four A100 GPUs (multi-view input). Stage-1 visual adaptation takes approximately 2 hours on a single GPU, and Stage-2 action-conditioned fine-tuning approximately 1.3 hours per camera view on a single GPU.

A.2 Policy Optimization

We use the official DexJoCo OpenPI $\pi_{0.5}$ initialization and LoRA (Hu et al., 2022) variants, with a Gemma-2B VLM LoRA and a Gemma-300M Action-Expert LoRA; only the LoRA parameters and the newly introduced modules (residual adapter, future-query tokens, alignment head, and privileged-context projections) are trainable, and all remaining weights are loaded from the released base checkpoint. Our recipe matches the official DexJoCo fine-tuning configuration: batch size 32, identical LoRA structure, 30K updates for single-arm tasks and 60K updates for bimanual tasks, and the same public 100-episode demonstration set per task. The optimizer is AdamW with gradient clipping at 1.0 and a 1K-step linear warmup to 5×10^{-5}, followed by a constant learning rate. No parameter Exponential Moving Average (EMA) is used. The action-chunk horizon is $H = 30$ at 30 fps, and checkpoints are saved every 10K updates.

For the full method, $\lambda_{\text{res}} = 1.0$, $\lambda_{\text{align}} = 0.1$, and $\lambda_{\text{priv}} = 0.1$. The alignment term is linearly warmed up over the first 1K policy updates. During policy training, the Action Expert and causal observation tokens are masked from attending to the $N_q = 64$ future-query positions. The privileged physical context is projected to a small fixed set of 4 physical tokens; the privileged visual context is pooled to 64 visual tokens before projection.

A.3 DexJoCo Interfaces and Privileged State

The privileged physical context is the task-relevant simulator object state—object pose and articulated joint state—read at a future offset of $\Delta = 15$ simulator frames, corresponding to 0.5 s at 30 fps; trajectories shorter than the offset are clamped at the final frame. The privileged state contains no velocity, contact-force, or object-velocity signals. Each privileged vector uses a uniform 32-D

Table A.2: Canonical DexJoCo policy interfaces. Task-specific camera aliases follow the released benchmark configuration; Click Mouse uses an `ego_right` third-person view in place of `front`.

Setting	RGB views	Robot state	Policy action	Horizon
Single-arm	2	23-D	22-D	30
Bimanual	3	46-D	44-D	30

format; task-specific state variables are inserted into fixed semantic slots and unused dimensions are zero-padded. The vector is normalized with a z-score whose per-dimension variance is floored by a data-driven bound, $\max(\mathrm{std},\ \mathrm{range}/4,\ 10^{-2})$, so that static dimensions are not amplified. The normalized vector is passed through LayerNorm and a linear projection to 4 VLM tokens with learned positional embeddings, which are appended to the privileged forward pass only. Table A.3 lists the per-task privileged object-state content. These labels are read directly from the simulator and are used only during training; analogous labels for real-world training data could be obtained offline from pose and articulation tracking while remaining unnecessary at deployment. For each view v, the recorded future visual latent from E_{dex} is pooled to the compact 8×8 layout. The V view-specific latents are concatenated and projected into the VLM token space as 64 visual tokens, which are combined with the physical context.

Table A.3: Per-task privileged object-state content. All tasks use the uniform 32-D container projected to 4 physical tokens.

Task	Privileged object state
Hammer Nail	hammer (tool) pose and nail state
Pinch Tongs	tongs pose
Water Plant	spray-bottle pose, plant pose, table height
Click Mouse	mouse pose, table height
Fold Glasses	glasses pose, case pose, table height
Pick Bucket	bucket pose, boxed-object pose, table height
Assembly /B	socket pose, peg pose
Unlock iPad /B	tablet pose
Hanoi /B	tower-base pose
Microwave /B	microwave pose and door articulation state
Photograph /B	camera pose

A.4 Stage 1: V-JEPA 2 Visual Adaptation

Stage 1 starts from the official V-JEPA 2 ViT-G encoder (`vjepa2_1_vitg_384`, 384×384 input resolution) (Assran et al., 2025) and adapts it to dexterous trajectories with LoRA. LoRA rank and alpha are both 16 and are applied to the qkv, projection, and feed-forward layers, giving 19.0M trainable parameters. Training uses Eq. (1): the adapted representation of the current frame is mapped by a lightweight predictor toward the frozen pretrained representation of the frame at $t + \Delta$ with a cosine loss. Optimization uses AdamW with learning rate 10^{-4}, zero weight decay, batch size 12, and two epochs. Frames are sampled with stride 3 and the future target uses $\Delta = 15$ frames. The split uses the first 80% of episodes for adaptation and the remaining 20% for validation. After adaptation, the view-specific predictors are discarded and the adapted encoders are frozen as $\{E^v_{\mathrm{dex}}\}_{v=1}^{V}$. Each view-specific encoder is shared across all tasks that use the corresponding camera view.

A.5 Stage 2: Action-Conditioned Future-Latent Prediction

Stage 2 trains a shared V-JEPA 2-AC predictor (Assran et al., 2025) (original full version, fine-tuned) on top of the frozen $\{E^v_{\mathrm{dex}}\}_{v=1}^{V}$ for 20 epochs. This predictor learns to anticipate how the demonstrated actions will transform the current visual state into future configurations. The predictor uses 24 layers, hidden width 1024, 16 attention heads, frame-causal attention, and rotary positional embeddings. Visual context contains eight frames sampled from 30 fps trajectories with stride 8, yielding an effective rate of approximately 3.75 fps. Training uses AdamW with learning rate 10^{-4}, cosine schedule, weight decay 0.04, batch size 8, and equal weighting of the teacher-forcing and

two-step autoregressive rollout losses. The final 10% of episodes form the validation split. The action condition is grouped into translation, rotation, and finger-articulation components for single-arm tasks and the corresponding six groups for bimanual tasks; per-step actions are the mean of the absolute actions over the corresponding stride window, normalized per dimension. Absolute actions are used for conditioning. For each camera view, the predictor produces a view-specific future latent. The V predicted latents are pooled to 8×8, concatenated across views, and projected/resampled to $N_q = 64$ target tokens for Eq. (6). The final predicted future latent $\widehat{\mathbf{z}}^{\mathrm{AC}}_{t+\Delta}$ is detached when used as the target.

A.6 Gradient Routing and Deployment

For the path ablation in Table 2, the four configurations differ only in the foresight objectives used during training. The Base policy sets $\lambda_{\mathrm{res}} = \lambda_{\mathrm{align}} = \lambda_{\mathrm{priv}} = 0$. Alignment Only uses $\lambda_{\mathrm{align}} = 0.1$ and $\lambda_{\mathrm{res}} = \lambda_{\mathrm{priv}} = 0$. Flow Path Only uses $\lambda_{\mathrm{res}} = 1.0$, $\lambda_{\mathrm{priv}} = 0.1$, and $\lambda_{\mathrm{align}} = 0$. Full DexForesight uses $\lambda_{\mathrm{res}} = 1.0$, $\lambda_{\mathrm{align}} = 0.1$, and $\lambda_{\mathrm{priv}} = 0.1$. For the privileged-context ablation in Table 3, we fix $\lambda_{\mathrm{align}} = 0$, $\lambda_{\mathrm{res}} = 1.0$, and $\lambda_{\mathrm{priv}} = 0.1$, and vary only the physical and visual components of the privileged future context. The residual adapter receives the base-path action representation, noisy action, and flow-time embedding; its output projection is zero-initialized, so training starts from the exact base-policy behavior. The privileged flow residual $\mathbf{r}_{\mathrm{flow}}$ is detached, and the action representation and flow-time embedding supplied to the residual adapter are also stop-gradient; consequently, $\mathcal{L}_{\mathrm{res}}$ updates only the adapter. The privileged flow-matching regularizer ($\lambda_{\mathrm{priv}} = 0.1$) keeps the privileged forward pass a valid velocity field without letting privileged information dominate the shared weights. The alignment objective updates the future-query embeddings, the alignment head, and the VLM parameters that produce the query representations; the action-conditioned target remains detached. The Stage-1 and Stage-2 visual modules are trained before policy optimization and never receive policy gradients.

At deployment, simulator object state, recorded-future visual context, E_{dex}, P_ψ, the privileged forward pass, and the future-query tokens are all removed. The policy retains the standard $\pi_{0.5}$ network and the residual adapter. Action generation uses $N_{\mathrm{Euler}} = 10$ Euler integration steps, with the residual adapter applied once per flow step; the measured inference-time overhead of the adapter is below 1%.

A.7 Evaluation Details

The primary evaluation uses DexJoCo `rand-obj`, which randomizes object placement and table height. The `rand-full` rows additionally randomize the full scene configuration, and the `multi-task` rows evaluate joint-training checkpoints per task; both follow the same per-task protocol as `rand-obj`. All multi-task and rand-full baselines are cited directly from the DexJoCo paper. For DexForesight configurations and the official rand-obj baseline, we evaluate 50 episodes for each of seeds 0, 1, and 2 and report mean $\pm$ standard deviation across seeds. The rand-full and multi-task baseline statistics are taken directly from the DexJoCo paper. Evaluation follows the official asynchronous OpenPI server–client protocol and its default replanning configuration: the next action chunk is generated while the current chunk executes, and flow integration uses 10 Euler steps.

B Qualitative Rollouts and Case Analysis

B.1 Matched-Initialization Rollout Comparison

Figure 3 presents four matched-initialization comparisons spanning finger-mediated interaction, articulated manipulation, and bimanual coordination. Across these examples, the main difference typically emerges before the final task event: DexForesight reaches a more favorable arm–hand configuration at the bottleneck interaction, after which the remaining action sequence succeeds.

B.2 Representative Failure Patterns

We annotate every evaluation episode of the main configurations as success or failure and organize the failure cases by their dominant mode, using frame-level review for the seven tasks with recorded videos (Fold Glasses, Click Mouse, Photograph /B, Unlock iPad /B, Hanoi /B, Assembly /B, Mi-

crowave /B); for the remaining four tasks (Water Plant, Pinch Tongs, Pick Bucket, Hammer Nail) we use simulator trajectory logs and per-episode outcomes to characterize coarse failure patterns. We identify six representative failure patterns covering the annotated cases:

1. **Stall (stationary attractor).** The largest mode, observed across five tasks: the arm enters a near-stationary state and remains until the horizon, typically *after* a successful grasp—the failure is not the grasp but the subsequent sub-action sequence that never unfolds.
2. **Last-millimeter button miss.** Visual localization is correct but contact-level precision is insufficient: the fingertip lands between keys (Unlock iPad /B), or the palm presses instead of the fingertip and the button is never actuated (Click Mouse).
3. **Undershoot.** The task-defining displacement is not completed: glasses are pressed flat on the table but not placed into the case, or placed with temples unfolded (Fold Glasses); the peg hovers above the socket without insertion (Assembly /B).
4. **Misalignment.** The tool or object pose is offset at the moment of interaction: the tool lands at an inclined angle beside the target with no effective stroke; a disk misses its peg.
5. **Grasp miss.** The object is never acquired or the task is never initiated.
6. **Overshoot with drop.** A single catastrophic case: the target object is struck off the table and leaves the scene—the only divergence-type failure among all reviewed cases, in contrast to the graceful partial completion elsewhere.