PRIVILEGED FORESIGHT DISTILLATION: AMORTIZED FUTURE CORRECTION FOR WORLD ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

World action models jointly predict future video and action during training, raising an open question about what role the future-prediction branch actually plays. A recent finding shows that this branch can be removed at inference with little to no loss on common manipulation benchmarks, suggesting that future information may act merely as a regularizer on the shared visual backbone. We propose instead that joint training induces an action-conditioned correction that privileged future observations impose on action denoising, and that current-only policies capture this correction only partially. Making the account precise, we formulate privileged foresight as a residual in the action-denoising direction—the difference between what a model predicts given the true future and what it predicts given only the current frame—and introduce *Privileged Foresight Distillation (PFD)*, which transfers this residual from a training-time teacher into a small adapter on a current-only student. The teacher and student share the same backbone and differ only in the attention mask over video tokens; future video is never generated at inference. Controlled experiments support that this gain reflects a future-conditioned correction rather than added capacity: with the total teacher weight held fixed, routing it directly rather than as a residual drops the policy below the undistilled baseline. Empirically, PFD improves over Fast-WAM on LIBERO and RoboTwin manipulation benchmarks while preserving the current-only inference interface with only a slight adapter-induced latency overhead. This view reframes the role of future information in world action models: not as a target to predict, nor as a branch to discard, but as a compressible correction to be distilled—one that the privileged path carries even when it is the worse predictor.

1 INTRODUCTION

Joint prediction of future video and action is a central design pattern in world action models, motivated by the intuition that visual foresight during training helps an agent choose better actions. A recent finding challenges this premise: a model trained jointly with video prediction can be deployed without test-time future generation while matching or exceeding the predictive variant (Yuan et al., 2026). The result has been read as evidence that test-time future imagination is unnecessary, but it leaves a deeper question unanswered. If the future branch is not used at inference, what role does future information play during training, and is any of its action-specific content lost when the branch is removed?

Two readings of this finding are possible. On a *regularizer* reading, future video shapes the shared visual backbone but contributes nothing action-specific; the current-only policy captures everything useful, and there is nothing to recover. On a *privileged-foresight* reading, future video induces a structured correction on the action-denoising direction itself—a correction that joint training transfers only partially to the current-only path. The two readings are observationally similar in the existing literature, yet they imply opposite methodological prescriptions: the first directs effort toward stronger visual backbones, the second toward better mechanisms for transferring the privileged signal. We find that the first reading is incomplete. Simply exposing the current-only policy to more training capacity—naïve finetuning of the same backbone layers—fails to improve performance (Section 4.3), so the gap between what joint training can teach and what the current-only policy learns is not a capacity gap. The interesting signal, if it exists, must lie in a direction that pure supervision on the action target does not reach.

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.

We locate this signal by asking what privileged access to the future would change in the action-denoising process. During training, we instantiate the same backbone as two parallel paths, identical except for the attention mask over video tokens: a current-only student that sees only the current frame (matching the standard joint-training setup), and a privileged teacher that attends to the full future video. The teacher's action-velocity prediction minus the student's defines a *foresight residual*—the component of the denoising direction that becomes predictable once future information is available. *Privileged Foresight Distillation (PFD)* trains a small adapter on the student path to predict this residual from current-only context. The residual target is detached before use, so the inherited joint-training objective is not pulled away from the action target by a moving teacher signal. At inference, the teacher is discarded and the adapter augments the student's prediction at each denoising step; the current-only inference interface is preserved exactly, with the foresight-induced correction restored through a residual head whose added latency is slight (Section 4).

We design controlled experiments to interpret PFD's gain, isolating it from confounds of capacity, regularization, and fine-tuning-budget reallocation. None of these alternatives accounts for the observed effect in our controls, supporting a specific reading of the transferred signal: privileged foresight is a future-conditioned correction that is not recovered by matched direct fine-tuning under the same budget, and a small adapter is sufficient to absorb it.

Contributions. We make the following contributions.

- **A new perspective on future information.** We propose that future information in world action models is best understood as an action-conditioned correction residual—a direction not recovered by matched direct fine-tuning under the same budget.
- **PFD.** We introduce a training-only teacher–student construction that makes this view operational: the teacher accesses real future during training, a small adapter distills the teacher-minus-student residual, and the adapter preserves the current-only inference interface with no future generation at test time.
- **Controlled evidence for the transferred signal.** We break the transfer one ingredient at a time and show that the residual routing itself, not only the added supervision, accounts for part of the gain.
- **Empirical results.** PFD improves over the Fast-WAM backbone on LIBERO and RoboTwin, matching or exceeding several methods that rely on embodied pretraining, while adding only a slight inference overhead from the adapter.

2 Related Work

World action models and future video. Recent robot policies combine video backbones with action heads, either by jointly predicting future frames and actions (Wu et al., 2024; Cheang et al., 2024; Hu et al., 2025) or by conditioning actions on externally generated future videos (Du et al., 2023; Black et al., 2024b). In both settings, future materialization—as pixels or latent rollouts—is required at inference and often dominates computation. Fast current-only policies remove this test-time future generation with a single forward pass (Yuan et al., 2026). We ask whether action-relevant future information can still benefit such current-only policies, and in what form.

Uses of future information. Prior work mainly uses future information in two ways. *Future-as-prediction* explicitly generates future frames for action conditioning (Du et al., 2023; Black et al., 2024b), while *future-as-representation* learns latent imagination rollouts for planning or representation learning (Hafner et al., 2023; Schrittwieser et al., 2020; Hansen et al., 2024). The former requires future generation at inference; the latter rolls the future out in latent space at planning time or during policy optimization. In contrast, PFD uses *future-as-correction*: future video is available only during training, where it reveals what a current-only policy misses, and is distilled into a residual correction that is not reconstructed at inference.

Privileged information and adapter heads. PFD builds on asymmetric teacher–student learning with privileged information (Vapnik & Vashist, 2009; Lopez-Paz et al., 2016; Chen et al., 2019; Robinet et al., 2020; Choudhury & Sodhi, 2025), commonly used to transfer supervision from more informed teachers to constrained students. Closest in spirit, Neitz et al. (2021) distill future trajectories into a present-only student via target activations from a separate, meta-learned teacher.

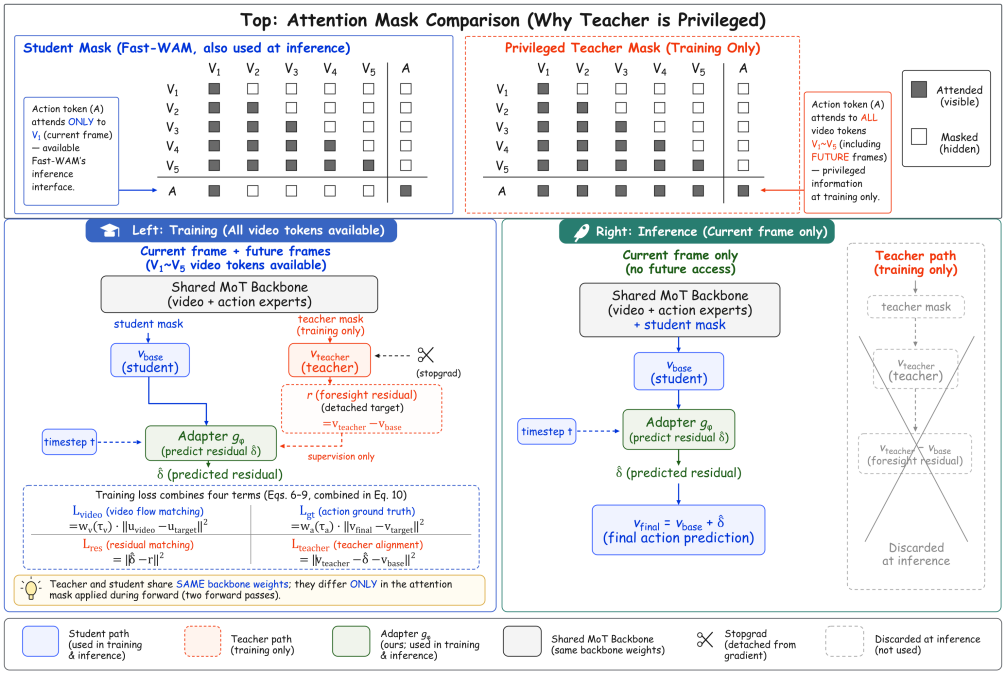


Figure 1: **PFD (Privileged Foresight Distillation). Top:** Student and privileged teacher paths differ only in their attention mask: the student action tokens attend to the current-frame video tokens and action tokens, matching the Fast-WAM current-only inference interface, whereas the teacher action tokens attend to all video tokens, including real future frames available only during training. **Left:** During training, the same MoT backbone is evaluated under both masks, yielding a live student prediction v_{base} and a stop-gradient privileged prediction v_{teacher}. The detached residual target $r = \text{sg}(v_{\text{teacher}} - v_{\text{base}})$ captures the action-denoising correction induced by future access. A small adapter g_ϕ takes the live v_{base} and predicts $\hat{\delta}$ to fit this residual, producing $v_{\text{final}} = v_{\text{base}} + \hat{\delta}$. **Right:** At inference, the teacher path and future video tokens are discarded. The model runs only the student mask and applies the adapter correction, preserving the current-only Fast-WAM interface with no test-time future generation and slight adapter-only compute cost.

Here, teacher and student share the same backbone parameters and differ only in their attention mask over video tokens, removing architectural confounds. Moreover, PFD defines the adapter target as the teacher–student residual rather than replacing the student with full teacher imitation; a weak teacher-consistency term is used only to stabilize the corrected output, isolating the component attributable to future access (Charikar et al., 2024; Menon et al., 2021; Silver et al., 2018). This residual is carried by a small action-stream adapter; unlike generic parameter-efficient adapters (Hu et al., 2022), it is explicitly sized and trained to encode the foresight residual.

3 METHOD

PFD is a training-time mechanism that operationalizes the future-as-correction view of §1: a privileged path with future access produces an action-side correction signal, and a small adapter on the current-only path absorbs it. Inference uses the current-only path and the adapter; the privileged path is not instantiated.

3.1 PRELIMINARIES

Let $X = (X_1, \ldots, X_T)$ denote a sequence of T video frames with X_1 the current frame, and let A denote the corresponding action chunk. Following Yuan et al. (2026), we adopt a Mixture-of-Transformers backbone with parameters θ, comprising a video expert and an action expert, with cross-stream information exchange controlled by a joint attention mask.

Both streams are trained with flow matching. We sample timesteps $\tau_v, \tau_a \in [0, 1]$ independently and draw independent Gaussian noise $\varepsilon_v, \varepsilon_a$, yielding the corrupted inputs

$$X_{\tau_v} \;=\; (1-\tau_v)\,\varepsilon_v + \tau_v\, X, \qquad A_{\tau_a} \;=\; (1-\tau_a)\,\varepsilon_a + \tau_a\, A, \tag{1}$$

with action-velocity target $v_{\text{target}} = A - \varepsilon_a$. We write u_{video} for the video-velocity output and $u_{\text{target}} = X - \varepsilon_v$ for its target. We write

$$v_{\text{act}}(X_{\tau_v},\, A_{\tau_a},\, \tau_v,\, \tau_a;\; M)$$

for the action-velocity output produced by the backbone when the joint self-attention is restricted by mask M. The current-only forward used at inference adopts the *student mask* M_{S}, under which each action-token query attends to the current-frame video tokens X_1 and to the other action tokens:

$$v_{\text{base}} \;=\; v_{\text{act}}(X_{\tau_v},\, A_{\tau_a},\, \tau_v,\, \tau_a;\; M_{\text{S}})\,. \tag{2}$$

Throughout the displayed equations we omit first-frame observation conditioning and the per-timestep scheduler weighting on the velocity outputs for clarity; both follow Yuan et al. (2026) and are applied identically to the student and teacher forwards introduced below.

3.2 A PRIVILEGED FORWARD VIA ATTENTION MASKING

PFD adds a second action forward, identical to equation 2 in every respect except for the attention mask:

$$v_{\text{teacher}} \;=\; \text{sg}[\, v_{\text{act}}(X_{\tau_v},\, A_{\tau_a},\, \tau_v,\, \tau_a;\; M_{\text{T}})\,]\,. \tag{3}$$

The *teacher mask* M_{T} allows each action-token query to attend to the full set of video tokens $X_1, \ldots, X_T$, including future frames; $\text{sg}(\cdot)$ denotes the stop-gradient operator. The teacher and student forwards share the same parameters θ at every step—there is no exponential moving average, no frozen copy, and no distinct teacher network—and they consume the same noisy inputs (X_{τ_v}, A_{τ_a}) generated from a single noise sample $(\varepsilon_v, \varepsilon_a)$. The two forwards differ only in the attention mask over video tokens, which isolates the effect of future access: capacity, parameterization, optimizer state, and noise realization are held identical, so any difference between v_{teacher} and v_{base} is attributable to the enlargement of the action queries' attention support. The stop-gradient on equation 3 further removes the teacher from the optimization graph, so it contributes no parameter update of its own and serves only as a target source for the residual we now define.

3.3 THE FORESIGHT RESIDUAL AND THE RESIDUAL ADAPTER

Rather than have the student imitate v_{teacher} in full, PFD distills only the component that future access changes. We define the *foresight residual* as

$$r \;:=\; \text{sg}(v_{\text{teacher}} - v_{\text{base}})\,; \tag{4}$$

since v_{teacher} is already detached, this is equivalent to $r = v_{\text{teacher}} - \text{sg}(v_{\text{base}})$. Targeting r rather than v_{teacher} confines supervision to the component that future access changes at the current θ; whenever the two masks induce identical predictions, the target is zero and no teacher signal enters the loss. This is what makes the construction viable even though the privileged forward is the *worse* predictor (Appendix B): the teacher is not a better policy to imitate but a probe of what future access moves.

The residual is absorbed by a small *residual adapter* g_φ placed at the output of the action expert and applied token-wise:

$$\hat{\delta} \;=\; g_\varphi(v_{\text{base}},\, \tau_a)\,, \qquad v_{\text{final}} \;=\; v_{\text{base}} + \hat{\delta}. \tag{5}$$

First, the adapter consumes v_{base} rather than $\text{sg}(v_{\text{base}})$; only the residual target r in equation 4 is detached. As we discuss in §3.4, this asymmetry is what allows residual supervision to influence the backbone subset θ' at all under partial fine-tuning, rather than reducing to a pure adapter-fitting problem on φ. Second, the adapter's output projection is zero-initialized, so $\hat{\delta} \equiv 0$ at the start of training and $v_{\text{final}} = v_{\text{base}}$ identically; the corrected forward equals the standard Fast-WAM student forward at initialization, and any departure from it develops only as training updates g_φ.

3.4 TRAINING OBJECTIVE AND GRADIENT ROUTING

Let $\theta' \subseteq \theta$ denote the subset of backbone parameters that are permitted to update; the adapter parameters φ always update. PFD trains (θ', φ) against the inherited video flow-matching loss together with three action-side losses:

$$\mathcal{L}_{\text{video}} = w_v(\tau_v) \left\| u_{\text{video}} - u_{\text{target}} \right\|^2, \tag{6}$$

$$\mathcal{L}_{\text{gt}} = w_a(\tau_a) \left\| v_{\text{final}} - v_{\text{target}} \right\|^2, \tag{7}$$

$$\mathcal{L}_{\text{res}} = \left\| \hat{\delta} - r \right\|^2, \tag{8}$$

$$\mathcal{L}_{\text{teacher}} = \left\| v_{\text{final}} - v_{\text{teacher}} \right\|^2, \tag{9}$$

where $w_v(\cdot)$ and $w_a(\cdot)$ are the per-timestep weighting schedules of Yuan et al. (2026). The full PFD objective is

$$\mathcal{L} \;=\; \lambda_{\text{video}}\,\mathcal{L}_{\text{video}} \;+\; \lambda_{\text{gt}}\,\mathcal{L}_{\text{gt}} \;+\; \lambda_{\text{res}}\,\mathcal{L}_{\text{res}} \;+\; \lambda_{\text{teacher}}\,\mathcal{L}_{\text{teacher}}, \tag{10}$$

with non-negative scalar coefficients; the values used in all experiments are reported in §4.1.

Gradient routing. The teacher forward is fully stop-gradiented at equation 3 and contributes no update to θ. The residual target r in equation 4 is also detached, which prevents $\mathcal{L}_{\text{res}}$ from being trivially reduced by moving its target instead of fitting it. However, because the adapter input in equation 5 is the live v_{base} rather than $\text{sg}(v_{\text{base}})$, $\mathcal{L}_{\text{res}}$ is not confined to updating φ: under partial fine-tuning, gradient also flows from $\mathcal{L}_{\text{res}}$ through the dependence of $\hat{\delta}$ on v_{base} and into θ'. Residual supervision therefore reshapes both the correction head and the backbone's emitted current-only velocity, and the detached target keeps that reshaping directed at the mismatch rather than at the target itself.

$\mathcal{L}_{\text{res}}$ versus $\mathcal{L}_{\text{teacher}}$. The two teacher-derived losses coincide in forward value but differ in gradient path. Substituting $v_{\text{final}} = v_{\text{base}} + \hat{\delta}$ into equation 9 gives $\|\hat{\delta} - (v_{\text{teacher}} - v_{\text{base}})\|^2$, which equals $\mathcal{L}_{\text{res}}$ as a number. They diverge once gradients are computed. Writing $e = \hat{\delta} - r = v_{\text{final}} - v_{\text{teacher}}$ and $J = \partial g_\varphi / \partial v_{\text{base}}$, the two terms produce identical adapter gradients, $2\,(\partial\hat{\delta}/\partial\varphi)^\top e$, and differ only in the gradient reaching the backbone: $\mathcal{L}_{\text{res}}$ contributes $2\,J^\top e$ through the adapter's dependence on v_{base}, while $\mathcal{L}_{\text{teacher}}$ contributes $2\,(I + J)^\top e$, adding a direct pull of the current-only velocity toward $v_{\text{teacher}} - \hat{\delta}$. PFD retains both terms: λ_{teacher} sets the strength of the direct pull, and $\lambda_{\text{res}} + \lambda_{\text{teacher}}$ that of the adapter-routed shaping.

PFD admits two regimes via the choice of θ': *adapter-only* ($\theta' = \varnothing$, the backbone is frozen) and *partial fine-tuning* (θ' unfreezes the last K_a blocks of the action expert and the last K_v blocks of the video expert); specific values and the default configuration are reported in §4.1.

3.5 INFERENCE

At inference, PFD preserves the current-only denoising interface of Fast-WAM. At each flow-matching denoising step, the model computes the student velocity v_{base} from equation 2 under the student mask M_{S}, applies the residual adapter, and uses

$$v_{\text{final}} \;=\; v_{\text{base}} + g_\varphi(v_{\text{base}},\, \tau_a)$$

for the sampling update on A_{τ_a}. The teacher mask M_{T} is never instantiated at inference, and the future video frames $X_2, \ldots, X_T$ are neither generated nor consumed. The only added cost relative to Fast-WAM is one forward pass through g_φ per denoising step.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Benchmarks. We evaluate on LIBERO (Liu et al., 2023) and RoboTwin 2.0 (Chen et al., 2025), following Fast-WAM (Yuan et al., 2026). LIBERO contains four suites (Spatial, Object, Goal, Long); we train a single model jointly on all four suites (2,000 demonstrations over 40 tasks) and report per-suite success rate over at least 500 trials. RoboTwin 2.0 is a bimanual dual-arm benchmark; we

use its multi-task setup with $2{,}500$ clean-scene and $25{,}000$ randomized-scene demonstrations across more than 50 tasks, reporting success over 100 trials per task in each condition.

Baselines. Our primary baseline is Fast-WAM in two forms: "Fast-WAM (released)" is the released model—published numbers on LIBERO, our evaluation of that checkpoint on RoboTwin—while "Fast-WAM (reproduced)" is re-trained with the released configuration, codebase, and schedule used by our PFD runs, and serves as the reference for reported gains. The reproduced numbers are slightly lower than the original report but follow consistent suite-level trends under a unified evaluation pipeline. For broader context, we also include published numbers for OpenVLA (Kim et al., 2024), π_0 (Black et al., 2024a), $\pi_{0.5}$ (Physical Intelligence et al., 2025), Motus (Bi et al., 2025), and LingBot-VA (Li et al., 2026), taken verbatim from Yuan et al. (2026). These five context baselines use embodied pretraining ("Emb. PT."), whereas Fast-WAM and PFD use the Wan2.2-5B backbone without embodied pretraining.

Training. We train for 30 epochs on LIBERO and 15 on RoboTwin using 8 H100 GPUs, matching Fast-WAM's batch size, schedule, and optimizer family. We use AdamW with cosine decay, weight decay 0.01, gradient clipping 1.0, and benchmark-specific learning rates following Fast-WAM defaults: $6{\times}10^{-5}$ for LIBERO and $1{\times}10^{-4}$ for RoboTwin. PFD adds only the privileged forward of §3.2, one additional no-grad attention pass per step; all runs are initialized from the released Fast-WAM checkpoint (training cost in Appendix G).

Inference. Following Fast-WAM, we use 10 flow-matching denoising steps with classifier-free guidance scale 1.0. At each step, PFD runs one student forward under M_{S} and applies the residual adapter, $v_{\mathrm{final}} = v_{\mathrm{base}} + g_\varphi(v_{\mathrm{base}}, \tau_a)$, as in §3.5. The teacher mask M_{T} is never instantiated at inference, and no future video frames are generated or consumed. End-to-end latency is reported in §4.4.

Implementation. All main results use the partial fine-tuning regime of §3.4, with trainable parameters $\theta' \cup \varphi$. Here, θ' contains the last K_a action-expert blocks and last K_v video-expert blocks, each expert having 30 blocks, and φ denotes the adapter. We set $(K_a, K_v) = (12, 12)$ for both benchmarks, unfreezing about 40% of blocks per expert. The adapter g_φ is a three-layer SiLU MLP of width 512; it takes a linear projection of the live base-action velocity v_{base} from equation 2 and a sinusoidal embedding of τ_a broadcast over tokens, with zero-initialized output projection. It contains only 1.2M trainable parameters, small relative to the Wan2.2-5B backbone. Loss weights are fixed for all PFD runs: $\lambda_{\mathrm{video}} = \lambda_{\mathrm{gt}} = 1.0$, $\lambda_{\mathrm{res}} = 0.5$, and $\lambda_{\mathrm{teacher}} = 0.1$.

4.2 Main results

LIBERO. Table 1 reports per-suite success rates. PFD raises the LIBERO average from 96.95 for the reproduced Fast-WAM to 98.10, a gain of $+1.15$ on the four-suite mean and $+0.50$ over the released model's 97.6; the $2{,}000$ evaluations are paired by initial state, and an exact McNemar test gives $p = 0.015$. PFD improves on three of the four suites, most on Goal; on Object, where both methods exceed 99%, the difference is within the binomial standard error of a 500-trial evaluation. Against methods that use embodied pretraining, PFD surpasses Motus, $\pi_{0.5}$ and π_0, and trails LingBot-VA by 0.40—without a separate embodied pretraining stage. Neither ingredient carries the gain alone. Adapter-only PFD ($\theta' = \varnothing$) reaches 96.60, below the reproduced baseline; disabling the correction instead, at the default θ', gives 97.15, which the paired test does not separate from the baseline (Appendix D). We adopt partial fine-tuning with the correction as the default.

RoboTwin 2.0. Table 2 reports clean-scene, randomized-scene, and average success rates. PFD reaches $94.23/93.59$ on clean and randomized respectively, with an average of 93.91, improving over the Fast-WAM row by $+1.23$ on clean, $+0.91$ on randomized, and $+1.07$ on the average. PFD's 93.91 also exceeds the strongest embodied-pretrain baseline (LingBot-VA at 92.2) by 0.7 despite using no embodied pretraining, and is the highest among all Wan2.2-based entries.

4.3 Isolating the foresight signal

We ask where the gain of §4.2 comes from. On LIBERO we run three controlled probes that share PFD's training budget and each break exactly one ingredient of the foresight transfer, and verify whether breaking it erases the gain.

Table 1: LIBERO success rate (%) over at least 500 trials per suite. "Emb. PT." indicates embodied pretraining; "(reproduced)" is re-trained under our codebase; ±: std over three evaluation runs. Best per column bold, second best underlined.

Method	Emb. PT.	Spatial	Object	Goal	Long	Average
OpenVLA (Kim et al., 2024)	✓	84.7	88.4	79.2	53.7	76.5
π_0 (Black et al., 2024a)	✓	96.8	98.8	95.8	85.2	94.1
$\pi_{0.5}$ (Physical Intelligence et al., 2025)	✓	**98.8**	98.2	<u>98.0</u>	92.4	96.9
Motus (Bi et al., 2025)	✓	96.8	<u>99.8</u>	96.6	<u>97.6</u>	97.7
LingBot-VA (Li et al., 2026)	✓	98.5	99.6	97.2	**98.5**	**98.5**
Fast-WAM (released) (Yuan et al., 2026)	–	98.2	**100.0**	97.0	95.2	97.6
Fast-WAM (reproduced)	–	97.0	99.4	96.6	94.8	$96.95^{\pm 0.08}$
PFD (partial fine-tune, ours)	–	<u>98.6</u>	99.2	**99.2**	95.4	<u>$98.10^{\pm 0.06}$</u>
PFD (adapter-only, $\theta' = \varnothing$)	–	97.2	98.8	96.6	93.8	96.6

Table 2: RoboTwin 2.0 success rate (%) over 100 trials per task. "Emb. PT." indicates embodied pretraining; "from Wan2.2" is re-trained on our backbone without embodied pretraining; "(released)": the released Fast-WAM checkpoint under our evaluation; ±: std over three evaluation runs. Best per column bold, second best underlined.

Method	Emb. PT.	Clean	Randomized	Average
π_0 (Black et al., 2024a)	✓	65.92	58.40	62.2
$\pi_{0.5}$ (Physical Intelligence et al., 2025)	✓	82.74	76.76	79.8
Motus (Bi et al., 2025)	✓	88.66	87.02	87.8
Motus from Wan2.2	–	77.56	77.00	77.3
LingBot-VA (Li et al., 2026)	✓	<u>92.90</u>	91.50	<u>92.2</u>
LingBot-VA from Wan2.2	–	80.60	–	80.6
Fast-WAM (released) (Yuan et al., 2026)	–	91.88	<u>91.78</u>	$91.83^{\pm 0.22}$
PFD (partial fine-tune, ours)	–	**94.23**	**93.59**	$\mathbf{93.91^{\pm 0.12}}$

Matched-capacity control. The first probe, *pure finetune*, unfreezes the same backbone subset $\theta' = (K_a, K_v) = (12, 12)$ and trains against the action ground truth alone, with no teacher forward and no adapter. If the PFD gain were attributable to the additional trainable capacity that θ' exposes, this control would match or exceed PFD. It does not: pure finetune scores $96.40/99.20/96.40/94.80$ for an average of 96.70, which is -0.25 below the reproduced Fast-WAM and -1.40 below PFD. Unfreezing the same subset of layers under direct action supervision slightly hurts the current-only policy at this training budget. The signal that PFD transfers is therefore not accessible to direct supervision on the action target, even when the layers permitted to update are identical.

Shuffled-future control. The second probe, *shuffled-future* PFD, replaces the teacher's future frames $X_2, \ldots, X_T$ at every training step with frames drawn from an unrelated trajectory in the same batch. The teacher mask M_{T}, the adapter, the loss weights, and the schedule are otherwise identical to the default PFD run. If the gain reflected auxiliary-loss regularization or the mere presence of a second supervisory target, destroying the temporal correspondence between X_1 and $X_{2:T}$ should leave it largely intact, since the teacher still receives real frames from the training distribution and the loss structure is unchanged. Instead, shuffled-future PFD scores $96.10/99.20/96.20/95.00$ for an average of 96.63, which is -0.32 below the reproduced Fast-WAM and -1.47 below PFD. The transferred signal therefore depends on genuine current-to-future correspondence, not on incidental properties of the teacher forward.

Depth–width trade-off probe. The third probe asks whether reducing video-side fine-tuning depth while increasing residual-head width can substitute for the default full-depth PFD configuration. We double the adapter hidden width from 512 to 1024 and, to test a practical depth–width trade-off, halve the video-expert fine-tune depth from $K_v = 12$ to $K_v = 6$ (the action expert is held at $K_a = 12$). This redirects the freed compute from updating the deeper video stack into a wider correction head. The resulting configuration scores $97.90/99.80/97.20/94.50$ for an average of 97.35, which is -0.75 below the default PFD at width 512. Redirecting fine-tuning budget from the video expert to a wider adapter therefore fails to recover the gain.

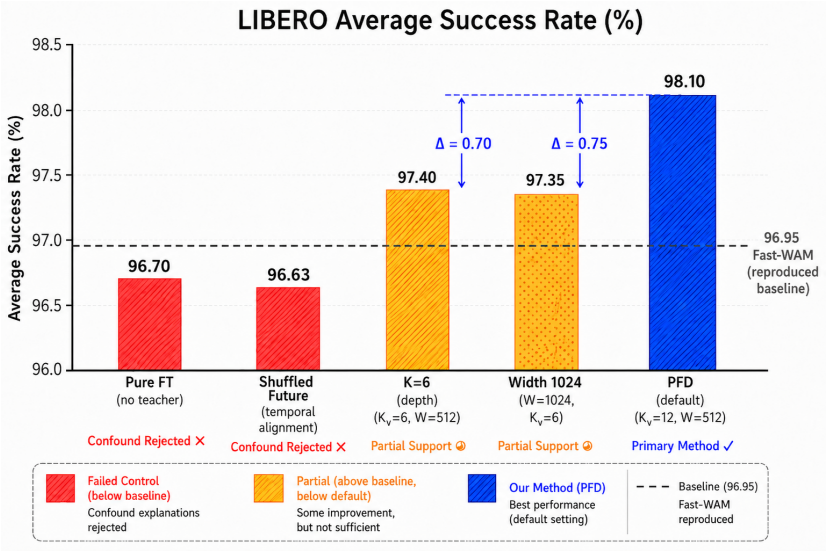


Figure 2: LIBERO average success rate for the three primary probes, Fast-WAM (reproduced), and PFD (default). Breaking matched-capacity (Pure FT) or temporal-correspondence (Shuffled Future) drops accuracy below the Fast-WAM baseline (red); the budget-reallocation control—redirecting video-side fine-tuning budget to a wider adapter (Width 1024 at $K_v = 6$)—closes only part of the PFD gap (amber). The supplementary fine-tune-depth check ($K = 6$) is shown for comparison and lands at a similar level.

Reading the probes together. Relative to the default PFD average of 98.10, the three probes land at deltas of -1.40 (matched-capacity), -1.47 (shuffled-future), and -0.75 (budget-reallocation). The account under test predicts the first two: if what transfers is the foresight residual r of equation 4, then keeping the capacity while removing the teacher, and keeping the teacher while destroying the correspondence that makes its prediction a *foresight* prediction, must both forfeit the gain. Both do, each landing at or below the frozen baseline. Budget reallocation is weaker evidence: it closes part of the gap, and the $(6, 6)$ row of Table 3 reaches 97.40 without any adapter widening, so we read the remaining gap to 98.10 as tracking fine-tune depth on the video expert rather than adapter width. None of the three confounds reproduces r, which is by construction the component of the action-velocity field that becomes available when the attention mask exposes future frames at the same parameters; the alternatives change capacity, target, or budget allocation while leaving that mask alone.

Routing control. The three probes each remove an ingredient of the transfer; a fourth keeps every ingredient and changes only how the teacher signal is routed. Holding architecture, schedule and the total teacher-derived weight fixed, we move that weight entirely onto the direct term, $(\lambda_{\text{res}}, \lambda_{\text{teacher}}) = (0, 0.6)$ instead of $(0.5, 0.1)$; by §3.4 the adapter receives identical gradients either way and only the update reaching the backbone differs. At the matched $(K_a, K_v) = (12, 6)$ setting this scores 96.20: below the reference run at 97.45, and below both the no-distillation control and the undistilled baseline. The same teacher, at the same total weight, helps as a residual and hurts as a direct target—the gain follows the routing, not the added supervision. Freezing the teacher at its initial weights instead destabilizes training altogether (Appendix E); what matters is how the teacher-derived gradient reaches a backbone that is itself still moving. Correction disabled at inference and further analyses are reported in Appendices B–G.

4.4 Inference Overhead

A central practical question for any future-aware policy is how much foresight costs at deployment time. Models that materialize future video at inference— either by jointly denoising future frames and actions, or by generating a future clip before an inverse-dynamics module (IDM)—pay a multiplicative latency penalty. We project their cost by applying the relative slowdowns reported by Fast-WAM to our measured cached-context baseline.

Table 3: LIBERO epistemic probes (success rate %). The first three probes break, in turn, capacity, temporal correspondence, and the allocation of the fine-tuning budget between backbone depth and adapter width. The last two rows are supplementary ablations on the adapter regime and on fine-tune depth.

Configuration	Spatial	Object	Goal	Long	Average
Fast-WAM (reproduced)	97.00	99.40	96.60	94.80	96.95
Pure finetune (θ', no teacher)	96.40	99.20	96.40	94.80	96.70
Shuffled-future PFD	96.10	99.20	96.20	95.00	96.63
PFD, width 1024 at $(K_a, K_v) = (12, 6)$	97.90	99.80	97.20	94.50	97.35
PFD (default) $(K_a, K_v) = (12, 12), W = 512$	**98.60**	99.20	**99.20**	**95.40**	**98.10**
adapter-only ($\theta' = \varnothing$)	97.20	98.80	96.60	93.80	96.60
fine-tune depth $(K_a, K_v) = (6, 6)$	97.90	99.70	97.30	94.70	97.40

Table 4: End-to-end inference latency per action chunk. In our LIBERO setting, one chunk contains 32 actions. Slowdown is reported relative to the current-only Fast-WAM cached-context baseline.

Method	Test-time future	Latency (ms/chunk)	Slowdown
Fast-WAM-Joint	joint video+action denoising	786.2 (proj.)	$3.05\times$
Fast-WAM-IDM	generate future, then IDM	1098.1 (proj.)	$4.26\times$
Fast-WAM	none (current-only)	257.7	$1.00\times$
PFD (ours)	none (current-only) + adapter	$\mathbf{271.0}$	$\mathbf{1.05}\times$

Table 4 reports end-to-end latency per action chunk on a single H100, measured over 20 trials after discarding 5 warmup runs, under the 10 flow-matching denoising steps used throughout this paper. In our LIBERO setup, each `infer_action` call predicts a chunk of 32 actions. The current-only Fast-WAM cached-context baseline runs at 257.77 ms per chunk, while PFD runs at 271.04 ms per chunk, adding 13.27 ms or 5.15% overhead. A prompt-mode implementation adds 7.80 ms (2.84%) over its own baseline.

These results show that the adapter cost remains small under both inference implementations. The added latency is attributable to one lightweight g_ϕ forward at each denoising step. The two future-materializing variants reach 98.5 and 98.0 on the LIBERO average at a projected $3.05\times$ and $4.26\times$ the current-only latency; PFD reaches 98.1 at $1.05\times$. PFD never instantiates the teacher mask and never generates future frames at inference (§3.5).

5 Conclusion

We revisited the role of future video in world action models once test-time imagination is removed, and argued that future is best understood not as a prediction target nor as a branch to discard, but as a compressible correction to be distilled. PFD operationalizes this view with a same-backbone teacher–student construction and a small output-side adapter that absorbs the foresight residual. Controlled probes attribute the gain to the foresight signal itself, and PFD improves over Fast-WAM on both LIBERO and RoboTwin with a slight adapter-only inference overhead while preserving the current-only inference interface exactly. More generally, our controls suggest that a privileged path can be worth distilling even when it is not the better predictor, provided its difference from the student is routed as a residual rather than imitated as a target.

Limitations. First, the construction is deliberately simple: a single output-side MLP adapter, a full-horizon teacher mask, and a single backbone family. Second, our claims are empirical: the alternatives we test do not account for the gain, but we give no formal characterization of when the residual admits a low-capacity approximation. Third, our evaluation is simulation-only.

AI USE STATEMENT

In this work, we used generative AI tools for writing assistance (language editing of author-written text) and for coding assistance (implementation and refactoring of experiment scripts). We have not used generative AI tools for research ideation, experimental design, or analysis of results. We have reviewed all AI-assisted work: all edited text was checked by the authors for fidelity to the intended claims, and all AI-assisted code was tested and its outputs verified against independently produced results.

ETHICS STATEMENT

This work uses public simulation benchmarks only and involves no human subjects or personal data. PFD improves the compute efficiency of robot policies without unlocking capabilities beyond the underlying backbone; the broader societal considerations are those common to manipulation-learning research.

REPRODUCIBILITY STATEMENT

All architectural components, attention masks, the residual definition, losses, and gradient routing are specified in Section 3; all training and inference hyperparameters are listed in Section 4.1. Benchmarks (LIBERO, RoboTwin 2.0) and the Wan2.2-5B backbone are public. Code, training/evaluation scripts, and checkpoints will be released upon publication.

REFERENCES

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model. *arXiv preprint arXiv:2512.13030*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024a.

Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer 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 (ICLR)*, 2024b. URL https://openreview.net/forum?id=c0chJTSbci.

Moses Charikar, Chirag Pabbaraju, and Kirankumar Shiragur. Quantifying the gain in weak-to-strong generalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, Hanbo Zhang, and Minzhao Zhu. GR-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024. URL https://arxiv.org/abs/2410.06158.

Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *Conference on Robot Learning (CoRL)*, 2019. URL https://arxiv.org/abs/1912.12294.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan-ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

Sanjiban Choudhury and Paloma Sodhi. Better than your teacher: LLM agents that learn from privileged AI feedback. In *International Conference on Learning Representations (ICLR)*, 2025.

Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2302.00111.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023. URL https://arxiv.org/abs/2301.04104.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.16828.

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. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *International Conference on Machine Learning (ICML)*, 2025. URL https://arxiv.org/abs/2412.14803. Spotlight.

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. *arXiv preprint arXiv:2406.09246*, 2024.

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

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023.

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. In *International Conference on Learning Representations (ICLR)*, 2016.

Aditya Krishna Menon, Ankit Singh Rawat, Sashank Reddi, Seungyeon Kim, and Sanjiv Kumar. A statistical perspective on distillation. In *International Conference on Machine Learning (ICML)*, 2021.

Alexander Neitz, Giambattista Parascandolo, and Bernhard Schölkopf. A teacher-student framework to distill future trajectories. In *International Conference on Learning Representations (ICLR)*, 2021.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

François Robinet, Antoine Demeules, Raphaël Frank, Georgios Varisteas, and Christian Hundt. Leveraging privileged information to limit distraction in end-to-end lane following. In *IEEE Consumer Communications & Networking Conference (CCNC)*, 2020. doi: 10.1109/CCNC46108.2020.9045110.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588 (7839):604–609, 2020. doi: 10.1038/s41586-020-03051-4.

Tom Silver, Kelsey Allen, Josh Tenenbaum, and Leslie Kaelbling. Residual policy learning. *arXiv preprint arXiv:1812.06298*, 2018.

Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5–6):544–557, 2009. doi: 10.1016/j.neunet.2009.06.042.

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 (ICLR)*, 2024. URL https://openreview.net/forum?id=NxoFmGgWC9.

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

A Training algorithm

For completeness, Algorithm 1 below summarizes one PFD training step in pseudocode form. The corresponding loss definitions and gradient-routing argument appear in §3.4 of the main paper.

Algorithm 1 One PFD training step.

Require: Batch (X, A); backbone f_θ; adapter g_φ; masks $M_{\mathrm{S}}, M_{\mathrm{T}}$; weights $\lambda_{\mathrm{video}}, \lambda_{\mathrm{gt}}, \lambda_{\mathrm{res}}, \lambda_{\mathrm{teacher}}$.

1: Sample $\tau_v, \tau_a \in [0,1]$ independently; sample Gaussian noise $\varepsilon_v, \varepsilon_a$.
2: Form noised inputs X_τ, A_τ.
3: $v_{\mathrm{base}} \leftarrow f_\theta(X_\tau, A_\tau, \tau_a;\, M_{\mathrm{S}})$ ▷ student forward
4: $v_{\mathrm{teacher}} \leftarrow \mathrm{sg}\big(f_\theta(X_\tau, A_\tau, \tau_a;\, M_{\mathrm{T}})\big)$ ▷ teacher forward
5: $r \leftarrow \mathrm{sg}(v_{\mathrm{teacher}} - v_{\mathrm{base}})$ ▷ foresight residual
6: $\hat{\delta} \leftarrow g_\varphi\big(v_{\mathrm{base}},\, \tau_a\big)$
7: $v_{\mathrm{final}} \leftarrow v_{\mathrm{base}} + \hat{\delta}$
8: $\mathcal{L} \leftarrow \lambda_{\mathrm{video}}\,\mathcal{L}_{\mathrm{video}} + \lambda_{\mathrm{gt}}\, w_a(\tau_a) \|v_{\mathrm{final}} - v_{\mathrm{target}}\|_2^2 + \lambda_{\mathrm{res}}\, \|\hat{\delta} - r\|_2^2 + \lambda_{\mathrm{teacher}}\, \|v_{\mathrm{final}} - v_{\mathrm{teacher}}\|_2^2$
9: Update (θ', φ) via an optimizer step on $\nabla\mathcal{L}$.

B Privileged-teacher offline analysis

Masked per-sample velocity MSE against v_{target} on 512 held-out LIBERO states with fresh noise draws, and open-loop action prediction (full 10-step denoising of a whole action chunk, L_1 against ground-truth actions, same initial noise per variant; the teacher variant receives the clean true future).

Table 5: Offline velocity analysis on two PFD checkpoints.

Quantity	**PFD** $(12, 6)$	**PFD** $(12, 12)$
student v_{base}	0.0099	0.0126
corrected v_{final}	0.0067 (-32%)	0.0106 (-16%)
teacher, noised future	0.0371	0.0427
teacher, clean true future	0.0529	0.0595
$\cos(\hat{\delta},$ true error direction$)$	$+0.64$	$+0.61$
$\cos(r,$ true error direction$)$	-0.2	-0.2

Why the raw residual can anti-align while the learned correction aligns. Fix the backbone and consider the population objective for the adapter over its input $x = (v_{\text{base}}, \tau_a)$. For the residual term alone, every g satisfies $\mathbb{E}\|g(x) - r\|^2 = \mathbb{E}\|g(x) - \mathbb{E}[r \mid x]\|^2 + \mathbb{E}\|r - \mathbb{E}[r \mid x]\|^2$—the cross term vanishes by the tower property—and the second term does not depend on g, so the minimizer over any adapter class is the L_2-projection of the conditional mean $\mathbb{E}[r \mid x]$ onto that class. Including the ground-truth term equation 7 (and noting $\mathcal{L}_{\text{teacher}} = \mathcal{L}_{\text{res}}$ as a function of $\hat{\delta}$ at fixed v_{base}), the same argument gives the unconstrained minimizer as the convex combination

$$\hat{\delta}^\star(x) \;=\; \frac{\lambda_{\text{gt}}\, w_a(\tau_a)\, \mathbb{E}[\, v_{\text{target}} - v_{\text{base}} \mid x\,] \;+\; \big(\lambda_{\text{res}} + \lambda_{\text{teacher}}\big)\, \mathbb{E}[\, r \mid x\,]}{\lambda_{\text{gt}}\, w_a(\tau_a) \;+\; \lambda_{\text{res}} + \lambda_{\text{teacher}}},$$

a weighted average of two *conditional means*. The adapter therefore does not track r alone: its optimum mixes the residual with the ground-truth correction $v_{\text{target}} - v_{\text{base}}$, and only the component of either target that is predictable from x is realizable. The learned correction can therefore align with the true error ($+0.61$ in Table 5) while the raw residual does not (-0.2); we do not claim that the residual term alone supplies that alignment.

Table 6: Open-loop action L_1 on 512 held-out states.

Variant	**Action L_1**
Fast-WAM baseline (student mask)	0.132
PFD, student path alone (v_{base})	0.118
PFD, student + adapter (v_{final})	0.072
teacher mask, clean true future	0.400

C CONTROLS AT $(K_a, K_v) = (12, 6)$

The *teacher-output routing* row isolates the residual formulation itself. It keeps the architecture, the schedule and the total teacher-derived weight of the reference run fixed and moves that weight entirely onto the direct term, $(\lambda_{\text{res}}, \lambda_{\text{teacher}}) = (0, 0.6)$ instead of $(0.5, 0.1)$; by §3.4 the two settings give the adapter identical gradients and differ only in how much of the update reaches the backbone. Routing the same supervision this way costs 1.25 points relative to the reference and also falls below both the no-distillation control and the Fast-WAM baseline, so the residual routing is not an incidental parameterization of the same signal.

Table 7: Per-suite success rate (%) for the $(12, 6)$ controls, at least 500 trials per suite.

Configuration	**Spatial**	**Object**	**Goal**	**Long**	**Average**
PFD $(12, 6)$, $W = 512$ (reference)	98.00	99.80	97.40	94.60	97.45
correction disabled at inference	97.20	99.80	96.60	94.80	97.10
no distillation ($\lambda_{\text{res}} = \lambda_{\text{teacher}} = 0$)	98.00	99.40	96.20	93.20	96.70
teacher-output routing ($\lambda_{\text{res}} = 0$, $\lambda_{\text{teacher}} = 0.6$)	98.00	98.40	96.00	92.40	96.20
Action expert only, $(12, 0)$	95.80	98.90	95.30	93.60	95.90
Action expert only, $(6, 0)$	96.60	98.90	95.60	94.20	96.33

D WHERE THE GAIN LIVES

The correction head and the fine-tuned backbone can be switched off independently, which separates their contributions at a fixed checkpoint. Table 8 does this at the default configuration; all rows use the same 500 trials per suite and the same initial states, so the comparisons are paired and we report exact McNemar tests on the 2,000 paired trials.

Three readings follow. First, the deployed correction carries most of the gain: disabling it costs 0.95 of the 1.15 points, and what remains of the fine-tuned backbone is $+0.20$ over the baseline, which the paired test does not separate from zero ($p = 0.75$). Second, the correction is not transferable on its own: with the backbone frozen the same adapter reaches 96.60, below the baseline. Third, the effect is specific to which parameters are free to move—replacing partial fine-tuning by rank-16

Table 8: Switching off each ingredient at the default $(K_a, K_v) = (12, 12)$ configuration. Paired McNemar against PFD unless noted.

Configuration	LIBERO average	paired p
PFD (default)	98.10	—
correction disabled at inference	97.15	0.029
adapter only ($\theta' = \varnothing$)	96.60	—
rank-16 LoRA on all blocks	97.15	0.037
Fast-WAM (reproduced)	96.95	0.015

LoRA across all $30 + 30$ blocks, with the adapter and the teacher losses unchanged, leaves 97.15, again indistinguishable from the baseline ($p = 0.71$).

Comparing with the $(12, 6)$ setting of Appendix C, where disabling the correction costs 0.35 rather than 0.95, the head's contribution grows with video-expert depth while the backbone's own contribution does not. This is consistent with the gradient routing of §3.4: residual supervision reaches θ' only through the adapter's dependence on the live v_{base}, so deeper video-side fine-tuning changes what v_{base} carries rather than improving the policy directly.

Is the privileged forward merely miscalibrated? On 256 held-out states with 16 independent noise draws each, we fit the best affine rescaling of the privileged prediction, $\min_{a,b} \|a\, v_{\text{teacher}} + b - v_{\text{target}}\|^2$. It removes under 11% of the teacher's error at either checkpoint ($0.0447 \to 0.0408$ at $(12, 12)$; $0.0414 \to 0.0371$ at $(12, 6)$), so the privileged forward is not a well-informed predictor on a different scale. Averaging the residual over the 16 draws at a fixed state does not make it agree with the true error either: $\cos$ moves from -0.10 to -0.20 at $(12, 12)$ and from -0.20 to -0.32 at $(12, 6)$, while the averaged correction sits at $+0.36$. The alignment of the learned correction therefore comes from the mixture in §B rather than from the residual term alone, as stated there.

E Frozen-teacher control

Replacing the live privileged forward by the same forward at the initial weights, all else identical, destabilizes training: the backbone's raw velocity MSE grows from 0.006 to 3.2×10^4 while the adapter degenerates into a counter-balancing term ($\cos(\hat{\delta}, \text{error}) = 1.00$), and the corrected policy succeeds in 0 of 400 evaluated rollouts.

F Evaluation-time perturbation study

Deterministic perturbations, no retraining, exactly reproducible: the agentview camera displaced by $(3, 2, 2)$ cm in the model frame and rotated 4° about the vertical axis (re-applied after each reset); diffuse lighting scaled $\times 0.55$ and ambient $\times 0.8$ with a 10° direction tilt; a fixed photometric shift on the policy inputs (brightness $\times 0.75$, contrast $\times 0.85$, red/blue channel gains $1.08/0.92$). All rows evaluate the PFD $(12, 6)$ checkpoint of Appendix C and the reproduced Fast-WAM baseline; each entry is the mean over the Spatial and Goal suites, with 400 trials per perturbed condition. †Not significant: two-proportion z-test, $z = 0.89$, $p = 0.37$.

Table 9: Success rate (%) under evaluation-time perturbations (clean row: 500 trials/suite).

Condition	Fast-WAM	PFD	Δ
clean	96.8	97.7	$+0.9$
photometric shift	96.5	98.0	$+1.5$
lighting change	96.5	96.8	$+0.3$
camera-pose shift	92.0	90.2	$-1.8^\dagger$

G Training cost

Table 10: Training and inference cost (LIBERO, default configuration).

Full LIBERO schedule	130,200 steps, $\approx$ 226 H100-hours (measured)
Per-step addition	one no-grad privileged forward ($\approx$ one student forward; no backward, no optimizer state)
Trainable parameters	2.37B backbone tails + 1.2M adapter (of 5B)
AdamW optimizer state	$\approx$ 20 GB (vs. $\approx$ 60 GB for a full fine-tune)
Inference overhead	$+13.3$ ms per 32-action chunk ($+5.15\%$), measured end-to-end (§4.4)