Denoising-Time Heterogeneity in VLA Action Generation: A Controlled Study via Step-Wise Expert Aggregation

Baolong Gao[1] **Yuhan Dong**[1] **Xudong Zhang**[2] **Yikai Wang**[3*]
[1]Tsinghua University [2]Tongji University
[3]Beijing Normal University

Abstract

Denoising-based Vision-Language-Action (VLA) policies are often parameterized by a single denoiser shared across denoising time. This paper provides controlled evidence that denoising-time assignment can be consequential in VLA action generation. We mix two full action experts at the velocity-field level with a fixed smooth schedule, enabling fixed-schedule interventions that keep expert capacity matched while changing when each expert is emphasized. The empirical evidence is organized as a matched chain in the studied settings: semantically distinct expert pairs test denoising-phase assignment through forward versus reversed schedules, while an anchor-free isomorphic-expert control tests schedule sensitivity without manually designed expert semantics. Experiments on LIBERO and CALVIN provide evidence of directionality under these matched interventions: aligned schedules can outperform their time-reversed counterparts under matched strength, suggesting that when each expert is emphasized can matter for action denoising. We complement these findings with a squared-error risk analysis showing that phase-wise sign changes in expert advantage imply a time-dependent optimal mixture, together with a schedule-weighted objective note explaining why copied experts can still functionally differentiate under non-constant schedules. These results provide theoretical guidance for future action-expert designs that exploit denoising-time structure.

1 Introduction

Vision-language-action (VLA) models have recently emerged as a promising route to general-purpose robot control by leveraging large-scale pretraining and multimodal conditioning. A common approach is to cast action generation as an iterative denoising process (e.g., diffusion or flow-matching formulations), where a policy refines a noisy action trajectory over a sequence of noise levels. Throughout this paper, "time" refers to denoising time or noise level, not the physical time index of the action trajectory. Despite strong empirical progress, a basic mechanism question remains under-explored: are denoising phases interchangeable for VLA action generation, or does expert usefulness change with denoising time?

This paper studies whether denoising-time assignment is consequential in VLA action generation: different noise levels may demand different computations and information sources, but this need not hold universally across all policies or benchmarks. We intentionally avoid introducing a flexible learned router and instead use a simple controlled probe: step-wise expert aggregation. We combine two full action experts with a denoising-time-dependent weight. In the full-expert experiments, each dual-expert variant starts from the same pretrained initialization by duplicating the action expert

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

module, and then uses a fixed aggregation schedule consistently during both training and inference. This setup enables matched interventions that keep architecture, initialization, and objective controlled while changing *when* each expert is emphasized.

Our empirical findings reveal an asymmetry in the studied settings: schedules that emphasize one expert at high noise and the other at low noise (an aligned or forward schedule) outperform uniform mixing in the main LIBERO setting, whereas reversing the schedule can degrade performance under matched schedule strength. This forward-reverse gap suggests that the effect is not explained by parameter count or naive ensembling alone. These comparisons intervene during both training and inference. To connect behavior to mechanism, we analyze expert denoising updates across noise levels and observe phase-structured differences consistent with denoising-time-varying expert advantage.

Motivated by these observations, we formalize step-wise aggregation as a denoising-time-dependent mixture of vector fields and analyze how per-timestep expert advantage determines the risk-minimizing mixture. We further introduce a router-free projection diagnostic in denoising-displacement space to visualize phase-wise differences consistent with the analysis. These tools are used as mechanistic explanations of schedule directionality and schedule-only sensitivity.

Prior diffusion and robotic generation works, including MoDE [16], already motivate the possibility that denoising stages differ. Our contribution is to test whether this hypothesis is observable in VLA action generation under matched schedule interventions, and to provide a risk analysis explaining why denoising-time-varying expert advantage can make schedule assignment consequential.

We use full duplicated action experts to isolate the mechanism, not to prescribe the most efficient deployment architecture. The same principle suggests lighter designs, such as schedule-weighted LoRA/adapters on a shared action expert or pretraining-time fission into smaller scheduled branches, exposing a design axis for future VLA policies. Our additional Fast-WAM control tests two narrower branches at nearly unchanged total parameter count, with a small gain over constant mixing across three seeds.

Our contributions are fourfold:

- We formulate step-wise expert aggregation as a controlled intervention for testing whether denoising-time assignment matters in VLA action generation under matched expert-capacity conditions.
- We develop a denoising-time expert-risk analysis: a closed-form pointwise optimal mixture, sufficient conditions for beating constant mixing, and an aligned-vs-reversed separation theorem under phase-wise expert advantage.
- We introduce a router-free denoising-space projection diagnostic that visualizes phase-wise expert contributions and makes the perception-to-memory handover observable without estimating u^* at deployment.
- Through semantically anchored, anchor-free, and cross-benchmark controls, we provide evidence that denoising-phase assignment—not merely expert count—contributes to the observed schedule sensitivity, offering guidance for future action-expert designs.

2 Preliminary

2.1 Flow Matching

Let x_0 be a clean action trajectory, $x_1 \sim \mathcal{N}(0, I)$, and c the visual, language, and optional proprioceptive/history conditioning. On the linear path

$$x_t = (1-t)x_0 + tx_1, \qquad u_t = x_1 - x_0, \qquad u^*(x,t,c) = \mathbb{E}[u_t \mid x_t = x, c], \tag{1}$$

the model learns v_θ by minimizing $\mathbb{E}\|v_\theta(x_t,t,c) - u_t\|^2$. At the population level, replacing u_t by u^* changes this squared loss only by a term independent of the predictor. Sampling integrates $dx/dt = v_\theta(x,t,c)$ from noise ($t=1$) to data ($t=0$).

2.2 Time-scheduled Mixture of Action Experts

We parameterize the field as $v_\theta = w v_{\theta_1} + (1-w) v_{\theta_2}$, where the branches are complete action denoisers, not token-level MoE blocks. A bounded, continuous $w(t) \in [0,1]$ preserves local Lipschitz

continuity in x when both expert fields satisfy it. Standard growth conditions are additionally needed for existence over the entire integration interval.

3 Method

3.1 Controlled Aggregation of Phase-Specialized Experts

We study denoising-based action generation under the hypothesis that different noise regimes require different action-generation behavior. To probe this effect in a controlled manner, our policy uses a shared visual-language encoder for the observation o (images, language, and proprioception) and two full downstream action denoisers obtained by duplicating the action expert module. For a noisy action trajectory $x_t \in \mathbb{R}^{H\times d}$ at diffusion time $t \in (0, 1]$, the two experts predict velocity fields $v_I(x_t, t)$ and $v_{II}(x_t, t)$. We compose them only at the output level, keeping the training objective, numerical solver, and upstream encoder architecture fixed.

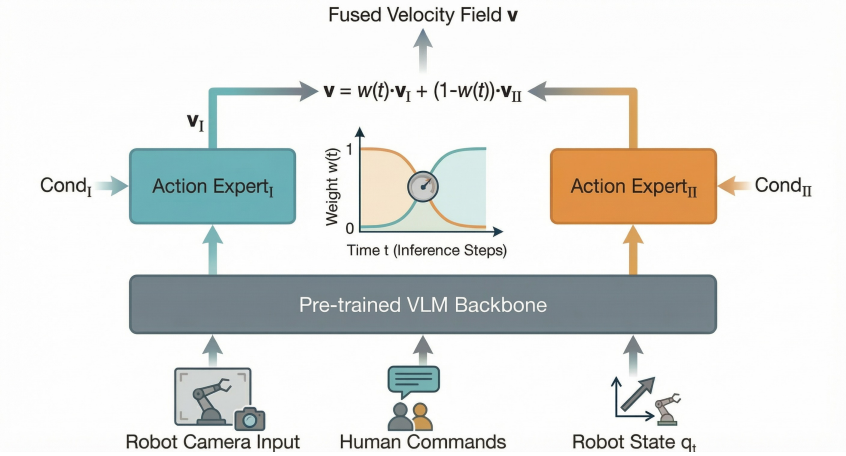


Figure 1: Full action-expert aggregation. The expert conditions are $\mathrm{Cond}_I = e(t) + c_1$ and $\mathrm{Cond}_{II} = e(t) + c_2$. They modulate each expert internally; the scalar schedule mixes the two output velocity fields.

At each denoising step, a scalar schedule $w(t) \in [0, 1]$ mixes the expert velocity fields:

$$v(x, t) = w(t)v_I(x, t) + (1 - w(t))v_{II}(x, t). \tag{2}$$

We compare three matched interventions: an **aligned** schedule, where expert I receives larger weight at high-noise steps ($t \to 1$) and expert II dominates low-noise refinement ($t \to 0$); **constant** mixing, $w(t) \equiv 0.5$, which removes denoising-time assignment; and a **reversed** schedule $w^{rev}(t) = w(1-t)$, which swaps the stage assignment and serves as a mechanistic mismatch control. Unless otherwise stated, the aligned schedule is an S-shaped function

$$s(t) = 3t^2 - 2t^3, \quad w(t) = \beta + \kappa s(t), \tag{3}$$

with (β, κ) chosen so that $w(t) \in [0, 1]$. For controlled reversal, we impose

$$2\beta + \kappa = 1 \implies w(t) + w(1 - t) = 1. \tag{4}$$

We use $\kappa = 1 - 2\beta$, so $w(t; \beta) = \beta + (1 - 2\beta)s(t)$ and $w(1 - t; \beta) = w(t; 1 - \beta)$. Center symmetry gives $\int_0^1 w(t)dt = 1/2$: forward and reverse have equal integrated weight under uniform time. This equality does not automatically hold under a shifted training density or an asymmetric finite solver grid. Under nonuniform measures, the schedules retain identical model capacity and complementary weights, while cumulative exposure depends on the time density and solver grid.

Anchored probes. To make phase-wise differences observable, we optionally provide the two experts with distinct anchors, such as image foresight (IF) [19] and episodic memory retrieval (Mem) [20]. Anchors are analysis probes rather than part of the aggregation rule in Eq. (2). They are

injected only into expert-specific action modules, not the shared prefix encoder. Let $e(t)$ be the time embedding and c_k the anchor for expert k (with $c_k = 0$ in the unconditional setting). We form

$$\gamma_k(t) = e(t) + c_k, \tag{5}$$

and use it to modulate the normalization layers inside each expert transformer block:

$$h \mapsto \mathrm{Norm}(h; \gamma_k(t)). \tag{6}$$

We implement $\mathrm{Norm}(\cdot;\cdot)$ as AdaRMS-style scale modulation. In Figure 1, $\mathrm{Cond}_I = \gamma_1(t)$ and $\mathrm{Cond}_{II} = \gamma_2(t)$; neither is a routing weight. Each c_k is projected to the dimension of $e(t)$ before addition. For Ori+Ori, $c_k = 0$ removes only the additional anchor, while observations and time conditioning remain. Appendix C details the conditioning interface and experimental settings.

Training and inference. For each schedule condition in the full-expert probe, we initialize from the same pretrained policy, duplicate the action expert module into two branches, and train the resulting dual-expert model end-to-end with the standard flow-matching objective. The narrower Fast-WAM initialization is described separately in Appendix C. Given a demonstration trajectory x_0 and Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, we sample $t \sim p(t)$ and form

$$x_t = (1-t)x_0 + t\epsilon, \quad u_t = \epsilon - x_0. \tag{7}$$

Each expert predicts a conditional velocity field from (o, x_t, t) and its optional anchor; the fixed schedule in Eq. (2) aggregates the two predictions before applying the squared-error loss

$$\mathcal{L}_{FM} = \mathbb{E}_{t,x_0,\epsilon}\left[||v(x_t,t) - u_t||_2^2\right]. \tag{8}$$

The schedule $w(t)$ is never learned. In the full-expert controls it is fixed before training and reused at inference. Thus, matched schedule variants differ only in the prescribed aggregation schedule, while sharing the same initialization, architecture, training objective, solver, and evaluation protocol. Because aggregation happens before the loss is applied, the schedule is a training-and-inference intervention: during fine-tuning it also changes the effective timestep weighting seen by each branch. At inference, we sample $x_1 \sim \mathcal{N}(0, I)$ and integrate from $t = 1$ to $t = 0$ with the same solver as the single-expert model, replacing only the vector field by the aggregated field in Eq. (2).

4 Theory: Denoising-Time Heterogeneity in Expert Risk

We use a squared-error surrogate to formalize the central mechanism tested by our schedules: expert advantage may vary with denoising time. For any trained pair of expert velocity fields v_1, v_2 and the flow-matching target u^* defined by the data path, the population mixture risk for scalar $w \in [0,1]$ at denoising time t is

$$\mathcal{R}_t(w) = \mathbb{E}_{x_t,c}\Big[\big\|wv_1(x_t,t,c) + (1-w)v_2(x_t,t,c) - u^*(x_t,t,c)\big\|_2^2\Big]. \tag{9}$$

Define $A(t) = \mathbb{E}\|v_1 - u^*\|_2^2$, $B(t) = \mathbb{E}\|v_2 - u^*\|_2^2$, and $C(t) = \mathbb{E}\|v_1 - v_2\|_2^2$ (expectations over (x_t, c) induced by the forward process at fixed t; Appendix H). The gap $\Delta(t) = B(t) - A(t)$ is the theoretical version of phase-wise expert advantage: positive values favor expert 1, negative values favor expert 2.

Pointwise optimum. If $C(t) > 0$, the unconstrained minimizer of $\mathcal{R}_t(w)$ is $w_t^* = \frac{1}{2} + \frac{B(t)-A(t)}{2C(t)}$, and the deployable weight is $w_t^{*,[0,1]} = \Pi_{[0,1]}(w_t^*)$. Time dependence is governed by the clipped ratio $\Pi_{[0,1]}(1/2 + \Delta(t)/(2C(t)))$, not by $\Delta(t)$ alone. A sign change in Δ on nonzero-measure phases with $C > 0$ implies different optimal weights. Variation in magnitude alone need not: Δ/C can stay constant, or clipping can produce the same boundary solution.

Aligned vs. reversed as a phase-assignment test. Under phase-wise dominance of one expert on $t > \frac{1}{2}$ and the other on $t < \frac{1}{2}$ (Assumption H.3), any aligned center-symmetric schedule w ($w(t) + w(1-t) = 1$) that differs from its reversal on a set with positive p-measure satisfies $\mathcal{R}(w) < \mathcal{R}(w^{\mathrm{rev}})$ for $\mathcal{R}(w) = \mathbb{E}_{t\sim p(t)}[\mathcal{R}_t(w(t))]$; see Theorem H.6. This is a comparison of fixed expert functions. Separately trained forward and reverse models do not by themselves test this theorem.

Coverage of anchored and isomorphic experts. For IF+Mem and DA+Mem, distinct anchors provide different inductive biases that make phase-wise advantage easier to observe. For Ori+Ori, the two branches begin as copies of the same action expert, but the fixed non-constant schedule is used during training as well as inference; it gives the two branches different timestep-weighted effective objectives (Appendix H). Ori+Ori therefore tests whether schedule sensitivity can arise without hand-designed semantic anchors. This control tests schedule sensitivity independently of manually chosen IF/Mem semantics; it does not establish representation disentanglement.

Full proofs, the schedule-weighted objective note, Pareto/Min-SNR context, sufficient conditions for beating constant mixing, and the formal scalarized diagnostic appear in Appendix H. **Scope.** The result is a mechanistic statement in squared-error denoising-field space, not a direct guarantee on task success rate.

5 Experiments

We organize the experiments as a chain of controlled comparisons: **(i)** multi-seed LIBERO/CALVIN readouts, **(ii)** a DA+Mem robustness stress test, **(iii)** router-free velocity-field diagnostics, **(iv)** an anchor-free Ori+Ori control, and **(v)** a near-capacity-matched Fast-WAM control. Reported success rates are behavioral readouts of schedule-level interventions in these settings; the theory concerns squared-error field risk.

5.1 Experimental Logic

The anchored IF/Mem and DA/Mem pairs probe different inductive biases, while Ori+Ori removes semantic anchors. Forward/mean/reverse training conditions control architecture and initialization within each experiment. Fast-WAM further controls total action capacity through two narrower branches. For a prescribed schedule to beat constant mixing, variation in expert advantage is insufficient by itself: the sufficient inequality in Appendix H must also hold.

5.2 Experimental Setup

Benchmarks and protocol. We evaluate on LIBERO [23] (Spatial, Object, Goal, and 10) and CALVIN [24] ABC-D. CALVIN policies are trained on environments A–C and evaluated zero-shot on D with 1,000 five-instruction rollout chains. We report Success Rate: LIBERO is averaged over suites; CALVIN columns 1–5 show a representative chain-completion breakdown, and cross-method comparisons use three-run mean $\pm$ std. Relative-effect tables report percentage-point (pp) gaps against matched mean or reverse references.

Model, experts, and schedules. The full-expert LIBERO experiments use a $\pi_{0.5}$ backbone [25], while CALVIN uses π_0. The additional Fast-WAM study is a separate backbone control [22]. We instantiate Image Foresight (IF) [19] for global spatial relations, Memory (Mem) [20] for episode-level history, and Depth-Anything (DA) [21] for robustness tests; Ori denotes the monolithic policy without expert aggregation. Stage-aware aggregation uses the S-curve family $w(t;\beta)$: Forward-S emphasizes IF/DA at high noise ($t \to 1$) and Mem at low noise ($t \to 0$), while Reverse-S swaps this assignment under mirrored schedule strength.

Intervention and compute. Unless stated otherwise, aggregation is applied at the velocity-field level during training and inference, adds no learned router, and uses the same fixed schedule in both phases. Matched comparisons start from the same pretrained initialization, so the controlled variable is the prescribed schedule rather than expert capacity; cumulative weight matching has the measure-specific qualification in Sec. 3. Our full dual-action-expert probe increases inference time by about $1.4\times$. The narrower Fast-WAM split matches action capacity but has a measured latency penalty (Sec. 5.7); the execution cost depends on both architecture and implementation. Primary multi-seed results are in Tables 1 and 2; LIBERO Ori+Ori and DA+Mem controls are in Tables 3 and 11; full IF+Mem sweeps and diagnostics are in Appendix E. Each full-expert LIBERO experiment fine-tunes $\pi_{0.5}$ for about one day on **8$\times$ NVIDIA A800 GPUs**; additional experimental settings are in Appendix C.

5.3 Evidence I: Multi-Seed Behavioral Evidence (LIBERO and CALVIN)

We first establish cross-benchmark, multi-run evidence under matched protocols. Table 1 reports LIBERO success as a downstream readout: the monolithic baseline (Ori) and single-expert IF-only / Mem-only rows show that both anchors are individually useful, but the targeted question is whether *when* each expert is emphasized along denoising time matters for the same expert pair.

Table 1: Main results on LIBERO. We compare single-expert baselines, constant expert averaging, and denoising-time scheduled aggregation. Suite-specific columns report the breakdown from the seed whose overall average is closest to the 4-seed mean, while the final column reports the mean $\pm$ std of the overall average across four seeds.

Method	Spatial	Object	Goal	10	Avg	4-seed Avg
Single-Expert Baselines						
Ori (Monolithic)	98.80	98.20	98.00	92.40	96.85	96.82 ± 0.14
IF-only	98.20	98.80	98.80	93.20	97.25	97.36 ± 0.13
Mem-only	99.00	98.20	98.00	93.00	97.05	97.02 ± 0.13
Ablation / Combinations						
IF+Ori $w(t; \beta = 0.2)$	99.40	97.40	98.20	93.20	97.05	97.01 ± 0.12
Ori+Mem $w(t; \beta = 0.2)$	99.60	99.00	98.40	92.80	97.45	97.40 ± 0.15
IF+Mem (Mean) $w(t) \equiv 0.5$	99.00	97.40	98.80	92.80	97.00	96.91 ± 0.13
Aligned schedule						
IF+Mem $w(t; \beta = 0.2)$	99.60	98.00	98.80	94.00	**97.60**	$\mathbf{97.64 \pm 0.12}$

Under the same IF+Mem experts, the aligned S-curve $w(t; \beta = 0.2)$ reaches 97.64 ± 0.12 over four seeds, above both monolithic Ori (96.82 ± 0.14) and static IF+Mem mean mixing (96.91 ± 0.13). Replacing the S-curve by mean aggregation reduces the representative Avg. from 97.60 to 97.00 and LIBERO-10 from 94.00 to 92.80, so the effect is not explained by expert count alone. We treat these gaps as evidence for denoising-phase assignment rather than deployment-level superiority on a saturated benchmark.

Table 2 provides an external full-VLA check under the standard 1,000-rollout ABC-D protocol. We report a representative 1–5 instruction-chain breakdown; cross-method comparisons use the three-run mean $\pm$ std. IF-only and Mem-only improve over π_0 by +1.95 and +2.97 pp. IF+Mem forward (IF at high noise, Mem at low noise) reaches 81.04 ± 0.41, above Mem-only by +1.37 pp and π_0 by +4.34 pp, while matched Mem+IF reverse reaches 77.89 ± 0.36. This gives a supporting directionality readout beyond LIBERO, though broader cross-backbone and cross-embodiment generality remains outside our evidence. Ori+Ori rows are analyzed with the LIBERO Ori+Ori control in §5.6.

Table 2: CALVIN ABC-D results under the standard 1,000-rollout evaluation protocol. Columns 1–5 report the success rate after completing the first k consecutive language instructions for one representative run. The final columns report the mean $\pm$ std of the overall average across three runs and its percentage-point gap over the π_0 baseline. IF-only and Mem-only are single-anchor baselines; Ori+Ori tests denoising-time scheduling without semantic anchors; IF+Mem vs. Mem+IF tests directionality under matched S-curve strength.

Method	1	2	3	4	5	3-run Avg.	$\Delta\pi_0$
π_0 (baseline)	93.8	85.0	76.7	68.1	59.9	76.70	+0.00
IF-only	96.2	86.7	78.6	69.6	60.8	78.65 ± 0.35	+1.95
Mem-only	94.1	87.7	81.0	72.5	63.9	79.67 ± 0.42	+2.97
Ori+Ori (Mean) $w(t) \equiv 0.5$	94.6	86.8	78.8	69.7	63.2	78.41 ± 0.27	+1.71
Ori+Ori (S-curve)	95.5	88.4	81.1	72.2	62.6	79.69 ± 0.33	+2.99
IF+Mem (Forward S-curve)	95.9	89.2	82.3	73.1	64.1	$\mathbf{81.04 \pm 0.41}$	**+4.34**
Mem+IF (Reversed S-curve)	94.4	86.4	78.2	69.3	61.6	77.89 ± 0.36	+1.19

Additional diffusion-policy controls. To test whether schedule sensitivity is tied to our main VLA implementation, we also ran controlled low-dimensional experiments in the `diffusion_policy`

Robomimic benchmark. These runs are not an additional full-VLA benchmark (inputs are low-dimensional states rather than image–language observations), but serve as matched action-generation controls in a different diffusion-policy training stack. Under matched architecture, training recipe, and evaluation protocol, changing only the denoising-time mixing schedule yields measurable differences on `transport_lowdim_abs` and `tool_hang_lowdim_abs`; see Appendix F.

5.4 Evidence II: DA+Mem Robustness and Directional Sensitivity

Replacing IF by Depth Anything [21] and training DA+Mem for 40k steps produces a weaker directionality pattern. Relative to constant mixing (96.35%), the forward/reverse gains are +0.65/+0.00 pp for $\beta = 0.2/0.8$ and +0.60/+0.00 pp for $0.3/0.7$, but +0.35/+0.35 pp for $0.1/0.9$. Thus, forward superiority is not uniform across strengths. The full numerical sweep is in Appendix D; Figure 2 presents the projection diagnostic.

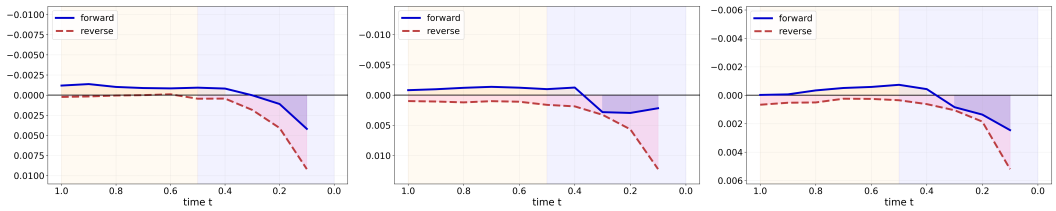


Figure 2: Velocity-field projection diagnostic for DA+Mem under mirrored S-curve schedules. The forward ordering preserves a perception-to-memory handover more clearly than the reversed ordering.

Figure 2 uses the same router-free projection construction as Appendix E: DA contributes more along the shared reference direction in high-noise phases and Mem in late denoising under Forward-S, while Reverse-S disrupts this handover. IF+Mem shows a similar shift from IF to Mem alignment in the two mirrored schedule pairs in Figure 4, with Reverse-S favoring Mem earlier in denoising.

5.5 Evidence III: Mechanistic Diagnostics and Theory Connection

At each denoising time t, the two experts produce velocity fields v_1 and v_2 (IF/Mem or DA/Mem), and the deployed field is $v_w = w(t)v_1 + (1 - w(t))v_2$. Because the flow-matching target u^* is unknown at test time, we scalarize unweighted expert displacements along a *shared* unit direction u derived from the forward mixed trajectory (formal definition: Appendix H, Eq. (31)–(32)), and report the normalized stage-advantage $\Delta g_0(t)$ between the two experts.

Under Forward-S, $\Delta g_0(t)$ tends to be positive at high noise and negative as $t \to 0$, consistent with a perception $\to$ memory handover; Reverse-S shifts relative alignment toward Mem earlier in denoising. This is the qualitative pattern predicted by Sec. 4: the pointwise optimal weight depends on $A(t), B(t), C(t)$, and when $B(t) - A(t)$ changes sign across phases, aligned schedules can be formally separated from their time reversals (Theorem H.6, Appendix H). The same appendix also explains why schedule weighting can make same-family Ori+Ori branches functionally different during fine-tuning.

5.6 Evidence IV: Anchor-Free Schedule Sensitivity (Ori+Ori)

We finally isolate schedule *shape* without semantic anchors: two isomorphic Ori experts are initialized from the same pretrained action expert and trained under the same recipe, while the fixed aggregation schedule is varied consistently across training and inference. Since the branches share the same expert family, this control asks whether denoising-time scheduling remains consequential without manually supplied IF/Mem-style inductive-bias labels.

On LIBERO (Table 3), the 3-seed Ori+Ori S-curve result improves over 3-seed mean aggregation. On CALVIN (Table 2), the same pattern holds among the Ori+Ori rows: S-curve aggregation (79.69 ± 0.33) improves over constant averaging (78.41 ± 0.27) under the shared π_0 initialization. We interpret this as evidence that denoising-time scheduling can matter even when expert semantics are not hand-designed; schedule-weighted objectives can allow same-family branches to become functionally different when the underlying denoising tasks are heterogeneous. The Ori+Ori result

Table 3: Isomorphic-expert control on LIBERO: non-constant schedules vs. constant mean. Ori+Ori rows report mean ± std over three seeds.

Method	**Avg. Success (%)**
Baseline (Monolithic)	96.85
Ori+Ori (Mean) $w(t; \beta = 0.5)$	96.45 ± 0.16
Ori+Ori (S-curve) $w(t; \beta = 0.2)$	97.35 ± 0.09

complements the anchored directionality evidence by demonstrating schedule sensitivity without IF/Mem conditioning; it does not establish representation disentanglement.

5.7 Evidence V: Anchor-Free Fast-WAM with Matched Action Capacity

We split the ActionDiT of Fast-WAM [22] into two narrower isomorphic experts without IF/Mem/DA anchors. Their combined 1,021,311,374 parameters differ from the monolithic 1,020,886,023 by only 0.0417%. Split-Mean and Split-S share the same architecture and seed-specific initial weights, with $w_M = 0.5$ and $w_F = 0.2 + 0.6s(t)$ respectively. Each narrow branch has width 576 and 30 layers. The two branches are initialized independently, while the matched Split-Mean and Split-S runs reuse the same paired initial weights for each seed. This tests scheduling in a different backbone without duplicating a full-sized action expert. Each model/seed uses the full 2,000-trial LIBERO protocol (40 tasks with 50 trials each).

Table 4: Capacity-controlled Fast-WAM on LIBERO (%). Variability is sample standard deviation over three training seeds. The monolithic row is our reproduction of the official model.

Method	Seed 1	Seed 2	Seed 3	Mean ± SD
Monolithic	97.20	97.40	97.50	97.37 ± 0.15
Split-Mean	97.10	97.60	97.40	97.37 ± 0.25
Split-S	97.55	97.80	97.70	97.68 ± 0.13

Split-S exceeds both controls in all three seeds, averaging +0.32 pp over each. Constant splitting gives no mean improvement over the monolithic reproduction. The gain is small and consistent across these seeds; statistical significance is not established. A same-A800 synthetic-input microbenchmark gives 570.95 ms for monolithic and 1057.81 ms for Split-S (approximately $1.85\times$), despite nearly equal peak memory (23.38 vs. 23.43 GiB). The measured implementation therefore trades higher inference latency for nearly unchanged parameter count (Appendix C.4).

Additional frozen-checkpoint schedule interventions and target-error diagnostics are reported in Appendices A and B.

6 Related Work

6.1 Diffusion and Flow Matching for Policy Learning

Generative policy learning has become central to robotic manipulation because it can model multimodal action distributions. Diffusion Policy [2] uses DDPM-style denoising to generate action trajectories, while recent work increasingly adopts Flow Matching and Rectified Flow [4, 11, 12], which learn velocity fields between noise and data distributions and enable faster deterministic sampling.

VLA policies including Octo [10] and π_0 [1] use generative action modules. DiT [7] and SiT [5] provide related Transformer-based generative architectures. We examine stage-dependent expert usefulness within action generation.

6.2 Denoising-Level Heterogeneity and Expert Denoisers

In diffusion generation, t indexes denoising level rather than trajectory time, and different noise regimes can impose different optimization and information demands. Min-SNR [3] addresses

conflicting timestep gradients; FreeU [9], SDXL [8], Dynamic CFG [6], and spectral-autoregression analyses [17] further suggest that sampling behavior changes across denoising time.

Several image-generation methods explicitly exploit this structure: eDiff-I [13] trains noise-specialized expert denoisers, MEME [14] assigns architectures to frequency-defined intervals, Remix-DiT [15] mixes basis Diffusion Transformers into stage-adapted denoisers, and Ambient Diffusion Omni [18] varies data-quality regimes across stages. These works motivate denoising-level heterogeneity but do not provide controlled analyses of action-expert aggregation in VLA control.

Most closely related, MoDE [16] learns noise-conditioned top-k routing among internal MLP experts, analyzes noise-dependent expert utilization, evaluates 2/4/6/8 experts, and uses expert caching for efficiency. Building on this evidence for noise-dependent specialization, our study examines aggregation of full action-expert fields through fixed schedule interventions.

7 Limitations

This study is a controlled mechanism analysis rather than a deployment-optimized VLA system. The full two-expert probe adds inference cost (about $1.4\times$ over the original single-action-expert policy), and the theory explains squared-error denoising-field risk rather than directly guaranteeing task success. The near-capacity-matched Fast-WAM split also increases measured inference latency, and its three-seed gain is small on the nearly saturated LIBERO benchmark. The projection diagnostic assumes alignment with a reference direction and serves as an explanatory proxy. Our evidence is limited to simulation; broader validation across additional backbones, embodiments, and adaptive routing remains future work. Scaling beyond two experts and deployment on real robots require further evaluation.

8 Conclusion

We presented step-wise expert aggregation as a controlled probe for denoising-time heterogeneity in VLA action generation. Denoising-time order can matter under matched interventions: forward/reverse comparisons favor perception-oriented experts at high noise and memory-oriented refinement at low noise, while Ori+Ori controls show schedule sensitivity without hand-designed semantic anchors. DA+Mem gives a qualifier: a similar directionality readout appears with a depth-based expert, but is less uniform across schedule strengths. The near-capacity-matched Fast-WAM control further supports schedule sensitivity with narrower, anchor-free experts.

Acknowledgments and Disclosure of Funding

We thank FreiNexus for providing the computing infrastructure used for part of the model training and validation experiments in this work, and part of this work was done while Baolong Gao was an intern at FreiNexus.

References

[1] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[2] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

[3] Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7441–7451, 2023.

[4] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[5] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pages 23–40. Springer, 2024.

[6] Pinelopi Papalampidi, Olivia Wiles, Ira Ktena, Aleksandar Shtedritski, Emanuele Bugliarello, Ivana Kajic, Isabela Albuquerque, and Aida Nematzadeh. Dynamic classifier-free diffusion guidance via online feedback. *arXiv preprint arXiv:2509.16131*, 2025.

[7] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[8] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[9] Chenyang Si, Ziqi Huang, Yuming Jiang, and Ziwei Liu. Freeu: Free lunch in diffusion u-net. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4733–4743, 2024.

[10] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[11] Fan Zhang and Michael Gienger. Affordance-based robot manipulation with flow matching. *arXiv preprint arXiv:2409.01083*, 2024.

[12] Qinglun Zhang, Zhen Liu, Haoqiang Fan, Guanghui Liu, Bing Zeng, and Shuaicheng Liu. Flowpolicy: Enabling fast and robust 3d flow-based policy via consistency flow matching for robot manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 14754–14762, 2025.

[13] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. eDiff-I: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

[14] Yunsung Lee, Jin-Young Kim, Hyojun Go, Myeongho Jeong, Shinhyeok Oh, and Seungtaek Choi. Multi-architecture multi-expert diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 13427–13436, 2024.

[15] Gongfan Fang, Xinyin Ma, and Xinchao Wang. Remix-DiT: Mixing diffusion transformers for multi-expert denoising. In *Advances in Neural Information Processing Systems*, 2024.

[16] Moritz Reuss, Jyothish Pari, Pulkit Agrawal, and Rudolf Lioutikov. Efficient diffusion transformer policies with mixture of expert denoisers for multitask learning. In *International Conference on Learning Representations*, 2025.

[17] Sander Dieleman. Diffusion is spectral autoregression. 2024. URL https://sander.ai/2024/09/02/spectral-autoregression.html.

[18] Giannis Daras, Adrian Rodriguez-Munoz, Adam Klivans, Antonio Torralba, and Constantinos Daskalakis. Ambient Diffusion Omni: Training good models with bad data. *arXiv preprint arXiv:2506.10038*, 2025.

[19] Chongkai Gao, Zixuan Liu, Zhenghao Chi, Junshan Huang, Xin Fei, Yiwen Hou, Yuxuan Zhang, Yudi Lin, Zhirui Fang, Zeyu Jiang, et al. VLA-OS: Structuring and dissecting planning representations and paradigms in vision-language-action models. *arXiv preprint arXiv:2506.17561*, 2025.

[20] Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. MemoryVLA: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2508.19236*, 2025.

[21] Haotong Lin, Sili Chen, Junhao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

[22] 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.

[23] 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.

[24] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *arXiv preprint arXiv:2112.03227*, 2021.

[25] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

A Supplementary Inference-Schedule Cross-Evaluation

We cross-evaluate existing IF+Mem checkpoints on $\pi_{0.5}$ under Forward (F), Mean (M), and Reverse (R) inference schedules. Rows identify the schedule used to train the evaluated models; columns identify the aggregation schedule used at inference. Within each row, model parameters are frozen and only the inference aggregation schedule is changed. The cross-evaluation requires no further parameter updates.

Table 5: Frozen-checkpoint cross-evaluation on LIBERO (%). Entries are aggregate success rates; bold indicates the largest reported value in each row.

Training \ inference	Forward	Mean	Reverse
Forward	**97.64**	93.82	87.25
Mean	95.35	**96.91**	93.77
Reverse	91.13	92.60	**96.80**

Each row has its largest reported success rate under the matching inference schedule. The Forward-trained models favor Forward inference, while the Mean- and Reverse-trained models favor their respective matching schedules. This pattern is consistent with sensitivity to the compatibility between learned expert functions and inference assignment.

B Fixed-Pair Error and Behavioral Counterfactuals

For one frozen Forward-S Fast-WAM checkpoint, we measure masked action MSE on 128 held-out samples, two noise draws, 21 uniform noise levels and the actual 10-step solver grid (7,936 records). Every schedule reuses the same raw v_1, v_2, x_t, u_t. Phase summaries in Table 6 reveal a sign change without assigned semantic roles. The statistic $\overline{\widehat{w}^*}$ averages sample-wise clipped optima; it is not the optimum computed from phase-averaged errors.

Table 6: Pathwise target-error diagnostic for one frozen, anchor-free pair. High/mid/low use $t \geq 0.55$, $0.45 < t < 0.55$, and $t \leq 0.45$. Values are rounded independently.

Noise phase	Expert 1 MSE	Expert 2 MSE	$\widehat{\Delta}$	$\overline{\widehat{w}^*}$
High	0.5399	3.7699	+3.2300	0.701
Mid	0.8714	0.8780	+0.0067	0.501
Low	8.0026	1.0837	−6.9189	0.302

Reaggregating the fields on the solver measure gives risks 0.00505 for Forward, 0.25641 for Mean, and 1.00778 for Reverse. The sample-cluster bootstrap 95% intervals for Mean−Forward and Reverse−Forward are [0.24837, 0.25496] and [0.99090, 1.01724]. The shifted training measure has the same ordering; full estimates and the 500-resample protocol are in Appendix C.3.

For the same frozen checkpoint, changing only the inference weights from Forward to Reverse lowers success from 97.30% (1,946/2,000) to 86.70% (1,734/2,000): 212 fewer successes, or a 10.60 pp drop. Forward suite rates are 98.40/99.60/97.20/94.00 for Spatial/Object/Goal/10. This single-checkpoint mechanism experiment is evaluated independently of the three-seed comparison in Table 4. It demonstrates non-exchangeability of the learned branches across stages under a fixed-capacity intervention; discrete-grid cumulative weights need not be identical (Sec. 3).

Scope beyond two experts. The formal extension is $v(x,t) = \sum_{k=1}^{K} w_k(t) v_k(x,t)$ with $w_k \geq 0$ and $\sum_k w_k = 1$. A held-out expert-by-stage risk matrix can guide partitions or router initialization; empirical validation for $K > 2$ remains future work. The two-expert schedule is indexed by noise level, not task identity. It supplies a diagnostic design principle rather than an empirically validated many-expert router.

C Implementation and Evaluation Details

C.1 Conditioning interface and anchored probes

The main LIBERO probe uses $\pi_{0.5}$; the CALVIN control uses π_0. Each has two full action experts sharing upstream visual-language processing. The expert-specific condition is $\gamma_k(t) = e(t) + c_k$. The anchor is projected into the time-embedding space and enters AdaRMS-style modulation inside the action transformer, not the shared prefix encoder. All models still receive their common observations and denoising time.

Table 7: Expert-specific anchor representations and conditioning interfaces. Each anchor is projected into the time-embedding space before modulation of the corresponding action expert.

Branch	Source feature	Conditioning interface
IF	Image-foresight representation, following VLA-OS [19]	Project to the time-embedding dimension; expert-specific AdaRMS modulation
Mem	Retrieved episodic-memory representation, following MemoryVLA [20]	Same addition/modulation interface, with a separate anchor
DA	Depth-based representation, following Depth Anything [21]	Same interface; replaces IF in the DA+Mem probe
Ori	No additional semantic anchor	$c_k = 0$; common observations and $e(t)$ remain

Full-expert LIBERO fine-tuning uses approximately one day on 8 NVIDIA A800 GPUs per experiment; DA+Mem uses 40k training steps. The main LIBERO results summarize four training seeds, and CALVIN summarizes three runs. The supplementary cross-evaluation in Appendix A uses existing frozen checkpoints and changes only the inference aggregation schedule. Repeated evaluations of a fixed checkpoint measure evaluation variability, rather than variability across independently trained models.

C.2 Fast-WAM split architecture and training

Each narrow ActionDiT has hidden width 576, FFN width 2304, 30 layers, 24 heads, and 510,655,687 parameters. The two branches have independent random initializations in the split architecture; matched Split-S and Split-Mean conditions reuse the same paired step-zero weights for a given seed. This differs from the copied pretrained full experts in the full-expert Ori+Ori control. The video backbone is Wan2.2 TI2V 5B (30 layers, width 3072, 24 heads). No additional IF/Mem/DA condition is introduced.

Table 8: Fast-WAM training and evaluation settings.

Setting	Value
Data	LIBERO Spatial/Object/Goal/10 no-noop LeRobot data
Views	Agent and wrist; each 224×224, concatenated to 224×448
Sequence; action/proprio	33 frames; dimensions 7/8
Batch and precision	12 per GPU, global 96; bf16; 8 A800 80GB GPUs
Learning rate; schedule	10^{-4}; cosine
Training length	10 epochs; maximum 28,930 steps
Parallelism	Accelerate with DeepSpeed ZeRO-1
Noise scheduler	1,000 training timesteps; training/inference shift 5
Evaluation	40 tasks $\times$ 50 trials; 10 inference steps; 10 executed actions before replanning

C.3 Fixed-pair pathwise risk protocol

The mechanism experiment freezes the Forward-S checkpoint at step 26,000. We construct a deterministic 5% held-out split with seed 42 and sample 128 examples without replacement. Two Gaussian action-noise draws are used per example. On $x_t = (1-t)x_0 + t\epsilon$, the pathwise target is $u_t = \epsilon - x_0$. Both raw expert fields are read at the same state and conditioning, and MSE is normalized over valid action tokens and dimensions. The 21-point uniform grid and 10-point actual solver grid are logged separately, yielding $128 \times 2 \times (21 + 10) = 7,936$ records over eight complete shards.

For each record we measure $\widehat{A} = \|v_1 - u_t\|^2$, $\widehat{B} = \|v_2 - u_t\|^2$, $\widehat{C} = \|v_1 - v_2\|^2$, and

$$\widehat{w}^* = \Pi_{[0,1]}\left(\frac{1}{2} + \frac{\widehat{B} - \widehat{A}}{2\widehat{C}}\right) \tag{10}$$

when $\widehat{C} > 0$, with a numerical safeguard near zero. The phase table averages these record-level statistics via timestep summaries. In particular, averaging clipped optima is not equivalent to optimizing a phase-averaged risk. At the population level, pathwise target risk and conditional-mean target risk differ by an irreducible conditional variance independent of the predictor. The finite-sample estimates here remain diagnostics and do not measure the exact unobserved conditional field.

The training measure combines the shift-induced density $q(t) = 5/(5-4t)^2$ and the scheduler weight $\omega(t) = \max\{\exp[-2(t-0.5)^2] - \omega_{\min}, 0\}$, then normalizes trapezoidal integration over the uniform grid. Here $\omega_{\min}$ is the minimum of that exponential over the scheduler's shifted 1,000-point training grid. The solver measure uses the actual shifted time points and step lengths:

$$\overline{R}_{\text{solver}} = \frac{\sum_k |\Delta t_k| R(t_k)}{\sum_k |\Delta t_k|}. \tag{11}$$

This is a diagnostic on interpolated data-path states sampled at solver noise levels, not an integral along each schedule's generated rollout trajectory. Complementary schedules have equal continuous uniform-time mass; equality is not implied under these nonuniform measures.

Table 9: Integrated pathwise FM risks for the same frozen expert fields. Brackets denote sample-cluster bootstrap 95% intervals (500 replicates).

Mixing	Solver measure	Training measure
Forward-S	0.00505 [0.00404, 0.00638]	0.00568 [0.00481, 0.00670]
Mean	0.25641 [0.25345, 0.26023]	0.34389 [0.33737, 0.35148]
Reverse-S	1.00778 [0.99585, 1.02229]	1.35473 [1.32889, 1.38409]

Each sample is the bootstrap cluster, retaining its two noise draws and all time points. The reported intervals use the 2.5th/97.5th percentiles of 500 bootstrap replicates. On the solver measure, Mean−Forward is 0.25135 [0.24837, 0.25496] and Reverse−Forward is 1.00273 [0.99090, 1.01724]. Differences are computed before rounding individual risks. The identity $R(w) = B - w\Delta - w(1-w)C$ agrees with direct reaggregation, with integration discrepancies below 1.8×10^{-8} in the numerical consistency check.

C.4 Training cost and inference microbenchmark

Representative training logs give approximately 215.4 GPU-hours for the baseline (6 A800s, 35.90 hours) and 222.3 GPU-hours for Split-S (8 A800s, 27.79 hours), a 3.2% difference. Different GPU counts prevent a strict wall-clock comparison. These logs describe particular runs and are not a complete budget for all seeds, ablations, and development experiments.

The inference microbenchmark uses one NVIDIA A800-SXM4-80GB, bf16, deterministic synthetic images of shape $[1, 3, 224, 448]$, proprioception $[1, 8]$, predicted action horizon 32, and 10 denoising steps. Five warmups and ten measurements are made in each of two model-loading orders (baseline then Split-S and the reverse), giving 20 measurements per model per timing mode. CUDA is synchronized before and after each invocation; loading is excluded.

Table 10: Same-GPU inference microbenchmark with synthetic observations. Latency is the mean of 20 measurements in milliseconds; measurements exclude simulator execution.

Model	End-to-end	Cached text context	Peak memory (GiB)
Monolithic	570.95	534.43	23.38
Split-S	1057.81	1032.81	23.43
Ratio	$1.853\times$	$1.933\times$	$1.002\times$

End-to-end timing includes text encoding, image VAE, video KV prefill, and action denoising; the second mode caches the text embedding outside the timer. The unfused implementation evaluates the two narrow experts sequentially at every denoising step. Both loading orders show increased end-to-end latency despite nearly identical parameter counts and peak memory. This benchmark characterizes model execution with synthetic inputs; end-to-end robot-control latency also depends on observation processing, simulator or hardware execution, and replanning.

D DA+Mem Robustness and Directional Sensitivity

We substitute the IF expert with a Depth Anything (DA) expert [21] and pair it with Mem [20], combining fields at the velocity level with the same S-curve family. We extend DA+Mem training to 40k steps to induce mild overfitting, stress-testing whether phase-wise complementarity and directional sensitivity persist when calibration is degraded. Table 11 reports mirrored schedules $(\beta, 1-\beta)$ relative to DA+Mem mean aggregation (96.35 Avg.).

Table 11: Directional sensitivity analysis (DA+Mem). We compare symmetric (time-reversed) S-curve schedules, where β and $1-\beta$ form a denoising-time reflection pair under mirrored schedule strength. Mean aggregation has 96.35 Avg.; ΔMean reports each schedule's gain over this static reference.

Forward	**Reverse**	**Fwd ΔMean**	**Rev ΔMean**
$\beta = 0.1$	$\beta = 0.9$	+0.35	+0.35
$\beta = 0.2$	$\beta = 0.8$	**+0.65**	+0.00
$\beta = 0.3$	$\beta = 0.7$	+0.60	+0.00
$\beta = 0.4$	$\beta = 0.6$	+0.35	-0.10

The pattern is **weaker but present** compared to IF+Mem (Appendix E): e.g. $(\beta,1-\beta) = (0.2, 0.8)$ and $(0.3, 0.7)$ favor forward schedules (Fwd ΔMean +0.65/+0.60 pp) while reversed sides collapse to +0.00 pp, whereas $(0.1, 0.9)$ is symmetric (+0.35 pp both sides). Forward sweeps stay positive (+0.35 to +0.65 pp), while reversed settings range from -0.10 to +0.35 pp relative to the mean baseline. Directional sensitivity therefore depends on schedule strength. **Additional** matched controls at $\beta = 0.2$ (DA+Ori, Ori+Mem, Ori+Ori) and preliminary learned-router experiments are reported in Appendix I.

E Additional IF+Mem Directional Results and Diagnostics

This appendix reports the IF+Mem mirrored-schedule success rates and router-free projection diagnostics. The two examples in Figure 4 show a shift from IF to Mem alignment consistent with the DA+Mem diagnostic in Figure 2.

E.1 Mirrored-parameter sensitivity (IF+Mem)

Denoising-time heterogeneity implies a **directionality** effect: even when two schedules have the same mixing strength, swapping the noise-level order of emphasis can change performance. We compare time-reversed S-curve schedules as symmetric pairs $(\beta, 1-\beta)$ relative to IF+Mem mean aggregation (97.00 Avg.).

Table 12: Directional sensitivity analysis (IF+Mem). Symmetric (time-reversed) S-curve schedules; β and $1-\beta$ form a denoising-time reflection pair under mirrored schedule strength. Mean aggregation has 97.00 Avg.; ΔMean reports each schedule's gain over this static reference.

Forward	Reverse	Fwd ΔMean	Rev ΔMean
$\beta = 0.1$	$\beta = 0.9$	+0.15	-0.05
$\beta = 0.2$	$\beta = 0.8$	+0.60	-0.20
$\beta = 0.35$	$\beta = 0.65$	+0.15	-0.30

Forward schedules are consistently above the static mean (up to +0.60 pp), whereas reversed schedules can fall below mean aggregation by as much as -0.30 pp. Paired comparisons (β vs. $1-\beta$) show a consistent ordering effect: expert aggregation is not time-symmetric under mirrored weights.

E.2 Router-free projection diagnostic (IF+Mem)

Since u^* is not observable at test time, we use a router-free proxy that measures which expert is more aligned with a shared reference direction. For each evaluation trajectory, let $\{t_k\}_{k=1}^{T}$ be the solver time grid with step sizes $s_k > 0$. Let u be the unit vector in the direction of the expected mixed displacement under a forward reference schedule, and define unweighted projections

$$g_{\mathrm{IF}}(t_k) = \langle s_k v_{\mathrm{IF}}(x_{t_k}, t_k, c), u \rangle, \qquad g_{\mathrm{Mem}}(t_k) = \langle s_k v_{\mathrm{Mem}}(x_{t_k}, t_k, c), u \rangle. \tag{12}$$

After trajectory-wise normalization (see Eq. (31)–(32) in Appendix H for the full construction), we report the stage-advantage $\Delta g_0(t) = \tilde{g}_{\mathrm{IF}}(t) - \tilde{g}_{\mathrm{Mem}}(t)$ averaged over episodes and noise draws.

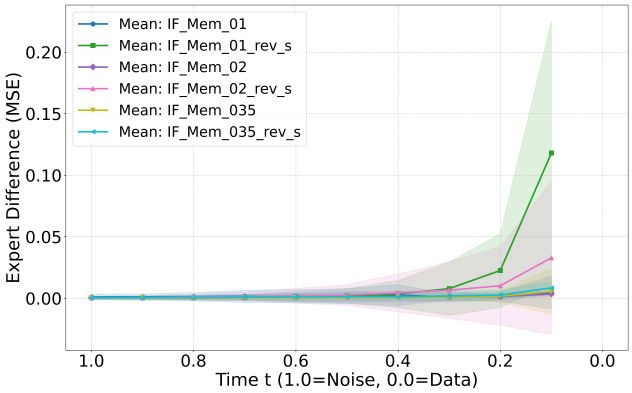


Figure 3: Squared differences between IF and Mem velocity fields under forward and reversed schedules. This diagnostic measures inter-expert disagreement; signed alignment with the shared reference direction is shown in Figure 4.

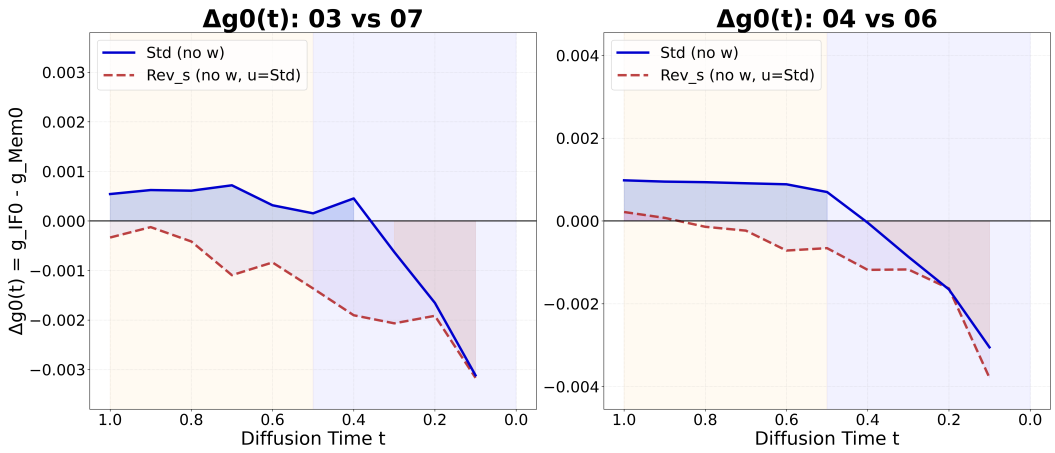


Figure 4: IF+Mem projection diagnostics for two mirrored S-curve pairs: $\beta = 0.3/0.7$ (left) and $0.4/0.6$ (right). Positive $\Delta g_0(t)$ indicates stronger IF alignment with the shared reference direction; negative values indicate stronger Mem alignment. Forward-S shifts from IF to Mem alignment as noise decreases, while Reverse-S favors Mem earlier.

Figure 4 plots $\Delta g_0(t)$ from $t = 1$ (high noise) to $t = 0$ (low noise). In both displayed pairs, the forward curve is positive at high noise and becomes negative at lower noise. The reversed curve favors Mem earlier, and the curves come closer at low noise. The timing and magnitude of this shift vary with schedule strength.

F Additional Low-Dimensional Diffusion-Policy Controls

To test whether the effect of denoising-time scheduling is tied to our main VLA implementation, we also ran controlled low-dimensional experiments in the `diffusion_policy` Robomimic benchmark. These experiments are *not* intended as an additional full-VLA benchmark: the inputs are low-dimensional states rather than image–language observations. Instead, they serve as action-generation controls in a different diffusion-policy training stack.

We evaluate `transport_lowdim_abs` and `tool_hang_lowdim_abs` under the Robomimic PH `low_dim_abs` setting with three seeds (24, 42, 43). All variants use the same policy stack and train/evaluation recipe. The `single` baseline uses one denoising expert. The `dual` variants share the same two-expert architecture; `dual_avg` uses constant mixing $w(t) = 0.5$, while `dual_scurve_035_065` and `dual_scurve_020_080` use mild and stronger forward S-curve schedules, respectively (the suffixes indicate the end weights, e.g., 0.35/0.65 vs. 0.20/0.80). Thus, among the dual variants, the only intervention is the prescribed denoising-time mixing schedule.

On `transport_lowdim_abs`, the stronger S-curve obtains the best 3-seed mean (0.900 ± 0.053), compared with 0.873 ± 0.012 for constant dual averaging and 0.870 ± 0.026 for the single baseline. On `tool_hang_lowdim_abs`, both S-curve variants reach a mean of 0.700, slightly above the single and constant-dual baselines (both 0.687), but the per-seed ordering is mixed.

Table 13: Low-dimensional `diffusion_policy` controls on Robomimic PH `low_dim_abs` tasks. All dual variants use the same architecture, training recipe, and evaluation protocol; only the denoising-time mixing schedule differs among `dual_avg` and the two S-curve schedules. Results report three seeds (24/42/43) and mean $\pm$ sample standard deviation.

Task	Method	Seeds 24/42/43	Mean $\pm$ std
`transport_lowdim_abs`	`single`	0.90 / 0.85 / 0.86	0.870 ± 0.026
`transport_lowdim_abs`	`dual_avg`	0.88 / 0.86 / 0.88	0.873 ± 0.012
`transport_lowdim_abs`	`dual_scurve_035_065`	0.94 / 0.87 / 0.86	0.890 ± 0.044
`transport_lowdim_abs`	`dual_scurve_020_080`	0.96 / 0.88 / 0.86	$\mathbf{0.900 \pm 0.053}$
`tool_hang_lowdim_abs`	`single`	0.73 / 0.66 / 0.67	0.687 ± 0.038
`tool_hang_lowdim_abs`	`dual_avg`	0.72 / 0.68 / 0.66	0.687 ± 0.031
`tool_hang_lowdim_abs`	`dual_scurve_035_065`	0.73 / 0.72 / 0.65	$\mathbf{0.700 \pm 0.044}$
`tool_hang_lowdim_abs`	`dual_scurve_020_080`	0.72 / 0.74 / 0.64	$\mathbf{0.700 \pm 0.053}$

G Broader Impacts

This work studies *mechanistic* properties of denoising-based vision–language–action (VLA) policies and proposes a controlled probe (fixed time-scheduled aggregation) to test whether expert usefulness varies across denoising levels. The primary expected positive impact is to improve scientific understanding of how generative policies behave across denoising phases, which may inform the design of more reliable and interpretable robot policies, and may reduce trial-and-error when selecting architectures or timestep-dependent components.

Potential negative impacts arise if improved action-generation methods are used in safety-critical or security-sensitive robotic settings without sufficient verification. More capable manipulation policies could be misused for harmful physical tasks, or could fail in unpredictable ways under distribution shift (e.g., new objects, viewpoints, or instructions), causing property damage or injury. Our experiments are confined to standard benchmarks and do not constitute a safety validation for deployment; any real-world use would require additional safety measures such as constrained action spaces, monitoring, fallback controllers, and extensive out-of-distribution testing.

H Full Mechanistic Derivations and Proofs

This appendix contains the schedule-weighted objective note, the complete squared-error analysis, Pareto background, proofs, and the formal scalarized diagnostic referenced from Sec. 4 and Sec. 5.5.

H.1 Background: per-timestep objectives and gradient conflict

We adopt the flow-matching view (the derivation applies similarly to diffusion training under standard reparameterizations). Let $x_t \in \mathbb{R}^d$ denote the latent/action state at diffusion time $t \in [0,1]$ and c the conditioning signal. The target velocity field is

$$u^*(x,t,c) = \mathbb{E}[\dot{x}_t \mid x_t = x, c], \tag{13}$$

and we train a parameterized field $v_\theta(x,t,c)$ by minimizing

$$\mathcal{L}(\theta) = \mathbb{E}_{t\sim p(t)}\Big[a(t)\mathcal{L}_t(\theta)\Big], \qquad \mathcal{L}_t(\theta) = \mathbb{E}_{x_t,c}\Big[\|v_\theta(x_t,t,c) - u^*(x_t,t,c)\|_2^2\Big] \tag{14}$$

where $a(t) \geq 0$ is an optional timestep weight (e.g., Min-SNR-type weighting).

A key observation behind Min-SNR is that the per-timestep tasks $\{\mathcal{L}_t\}_{t\in[0,1]}$ can have conflicting gradients. Define $g_t(\theta) = \nabla_\theta \mathcal{L}_t(\theta)$. When $\langle g_{t_1}, g_{t_2}\rangle < 0$, progress on one timestep can hurt another. From a multi-objective optimization standpoint, one seeks a Pareto-improving direction.

Definition H.1 (Pareto stationarity (continuous-time index))**.** *Let the vector objective be* $\mathbf{L}(\theta) = \{\mathcal{L}_t(\theta)\}_{t\in[0,1]}$. *We say* θ *is Pareto stationary if there exists a nonnegative weighting density* $\lambda(t) \geq 0$ *with* $\int_0^1 \lambda(t)dt = 1$ *such that*

$$\int_0^1 \lambda(t) g_t(\theta) dt = 0. \tag{15}$$

Equivalently, 0 lies in the convex hull of $\{g_t(\theta)\}$ in the limit of discretizing t. Min-SNR can be interpreted as a practical choice of $a(t)$ that reduces gradient domination and conflict across t. We study whether a structured two-expert mixture can better match phase-dependent denoising requirements.

H.2 Time-scheduled mixture of two expert fields

We parameterize the velocity field as a convex mixture of two full experts:

$$v(x_t, t, c) = w(t)v_1(x_t, t, c) + (1 - w(t))v_2(x_t, t, c), \tag{16}$$

Here v_1 and v_2 may be anchored experts with different inductive biases (e.g., IF/Mem or DA/Mem), or two copied Ori branches with the same architecture and initialization. We consider three schedules: aligned $w(t)$ (high at $t \to 1$ and low at $t \to 0$), constant $w(t) \equiv \frac{1}{2}$, and reversed $w_{\text{rev}}(t) = w(1 - t)$.

H.3 Schedule-weighted expert objectives

The schedule is fixed during both training and inference. For a training sample at time t, write the residual as

$$r_t = w(t)v_{\theta_1}(x_t, t, c) + (1 - w(t))v_{\theta_2}(x_t, t, c) - u_t. \tag{17}$$

The gradients of the squared flow-matching loss with respect to the two expert branches are

$$\nabla_{\theta_1}\mathcal{L} = \mathbb{E}_t\left[2w(t)J_{\theta_1}(t)^\top r_t\right], \qquad \nabla_{\theta_2}\mathcal{L} = \mathbb{E}_t\left[2(1 - w(t))J_{\theta_2}(t)^\top r_t\right], \tag{18}$$

where $J_{\theta_k}(t)$ is the Jacobian of expert k's velocity prediction with respect to its parameters. Thus expert 1 and expert 2 are trained under effective timestep weights proportional to $p(t)w(t)$ and $p(t)(1 - w(t))$, respectively.

This observation is not a convergence theorem. It only states that, when per-timestep objectives or gradients are heterogeneous, a non-constant schedule supplies different effective denoising-time objectives to the two branches. For IF/Mem and DA/Mem, the expert families also differ through their inductive biases. One can view this by defining a family-wise approximation profile

$$A_k(t) = \inf_{v \in \mathcal{F}_k} \mathbb{E}_{x_t, c}\|v(x_t, t, c) - u^*(x_t, t, c)\|_2^2, \tag{19}$$

where $\mathcal{F}_k$ represents the function class induced by expert k's conditioning and architecture. IF, DA, and Mem can therefore have different approximation profiles across t. For Ori+Ori, the two branches share the same family and initialization, but their schedule-weighted objectives can still drive functionally different solutions when the denoising-time tasks are not homogeneous. This is the theoretical role of the isomorphic-expert control: it tests whether schedule sensitivity persists without manually assigned expert semantics.

H.4 A closed-form optimal schedule per timestep (geometric projection)

For a fixed timestep t, define the *population* squared error of the mixture field:

$$\mathcal{R}_t(w) = \mathbb{E}_{x_t, c}\Big[\|wv_1(x_t, t, c) + (1 - w)v_2(x_t, t, c) - u^*(x_t, t, c)\|_2^2\Big] \tag{20}$$

where here $w \in [0, 1]$ is a scalar.

To simplify notation at fixed t, denote random vectors

$$a = v_1(x_t, t, c), \quad b = v_2(x_t, t, c), \quad u = u^*(x_t, t, c),$$

and define the second moments

$$A(t) = \mathbb{E}\|a - u\|_2^2, \quad B(t) = \mathbb{E}\|b - u\|_2^2, \quad C(t) = \mathbb{E}\|a - b\|_2^2. \tag{21}$$

Proposition H.2 (Pointwise optimal mixture weight)**.** *Assume $C(t) > 0$. The minimizer of $\mathcal{R}_t(w)$ over $w \in \mathbb{R}$ is*

$$w_t^* = \frac{1}{2} + \frac{B(t) - A(t)}{2C(t)}. \tag{22}$$

The constrained minimizer over $w \in [0, 1]$ is the clipped version

$$w_t^{*,[0,1]} = \Pi_{[0,1]}\left(\frac{1}{2} + \frac{B(t) - A(t)}{2C(t)}\right), \tag{23}$$

where $\Pi_{[0,1]}(x) = \min\{1, \max\{0, x\}\}$.

Proof. Write the mixture error as

$$wa + (1-w)b - u = (b-u) + w(a-b).$$

Thus $\mathcal{R}_t(w) = \mathbb{E}\|e + wd\|_2^2$ with $e = b - u$ and $d = a - b$. Differentiate:

$$\frac{d}{dw}\mathcal{R}_t(w) = 2\,\mathbb{E}\langle e + wd, d\rangle = 2\big(\mathbb{E}\langle e, d\rangle + w\,\mathbb{E}\|d\|_2^2\big).$$

Set derivative to zero:

$$w_t^* = -\frac{\mathbb{E}\langle e, d\rangle}{\mathbb{E}\|d\|_2^2}.$$

Compute $\mathbb{E}\langle e, d\rangle = \mathbb{E}\langle b-u, a-b\rangle = \mathbb{E}\langle b-u, a-u\rangle - \mathbb{E}\|b-u\|_2^2$. Let $E(t) = \mathbb{E}\langle a-u, b-u\rangle$. Then $\mathbb{E}\langle e, d\rangle = E(t) - B(t)$. Also note $C(t) = \mathbb{E}\|a-b\|_2^2 = A(t) + B(t) - 2E(t) \implies E(t) = \frac{A(t)+B(t)-C(t)}{2}$. Substitute:

$$w_t^* = \frac{B(t) - E(t)}{C(t)} = \frac{B(t) - \frac{A(t)+B(t)-C(t)}{2}}{C(t)} = \frac{1}{2} + \frac{B(t) - A(t)}{2C(t)}$$

Finally, projecting onto $[0,1]$ yields (23). □

Interpretation. Equation (22) shows that the optimal schedule is governed by the *relative per-timestep approximation errors* $A(t)$ vs $B(t)$. If $A(t) < B(t)$, then $w_t^* > \frac{1}{2}$ (favor expert 1); if $A(t) > B(t)$, then $w_t^* < \frac{1}{2}$ (favor expert 2). The term $C(t)$ is the curvature of the one-dimensional mixture risk. When the experts are nearly identical, changing w has little effect on the mixture output, so the precise value of the optimal weight is less identifiable and is clipped to $[0,1]$ in deployment.

H.5 Phase-wise heterogeneity implies a non-constant optimal schedule

Assumption H.3 (Phase-wise approximation gap / information heterogeneity)**.** *There exists a boundary $t^* \in (0,1)$ and a margin $\delta > 0$ such that*

$$B(t) - A(t) \geq \delta \quad \textit{for a.e. } t \in (t^*, 1], \qquad B(t) - A(t) \leq -\delta \quad \textit{for a.e. } t \in [0, t^*). \tag{24}$$

Equivalently, expert 1 has strictly smaller error at high noise, while expert 2 has strictly smaller error at low noise. In our anchored experiments, this corresponds to the perception-oriented expert dominating earlier denoising and the memory-oriented expert dominating later refinement.

Under Assumption H.3, Proposition H.2 implies that the unconstrained optimum satisfies $w_t^* > \frac{1}{2}$ for a.e. $t \in (t^*, 1]$ and $w_t^* < \frac{1}{2}$ for a.e. $t \in [0, t^*)$ (when $C(t) > 0$), i.e., an *aligned, non-constant* schedule is optimal away from the phase boundary.

Corollary H.4 (Pointwise optimum improves over constant mixing)**.** *If $C(t) > 0$ and $B(t) - A(t)$ is nonzero on a set of positive p-measure, then the pointwise optimal schedule $w_t^{*,[0,1]}$ has no larger integrated risk than constant mixing $w(t) \equiv \frac{1}{2}$, and the inequality is strict whenever the clipped optimum differs from $\frac{1}{2}$ on a set of positive p-measure. For a chosen parametric aligned schedule w, a sufficient condition for improving over constant mixing is*

$$\mathbb{E}_t\left[(w(t) - \tfrac{1}{2})\Delta(t)\right] > \mathbb{E}_t\left[C(t)(w(t) - \tfrac{1}{2})^2\right], \qquad \Delta(t) = B(t) - A(t). \tag{25}$$

This follows from the identity $\mathcal{R}_t(w) - \mathcal{R}_t(\frac{1}{2}) = C(t)(w - \frac{1}{2})^2 - (w - \frac{1}{2})\Delta(t)$.

Scope note. Corollary H.4 is an existence/sufficiency statement in squared-error denoising-field risk space. In experiments, success rate is a downstream behavioral readout.

H.6 Aligned schedule vs reversed schedule: a formal separation

We compare the population risks of aligned vs reversed schedules at the *integrated* level:

$$\mathcal{R}(w) = \mathbb{E}_{t\sim p(t)}\big[\mathcal{R}_t(w(t))\big]. \tag{26}$$

Definition H.5 (Aligned / reversed schedules)**.** *A schedule $w : [0,1] \to [0,1]$ is aligned w.r.t. t^* if $w(t) \geq \frac{1}{2}$ for $t \in [t^*, 1]$ and $w(t) \leq \frac{1}{2}$ for $t \in [0, t^*]$. Its reversed counterpart is $w_{rev}(t) = w(1-t)$. The S-shaped schedules used in our mirrored controls additionally satisfy center symmetry, $w(t) + w(1-t) = 1$.*

Theorem H.6 (Aligned schedules dominate reversed schedules under phase-wise gap)**.** *Assume Assumption H.3 holds with $t^* = \frac{1}{2}$. Let w be aligned, non-constant, and center-symmetric, i.e., $w(t)+w(1-t) = 1$, and let $w_{rev}(t) = w(1-t)$. Suppose additionally that the set $\{t : w(t) \neq w_{rev}(t)\}$ has positive measure under $p(t)$. Then*

$$\mathcal{R}(w) < \mathcal{R}(w_{rev}). \tag{27}$$

Moreover, the separation is

$$\mathcal{R}(w_{rev}) - \mathcal{R}(w) = \mathbb{E}_t\Big[\big(w(t) - w_{rev}(t)\big)\Delta(t)\Big] > 0, \qquad \Delta(t) = B(t) - A(t). \tag{28}$$

Proof. Fix t and abbreviate $\Delta(t) = B(t) - A(t)$. From (20), a direct expansion yields, for any scalars w, w',

$$\mathcal{R}_t(w) - \mathcal{R}_t(w') = C(t)(w - w')(w + w' - 1) - (w - w')\Delta(t). \tag{29}$$

Now set $w' = w_{\text{rev}}(t)$. Center symmetry gives $w(t) + w_{\text{rev}}(t) = 1$, so the curvature term cancels:

$$\mathcal{R}(w) - \mathcal{R}(w_{\text{rev}}) = -\mathbb{E}_t\left[(w(t) - w_{\text{rev}}(t))\Delta(t)\right].$$

Rearranging gives (28).

It remains to show $(w(t) - w_{\text{rev}}(t))\Delta(t) \geq 0$ and positive on a set with nonzero p-measure. For a.e. $t > \frac{1}{2}$, alignment implies $w(t) \geq \frac{1}{2}$ while $w_{\text{rev}}(t) = w(1-t) \leq \frac{1}{2}$, hence $w(t) - w_{\text{rev}}(t) \geq 0$. Assumption H.3 gives $\Delta(t) = B(t) - A(t) \geq \delta > 0$ for a.e. $t > \frac{1}{2}$, so the product is ≥ 0. Similarly, for a.e. $t < \frac{1}{2}$ we have $w(t) - w_{\text{rev}}(t) \leq 0$ and $\Delta(t) \leq -\delta < 0$, again giving $(w - w_{\text{rev}})\Delta \geq 0$. By the positive-$p$-measure condition on $\{t : w(t) \neq w_{\text{rev}}(t)\}$, the product is strictly positive in expectation. Therefore $\mathcal{R}(w) < \mathcal{R}(w_{\text{rev}})$, proving (27). □

Implication. Theorem H.6 formalizes a mechanistic control: when expert advantage truly switches across phases, reversing a center-symmetric schedule systematically assigns more weight to the worse expert at each phase, yielding a provable degradation.

Connection to end-to-end fine-tuning. In our experiments, the schedule is fixed during training as well as inference. Different schedules therefore allocate gradients differently across denoising time and can allow duplicated experts to functionally diverge even from the same pretrained initialization; the Ori+Ori control is an empirical readout of schedule sensitivity under this same-family setting.

H.7 A scalarized stage-advantage statistic and why the visualization detects phase heterogeneity

Let $s(t) > 0$ denote the effective solver step size (or the logged Δt proxy). Define per-step displacements (unweighted by the router):

$$\Delta x_{\text{IF}}(t) = s(t)v_{\text{IF}}(x_t, t, c), \qquad \Delta x_{\text{Mem}}(t) = s(t)v_{\text{Mem}}(x_t, t, c). \tag{30}$$

Let u be the normalized (aligned/Std) mixed displacement direction. With

$$m = \mathbb{E}\left[\sum_t s(t)\big(w(t)v_{\text{IF}} + (1 - w(t))v_{\text{Mem}}\big)\right], \qquad u = \frac{m}{\|m\|_2} \quad \text{when } \|m\|_2 > 0. \tag{31}$$

In numerical plots we use the stabilized denominator $\|m\|_2 + \epsilon$; the diagnostic depends on the shared direction, not on the exact normalization constant. We then scalarize the displacements by projection:

$$g_{\text{IF}}(t) = \langle \Delta x_{\text{IF}}(t), u\rangle, \qquad g_{\text{Mem}}(t) = \langle \Delta x_{\text{Mem}}(t), u\rangle, \qquad \Delta g_0(t) = g_{\text{IF}}(t) - g_{\text{Mem}}(t). \tag{32}$$

Assumption H.7 (Dominant-direction scalarization)**.** *Along the aligned trajectory, the projection onto u captures the dominant denoising progress, i.e., there exists $\kappa > 0$ such that for most t,*

$$\mathbb{E}\big[\langle s(t)u^*(x_t, t, c), u\rangle\big] \geq \kappa, \tag{33}$$

and the relative ordering of experts w.r.t. approximation error correlates with their projected progress along u.

Diagnostic implication. Under Assumptions H.3 and H.7, the projection statistic is expected to show $\Delta g_0(t) > 0$ over high-noise phases where IF has the smaller approximation error, whereas a reversed schedule can collapse this structure due to phase mismatch. We use this as an explanatory diagnostic rather than as a formal estimator of the unobserved target field u^*.

Connection to Min-SNR. Min-SNR reweights timesteps via $a(t)$ in (14) to alleviate multi-task gradient domination and conflict. Structural time-dependent aggregation (two experts + $w(t)$) yields a provable advantage when per-timestep approximation gaps change sign across phases (Assumption H.3). These approaches are complementary: $a(t)$ addresses optimization dynamics, while $w(t)$ addresses representational mismatch across phases.

I Additional DA+Mem Controls and Learned-Router Attempts

I.1 Matched controls at $\beta = 0.2$

In addition to Table 11 (Appendix D), we evaluate three controls at $\beta = 0.2$: DA+Ori, Ori+Mem, and Ori+Ori. Relative to DA+Mem mean aggregation, Ori+Ori gives no gain (+0.00 pp), while DA+Ori and Ori+Mem give only +0.25 and +0.30 pp. The forward DA+Mem schedule reaches +0.65 pp, outperforming these controls and indicating that the gain depends on the paired experts being aligned with the schedule.

I.2 Visual correspondence (DA+Mem vs. IF+Mem)

The DA+Mem projections in Figure 2 and the two IF+Mem examples in Figure 4 show a similar shift from perception-oriented to memory-oriented alignment under Forward-S. Reverse-S favors Mem earlier in denoising. The timing and magnitude of the shift vary with the schedule, connecting the aggregation rule to the relative contributions of the two expert fields.

I.3 Learning to route: preliminary attempts

We explored learning the time-dependent aggregation weight with DA+Mem as a testbed, using lightweight routers that take diffusion timestep t as input.

Free-form router. The router predicts $w(t) \in [0, 1]$ via sigmoid with no parametric structure beyond network smoothness. Optimization often collapsed early to near-constant gating (one expert dominates most timesteps), consistent with ill-conditioned mixture training without regularization or identifiability constraints.

β-router (structured schedule). We retain the S-curve and only learn β in $w(t; \beta)$. Initializing $\beta \in \{0.2, 0.4, 0.6, 0.8\}$, optimization consistently converged near ≈ 0.46, largely independent of initialization—suggesting that recovering the most effective routing pattern from data remains non-trivial and may bias toward a near-symmetric compromise under our objective.

These preliminary failures apply to our router implementations and budgets; they do not establish that learned routing in general is harder or less effective than fixed scheduling.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction describe controlled schedule interventions, distinguish the joint training-and-inference setting, and restrict claims to the studied settings. The theory conditions and experimental limitations are stated explicitly.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 7 discusses simulation-to-deployment generality, latency, limited seeds, the scope of empirical comparisons, and theoretical assumptions.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Section 4 states the risk interpretation; Appendix H gives assumptions and proofs. The pointwise optimum depends on the clipped advantage-to-disagreement ratio.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [No]

   Justification: We specify the schedules and evaluation procedures in the main text and Appendix C, but do not provide every anchor-extraction parameter and run configuration.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We use public benchmarks; a complete public release of our training and evaluation implementation is not included.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [No]

Justification: Appendix C gives the Fast-WAM configuration and evaluation protocols; not all hyperparameters for the anchored experiments are reported.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The full-expert comparisons include multi-run mean and standard deviation; Fast-WAM uses sample SD over three seeds. The fixed-pair risk diagnostic reports sample-cluster bootstrap 95% intervals (500 resamples). The supplementary cross-evaluation and behavioral reversal are descriptive and have no claimed paired significance.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Appendix C.4 reports representative training GPU-hours and a same-GPU inference microbenchmark, but does not enumerate the compute budget of every training run and preliminary experiment.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: To the best of our knowledge, the study conforms to the NeurIPS Code of Ethics; it uses simulated robotics benchmarks and explicitly discusses reporting and deployment limits.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: Appendix G discusses possible scientific benefits, physical misuse and distribution-shift risks, and the latency/energy implications of expert aggregation.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: This manuscript does not release a high-misuse-risk model or a newly scraped dataset; the experiments use existing robotics benchmarks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [No]

Justification: Primary model sources are cited, but complete asset versions, licenses, and terms of use are not enumerated in this manuscript.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new dataset or public model/code asset release is included with this manuscript.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The research does not involve crowdsourcing or human-subject experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The research does not involve human subjects and therefore does not require IRB or equivalent review.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The study uses existing VLA backbones and does not introduce an LLM as a novel component of the research methodology.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.