BEYOND NOISE: REGULATING HOW MUCH OF THE PRIOR SURVIVES IN FLOW-MATCHING ROBOT POLICIES

Anonymous authors
Paper under double-blind review

Code: `https://anonymous.4open.science/r/Beyond-Noise-56D4/`

ABSTRACT

Most flow-matching robot policies generate actions from Gaussian noise, ignoring the temporal continuity available in robot interaction. Recent methods warm-start the flow with informative sources such as previous actions or visual forecasts, implicitly assuming that a better source produces a better action. We show that this assumption is incomplete: informative sources help, but much of their information is attenuated by the flow before it reaches the executed action, so what must be controlled is how much of the prior survives, not only which prior enters. We therefore propose Beyond Noise. It enters the flow at a state that couples the prior's weight, the injected noise and the remaining path through a single entry time q. Beyond Noise selects a default entry on validation data and applies the regulator's per-call refinement where validated, dynamically adjusting q according to the current state; the regulator learns from counterfactual closed-loop returns. A patch-token memory lets the policy use where objects are once the prior is kept. On four LIBERO suites, under a protocol that separates every choice from the held-out test states, Beyond Noise reaches 81.0% average success with a single function evaluation, above matched Gaussian, visual-prior and external baselines, with the largest gains on long-horizon tasks. Exploiting a better prior requires deciding how much of it survives.

1 INTRODUCTION

Generative policies based on diffusion and flow matching have become a standard approach to visuomotor control (Chi et al., 2023; Ze et al., 2024; Black et al., 2025a). Building on score-based generative models (Ho et al., 2020; Song et al., 2021) and flow matching (Lipman et al., 2023), they represent multimodal action distributions and emit temporally extended action chunks (Shafiullah et al., 2022; Zhao et al., 2023), each produced by transporting a sample from a *source* distribution to the action manifold. The source is thus an explicit—if rarely examined—part of the policy.

Almost all such policies draw that sample from an observation-independent Gaussian (Chi et al., 2023; Black et al., 2025a). The default ignores what the controller already knows: the previous chunk and recent frames say much about the next one. Recent work therefore builds the source from a prior policy (Chen et al., 2024), from visual representations (Gao et al., 2026), from recent actions (Jia et al., 2026; Kang et al., 2026) or earlier predictions (Duan et al., 2025; Chen et al., 2025; Li et al., 2026b), or learns a state-adaptive Gaussian (Dai et al., 2026). A matched study—one architecture, one training recipe and one evaluator on four LIBERO suites—confirms the premise: warm-starting a one-step flow raises success by 15.7–16.9 percentage points (pp) over a Gaussian start. Within the budgets we test, more integration steps do not close this gap.

The same study yields a result we did not anticipate. If an informative source helps, a better-informed one should help more, but it does not. A visual source that forecasts the upcoming chunk from recent frames and a temporal source that reuses the previous chunk are statistically indistinguishable ($+1.2\,\mathrm{pp}$; 95% CI $[-2.7, +5.2]$), and a paired intervention that adds an explicit previous chunk to the visual source changes success by $+0.08\,\mathrm{pp}$. More information at the source does not by itself buy better actions.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

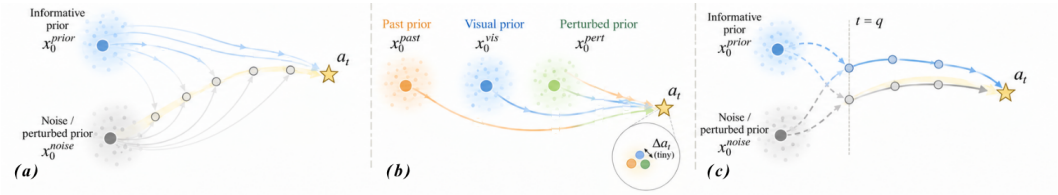

Figure 1: Why Beyond Noise. (a) An informative prior (blue) starts the flow closer to the action than Gaussian noise (gray), so warm starts help. (b) Yet the flow absorbs most of the difference between sources: past, visual and perturbed priors lead to nearly the same action (inset). (c) Beyond Noise controls how much of the prior survives by entering the flow at $t = q$ (flow time; s in the text) from a mixture of prior and noise and integrating only the remaining path. Schematic, not to scale.

Part of the reason lies downstream of the source. When we displace a policy's own source while holding the observation and noise fixed, only 6% (Gaussian) to 13–20% (warm starts) of the displacement reaches the executed action (Section 3.3); in a policy trained on both sources, exchanging the visual and temporal sources moves the action by only 20–27% of the distance between them. Much of what distinguishes one informative source from another is absorbed before execution.

If refinement limits how much of the source survives, then how much survives should be a decision. Beyond Noise makes it one (Figure 1c). It enters the flow at an intermediate time q from a mixture $qB_t + (1 - q)\varepsilon$ of a predicted action anchor B_t and noise, and integrates only the remaining path $[q, 1]$: one scalar sets the anchor weight, the noise scale and the remaining path. Beyond Noise selects a default entry on validation data and applies the regulator's per-call refinement where validated, dynamically adjusting q according to the current state. The *adaptive regulator* learns by counterfactual closed-loop regulation: from shared situations, it compares what happens when the controller keeps more or less of its prior, and enters where the return is highest. Because a kept prior is useful only if the policy can see where the objects are, the flow also attends to a small grid of image patch tokens.

Our contributions are as follows.

- **A matched study of the source.** Informative sources improve one-step policies by 15.7–16.9 pp through cross-step consistency; the choice among them and additional history are inert, and most of a source perturbation is attenuated before it reaches the action (Section 3).
- **Beyond Noise.** A coupled entry, an adaptive regulator that sets the entry time from counterfactual closed-loop returns, and a patch-token memory initialized from the coupled policy (Section 4).
- **Evidence.** The coupled entry raises transmission about sevenfold. Under a held-out protocol, Beyond Noise averages 81.0% against 73.3% for a fixed visual prior with the same recipe and capacity. Adaptive regulation adds 4.0 pp over a fixed entry and recovers tasks that any single entry fails; on 500 freshly sampled LIBERO-10 initial states, its per-call refinement raises success from 70.0 to 74.6% (Section 5).

2 Preliminaries

Closed-loop chunked control. At environment step t the policy receives the current observation o_t and the interaction history h_t, predicts an action chunk $A_t = (a_t, \ldots, a_{t+H-1})$, executes its first H_a actions and replans; we call each replanning a *planning call*. The objective is the closed-loop return, which on LIBERO is the probability of task success.

Flow-matching policies. Flow matching learns a velocity field that transports a source distribution to the action distribution (Lipman et al., 2023). We write $s \in [0, 1]$ for flow time, reserving t for environment time, and c_t for the conditioning information (the current observation and proprioception). For a source sample $Z \sim p_0(\cdot \mid c_t)$ and a demonstrated chunk A_t, the linear interpolant

Table 1: What the source contributes (matched-source protocol). Success: 200 episodes per suite (Goal: mean of three training seeds; Pooled: all six suite–seed cells). Δ: pooled difference from the Gaussian start with a 95% task-cluster bootstrap interval. T_x, T_y: transmission of a source displacement to the executed action at the deployed single step; $d(1)$: separation remaining after 16 Euler steps, measured with a separate protocol.

	Success (%), one step						Attenuation		
Source	Object	Spatial	Goal	LIBERO-10	Pooled	Δ vs. Gaussian	T_x	T_y	$d(1)$
Gaussian	64.5	67.5	57.5	23.0	54.6	—	0.062	0.060	0.176
Temporal	90.0	63.0	78.3	33.5	70.3	$+15.7\,[+10.3, +21.2]$	0.125	0.134	0.316
Visual	85.5	73.0	78.3	35.5	71.5	$+16.9\,[+11.7, +22.2]$	0.195	0.166	0.344

$X_s = (1-s)Z + sA_t$ defines the objective

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{(c_t, A_t), Z, s}\left[\left\| v_\theta(X_s, s \mid c_t) - (A_t - Z)\right\|_2^2\right], \qquad s \sim \mathcal{U}[0,1]. \tag{1}$$

At inference, $\mathrm{d}X_s/\mathrm{d}s = v_\theta(X_s, s \mid c_t)$ is integrated from $X_0 = Z$ to $\widehat{A}_t = X_1$ with n Euler steps, that is, n function evaluations (NFE). A *cold start* draws $Z = \varepsilon \sim \mathcal{N}(0, I)$. A *warm start* centers the source on an action anchor B_t, $Z = B_t + 0.5\,\varepsilon$, and still integrates the full interval (Kang et al., 2026); we call this the *legacy contract*. We call the integrated interval $[s_0, 1]$ the *remaining path*.

3 What the source contributes

Setup. A comparison of sources is meaningful only if nothing else differs: batch size and learning rate alone can reverse its outcome. We therefore train three one-step policies that differ only in their source: a Gaussian *cold start*; a *temporal* warm start that reuses the previous predicted chunk (Kang et al., 2026); and a *visual* warm start whose anchor comes from AVIS, a learned two-frame predictor over DINOv2 features (Oquab et al., 2024). Both warm starts use the legacy contract, and all three share one 21.9 M transformer action head, one training recipe and one evaluator. We run receding-horizon control (prediction horizon $H = 16$, executed horizon $H_a = 8$) on LIBERO-Object, -Spatial, -Goal and LIBERO-10 (Liu et al., 2023a), 200 episodes per suite. Object, Spatial and LIBERO-10 contribute one suite–seed cell each and Goal three training seeds, for six cells and 1,200 episodes per method; we base conclusions on pooled 95% task-cluster bootstrap intervals, not on single cells.

3.1 An informative source helps, and more steps do not close the gap

An informative start pays off: pooled over 1,200 episodes per method, the temporal and visual warm starts beat the Gaussian start by $+15.7$ and $+16.9$ pp (Table 1), with large gains on Object, Goal and LIBERO-10; on Spatial, only the visual source helps.

More steps and more capacity do not close the gap. Extra integration steps do not let the Gaussian start catch up: over NFE $\in \{1, 4, 8, 16\}$ on four suites, both warm starts score best at one step, the Gaussian start gains at most $+2.2$ pp, and the warm-start margin persists at every budget ($+9.6$ to $+16.5$ pp). We report this as an observation under our recipe and budgets, not as a property of numerical integration. Capacity does not change the picture either: a $4\times$ larger action head moves the Goal mean by -1.6 pp, and the margin persists across three training configurations.

The source buys consistency, not accuracy. A warm start does not help by predicting better: offline, the Gaussian and temporal policies predict chunks equally well (error 0.1330 versus 0.1329 on Goal). It helps in closed loop, where it makes consecutive plans agree, cutting their disagreement by 27–30% and their jerk by 16–29%.

3.2 More source information does not necessarily help

The visual source forecasts the upcoming chunk from recent frames; the temporal source merely reuses a chunk planned eight steps earlier. Yet neither is reliably better: their difference changes sign across suites (-4.5 pp on Object, $+10.0$ on Spatial, $+2.0$ on LIBERO-10) and across Goal seeds ($+2.5$, -0.5, -2.0), and pools to $+1.2$ pp (95% CI $[-2.7, +5.2]$). The visual policy is even more accurate offline (error 0.128 versus 0.148), but no more successful in closed loop.

Since AVIS already conditions on past frames and action history, this does not show that history is useless; it asks what *additional* information buys. We test that directly: freezing the policy, visual predictor and backbone, we add an explicit previous predicted chunk to the AVIS-based source (Appendix B.5). Across 40 decision states with 32 independent paired seeds each, success moves from 66.17% to 66.25% ($+0.08$ pp; episode-cluster 95% CI $[-1.25, +1.41]$), and no state changes after Holm correction. Once one informative source is in place, more source information buys little.

3.3 Source perturbations are substantially attenuated

Why does extra information at the source buy so little? To see how much of a source reaches the action, we shift each policy's own deployment source by a small offset ΔX along an end-effector axis, hold the observation and noise fixed, and record two quantities (Table 1): the normalized separation $d(s) = \|X_s^{(a)} - X_s^{(b)}\|/\|\Delta X\|$ between the two resulting trajectories along a 16-step integration, and the transmission $T = \mathrm{RMS}(\Delta\widehat{A})/\mathrm{RMS}(\Delta X)$ to aligned executed-action coordinates at the deployed single step.

Only a small fraction of the source reaches the action. All three policies contract strongly: 66–82% of the separation disappears over 16 steps, and only 6–20% of a displacement reaches the executed action at the deployed step. The remaining separation is ordered Gaussian $<$ temporal $<$ visual and shrinks at every step, except the last for the temporal source, where it rises slightly.

The source is used, but only to seed the path. Attenuation does not mean the source is ignored: replacing a warm-start policy's source with Gaussian noise raises its error by a factor of 1.6 (temporal) or 2.9 (visual). Yet small perturbations of it are absorbed. Weakening the observation condition does not raise transmission, so we do not attribute the attenuation to observation conditioning.

From attenuation to the inert choice of source. To swap the two sources within the training distribution, we train one policy on both, drawn with equal probability and unlabeled. Swapping them moves the executed action by only 20% (Goal) and 27% (LIBERO-10) of the distance between them, and the policy succeeds about equally often with either (146 versus 149 and 84 versus 94 of 200 episodes). The absorption is generic, not specific to these sources: a displacement of the same size taken from another window is absorbed just as much. Attenuation is not the whole story: trained on both sources, the policy loses the visual-only policy's advantage on Goal (146 versus 165 of 200), so per-source training and the redundancy between sources (AVIS already encodes history) also play a role. Attenuation thus limits how much any source can shape the action; the amount that survives is the quantity to design.

4 Beyond Noise: regulating how much of the prior survives

Section 3 showed that an informative source beats Gaussian noise, but that the flow absorbs most of the difference between sources, so improving the source alone is a weak lever. Beyond Noise changes the question from *which* prior to supply to *how much* of it the flow keeps, and lets the policy set that amount for each situation (Figure 2). A single scalar q then sets both the mix of anchor and noise in the entry state and the time at which the flow starts (Section 4.1). The flow is trained on the entry states this coupling creates, so one checkpoint covers the whole range of q (Section 4.2). At deployment, q is set by counterfactual closed-loop regulation (Section 4.3). The policy relies on the anchor when it is trustworthy and moves toward noise when it is not. This adaptivity, rather than a better prior, is what raises success and robustness.

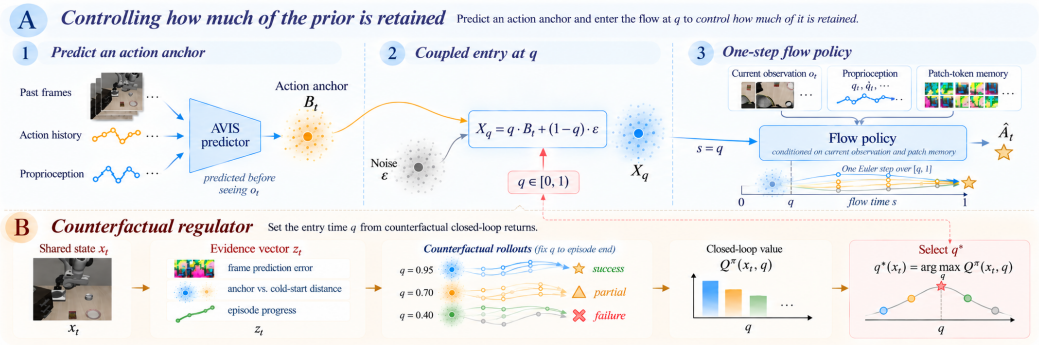

Figure 2: Overview of Beyond Noise. (A) Controlling how much of the prior is retained: (1) AVIS predicts an action anchor B_t from past frames, action history and proprioception, before seeing the current observation o_t; (2) the coupled entry state $X_q = q\,B_t + (1-q)\,\varepsilon$ ties the anchor's weight, the injected noise and the remaining path to one scalar q; (3) the one-step flow policy, conditioned on the current observation, proprioception and patch-token memory, enters at $s = q$ and integrates $[q, 1]$ in one step to the action chunk $\widehat{A}_t$. (B) The counterfactual regulator sets q from closed-loop returns: from a shared state x_t, summarized by an evidence vector z_t (frame prediction error, anchor versus cold-start distance, episode progress), the controller is continued with each candidate q fixed to the end of the episode; the outcomes give the closed-loop value $Q^\pi(x_t, q)$, and the regulator selects $q^\star(x_t) = \arg\max_q Q^\pi(x_t, q)$.

4.1 Coupling source proportion and flow entry time via q

At each planning call, let $B_t \in \mathbb{R}^{H\times d}$ be an *action anchor*: a prediction of the next H actions based on information available before the current observation. Our anchor is produced by AVIS, a frozen predictor that maps past frames, action history and proprioception to an action chunk. The anchor is a prediction rather than the demonstrated chunk A_t, which is available only during training. It enters the policy only through the initial state, while the flow remains conditioned on the current observation throughout generation.

We write the flow entry state as $\alpha B_t + \sigma\varepsilon$, introduced at flow time s_0. The conventional warm start uses $(\alpha, \sigma, s_0) = (1, 0.5, 0)$: a full-weight anchor with substantial noise, followed by refinement over the entire path. This gives the flow the full path over which to revise or overwrite the anchor. Beyond Noise instead ties all three quantities to a single entry parameter $q \in [0, 1)$:

$$X_q = qB_t + (1-q)\varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, I), \qquad s_0 = q, \tag{2}$$

and integrates $\mathrm{d}X_s/\mathrm{d}s = v_\theta(X_s, s \mid c_t)$ over the remaining path $[q, 1]$, yielding $(\alpha, \sigma, s_0) = (q, 1-q, q)$.

This coupling gives q a single operational meaning: how strongly the policy commits to the prior. At $q = 0$, the entry state is pure noise and the full path remains, recovering the cold start. As q increases, the anchor receives greater weight, the injected noise decreases and less of the path remains to revise the anchor. These effects act in the same direction: a larger q provides a stronger anchor and gives the flow less opportunity to overwrite it, thereby increasing the amount of prior information retained in the output. The entry parameter therefore interpolates between ignoring the prior and committing to it. Appendix C measures how much of the anchor survives at each value of q.

4.2 Training on coupled entry states

The coupling in Section 4.1 is useful only if a single flow can operate across the entry states induced by the full range of q. Because the anchor is a prediction, X_q need not lie on the interpolant between ε and A_t, and a flow trained only from Gaussian starts is not guaranteed to handle it. We therefore train the flow directly on the distribution of coupled entry states it will encounter at deployment. For

a sampled q, the reference path runs in a straight line from the entry state to the demonstrated chunk,

$$X_s^{(q)} = (1-u)\,X_q + u\,A_t, \qquad s = q + u(1-q), \quad u \in [0,1], \tag{3}$$

with constant velocity $(A_t - X_q)/(1-q)$; at $q = 0$ it reduces to standard flow matching. Because this velocity grows as $1/(1-q)$, we regress the displacement, scaling the prediction by the remaining path:

$$\mathcal{L}_{\text{path}}(\theta) = \mathbb{E}\Big[\big\|(1-q)\,v_\theta(X_s^{(q)}, s \mid c_t) - (A_t - X_q)\big\|_2^2\Big], \tag{4}$$

which has the same pointwise minimizer and keeps the target of order one at every q. Half of the training samples are drawn at the entry state ($u = 0$), the only point a one-step policy evaluates, and the rest along the path.

Which anchors the flow encounters at each q depends on how the regulator (Section 4.3) will use it: low entries are selected precisely when the anchor cannot be trusted. Stage 1 samples $q \sim \mathcal{U}(0, 0.95)$, covering both cold and informative entries. Stage 2 is a short continuation restricted to the high-entry band $q \in [0.85, 0.95]$, where most deployment situations offer a reliable anchor. It uses clean, current anchors and concentrates training on preserving and refining a trustworthy prior. Initialized from Stage 1, the resulting band *specialist* inherits its low-q behavior from training over the full range while specializing the high-q regime used most often at deployment.

Patch-token memory. Because a high entry keeps more of the anchor, correcting it where the scene has changed falls to the flow's conditioning. We therefore add pooled DINOv2 patch tokens from the two observation frames to the decoder's cross-attention memory, initialized from the coupled policy, so the flow can localize where the scene departs from the anchor (Appendix D).

4.3 Counterfactual closed-loop regulation of q

What remains is the value of q, and on this question imitation is silent: a demonstration records what the expert did, not what would have happened had the flow kept more or less of its prior. The right entry time is a property of closed-loop consequences, so we define it through them. Let x_t denote the situation of the controller at a planning call, and let

$$Q^\pi(x_t, q) = \Pr[\text{success} \mid x_t,\ \text{entry time } q \text{ from } t \text{ onward}] \tag{5}$$

be the closed-loop value of keeping that much of the prior. The ideal regulator chooses

$$q^\star(x_t) = \arg\max_q Q^\pi(x_t, q).$$

It keeps the prior where keeping it leads to success and releases it where it does not. The entry time is therefore defined by consequences, not by a hand-set rule on the anchor's reliability.

Estimating value by intervention. Q^π cannot be read from logged experience. An episode reveals the outcome of the entry time it used, never those of the alternatives, and observationally chosen entry times may be correlated with situation difficulty. We therefore estimate Q^π by counterfactual intervention. From a shared situation—the same task, initial state and branch call—the controller is continued in closed loop under each candidate entry time, which is held fixed to the end of the episode. Because the entry time is assigned independently of the situation, outcome differences between branches estimate the effect of the entry time without confounding, and aggregating them over situations yields an unbiased estimate of the advantage of one entry time over another.

Default entry (task level). The regulator answers the same question—which q maximizes Q^π—at two levels that differ in what they condition on. At the task level, x is the task, and Q^π is averaged over its initial states. For each candidate q, the controller runs complete episodes using that entry time, and the regulator selects the q with the highest estimated success rate. This produces one default entry per task, used at every planning call. The task-level estimate is coarse but statistically reliable. Its rollouts also include the first planning call, where the choice of entry time matters most, and therefore capture most of the available benefit (Appendix E).

Table 2: Success (%) on held-out test states (states 15–49, 350 episodes per cell); vanilla FM, fixed visual prior and Beyond Noise: mean over three training seeds; WarmPrior, LeaP, Diffusion Policy and STEP: one training run. NFE: sampling steps per planning call. Latency: median ms per planning call (one RTX PRO 5000 GPU, batch 1). Bold: within 3 pp of the best. ‡With the per-call refinement (LIBERO-10), which also computes a cold-start action for its evidence. *On the first call 100 steps (479 ms). §Executes one action per call.

Method	NFE	Params	Object	Spatial	Goal	LIBERO-10	Average	Latency
Vanilla FM	1	86.2 M	76.6	78.3	72.0	30.9	64.4	8.7
WarmPrior (Kang et al., 2026)	1	21.9 M	**93.7**	76.3	78.3	40.0	72.1	8.8
Fixed visual prior	1	86.2 M	87.7	**82.6**	78.9	44.0	73.3	11.4
LeaP (Dai et al., 2026)	1	22.1 M	56.6	58.3	52.0	16.6	45.9	9.1
Diffusion Policy (Chi et al., 2023)	100	88.4 M	**92.0**	77.2	54.6	28.9	63.2	284.2§
STEP (Li et al., 2026b)	2*	265.8 M	85.4	62.3	58.9	30.0	59.1	13.2*
Beyond Noise	1	86.9 M	**92.9**	79.4	**87.4**	**64.3**	**81.0**	17.6/20.5‡

Per-call refinement (state level). At the state level, $x = z_t$ denotes the controller's evidence at the current planning call: how well the predictor anticipated the current frame, how far the anchor lies from a cold-start action and how far the episode has progressed. Training data come from *branch-and-hold* continuations. At a selected mid-episode branch call, the controller continues under each candidate q, which is held fixed until the end of the episode. A value model $\widehat{Q}_\psi(z_t, q)$ fitted to these outcomes allows the regulator to choose an entry time at every planning call without performing rollouts at deployment.

Combining the two levels. Beyond Noise selects a default entry on validation data and applies the regulator's per-call refinement where validated, dynamically adjusting q according to the current state. The refinement is enabled for a policy and benchmark suite only if it outperforms the default entry on held-out branch-and-hold groups. When the refinement is disabled, or when the current state provides no evidence supporting a different choice, the regulator falls back to the default entry. In either case, the flow carries $X_{q^\star}$ to the action in one step.

5 Experiments

5.1 Setup

We evaluate on LIBERO-Object, -Spatial, -Goal and LIBERO-10 with prediction horizon 16, action horizon 8, a 600-step limit, deterministic scene geometry and one function evaluation (NFE) per planning call. LIBERO fixes 50 initial states per task; we split them into a *validation* bank (states 0–14, 150 episodes per suite), on which every entry time, checkpoint and regulator rule is chosen, and a *test* bank (states 15–49, 350 episodes per suite), on which each chosen configuration is evaluated. For vanilla flow matching, the fixed visual prior, Beyond Noise and its ablations we report the mean over three training seeds; external baselines are single training runs.

Comparators. Table 2 compares Beyond Noise with vanilla flow matching (a Gaussian start of the same architecture); the *fixed visual prior*, the AVIS anchor under the legacy contract, trained with our recipe and capacity; Diffusion Policy (Chi et al., 2023); and three informative-source policies: WarmPrior (Kang et al., 2026) (previous chunk), LeaP (Dai et al., 2026) (learned state-conditioned source) and STEP (Li et al., 2026b) (predicted chunk).

5.2 Main results

On held-out initial states, Beyond Noise reaches 81.0% average success (Table 2), against 73.3% for the fixed visual prior and 64.4% for vanilla flow matching, each with one function evaluation per planning call, and above every external baseline (WarmPrior 72.1%, Diffusion Policy 63.2%, STEP 59.1%, LeaP 45.9%). The gains concentrate on long-horizon and multi-object tasks: $+20.3$ pp over

Table 3: What each component adds (test states, success %, mean over three training seeds); each row adds to the row above. The middle row uses $q = 0.95$.

	Object	Spatial	Goal	LIBERO-10	Average
Fixed visual prior ($B_t + 0.5\varepsilon$ at $s_0 = 0$)	87.7	82.6	78.9	44.0	73.3
+ coupled entry	85.4	76.9	85.4	60.1	77.0
+ adaptive regulator (Beyond Noise)	92.9	79.4	87.4	64.3	81.0

the fixed visual prior and $+24.3$ over WarmPrior on LIBERO-10, and $+8.5$ and $+9.1$ on Goal. The fixed visual prior remains stronger on Spatial (82.6 against 79.4%), whose tasks are short pick-and-place episodes with nearly fixed object placements. Beyond Noise costs 17.6 ms per planning call (20.5 ms with the per-call refinement) against 8.7–13.2 ms for the other one- and two-step policies, mostly for frame encoding.

5.3 Why Beyond Noise works

The coupled entry lets more of the source reach the action. Under matched retraining on LIBERO-10, 0.69–0.73 of an anchor displacement reaches the executed action under the coupled entry at $q = 0.95$, against 0.09–0.10 under the legacy contract (two seeds). The noise scale is the main factor: at the coupled entry, transmission falls from 0.89 to 0.71, 0.48 and 0.20 as the noise rises from 0.025 to 0.05, 0.1 and 0.5, while the anchor weight barely matters. Higher transmission is not by itself higher success (0.46–0.90 across low-noise contracts, with success within 272–298 of 500): more of the anchor helps only where it is worth keeping, which the regulator decides. At $q = 0.95$ on LIBERO-10, the coupled entry reaches 55.2 and 57.4% (two seeds) against 46.6 and 43.6% for the legacy contract with the same anchor and recipe.

Which parts of the coupling matter. Breaking the coupling's ties one at a time and retraining, only a fixed noise scale of 0.5 costs success (11.7 pp, two seeds): a heavily noised anchor is refined away. Entering at $s_0 = 0$ leaves success unchanged but lowers transmission from 0.71 to 0.53: the entry time shapes how much survives, though less than the noise scale.

Adaptive regulation pays off where a fixed entry fails. Held at $q = 0.95$, the coupled entry lowers success on Object and Spatial (Table 3). The adaptive regulator raises success over this fixed entry on every suite, lifting the average from 77.0 to 81.0%. Choosing the default entry per task recovers tasks that any single entry fails. On Spatial task 7 (the bowl on the stove), success falls from 44 to 20 of 50 as q rises from 0.60 to 0.95; the regulator selects a lower entry and lifts the task from 21 to 32 of 35 held-out episodes, and LIBERO-10 task 3 improves from 16 to 20 of 35.

Per-call refinement. Validation decides where the per-call refinement is applied, and where it is applied it pays. On LIBERO-10, validation enables it ($+4.5$ pp on held-out groups), and it raises success on LIBERO's 350 test states from 60.1 to 64.3% and on 500 freshly sampled LIBERO-10 initial states, never used for training or selection, from 70.0 to 74.6% (paired exact McNemar $p = 0.04$). On the other suites' fixed initial states, validation keeps the default entry, as designed. The refinement pays most where states vary beyond the task. Across LIBERO-plus, whose variants perturb camera, lighting, layout and robot state, a regulator fitted for the jointly trained model on 150 variants per suite raises success on the remaining variants of every suite (Object $+2.6$ pp, Spatial $+2.7$, Goal $+3.1$ with $p < 10^{-3}$, LIBERO-10 $+3.4$), and by 3.0 pp over all four suites. Without refitting, the same regulator also raises success on LIBERO's clean Goal test states by 5.1 pp ($p = 0.03$).

5.4 Robustness to invalidated priors

Transient blur. Blur costs most methods little. Beyond Noise changes by -6.9 to $+1.7$ pp, its largest loss on Object, and remains above STEP and LeaP on every suite; on LIBERO-10 it keeps 66.0%, more than twice the best baseline (STEP, 31.1%). WarmPrior loses 5.4–10.0 pp.

Table 4: Robustness: success (%) and, in grey, change from clean (pp). Blur (20-step, radius-4 window) and object shift (3–5 cm at step 20): test states, policies of Table 2. LIBERO-plus (Fei et al., 2026): one model per method for all four suites; Beyond Noise's per-call refinement is fitted on 150 variants per suite, scored on the rest. Bold: within 3 pp of the best.

	Blur				Object shift				LIBERO-plus			
Method	Object	Spatial	Goal	LIBERO-10	Object	Spatial	Goal	LIBERO-10	Object	Spatial	Goal	LIBERO-10
WarmPrior	**86.9** −6.8	70.9 −5.4	72.3 −6.0	30.0 −10.0	18.0 −75.7	33.7 −42.6	39.6 −38.7	23.4 −16.6	44.4	37.5	30.3	13.2
LeaP	56.3 −0.3	59.1 +0.8	52.3 +0.3	16.3 −0.3	16.0 −40.6	28.6 −29.7	28.9 −23.1	8.6 −8.0	59.5	54.8	41.8	22.7
STEP	**85.1** −0.3	60.6 −1.7	58.6 −0.3	31.1 +1.1	34.9 −50.5	30.6 −31.7	29.6 −29.3	14.3 −15.7	65.2	50.6	38.3	3.5
Beyond Noise	**86.0** −6.9	**80.0** +0.6	**86.6** −0.8	**66.0** +1.7	**52.0** −40.9	**50.0** −29.4	**49.5** −37.9	**40.8** −23.5	**81.6**	**77.2**	**71.2**	**51.5**

Abrupt object displacement. Moving the object invalidates the prior at once, and every policy collapses, yet Beyond Noise retains the most success on every suite (40.8–52.0%, against at most 39.6% for WarmPrior, 34.9% for STEP and 28.9% for LeaP), with its largest lead on LIBERO-10 (40.8 against 23.4%). On Object and Spatial only, whose demonstrations barely move the target, Beyond Noise also uses extra moved-object demonstrations (Appendix G: results without them).

Systematic variation. Across LIBERO-plus, Beyond Noise succeeds on 51.5–81.6% of variants per suite (70.3% overall), against 3.5–65.2% for STEP, 22.7–59.5% for LeaP and 13.2–44.4% for WarmPrior, and more than doubles the best baseline on LIBERO-10 (51.5 against 22.7%).

6 Related work

Generative policies and informative sources. Distillation (Prasad et al., 2024; Wang et al., 2025; Sheng et al., 2026), guided sampling (Liu et al., 2025; Black et al., 2025b) and value-based methods (Wang et al., 2023; Park et al., 2025) change the policy or its sampler; we learn one sampler scalar from returns. BRIDGER bridges a source policy to expert actions (Chen et al., 2024; Albergo et al., 2025); other sources come from visual latents, executed-action history or observation features (Gao et al., 2026; Jia et al., 2026; Dong et al., 2025; Jiang et al., 2025; Machado et al., 2026). WarmPrior centers a fixed-variance Gaussian on the previous chunk or forecast (Kang et al., 2026; Liu et al., 2023b), and LeaP learns a state-conditioned mean and variance (Dai et al., 2026; Lee et al., 2022). These works seek a better source; we regulate how much of it survives.

Entering the path partway. SDEdit trades faithfulness against realism through the noise level of a partially noised guide (Meng et al., 2022)—the trade-off our entry time controls. Truncated diffusion starts from a learned prior or anchored Gaussians (Zheng et al., 2023; Lyu et al., 2022; Liao et al., 2025; Lim et al., 2026). Robot policies enter partway from a previous, retrieved, predicted or partially denoised chunk (Duan et al., 2025; Odonchimed et al., 2025; Li et al., 2026b; Høeg et al., 2025; Liu et al., 2026) at a fixed step or by a hand-set rule; Falcon chooses its start per call with a training-free consistency test (Chen et al., 2025). Learned control of the sampler has targeted compute or exploration (Hu et al., 2024; Yu et al., 2025; Li et al., 2026a; Hong et al., 2026); we learn where to enter the flow from counterfactual closed-loop returns.

7 Conclusion

A better prior need not yield a better action: informative sources help, but the choice among them buys little and the flow absorbs most of the source. Beyond Noise therefore decides how much of the prior survives: it selects a default entry on validation data, lets the regulator adjust q per call where validated, and gives the flow patch tokens to use what it keeps, improving on matched and external baselines, most on long-horizon tasks.

AI USE STATEMENT

In this work, we used generative AI tools for writing assistance (editing the prose of this manuscript), for retrieval and discovery (searching for and summarizing related work) and for research ideation and execution (discussing experimental designs and helping write and orchestrate experiment scripts). We have reviewed all AI-assisted work: the authors checked retrieved references against the original sources and verified AI-assisted code and its outputs before using any result. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies control policies for simulated robot manipulation on a public benchmark. It involves no human subjects, no personal data and no deployment on physical systems. We see no ethical concerns beyond those common to learned robot control.

REPRODUCIBILITY STATEMENT

All results use two fixed protocols, each with one training recipe and one evaluator shared by every compared method; model selection uses only the validation bank of initial states, and each chosen configuration is evaluated on the test bank. Vanilla FM, the fixed visual prior, Beyond Noise and its ablations are trained with three seeds and we report their mean; external baselines are trained once; every evaluation uses evaluation seed 42. For every reported cell we record the protocol, training and evaluation seeds, checkpoint and raw per-task report. Appendix A gives the training recipes, evaluators, baseline implementations and split of initial states, and Appendices B–G document the experiments behind each main-text claim. Code is available at `https://anonymous.4open.science/r/Beyond-Noise-56D4/`; checkpoints, the fresh initial states and raw reports will be released upon acceptance.

REFERENCES

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems (RSS)*, 2025a. doi: 10.15607/RSS.2025.XXI.010.

Kevin Black, Manuel Y. Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 33383–33407, 2025b.

Haojun Chen, Minghao Liu, Chengdong Ma, Xiaojian Ma, Zailin Ma, Huimin Wu, Yuanpei Chen, Yifan Zhong, Mingzhi Wang, Qing Li, and Yaodong Yang. Falcon: Fast visuomotor policies via partial denoising. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 8552–8573. PMLR, 2025.

Kaiqi Chen, Eugene Lim, Kelvin Lin, Yiyang Chen, and Harold Soh. Don't start from scratch: Behavioral refinement via interpolant-based policy diffusion. In *Robotics: Science and Systems (RSS)*, 2024. doi: 10.15607/RSS.2024.XX.122.

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

Meipo Dai, Qiyuan Zhuang, He-Yang Xu, Ying-Jie Shuai, Yijun Wang, Qi Dou, and Xiu-Shen Wei. Where should action generation begin? A learnable source prior for generative robot policies. *arXiv preprint arXiv:2606.17408*, 2026.

Zibin Dong, Yicheng Liu, Yinchuan Li, Hang Zhao, and Jianye Hao. Conditioning matters: Training diffusion policies is faster than you think. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 119651–119676, 2025.

Yufei Duan, Hang Yin, and Danica Kragic. Real-time iteration scheme for diffusion policy. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11758–11764, 2025. doi: 10.1109/IROS60139.2025.11247391.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: A progressive robustness benchmark for visual-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 38574–38583, 2026. arXiv:2510.13626.

Dechen Gao, Boqi Zhao, Andrew Lee, Ian Chuang, Hanchu Zhou, Hang Wang, Zhe Zhao, Junshan Zhang, and Iman Soltani. VITA: Vision-to-action flow matching policy. In *International Conference on Learning Representations (ICLR)*, 2026.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Sigmund H. Høeg, Yilun Du, and Olav Egeland. Fast policy synthesis with variable noise diffusion models. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4821–4828, 2025. doi: 10.1109/ICRA55743.2025.11127858. Streaming Diffusion Policy; arXiv:2406.04806.

Matthew M. Hong, Jesse Zhang, Anusha Nagabandi, and Abhishek Gupta. TMRL: Diffusion timestep-modulated pretraining enables exploration for efficient policy finetuning. In *Robotics: Science and Systems (RSS)*, 2026. doi: 10.15607/RSS.2026.XXII.208.

Xixi Hu, Bo Liu, Xingchao Liu, and Qiang Liu. AdaFlow: Imitation learning with variance-adaptive flow-based policies. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 138836–138858, 2024.

Jindou Jia, Gen Li, Xiangyu Chen, Tuo An, Yuxuan Hu, Jingliang Li, Xinying Guo, and Jianfei Yang. Action-to-action flow matching. In *Robotics: Science and Systems (RSS)*, 2026. doi: 10.15607/RSS.2026.XXII.209.

Sunshine Jiang, Xiaolin Fang, Nicholas Roy, Tomás Lozano-Pérez, Leslie Pack Kaelbling, and Siddharth Ancha. Streaming flow policy: Simplifying diffusion/flow-matching policies by treating action trajectories as flow trajectories. In *Conference on Robot Learning (CoRL)*, volume 305 of *Proceedings of Machine Learning Research*, pp. 238–257, 2025.

Sinjae Kang, Chanyoung Kim, Kaixin Wang, Li Zhao, and Kimin Lee. WarmPrior: Straightening flow-matching policies with temporal priors. *arXiv preprint arXiv:2605.13959*, 2026.

Sang-gil Lee, Heeseung Kim, Chaehun Shin, Xu Tan, Chang Liu, Qi Meng, Tao Qin, Wei Chen, Sungroh Yoon, and Tie-Yan Liu. PriorGrad: Improving conditional denoising diffusion models with data-dependent adaptive prior. In *International Conference on Learning Representations (ICLR)*, 2022.

Andrew Zou Li, Gokul Swamy, Yonatan Bisk, and Andrea Bajcsy. ELASTIC: Efficiently learning to adaptively scale test-time compute for generative control policies. *arXiv preprint arXiv:2606.31132*, 2026a.

Jinhao Li, Yuxuan Cong, Yingqiao Wang, Hao Xia, Shan Huang, Yijia Zhang, Ningyi Xu, and Guohao Dai. STEP: Warm-started visuomotor policies with spatiotemporal consistency prediction. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026b.

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, and Xinggang Wang. DiffusionDrive: Truncated diffusion model for end-to-end autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12037–12047, 2025.

Jia Syuen Lim, Zhizhen Zhang, Peter Bohm, Brendan Tidd, Zi Huang, and Yadan Luo. AnchorVLA: Anchored diffusion for efficient end-to-end mobile manipulation. *arXiv preprint arXiv:2604.01567*, 2026.

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

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 36, pp. 44776–44791, 2023a.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023b.

Yuejiang Liu, Jubayer Ibn Hamid, Annie Xie, Yoonho Lee, Maximilian Du, and Chelsea Finn. Bidirectional decoding: Improving action chunking via guided test-time sampling. In *International Conference on Learning Representations (ICLR)*, 2025.

Yufeng Liu, Hang Yu, Juntu Zhao, Bocheng Li, Di Zhang, Mingzhu Li, Wenxuan Wu, Yingdong Hu, Junyuan Xie, Junliang Guo, Dequan Wang, and Yang Gao. Learning native continuation for action chunking flow policies. In *Robotics: Science and Systems (RSS)*, 2026. doi: 10.15607/RSS.2026.XXII.058.

Zhaoyang Lyu, Xudong Xu, Ceyuan Yang, Dahua Lin, and Bo Dai. Accelerating diffusion models via early stop of the diffusion process. *arXiv preprint arXiv:2205.12524*, 2022.

Bruno Machado, Alexandre Chapin, Emmanuel Dellandrea, and Liming Chen. Flowing with purpose: Latent action guided flow matching policies for robotic manipulation. *arXiv preprint arXiv:2606.23420*, 2026.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2022.

Sodtavilan Odonchimed, Tatsuya Matsushima, Simon Holk, Yusuke Iwasawa, and Yutaka Matsuo. Retrieve-augmented generation for speeding up diffusion policy without additional training. *arXiv preprint arXiv:2507.21452*, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 2024. ISSN 2835-8856.

Seohong Park, Qiyang Li, and Sergey Levine. Flow Q-learning. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 48104–48127. PMLR, 2025.

Aaditya Prasad, Kevin Lin, Jimmy Wu, Linqi Zhou, and Jeannette Bohg. Consistency policy: Accelerated visuomotor policies via consistency distillation. In *Robotics: Science and Systems (RSS)*, 2024. doi: 10.15607/RSS.2024.XX.071.

Nur Muhammad Mahi Shafiullah, Zichen Cui, Ariuntuya Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Juyi Sheng, Ziyi Wang, Peiming Li, and Mengyuan Liu. MP1: MeanFlow tames policy learning in 1-step for robotic manipulation. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 40, pp. 18532–18539, 2026. doi: 10.1609/aaai.v40i22.38919.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021.

Zhendong Wang, Jonathan J. Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2023.

Zhendong Wang, Max Li, Ajay Mandlekar, Zhenjia Xu, Jiaojiao Fan, Yashraj Narang, Linxi Fan, Yuke Zhu, Yogesh Balaji, Mingyuan Zhou, Ming-Yu Liu, and Yu Zeng. One-step diffusion policy: Fast visuomotor policies via diffusion distillation. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 63399–63416, 2025.

Shu-Ang Yu, Feng Gao, Yi Wu, Chao Yu, and Yu Wang. D3P: Dynamic denoising diffusion policy via reinforcement learning. *arXiv preprint arXiv:2508.06804*, 2025.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3D diffusion policy: Generalizable visuomotor policy learning via simple 3D representations. In *Robotics: Science and Systems (RSS)*, 2024.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems (RSS)*, 2023. doi: 10.15607/RSS.2023.XIX.016.

Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Truncated diffusion probabilistic models and diffusion-based adversarial auto-encoders. In *International Conference on Learning Representations (ICLR)*, 2023.

A PROTOCOLS AND REPRODUCIBILITY

The paper uses two protocols. The *matched-source protocol* (Section 3) isolates the source: three policies that differ only in their source share one architecture, one training recipe and one evaluator. The *path-time protocol* (Sections 4–5) trains and evaluates Beyond Noise, its ablations and the fixed visual prior, and separates every choice from its evaluation. Success rates are never pooled across the two protocols.

A.1 RECIPES AND EVALUATORS

Table 5: Training recipes and evaluators of the two protocols.

	Matched-source (Section 3)	Path-time (Sections 4–5)
Action head	transformer, 8 layers, width 384, 6 heads	transformer, 8 layers, width 768, 12 heads
Parameters	21.9 M	86.2 M (86.9 M with patch memory)
Visual input	DINOv2-base CLS, 224 px	DINOv2-base CLS (+ 4×4 patch grid), 224 px
Horizons	observation 2, prediction 16, action 8	observation 2, prediction 16, action 8
Source	Gaussian, temporal or visual (legacy contract)	coupled entry $X_q = qB_t + (1-q)\varepsilon$ at $s_0 = q$
Optimization	batch 1024, 6,250 steps; lr 10^{-4}, weight decay 10^{-5}, warmup 0.2	global batch 512; Stage 1 50 k, Stage 2 12 k, patch memory 12 k steps; lr 10^{-4}, weight decay 10^{-5}, warmup 0.2, EMA, dropout 0.1
Training seeds	42 (Goal also 43, 44)	45; three-seed runs 42, 43, 123
Episodes	10 tasks $\times$ 20 per suite	10 tasks $\times$ 50 initial states; validation 0–14, test 15–49
Step limit	280 (520 on LIBERO-10)	600
Evaluator	anchored, seed 42, NFE 1; render-verified GPUs (mujoco 2.3.7)	anchored, seed 42, NFE 1, deterministic scene geometry; render-verified GPUs

AVIS predictor. Both protocols draw the visual anchor from AVIS, a transformer (6 layers, width 384, 6 heads, MLP width 1536, dropout 0.15; 23.9 M parameters) over the 16×16 DINOv2-base patch grids and CLS features of the two frames preceding the current observation, conditioned by adaptive layer normalization on proprioception and the recent executed actions. It predicts the next frame's patch and CLS features and, through a 16-query action head, the next 16-step action chunk (DCT parameterization, Gaussian likelihood with a Bernoulli gripper). One predictor is trained per suite for 20 k steps and selected on a joint feature and action validation score; it is frozen in every policy that uses it.

A.2 SEPARATING CHOICES FROM EVALUATION

LIBERO fixes 50 initial states per task. The path-time protocol splits them once (Figure 3): every choice—entry time, checkpoint and initialization, regulator rule and its hyperparameters—is made on states 0–14 (150 episodes per suite), and each chosen configuration is evaluated once on states 15–49 (350 episodes per suite). Selection files record the validation counts and are frozen before the corresponding test run. The per-call refinement is additionally tested on 500 fresh LIBERO-10 initial states that played no role in any choice (Appendix F).

A.3 TRAINING BEYOND NOISE

Beyond Noise is trained in three stages (Figure 4). Stage 1 trains the coupled flow from random initialization with $q \sim \mathcal{U}(0, 0.95)$ (50 k steps). Stage 2 continues it with the anchor curriculum of Section 4.2 (12 k steps). The patch-memory stage then adds the 4×4-pooled DINOv2 patch tokens of the two observation frames to the decoder's cross-attention memory, next to the time, observation and proprioception tokens; it is initialized from the coupled policy, with the initialization and checkpoint chosen on validation states, and trained for 12 k steps with the Stage-2 recipe. Every Beyond Noise policy, including those in Table 2, uses the patch tokens.

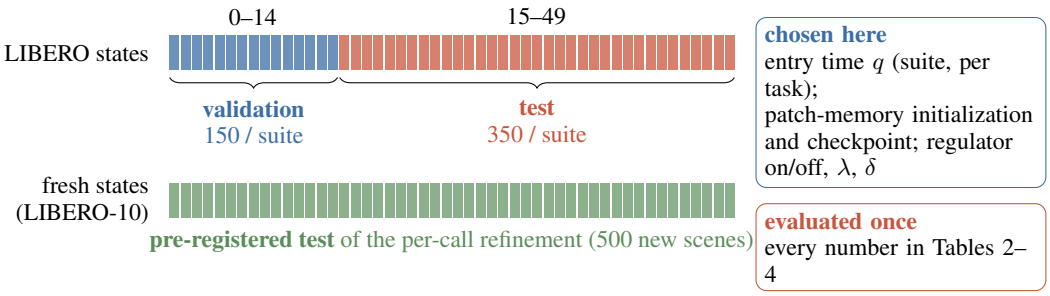

Figure 3: The path-time protocol separates choices from evaluation. Each suite has 10 tasks with 50 fixed initial states; states 0–14 are used for every choice and states 15–49 only to evaluate the chosen configuration. The per-call refinement is also evaluated on 500 freshly sampled LIBERO-10 scenes.

Band specialist. Stage 2 is restricted to the band $q \in [0.85, 0.95]$, where most deployment situations offer a reliable anchor; behavior at lower entries is inherited from Stage 1. A direct comparison supports this choice: at $q = 0.95$ on LIBERO-10 test states, the 86.2 M band specialist reaches 58.3% against 44.6% for a generalist of the same size whose Stage 2 covers $q \in [0, 0.95]$, and larger generalists (124 M and 219.9 M) do not close the gap.

Patch tokens and object displacement. Patch tokens let the flow see where objects are, so the action can follow a displaced object; they improve success under object shift on Goal and LIBERO-10. There the demonstrations already vary the target's placement (placement s.d. 1.45 cm on LIBERO-10), and the cross-attention learns the displacement relationships between patches from the original data alone, without any augmentation. On Object the target barely moves across the original demonstrations (placement s.d. 0.14 cm; 6 of the 10 tasks place it at exactly the same position), and on Spatial the spread is also small (0.93 cm); such displacements give too little supervisory signal to learn how the action should move with the object. We therefore augment Object and Spatial with simulated expert demonstrations under perturbation, raising the displacement to a clearly observable level. In each demonstration's scene the target object is moved by 0.5–3 cm (uniform over a disc), either before the episode starts or at a step between 10 and 30 before the grasp; the expert trajectory is retargeted to the new position and replayed in simulation, and only successful, collision-free episodes are kept. This adds 1,742 episodes on Object and 1,563 on Spatial to the 500 original demonstrations and raises the per-task placement spread to 1.28–1.44 cm and 1.27–2.15 cm, comparable to LIBERO-10. All three stages are then trained on the augmented data with the same recipe. Compared with training on the original data only, Beyond Noise succeeds far more often under object shift (Object: 29.6 to 52.0%; Spatial: 33.4 to 50.0%; Appendix G).

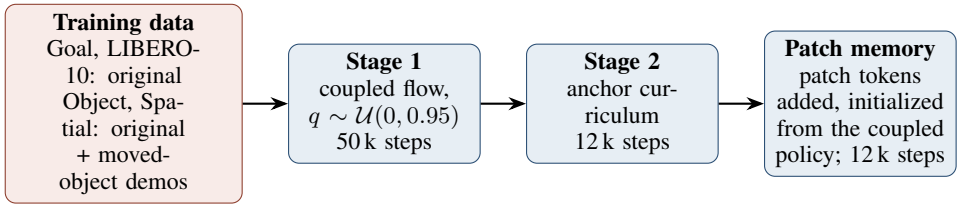

Figure 4: Training Beyond Noise. The coupled flow is trained in two stages; the patch-memory stage then adds patch tokens to the decoder's cross-attention and is initialized from the coupled policy. On Object and Spatial the original demonstrations are augmented with simulated demonstrations in which the target object is moved.

Entry-time sweeps. Suite entry times are the argmax of validation success over $q \in \{0.60, 0.75, 0.85, 0.90, 0.95\}$ (ties to the higher entry). Spatial: 94, 112, **118**, 116 and 109 of 150; Goal: 125, 128, 130, **132** and 128. On Object, $q = 0.85$ is the validation-best entry of the branch-and-hold held-q curves on states 0–14. A task departs from its suite's entry only if another entry

wins on that task by at least 3 of its 15 validation episodes; this happens once (Spatial task 7, the bowl on the stove, $q = 0.75$).

A.4 BASELINES

All baselines are evaluated with the path-time evaluator on the same test states 15–49 (Table 6); where a baseline has several checkpoints, the checkpoint is chosen on validation states 0–14.

Table 6: Baseline implementations.

Method	Code	Training	Sampler	Params
Vanilla FM	standard implementation	standard flow matching from Gaussian noise	1 Euler step	86.2 M
Fixed visual prior	ours	legacy contract $B_t + 0.5\varepsilon$ at $s_0 = 0$, 12 k from Stage 1	1 Euler step	86.2 M
WarmPrior	official release	temporal prior (previous chunk, $\sigma = 0.5$), per suite	1 step	21.9 M
LeaP	reimplemented in the WarmPrior release	learned state-conditioned source, per suite (Object: task-conditioned variant)	1 step	22.1 M
Diffusion Policy	official release	per suite	100 steps	88.4 M
STEP	official release	25 k steps, EMA weights	$k' = 2$ (100 on first call)	265.8 M

LIBERO-plus models. For Table 4's LIBERO-plus columns each method is trained once on the four suites jointly (one model per method, task-conditioned) and evaluated on the 10,030 perturbed variants, one episode each, in all seven perturbation categories. Beyond Noise is reported with the per-call refinement of Appendix E, fitted on 150 variants per suite, on the remaining 9,430 variants. For the per-category breakdown (Figure 17), its default entries are chosen per task on clean validation states before any perturbed episode is run (Object 0.90, Spatial 0.85, Goal 0.60, LIBERO-10 0.90 with task 3 at 0.95).

A.5 CONTROLLED EVALUATION

Render-verified GPUs. Every reported evaluation uses fixed seeds and the same evaluator and flags for all methods, and runs on GPUs verified to render a frozen simulator state to the same frame in 40 of 40 renders. We introduced this check after finding that some GPUs render a frozen state to slightly different frames on repeated calls: three repeats of one unchanged matched-source arm, run before the check, scored 153, 148 and 145 of 200, a spread of 4.0 pp (Figure 5). The path-time protocol evaluates only on render-verified GPUs, and every matched-source result in Section 3 and Appendix B.2 was rerun on render-verified GPUs of a single server with the same checkpoints, evaluator and flags. In the rerun, all three arms start from an identical frame in every episode of every cell. The rerun confirms every conclusion: the warm-start gains grow slightly ($+15.7$ and $+16.9$ pp against $+13.9$ and $+15.2$ pp) and the visual–temporal difference stays null ($+1.2$ pp, 95% CI $[-2.7, +5.2]$).

Paired comparisons. All methods are evaluated on the same initial states with the same evaluation seed, and comparisons are paired by initial state; where initial frames are recorded, paired tests are also reported on the episodes whose first frames match across arms, and conclusions are drawn only where the two agree. On the path-time protocol, evaluation noise alone gives the paired difference between two arms a standard deviation of about 1.9 pp at 350 episodes; we treat smaller differences as inconclusive.

B THE SOURCE STUDY IN DETAIL

This appendix supports each finding of Section 3 with the underlying measurements. Results use the matched-source protocol (Appendix A) unless a subsection states otherwise.

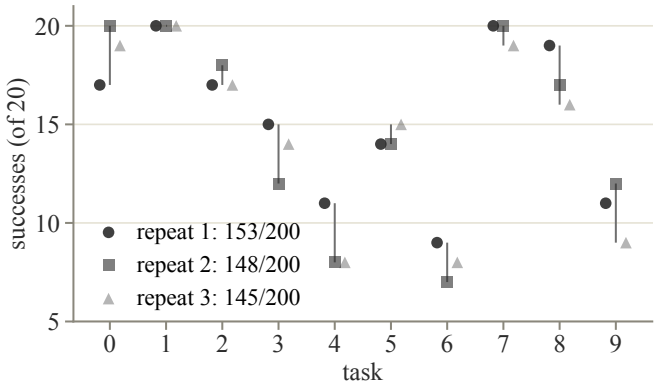

Figure 5: Three repeats of one matched-source arm (Goal, temporal source, seed 42), run before render verification: successes per task out of 20. All reported results use render-verified GPUs.

B.1 The training configuration can decide a source comparison

With the same Goal data, architecture, seed and 6.4 M sample exposures, the gap between the temporal and Gaussian sources depends strongly on the optimization configuration (Figure 6): at batch 4096 it grows from $+1.0$ to $+14.5$ pp as the learning rate rises from 10^{-4} to 4×10^{-4}, and at batch 1024 and learning rate 10^{-4} it reaches $+25.5$ pp (95% CI $[+8.0, +44.5]$). The temporal arm improves monotonically with the learning rate while the Gaussian arm stays flat: an unscaled learning rate at large batch starves the informative-source arm specifically. This sweep was scored with the evaluator of the WarmPrior release; with our evaluator the batch-1024 checkpoints give 56.0 and 80.5% (Goal seed 42 of Table 1), so the direction holds and no comparison crosses evaluators. The matched-source protocol therefore fixes batch 1024, and every path-time comparison keeps global batch 512.

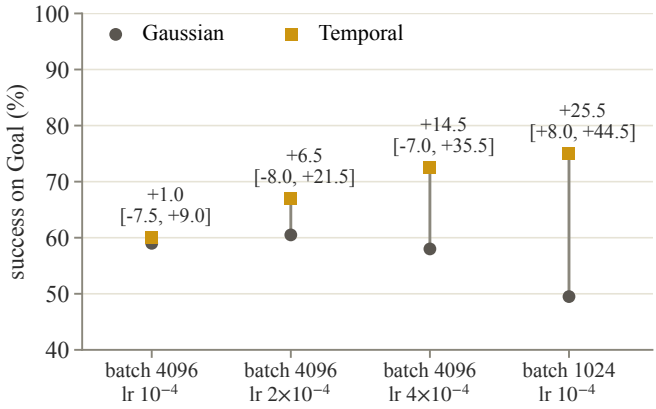

Figure 6: The training configuration can decide a source comparison (Goal, 6.4 M sample exposures, 200 episodes per point). Labels: temporal−Gaussian gap in pp with its 95% task-cluster bootstrap interval.

B.2 More integration steps cannot replace a prior

Figure 7 repeats the matched-source evaluation at NFE $\in \{1, 4, 8, 16\}$ for all three sources on four suites (200 episodes per point, render-verified GPUs).

Informative sources help at every budget. Pooled over the four suites, the temporal and visual warm starts exceed the Gaussian start at every number of integration steps, by $+9.6$ to $+16.5$ pp, with every 95% interval above zero.

More steps do not replace the prior. The Gaussian start does not catch up as steps are added (52.8, 55.0, 50.9 and 49.2% at NFE 1, 4, 8 and 16): even at its best budget it stays below every warm start at every budget (60.8–69.3%). Individual cells vary by a few points—on LIBERO-10 four steps help the Gaussian and temporal sources by 5.0–5.5 pp—but no budget closes the gap. More steps reduce the error of integrating the learned field; they do not supply the information a prior carries. The warm starts need no extra steps either: both are at their best at one step (temporal 66.7, 64.6, 60.8 and 61.4%; visual 69.3, 64.9, 64.3 and 64.1%), so one step suffices.

But much of the prior is attenuated. Although more steps cannot replace the prior, the flow transmits only part of it: most of a displacement of the source is absorbed before it reaches the action (Appendix B.6). Beyond Noise therefore controls how much of the prior survives (Section 4).

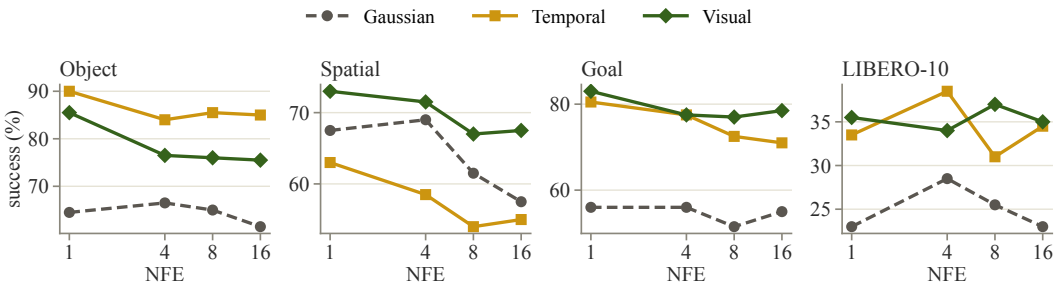

Figure 7: Success versus integration steps for the three sources on four suites (matched-source protocol, training seed 42, 200 episodes per point). Pooled over the four suites, both warm starts exceed the Gaussian start at every budget.

B.3 Capacity

A 4× larger action head does not change the picture (Goal, seed 42, 200 episodes per arm; evaluated before the rerun of Appendix A.5, so the table is comparable only within itself):

Configuration	Params	Gaussian	Temporal	Visual	Mean	Temp. − Gauss.
width 384, batch 1024 × 6,250	21.9 M	58.5	78.0	82.0	72.8	+19.5
width 768, batch 1024 × 6,250	86.2 M	56.0	76.5	81.0	71.2	+20.5
width 768, batch 256 × 25,000	86.2 M	59.0	68.0	77.0	68.0	+9.0

The mean changes by -1.6 pp with capacity, and the warm-start margin persists across all three configurations.

B.4 The source buys consistency, not accuracy

Consecutive plans overlap: plan k covers control steps $[j, j{+}16)$ and plan $k{+}1$ covers $[j{+}8, j{+}24)$, so both describe the same eight steps. Over the first 10 decisions of every Goal episode, measured blind to outcome with decision counts matched within 0.7%, an informative source reduces the disagreement between overlapping plans by 26.8% (temporal) and 29.9% (visual), their jerk by 29.1% and 16.3%, and direction reversals by 9.4% and 8.7% (Figure 8); restricted to successful episodes, the reduction in disagreement persists (-9.2% and -11.4%). Offline, the Gaussian and temporal policies predict chunks equally well (error 0.1330 against 0.1329). Traces were recorded by a recording-only copy of the evaluator that alters no action, random draw or control-flow decision.

B.5 More source information: the one-chunk intervention

To test whether additional history helps beyond what AVIS already encodes, the intervention takes an AVIS-based warm-start policy trained for this study on Goal, freezes the policy, visual predictor and backbone, and adds an explicit previous predicted chunk to its source. Configurations are selected

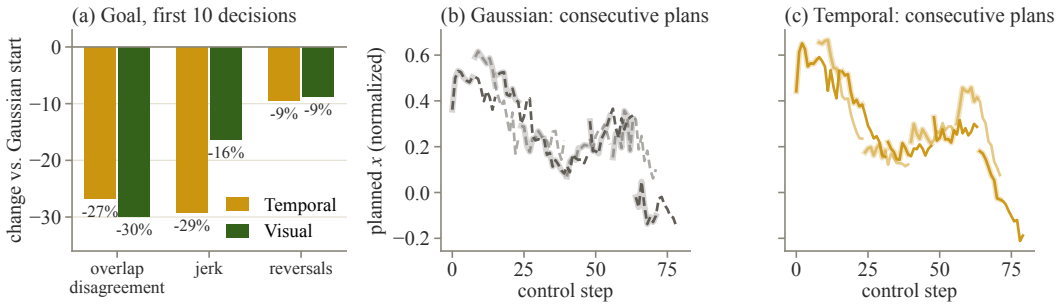

Figure 8: An informative source makes consecutive plans agree. (a) Change of three consistency measures relative to the Gaussian start (Goal, first 10 decisions of every episode). (b,c) The first nine plans of one episode (Goal task 0), end-effector x; the executed first half of each plan is shaded. Under the Gaussian start successive plans disagree where they overlap.

on four paired discovery seeds and evaluated on 32 independent paired seeds in each of 40 decision states, with branches sharing state, observation, continuation policy and noise. Success changes from 66.17% to 66.25% ($+0.08$ pp; episode-cluster 95% CI $[-1.25, +1.41]$); five states favor the additional chunk, four the alternative and 31 tie, and none remains significant after Holm correction (Figure 9).

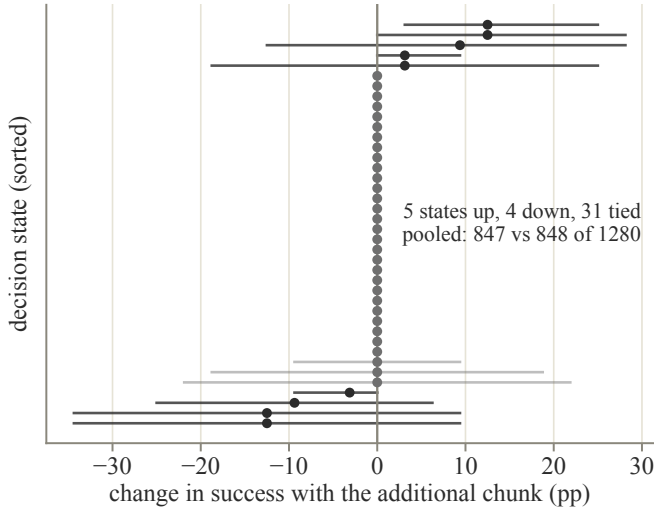

Figure 9: Per-state effect of adding the previous chunk to the visual source (40 decision states, 32 paired seeds each; 95% intervals). No state changes after Holm correction.

B.6 How much of a source reaches the action

For each policy we displace its own deployment source by an offset ΔX along an end-effector axis (0.02, 0.05 and 0.10 raw units, both signs), hold the observation and noise fixed, and record the normalized separation $d(s) = \|X_s^{(a)} - X_s^{(b)}\|/\|\Delta X\|$ along a 16-step integration and the transmission $T = \mathrm{RMS}(\Delta\widehat{A})/\mathrm{RMS}(\Delta X)$ to the executed action at the deployed single step (Figure 10).

Weakening the condition does not raise transmission. Repeating the probe with the condition replaced by a frame lagged by $k \in \{2, 4, 8\}$ control steps or by a static first frame degrades the action (offline error rises, e.g. from 0.147 to 0.307 for the Gaussian policy at lag 8 and to 0.561 with a static frame) while its sensitivity to the source stays flat under lag and falls with a static frame

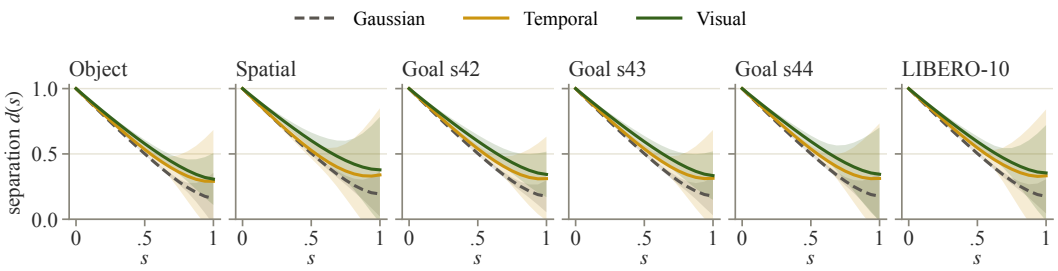

Figure 10: Normalized separation $d(s)$ between trajectories launched from a policy's deployment source and a displaced copy, along a 16-step integration (mean $\pm$ s.d. over windows), in each of the six suite–seed cells.

(Figure 11). The attenuation is therefore not a consequence of observation conditioning overriding the source.

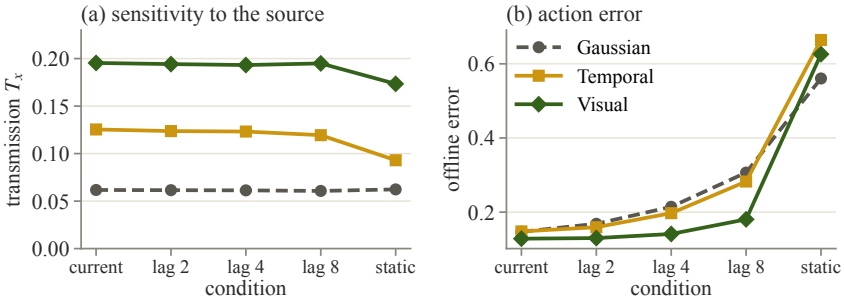

Figure 11: Weakening the condition degrades the action (b) but does not raise its sensitivity to the source (a). Mean over six cells.

B.7 The source is used, but only to seed the path

Replacing a warm-start policy's source at inference raises its error: by a factor of 1.6 for the temporal policy and 2.9 for the visual policy when the source is replaced by Gaussian noise (Figure 12). The Gaussian policy is indifferent to the source it receives.

	source at inference Gaussian	previous chunk	visual
Gaussian policy	**0.147**	0.145	0.148
temporal policy	0.244	**0.148**	0.165
visual policy	0.375	0.281	**0.128**

Figure 12: Offline error of the executed window when each policy receives each source (mean over six cells; bold: the deployed source). Warm-start policies rely on their source; the Gaussian policy does not.

B.8 Exchanging the sources

To exchange the two informative sources in distribution, one policy per suite (Goal and LIBERO-10, seed 42) is trained with the matched-source recipe, changed only in its source: for each training sample a fair coin selects the visual or the temporal source ($B_t + 0.5\varepsilon$), and no input identifies which was drawn. On every window where both sources exist (3,647 on Goal, 3,880 on LIBERO-10),

the policy receives each source with the same noise and observation. Exchanging them moves the executed action by only 20.0% (Goal) and 26.6% (LIBERO-10) of the distance between the sources, with a directional gain of 0.07 and 0.16; a displacement of the same size taken from another window is absorbed as much (0.20–0.23), and an i.i.d. Gaussian displacement slightly more (0.14–0.16). In closed loop the mixed policy succeeds about equally often with either source (Goal 146 and 149, LIBERO-10 84 and 94 of 200; Figure 13).

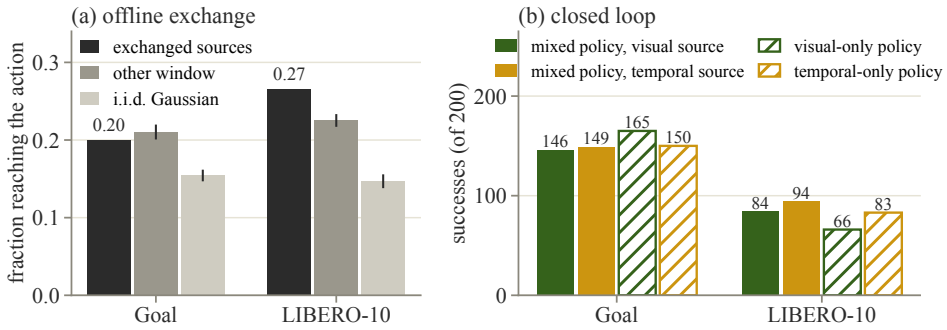

Figure 13: Exchanging the informative sources barely changes the action or the outcome. (a) Fraction of the source difference that reaches the executed action, against two displacements of the same size (bars: range over the two base sources). (b) Closed-loop successes of the mixed policy with each deployed source, and of the single-source policies.

C How the entry time controls survival

The coupled entry is meant to turn *how much of the prior survives* into one scalar. We measure survival with the transmission probe of Appendix B.6, applied to the anchor rather than to a whole source: raw end-effector offsets of 0.02, 0.05 and 0.10 in both signs are added to B_t, so the entry state moves by $\alpha\,\Delta B$, while the noise, observation and flow time are held fixed, and $T = \mathrm{RMS}(\Delta\widehat{A})/\mathrm{RMS}(\Delta B)$ is measured on the executed window (3,880 LIBERO-10 windows).

The entry time controls how much of the anchor survives. On the 86.2 M LIBERO-10 specialist, transmission rises steadily with the entry time, from 0.004 at $q = 0.05$ to 0.028, 0.074 and 0.50 at $q = 0.30$, 0.60 and 0.95 (Figure 14), while the offline error of the action falls from 0.181 to 0.038. One scalar thus moves the policy across two orders of magnitude, from nearly discarding the anchor to passing half of any change in it to the action. Which entry is best is a closed-loop question, which the regulator answers (Appendix E). Entries below 0.85 lie outside this specialist's Stage-2 band, so the low-q values bound rather than measure what a full-band policy would keep.

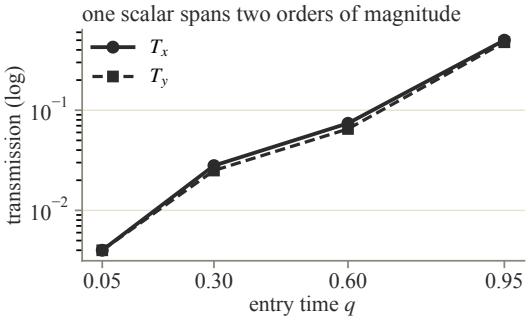

Figure 14: Transmission of an anchor displacement to the executed action along the entry time (86.2 M LIBERO-10 specialist, log scale).

Lower noise preserves the source; transmission alone does not set success. We break the ties of the coupled entry $(\alpha, \sigma, s_0) = (q, 1-q, q)$ one at a time and retrain (21.9 M LIBERO-10 policies,

12 k steps from a shared Stage-1 policy, training seeds 45 and 46, 500 episodes each; Table 7). Two findings follow. First, the noise scale sets how much of the source survives: at the same anchor weight and entry time, halving the noise raises transmission from 0.73 to 0.90, doubling it lowers transmission to 0.50, and a fixed $\sigma = 0.5$ leaves 0.20; the legacy contract transmits least (0.10). Second, higher transmission does not by itself mean higher success: among the low-noise contracts, which transmit 0.50–0.90 of the anchor, success does not rise with transmission. Keeping more of the anchor pays only where the anchor is worth keeping, which is what the regulator decides.

Table 7: Source contracts at $q = 0.95$ (LIBERO-10, 500 episodes per seed). T_x, T_y: transmission of an anchor displacement to the executed action.

Arm	α	σ	s_0	Successes (seed 45 / 46)	T_x	T_y	Offline error
Coupled (ours)	q	$1-q$	q	276 / 287	0.727	0.687	0.057
Unshrunk anchor	1	$1-q$	q	298 / 272	0.751	0.709	0.057
Half noise	q	$\frac{1}{2}(1-q)$	q	279 / —	0.903	0.868	0.053
Double noise	q	$2(1-q)$	q	276 / —	0.504	0.464	0.063
$\sigma = 0.5$	q	0.5	q	214 / 232	0.199	0.192	0.099
Legacy	1	0.5	0	233 / 218	0.096	0.088	0.075

D PATCH MEMORY

Patch tokens localize objects. On 8,400 LIBERO-10 planning-call states, a linear read-out of the object's position relative to the end effector errs by 2.44 cm from the DINOv2 CLS feature and by 1.42 cm from a 4×4 pooled patch grid; finer grids gain at most 0.05 cm (Figure 15a).

The action follows a moved object. When the target object is moved (170 approach states, $q = 0.95$), the planned action of the policy with patch tokens follows it with a gain of 7.42, about 80% of the expert's 9.06, against 1.11 without them, and 5.23 against 1.18 on held-out tasks. Shuffling the patch grid removes the response (0.20), so it comes from the tokens' content (Figure 15b).

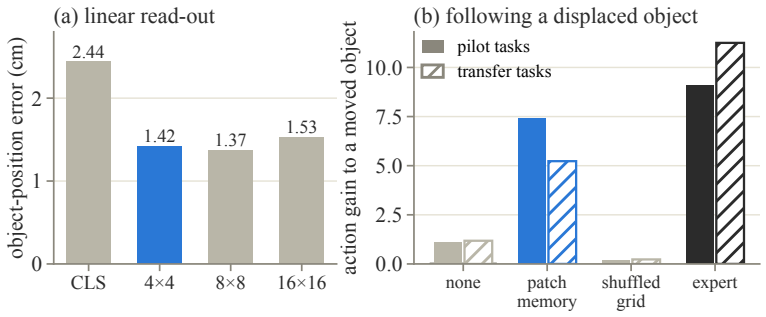

Figure 15: (a) Object-position error of a linear read-out from the global feature and from patch grids. (b) Action gain toward a moved object on pilot and held-out (transfer) tasks: without patch tokens, with them, with the patch grid shuffled, and the expert's.

E COUNTERFACTUAL REGULATION IN PRACTICE

This appendix gives the concrete instantiation of Section 4.3: why the regulator learns from counterfactual rather than logged returns, how the default entry is selected, and how a regulator fitted where states vary behaves.

E.1 WHY COUNTERFACTUAL RETURNS

Regulators fitted to logged experience fail. On three LIBERO-10 generalists, regulators fitted to per-call perturbations of logged rollouts scored 197, 220 and 240 of 500 against 191, 230 and 247 at

a fixed entry; de-confounding the best of them lowered it from 240 to 213, and their held-out ranking accuracy is about 55%, near chance. In logged rollouts the entry time that was used is correlated with how difficult the situation was, which is the confounding that counterfactual continuations remove: branch-and-hold regulators rank held-out arms correctly 63–90% of the time.

E.2 Per task: full-episode counterfactuals

For every task, each validation initial state is run to completion at $q \in \{0.60, 0.75, 0.85, 0.90, 0.95\}$. The suite's entry is the argmax of validation success, and a task departs from it only if another entry wins on that task by at least 3 of its 15 validation episodes. Most tasks are flat across entries or peak at the suite's entry; Spatial task 7 is the exception: for the coupled specialist, success falls from 44 to 20 of 50 as q rises from 0.60 to 0.95. Its decisive choice is made early: holding $q = 0.60$ from the second planning call leaves the task at 20–33% success, whereas entering at 0.60 from the first anchored call reaches 80%. Full-episode counterfactuals include that first call. Conditioning the first call's entry on the regulator's state signals, with the penalty and margin chosen by cross-validation on validation episodes, matches the default entry on every suite (Object 140, Spatial 124 against 122, Goal 127, LIBERO-10 92 of 150) and does not improve on it on test states (Spatial 281 against 284): on LIBERO's fixed initial states the recoverable headroom at the first call is task-level.

E.3 Where states vary: LIBERO-plus

The per-call refinement needs states that differ in ways the task does not already predict. LIBERO-plus supplies them. For the model trained jointly on the four suites, branch-and-hold data were collected on 150 LIBERO-plus variants per suite and one regulator was fitted (held-out ranking accuracy $64.6 \pm 3.4\%$) and deployed with a rule fixed in advance (argmax, margin 0.10) on all suites. On the remaining variants, paired against the same model at a fixed entry:

	Object	Spatial	Goal	LIBERO-10	All suites
Variants	2,368	2,252	2,441	2,369	9,430
Fixed entry (%)	79.0	74.5	68.1	48.1	67.3
Regulated (%)	81.6	77.2	71.2	51.5	70.3
Difference (pp)	+2.6	+2.7	+3.1	+3.4	+3.0

On Goal the paired difference is significant (exact McNemar $p = 10^{-4}$). The same regulator, without refitting, also transfers to LIBERO's clean test states (states 15–49, 350 episodes per suite, every episode in a freshly built environment so that both arms start from identical frames): it raises success on Goal from 85.1 to 90.3% ($+5.1$ pp, $p = 0.03$; $+4.8$ pp, $p = 0.012$ over all 50 states) and on Object from 94.0 to 96.3% ($+2.3$ pp, $p = 0.04$), and leaves Spatial (-0.3 pp) and LIBERO-10 (-2.3 pp, $p = 0.42$) within noise; pooled over the four suites the change is $+1.2$ pp ($p = 0.23$). It departs from the base entry on 32% of Goal and 41% of LIBERO-10 episodes, against 23% on Object and 3% on Spatial. A regulator fitted where states vary thus carries its gain to clean suites other than LIBERO-10, the opposite of the pattern of the regulator fitted on LIBERO's fixed states.

F Fresh initial states

To test the per-call refinement on states that played no role in any choice, we sampled 50 new initial states for each LIBERO-10 task with LIBERO's own reset and new seeds, storing each complete scene (fixture layout and object placements) so that every arm starts from the same scene. Four arms were evaluated on all 500 states under a protocol written before any state was generated (Figure 16). Paired by state, the validation-selected refinement beats the fixed entry by 4.6 pp (69 against 46 discordant episodes; exact McNemar $p = 0.04$); the development rule with margin 0.10, the pre-registered primary arm, does not ($+2.2$ pp, $p = 0.29$); and patch memory adds 6.6 pp at the fixed entry ($p = 0.009$). The first rendered frame is identical across all four arms in 369 of the 500 episodes; restricted to those, the differences are unchanged ($+4.8$ pp, $p = 0.05$; $+2.6$ pp, $p = 0.23$; $+5.8$ pp, $p = 0.03$).

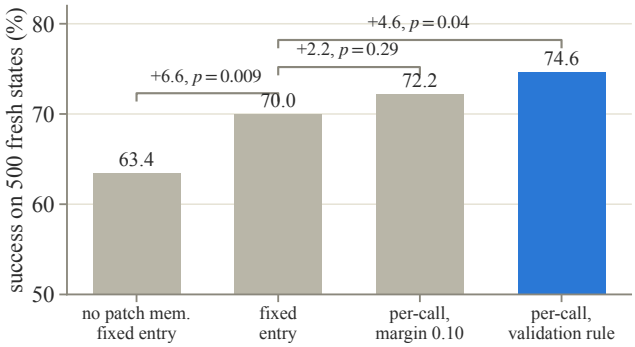

Figure 16: Success on 500 freshly sampled LIBERO-10 scenes, with paired differences and exact McNemar p-values.

G Robustness details

Without the moved-object demonstrations. To separate the effect of the additional Object and Spatial demonstrations (Appendix A.3), we also evaluated Beyond Noise trained on the original demonstrations only (test states; success in %, fixed entry / adaptive regulator):

	Clean	Blur	Object shift
Object	88.6 / 89.7	87.2 / 87.4	29.6 / 29.6
Spatial	69.0 / 66.9	68.4 / 66.0	33.4 / 33.4

The additional demonstrations matter most under object shift: with them, Beyond Noise reaches 52.0% (Object) and 50.0% (Spatial) under shift (Table 4), against 29.6 and 33.4% without them; without them, it falls below STEP on Object (34.9%) and is on par with WarmPrior on Spatial (33.7%). With the regulator, they also change clean success from 89.7 to 92.9% on Object and from 66.9 to 79.4% on Spatial (Table 2), and success under blur from 87.4 to 86.0% and from 66.0 to 80.0%.

LIBERO-plus by category. LIBERO-plus perturbs each suite along seven dimensions. One model per method, trained jointly on the four suites, is evaluated on all 10,030 variants with one episode each (Figure 17; Beyond Noise with its default entries, without the per-call refinement). Beyond Noise leads in every category. Its margin over the best baseline is largest under lighting, language and robot-state changes ($+28$ to $+34$ pp) and smallest under camera and layout changes ($+10$ and $+12$ pp), which change where objects appear in the image.

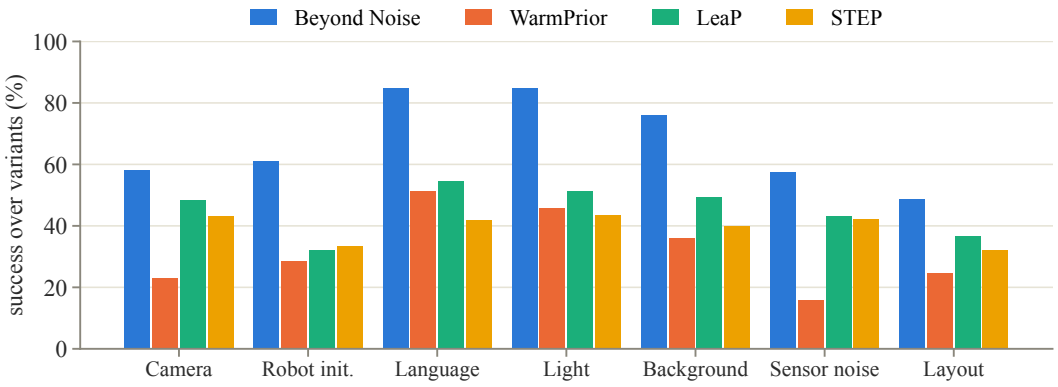

Figure 17: LIBERO-plus success by perturbation category, pooled over the four suites (one model per method trained on all suites; Beyond Noise with default entries chosen per task on clean validation states, without the per-call refinement).