A TRAINING-FREE APPROACH FOR ADAPTING FLOW-MATCHING POLICIES TO SPATIAL SHIFTS

Anonymous authors
Paper under double-blind review

ABSTRACT

A deployed pre-trained policy often has to operate in a fixed spatial layout that differs from the one it was trained on. This spatial shift places the policy outside its training distribution, where it frequently fails. We seek a training-free adaptation that raises the success rate at run time using only the policy's own attempts at that layout: no demonstrations, no weight updates, only the outcomes of its own attempts. A success recorded there represents a complete trajectory that solves the layout; with the weights frozen, that trajectory can only be exploited through the sampler. For a chunked flow-matching policy, the sampler offers three entry points: the measure it starts from, the time at which it enters the flow, and the action it emits. We inject the stored success at all three points. Specifically, warm-started denoising re-enters the flow from the success under the current observation; a density gate, using the density defined by the same velocity field, returns control to the policy once the re-solved chunk is no longer one the policy would produce; and an action-space residual search adds a constant offset fitted using terminal outcomes alone. We consolidate these into WDRS, a plug-and-play adaptation framework, and conduct extensive empirical validation. On LIBERO, WDRS raises single-draw success from .247 to .720 at the smallest of three jitter scales and matches or exceeds a per-site weight-update baseline in five of six suite-by-jitter cells; it carries over to two further frozen flow policies (up to $+.474$), and to the Layout perturbations of LIBERO-plus ($+.290$) and RoboLab ($+.109$, warm-started only). This demonstrates that WDRS is a highly transferable and readily deployable solution.

1 INTRODUCTION

Driven by rapid advances in foundation models and generative AI, generative policies (Kim et al., 2025; Shukor et al., 2025; Black et al., 2026) for robotic control have attracted significant attention. Leveraging the powerful representations of pre-trained vision and language models (Beyer et al., 2024), these approaches exhibit remarkable proficiency in instruction following and complex scene comprehension. Consequently, mainstream architectures typically adopt a decoupled paradigm comprising a perception module and an action head: the former encodes the environmental context, while the latter generates the corresponding actions (NVIDIA et al., 2025). Meanwhile, given the profound capacity to model complex, multi-modal distributions, action heads based on Diffusion (Ho et al., 2020; Song et al., 2021) or Flow Matching (Lipman et al., 2023) have been widely adopted in state-of-the-art frameworks (Intelligence et al., 2025; 2026), substantially enhancing the generalization capabilities of the generative policies. In such a head the action is not read out in one step: it is produced by integrating a learned velocity field from a noise sample to an action chunk over a time interval.

Although this paradigm significantly enhances the feasibility of real-world deployment, the transition still confronts substantial sim-to-real gaps (Zhao et al., 2026b), encompassing visual domain shifts, instruction misalignment, sensor noise, and spatial relationship perturbations (Jiang et al., 2023; Dalal et al., 2025; Xu et al., 2026). The vulnerability of pretrained policies to spatial perturbations is unsurprising. During the training phase, data collection and evaluation are typically restricted to fixed layouts, compelling the model to overfit to rigid behavioral patterns.

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.

This kind of spatial perturbation is a common condition of deployment: each time a policy is deployed it may face a layout that differs from the one it was trained on. Its effect on a pre-trained policy is substantial (e.g., success rate on LIBERO (Liu et al., 2023) drops from .990 to .247). Our goal is to **adapt the policy to the spatially shifted layout it is deployed in.** The deployment has a structure we can use. A fixed workstation does not present a new layout on every attempt: its displacement is drawn once at deployment and then held, and successive attempts differ only by a small residual jitter. We call such a layout a *site* (Figure 1). The open question here is how we can correct the policy for the spatial bias of its site using only the rollouts it produces there.

A direct approach is to fine-tune the policy on these rollouts. Fine-tuning, however, needs far more data than the few successes a site provides and compute that a local deployment rarely has, and it must be repeated at every site; reinforcement-learning fine-tuning (Yu et al., 2026; Xiao et al., 2026) is more expensive still, spending many attempts, most of them failures, before any improvement is seen. We therefore seek training-free adaptation (Zhao et al., 2026a; Patil et al., 2026), which uses no new demonstrations and no weight update, only what the policy itself produces at the site.

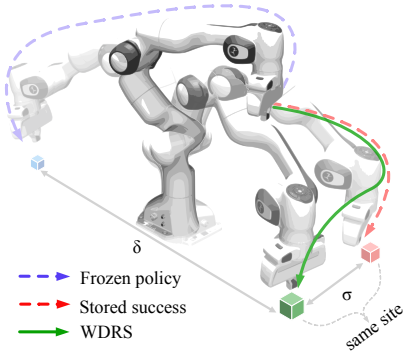


Figure 1: A site. Red and green differ by a jitter of scale σ and form one site; blue is displaced by $\delta \gg \sigma$ and is a different site.

With the weights frozen, an intervention can only enter through the sampler. Existing training-free methods for VLA policies each use one of its entry points. Most act at the start of the flow, on the initial noise vector (Wagenmaker et al., 2025; Simsir & Oguz, 2026; Patil et al., 2026); some (Zhao et al., 2026a) enters mid-flow with a retrieved stored chunk, but keeps the injection for the length of the stored trajectory, with no test of whether it still applies. A flow-matching sampler, however, has three entry points: the measure it starts from, the time at which it enters the flow, and the action it emits. We put the site's own success, a complete trajectory that solved this site's displacement and lies in the policy's own distribution, through all three, and add a rule for when to stop. The success is re-entered at an interior time and re-solved under the current observation, so that the output is again the policy's own draw, anchored to the success; the density defined by the same velocity field tells when the re-solved chunk is no longer one the policy would produce here, and control is returned to the policy; and a low-dimensional correction fitted from terminal outcomes alone is added at the output. Our primary contributions are as follows:

- We formalize the site-specific adaptation problem for spatial perturbations, and propose two interventions that between them occupy all three entry points of a chunked flow sampler: Warm-Started Denoising (Ws), which sets the starting measure and the entry time, and Action-Space Residual Search (RS), which corrects the emitted action.
- We design a Density Gate (Dg), which scores each re-solved chunk against what the policy would draw unaided under the current observation, and returns control the moment the two disagree. This replaces the stored success's own length with a decision read off the current attempt, and yields WDRS, a training-free adaptation framework for closed-loop control.
- We evaluate on two LIBERO suites under three jitter levels, where WDRS raises the frozen policy's success rate from .247 to .720 at the smallest jitter, and is the strongest of all six methods, including per-site weight fine-tuning, in five of the six suite-by-jitter cells. It ports unchanged in form to two further frozen flow policies (up to $+.474$ over the frozen policy), and to the Layout perturbations of LIBERO-plus ($+.290$) and RoboLab ($+.109$, Ws only).

2 Background and Setting

Chunked flow-matching policies. Recent work in robot learning has shifted from deterministic mappings to generative policies. It treats action prediction as conditional generation and can carry the multi-modality that a manipulation task actually has. To limit compounding error over a sequence, it is usually combined with *action chunking* (Zhao et al., 2023): instead of one action per

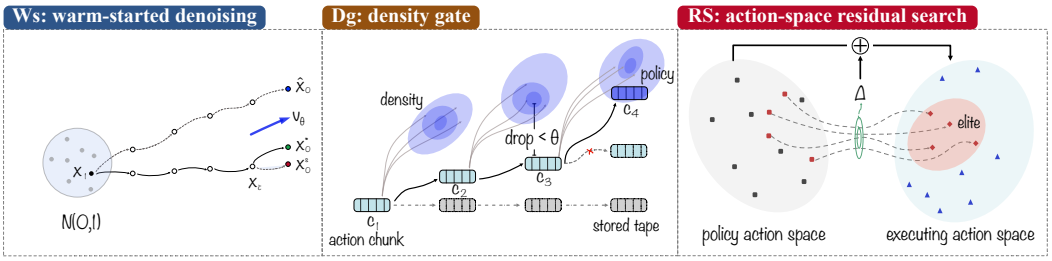


Figure 2: The three operators and the part of the sampler each one operates. **Left, Ws**: instead of integrating the whole flow from a draw of the base measure ($x_1 \to \hat{x}_0$, dashed), we re-enter the trajectory of a stored success at t_0 and integrate only the last few steps. **Middle, Dg**:each light-blue c_k is the chunk it produces under the current observation, scored against what the policy would draw on its own; at the chunk with $\mathrm{drop} < \theta_w$ the injected chunk is discarded (red cross) and the episode continues from the policy alone. **Right, RS**: a constant Δ carries the action the policy emits to the one actually executed, fitted across repeated attempts from terminal labels.

control step, the policy emits a whole chunk from observation o:

$$a \;=\; [\,a^{(1)}, a^{(2)}, \ldots, a^{(H)}\,] \;\in\; \mathbb{R}^{H \times d_a}, \tag{1}$$

which is then executed open loop for H control steps before the next observation is taken.

Among generative objectives, flow matching (Lipman et al., 2023) trains a time-dependent velocity field that transports a tractable base measure to the distribution of expert chunks. With the conditional path

$$x_t \;=\; (1-t)\,a \;+\; t\,\varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, I), \qquad t \in [0,1], \tag{2}$$

so that $t = 0$ is a chunk and $t = 1$ is noise, the network is trained to regress the velocity of that path, and sampling integrates the resulting ordinary differential equation backwards from $t = 1$. Two properties of this family matter here: the velocity field is *explicit*, so it can be evaluated at states the policy did not itself produce and in either direction of time; and sampling is an *integration*, so it can be entered part of the way along rather than only at its start.

The deployment: one site. Once deployed, the robot faces a fixed layout. Relative to the layout the policy was trained on, the n movable objects of the scene sit at displaced positions, and the displacement does not change from one attempt to the next; what does change between attempts is a small variation in how the objects are placed. We call the displaced layout a *site* w, with displacement δ_w, and the layout of one particular attempt, the pair (w, j), a *scene*, where j is that attempt's residual jitter. An attempt ends with a terminal outcome $y \in \{0, 1\}$, and $p_0(w) = \Pr[y = 1 \mid w]$ denotes the success rate of the frozen policy at the site, the probability being over the jitter and over the policy's own sampling noise. In our experiments $\delta_w \in \mathbb{R}^{2n}$ is a planar, per-object displacement drawn once and held fixed at the site, the jitter is $j \sim \mathcal{N}(0, \sigma^2 I_{2n})$ drawn independently per attempt, and δ_w is large relative to σ for every object. The pair (δ_w, σ) is the entire perturbation model.

What the method observes. Only the outcome y of each attempt: no reward, no demonstration, and no access to δ_w, j or σ. Every attempt spent on adaptation at the site counts against a fixed budget (72 in our experiments).

3 Methodology

This section details our proposed framework. Driven by the core motivation of repurposing prior successful rollouts, the subsequent three subsections address three questions regarding trajectory reuse: *1) Where should the sampler start?* (Ws, Sec. 3.2); *2) When to terminate the intervention?* (Dg, Sec. 3.3); *3) What actionable priors can be extracted from repeated attempts?* (RS, Sec. 3.4), as illustrated in Figure 2.

3.1 The policy sampler

Given a frozen policy π_θ, at chunk index k it receives an observation o_k and emits a chunk a_k as in equation 1, so an episode is determined by the sequence $(a_0, a_1, \ldots)$ and the environment dynamics.

The policy provides the velocity field $v_\theta(x, t; o)$ of equation 2, and we write $\Phi^o_{t\to t'}$ for the flow map of $\dot{x}_r = v_\theta(x_r, r; o)$, i.e. the solution operator carrying a state at time t to its image at time t'. Ordinary sampling integrates the whole interval from the base measure:

$$a_k = \Phi^{o_k}_{1\to 0}(\varepsilon), \quad \varepsilon \sim \mathcal{N}(0, I) \qquad \Longleftrightarrow \qquad a_k \sim \pi_\theta(\cdot \mid o_k) = \big(\Phi^{o_k}_{1\to 0}\big)_{\#}\mathcal{N}(0, I), \tag{3}$$

where $(\cdot)_{\#}$ denotes the pushforward of a measure. In practice $\Phi^o_{1\to 0}$ is discretised into $K = 10$ Euler steps, $x^{(i-1)} = x^{(i)} - \frac{1}{K} v_\theta(x^{(i)}, \frac{i}{K}, o_k)$ from $x^{(K)} = \varepsilon$, returning $a_k = x^{(0)}$.

A common form. Both interventions are a choice of a base measure ν_k, an entry time t_k into the flow, and a translation Δ of the executed chunk:

$$a_k = \Phi^{o_k}_{t_k\to 0}(\xi_k) + \beta_k \Delta, \qquad \xi_k \sim \nu_k. \tag{4}$$

with the frozen policy at $(\nu_k, t_k, \beta_k) = (\mathcal{N}(0, I), 1, 0)$. The velocity field, and the geometry the policy has learned with it, are identical in all arms. Constants are collected in Appendix A.2.

These three entries are the three arguments of a single sampler call, and each can be configured individually without structurally interfering with the other two. They are not the only way to act at fixed θ (e.g., editing the velocity field, see Appendix H), but they are the ones that leave the transport the policy learned untouched. Meanwhile, it's worth noting that during closed-loop execution, their practical impacts are not isolated but coupled.

3.2 Ws : Warm-Started Denoising

Where should the sampler start? Let $\bar{a} = (\bar{a}_0, \ldots, \bar{a}_{L-1})$ be the chunks executed on the first attempt at this site that terminated in success. While producing each $\bar{a}_k$ the sampler passed through a sequence of intermediate states on the Euler grid, and it is these that we store: the *tape* is the set of sampler trajectories $\{x^{\text{rec}}_{k,t}\}$, with $\bar{a}_k = x^{\text{rec}}_{k,0}$ as the endpoint of each. Neither the observations of that attempt nor any reward is stored, and the tape is the only object Ws carries to the next attempt.

The injection operator. For $k < L$, we replace the base measure of chunk k by a point mass at the state the recording's own sampler occupied at time t_0, and re-enter the flow there instead of at $t = 1$:

$$\nu_k = \delta_{x^{\text{rec}}_{k,t_0}}, \qquad t_k = t_0, \qquad \text{so that} \qquad a_k = \Phi^{o_k}_{t_0\to 0}\big(x^{\text{rec}}_{k,t_0}\big), \tag{5}$$

integrated for $m(t_0) = K - \text{round}\big((1 - t_0)K\big)$ Euler steps down to $t = 0$. For $k \geq L$ the tape is exhausted and the sampler reverts to equation 3. Because t_0 must be a time the recording actually visited, it ranges over the K points of the Euler grid and the operator is indexed by the integer m.

Two properties follow. *The stored chunk is not replayed*: the transport is generated by $v_\theta(\cdot, \cdot\,; o_k)$ under the *current* observation, and at recording time the same m steps from the same state reproduce $\bar{a}_k$ by construction, so $a_k \neq \bar{a}_k$ for exactly one reason, that the observation changed from o^{rec}_k to o_k (Proposition 3, Appendix B). The operator is a re-solution of the stored computation rather than a perturbation of its answer. *And the injected chunk carries no sampler randomness*: the pushforward of a point mass is a point mass, so given the tape and o_k the chunk is determined and all variation across attempts enters through the scene. Injected episodes are bit-reproducible up to the handoff, which is what makes the gate's statistic below a distribution over scenes rather than over scenes and sampler draws. At $t_0 \to 0$ the stored state is $\bar{a}_k$ itself and the operator degenerates to open-loop replay, which ignores o_k entirely; at $t_0 = 1$ it integrates the whole flow from the noise the recording was drawn from, which is not the frozen policy (that draws fresh noise) but a fixed-noise arm.

3.3 Dg: Density Gate

When to terminate the intervention? The tape's length L is a property of the *recording*: it says how long that one success happened to be, and carries nothing about whether, at chunk k of the attempt now running, the observation still agrees with the scene the tape was recorded in. The same tape helps on some attempts and hurts on others according to how far this attempt's jitter has carried the scene away from the recording, so the horizon must be replaced by a rule read off the current attempt. The gate is that rule, quantifying the tape's agreement with the scene.

A self-referential density statistic. The velocity field that generates actions also defines a log-density on chunks. Along the trajectory of $\dot{x}_t = v_\theta(x_t, t; o)$ the instantaneous change of variables

gives $\frac{\mathrm{d}}{\mathrm{d}t}\log p_t(x_t) = -\nabla\cdot v_\theta(x_t,t;o)$, so integrating from $t{=}0$ to $t{=}1$ and using the Gaussian law at the terminal time,

$$\log p_\theta(a \mid o) \;=\; \log \mathcal{N}(x_1;0,I) \;+\; \int_0^1 \nabla\cdot v_\theta(x_t,t;o)\,\mathrm{d}t, \qquad \dot{x}_t = v_\theta(x_t,t;o),\; x_0 = a. \tag{6}$$

The integral is discretised on the same Euler grid and the divergence is estimated by a Hutchinson trace estimator. The estimator runs the same v_θ that produces actions, in reverse, and takes no gradient with respect to θ.

The statistic is a typicality score in units of the policy's own entropy. At an injected chunk k, let a_k^{tape} be the chunk produced by equation 5 under o_k, and let $a_k^{(1)},\dots,a_k^{(J)}$ be J chunks drawn from the ordinary sampler equation 3 under the same o_k. The statistic is the drop

$$\mathrm{drop}_k \;=\; \log p_\theta\big(a_k^{\mathrm{tape}} \mid o_k\big) \;-\; \frac{1}{J}\sum_{j=1}^{J} \log p_\theta\big(a_k^{(j)} \mid o_k\big). \tag{7}$$

The subtracted term is a Monte-Carlo estimate of $-\mathcal{H}\big(\pi_\theta(\cdot \mid o_k)\big)$, so drop_k is the log-density of the injected chunk centred by the differential entropy of the conditional the policy currently holds. The centring is exact and assumes no model: a chunk drawn from that conditional scores zero in expectation, so the statistic has a fixed natural zero on any base policy and in any units (Appendix B). Two things follow. No reference to the recording scene is needed, because the comparison is drawn against the policy's own unforced behaviour at the same chunk under the same observation. And the centring is done afresh at each chunk, so the threshold below can be a single scalar per site rather than one number per chunk index.

The threshold is a rate, not a level. A fixed constant in nats does not transfer across sites. We therefore threshold on a *rate*. Let P be the pool of drop values of all chunks of the first n_{warm} injected episodes at this site (the warm-up), frozen thereafter, and $\widehat{F}_P$ its empirical distribution; fix $q \in (0,1)$. The threshold is the q-quantile of that pool,

$$\theta_w \;=\; \widehat{F}_P^{-1}(q), \qquad \text{and the handoff index is} \qquad \tau \;=\; \min\{\, k < L :\; \mathrm{drop}_k < \theta_w \,\} \wedge L. \tag{8}$$

Here τ is the index of the first unaided chunk: injection is withdrawn there, and the policy runs unaided for the rest of the episode, exactly as in the $k \geq L$ branch of equation 5. What needs to be set is a firing rate rather than a level in nats, so q is shared across sites while the θ_w it induces is not. The warm-up, the pool floor and the freezing of P are given in Algorithm 1.

3.4 RS: Action-Space Residual Search

What actionable priors can be extracted from repeated attempts? The answer is that the third operator sets the last slot of equation 4: a constant offset added to the action the sampler emits, fitted at one site from terminal labels alone. A constant can only cancel what is constant, so RS can pay only if the command carries a component that repeats across attempts (Proposition 1, Appendix B). That is what a site supplies and an i.i.d. perturbation does not: the displacement is drawn once and held, while the residual jitter is not.

Where the search acts. The intervention is a translation of the executed action,

$$a_k \;\longmapsto\; a_k + \beta_k\,\Delta, \qquad \Delta \in \mathbb{R}^{d_a}. \tag{9}$$

Here, Δ is a single vector drawn once at the start of an attempt, held constant across every step of it, and added to each action in the chunk. **Every executed action is translated**, including the chunks the tape still drives. β_k is the third slot of equation 4. If the intervention is confined to the subset $\{k \geq \tau\}$, achieved by setting $\beta_j = 0$ for all $j < \tau$, the routing decision of the gate remains unaffected. By default, we set $\beta_k \equiv 1$, which alters the behavior of Dg.

What kind of object Δ is. The translation post-composes the sampler with a fixed map, so the executed law is the pushforward of the policy's own conditional under $x \mapsto x + \Delta$: every attainable command remains one the policy could itself have produced, displaced by a known constant, and no transport map is edited. Because the policy re-observes the displaced state and corrects for it, a constant command offset does not accumulate as drift, even though within a chunk it is applied open loop.

Search by the cross-entropy method. Δ is fitted by a cross-entropy method run across repeated attempts at the site, maintaining a mean μ and an isotropic scale s, initialised at $\mu = 0$ and $s = s_0$.

Table 1: Comparison of success rates varying with σ across different suites

Method	libero_object ($G^{\dagger}$=10)			libero_spatial (G=9)		
	σ=0.5	σ=1	σ=2	σ=0.5	σ=1	σ=2
Frozen policy	.247	.275	.300	.581	.520	.567
Open-loop replay	.233 (−.013)	.227 (−.048)	.053 (−.247)	.296 (−.285)	.259 (−.261)	.126 (−.441)
Retrieve-then-steer (RTS)*	.107 (−.140)	.113 (−.162)	.280 (−.020)	.437 (−.144)	.474 (−.046)	.348 (−.219)
Golden ticket	.367 (+.120)	.373 (+.098)	.313 (+.013)	.578 (−.004)	.548 (+.028)	.548 (−.019)
Latent-noise search	.540 (+.293)	.453 (+.178)	.400 (+.100)	.637 (+.056)	.593 (+.072)	.578 (+.011)
Per-site LoRA	.247 (+.013)	.373 (+.073)	.380 (+.113)	.593 (−.015)	.563 (+.044)	**.652** (+**.067**)
WDRS (ours)	**.720** (+**.473**)	**.687** (+**.412**)	**.467** (+**.167**)	**.704** (+**.122**)	**.644** (+**.124**)	.548 (−.019)

† G is the number of sites. * RTS is reported at the stronger of two memory sizes.

Attempts are drawn in generations of B; within a generation, attempt i draws

$$\Delta_i \;=\; \mu + s \cdot z_i, \qquad z_i \sim \mathcal{N}(0, I_6), \qquad \text{clipped to } \|\Delta_i\|_\infty \le c. \tag{10}$$

Every attempt executes Δ from its first chunk, so the *elite* set of a generation is simply its successful attempts, and the update is

$$\mu \;\leftarrow\; \frac{1}{|\text{elite}|} \sum_{i \in \text{elite}} \Delta_i, \qquad s \;\leftarrow\; \max(\gamma \cdot s,\, s_{\min}), \tag{11}$$

with $\gamma < 1$; a generation whose elite set is empty leaves both μ and s unchanged. This is the usual cross-entropy step, the new mean is the M-projection of the elite empirical measure onto the sampling family, with a floor $s_{\min} > 0$ on the scale so that later generations can still leave a bad mean. What the search maximises is $\Pr[\, y = 1 \mid \Delta \,]$, estimated from terminal labels; the variant that withholds Δ until the handoff needs a different conditioning for that estimand, and is in Appendix E.

Synthesis. When deployed jointly at a specific site, these three mechanisms simultaneously populate the three arguments in Eq. equation 4: the tape supplies the base measure and the entry time until Dg withdraws it at τ, and RS translates every executed chunk throughout. A unifying characteristic of these interventions is that each moves the *location* of the chunk distribution and holds it moved: where sampling starts, when it starts, or what is finally executed, and none of them edits the field that transports it. The criterion is location rather than spread.

By integrating these operators, we obtain **WDRS** (**W**arm-started denoising with **D**ensity gate and **R**esidual **S**earch in action space), a closed-loop adaptation framework. The full procedure is detailed in Algorithm 1, which outlines the boarding attempts that earn the tape, the warm-up that fixes θ_w, and the generations over which Δ is fitted.

The supervision consumed. The method consumes terminal labels and nothing else: one label to accept the tape, and one label per attempt to form the elite set. No reward model, no value function, no intermediate success predicate and no demonstration enters the loop, and model parameter θ is never updated.

4 EXPERIMENTS

Setup. As a plug-and-play adaptation module, WDRS is compatible with most chunked flow-matching policy backbones. We conduct experiments utilizing three mainstream architectures: $\pi_{0.5}$, GR00T N1.7 and π_{RL}. For each architecture, we adopt the respective suite-specific fine-tuned variant as our foundational base policy.

Benchmarks. We evaluate on two task suites of the LIBERO benchmark (Liu et al., 2023) (capped at a 300-step horizon). While fine-tuned base models achieve $> 95\%$ SR on the original layouts (Appendix C), we simulate spatial displacement scenarios by setting $\delta = 10$ cm for all scene objects, alongside varying local jitter levels, $\sigma \in \{0.5, 1, 2\}$ cm. Additionally, we conducted evaluations on the Layout axis of LIBERO-Plus (Fei et al., 2025), which is a standardized spatial displacement benchmark, and RoboLab (Yang et al., 2026), a dedicated framework based on IsaacSim (NVIDIA, 2025) for evaluating generalization. During adaptation, each site is constrained to a 72-attempt budget, and final performance is aggregated over 15 evaluation rollouts. At evaluation the site runs WDRS with the tape, the frozen threshold θ_w and the final search mean $\Delta = \mu$, on scenes

Table 2: Evaluation of the portability across multiple base models.

Base policy	H / K	t_0	δ	σ	med. θ_w	Frozen policy	Open-loop replay	WDRS
$\pi_{0.5}$	16 / 10	0.1	10 cm	1	−435	.275	.227 (−.048)	**.687** (+**.412**)
GR00T N1.7	16 / 4	0.25	6 cm	1	−458	.311	.104 (−.207)	**.715** (+**.404**)
π_{RL}	10 / 10	0.5	10 cm	1	−48	.348	.215 (−.133)	**.481** (+**.133**)

Table 3: Performance evaluation on Libero-Plus Layout and RoboLab

	Frozen policy	Open-loop replay	WDRS
`libero_goal` × `Layout` (G=31)	.455	.419 (−.035)	**.745** (+**.290**)
`RoboLab` (σ=2.0, G=8)	.500	.047 (−.453)	.609† (+.109)

† Ws only.

disjoint from the search. Since the robots in these benchmarks are 7-DoF, Δ is confined to a 6-dimensional search space without the gripper. Full hyperparameters and the complete loop are given in Appendix A.

Baselines. 1) *Open-loop replay*: Executes the cached successful trajectory as a rigid, fixed sequence of actions, entirely blinding the policy to current perceptual observations. This represents a degenerate case of Ws. 2) *Retrieve-then-steer* (Zhao et al., 2026a): Intervenes along the same architectural axis as Ws. It leverages DTW to retrieve and dynamically re-inject stored action chunks based on state similarity. 3) *Golden ticket* (Patil et al., 2026): Searches for and reuses a single, optimal initial sampling noise vector that yields a successful rollout in the current site. 4) *Latent-noise search*: Serves as the counterpart to RS, but operates within the sampler's latent noise space, it optimises a constant translation of the initial noise ε. 5) *Per-site LoRA*: A weight-updating baseline that fine-tunes the base policy using its own successful rollouts collected at each specific site.

Main results. As shown in Table 1, across the six jitter-injected evaluation suites, WDRS outperforms other baselines in five of them. Specifically, under minor to moderate perturbations ($\sigma = 0.5$ and 1), WDRS raises the success rate by more than .40 in the Object suite and .12 in the Spatial suite. Notably, the relative performance gain diminishes as the spatial jitter intensifies. At a larger perturbation level ($\sigma = 2$), the utility extracted from the "tape" becomes marginal; in the Spatial suite, it even underperforms the base policy. When the base policy already exhibits high intrinsic robustness (i.e., a high SR), the marginal utility provided by trajectory reuse becomes limited.

It is important to clarify that due to the existence of jitter, action chunks retrieved via observational state similarity originate from a spatial layout that is misaligned with the geometry of the current execution scene. RTS is therefore reported with its memory limited to the single tape our method boards; given our 72 attempts to record instead, its memory grows to 1–68 tapes drawn from different jitters and collapses to .000–.111.

Portability across base policies. We re-established the whole construction on two further frozen flow policies (Table 2). No operator was modified for any of them: the same injection, density statistic, and search. Since the denoising horizons and execution steps vary across foundational models, we recalibrate the integration starting point t_0 accordingly, while holding all other hyperparameters constant. As demonstrated in Table 2, WDRS yields substantial performance gains (ranging from $+.133$ to $+.412$) across disparate backbones, evidencing its cross-architecture portability. In fact, while Ws and the Dg are structurally tailored to FM formulations, RS is model-agnostic and does not mandate an FM backbone.

Notably, the empirical median of the density gate threshold θ_w fluctuates significantly across different models. This divergence validates our algorithmic design choice to define the gating threshold via statistical percentiles rather than absolute values.

A benchmark that supplies its own displacement. The sites above are built by displacing a suite's objects ourselves. LIBERO-plus ships a `Layout` axis that does it instead, and we ran the method on the 31 `Layout` variants of `libero_goal` on which it boards a tape and the frozen policy is below ceiling (78 of 185 board; on 47 of those the policy already scores 1.000). This axis displaces every object independently, in five graded levels, so it carries no displacement shared across objects and no jitter of ours, and the method still moves the success rate from .455 to .745 (Table 3). We also validated WDRS on RoboLab by using a DROID (Khazatsky et al., 2025) fine-tuned $\pi_{0.5}$ model.

Table 4: Ablation study on $\pi_{0.5}$ and GR00T N1.7

	$\pi_{0.5}$ (δ=10 cm, G=10)			GR00T N1.7 (δ=6 cm, G=9)		
	σ=0.5	σ=1	σ=2	σ=0.5	σ=1	σ=2
Frozen policy	.247	.275	.300	.319	.311	.341
+ Ws	.433 (+.187)	.373 (+.098)	.160 (−.140)	.333 (+.015)	.281 (−.030)	.230 (−.111)
+ Dg	.473 (+.040)	.393 (+.020)	.287 (+.127)	.415 (+.081)	.341 (+.059)	.281 (+.052)
+ RS = WDRS	**.720** (+**.247**) (+.473)	**.687** (+**.293**) (+.412)	**.467** (+**.180**) (+.167)	**.793** (+**.378**) (+.474)	**.715** (+**.374**) (+.404)	**.400** (+**.119**) (+.059)

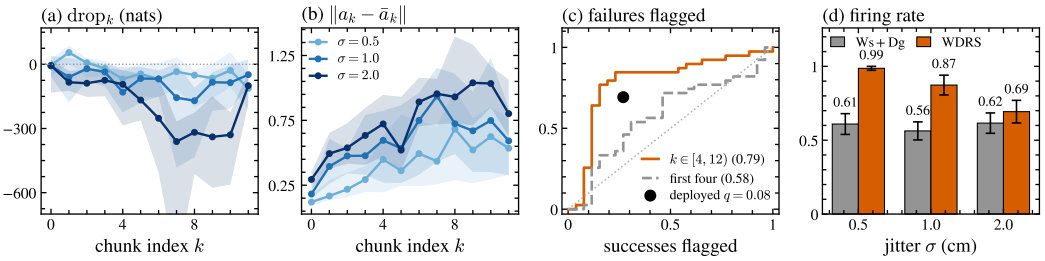


Figure 3: **(a)** drop$_k$ and **(b)** how far the executed chunk has left the tape chunk it was started from, along the episode. **(c)** how well $\min_{k\in[4,12)} \text{drop}_k$ orders episodes by whether they fail with no gate, against the first four chunks as a placebo; the deployed point is off-curve because the threshold is per-site and the curve one pooled ranking. **(d)** fraction of episodes in which the gate fires at all.

Since this OOD scenario lacks an explicit train-deploy spatial shift, leaving no systematic geometric bias to be learned, we deploy Ws alone ($t_0 = 0.1$), which yields $+.109$.

Ablation study. We conducted comprehensive ablation studies across two base policies, with the results detailed in Table 4. An intuitive observation is that the efficacy of rollout injection degrades as the spatial jitter σ intensifies, even yielding negative transfer (-0.140 at $\sigma = 2$). This outcome is theoretically sound, given that the spatial validity of a successful tape is inherently bounded. The Ws serves primarily as a reference for the Dg, explicitly delineating the trust region for adaptation. Meanwhile, Dg effectively mitigates the limitations of tape reuse. By handing control back to the base policy, Dg ensures consistent positive gains over the policy up to $\sigma = 1$. RS delivers the most substantial improvements, achieving an SR boost between $+.247$ and $+.378$ within the $\sigma \leq 1$ regime. Mechanistically, while the Ws+Dg combination requires only a single successful tape to bootstrap improvement, RS iteratively accumulates optimization benefits across multiple attempts.

Notably, the efficacy of RS also decays while $\sigma = 2$. This indicates that its optimization landscape is similarly bounded; excessive spatial jitter renders the constant spatial offset difficult to fit reliably. The last rung is not a pure increment. RS is supported on every chunk, so it displaces the observation Dg scores under, and the firing rate rises accordingly (Fig. 3(d); the two arms there are not paired). A higher success rate can therefore arise either from a better Δ or from more of the tape being surrendered, and the outcome alone does not separate them. Appendix E defines the variant whose effect is a pure increment, and explains why we deploy the other one.

Drift of the gate statistic. Figure 3(a)–(c) read the per-chunk records of the ungated warm-up episodes, and (d) those of the gated arms. Along an episode drop$_k$ falls and its dispersion grows, and the executed chunk diverges from the tape chunk it was started from. The base measure is a Dirac at a stored path point and is identical on every attempt, so this variation can enter only through the conditioning observation. The validity of a stored chunk therefore decays with the chunk index, which is what a stopping rule is for. Figure 3 (a)–(c) use episodes recorded before the gate is armed, so that no record is truncated at its own firing. Appendix D gives the filters and the measurements; Appendix F gives the alternative of deciding before the episode begins.

Which episodes the rule selects. We also examined whether the rule identifies the episodes that need the intervention. Warm-up episodes recorded before the gate is armed run the injection to the end of the tape, so their outcomes are those of an ungated run; applying the site's threshold to them afterwards labels each episode by whether the gate would have fired. Among 65 such episodes, those that would fire failed at a rate of .794, against .387 for those that would not. The gate is therefore triggered on the episodes that were going to fail.

5 Related work

Generative policies have rapidly established themselves as the dominant paradigm in robotic learning. They have transitioned from autoregressive architectures, such as OpenVLA (Kim et al., 2025), toward recent diffusion (Chi et al., 2023; 2024) or flow matching (Lipman et al., 2023) based models, which chunked generative policy produces a short action sequence, such as π_0 (Black et al., 2026), $\pi_{0.5}$ (Intelligence et al., 2025), GR00T N1 (NVIDIA et al., 2025) and SmolVLA (Shukor et al., 2025). While these models leverage massive datasets to acquire broad, general-purpose capabilities during pre-training, their deployment in specific environments necessitates targeted adaptation to effectively resolve domain-specific challenges.

Post-training adaptation. One family improves the policy by updating its weights: online reinforcement learning on a flow VLA (Chen et al., 2026); residual actors that probe the generalist's failure regions and are distilled back by supervised fine-tuning (Xiao et al., 2026); reinforcement learning requires reward design and simulation support (Yu et al., 2026); and test-time training of fast weights from long context and human video (Jiang et al., 2026). These approaches suffer from sample and computational inefficiencies. They rely on human expert demonstrations or massive online interaction budgets, and also necessitate burdensome gradient computations.

Training-free test-time adaptation. A second family leaves θ fixed and intervenes in the sampler. Latent-space reinforcement learning optimises a policy over the sampling noise (Wagenmaker et al., 2025), and LP-DS constrains that perturbation to remain near the latent prior (Simsir & Oguz, 2026); both require a return. The golden ticket searches a single noise vector and reuses it across episodes, from terminal outcomes alone (Patil et al., 2026). All three act at $t{=}1$, where the available object is a noise vector, whereas Ws acts at an interior time, where it is a state the policy itself passed through on a success. LP-DS keeps the perturbation admissible by a penalty in noise space; Dg instead compares the resulting chunk, in action space, with what the policy would draw unaided, and responds by stopping rather than by penalising. Retrieve-then-steer writes the same argument, re-injecting stored successes retrieved by state similarity (Zhao et al., 2026a); it models failure as perception noise about a competent mean, so reuse acts as variance reduction. Q-Planning leaves the sampler untouched and fine-tunes only a Q-function, which arbitrates the policy's own draws (Giridhar et al., 2026).

Diverging from continuous intervention, WDRS applies localized steering via an explicit handoff mechanism, seamlessly relinquishing control to the base policy mid-episode. Furthermore, it operates purely on the sparse binary outcome of a trial, eschewing dense rewards, value networks, or demonstrations. Mechanistically, both interventions move the location of the chunk distribution rather than its spread: re-entering from a stored success starts the sampler on a trajectory that solved this site, and the fitted offset cancels the component of the required correction that persists across attempts.

6 Discussion and limitations

We have presented WDRS, a training-free procedure that adapts a frozen chunked flow policy to a fixed spatial displacement by acting only on its sampler. It re-enters the denoising trajectory at a path point stored from one of the policy's own successes, withdraws that injection once the injected chunk becomes atypical under the policy's own density, and adds a constant offset to the executed actions, fitted at the site from terminal outcomes alone. Across three base policies, two LIBERO suites it is the strongest method in five of six cells and raises the success rate by $+.412$, $+.404$ and $+.133$ at $\sigma = 1$ on the three policies; on the `Layout` axis of LIBERO-plus, it raises it from $.455$ to $.745$. These results demonstrate the effectiveness of the construction and its plug-and-play character.

Limitations. *A success is required.* The method starts from one of the policy's own successes, so a site at which the policy never succeeds is outside it by construction. *The gain is confined to small jitter.* At $\sigma = 2\,\text{cm}$ the tape alone is harmful on both base policies, and the full method retains only part of its gain, or falls below the frozen policy. *The gain depends on the tape.* Everything Ws and Dg do is conditioned on the single recorded success they are given, and we do not select it: the tape is whichever attempt succeeded first. Selecting which success to keep, fitting the offset by something better than a search over terminal outcomes, and transferring either across sites are left to future work.

AI USE STATEMENT

In this work, we used generative AI tools for assisting with translation and code writing. We have not used generative AI tools for developing theoretical models or conceptual frameworks, literature reviewing, or experimental design.

REPRODUCIBILITY STATEMENT

All datasets used are publicly available. We include the core pseudo-codes and most implementation details in the paper. In addition, we commit to releasing the code in the future.

REFERENCES

T. W. Anderson. The integral of a symmetric unimodal function over a symmetric convex set and some probability inequalities. *Proceedings of the American Mathematical Society*, 6(2):170–176, 1955.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. Paligemma: A versatile 3b vlm for transfer, 2024. URL https://arxiv.org/abs/2407.07726.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2026. URL https://arxiv.org/abs/2410.24164.

Kang Chen, Zhihao Liu, Tonghe Zhang, Zhen Guo, Si Xu, Hao Lin, Hongzhi Zang, Xiang Li, Quanlu Zhang, Zhaofei Yu, Guoliang Fan, Tiejun Huang, Yu Wang, and Chao Yu. π_{RL}: Online rl fine-tuning for flow-based vision-language-action models, 2026. URL https://arxiv.org/abs/2510.25889.

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

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*, 2024.

Murtaza Dalal, Min Liu, Walter Talbott, Chen Chen, Deepak Pathak, Jian Zhang, and Ruslan Salakhutdinov. Local policies enable zero-shot long-horizon manipulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13875–13882, 2025. doi: 10.1109/ICRA55743.2025.11128407.

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: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Varun Giridhar, Anant Khandelwal, Jeremy A. Collins, Ignat Georgiev, and Animesh Garg. Beyond imitation: Self-improving robot policies via off-policy q-planning, 2026.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

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

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, Vedant Choudhary, Foster Collins, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Maitrayee Dhaka, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachlan Groom, Haroun Habeeb, Hunter Hancock, Karol Hausman, Gashon Hussein, Victor Hwang, Brian Ichter, Connor Jacobsen, Szymon Jakubczak, Rowan Jen, Tim Jones, Gregg Kammerer, Ben Katz, Liyiming Ke, Mairbek Khadikov, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Brendon LeCount, Sergey Levine, Xinyu Li, Adrian Li-Bell, Vladislav Lialin, Zhonglin Liang, Wallace Lim, Yao Lu, Enyu Luo, Vishnu Mano, Nandan Marwaha, Aikys Mongush, Liam Murphy, Suraj Nair, Tyler Patterson, Karl Pertsch, Allen Z. Ren, Gavin Schelske, Charvi Sharma, Baifeng Shi, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Jiaming Tang, Jimmy Tanner, Shalom Tekeste, Marcel Torne, Kyle Vedder, Quan Vuong, Anna Walling, Haohuan Wang, Jason Wang, XuDong Wang, Chris Whalen, Samuel Whitmore, Blake Williams, Charles Xu, Sukwon Yoo, Lili Yu, Wuming Zhang, Zhuoyang Zhang, and Ury Zhilinsky. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities, 2026. URL https://arxiv.org/abs/2604.15483.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: robot manipulation with multimodal prompts. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Yunfan Jiang, Yevgen Chebotar, Ruijie Zheng, Fengyuan Hu, Yunhao Ge, Jimmy Wu, Tianyuan Dai, Scott Reed, Li Fei-Fei, Yuke Zhu, and Linxi "Jim" Fan. Robottt: Context scaling for robot policies, 2026. URL https://arxiv.org/abs/2607.15275.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, Vitor Guizilini, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Muhammad Zubair Irshad, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. Droid: A large-scale in-the-wild robot manipulation dataset, 2025. URL https://arxiv.org/abs/2403.12945.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of

Proceedings of Machine Learning Research, pp. 2679–2713. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning, 2023. URL https://arxiv.org/abs/2306.03310.

NVIDIA. Isaac Sim, 2025. URL https://github.com/isaac-sim/IsaacSim.

NVIDIA, Johan Bjorck, Nikita Cherniadev Fernando Castañeda, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots. In *ArXiv Preprint*, March 2025.

Omkar Patil, Ondrej Biza, Thomas Weng, Karl Schmeckpeper, Wil Thomason, Xiaohan Zhang, Kausik Sivakumar, Robin Walters, Nakul Gopalan, Sebastian Castro, Stephen Hart, and Eric Rosen. You've got a golden ticket: Improving generative robot policies with a single noise vector, 2026. URL https://arxiv.org/abs/2603.15757.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. Smolvla: A vision-language-action model for affordable and efficient robotics, 2025. URL https://arxiv.org/abs/2506.01844.

Hikmet Simsir and Ozgur S. Oguz. Lagrangian perturbation diffusion steering: Latent reinforcement learning for generative policies. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=HsdgKuOV8n.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

Andrew Wagenmaker, Mitsuhiko Nakamoto, Yunchu Zhang, Seohong Park, Waleed Yagoub, Anusha Nagabandi, Abhishek Gupta, and Sergey Levine. Steering your diffusion policy with latent space reinforcement learning. *Conference on Robot Learning (CoRL)*, 2025.

Wenli Xiao, Haotian Lin, Andy Peng, Haoru Xue, Tairan He, Zhengyi Luo, Yuqi Xie, Fengyuan Hu, Linxi Fan, Guanya Shi, and Yuke Zhu. Self-improving vision-language-action models with data generation via residual RL. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=eUGoqrZ6Ea.

Kechun Xu, Zhenjie Zhu, Anzhe Chen, Rong Xiong, and Yue Wang. Apt: Action expert pretraining improves instruction generalization of vision-language-action policies, 2026. URL https://arxiv.org/abs/2606.12366.

Jenai Xuning Yang, Rishit Dagli, Alex Zook, Hugo Hadfield, Ankit Goyal, Stan Birchfield, Fabio Ramos, and Jonathan Tremblay. Robolab: A high-fidelity simulation benchmark for analysis of task generalist policies, 2026. URL https://arxiv.org/abs/2604.09860.

Raymond Yu, William Huey, Mustafa Mukadam, Anusha Nagabandi, and Abhishek Gupta. Support-constrained rl enables real-world policy improvement without real-world experience, 2026. URL https://arxiv.org/abs/2606.27475.

Jianchao Zhao, Huoren Yang, Yusong Hu, Yuyang Gao, Qiguan Ou, Cong Wan, SongLin Dong, Zhiheng Ma, and Yihong Gong. Retrieve-then-steer: Online success memory for test-time adaptation of generative vlas, 2026a. URL https://arxiv.org/abs/2605.10094.

Runyi Zhao, Sheng Xu, Ruixing Jin, Yueci Deng, Yunxin Tai, Kui Jia, and Guiliang Liu. Sim2real VLA: Zero-shot generalization of synthesized skills to realistic manipulation. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=H4SyKHjd4c.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023. URL https://arxiv.org/abs/2304.13705.

A METHOD DETAILS

This appendix makes explicit what Sec. 3 describes one operator at a time: the order in which boarding, the warm-up and the search run at a site, and the constants each step carries.

A.1 PSEUDO-CODE

Algorithm 1 Warm-started denoising + density gate + action-space residual search, at one site. π_θ is frozen; no parameter of it is updated. Written at the deployed setting, in which every executed chunk is translated.

input firing rate q, pool floor $P_{\min}$, warm-up length n_{warm}, steps m, clip c; CEM mean $\mu \leftarrow 0$, scale $s \leftarrow s_0$, shrink γ, floor $s_{\min}$, generation size B
$\bar{a} \leftarrow \varnothing$ ▷ tape
while $\bar{a} = \varnothing$ **do** ▷ boarding: the site is run until it succeeds once
    roll out an unmodified episode; **if** it succeeds **then** $\bar{a} \leftarrow$ its chunk sequence, $\{x^{\text{rec}}_{k,t}\} \leftarrow$ the sampler states that produced it, and $L \leftarrow |\bar{a}|$
end while
$P \leftarrow \varnothing$; $\theta_w \leftarrow \varnothing$
for $j = 1, \dots, n_{\text{warm}}$ **do** ▷ fill the pool; injected episodes with $\Delta = 0$
    if $|P| \geq P_{\min}$ **then** $\theta_w \leftarrow \widehat{F}_P^{-1}(q)$ ▷ equation 8; while $\theta_w = \varnothing$ the gate cannot fire
    run lines 16–29 with $\Delta = 0$, appending every drop_k it computes to P *once the episode has ended*
end for
freeze P, and with it θ_w ▷ the pool does not grow during the search
for generation $g = 1, \dots, N_{\text{gen}}$ **do**
    elite $\leftarrow \varnothing$
    for $i = 1, \dots, B$ **do**
        draw $\Delta_i = \mu + s\, z_i$, clipped to $\|\Delta_i\|_\infty \leq c$ ▷ equation 10
        $k \leftarrow 0$
        while $k < L$ **do**
            produce $a_k = \Phi^{o_k}_{t_0 \to 0}\big(x^{\text{rec}}_{k,t_0}\big)$ by integrating m steps from the stored path point ▷ equation 5
            compute drop_k ▷ equation 7
            if $\theta_w \neq \varnothing$ **and** $\text{drop}_k < \theta_w$ **then**
                break ▷ the gate fires; a_k is discarded
            end if
            execute $a_k + \Delta_i$; $k \leftarrow k + 1$
        end while
        $\tau \leftarrow k$
        for $k = \tau, \tau + 1, \dots$ until the episode ends **do**
            produce a_k from the ordinary sampler equation 3 and execute $a_k + \Delta_i$ ▷ equation 9
        end for
        observe the outcome $y_i \in \{0, 1\}$
        if $y_i = 1$ **then** ▷ every attempt is acted, so the outcome is the whole criterion
            elite $\leftarrow$ elite $\cup \{\Delta_i\}$
        end if
    end for
    if elite $\neq \varnothing$ **then**
        update μ, s ▷ equation 11
    end if
end for

A.2 CONSTANTS

We list every constant the method carries together with what it governs. The first three groups belong to the three operators of Sec. 3; the fourth is the evaluation protocol.

	Value	What it governs
Warm-started denoising (Sec. 3.2)		
t_0	0.1	Where on the conditional path the sampler re-enters. It reaches the sampler only through $m = K - \mathrm{round}((1 - t_0)K)$, so with $K = 10$ it means one Euler step.
K	10	Euler steps of a full draw.
H	16	Control steps executed per chunk.
recording cap	60	Attempts allowed to earn a tape.
Density gate (Sec. 3.3)		
q	0.08	The firing rate, and the only free constant of the rule: the threshold is the q-quantile of the site's own pool, so q is shared across sites while the level it induces is not.
pool minimum	40	Chunks; below it θ_w is undefined and the gate never fires.
n_{warm}	10	Injected episodes that fill the pool, after which it is held fixed so that gate against the same θ_w.
J	2	Policy draws per chunk used to centre equation 7 by the conditional entropy.
Hutchinson probes	1	Probe vectors per divergence estimate.
difference step	10^{-2}	The Jacobian–vector product in the trace estimator is taken by central differences; this is its step.
reverse steps	50	Integration steps of equation 6.
Action-space residual search (Sec. 3.4)		
α	1	Weight of Δ on the chunks still driven by the tape. $\alpha = 1$ is the deployed setting and puts Δ on every executed chunk.
s_0	0.02	Initial exploration scale of the search, per coordinate, in the units of the pose command.
$s_{\min}$	0.006	Floor on that scale, so the sampling family stays non-degenerate and a later generation can still leave a bad mean.
γ	0.85	Geometric annealing per generation.
c	0.06	Clip on $\|\Delta_i\|_\infty$.
B	15	Attempts per generation.
N_{gen}	4	Search generations.
Evaluation protocol (not part of the method)		
held-out episodes	15	Reported per site, on scenes disjoint from the search.
episode cap	300	Maximum steps of each episode.
δ, σ	—	Site displacement 10 cm; residual jitter $\sigma \in \{0.5, 1, 2\}$ cm.

B PROOFS

This appendix holds the propositions behind Secs. 3.2 and 3.4, extends the first from a surrogate loss to the success probability itself, and derives the natural zero of the gate statistic that Sec. 3.3 relies on.

Proposition 1 (a constant buys nothing without a persistent bias). *Let C be the correction an attempt at the site would require, a random vector over attempts, and let ℓ be convex with $\ell(x) = \ell(-x)$. If C and $-C$ have the same law, then $\Delta \mapsto \mathbb{E}\,\ell(C - \Delta)$ is minimised at $\Delta = 0$. For $\ell(x) = \|x\|^2$ it is enough that $\mathbb{E}[C] = 0$, and then $\mathbb{E}\|C - \Delta\|^2 = \mathbb{E}\|C\|^2 + \|\Delta\|^2$.*

Write $f(\Delta) = \mathbb{E}\,\ell(C - \Delta)$. It is convex in Δ, being an expectation of convex functions of an affine argument. It is also even: $f(-\Delta) = \mathbb{E}\,\ell(C + \Delta) = \mathbb{E}\,\ell(-C - \Delta) = \mathbb{E}\,\ell(C - \Delta) = f(\Delta)$, using the symmetry of ℓ and then that C and $-C$ have the same law. A convex even function on $\mathbb{R}^d$ attains its minimum at the origin, since $f(0) = f(\frac{1}{2}\Delta + \frac{1}{2}(-\Delta)) \le \frac{1}{2}f(\Delta) + \frac{1}{2}f(-\Delta) = f(\Delta)$. For the quadratic case, expanding gives $\mathbb{E}\|C - \Delta\|^2 = \mathbb{E}\|C\|^2 - 2\langle \mathbb{E}[C], \Delta\rangle + \|\Delta\|^2$, which reduces to $\mathbb{E}\|C\|^2 + \|\Delta\|^2$ when $\mathbb{E}[C] = 0$. □

Corollary 1.1 (the persistent share). *Write $C = b_{\mathrm{persist}} + b_{\mathrm{episode}}$ with $b_{\mathrm{persist}} = \mathbb{E}[C]$ fixed across attempts and $\mathbb{E}[b_{\mathrm{episode}}] = 0$. For $\ell(x) = \|x\|^2$ the minimiser of $\Delta \mapsto \mathbb{E}\,\ell(C - \Delta)$ is $\Delta^\star = b_{\mathrm{persist}}$, and the loss it removes is $\|b_{\mathrm{persist}}\|^2$; no constant acts on b_{episode}.*

Expanding as in the proof of Proposition 1, $\mathbb{E}\|C-\Delta\|^2 = \mathbb{E}\|C\|^2 - 2\langle \mathbb{E}[C], \Delta\rangle + \|\Delta\|^2$, which is minimised at $\Delta = \mathbb{E}[C]$ with value $\mathbb{E}\|C\|^2 - \|\mathbb{E}[C]\|^2 = \mathbb{E}\|b_{\text{episode}}\|^2$. A constant offset can therefore absorb only the component of the required correction that persists across attempts at the site — the component a persistent displacement induces — and leaves the per-attempt component to the re-solved tape. □

Proposition 2 (switching is not selection). *Let A and B be two policies run under common random numbers. Call a rule a* selector *if it commits, before the episode starts, to running one of them throughout, using any pre-episode information and any randomisation.* ***(a)*** *Every selector succeeds with probability at most $\Pr[\,A \text{ succeeds} \;\vee\; B \text{ succeeds}\,]$.* ***(b)*** *The switching family π_c — follow A on chunks $k < c$ and B on $k \geq c$ — contains members that exceed that bound; there are tasks on which some π_c succeeds with probability one while the bound is zero.*

(a) Fix the random numbers. A selector runs A throughout or runs B throughout, so on every realisation the event that it succeeds is contained in the union of the event that A succeeds and the event that B succeeds; pre-episode information and randomisation choose *which* of the two is run, and so cannot leave that union. Taking probabilities gives the bound, and it is attained only by a selector that picks a succeeding arm whenever one exists.

(b) Two chunks suffice. Let an episode consist of chunks 0 and 1 and let it succeed exactly when both are executed correctly. Let A be correct at chunk 0 and wrong at chunk 1, and B the reverse, both deterministically. Then A fails and B fails, so the bound of (a) is zero, while π_1 — A on chunk 0, B from chunk 1 — succeeds with probability one. The construction uses nothing about flow matching; what it uses is that the episode has an interior, and that the two policies are good on different parts of it. □

Success rate, without the surrogate. Proposition 1 is stated for a convex loss and success rate is not one, so as it stands it governs a surrogate for the correction rather than $\Pr[y = 1 \mid \Delta]$. Anderson's inequality closes that gap under two further assumptions. Suppose the law of C has a density f that is symmetric about the origin and unimodal, in the sense that every superlevel set $\{f \geq u\}$ is convex; and suppose an attempt succeeds exactly when the residual $C - \Delta$ falls in a tolerance region K that is convex and symmetric about the origin. Substituting $x = c - \Delta$,

$$\Pr[\,C - \Delta \in K\,] \;=\; \int_K f(x+\Delta)\,\mathrm{d}x \;\leq\; \int_K f(x)\,\mathrm{d}x \;=\; \Pr[\,C \in K\,],$$

where the inequality is Anderson's (Anderson, 1955). So $\Delta = 0$ maximises the success probability itself. The price is the second assumption: that the set of corrections an attempt tolerates is convex and centred. We do not verify it, and a task with two disjoint ways of succeeding violates it.

Proposition 3 (one-step warm start is a posterior mean). *Let $v_\theta(x, t; o) = \mathbb{E}[\varepsilon - a \mid x_t = x,\, o]$ be the exact velocity field of the path equation 2. Then for every x, $t \in (0, 1]$ and o,*

$$x - t\, v_\theta(x, t; o) \;=\; \mathbb{E}[\,a \mid x_t = x,\; o\,].$$

At the deployed setting $m(t_0) = 1$, Ws *therefore emits $a_k = \mathbb{E}[\,a \mid x_t = x^{\text{rec}}_{k,t_0},\, o_k\,]$, and*

$$a_k - \bar{a}_k \;=\; \mathbb{E}[\,a \mid x^{\text{rec}}_{k,t_0},\, o_k\,] - \mathbb{E}[\,a \mid x^{\text{rec}}_{k,t_0},\, o^{\text{rec}}_k\,].$$

From $x = (1-t)a + t\varepsilon$, $\mathbb{E}[\varepsilon \mid x, o] = (x - (1-t)\,\mathbb{E}[a \mid x, o])/t$, so $t\, v_\theta(x, t; o) = t\,\mathbb{E}[\varepsilon \mid x, o] - t\,\mathbb{E}[a \mid x, o] = x - \mathbb{E}[a \mid x, o]$. One Euler step of size t_0 from x^{rec}_{k,t_0} is exactly $x - t_0\, v_\theta(x, t_0; o_k)$, which gives the first claim; at recording time the same step under o^{rec}_k returned $\bar{a}_k$, and subtracting the two gives the second. The identity is exact for one step only; for $m > 1$ the output is the endpoint of the flow from x^{rec}_{k,t_0}, not a conditional mean. It holds for any one-step estimate, the policy's own included; what is specific to Ws is the state it is applied to. □

Proposition 4 (the tape kernel). *Under the forward law $p(x_t \mid a) = \mathcal{N}((1-t)\,a,\; t^2 I)$ of the path equation 2,*

$$p(a \mid x_{t_0}, o) \;\propto\; \pi_\theta(a \mid o)\, \mathcal{N}\big(a;\, c,\, s^2 I\big), \qquad c = \frac{x_{t_0}}{1 - t_0}, \quad s = \frac{t_0}{1 - t_0}.$$

By Proposition 3, the one-step output of Ws *is the mean of the policy's own conditional $\pi_\theta(\cdot \mid o_k)$ restricted by a Gaussian kernel of width $s(t_0)$ centred at the stored chunk (up to the stored noise);*

t_0 sets how far from the tape the policy is allowed to move. As $t_0 \to 0$ the kernel collapses onto the stored chunk (open-loop replay); as $t_0 \to 1$ it becomes flat and the output is the policy's own conditional mean.

Bayes' rule with the Gaussian forward law gives $p(a \mid x_{t_0}, o) \propto \pi_\theta(a \mid o) \exp\big(-\|x_{t_0} - (1-t_0)a\|^2/2t_0^2\big)$, and $\|x_{t_0} - (1-t_0)a\|^2/t_0^2 = \|a-c\|^2/s^2$. The two limits follow from $s \to 0$ and $s \to \infty$. The statement assumes that the stored state x^{rec}_{k,t_0} is distributed as the forward marginal at t_0, which the recording only approximates. □

Corollary 4.1 (what warm-starting carries over). *If, locally, $a \mid o \sim \mathcal{N}(\mu(o), \sigma^2 I)$, then*

$$\mathbb{E}[a \mid x_{t_0}, o] - c \;=\; \big(\mu(o) - c\big)\,\frac{s^2}{s^2+\sigma^2}.$$

Along directions in which the policy is confident ($\sigma \ll s$) the stored chunk is replaced by the policy's mean; along directions in which it is uncertain ($\sigma \gg s$) the stored chunk passes through. Writing $c = \bar{a}_k + \frac{s}{1-t_0}\varepsilon^{\text{rec}}$ and $\bar{a}_k = \mu(o_k^{\text{rec}}) + (\bar{a}_k - \mu(o_k^{\text{rec}}))$, the only content the output retains from the recording, relative to the policy's own mean, is the tape's deviation from its own scene, $\bar{a}_k - \mu(o_k^{\text{rec}})$, attenuated by $\sigma^2/(s^2+\sigma^2)$; no term of the operator estimates the shift $\mu(o_k) - \mu(o_k^{\text{rec}})$.

The mean of a Gaussian prior times a Gaussian kernel is the precision-weighted average of their centres, which is the displayed formula; the decomposition of c substitutes the stored chunk and rearranges. This is the formal counterpart of the division of labour in Sec. 3.4: a constant offset is the only element of the construction that can address a displacement that persists across attempts (Corollary 1.1), and the warm start is the only one that carries the recording. □

The gate statistic's natural zero. Write drop_k as in equation 7. If a_k^{tape} is itself a draw from $\pi_\theta(\cdot \mid o_k)$, the two terms have the same expectation and $\mathbb{E}[\text{drop}_k] = 0$ exactly — for any conditional, on any base policy, in any units, with nothing Gaussian or otherwise assumed about the action distribution. The statistic therefore carries a natural zero, *the tape is as typical here as the policy's own plan*, and what a site's threshold θ_w records is how far below that zero the site's tape sits. It is also why a firing rate transfers across sites and base policies while the level it induces does not: the zero is shared, the quantile is shared, and the nats are not (Table 2).

C LIBERO AND ROBOLAB BENCHMARKS

This appendix shows what a site looks like on each benchmark and reports the one setting the main text leaves out: the method switched on where nothing has been displaced. We build the sites of the main experiments on the LIBERO benchmark's own task layouts. Figure 4 shows, for one task of each suite, the benchmark's own initial state and the displaced site we derive from it.

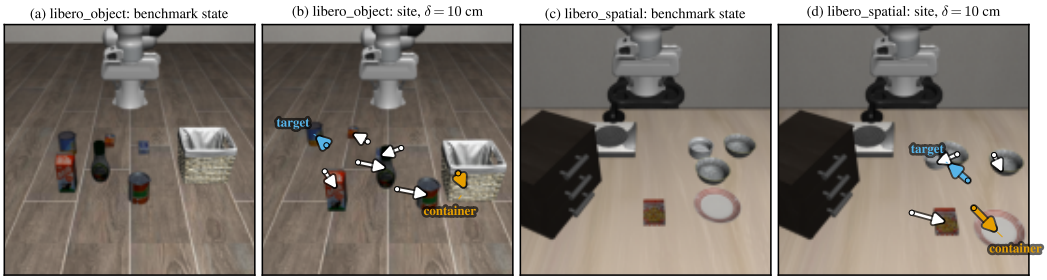


Figure 4: A site on each suite. **(a,b)** `libero_object` task 0; **(c,d)** `libero_spatial` task 0; site 1001 in both. Left of each pair: the benchmark's own initial state. Right: every movable body displaced by $\delta = 10$ cm in its own direction, drawn once and held for every attempt at the site, plus this attempt's residual jitter ($\sigma = 0.5$ cm). Arrows join each body's two positions, projected through the `agentview` camera. The container moves too (the basket in (b), the plate in (d)), so the pick point and the place point move independently.

We also ran the method where nothing is displaced at all: on the four standard LIBERO suites, with the benchmark's own initial states and no perturbation of any kind, and with every constant left at its deployed value (Table 5).

	success rate			paired difference		
Suite	policy	Ws	+Dg+RS	Ws−pol.	(+Dg+RS)−Ws	(+Dg+RS)−pol.
`libero_spatial`	.955	.955	.965	+.000	+.010	+.010
`libero_object`	.990	.905	.975	−.085	+.070	−.015
`libero_goal`	.975	.865	.935	−.110	+.070	−.040
`libero_10`	.965	.800	.955	−.165	**+.155**	−.010
pooled, G=40	—	—	—	−.090	**+.076**	−.014

Table 5: The method on the four standard suites, *no displacement and no jitter*, using each benchmark task's own initial states. One site per task, 10 tasks per suite, 190 episodes per task; differences are paired per task.

At this saturated operating point the procedure is not harmful. The tape alone loses (−.090 pooled), the gate and the residual recover it, and the composition lands on zero, with the same shape in all four suites. This is the failure mode a tape-based method is most exposed to — replaying a stored success where the policy did not need help — and what the gate buys is the ability to leave such a site alone. The statement is weaker than it looks, since three quarters of the tasks leave no room to improve, but it is the one that matters for a method meant to be switched on at a site whose difficulty is not known in advance.

RoboLab. RoboLab (Yang et al., 2026) is a high-fidelity simulation benchmark built on Isaac Sim for analysing task-generalist policies: photorealistic indoor scenes, a Franka arm, and the DROID camera set — one exterior and one wrist camera at 1280×720 — so that a policy fine-tuned on DROID (Khazatsky et al., 2025) runs on it with its inputs unchanged. We use the DROID-fine-tuned $\pi_{0.5}$ with joint-position control and evaluate on eight sites of one task, each a fixed layout of the objects with a per-attempt jitter of $\sigma = 2\,\text{cm}$. The sites differ in where the objects sit, but no shift relative to the policy's training layout is imposed. Figure 5 shows the two camera images the policy receives at one such site.



Figure 5: RoboLab. The two images the policy receives at a site of the task *put the banana, then the cube, in the bowl*: left, the exterior camera; right, the wrist camera. The layout carries a per-attempt jitter of $\sigma = 2\,\text{cm}$.

D THE DRIFT THE DENSITY GATE READS

Sec. 4 reads Fig. 3(a)–(c) as evidence that a stored chunk's validity decays along the episode. This appendix gives the filters behind those panels and the measurements behind the sentences. We read panels (a)–(c) of Fig. 3 on the warm-up episodes recorded before the gate is armed, so that no trajectory is truncated by its own firing, and only on episodes of at least 12 chunks, so that the same episodes are present along the whole axis. Within this set we compare late chunks ($k \geq 4$) with early ones ($k \leq 3$) on the same episodes.

Both moments of drop move, and at different speeds. The median falls by 69, 58 and 170 nats at $\sigma = 0.5/1/2$, while the standard deviation grows by factors of 2.55, 2.41 and 4.91. In the space the environment sees, the executed chunk departs from the tape chunk it was started from by factors of 2.74, 1.79 and 1.95 over the same comparison: the same stored start point, run under an observation the loop has moved, no longer returns the stored action.

Two consequences follow. The single scalar θ_w is a rate only in aggregate: on these episodes it is crossed on 2%, 1% and 0% of chunks at $k \leq 3$ against 14%, 14% and 16% at $k \geq 4$. And because the dispersion grows faster than the level, equation 8 reads the level rather than a trend: a quantile of the pooled drop distribution fires exactly when a chunk lands in the tail that this growth creates.

The aiming test. The gap reported in Sec. 4, would-fire episodes failing at .794 against .387, holds at all three jitters (+.52, +.49, +.29), survives stratification on tape length, and rests on no single site. Recomputing the threshold from the other episodes at each site leaves it at +0.363, and within the firing window the rule separates failing from succeeding episodes with an AUC of .79, against .58 for a placebo read off the first four chunks. Both drift quantities are descriptive; the gate's effect on success is what the ablation study measures.

The estimator's own noise. Each $\log p_\theta$ in equation 7 is a discretised integral with a single-probe Hutchinson trace, so drop carries estimator noise. We measured it from the two policy draws of equation 7, whose difference scaled by $\sqrt{2}$ estimates the spread of a single $\log p_\theta$: it grows by factors of 1.37, 1.09 and 1.30 between early and late chunks, against 2.55, 2.41 and 4.91 for drop itself, and accounts for about half of $\mathrm{Var}(\mathrm{drop})$ early and under 15% late. Subtracting it raises the reported ratios; the figure is the conservative reading.

The rule across jitter levels. Table 6 reads the same rule as a detector. When we build θ_w from the episodes of one jitter level and run those of another, the firing rate rises to $4.3\,q$ when the scenes are worse than the pool and falls to $0.26\,q$ when they are better, while the diagonal stays at q by construction.

	scenes drawn at		
pool built at	σ=0.5	σ=1	σ=2
$\sigma = 0.5$	**.066**	.163	.285
$\sigma = 1$	.032	**.065**	.150
$\sigma = 2$	.017	.039	**.063**

Table 6: The gate as an outlier detector with respect to its pool. Per-chunk firing rate over 10 sites when θ_w is built from the pool episodes of one jitter level and the episodes of another are run. The diagonal is the realised rate at matched jitter, 0.063–0.066 against the nominal $q = 0.08$; off it the rate rises to 4.3× the diagonal when the scenes are worse than the pool and falls to 0.26× when they are better.

E THE POST-HANDOFF VARIANT AND ITS INVARIANCE

Sec. 3.4 deploys the residual on every executed chunk and notes that the variant which withholds it until the handoff estimates a different quantity; the ablation in Sec. 4 adds that the deployed setting is not a pure increment over the gated arm. This appendix gives the reason for both. Write the weight of equation 9 as $\beta_k = \alpha\,\mathbb{1}\{k < \tau\} + \mathbb{1}\{k \geq \tau\}$, so that $\alpha \in [0, 1]$ sets how much of the translation the chunks still driven by the tape receive. We deploy $\alpha = 1$ (Sec. 3.4). The variant at $\alpha = 0$, which applies the translation only after the gate has handed back, is the one whose effect can be read as a pure increment over the gated arm, and we collect here why that is so and what the deployed setting gives up.

The handoff is a stopping time. Let $\mathcal{F}_k$ be the σ-algebra generated by the tape, the observations $o_0, \ldots, o_k$, the drops computed up to chunk k and the pool inherited from earlier episodes. The event $\{\tau = k\}$ is $\mathcal{F}_k$-measurable, so τ is a stopping time, and the indicator of "still following the tape" is monotone: it has two exits, firing in equation 8 and exhaustion at $k = L$, and neither is reversible. Since the gate evaluates equation 7 only while it is following, all of its inputs are measurable with respect to $\mathcal{F}_\tau$.

An intervention supported on the chunks $k \geq \tau$ only — one with $\beta_j = 0$ for every $j < \tau$ — therefore leaves both τ and the fired chunk unchanged, pathwise under common random numbers: it modifies nothing that enters $\mathcal{F}_\tau$, and monotonicity forbids re-entry. We verified this identity on every logged episode of the rounds that ran such an intervention, which doubles as a check that the operator is attached where it claims to be. The deployed residual, supported on every chunk, falls outside the proposition by design.

What the invariance buys, and what the deployed setting gives up. At $\alpha = 0$ the fired chunk and τ are unchanged, so the difference against the same arm with $\Delta = 0$ is a pure increment, with no accompanying change in how much of the tape was used. At $\alpha = 1$ this invariance is forfeited. The gate still scores the injected chunk before the translation is applied to it, but the observation it scores under has already been displaced by the translations executed at earlier chunks, so it does not read the statistic it would have read at $\Delta = 0$. The consequence is visible in Fig. 3(d): with the residual on every chunk the gate fires in almost every episode rather than in three of five, and its median first firing moves earlier, from chunks 8, 5 and 7 to 3, 4 and 5 at the three jitters. Two things follow. The deployed arm is not a pure increment over the gated arm, since the search can raise the outcome either by finding a better Δ or by surrendering more of the tape, and a single success rate does not separate the two. And because chunk k is scored before Δ is added to it, the gate's response to the translation lags it by one chunk.

The estimand at $\alpha = 0$. We call an attempt *acted* if Δ had support in it. At $\alpha = 1$ every attempt is acted and the search estimates $\Pr[y = 1 \mid \Delta]$. At $\alpha = 0$ an episode that concludes while still inside the tape carries no information about Δ, so we restrict the elite set to acted attempts and the quantity the search estimates is $\Pr[y = 1 \mid \text{acted}, \Delta]$; including the other episodes would attenuate the update towards 0 by the fraction of successes decided before the handoff.

F An Entry Test on the Density

The gate of Sec. 3.3 decides when to stop using a tape that an episode has already started to use. A second, logically prior decision is available — whether to use the tape on this episode at all. Therefore, we conducted an extended test to verify whether density could be used as an entry.

The entry test requires the tape to be stored together with the observations $(o_0^{\text{rec}}, \ldots, o_{L-1}^{\text{rec}})$ under which it was produced, which the deployed method does not need. Before an episode is injected, we score the tape's first chunk under the current first observation and reference it to its density at the moment it was recorded, a number computed once and stored with the tape:

$$\text{drop}_0 \;=\; \log p_\theta(\bar{a}_0 \mid o_0) \;-\; \log p_\theta(\bar{a}_0 \mid o_0^{\text{rec}}) \;<\; \theta_{\text{in}} \;\Longrightarrow\; \text{do not inject this episode.} \quad (12)$$

Because the same chunk is scored under two observations, chunk identity cancels and what remains is the displacement of the scene relative to the recording. The quantity therefore has a fixed zero and is comparable across sites, which is what licenses a single constant θ_{in} shared by all of them, where equation 8 has to threshold on a rate.

G Which Stored Success You Record

The deployed method records the first success at a site and uses that one tape; the limitations in Sec. 6 note that we do not select it. To see how much the choice of tape matters, we recorded three tapes at each of sixteen sites and ran Ws with each. The effect varies substantially across the tapes of one site — the between-tape spread of the effect is about 0.17–0.19 in success rate — and it varies from site to site. Selecting the best of three tapes rather than drawing one would be worth about $+0.09$ in success rate after correcting for the selection itself, but this is an oracle quantity: we propose no rule for choosing a tape, and we leave tape selection to future work. Taking the first success, as the deployed method does, introduces no detectable selection bias relative to a random tape ($+0.038$).

H Constructions That Did Not Survive the Closed Loop

Sec. 3.1 says that the three entry points are not the only way to act at fixed θ. This appendix lists the other constructions we tried. We ran six of them, and none survived the closed loop. Table 7 lists

each with what we measured and the mechanism we read from its own measurement; none failed for want of tuning.

Construction	Measured	Why it fails
Retrieve the tape by state, so it lasts longer	-0.156 ± 0.037	Coverage rises to 1.000 and SR does not; at the correct operating point it is harmful, not merely useless.
Average K samples	$1/\sqrt{K}$ holds at source, 0.79 by divergence, zero by step 191	The loop does not retain a variance reduction. *Scope:* this is uniform averaging, whose whole content is variance reduction; it says nothing about a weighting that carries value information.
Store the noise instead of the action	-0.030 ± 0.029; dispersion ratio $\approx$1.0	Fixed ε is not a persistent bias. *Scope:* this kills replaying a whole ε at unit amplitude, not searching its low-dimensional mean component.
Return to a reached state, then branch	pass@3 -0.333 ± 0.082; within-scene ICC $+0.605$ above the policy's	A shared prefix destroys the independence repeated draws rely on.
Use another site's tape	-0.101 ± 0.034, three rounds	Specificity.
Guide the velocity field	$18.2 \to 5.5 \to \leq 0.8$ cm	The loop compensates $\approx$96%.

Table 7: Each row is a measured effect with the mechanism we read from it, not an absence of effect.

Individually each row is a negative result. Together they are the empirical content of the design: every construction that injected a bias at the start survived, and every construction that reduced variance, edited the velocity field or borrowed another site's success did not.

I TRAINING-TIME RANDOMISATION, IN FULL

Before adapting at deployment time, we asked whether the deficit could be removed at training time instead. We fine-tuned three arms of the base policy on placement randomisation of radius 5, 8 and 12 cm, balanced in episode count so that the width of the support is decoupled from the volume of data, and evaluated them against the frozen policy on a paired grid of displacements (Table 8).

Arm	$\delta = 0$	$\delta = 5$	$\delta = 8$	$\delta = 10$	$\delta = 12$
A0 (frozen)	99.7	86.7	65.0	50.7	40.0
S5	98.3	84.3	62.3	47.0	37.3
S8	100.0	86.3	64.7	52.7	40.0
S12	99.3	89.3	63.7	55.0	42.3

Table 8: Success rate (%) of the frozen policy (A0) and of three arms fine-tuned on placement randomisation of radius 5/8/12 cm, on a paired grid of displacements δ. Displacements are per-object, of magnitude δ, in directions drawn afresh for each episode (no persistent site; 10 tasks $\times$ 30 episodes per cell; 600-step cap), so these populations are not comparable to the fixed sites of Table 1. None of the fifteen cells differs from the frozen policy; S8, whose training support contains $\delta = 8$, is null there too.

None of the fifteen cells differs from the frozen policy. The arm whose training support contains the test displacement is null there as well, so the deficit is not a matter of the randomisation never having touched this configuration; what matters is the density of coverage, not its extent.

Why covering scales badly here. A site is $\sim$7 objects $\times$ 2D $=$ 14 dimensions. Sampling N training configurations uniformly, the nearest-neighbour distance to a deployment configuration, as a fraction of the randomisation radius, is 0.611 at $N = 10^3$ and still 0.373 at $N = 10^6$ in $d = 14$, against 0.032 and 0.001 in $d = 2$. Pressing it to 1/8 of the radius needs 64 configurations at $d = 2$ and 4.4×10^{12} at $d = 14$. We take this as motivation rather than a demonstrated cause.