RETHINKING WHERE ROBOT ACTIONS BEGIN: LEARNING AN EXPERT-INVERSION PRIOR

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow-matching robot policies have demonstrated strong performance, yet typically sample from random *Gaussian* noise without explicitly guiding starting-point sampling using expert demonstrations. We investigate action inversion as an interface for such guidance. Our theoretical analysis shows that matching expert conditional starting-point distributions reduces expert–policy action-distribution mismatch, while experiments reveal the behavioral value of expert starting points and their correspondence with the current observation. Motivated by these findings, we introduce Expert-Inversion Prior (EIP), which learns observation-conditioned priors from expert-inverted starting points, incorporating expert information while retaining distributional action generation. We instantiate EIP with conditional *Gaussian* and conditional Flow priors. Across 12 simulation tasks, EIP-Gaussian and EIP-Flow improve average success over the base policies by 25.5 and 25.8 percentage points on FM-UNet, and by 20.5 and 21.8 percentage points on FM-DiT, respectively. Across four real-world tasks, the corresponding gains are 27.5 and 25.0 percentage points. These improvements require neither action-generator updates nor additional demonstrations. Reduced-step inference further improves success over the standard baselines while reducing latency by 20%-58% in our evaluations. These findings provide theoretical and empirical insights into expert-guided starting-point sampling for robot policies.

1 INTRODUCTION

Flow-matching robot policies formulate action generation as a conditional transport process that transforms a *Gaussian* noise sample into an action chunk conditioned on the current observation (Lipman et al., 2023; Black et al., 2025; Physical Intelligence et al., 2025; NVIDIA et al., 2025; Shukor et al., 2025). By modeling conditional action distributions , these policies can represent multiple modes of behavior under the same observation and have demonstrated strong performance across manipulation tasks. Variation in generated actions under a fixed observation arises from the sampled starting points. However, these starting points are typically drawn from a fixed standard *Gaussian* distribution: expert demonstrations supervise the action generator, but do not explicitly shape a starting-point prior. When a gap remains between policy-generated and expert behavior, a natural question arises: can existing expert demonstrations guide starting-point sampling while retaining the capacity to represent multimodal behaviors, improving inference without updating the action generator or collecting additional demonstrations?

Recent work has explored starting-point selection and noise-space guidance for robot policies through reward optimization, historical information reuse, and execution constraints (Wagenmaker et al., 2025; Simsir & Oguz, 2026; Jia et al., 2026b; Kang et al., 2026; Zhang et al., 2026; Chen et al., 2026), establishing starting points as an important interface for influencing policy behavior. Meanwhile, image inversion recovers sampling starting points from reference images to support reconstruction and editing (Song et al., 2021; Mokady et al., 2023), and recent action-inversion methods extend this perspective to reference-guided control, temporal plan reuse, and adaptation from expert corrections (Tang et al., 2026; Cao et al., 2026; Murray et al., 2026; Lu et al., 2026; Ho et al., 2026). These advances show that reference behaviors can be mapped into starting-point information that guides subsequent generation, providing an interface for guiding starting-point sampling using expert demonstrations. Unlike prior work, we take a deep dive into the distributions produced

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

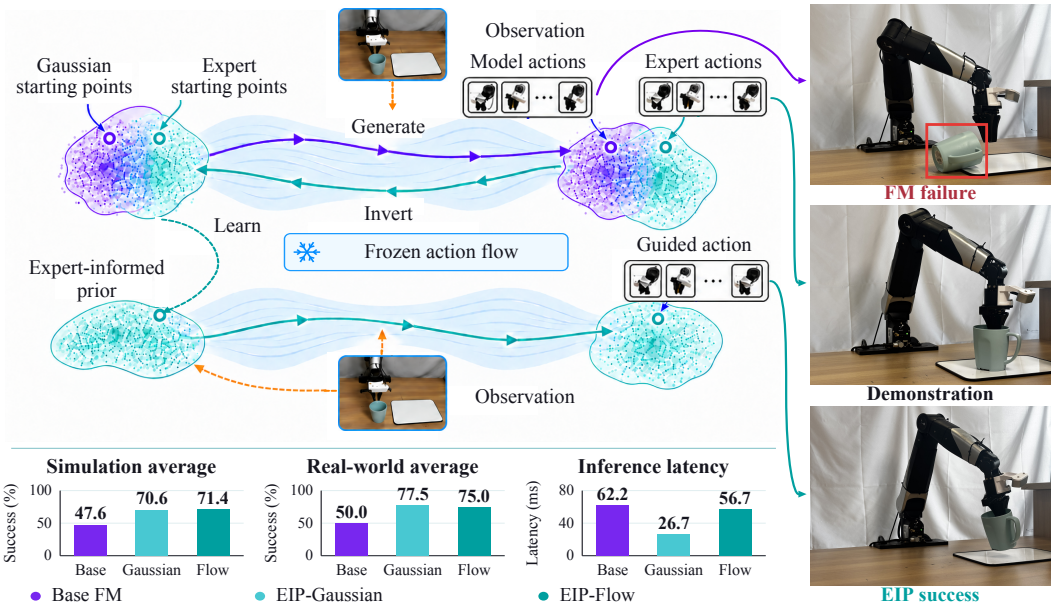

Figure 1: Overview of Expert-Inversion Prior (EIP). Expert actions are inverted through a frozen action generator to supervise observation-conditioned starting-point sampling. At inference, the learned prior guides the same generator without additional demonstrations or generator updates. Paired rollout examples illustrate different execution outcomes, while the plots summarize average simulation and real-world success rates and inference latency.

by expert action inversion, their relationships with observations and generated actions, and how they can guide starting-point sampling.

We organize our study around three connected questions: *What distribution does expert action inversion produce? How does this distribution relate to observations and generated actions? How should this information be used?* First, through comparisons among Gaussian starting points, starting points obtained by inverting model-generated actions, and those inverted expert actions, we observe differences between the distribution of expert-inverted starting points and the Gaussian prior. Analysis of conditional invertible transport explains this observation: The KL divergence between the expert-inverted starting-point distribution and the Gaussian prior equals that between the expert action distribution and the policy-generated action distribution. Second, we use a conditional KL decomposition to distinguish information in the marginal starting-point distribution from the correspondence between observations and starting points, and compare Gaussian sampling, random expert starting-point sampling, and observation-based retrieval in closed-loop execution. The results show that marginal expert starting points provide useful guidance in some configurations, while observation-based starting-point selection yields more consistent improvements. Finally, as illustrated in Figure 1, we introduce Expert-Inversion Prior (EIP) to guide starting-point sampling with expert information while retaining the capacity to represent multimodal action distributions. EIP learns conditional starting-point priors from inverted expert demonstrations and decodes sampled starting points through the frozen action generator. We consider two implementations, EIP-Gaussian and EIP-Flow, and conduct extensive experiments demonstrating their effectiveness in improving task success and reducing inference latency, without updating the action generator or collecting additional demonstration data.

Our main contributions are:

- We show that, under inversion, the KL divergence between the expert-inverted starting-point distribution and the Gaussian distribution equals that between the expert and policy action distributions. This provides a theoretical basis for improving starting-point sampling through expert-inversion supervision.
- We distinguish the marginal expert starting-point distribution from observation–starting-point correspondence, showing how both guide action generation and how observation-based selection can provide additional benefits.

- We introduce EIP-Gaussian and EIP-Flow for observation-conditioned starting-point sampling with a frozen action generator and no additional demonstrations. They improve average success over the baselines by 20.5-25.8 percentage points across 12 simulation tasks and by 27.5 and 25.0 percentage points, respectively, across four real-world tasks. Reduced-step inference reduces latency by 20-58% relative to the baselines in three of four evaluated configurations.

2 Related Work

2.1 Image Inversion and Controllable Generation

Image inversion maps reference images into the latent representations of pretrained generative models to support reconstruction and subsequent editing. Early work such as Image2StyleGAN investigates embedding real images into GAN latent spaces (Abdal et al., 2019), while subsequent studies examine reconstruction fidelity and editability as related but distinct objectives. In diffusion models, DDIM provides deterministic sampling trajectories that support approximate inversion (Song et al., 2021), while Null-text Inversion and EDICT improve reconstruction and editing through conditioning optimization and invertible process construction, respectively (Mokady et al., 2023; Wallace et al., 2023). Prompt-to-Prompt further demonstrates how internal representations can control image structure and semantic changes (Hertz et al., 2023). These studies motivate using reference data to guide frozen generators while distinguishing reconstruction fidelity from downstream usefulness. We investigate this distinction in robot action generation by examining expert-inverted starting-point distributions and their value under the current observation.

2.2 Robot Action Inversion

Recent studies use action inversion to connect reference behaviors with generative robot policies. SegDiff employs DDIM inversion to reuse and refine historical action plans (Cao et al., 2026). Flow Reversal Steering (FRS) uses reverse flow integration to guide generation from reference actions; its DSBC method also learns a noise policy from inversion-derived targets, including offline demonstrations (Tang et al., 2026). PAINT selects initial noise through backward inversion to improve consistency between asynchronously executed action chunks (Ho et al., 2026). FlowDAgger and UniSteer use expert corrections for noise-space supervision and policy adaptation (Murray et al., 2026; Lu et al., 2026). Our study examines the distribution produced by inverting expert demonstrations, separates numerical inversion error from expert–policy distribution mismatch, and evaluates the role of observation–starting-point correspondence. We then model an observation-conditioned distribution of expert-inverted starting points for sampling through a frozen action generator.

2.3 Starting-Point Modeling and Guidance for Robot Policies

Generative robot policies commonly sample actions from a simple source distribution; Diffusion Policy established conditional action generation as an effective approach to visuomotor control (Chi et al., 2023). Subsequent work has explored how the choice of starting points affects behavior. DSRL learns a reinforcement learning policy in the noise space of a frozen generator (Wagenmaker et al., 2025), while LP-DS uses constrained noise perturbations (Simsir & Oguz, 2026). A2A, FLASH, MARS, and WarmPrior incorporate information from execution or action history into initialization (Jia et al., 2026b; Bai et al., 2026; Jia et al., 2026a; Kang et al., 2026). Streaming Flow Policy samples near the previous action to support incremental action generation (Jiang et al., 2025). Closely related to our setting, LAFM uses a latent action model to select an observation-dependent source distribution from a learned library (Machado et al., 2026). EIP instead derives supervision for the starting-point distribution by inverting existing expert actions through a frozen generator. We study the resulting marginal and observation-conditioned distributions and instantiate conditional sampling with EIP-Gaussian and EIP-Flow.

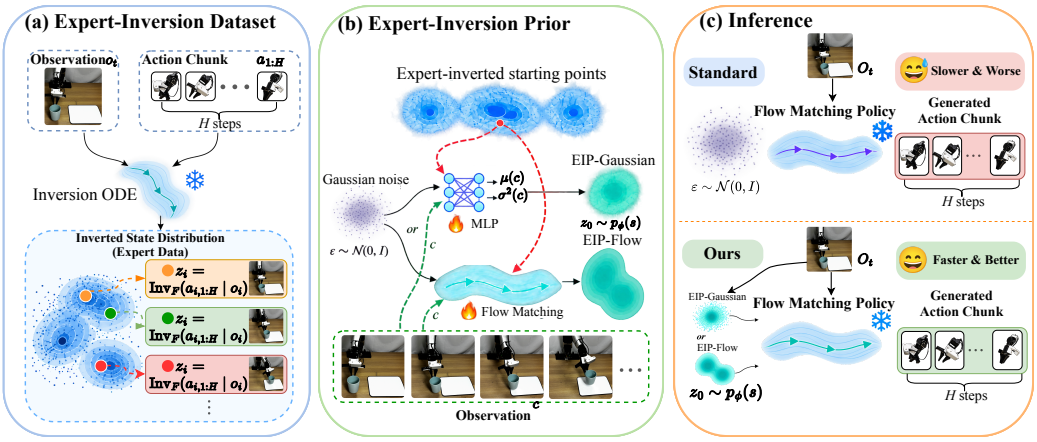

Figure 2: Overview of Expert-Inversion Prior (EIP). (a) Expert action chunks are inverted under their associated observations using the frozen action generator to construct starting-point supervision. (b) The illustrated EIP-Flow learns a conditional transport from Gaussian noise to expert-inverted starting points; EIP-Gaussian instead predicts a conditional diagonal Gaussian. (c) At inference, either learned prior samples an observation-conditioned starting point, which is decoded by the same frozen action generator. Prior learning uses existing demonstrations without updating the action generator or collecting additional data.

3 What Are Inverted Starting Points?

Standard flow-matching policies generate actions from starting points sampled from a *Gaussian* distribution. To use expert demonstrations to guide this sampling process, we need to understand whether inverting expert actions yields a different starting-point distribution and what such differences represent. Section 3.1 introduces flow matching and action inversion. Section 3.2 compares expert-inverted starting points with Gaussian starting points and inversions of model-generated actions, using numerical controls to assess the contribution of inversion error. Section 3.3 explains the relationship between starting-point and action distributions, providing a theoretical basis for learning expert-informed starting-point priors.

3.1 Flow Matching and Action Inversion

Let c denote the observation condition and $A \in \mathbb{R}^D$ a complete normalized action chunk. Flow matching learns a time-dependent velocity field $v_\theta(x_t, t, c)$ that transports noise into actions (Lipman et al., 2023). Starting from $z \sim \gamma = \mathcal{N}(0, I_D)$, generation integrates the following ODE, defining the conditional map $A = F_c(z)$:

$$\frac{\mathrm{d}x_t}{\mathrm{d}t} = v_\theta(x_t, t, c), \qquad x_0 = z, \quad A = x_1. \tag{1}$$

Action inversion uses the same velocity field with fixed model parameters and observation condition to integrate backward from an expert action chunk A^E at $t = 1$ to $t = 0$ (Hairer et al., 1993; Chen et al., 2018; Grathwohl et al., 2019). For an invertible flow solved exactly, this recovers $z^E = F_c^{-1}(A^E)$. Equivalently, setting $s = 1 - t$ gives:

$$\frac{\mathrm{d}y_s}{\mathrm{d}s} = -v_\theta(y_s, 1 - s, c), \qquad y_0 = A^E, \quad z^E = y_1. \tag{2}$$

3.2 Numerically Controlled Distribution Diagnostics

We compare three sets of starting points across frozen FM-UNet, FM-DiT, and π_0 policies: *Gaussian* samples, inverted model-generated actions, and inverted expert actions. FM-UNet and FM-DiT are evaluated on the same two tasks using 128 test-split action chunks per task. For π_0, we evaluate 50 samples per task and compute statistics over the seven active action channels across the 50-step prediction horizon, excluding padded channels. We report the sliced Wasserstein distance (SW_2)

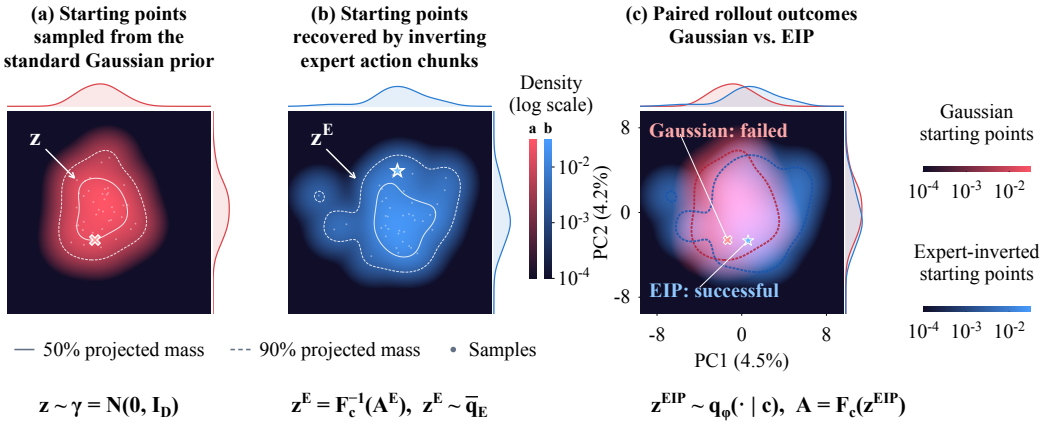

Figure 3: Starting-point distributions and paired rollout outcomes on RoboMimic Square with a frozen FM-UNet. (a,b) Gaussian and expert-inverted starting-point densities in a shared PCA projection; solid and dashed contours enclose 50% and 90% of projected mass. Markers identify a Gaussian sample and the corresponding expert-inverted point. (c) Overlaid densities with a Gaussian-failure/EIP-success example under the same initial observation; the successful EIP sample is distinct from the expert-inverted point in (b). Top and right curves show marginal densities. Colors indicate density on a logarithmic scale.

to an independent standard Gaussian sample set to quantify distributional differences, and the mean squared norm, $N^{-1}\sum_{i=1}^{N}\|z_i\|_2^2$, to characterize starting-point scale. Within each configuration, all three sets use matched sample counts, the same Gaussian reference, and the same projection directions. The Gaussian-to-Gaussian distance provides a finite-sample baseline rather than an expected zero value. Metric definitions are provided in Appendix B.3. Numerical recovery protocols and solver-convergence studies are detailed in Appendices B.2, B.5, and B.6.

Expert-inverted starting points exhibit larger sliced Wasserstein distances to the Gaussian reference than the Gaussian and model-inversion controls across the evaluated configurations. A separate CloseBox convergence study shows that this mismatch persists as numerical error decreases: at FP32 RK4-256, starting-point recovery and expert-action reconstruction RMSEs reach 1.006×10^{-6} and 3.941×10^{-8}, while the expert-to-Gaussian distance remains approximately 0.4033, compared with 0.2078 for the Gaussian control (Appendix B.5). Figure 3 visualizes the projected distributions on Square and a paired Gaussian-failure/EIP-success example under the same initial observation.

3.3 Distributional Interpretation

To explain the observed deviations of expert-inverted starting-point distributions from the Gaussian prior, we analyze how inversion transforms the discrepancy between expert and policy-generated action distributions. Fix an observation condition c. Assume F_c is a diffeomorphism and the relevant distributions admit densities with finite KL divergences. Then, the following KL identity holds:

$$D_{\mathrm{KL}}(q_E(\cdot \mid c) \,\|\, \gamma) = D_{\mathrm{KL}}(p_E(\cdot \mid c) \,\|\, p_\theta(\cdot \mid c))\,. \tag{3}$$

The derivation is provided in Appendix A.1. This relation explains why expert-inverted starting-point distributions can differ from the standard Gaussian prior even when inversion is accurate: inversion preserves the underlying expert–policy distribution mismatch, expressing it in starting-point space. The Gaussian prior is recovered exactly only when the base policy perfectly matches the expert conditional action distribution under the same observation condition.

This analysis also suggests how the discrepancy can be corrected. Let $q_\phi(\cdot \mid c)$ be a learned starting-point distribution and $p_\phi(\cdot \mid c) = (F_c)q_\phi(\cdot \mid c)$ the action distribution induced by the frozen generator. Under the same assumptions,

$$D_{\mathrm{KL}}(q_E(\cdot \mid c) \,\|\, q_\phi(\cdot \mid c)) = D_{\mathrm{KL}}(p_E(\cdot \mid c) \,\|\, p_\phi(\cdot \mid c))\,. \tag{4}$$

Therefore, reducing the discrepancy between the learned and expert starting-point distributions equivalently reduces the discrepancy between their generated action distributions. Exact starting-

Table 1: Closed-loop success rates (%) on LIBERO-Goal. Each method uses the same 50 initial states and three sampling seeds. Expert starting points come exclusively from the training bank.

Model	Task	Gaussian	Random expert	Retrieval
UNet	OpenDrawer	36.7	82.0	80.0
	BowlOnPlate	89.3	89.3	90.0
DiT	OpenDrawer	46.7	40.0	86.0
	BowlOnPlate	70.7	90.0	96.0

point matching recovers the expert conditional action distribution without modifying the action generator. This identifies expert inversion as a target for conditional starting-point learning. The next section examines the relationships among starting points, observations, and generated actions.

4 How Do Starting Points and Observations Jointly Shape Action Generation?

Expert inversion expresses expert–policy distribution mismatch in starting-point space, but generation depends jointly on the starting point and the observation. We therefore distinguish information in the marginal expert starting-point distribution from its correspondence with observations.

Let $\rho(c)$ denote the observation distribution, and define

$$q_E(c, z) = \rho(c) q_E(z \mid c), \qquad \overline{q}_E(z) = \int \rho(c) q_E(z \mid c)\, \mathrm{d}c. \tag{5}$$

For an observation-independent starting-point distribution $r(z)$, let $p_r(\cdot \mid c) = (F_c)_{\#} r$. Under the invertibility, density, and finite-divergence assumptions of Section 3,

$$\begin{aligned} \mathbb{E}_{c\sim\rho} D_{\mathrm{KL}}(p_E(\cdot \mid c) \,\|\, p_r(\cdot \mid c)) &= \mathbb{E}_{c\sim\rho} D_{\mathrm{KL}}(q_E(\cdot \mid c) \,\|\, r) \\ &= D_{\mathrm{KL}}(\overline{q}_E \,\|\, r) + I_{q_E}(C; Z). \end{aligned} \tag{6}$$

The derivation appears in Appendix A.2. The first term measures marginal distribution mismatch, while the second captures observation–starting-point dependence. Setting $r = \overline{q}_E$ eliminates the first term but leaves a residual conditional action-distribution mismatch whenever $I_{q_E}(C; Z) > 0$.

We examine the behavioral implications using a training-only bank $\mathcal{B} = \{(c_i, z_i^E)\}_{i=1}^M$, comparing Gaussian sampling, uniform random expert starting-point sampling, and observation-based nearest-neighbor retrieval. Random sampling uses marginal expert information; retrieval additionally exploits observation correspondence. All methods select starting points at each step with the same frozen generator, normalization, decoder, and evaluation initial states; see Appendix C.

Table 1 illustrates the value of expert starting-point information and observation correspondence. For UNet, random expert sampling and retrieval perform similarly, with random sampling improving OpenDrawer from 36.7% to 82.0%. For DiT, retrieval further improves success over random expert sampling from 40.0% to 86.0% on OpenDrawer and from 90.0% to 96.0% on BowlOnPlate. Together with the analysis, these results motivate learning observation-conditioned starting-point distributions to exploit both sources of information.

5 How Can Inverted Starting Points Improve Inference?

The preceding section shows that expert-inverted starting points can guide action generation and that their correspondence with the current observation matters. Our goal is to use expert-inversion supervision to shape the conditional starting-point sampling distribution. We therefore introduce *Expert-Inversion Prior* (EIP): we invert existing expert demonstrations, learn an observation-conditioned prior, and sample from it to initialize the frozen action generator. We consider two implementations, *EIP-Gaussian* and *EIP-Flow*, which differ in how they represent and sample this conditional distribution. Figure 2 illustrates the framework.

5.1 Learning Expert-Inversion Priors

For each expert action chunk A_i^E and its observation encoding c_i, inversion produces $\widetilde{z}_i^E = \widetilde{I}_{c_i}(A_i^E)$. These pairs form the supervision dataset $\mathcal{D}_{\text{inv}} = \{(c_i, \widetilde{z}_i^E)\}_{i=1}^M$. The action generator, observation encoder, and base-policy normalization remain fixed. To reduce inversion cost, we retain every eighth eligible window and the final eligible window of each demonstration, while preserving every coordinate of each complete action chunk. Details are provided in Appendix E.

Our goal is to guide starting-point sampling with observation-dependent expert information while retaining the capacity to represent multimodal action distributions. For a frozen deterministic action flow, predicting a single starting point yields a single action chunk under a fixed observation. Instead, we learn a conditional starting-point distribution from expert-inverted demonstrations, allowing expert supervision to shape both where starting points are sampled and how they vary. We instantiate this principle with EIP-Gaussian for lightweight mean-and-variance modeling and EIP-Flow for more flexible joint distribution modeling.

EIP-Gaussian. A network predicts the conditional mean $\mu_\phi(c)$ and positive standard deviations $\sigma_\phi(c)$. It is trained by minimizing Gaussian negative log-likelihood, up to an additive constant:

$$\mathcal{L}_{\text{G}} = \mathbb{E}_{(c,\widetilde{z}_E)\sim\mathcal{D}_{\text{inv}}} \frac{1}{D} \sum_{j=1}^{D} \left[\frac{(\widetilde{z}_{E,j} - \mu_{\phi,j}(c))^2}{2\sigma_{\phi,j}(c)^2} + \log \sigma_{\phi,j}(c) \right]. \tag{7}$$

At inference, it samples $z = \mu_\phi(c) + \sigma_\phi(c) \odot \epsilon$, where $\epsilon \sim \mathcal{N}(0, I_D)$.

EIP-Flow. A separate conditional velocity field u_ϕ transports Gaussian noise to expert-inverted starting points. For $\epsilon \sim \mathcal{N}(0, I_D)$ and $\tau \sim \mathcal{U}[0, 1]$, define $z_\tau = (1-\tau)\epsilon + \tau\widetilde{z}_E$ and optimize

$$\mathcal{L}_{\text{F}} = \mathbb{E}_{(c,\widetilde{z}_E),\epsilon,\tau} \frac{1}{D} \left\| u_\phi(z_\tau, \tau, c) - (\widetilde{z}_E - \epsilon) \right\|_2^2. \tag{8}$$

At inference, integrating $\mathrm{d}z_\tau/\mathrm{d}\tau = u_\phi(z_\tau, \tau, c)$ from $z_0 = \epsilon$ yields a starting point z_1. The learned prior Flow operates in starting-point space, while the frozen action Flow decodes the resulting point as $A = F_c(z_1)$. Appendix D details the architectures, optimization, and checkpoint-selection rules.

Deterministic MLP control. As a deterministic control, we train an observation-conditioned MLP $f_\psi(c)$ on the same inversion targets by minimizing

$$\mathcal{L}_{\text{MLP}} = \mathbb{E}_{(c,\widetilde{z}_E)\sim\mathcal{D}_{\text{inv}}} \frac{1}{D} \| f_\psi(c) - \widetilde{z}_E \|_2^2. \tag{9}$$

At inference, it returns a single starting point for each observation. Unlike EIP-Gaussian and EIP-Flow, it does not model or sample a conditional starting-point distribution; we therefore use it as a deterministic comparison to assess the benefits of distributional starting-point modeling.

5.2 Evaluation across Tasks and Policy Architectures

Simulation Evaluation. To evaluate starting-point guidance across manipulation tasks and benchmarks, we consider 12 tasks from LIBERO-90 (Liu et al., 2023), RoboVerse (Geng et al., 2025), RLBench (James et al., 2020), and RoboMimic (Mandlekar et al., 2022). We evaluate EIP-Gaussian and EIP-Flow with FM-UNet (Lipman et al., 2023) and FM-DiT (Lipman et al., 2023; Peebles & Xie, 2023), comparing them against the corresponding base policies. All methods share the same frozen action checkpoint within each task–architecture configuration and use no additional demonstration data. Table 2 reports fixed-step success rates, while Figure 4 shows representative task scenes. Experimental details are provided in Appendix F.

Real-World Evaluation. We further evaluate EIP on a physical robot across four manipulation tasks: picking up a red cube, placing a mug on a white board, opening an orange drawer, and stacking a black cube on a red box. We compare standard Gaussian initialization with EIP-Gaussian and EIP-Flow using the same fixed action generator. Each method is evaluated over 20 rollouts per task under the same task setup. Figure 5 presents the task procedures and success rates. Averaged

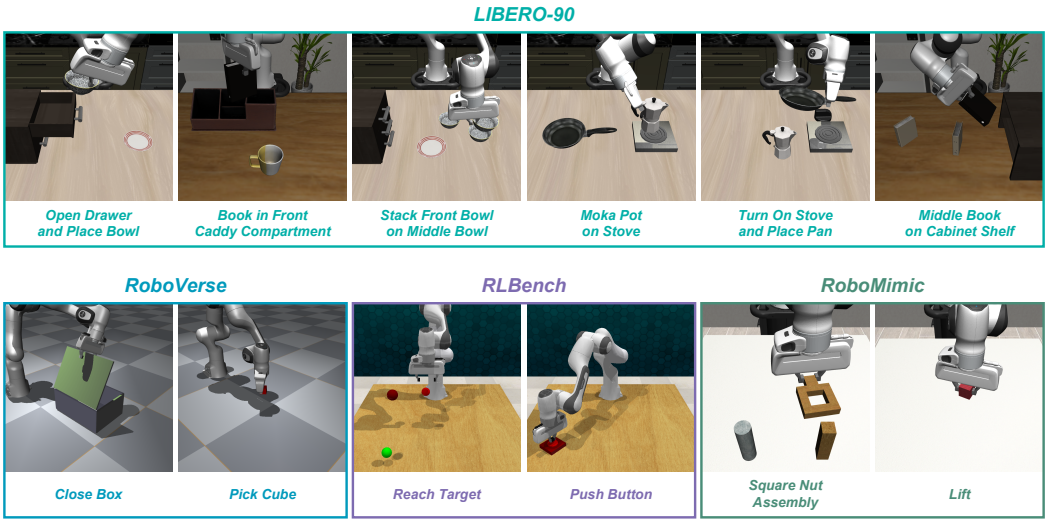

Figure 4: Representative task scenes from the simulation benchmarks used in our evaluation.

Table 2: Success rates (%) on 12 simulation tasks. Each cell reports UNet / DiT. The four row groups correspond to LIBERO-90 (Liu et al., 2023), RoboVerse (Geng et al., 2025), RLBench (James et al., 2020), and RoboMimic (Mandlekar et al., 2022). All methods use 50 matched initial states and the same frozen action checkpoint per task and backbone; learned priors use seed 0. Action generation uses 10 integration steps, and EIP-Flow uses 16 prior steps. Bold marks the highest success rate among the displayed methods for each task and backbone, including ties.

Task	Base FM	Conditional MLP	EIP-Gaussian	EIP-Flow
Open Drawer & Place Bowl	56 / 42	96 / **100**	96 / **100**	**98** / 98
Book in Front Caddy	4 / 52	**40** / 58	36 / 60	**40** / **66**
Stack Front Bowl	46 / 24	**94** / 72	86 / 68	92 / **82**
Moka Pot on Stove	38 / 66	84 / 96	**88** / **100**	86 / 98
Turn On Stove & Place Pan	62 / 54	92 / 82	**94** / **98**	90 / 92
Middle Book on Cabinet Shelf	22 / 94	50 / 92	52 / **96**	**76** / 94
Close Box	54 / 88	70 / 94	**78** / **96**	70 / 92
Pick Cube	8 / 24	4 / 20	**14** / 22	6 / **42**
Reach Target	14 / 18	28 / 24	**30** / 18	28 / **30**
Push Button	50 / 66	44 / 70	**56** / **74**	46 / 66
Square	46 / 16	70 / **58**	**76** / **58**	**76** / 46
Lift	98 / **100**	98 / **100**	98 / **100**	**100** / **100**

across the four tasks, EIP-Gaussian and EIP-Flow improve success over standard initialization by 27.5 and 25.0 percentage points, respectively. These results show that expert-informed starting points can improve physical manipulation without updating the action generator.

Reduced-step inference has different costs for the two priors. EIP-Gaussian reduces measured latency from 77.5 to 32.2 ms with FM-UNet and from 47.0 to 21.1 ms with FM-DiT, while improving success in both cases. EIP-Flow requires additional integration steps to sample a starting point: its latency decreases by 19.6% with FM-UNet but increases by 8.9% with FM-DiT relative to the corresponding ten-step base policy.

Offline Inversion Cost and Inference Latency. We apply stride-8 temporal thinning before expert inversion, retaining the final eligible window of each demonstration. On OpenMicrowave and FrontBowlOnPlate, the eligible training windows decrease from 4,612 to 620 (86.6%) and from 4,271 to 577 (86.5%), respectively, reducing the number of inversions needed to prepare prior train-

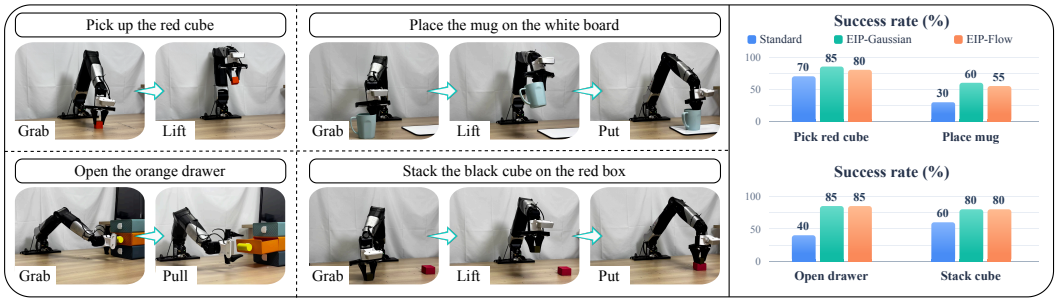

Figure 5: **Real-world manipulation tasks and success rates.**We compare standard Gaussian initialization, EIP-Gaussian, and EIP-Flow on four tasks. Each method was evaluated over 20 rollouts per task under the same task setup.

Table 3: Success rate and inference latency on LIBERO-90 Open Drawer & Place Bowl, using 50 matched evaluation initial states. Latency is averaged over 30 timed repetitions after warm-up on an RTX 5090, using cached observation conditions and including learned-prior sampling, action generation, and output transfer to CPU. Observation encoding, preprocessing, and simulation are excluded. a and p denote action and prior integration steps, respectively. Latency changes are relative to the corresponding ten-step base policy.

Backbone	Method	Steps	Success (%)	Latency (ms)	vs. Base
FM-UNet	Base FM	$a10$	56	77.5	–
	EIP-Gaussian	$a4$	92	32.2	-58.4%
	EIP-Flow	$p8{+}a4$	88	62.2	-19.6%
FM-DiT	Base FM	$a10$	42	47.0	–
	EIP-Gaussian	$a4$	98	21.1	-55.1%
	EIP-Flow	$p8{+}a4$	100	51.2	$+8.9\%$

Table 4: Temporal thinning on StackCube with EIP-Flow: 5,000 training updates, $p8{+}a4$ inference, and two evaluation seeds of 50 episodes each. Training time includes optimization and validation. Stride-8 subsamples the existing cache; inversion-time speedup is not measured.

Cache	Windows	Training time (s)	Success (%)
Full	15,264	195.0	78
Stride-8	2,041	182.8	86

ing data. Complete action chunks and inverse-solver settings are preserved. In a matched StackCube pilot with 5,000 prior-training updates and $p8{+}a4$ inference, success increases from 78% with the full cache to 86% with stride-8 thinning (Table 4). We report online cost separately: Table 3 compares the success and latency of reduced-step EIP inference with the standard ten-step base policy.

6 Conclusion

We study what expert action inversion reveals about the starting points of generative robot policies and how this information can improve inference. Under invertible transport, inversion preserves expert–policy distribution mismatch in starting-point space, while observation–starting-point dependence explains why marginal matching alone can be insufficient. These insights motivate observation-conditioned priors learned from expert-inverted actions with a frozen action generator. Experiments show that EIP-Gaussian and EIP-Flow improve task success across simulation and real-world tasks, while reduced-step inference lowers latency. Our findings establish expert inversion as an interface for inference-time adaptation and motivate further study of expert guidance.

AI USE STATEMENT

We used generative AI tools to assist with manuscript polishing, the writing of mathematical proofs, the refinement of existing hypotheses, code development, and figure generation. We did not use these tools to originate the core research ideas or claims. The use of generative AI tools to generate synthetic datasets is not applicable to this work. In particular, AI-assisted proofs and hypothesis refinements were checked for logical consistency, code was reviewed and verified for correctness, and figures were inspected for accuracy and scientific plausibility. We take responsibility for the final content of this work, including all text, claims, code, and figures produced with the assistance of generative AI tools.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide theoretical proofs, data-splitting protocols, training-only normalization procedures, checkpoint-selection criteria, inversion and decoding settings, and random seeds in the appendix.

REFERENCES

Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2StyleGAN: How to embed images into the StyleGAN latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019. URL https://arxiv.org/abs/1904.03189.

Jiaqi Bai, Jindou Jia, Yuxuan Hu, Gen Li, Xiangyu Chen, Tuo An, Kuangji Zuo, and Jianfei Yang. FLASH: Efficient visuomotor policy via sparse sampling. arXiv preprint arXiv:2605.15492, 2026. URL https://arxiv.org/abs/2605.15492.

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 *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.010. URL https://www.roboticsproceedings.org/rss21/p010.html.

Haidong Cao, Wenjun Cao, Quanhao Li, Sicheng Xie, Zhiying Du, Jiaqi Leng, Zuxuan Wu, and Yu-Gang Jiang. SegDiff: Segmented trajectory diffusion for consistent and adaptive robot manipulation. arXiv preprint arXiv:2607.11027, 2026. URL https://arxiv.org/abs/2607.11027.

Qingyi Chen, Joseph Ruan, and Zachary Kingston. Safe real-time policy steering via noise-space trajectory optimization for one-step generative policies. arXiv preprint arXiv:2609.21220, 2026. URL https://arxiv.org/abs/2609.21220.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://arxiv.org/abs/1806.07366.

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. *arXiv preprint arXiv:2303.04137*, 2023.

Haoran Geng, Feishi Wang, Songlin Wei, Yuyang Li, Bangjun Wang, Boshi An, Haozhe Lou, Charlie Tianyue Cheng, Peihao Li, Haozhe Chen, Yutong Liang, Yuxi Qian, Jiageng Mao, Weikang Wan, Yiran Geng, Mingtong Zhang, Jiangran Lyu, Siheng Zhao, Jiazhao Zhang, Chaoyi Xu, Jialiang Zhang, Chengyang Zhao, Haoran Lu, Yufei Ding, Ran Gong, Yuran Wang, Yuxuan Kuang, Ruihai Wu, Baoxiong Jia, Hao Dong, Siyuan Huang, Yue Wang, Jitendra Malik, and Pieter Abbeel. RoboVerse: A unified platform, benchmark and dataset for scalable and generalizable robot learning. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.022. URL https://www.roboticsproceedings.org/rss21/p022.html.

Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. FFJORD: Free-form continuous dynamics for scalable reversible generative models. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJxgknCcK7.

Ernst Hairer, Syvert P. Nørsett, and Gerhard Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*, volume 8 of *Springer Series in Computational Mathematics*. Springer, 2 edition, 1993. doi: 10.1007/978-3-540-78862-1.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross-attention control. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=_CDixzkzeyb.

Trong-Bao Ho, Quang-Tan Nguyen, Thien-Loc Ha, Gia-Binh Nguyen, Viet-Thanh Nguyen, Long Dinh, Minh N. Vu, Duy M. H. Nguyen, An Thai Le, and Ngo Anh Vien. Start right, arrive right: Asynchronous execution via initial noise selection. arXiv preprint arXiv:2606.19774, 2026. URL https://arxiv.org/abs/2606.19774.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. RLBench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 2020. URL https://arxiv.org/abs/1909.12271.

Jindou Jia, Tuo An, Yuxuan Hu, Gen Li, Jingliang Li, Bohan Hou, Xiangyu Chen, Jiaqi Bai, Bofan Lyu, and Jianfei Yang. MARS policy: Multimodality only when it matters. arXiv preprint arXiv:2605.29766, 2026a. URL https://arxiv.org/abs/2605.29766.

Jindou Jia, Gen Li, Xiangyu Chen, Tuo An, Yuxuan Hu, Jingliang Li, Xinying Guo, and Jianfei Yang. Action-to-action flow matching. In *Proceedings of Robotics: Science and Systems*, 2026b. doi: 10.15607/RSS.2026.XXII.209. URL https://www.roboticsproceedings.org/rss22/p209.html.

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. *arXiv preprint arXiv:2505.21851*, 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. URL https://arxiv.org/abs/2605.13959.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.02747.

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*, volume 36, 2023. URL https://arxiv.org/abs/2306.03310.

Junjie Lu, Xinyao Qin, Yuhua Jiang, Kaixin Wang, Chuheng Zhang, Bin Liang, Jun Yang, Min Xu, and Li Zhao. UniSteer: Unified noise steering for efficient human-guided VLA adaptation. arXiv preprint arXiv:2605.10821, 2026. URL https://arxiv.org/abs/2605.10821.

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.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1678–1690. PMLR, 2022. URL https://proceedings.mlr.press/v164/mandlekar22a.html.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. NULL-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6038–6047, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Mokady_NULL-Text_Inversion_for_Editing_Real_Images_Using_Guided_Diffusion_Models_CVPR_2023_paper.html.

Michael Murray, Daphne Chen, Simran Bagaria, Dean Fortier, Tess Hellebrekers, Galen Mullins, Harshavardhan Gajarla, Oier Mees, Maya Cakmak, and Andrey Kolobov. FlowDAgger: Human-in-the-loop adaptation of generative robot policies in latent space. arXiv preprint arXiv:2607.08877, 2026. URL https://arxiv.org/abs/2607.08877.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, 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. arXiv preprint arXiv:2503.14734, 2025. URL https://arxiv.org/abs/2503.14734.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023. URL https://arxiv.org/abs/2212.09748.

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

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. arXiv preprint arXiv:2506.01844, 2025. URL https://arxiv.org/abs/2506.01844.

Hikmet Simsir and Ozgur S. Oguz. Lagrangian perturbation diffusion steering: Latent reinforcement learning for generative policies. arXiv preprint arXiv:2606.01151, 2026. URL https://arxiv.org/abs/2606.01151.

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.

Andy Tang, William Chen, Andrew Wagenmaker, Chelsea Finn, and Sergey Levine. Improving robotic generalist policies via flow reversal steering. arXiv preprint arXiv:2606.13675, 2026. URL https://arxiv.org/abs/2606.13675.

Andrew Wagenmaker, Yunchu Zhang, Mitsuhiko Nakamoto, Seohong Park, Waleed Yagoub, Anusha Nagabandi, Abhishek Gupta, and Sergey Levine. Steering your diffusion policy with latent space reinforcement learning. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 258–282. PMLR, 2025. URL https://proceedings.mlr.press/v305/wagenmaker25a.html.

Bram Wallace, Akash Gokul, and Nikhil Naik. EDICT: Exact diffusion inversion via coupled transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22532–22541, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Wallace_EDICT_Exact_Diffusion_Inversion_via_Coupled_Transformations_CVPR_2023_paper.html.

Pengfei Zhang, Teng Sun, and Xianchao Xiu. Beyond noise steering: Dual-latent space reinforcement learning for generative robot policy. arXiv preprint arXiv:2609.11270, 2026. URL https://arxiv.org/abs/2609.11270.

A Theoretical Details

A.1 Conditional Transport and KL Preservation

Fix an observation-history condition c. Let A denote a complete normalized action chunk and let $\gamma = \mathcal{N}(0, I_D)$. For an invertible generation map F_c, define

$$p_\theta(\cdot \mid c) = (F_c)_{\#}\gamma, \qquad q_E(\cdot \mid c) = (F_c^{-1})_{\#}p_E(\cdot \mid c). \tag{10}$$

Here $T_{\#}p$ is the distribution of $T(X)$ for $X \sim p$. Assume F_c is a diffeomorphism on the relevant full-dimensional domains, the distributions admit densities, and the required absolute-continuity and finite-divergence conditions hold. Abbreviating the conditional densities, change of variables gives

$$q_E(z) = p_E(F_c(z))|\det JF_c(z)|, \qquad \gamma(z) = p_\theta(F_c(z))|\det JF_c(z)|. \tag{11}$$

The Jacobians cancel in the density ratio, so

$$D_{\mathrm{KL}}(q_E\|\gamma) = \int q_E(z)\log\frac{p_E(F_c(z))}{p_\theta(F_c(z))}\,dz \tag{12}$$

$$= \int p_E(A)\log\frac{p_E(A)}{p_\theta(A)}\,dA = D_{\mathrm{KL}}(p_E\|p_\theta). \tag{13}$$

This applies the invariance of KL under a common invertible transformation; it does not introduce a new invariance theorem. For a learned source law $\widehat{q}_\phi$ and $\widehat{p}_\phi = (F_c)_{\#}\widehat{q}_\phi$, the same argument yields

$$D_{\mathrm{KL}}(p_E\|\widehat{p}_\phi) = D_{\mathrm{KL}}(q_E\|\widehat{q}_\phi). \tag{14}$$

Thus exact conditional source matching recovers the expert conditional action law without updating F_c.

A.2 Pooled Mismatch and Conditional Correspondence

Let $q_E(c, z) = \rho(c)q_E(z \mid c)$ and $\overline{q}_E(z) = \int q_E(c, z)\,dc$. For any observation-independent reference law $r(z)$ with the required domination and finite quantities,

$$\log\frac{q_E(z \mid c)}{r(z)} = \log\frac{q_E(z \mid c)}{\overline{q}_E(z)} + \log\frac{\overline{q}_E(z)}{r(z)}. \tag{15}$$

Integration against $q_E(c, z)$ gives

$$\mathbb{E}_{c\sim\rho}D_{\mathrm{KL}}(q_E(\cdot \mid c)\,\|\,r) = I_{q_E}(C; Z) + D_{\mathrm{KL}}(\overline{q}_E\|r). \tag{16}$$

Applying the preceding conditional change of variables gives the corresponding average action-space identity. Among unrestricted observation-independent laws, $r = \overline{q}_E$ minimizes this expression, leaving $I_{q_E}(C; Z)$. Relative to $r = \gamma$, its ideal KL improvement is precisely $D_{\mathrm{KL}}(\overline{q}_E\|\gamma)$.

B Additional Experiments on Action Inversion

B.1 Datasets, Models, and Evaluation Coordinates

Matched FM-UNet and FM-DiT evaluations. The cross-architecture experiments use two public LIBERO-Goal tasks: `open_the_middle_drawer_of_the_cabinet` (OpenDrawer) and `put_the_bowl_on_the_plate` (BowlOnPlate). Each task contains 50 demonstration episodes, which we partition into 35 training, 5 validation, and 10 test episodes using split seed 20260915. This is our episode-level research split. Normalization is fitted on training episodes, and checkpoints are selected using validation flow-matching loss. For each model–task pair, we sample 128 complete test windows without replacement, balancing their allocation across episodes. Each action chunk contains 16 steps and 7 action channels ($D = 112$); windows requiring terminal padding are excluded. Each model is trained for 10,000 updates with batch size 32, AdamW, learning rate 10^{-4}, weight decay 10^{-6}, and a cosine schedule following 200 warm-up updates. These experiments use training seed 0; the backbone parameter counts are not matched.

Pretrained π_0 evaluations. We evaluate the existing `pi0_libero` checkpoint on OpenDrawer, BowlOnPlate, and the LIBERO-90 task `KITCHEN_SCENE7_open_the_microwave` (OpenMicrowave). Each task contributes one complete window from each of 50 episodes. Checkpoint preprocessing and normalization are retained. The model integrates the full 50×32 tensor throughout generation and inversion. For the main distribution statistics, we retain the source coordinates corresponding to the robot's 7 action channels at all 50 prediction steps, giving $D = 350$; the other 25 channels are padding channels.

Separate numerical calibration. We additionally use an earlier frozen FM-UNet checkpoint on CloseBox, with 128 complete 16×9 windows from 50 demonstration episodes. This experiment provides a detailed solver-convergence study and is separate from the LIBERO configurations in the main-text source-distribution table. Its checkpoint normalization is left unchanged.

B.2 Source Sets and Numerical Protocol

For an observation condition c_i, let $F^{\mathrm{ref}}_{c_i}$ be a fixed numerical forward reference and let $\widetilde{I}_{c_i,K}$ denote inversion with K integration steps. We construct the three source sets as

$$\begin{aligned} \mathrm{G}: &\quad z_i \sim \mathcal{N}(0, I), \\ \mathrm{M}: &\quad \widehat{z}_i = \widetilde{I}_{c_i,K}\big(F^{\mathrm{ref}}_{c_i}(z_i)\big), \\ \mathrm{E}: &\quad z_i^E = \widetilde{I}_{c_i,K}(A_i^E). \end{aligned} \tag{17}$$

The model-generated actions in M are not filtered by task success. G and M form a paired numerical control: the initial source is known, and generation and inversion use the same observation condition. E instead starts from the recorded expert action paired with that condition. All model parameters and normalization statistics remain fixed during evaluation. We use explicit Euler, explicit midpoint, and classical fourth-order Runge–Kutta (RK4) integration where specified below (Hairer et al., 1993). RK4 uses four velocity-field evaluations per integration step; RK4-K therefore denotes K steps, not four total steps. The main-table FM-UNet and FM-DiT results use FP32 RK4-256 for both directions, while π_0 uses FP32 RK4-1280. Within each inverse-step sweep, the forward reference is held fixed. For a d-dimensional evaluated coordinate view, define $r(u, v) = \|u - v\|_2/\sqrt{d}$. We separately measure known-source recovery, $r(\widehat{z}_i, z_i)$, and expert-action reconstruction, $r(F^{\mathrm{ref}}_{c_i}(z_i^E), A_i^E)$. We also compare expert sources at successive inverse resolutions. For an expert-neighborhood control, we form known sources $z_i^{\mathrm{near}} = \widetilde{I}_{c_i,K_{\mathrm{ref}}}(A_i^E) + 0.1\epsilon_i$, with $\epsilon_i \sim \mathcal{N}(0, I)$, generate their actions, and test recovery of these known sources.

B.3 Distribution Diagnostics

All source groups retain their original coordinates without group-specific standardization, rescaling, or moment matching. For each configuration, we draw an independent Gaussian reference $\{g_i\}_{i=1}^N$ with the same dimension and sample count as the source groups. We use $L = 128$ shared random

Table 5: CloseBox numerical calibration with 128 complete 16×9 chunks. Errors are mean per-sample RMSE. Source change compares expert inversions at consecutive resolutions listed in the table.

Inverse steps	Gaussian recovery	Near-expert recovery	Action reconstruction	Source change
32	6.117×10^{-5}	2.582×10^{-4}	3.613×10^{-6}	–
64	5.434×10^{-6}	2.111×10^{-5}	3.116×10^{-7}	2.549×10^{-4}
128	1.607×10^{-6}	3.212×10^{-6}	1.186×10^{-7}	2.107×10^{-5}
256	1.006×10^{-6}	2.146×10^{-6}	3.941×10^{-8}	3.242×10^{-6}

unit directions $\{u_\ell\}_{\ell=1}^{L}$. If $a_{(i)}^{\ell}$ and $b_{(i)}^{\ell}$ denote the sorted projections of a source group and the reference onto u_ℓ, respectively, we compute

$$\widehat{\mathrm{SW}}_2 = \left[\frac{1}{LN} \sum_{\ell=1}^{L} \sum_{i=1}^{N} \left(a_{(i)}^{\ell} - b_{(i)}^{\ell} \right)^2 \right]^{1/2}. \tag{18}$$

The reported mean squared source norm is $N^{-1} \sum_i \|z_i\|_2^2$. Because G and the reference are independently sampled, their empirical sliced Wasserstein distance is generally nonzero. All three groups share the same reference and projection directions within a configuration. For source visualizations, PCA is fitted once to the pooled G/M/E samples, and the same centering and projection are used for all groups.

B.4 Square Starting-Point Visualization

Figure 3 uses a separate RoboMimic Square evaluation with a frozen FM-UNet. The density groups contain 40 standard Gaussian starting points and 40 expert-inverted starting points, each representing a complete 16×7 action chunk ($D = 112$). Gaussian and EIP-Gaussian rollouts are paired by initial observation and frozen checkpoint; the plotted policy samples are taken at the first action query.

Shared projection and density estimation. PCA is fitted once to the pooled 40 Gaussian and 40 expert-inverted vectors, with a common mean. The EIP samples are projected using this fitted transformation and are not included in the PCA fit. The first two components explain 4.5% and 4.2% of the pooled variance, respectively. Both projected groups use an isotropic Gaussian KDE with the same bandwidth $h = 1.19496$, obtained from the pooled projected standard deviations and the factor $40^{-1/6}$. Densities are evaluated on a shared 320×320 grid with both axes spanning approximately $[-9.578, 9.578]$. The color ramps share logarithmic density limits of 10^{-4} and 0.031122; values below the lower limit use the background color. Contour thresholds are computed from cumulative KDE mass on this grid: panels (a,b) display 50% and 90% contours, and panel (c) displays 90% contours. Top and right curves show one-dimensional KDE marginals with shared bandwidth and scales.

B.5 CloseBox Solver Convergence

Table 5 reports the CloseBox calibration with a fixed FP32 RK4-256 forward reference. Increasing inverse resolution reduces both known-source recovery and expert-action reconstruction errors. The recovered expert coordinates also stabilize under step refinement.

The RK4-128 and RK4-256 forward outputs differ by 1.709×10^{-7} mean per-sample action RMSE. Meanwhile, the expert-source sliced Wasserstein distance changes only from 0.403292 at 32 inverse steps to 0.403282 at 256 steps. At 256 steps, the corresponding G and M distances are both 0.207777 at the reported precision. Thus, the source-distribution discrepancy in this particular checkpoint–dataset pair persists after substantial numerical refinement; it is not explained solely by coarse inversion. To reduce the contribution of correlated windows, we also retain one window per CloseBox episode, giving 50 samples, and repeat the distribution diagnostics with three statistical random seeds. Expert distances range from 0.425 to 0.448, compared with 0.312 to 0.355 for the

Table 6: Separate four-window π_0 OpenMicrowave pilot. The forward reference is fixed at FP32 RK4-1280. Errors are global RMSE over the 50×7 action-coordinate view.

Inverse steps	Gaussian recovery	Near-expert recovery	Action reconstruction	Source change
320	1.8810×10^{-2}	1.8008×10^{-2}	1.925×10^{-5}	–
640	2.499×10^{-3}	2.351×10^{-3}	1.852×10^{-6}	1.7260×10^{-2}
1280	5.30×10^{-4}	3.94×10^{-4}	1.055×10^{-7}	2.753×10^{-3}

Gaussian controls. These ranges describe the repeated diagnostics, not confidence intervals. In the LIBERO test evaluations, 128 windows come from only 10 held-out episodes. They are therefore not 128 independent demonstrations.

B.6 Numerical Precision in a Separate π_0 Pilot

We use a separate four-window OpenMicrowave pilot to examine numerical precision. These windows are not the 50-sample main-table evaluation. FP32 computation retains the same BF16-rounded parameter values while increasing the precision of model computation and cached features; it does not recover parameter information discarded during BF16 loading. With a fixed RK4-320 forward reference within each precision setting, changing computation from BF16 to FP32 reduces known-source recovery RMSE from 0.17577 to 0.04960 and expert-source change between 160 and 320 inverse steps from 0.22250 to 0.06020. The forward references themselves differ across precision settings, so this comparison is not inversion of an identical numerical map. A separate FP32 sweep holds a higher-resolution RK4-1280 forward reference fixed, as shown in Table 6.

B.7 Compatibility with Few-Step Decoders

Finally, we freeze the saved high-resolution expert sources and their observation conditions, and change only the decoder configuration. Table 7 uses the CloseBox calibration and the separate four-window π_0 pilot.

Table 7: Reconstruction from fixed high-resolution expert sources using different decoders. The UNet rows use mean per-sample RMSE; π_0 rows use global RMSE. The last column restricts evaluation to the executed time segment.

Model	Decoder	Full-horizon RMSE	Executed-segment RMSE
UNet	FP32 RK4-256	3.941×10^{-8}	3.794×10^{-8}
	FP32 midpoint-10	0.006534	0.006921
	FP32 midpoint-30	0.000436	0.000456
π_0	FP32 RK4-1280	1.055×10^{-7}	8.117×10^{-8}
	FP32 Euler-10	0.104546	0.085044
	BF16 Euler-10	0.103195	0.086503

For CloseBox, the full view is 16×9 and the executed segment is indices $[7 : 15)$ under zero-based indexing. For π_0, the full-horizon view retains the 7 action channels at all 50 steps, while the executed view retains the first 10 steps. The FP32 RK4 versus FP32 Euler comparison holds arithmetic precision fixed; the BF16 Euler row additionally changes computation precision. Thus, accurate reconstruction with a high-resolution reference does not automatically transfer to the deployed few-step decoder. This motivates checking decoder compatibility when preparing expert-source supervision. These are offline reconstruction measurements, rather than closed-loop task-success evaluations.

Table 8: Completed LIBERO-Goal source-guidance controls. Entries are successes out of 150 rollouts (50 initial states, three sampling seeds). All source selections use the training bank.

Model	Task	Gaussian	Random bank	Retrieval
UNet	OpenDrawer	55/150	123/150	120/150
	BowlOnPlate	134/150	134/150	135/150
DiT	OpenDrawer	70/150	60/150	129/150
	BowlOnPlate	106/150	135/150	144/150

Table 9: Retrieval minus random-bank success (percentage points), with paired initial-state bootstrap intervals.

Task	Backbone	Difference	95% interval
OpenDrawer	UNet	-2.0	[-16.0, 12.0]
OpenDrawer	DiT	+46.0	[31.3, 59.3]
BowlOnPlate	UNet	+0.7	[-9.3, 10.0]
BowlOnPlate	DiT	+6.0	[-2.0, 14.0]

C Observation-Based Source Guidance

Protocol. The LIBERO-Goal OpenDrawer and BowlOnPlate controls use the same frozen FM-UNet and FM-DiT checkpoints and episode splits as the cross-architecture experiments. The bank contains all eligible unpadded training windows: 4,354 for OpenDrawer and 2,762 for BowlOnPlate. Sources and conditions remain paired in the training bank. At every policy update, the Gaussian baseline draws a fresh standard Gaussian source. Random expert sampling draws uniformly over bank windows. Retrieval selects the nearest training condition under RMS distance after coordinate-wise standardization using training-bank statistics. Keys include the complete frozen visual-history and proprioceptive condition. All methods use FP32 midpoint-10 decoding, complete 16×7 chunks, and execution indices $[1 : 9)$. Observations contain the latest two actual environment frames. Each method uses the same 50 packaged initial states and sampling seeds 0, 1, and 2, with environment seed 7, 10 settling steps, and at most 300 task steps.

C.1 Paired Uncertainty for Observation-Based Retrieval

The three sampling seeds in Table 8 reuse the same 50 initial states. We therefore first average each method's success indicator across sampling seeds within an initial state, then bootstrap the 50 paired state-level differences. Table 9 uses 10,000 bootstrap resamples, random seed 20260926, and the 2.5th and 97.5th percentiles. These are descriptive, pointwise intervals conditional on the frozen model and evaluation protocol; no multiple-comparison adjustment is applied. Repeated sampling seeds are not 150 independently sampled initial states.

The retrieval advantage is clearest for OpenDrawer with FM-DiT. For the other three configurations the intervals include zero. Thus the comparison supports a task-dependent value of observation–source correspondence, rather than a uniform advantage of nearest-neighbor retrieval over marginal expert-source sampling.

D Conditional Source Models and Training

D.1 Supervision and Normalization

For the LIBERO experiments, the frozen visual encoder produces a 2,064-dimensional condition from two causal frames, two RGB views, and proprioception. Each inverse target is a complete 16×7 native source. The base checkpoint's train-fitted action and proprioceptive normalization are retained without refitting. Source-model condition means and standard deviations are computed only from its training cache, with standard deviations floored at 10^{-5}.

D.2 Objectives and Parameterizations

Deterministic regression. The point baseline minimizes $\mathbb{E}\|f_\phi(c) - \widetilde{z}_E\|_2^2/D$. Its unrestricted population optimum is the conditional mean.

Conditional diagonal Gaussian. The model predicts $\mu_\phi(c)$ and $\ell_\phi(c) = \log \sigma_\phi(c)$ for all $D = 112$ coordinates. The implemented log-standard-deviation range is $[-5, 2]$. Up to an additive constant, the training loss is

$$\mathcal{L}_{\mathrm{CG}} = \mathbb{E}\frac{1}{D}\sum_{j=1}^{D}\left[\frac{(\widetilde{z}_{E,j} - \mu_{\phi,j}(c))^2}{2\sigma_{\phi,j}(c)^2} + \log \sigma_{\phi,j}(c)\right]. \tag{19}$$

Sampling uses $z = \mu_\phi(c) + \sigma_\phi(c) \odot \epsilon$ with independent $\epsilon \sim \mathcal{N}(0, I_D)$.

Conditional Flow. For independent $\epsilon \sim \mathcal{N}(0, I_D)$ and $s \sim \mathcal{U}[0, 1]$, define

$$z_s = (1 - s)\epsilon + s\widetilde{z}_E, \qquad \mathcal{L}_{\mathrm{CF}} = \mathbb{E}\frac{1}{D}\|v_\phi(z_s, s, c) - (\widetilde{z}_E - \epsilon)\|_2^2. \tag{20}$$

The source transport is integrated with 16 left-Euler steps:

$$z^{(0)} = \epsilon, \qquad z^{(k+1)} = z^{(k)} + \frac{1}{16}v_\phi(z^{(k)}, k/16, c), \quad k = 0, \ldots, 15. \tag{21}$$

The final complete source is decoded by the unchanged action policy. An unconditional Flow control, where used, has the same network but receives a zero conditioning vector during training and sampling.

Architectures. The conditional source Flow uses an author-implementation ConditionalUnet1D with channels $(128, 256, 512)$, a 128-dimensional time embedding, kernel size 5, eight GroupNorm groups, and condition-dependent scale prediction. The conditional Gaussian uses three hidden linear layers with LayerNorm and SiLU. Hidden width 3,412 approximately matches the source-Flow parameter budget: 31,121,076 versus 31,118,471 parameters for the current $D = 112$, condition-dimension-2,064 configuration. The deterministic MLP uses hidden width 3,436 and has 31,119,964 parameters. The unconditional Flow has the same 31,118,471 parameters as the conditional Flow and receives a zero condition vector.(Appendix F).

D.3 Optimization and Model Selection

Both source priors use AdamW, learning rate 10^{-4}, weight decay 10^{-6}, batch size 32, 5,000 optimizer updates, 200 warm-up updates followed by cosine decay, and gradient norm clipping at 1.0. Validation diagnostics are recorded every 500 updates. Source-model validation losses are diagnostics of the training objective. Closed-loop success is measured separately by decoding predicted starting points through the frozen action generator.

E Temporal Thinning before Expert Inversion

For each episode and split, let $(w_0, \ldots, w_{n_e-1})$ be the eligible complete windows sorted by their start indices. With stride $s = 8$, retain

$$\mathcal{J}_e = \{0, s, 2s, \ldots, \lfloor (n_e - 1)/s \rfloor s\} \cup \{n_e - 1\}. \tag{22}$$

The terminal eligible window is included only once. Selection is performed *before* numerical inversion. Every selected window retains its full observation history, complete action chunk, episode identifier, and start index. Splits are assigned at the episode level before selection.

Table 10: Verified eligible-window counts before and after stride-8 selection. Entries are dense → retained counts. These quantify sample reduction, not measured wall-clock speedup.

LIBERO-90 task	Train	Validation	Test	Total
OpenMicrowave	$4612 \to 620$	$679 \to 90$	$1373 \to 186$	$6664 \to 896$
FrontBowlOnPlate	$4271 \to 577$	$680 \to 92$	$1209 \to 164$	$6160 \to 833$

Table 11: Simulation protocols. Splits count demonstration episodes; C is condition dimension. RK4 columns list inverse steps for UNet / DiT, not action-decoder steps. Limit counts environment actions.

Task	Train/val/ test	C	Chunk	RK4 U / D	Limit
LIBERO-90 (each task)	35/5/10	2064	16×7	256 / 256	400
RoboVerse CloseBox	35/5/10	34	16×9	512 / 256	250
RoboVerse PickCube	35/5/10	32	16×9	1024 / 256	250
RLBench ReachTarget	70/10/20	40	16×8	2048 / 2048	400
RLBench PushButton	70/10/20	92	16×8	1024 / 256	400
RoboMimic Square	140/20/40	46	16×7	256 / 256	400
RoboMimic Lift	140/20/40	38	16×7	256 / 256	400

For the two audited tasks, selection reduces the number of inverted windows by approximately 86.6% and 86.5%, respectively. The corresponding ratios, approximately $7.4\times$, are window-count ratios. Uniform training over retained windows also changes the empirical window distribution.

E.1 Matched StackCube Cache Pilot

Table 4 reports a separate StackCube EIP-Flow pilot comparing the full cache of 15,264 windows with its stride-8 subset of 2,041 windows. The subset preserves the stored conditions and complete inversion targets. Both configurations use 5,000 prior-training updates and $p8{+}a4$ inference, with the same frozen action generator within the comparison. Success is pooled over two evaluation seeds of 50 episodes each, yielding 78% for the full cache and 86% for the thinned cache.

The reported training times, 195.0 s and 182.8 s, sum the 30 epoch-level timers for optimization and validation. They exclude initialization, checkpoint writing, and additional diagnostics. The observed difference is 12.2 s (6.3%) at the matched update budget. Because the stride-8 cache is extracted from an existing inversion cache, this pilot does not measure inversion-time speedup.

F Simulation Experimental Setup

F.1 Dataset Splits, Inputs, and Numerical Settings

For LIBERO-90, each task's 50 demonstration episodes are split into 35/5/10 training/validation/test episodes. The complete action chunk is 16×7. Inverse caches use FP32 RK4-256 and stride 8, retaining each episode's final eligible window. Base policies are trained per task. For the state-input benchmarks, Table 11 records the task-specific dimensions and inverse resolutions. Each condition concatenates two observations. RoboMimic uses end-effector position and quaternion, gripper position, and object state from the proficient-human data. RLBench uses robot proprioception and named physical object states; planning waypoints are not policy inputs. RoboVerse uses native robot and object state. Action and state normalization are fitted only on the training episodes. State-input episode splits use seed 20260923. All inverse caches retain complete chunks and use stride 8 plus the final eligible window.

The state-input bases use the same UNet and DiT velocity-network implementations with benchmark-specific input/output dimensions, 10,000 training updates, batch size 32, AdamW at 10^{-4} with weight decay 10^{-6}, and training seed 0. Base checkpoints are selected by fixed-noise validation flow-matching loss evaluated every 500 updates (including initialization). Learned priors use the shared training implementation and the final 5,000-update checkpoint.

Numerical qualification. The cross-benchmark cache pipeline checks representative training conditions for inverse-step doubling, expert-action reconstruction, and known-Gaussian-source recovery before constructing the cache. The nominal thresholds are respectively 10^{-3}, 10^{-4}, and 10^{-3} normalized RMSE. Some state-input models require the higher inverse resolutions in Table 11.

Environment execution. Each query predicts 16 actions and executes indices $[1 : 9)$ before replanning. LIBERO uses environment seed 7, 10 settling actions, and 50 packaged initial states, with history updated after each environment step. State-input evaluations use reset seed $10000 + i$ and policy-noise seed $20000 + i$ for trial $i = 0, \ldots, 49$. RoboVerse and RLBench use a fresh simulator per trial to avoid dependence on previous rollouts. Across the completed state-input comparisons, recorded initial state vectors and reset/noise seeds match the baseline; the saved numerical pairing tolerance is 10^{-3} and the observed maximum initial-state discrepancy is zero. RLBench retains the native success predicate but uses a custom action adapter with seven absolute arm-joint targets and a thresholded gripper command, including a collection-aligned closure macro. RoboVerse uses nine native absolute joint targets, including two finger targets. RoboMimic uses the native controller and success predicate associated with its dataset.

G Measured Inference Cost

We measure inference latency on LIBERO-90 Open Drawer & Place Bowl using an NVIDIA RTX 5090, batch size 1, and FP32 with TF32 disabled. Each configuration uses 10 warm-up queries followed by 30 timed queries, with GPU synchronization around the timed region. Timing includes starting-point sampling, action generation, and output transfer to CPU from cached observation conditions, excluding observation encoding, preprocessing, and simulation. We compare Base FM and EIP-Gaussian with 4 or 10 action steps, and EIP-Flow with $p8{+}a4$ or $p16{+}a10$. Action and prior integration use midpoint and Euler solvers, respectively. Success rates use 50 matched initial states per configuration.