Source-Lifted Flow Matching for Intervenable Multimodal Imitation

He Zhang[1], Ying Sun[1]*, Pengteng Li[1], Ziyang Chen[1],
Yiren Zhao[1], Ziyang Rao[1], Weiyu Guo[1], Yandong Guo[2], Hui Xiong[1]*

[1]The Hong Kong University of Science and Technology (Guangzhou)
[2]AI[2] Robotics

Abstract

Flow-matching policies are promising for imitation learning because they model complex multimodal action distributions. However, their stochasticity is largely passive: repeated sampling may yield diverse behaviors, but users cannot directly choose among valid continuations from the same state. We propose **Source-Lifted Flow Matching (SL-FM)**, a source-intervenable flow-matching policy that exposes such a handle while keeping the velocity field shared and latent-free. The handle selects only the source endpoint of the conditional flow, not a mode-specific field, preserving the standard formulation while avoiding decomposition into separate mode-conditioned dynamics. The core mechanism is **Orthogonal Source Lifting**, designed to prevent path-crossing ambiguity. Instead of partitioning target actions by mode, SL-FM lifts handle-specific sources into auxiliary orthogonal coordinates and keeps targets in the original action subspace. This preserves the demonstrated action distribution while allowing one shared field to carry different branches without merging at crossings. To keep handles usable across states, we learn a state-dependent source mixture end to end and use a responsibility floor, giving each handle weak supervision and mitigating dead modes. Experiments on crossing-flow diagnostics and robot-control benchmarks show that SL-FM converts passive source randomness into an actionable intervention variable. It removes crossing-induced composite trajectories, changes future routes in 91.1% of matched-prefix interventions, and achieves strong free-deployment performance, with improvements in several benchmark settings. Overall, source geometry provides actionable multimodal control without conditioning the velocity field on the selected mode.

Introduction

Recent advances in imitation learning have increasingly leveraged generative policies to model complex, multimodal action distributions (Chi et al. 2023; Pearce et al. 2023; Jiang et al. 2025). Unlike conventional behavioral cloning methods that often assume a unimodal Gaussian policy (Florence et al. 2022; Shafiullah et al. 2022), diffusion and flow-matching policies can represent multiple plausible actions for the same observation (Chi et al. 2023; Lipman et al. 2023; Jiang et al. 2025), which is essential in robot tasks with several valid strategies (Jia et al. 2024; Chi et al. 2023). For example, in obstacle avoidance, demonstrations may pass an obstacle from either side; a good policy should preserve both alternatives rather than collapse them into an averaged action.

*Corresponding author.

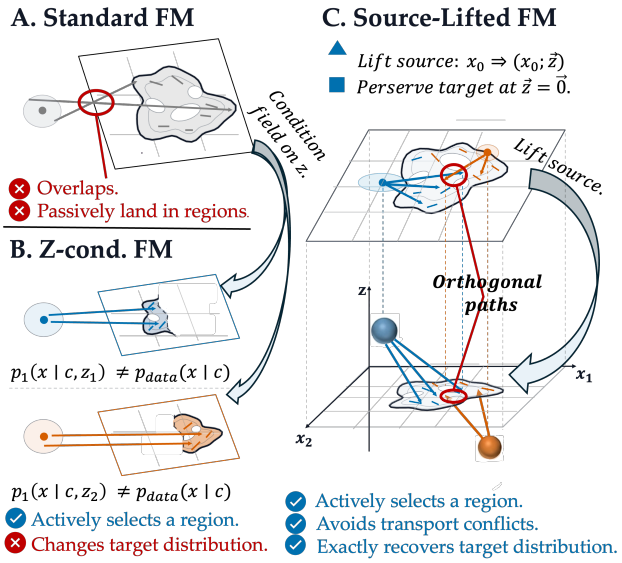

Figure 1: Main idea of Source-Lifted Flow Matching. Standard flow matching passively samples multimodal behaviors, while mode-conditioned fields may split both the target distribution and velocity field.. SL-FM instead lifts handle-specific sources into orthogonal coordinates, keeps all targets in the original action subspace, and uses one shared latent-free field to preserve source identity through crossings.

However, representing diverse behaviors is not the same as controlling them. Existing flow-matching policies usually use source noise as a passive sampling mechanism (Lipman et al. 2023; Jiang et al. 2025): resampling the source may produce different rollouts, but the user cannot directly choose which continuation will be realized from a fixed decision state. This distinction matters for downstream planning, human-in-the-loop control, and robot tasks where several valid futures share the same prefix. A policy that sometimes goes left and sometimes goes right is useful; one that lets a planner choose the continuation from the same local state is more useful.

A straightforward way to expose such a choice is to condition the policy or the velocity field on a discrete mode variable (Guo and Schwing 2025; Zhai et al. 2025). While effective, this changes the nature of the model. A velocity field of the form $v_\theta(x, t, s, z)$ can resolve multimodality by learning separate mode-specific dynamics, making the dis-

crete input an expert selector rather than an intervention on the source of a single shared flow. Our goal is different: we ask whether a conditional flow-matching policy can retain one shared velocity field while exposing a source handle that can be selected at test time.

To this end, we introduce **Source-Lifted Flow Matching (SL-FM)**, a source-intervenable flow-matching policy for multimodal imitation learning. SL-FM assigns each state a small set of source handles. In free deployment, the handle is sampled from a learned state-dependent prior. Under intervention, it can be set externally at a decision state. Crucially, the handle only determines where the conditional flow starts. The velocity field receives the intermediate point, time, and observation, but never the discrete handle itself. This preserves the standard conditional flow-matching formulation and forces behavior selection to arise from source geometry rather than from mode-conditioned subfields (Figure 1).

The key technical challenge is that source-only handles are not automatically reliable. If two multimodal paths cross in action space, a shared field may lose the identity of the chosen source branch and average incompatible velocities near the crossing. We address this problem with **Orthogonal Source Lifting**. Instead of partitioning the demonstrated action distribution into $p(a \mid s, z)$, SL-FM lifts handle-specific sources into auxiliary orthogonal coordinates while keeping every target action in the original action subspace. The target action distribution is therefore unchanged, but different handles remain separated during transport, allowing one latent-free field to carry different branches without losing identity at crossings. We further learn a state-dependent source mixture end to end and introduce a responsibility floor, which gives every handle weak transport supervision and mitigates dead modes under redundant handle counts.

Our experiments evaluate both ordinary imitation performance and the proposed intervention interface. On a crossing-flow diagnostic, SL-FM removes composite trajectories caused by path crossings and enables direct branch selection through the source handle. On D3IL Avoiding, same-prefix counterfactual interventions show that changing only the local source handle redirects future behavior in 91.1% of matched pairs. Across multimodal robot-control benchmarks, including D3IL and PushT, SL-FM maintains strong free-deployment performance and achieves the best average score among same-harness mechanism-matched baselines. These results show that source geometry can turn passive flow-matching randomness into an actionable multimodal control interface without conditioning the velocity field on the selected mode.

We summarize our **contributions** as follows:

- We formulate source-intervenable conditional flow matching for imitation learning, where a discrete handle selects the source endpoint but never conditions the velocity field.
- We identify path-crossing identity collapse as a failure mode of source-only intervention in shared flow fields and introduce Orthogonal Source Lifting to preserve source identity through crossings.
- We learn a state-dependent source mixture with a responsibility floor to keep redundant handles trainable and mitigate dead modes.
- We evaluate SL-FM using both free-deployment benchmarks and same-prefix interventions, demonstrating an actionable source interface with strong imitation performance.

Related Work

Generative policies for multimodal imitation. Diffusion and flow-based policies have become strong generative models for multimodal imitation learning. Diffusion Policy, flow-matching policies, Streaming Flow Policy, and related implicit, tokenized, or score-based policies represent diverse action modes beyond unimodal Gaussian cloning (Chi et al. 2023; Lipman et al. 2023; Jiang et al. 2025; Florence et al. 2022; Shafiullah et al. 2022; Reuss et al. 2023; Pearce et al. 2023). Benchmarks such as D3IL and PushT further highlight the need to preserve multiple valid robot strategies rather than collapse them into averaged actions (Jia et al. 2024; Chi et al. 2023). This trend also appears in VLA and generalist robot policies, where systems such as π_0, $\pi_{0.5}$, and GR00T N1 use diffusion or flow-based action heads for continuous control (Black et al. 2025; Physical Intelligence et al. 2025; NVIDIA et al. 2025). These works establish generative action modeling as a strong tool for marginal coverage; SL-FM instead asks whether the source randomness can be made locally intervenable under a matched rollout prefix.

Policy steering and latent conditioning. Another line of work exposes control by steering, adapting, or conditioning generative policies. Inference-time methods such as DynaGuide guide pretrained diffusion policies with external dynamics or task objectives (Du and Song 2025). Adaptation and distillation methods, including DSRL, ReinFlow, GoldenStart, RFS, and related latent policy steering approaches, optimize noise variables, priors, entropy, residual actions, or latent actors for policy improvement (Wagenmaker et al. 2025; Zhang et al. 2025; Zhang, Sun, and Xiong 2026; Su et al. 2026; Ren et al. 2024; Park, Li, and Levine 2025). Variational Flow-Matching Policy is closer to our setting, using latent priors and mode-aware decoding for multimodal manipulation (Zhai et al. 2025). These methods are complementary, but they expose control through guidance, value optimization, latent adaptation, or mode-conditioned generation. SL-FM targets a stricter interface: the handle only chooses where the conditional flow starts, while the velocity field remains shared and latent-free.

Source and coupling design in generative flows. The source distribution and source–target coupling strongly shape flow-matching geometry. OT-CFM, minibatch OT, rectified or optimal flow methods, CPD, MM-FM, and modal coupling design priors or pairings to shorten transport, reduce crossings, or improve sample efficiency, mostly in vision, generic, or motion generation (Tong et al. 2024; Pooladian et al. 2023; Liu, Gong, and Liu 2023; Albergo and Vanden-Eijnden 2023; Issachar et al. 2025; Luo et al. 2026; Wang et al. 2025). Control tasks add sequential decisions, state-dependent branching, and closed-loop feedback, so better transport geometry

alone does not necessarily yield an actionable decision variable. SL-FM uses source structure for intervention rather than generation quality: it keeps the demonstrated target distribution intact, places the selectable handle only at the source, and tests whether changing that handle causally changes future behavior under a matched prefix.

Preliminaries

Notation. We consider an imitation dataset $\mathcal{D} = \{(s_i, a_i)\}_{i=1}^{N}$, where $s \in \mathcal{S}$ denotes the policy observation and $a \in \mathbb{R}^{d_a}$ denotes a demonstrated action or a flattened action chunk. Here d_a is the action dimension, $t \in [0, 1]$ for the artificial flow-matching time and the timesteps in tasks are indexed by h. The goal of imitation learning is to sample actions from the conditional demonstration distribution $p_{\text{data}}(a \mid s)$ while executing successfully in closed loop.

Conditional flow matching for actions. Conditional flow matching learns a velocity field $v_\theta(x_t, t, s)$ that transports a source endpoint $x_0 \sim p_0(\cdot \mid s)$ to a target endpoint $x_1 = a \sim p_{\text{data}}(\cdot \mid s)$ (Lipman et al. 2023; Tong et al. 2024). For the linear conditional path

$$x_t = (1 - t)x_0 + tx_1, \qquad u_t = x_1 - x_0, \tag{1}$$

the standard regression objective is:

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{(s,a)\sim\mathcal{D},\, x_0\sim p_0,\, t\sim\mathcal{U}[0,1]} \left[\|v_\theta(x_t, t, s) - u_t\|^2 \right]. \tag{2}$$

At deployment, one samples x_0 from the source and integrates the ODE:

$$\frac{dx_t}{dt} = v_\theta(x_t, t, s) \tag{3}$$

from $t = 0$ to $t = 1$. The results are then executed as actions.

When $p_{\text{data}}(a \mid s)$ is multimodal, a standard flow-matching policy can still produce diverse behaviors because different regions of the source space are transported to different action modes (Lipman et al. 2023; Jiang et al. 2025). This is sufficient for marginal multimodal sampling, but the induced partition of the source space is implicit (Jiang et al. 2025). Resampling the source may change the rollout, yet the user or planner is not given an explicit local variable that selects which continuation should occur.

From passive sampling to source intervention. Figure 2 illustrates the distinction. At a decision state, multiple demonstrated futures may share the same prefix and branch into different valid continuations. A standard flow-matching policy may sample either branch, but this choice is passive: it is determined by source noise rather than by an externally selectable handle (Jiang et al. 2025). We instead seek a source-intervenable policy with a discrete handle $z \in \{1, \ldots, K\}$ satisfying three requirements. First, in free deployment, sampling z should recover ordinary stochastic imitation behavior and preserve the target action marginal. Second, under intervention, a planner should be able to set $z = k$ at a local decision state while keeping the prefix fixed. Third, the velocity field must remain shared and latent-free, i.e., the model uses $v_\theta(\bar{x}_t, t, s)$ rather than $v_\theta(x_t, t, s, z)$.

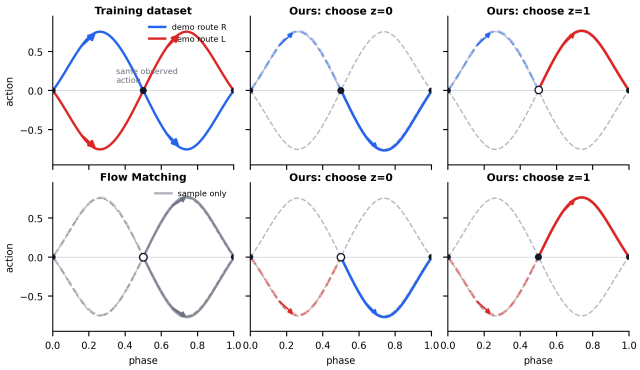

Figure 2: Same-prefix source-intervention problem. Standard flow matching can passively sample different continuations from a shared prefix, but the choice is not externally selectable. SL-FM asks for a local source handle z whose intervention selects among valid futures while the velocity field remains shared.

More formally, consider two rollouts that share the same environment seed, state-action prefix, and stochastic draws before a decision step h. A source intervention changes only the local handle, from $z = k$ to $z = k'$, and then resumes the same policy interface. Let $\rho(\cdot)$ denote an evaluation-time route or continuation label, not a training label. The desired behavior is that changing the handle can change the future continuation,

$$\rho(\xi_{h:}^{k}) \neq \rho(\xi_{h:}^{k'}), \tag{4}$$

while both branches remain valid task executions. This is stronger than showing diversity across independent rollouts: it asks whether the policy exposes a causal local variable for choosing among futures from the same prefix.

This problem setup motivates the construction in the next section. Since the handle is not allowed to condition the velocity field directly, any intervention effect must be carried by the geometry of the source. The remaining challenge is to make that source identity survive multimodal path crossings, where a shared field may otherwise average incompatible branch velocities.

Methodology

The previous section defines the desired interface: a discrete source handle should be selectable at a local decision state, while the velocity field itself remains shared and not explicitly conditioned on that handle. We now introduce **Source-Lifted Flow Matching (SL-FM)**, which realizes this interface through three components, as illustrated in Figure 3: a state-conditioned source prior, Orthogonal Source Lifting, and floor-weighted flow training.

Modeling the Source Prior

Standard flow matching starts generation from an unstructured source, such as a standard Gaussian. SL-FM instead models the source as a small state-conditioned mixture, where each mixture component corresponds to a selectable source handle (Bishop 1994; Jacobs et al. 1991). Let $a \in \mathbb{R}^{d_a}$ be the demonstrated action, and let $x_0^a \in \mathbb{R}^{d_a}$ denote the action-space source point before lifting. We use

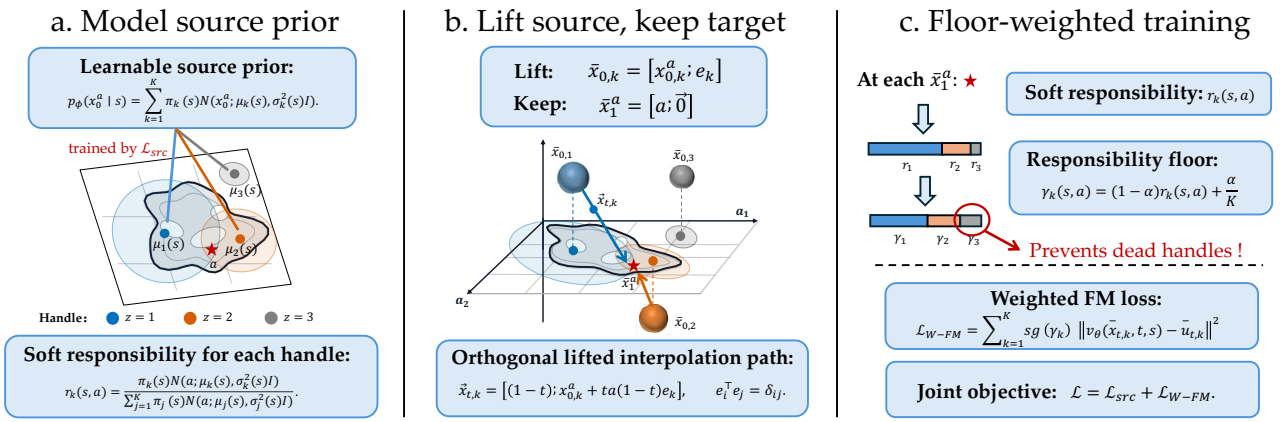

Figure 3: Method overview of Source-Lifted Flow Matching. SL-FM models a state-conditioned source prior, lifts handle-specific sources into auxiliary orthogonal coordinates while keeping all targets on the zero-lift plane, and trains one shared latent-free velocity field with floor-weighted responsibilities. For visual clarity, the schematic shows fixed anchors and omits scalar loss weights.

$z \in \{1, \dots, K\}$ to index K source handles. For each state s, a prior network predicts the parameters of an isotropic Gaussian mixture:

$$\{\pi_k(s), \mu_k(s), \sigma_k(s)\}_{k=1}^K,$$

where $\pi_k(s)$ are mixture weights with $\sum_k \pi_k(s) = 1$, $\mu_k(s) \in \mathbb{R}^{d_a}$ are component means, $\sigma_k(s) > 0$ are diagonal scales, and I_{d_a} is the d_a-dimensional identity matrix. The resulting source prior is:

$$p_\phi(x_0^a \mid s) = \sum_{k=1}^K \pi_k(s) \mathcal{N}\big(x_0^a; \mu_k(s), \sigma_k^2(s) I_{d_a}\big). \quad (5)$$

Free deployment samples $z \sim \pi_\phi(\cdot \mid s)$ and then samples from the selected Gaussian component,

$$x^a_{0,k} = \mu_k(s) + \sigma_k(s)\epsilon_k, \qquad \epsilon_k \sim \mathcal{N}(0, I_{d_a}). \quad (6)$$

Under intervention, the evaluator directly sets $z = k$ and uses the corresponding component. Thus, the handle first appears as a choice of source component, not as an input label to the velocity field.

We train this source prior to fit the demonstrated action distribution. Motivated by OT-CFM and related coupling-based flow-matching methods (Tong et al. 2024; Pooladian et al. 2023; Liu, Gong, and Liu 2023), and following the observation that flow matching tends to prefer short and geometrically simple transports, we encourage each Gaussian component to lie near the actions it is responsible for, instead of forcing all handles to start from a fixed uninformed Gaussian. We therefore fit the mixture to demonstrated actions:

$$\mathcal{L}_{src} = -\mathbb{E}_{(s,a)\sim\mathcal{D}} \log \sum_{k=1}^K \pi_k(s) \mathcal{N}\big(a; \mu_k(s), \sigma_k^2(s) I_{d_a}\big). \quad (7)$$

This density model induces a soft responsibility for each handle:

$$r_k(s,a) = \frac{\pi_k(s) \mathcal{N}\big(a; \mu_k(s), \sigma_k^2(s) I_{d_a}\big)}{\sum_{j=1}^K \pi_j(s) \mathcal{N}\big(a; \mu_j(s), \sigma_j^2(s) I_{d_a}\big)}. \quad (8)$$

Here $r_k(s,a)$ is a training assignment. It measures how much component k is responsible for the demonstrated action under the current state. Unlike global source or coupling designs in computer-vision generative modeling, our assignment is conditioned on the robot state and is recomputed at every decision step. This distinction is important for control: the policy condition s changes continuously along a closed-loop trajectory, and the same action vector can have different meanings depending on the current observation, history, and future branch.

Orthogonal Source Lifting

The source prior above gives us selectable components, but source selection alone does not guarantee an intervenable shared field. If two branches cross in action space, two different source components may reach the same intermediate action point with incompatible target velocities. A velocity field conditioned only on the action coordinate would then average these velocities and erase the selected source identity.

Orthogonal Source Lifting prevents this identity collapse by transporting points in a lifted space. Let

$$\bar{x} = [x^a, x^g] \in \mathbb{R}^{d_a + d_g},$$

where $x^a \in \mathbb{R}^{d_a}$ is the action component and $x^g \in \mathbb{R}^{d_g}$ is an auxiliary lift coordinate. We use orthogonal anchors $\{e_k\}_{k=1}^K$, with $e_k^\top e_j = \delta_{kj}$ and typically $d_g \geq K$. Figure 3(b) shows the fixed-anchor case for readability. In the full model, we use state-dependent lift embeddings $\tau_k(s)$ regularized toward these anchors:

$$\mathcal{L}_{anc} = \mathbb{E}_s \sum_{k=1}^K \|\tau_k(s) - e_k\|^2. \quad (9)$$

For handle k, SL-FM lifts only the source endpoint. Given an action-space source $x^a_{0,k}$ and a demonstrated target action a, we define

$$\bar{x}_{0,k} = [x^a_{0,k}, \lambda \tau_k(s)], \qquad \bar{x}_1 = [a, 0], \quad (10)$$

where $\bar{x}_{0,k}$ is the lifted source endpoint, $\bar{x}_1$ is the lifted target endpoint, and λ controls the lift scale.

Importantly, the target distribution is not split by z. Every demonstrated action is embedded into the same zero-lift plane:

$$p_1(\bar{x} \mid s) = p_{\text{data}}(a \mid s)\delta(x^g = 0). \tag{11}$$

Equivalently, projecting the lifted target distribution back to the action component exactly recovers the original demonstration distribution. Thus, SL-FM preserves the standard imitation target rather than replacing it with mode-conditioned targets $p(a \mid s, z)$. The handle affects only the source geometry before transport, not which target actions are included in training.

The lifted interpolation path is :

$$\begin{aligned} \bar{x}_{t,k} &= (1-t)\bar{x}_{0,k} + t\bar{x}_1 \\ &= \left[(1-t)x^a_{0,k} + ta,\ (1-t)\lambda\tau_k(s)\right], \end{aligned} \tag{12}$$

with target velocity:

$$\bar{u}_{t,k} = \bar{x}_1 - \bar{x}_{0,k} = \left[a - x^a_{0,k},\ -\lambda\tau_k(s)\right]. \tag{13}$$

For $t < 1$, different handles remain separated in the lift coordinate even when their action coordinates overlap. At execution, only the terminal action component is applied to the robot. The lift coordinate is not a second robot action and is never sent to the environment. It is an internal integration coordinate that preserves source identity until all targets meet on the zero-lift plane. The velocity field observes the continuous lifted state $\bar{x}_t$, but it never receives a separate discrete expert label.

Floor-Weighted Flow Training

Given the lifted paths above, SL-FM trains one shared velocity field over all handles. The soft responsibility $r_k(s, a)$ from Eq. (8) determines how much handle k should contribute to the flow regression for the demonstrated pair (s, a). A natural choice is to weight each lifted path by $r_k(s, a)$, so that handles closer to the demonstrated action receive larger supervision.

However, pure responsibility weighting can create dead handles when K is redundant. In practice, some components may receive almost no assignment for many states, as illustrated by the low-responsibility handle in Figure 3(c). If such handles rarely participate in flow training, their lifted trajectories can be poorly learned, leading to unreliable behavior when the evaluator switches to them at test time. To keep every handle trainable while preserving the responsibility structure, we use a responsibility floor:

$$\gamma_k(s, a) = (1-\alpha)r_k(s, a) + \frac{\alpha}{K}, \tag{14}$$

where $\alpha \in [0, 1]$ controls the amount of uniform supervision in the objective. When $\alpha = 0$, the method naturally reduces to pure responsibility weighting. Futhermore, larger α gives more training signal to low-responsibility handles. The floor affects only the flow-regression weights. Again, it does not change the deployment prior $\pi_\phi(z \mid s)$, add a policy head, or condition the velocity field on z.

The floor-weighted flow-matching objective is

$$\mathcal{L}_{W\text{-}FM} = \mathbb{E}_{(s,a),t,\{\epsilon_k\}} \sum_{k=1}^{K} \text{sg}(\gamma_k(s, a)) \left\| v_\theta(\bar{x}_{t,k}, t, s) - \bar{u}_{t,k} \right\|^2, \tag{15}$$

where the time step $t \sim \mathcal{U}[0, 1]$ and $\text{sg}(\cdot)$ stops gradients through the assignment weights. This prevents the flow loss from directly reshaping the posterior assignments, while still allowing the lifted source construction and the shared field to be optimized jointly during training.

Figure 3(c) summarizes the two main training signals as source fitting and weighted flow matching. In the implementation, we also use scalar weights and the anchor regularizer:

$$\mathcal{L} = \mathcal{L}_{W\text{-}FM} + w_{\text{src}}\mathcal{L}_{src} + \beta\mathcal{L}_{anc}. \tag{16}$$

Each minibatch predicts $\{\pi_k, \mu_k, \sigma_k, \tau_k\}_{k=1}^K$ from s, computes r_k and γ_k from the demonstrated action, constructs all lifted paths, and updates the source prior and shared velocity field jointly.

Deployment and source intervention.

At inference time, free deployment samples $z \sim \pi_\phi(\cdot \mid s)$:

$$x^a_{0,z} = \mu_z(s) + \sigma_z(s)\epsilon_z, \quad \bar{x}_{0,z} = [x^a_{0,z}, \lambda\tau_z(s)],$$

and integrates the shared field $d\bar{x}_t/dt = v_\theta(\bar{x}_t, t, s)$. Only the terminal action component is executed. For intervention, the evaluator sets $z = k$ at a chosen decision state and uses the same field. Therefore, the intervention changes only the local source endpoint, not the policy architecture or dynamics model. Our same-prefix protocol realizes Figure 2 by replaying an identical prefix and changing only this local handle. We provide the pseudocode in Algorithm 1.

Algorithm 1 SL-FM Training and Source Intervention

Input: demonstrations $\{(s_i, a_i)\}$, handles K, anchors $\{e_k\}$, lift scale λ, floor α, weights w_{src}, β.

Output: source prior p_ϕ, shared field v_θ, and source-handle rollout interface.

for minibatch (s, a) **do**
  Predict $\{\pi_k, \mu_k, \sigma_k, \tau_k\}_{k=1}^K$ from s.
  Compute $r_k(s, a)$ and $\gamma_k = (1-\alpha)r_k + \alpha/K$.
  for handle $k = 1, \ldots, K$ **do**
    Sample $\epsilon_k \sim \mathcal{N}(0, I_{d_a})$ and set $x^a_{0,k} = \mu_k(s) + \sigma_k(s)\epsilon_k$.
    Lift source and target: $\bar{x}_{0,k} = [x^a_{0,k}, \lambda\tau_k(s)]$, $\bar{x}_1 = [a, 0]$.
    Sample $t \sim \mathcal{U}[0, 1]$; compute $\bar{x}_{t,k}$, $\bar{u}_{t,k}$, and $\ell_{t,k}$.
  end for
  Update (θ, ϕ) with $\mathcal{L}_{W\text{-}FM} + w_{\text{src}}\mathcal{L}_{src} + \beta\mathcal{L}_{anc}$.
end for
Free rollout: sample $z \sim \pi_\phi(\cdot \mid s)$, sample $\bar{x}_{0,z}$, integrate $v_\theta(\bar{x}_t, t, s)$, and execute the action component.
Intervention: set $z = k$, sample $\bar{x}_{0,k}$, integrate the same field, and execute only the action component.

Experiments

Protocol and Metrics

We evaluate SL-FM on D3IL Avoiding and Aligning, PushT, and D4RL Kitchen (Jia et al. 2024; Chi et al. 2023; Fu et al.

Table 1: D3IL Avoiding results. Type denotes the main multimodal mechanism: global-src = global multimodal source/coupling; cond-src = conditional source prior; handle-src = state-conditioned source handle.

Method	Success	Entropy	Type
MM-FM	0.225	0.619	global-src
Modal coupling	0.335	0.521	modal-cpl
CPD	0.692	**0.944**	cond-src
BeT	0.747	0.844	implicit
IBC	0.760	0.850	implicit
FM	0.798	0.933	single-src
SL-FM w/o lift	0.733	0.907	handle-src
SL-FM	**0.825**	0.921	handle-src + lift

Table 2: Comparison across multimodal benchmarks. Avoid. and Align. report D3IL success. PushT reports mean maximum coverage. Avg. is the unweighted average across the three columns.

Method	Avoid.	Align.	PushT	Avg.
FM	0.798	**0.808**	0.801	0.802
CPD	0.692	0.715	0.814	0.740
MM-FM	0.225	0.000	0.279	0.168
Modal cpl.	0.335	0.623	0.839	0.599
Z-cond. field	0.708	0.750	0.833	0.764
SL-FM	**0.825**	0.792	**0.852**	**0.823**

2020). The Avoiding task is the main source-intervention testbed because it has shared starts and frequent route decisions. We compare against FM, source and coupling baselines (CPD, MM-FM, and Modal coupling), a direct z-conditioned field (Z-cond. field), and official BeT and IBC references. Avoiding and Aligning report success, Avoiding also reports route entropy, PushT reports mean maximum coverage, and Kitchen reports 500-step task completion. Additional baseline definitions, protocol details, and hyperparameters are provided in detail in the appendix.

Task Performance Across Multimodal Benchmarks

We first test whether the source-intervention interface preserves ordinary free-deployment imitation. Table 1 shows that SL-FM achieves the best Avoiding success while retaining high route entropy. The gap between *SL-FM w/o lift* and full SL-FM indicates that learning a state-local source handle is not enough. Also, orthogonal lifting is needed to preserve handle identity when paths cross. MM-FM instead uses global action-space modes, which are not tied to local decision states. Across Avoiding, Aligning, and PushT, Table 2 shows the best average score for SL-FM. The gains concentrate on tasks with more frequent local branching, while Aligning stays close to the base FM policy. CPD, modal coupling, and direct z-conditioning can be competitive in some settings, but they lack the same state-local, source-only intervention interface.

We also evaluate SL-FM on D4RL Kitchen, a multi-task robot manipulation benchmark (Fu et al. 2020). Figure 4A shows that the $K = 8$ SL-FM variant improves over FM by $+0.42$ completed tasks in 500-step rollouts, suggesting that the source-structured design remains useful beyond the

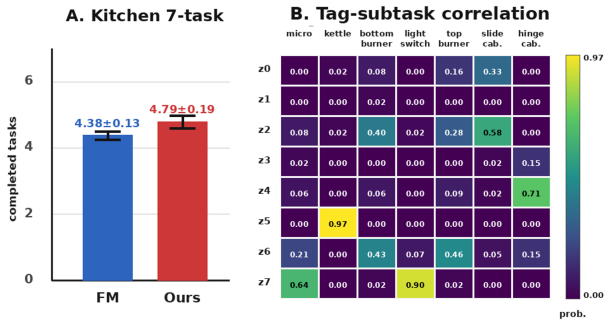

Figure 4: D4RL Kitchen diagnostic. A: 7-task completion mean and standard deviation for FM and the $K = 8$ SL-FM variant. B: source-handle–subtask correlation in rollouts.

Table 3: Same-prefix source intervention on D3IL Avoiding. From an identical pre-decision prefix, paired rollouts differ only in the five-policy-call local intervention. Outside the window, stochastic draws are matched and the policies resume free sampling.

Method	Intervention	Future sep.	Route changed	Both success	Both-succ. route chg.
FM	noise resample	0.190 m	0.767	0.428	0.286
z-field	change z input	0.273 m	**0.930**	0.455	0.421
SL-FM	same handle	0.000 m	0.000	0.787	0.000
SL-FM	change handle	**0.287 m**	0.911	**0.639**	**0.578**

route-control setting. Figure 4B reports the source-handle–subtask correlation, which we discuss in the next section.

Source-Intervention Controllability

Task performance alone does not show that the sampled handle is an actionable cause of future behavior. We therefore use a same-prefix counterfactual on D3IL Avoiding. Two rollouts share an identical prefix and then differ only in the local source handle at the first high-entropy decision point in the pre-obstacle band $y \in [-0.30, -0.02]$. For each matched state, we replay the prefix, branch into all six unordered pairs of $z \in \{0, 1, 2, 3\}$, force each selected handle for five consecutive policy calls, and then return both branches to free source resampling.

Table 3 shows that changing the source handle redirects the future while preserving task execution. SL-FM changes the final route in 91.1% of paired interventions and achieves a 57.8% both-success route-change rate. This last metric is stricter than route change alone: it counts only pairs where both branches succeed and nevertheless take different routes. FM noise resampling and direct z-conditioning can also change routes, but they produce fewer successful route changes. The same-handle control produces no route changes, confirming that the effect comes from changing the source handle rather than replay instability.

Free-rollout diagnostics show the same alignment without manual intervention. In Avoiding, each successful trajectory can be encoded by three route digits, where digit0, digit1, and digit2 denote the branch choices around the first, second, and third obstacle regions, respectively. Figure 6 reports the mutual information between sampled handles and these realized

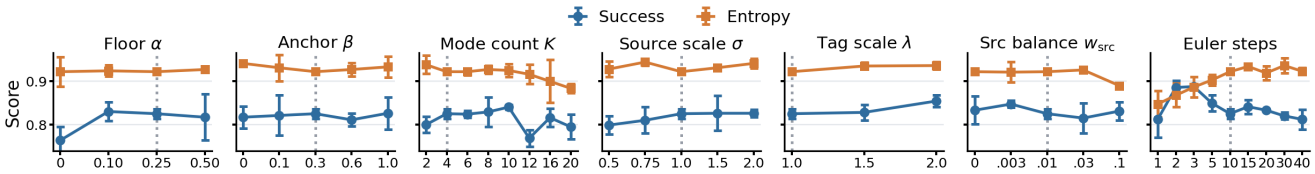

Figure 5: Hyperparameter and deployment-cost sensitivity on D3IL Avoiding, varying one setting per panel and keeping the remaining defaults fixed.

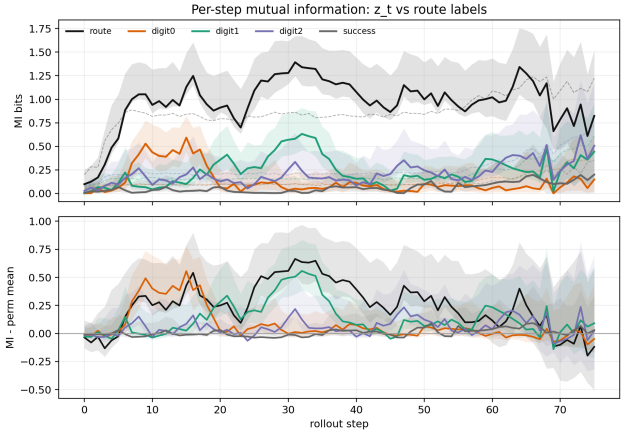

Figure 6: Per-step mutual information between sampled source handles and realized route labels under free source resampling, shown up to step 75 with bootstrap 95% intervals. Peaks align with route decision phases.

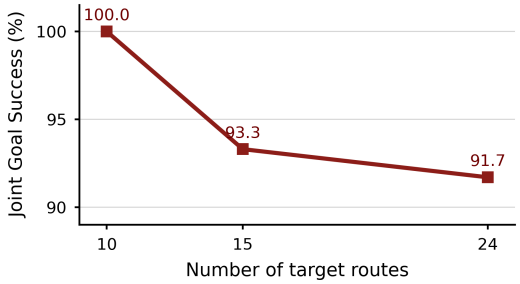

Figure 7: Fixed-state downstream selector diagnostic on Avoiding route targets. Joint goal success means that the selected route matches the target and the task succeeds.

route digits over time. The digit0 curve reaches its peak before the first obstacle, showing that early handle samples are most informative about the first branch choice. The digit1 curve peaks later, before the second obstacle, consistent with the second route decision occurring after the first branch has been resolved. The digit2 curve is weaker: near the final obstacle, the current position already strongly constrains the remaining route, so z carries less additional information under free rollout. Nevertheless, the intervention results show that changing z can still force meaningful future changes. Mutual information with success remains small throughout, indicating that the handles track branch identity rather than merely success or failure. Figure 4B extends this diagnostic to Kitchen, where the source-handle–subtask heat map is non-uniform. The strongest entries align specific handles with subtasks: $z5$ with kettle, $z7$ with microwave and light switch, $z4$ with hinge cabinet, and $z2$ with slide cabinet and bottom burner. Notably, slide cabinet and bottom burner are spatially and procedurally related subtasks, and their shared association with $z2$ suggests that the source handles can capture semantically meaningful structure rather than only arbitrary rollout variation. Together, these results suggest that the handles capture subtask-specific progress patterns in long-horizon manipulation.

Downstream Route Control

We next test whether a downstream module can use the exposed handle. With the low-level SL-FM policy frozen, a high-level selector chooses z every five consecutive policy calls from a fixed initial state. Free source resampling still solves the task, but it achieves 0% commanded-route success for the specified targets. CEM-retrieved source schedules and a PPO selector initialized from CEM elites reach 91.7% joint route-and-task success over all 24 route targets (Rubinstein 1999; Schulman et al. 2017). Figure 7 shows the same trend under increasingly broad route sets: the selector reaches 100.0% joint success over 10 target routes, 93.3% over 15 routes, and 91.7% over all 24 routes. This result demonstrates that the source handle can serve as an action space for high-level control, analogous to skill-level decision abstractions, while the same-prefix experiment remains the primary causal evidence.

Design Ablations and Sensitivity

Finally, Figure 5 summarizes dense Avoiding sensitivity sweeps. The responsibility floor only changes the flow-regression weights γ_k. Moderate floors improve success while preserving route entropy, supporting their role as dead-handle robustness rather than collapse-inducing regularization. Nearby anchor penalties, source scales, source-alignment weights, and mode counts $K = 4$–10 remain viable. Very large K is less reliable, likely because a fixed minibatch provides noisier responsibility-weighted updates for each handle. Unstable lift scales and excessive Euler steps also reduce robustness. All plotted Avoiding points use the official 480-trajectory, three-seed protocol.

Conclusion

We presented Source-Lifted Flow Matching, a flow-matching policy that exposes an intervenable source interface without conditioning the velocity field on a discrete code. The central

mechanism is Orthogonal Source Lifting: sources are lifted into orthogonal coordinates while every target remains on the zero-tag plane. This preserves the original conditional data distribution at $z = 0$, but makes source choices externally addressable during deployment. Across crossing-flow, D3IL, PushT, and Kitchen diagnostics, SL-FM remains competitive in free deployment and redirects behavior under same-prefix source interventions.

References

Albergo, M. S.; and Vanden-Eijnden, E. 2023. Building Normalizing Flows with Stochastic Interpolants. In *International Conference on Learning Representations*.

Bishop, C. M. 1994. Mixture Density Networks. Technical Report NCRG/94/004, Aston University.

Black, K.; Brown, N.; Driess, D.; Esmail, A.; Equi, M. R.; Finn, C.; Fusai, N.; Groom, L.; Hausman, K.; Ichter, B.; Jakubczak, S.; Jones, T.; Ke, L.; Levine, S.; Li-Bell, A.; Mothukuri, M.; Nair, S.; Pertsch, K.; Shi, L. X.; Smith, L.; Tanner, J.; Vuong, Q.; Walling, A.; Wang, H.; and Zhilinsky, U. 2025. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of Robotics: Science and Systems*. Los Angeles, CA, USA.

Chi, C.; Feng, S.; Du, Y.; Xu, Z.; Cousineau, E.; Burchfiel, B. C. M.; and Song, S. 2023. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Proceedings of Robotics: Science and Systems*. Daegu, Republic of Korea.

Du, M.; and Song, S. 2025. DynaGuide: Steering Diffusion Polices with Active Dynamic Guidance. In *Advances in Neural Information Processing Systems*.

Florence, P.; Lynch, C.; Zeng, A.; Ramirez, O. A.; Wahid, A.; Downs, L.; Wong, A.; Lee, J.; Mordatch, I.; and Tompson, J. 2022. Implicit Behavioral Cloning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, 158–168. PMLR.

Fu, J.; Kumar, A.; Nachum, O.; Tucker, G.; and Levine, S. 2020. D4RL: Datasets for Deep Data-Driven Reinforcement Learning. *arXiv preprint arXiv:2004.07219*.

Guo, P.; and Schwing, A. G. 2025. Variational Rectified Flow Matching. *arXiv preprint arXiv:2502.09616*.

Issachar, N.; Salama, M.; Fattal, R.; and Benaim, S. 2025. Designing a Conditional Prior Distribution for Flow-Based Generative Models. *arXiv preprint arXiv:2502.09611*.

Jacobs, R. A.; Jordan, M. I.; Nowlan, S. J.; and Hinton, G. E. 1991. Adaptive Mixtures of Local Experts. *Neural Computation*, 3(1): 79–87.

Jia, X.; Blessing, D.; Jiang, X.; Reuss, M.; Donat, A.; Lioutikov, R.; and Neumann, G. 2024. Towards Diverse Behaviors: A Benchmark for Imitation Learning with Human Demonstrations. In *International Conference on Learning Representations*.

Jiang, S.; Fang, X.; Roy, N.; Lozano-Pérez, T.; Kaelbling, L. P.; and Ancha, S. 2025. Streaming Flow Policy: Simplifying Diffusion/Flow-Matching Policies by Treating Action Trajectories as Flow Trajectories. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, 238–257. PMLR.

Lipman, Y.; Chen, R. T. Q.; Ben-Hamu, H.; Nickel, M.; and Le, M. 2023. Flow Matching for Generative Modeling. In *International Conference on Learning Representations*.

Liu, X.; Gong, C.; and Liu, Q. 2023. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *International Conference on Learning Representations*.

Luo, G.; Cole, F.; Zhang, S.; Wan, Y.; Lu, Y.; and Sun, J. 2026. Flow Matching for Multimodal Distributions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23260–23271.

NVIDIA; Bjorck, J.; Castañeda, F.; Cherniadev, N.; Da, X.; Ding, R.; Fan, L. J.; Fang, Y.; Fox, D.; Hu, F.; Huang, S.; Jang, J.; Jiang, Z.; Kautz, J.; Kundalia, K.; Lao, L.; Li, Z.; Lin, Z.; Lin, K.; Liu, G.; Llontop, E.; Magne, L.; Mandlekar, A.; Narayan, A.; Nasiriany, S.; Reed, S.; Tan, Y. L.; Wang, G.; Wang, Z.; Wang, J.; Wang, Q.; Xiang, J.; Xie, Y.; Xu, Y.; Xu, Z.; Ye, S.; Yu, Z.; Zhang, A.; Zhang, H.; Zhao, Y.; Zheng, R.; and Zhu, Y. 2025. GR00T N1: An Open Foundation Model for Generalist Humanoid Robots. *arXiv preprint arXiv:2503.14734*.

Park, S.; Li, Q.; and Levine, S. 2025. Flow Q-Learning. *arXiv preprint arXiv:2502.02538*.

Pearce, T.; Rashid, T.; Kanervisto, A.; Bignell, D.; Sun, M.; Georgescu, R.; Macua, S. V.; Tan, S. Z.; Momennejad, I.; Hofmann, K.; and Devlin, S. 2023. Imitating Human Behaviour with Diffusion Models. In *International Conference on Learning Representations*.

Physical Intelligence; Black, K.; Brown, N.; Darpinian, J.; Dhabalia, K.; Driess, D.; Esmail, A.; Equi, M.; Finn, C.; Fusai, N.; Galliker, M. Y.; Ghosh, D.; Groom, L.; Hausman, K.; Ichter, B.; Jakubczak, S.; Jones, T.; Ke, L.; LeBlanc, D.; Levine, S.; Li-Bell, A.; Mothukuri, M.; Nair, S.; Pertsch, K.; Ren, A. Z.; Shi, L. X.; Smith, L.; Springenberg, J. T.; Stachowicz, K.; Tanner, J.; Vuong, Q.; Walke, H.; Walling, A.; Wang, H.; Yu, L.; and Zhilinsky, U. 2025. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. *arXiv preprint arXiv:2504.16054*.

Pooladian, A.-A.; Ben-Hamu, H.; Domingo-Enrich, C.; Amos, B.; Lipman, Y.; and Chen, R. T. Q. 2023. Multisample Flow Matching: Straightening Flows with Minibatch Couplings. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, 28100–28127. PMLR.

Ren, A. Z.; Lidard, J.; Ankile, L. L.; Simeonov, A.; Agrawal, P.; Majumdar, A.; Burchfiel, B.; Dai, H.; and Simchowitz, M. 2024. Diffusion Policy Policy Optimization. *arXiv preprint arXiv:2409.00588*.

Reuss, M.; Li, M.; Jia, X.; and Lioutikov, R. 2023. Goal-Conditioned Imitation Learning using Score-Based Diffusion Policies. In *Proceedings of Robotics: Science and Systems*. Daegu, Republic of Korea.

Rubinstein, R. Y. 1999. The Cross-Entropy Method for Combinatorial and Continuous Optimization. *Methodology and Computing in Applied Probability*, 1(2): 127–190.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*.

Shafiullah, N. M. M.; Cui, Z. J.; Altanzaya, A.; and Pinto, L. 2022. Behavior Transformers: Cloning k Modes with One Stone. In *Advances in Neural Information Processing Systems*, volume 35, 22955–22968.

Su, E.; Westenbroek, T.; Nagabandi, A.; and Gupta, A. 2026. RFS: Reinforcement Learning with Residual Flow Steering for Dexterous Manipulation. *arXiv preprint arXiv:2602.01789*.

Tong, A.; Fatras, K.; Malkin, N.; Huguet, G.; Zhang, Y.; Rector-Brooks, J.; Wolf, G.; and Bengio, Y. 2024. Improving and Generalizing Flow-Based Generative Models with Minibatch Optimal Transport. *Transactions on Machine Learning Research*.

Wagenmaker, A.; Zhang, Y.; Nakamoto, M.; Park, S.; Yagoub, W.; Nagabandi, A.; Gupta, A.; and Levine, S. 2025. 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*, 258–282. PMLR.

Wang, L.; Yu, H.; Yu, C.; Gao, S.; and Christensen, H. 2025. Controllable Motion Generation via Diffusion Modal Coupling. *arXiv preprint arXiv:2503.02353*.

Zhai, X.; Zhao, Q.; Yu, Q.; and Hao, C. 2025. VFP: Variational Flow-Matching Policy for Multi-Modal Robot Manipulation. *arXiv preprint arXiv:2508.01622*.

Zhang, H.; Sun, Y.; and Xiong, H. 2026. GoldenStart: Q-Guided Priors and Entropy Control for Distilling Flow Policies. In *The Fourteenth International Conference on Learning Representations*.

Zhang, T.; Yu, C.; Su, S.; and Wang, Y. 2025. ReinFlow: Fine-Tuning Flow Matching Policy with Online Reinforcement Learning. In *Advances in Neural Information Processing Systems*.

Experimental Details

Baselines and ablations. Unless labeled as an ablation, SL-FM uses source-only z, Orthogonal Source Lifting, one latent-free shared velocity field, and the responsibility floor.

FM. The standard conditional flow-matching policy uses a single Gaussian source and one field; its intervention baseline is source-noise resampling rather than handle selection.

CPD. Conditional Prior Distribution replaces the fixed source with a state-conditioned single Gaussian trained toward demonstrated actions, but has no K-way handle, responsibility assignment, or lifted tag coordinate.

MM-FM. MM-FM uses a global action-space GMM source; its component index is a global action cluster rather than a state-local decision handle.

Modal coupling. Modal coupling uses fixed global source anchors and hard action-cluster/source-target pairings, so its pairings are not recomputed as state-local decisions.

z-conditioned field. This direct-conditioning baseline feeds the discrete handle into $v_\theta(x_t, t, s, z)$, allowing mode-specific dynamics; SL-FM keeps the field shared and lets z affect only the source endpoint.

BeT. Behavior Transformer is included as an official D3IL Avoiding reference and represents multimodality through discrete behavior/action tokens rather than flow source geometry.

IBC. Implicit Behavioral Cloning is another official Avoiding reference; it scores candidate actions with an implicit model and has no flow source or source-level intervention variable.

SL-FM w/o lift. This ablation keeps the state-conditioned source mixture and source-only handle but removes the auxiliary lift coordinate, testing whether a learned local handle alone preserves identity through crossings.

Evaluation protocol. Avoiding follows the official D3IL state benchmark protocol and reports success and normalized route entropy over the 480-trajectory, three-seed evaluation. PushT reports mean maximum coverage, and Kitchen reports 500-step task completion with eight environments and three evaluation seed offsets.

Representative defaults. A representative D3IL Avoiding run uses $K = 4$, $d_g = K$, $\sigma_{\text{iso}} = 1.0$, responsibility floor $\alpha = 0.25$, anchor/balance weights $0.3/0.01$, 10 Euler steps, and 2000 training epochs. The encoder, shared field, and source heads are residual MLPs with six 256-wide Mish layers; other domains keep the same interface while adjusting task-specific backbones or K.

Additional Discussion

Planning and reinforcement learning. The exposed source handle is an interface rather than a complete planner. It can be combined with MPC or RL by letting a planner, value function, or high-level policy choose or constrain z, while the flow policy executes low-level continuous actions.

Semantic conditioning. The handle can also be grounded by semantic signals. Text instructions, goal images, or vision-language embeddings could bias $\pi_\phi(z \mid s)$ or select interventions, connecting high-level intent to continuous control without conditioning the shared velocity field directly on a discrete mode.

Large heterogeneous imitation data. Because responsibilities are learned without manual mode labels, source handles are promising for large imitation corpora that mix styles, imbalanced modes, and non-expert or mixed-quality trajectories. In this setting, the source prior can provide a compact interface for separating reusable behavior patterns while retaining a single shared action generator.

Additional Kitchen Results

Table 4 gives the full 500-step Kitchen diagnostic behind Figure 4A, using eight environments and seed offsets $0, 8, 16$. Here r_f is the responsibility floor, α the anchor weight, b_w the source-balance weight, and s_s, t_s the source and tag scales. Four of five source-structured variants exceed FM on 7-task completion, and the balance-weighted variant gives the best four-target score.

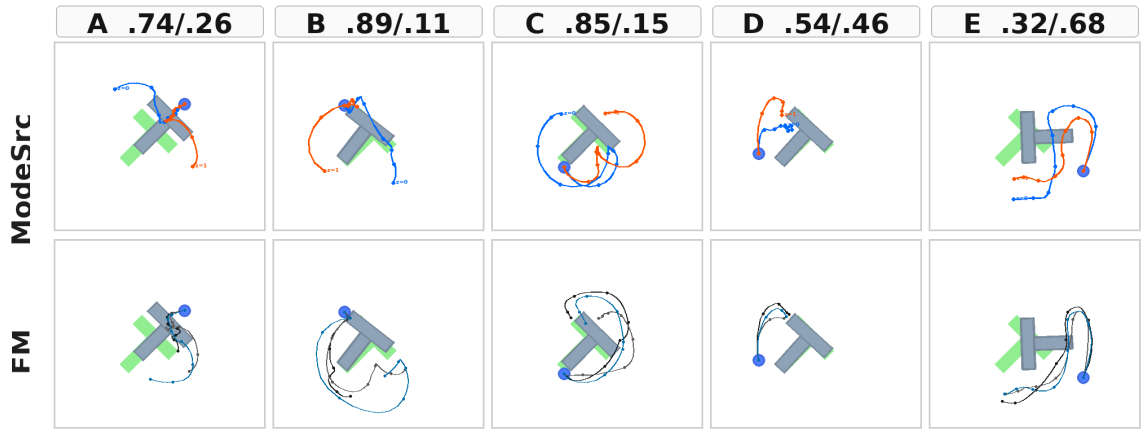

Figure 8: Additional PushT branching examples. Columns show selected PushT states. Forced source choices in the source-structured policy expose visibly separated contact strategies, whereas FM source-noise resampling provides stochastic variation without an explicit persistent handle.

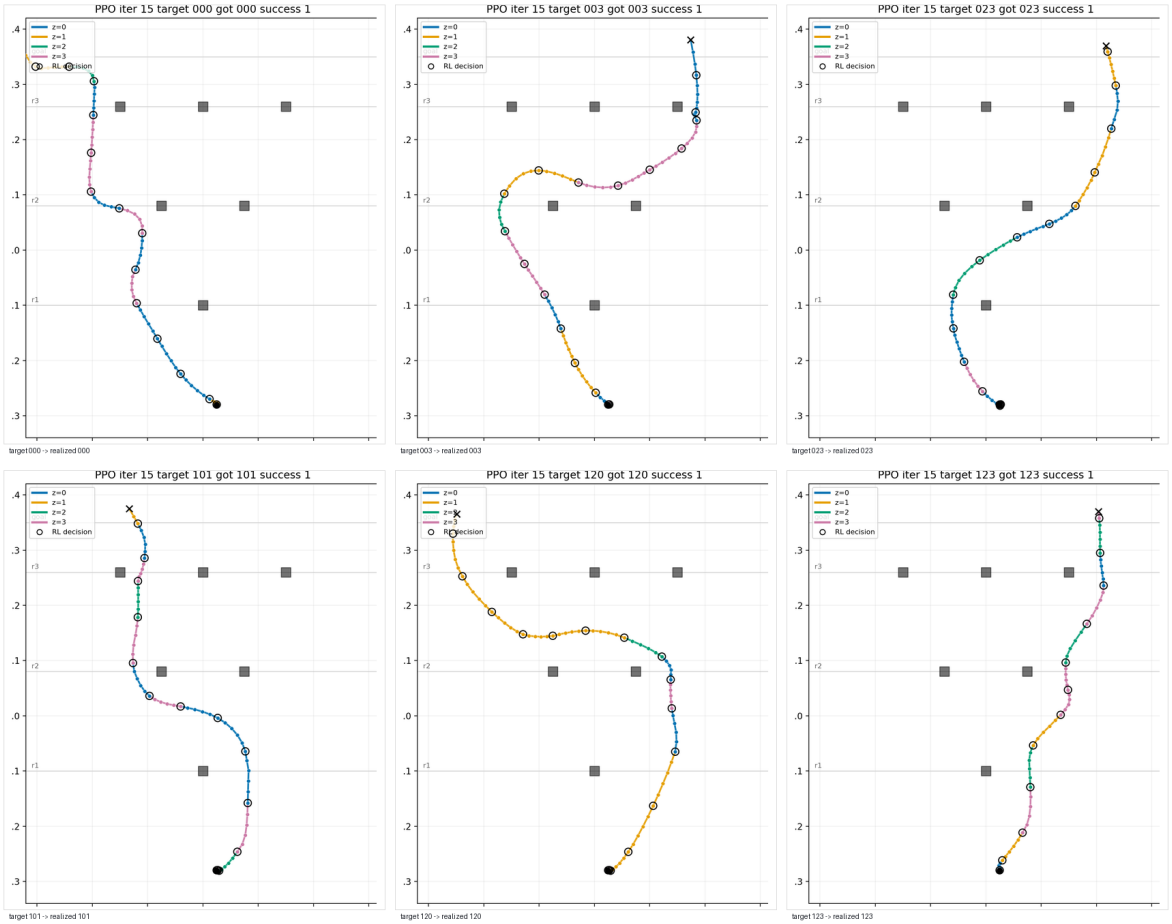

Figure 9: Representative successful downstream route-control trajectories on Avoiding. The low-level SL-FM policy is frozen, while a high-level selector chooses one source handle every five policy calls. The shown fixed-start PPO checkpoint reaches $22/24$ joint route-and-task success; colors indicate the executed source handle and black circles mark selector decisions.

Table 4: D4RL Kitchen 500-step diagnostic at checkpoint 500. Four-target scores sum microwave, kettle, bottom burner, and light switch.

Method / variant	7-task mean	4-target mean
SL-FM r_f=0.10, α=0.3, s_s=10, t_s=10	**4.79 ± 0.19**	2.71
SL-FM r_f=0.25, α=0.3, b_w=0.005, s_s=10, t_s=10	4.71 ± 0.14	**2.96**
SL-FM r_f=0.25, α=0.3, s_s=10, t_s=10	4.67 ± 0.80	2.83
SL-FM r_f=0.10, α=0.5, s_s=10, t_s=15	4.62 ± 0.22	2.62
SL-FM r_f=0.25, α=0.5, s_s=10, t_s=10	4.29 ± 0.26	2.54
FM	4.38 ± 0.13	2.54

Additional Qualitative Results

Figure 8 visualizes PushT branching at matched states. Forced source choices in the source-structured policy produce distinct contact strategies around the T block, while FM source-noise resampling gives stochastic variation without a persistent index for selecting one strategy. These examples complement the aggregate PushT scores by showing that the exposed handle corresponds to qualitatively different manipulation choices.

Figure 9 shows successful Avoiding trajectories from the downstream selector. Black circles mark selector decisions and colors indicate the active source handle. The trajectories reach different commanded routes from the same start, illustrating that a high-level module can use the handle as a compact route-control interface after the low-level SL-FM policy is frozen.