MODEFLOW: BREAKING THE CONNECTIVITY BARRIER OF ONE-STEP FLOW POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

One-step flow policies enable efficient action generation. However, we identify a fundamental connectivity barrier. Mapping connected Gaussian noise through a continuous generator produces a proposal distribution with connected support, so when high-value behaviors occupy separated regions, the policy necessarily places probability mass on the low-value bridges between them and suppressing that mass below δ forces the generator's Lipschitz constant to grow as $\Omega(D/\delta)$. We formalize this barrier and derive a Gaussian-isoperimetric lower bound on the bridge mass. To overcome this challenge, we propose MODEFLOW, a mode-conditioned one-step MeanFlow policy that discovers discrete behavior modes from offline trajectories, conditions the generator and critic on mode labels, and stratifies a fixed proposal budget across modes. This allows the policy to represent separated behaviors as distinct proposal branches while retaining efficient one-step generation. We further prove the return improvement of MODEFLOW over one-step policies of this class, as the bridge mass induces a nonzero probability of missing the optimal behavior. Experiments demonstrate that MODEFLOW consistently outperforms the baselines across the Robomimic and OGBench tasks, with particularly large gains on long-horizon manipulation. On the most challenging Cube-triple-task5, MODEFLOW achieves 98.4% success while all baselines fail.

1 INTRODUCTION

Generative policies have emerged as a powerful class of policies for reinforcement learning (RL), particularly for multimodal behavior in demonstrations and offline data. Diffusion and flow-based policies can capture such multimodality (Chi et al., 2025; Wang et al., 2023; Park et al., 2025b), but their iterative sampling is costly for online RL, where actions must be generated repeatedly during interaction and training. The mean velocity policy (MVP) (Zhan et al., 2026) addresses this efficiency challenge with one-step MeanFlow generation (Geng et al., 2026), achieving strong performance with a single network evaluation per action chunk. This efficiency, however, raises a fundamental question: can a one-step continuous generator represent and reliably select among multiple, well-separated behavioral modes?

The difficulty comes from the geometry of one-step generation. At a fixed state s, a one-step flow policy generates an action as $A = F_s(\epsilon)$ from Gaussian noise ϵ, where F_s is a continuous neural network. Consequently, the support of the resulting proposal distribution is connected: a continuous map cannot map a connected noise space into several disconnected action regions. This becomes problematic when high-value behaviors form multiple separated modes in the action space. For example, rearranging three cubes admits different valid orders of manipulation, each corresponding to a distinct behavioral mode whose action chunks occupy a separate region of the action space at the same state. Because the proposal support is connected, assigning substantial probability to two separated regions necessarily introduces probability mass in the region between them. We call this the *connectivity barrier*, and refer to the induced probability mass between separated regions as *bridge mass*.

The connectivity constraint becomes increasingly costly as behavioral modes move farther apart. We show that, for an L-Lipschitz one-step policy, the bridge mass between two high-value regions admits a Gaussian-isoperimetric lower bound that increases with their separation D. In particular, suppressing the bridge mass below δ requires $L = \Omega(D/\delta)$. Thus, concentrating a single continuous generator

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.

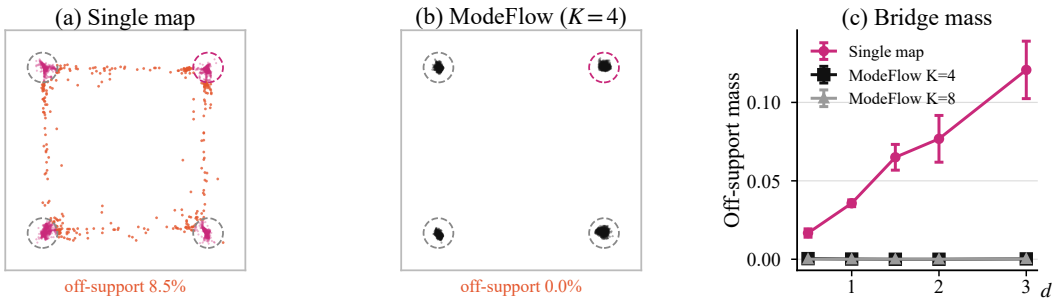

Figure 1: **The connectivity barrier and our mode-conditioned solution.** A controlled 2-D setting with four separated high-value regions. (a,b) A single one-step map (MeanFlow) induces bridge mass between separated regions, whereas mode conditioning represents them with distinct conditional mappings. (c) Bridge mass increases with mode separation d for the single map but remains negligible with mode conditioning (details in Appendix D)

on well-separated behaviors requires increasingly steep mappings, creating a fundamental tension with the smooth average-velocity regression used by MeanFlow. This limitation also propagates to value-based selection: under a finite proposal budget, probability assigned to the regions between behavioral modes reduces the number of proposals available within those modes, while modes with low marginal probability are sampled less frequently. Since the critic can only rank the candidates that are proposed, the resulting proposal allocation directly affects best-of-N policy selection.

Existing approaches address individual aspects of this problem, but not their combination. Multi-step diffusion and flow policies represent complex multimodal distributions (Chi et al., 2025; Li et al., 2026), but iterative generation sacrifices the one-step efficiency of MVP. Mixture-based and sub-mode-conditioned generators can model disconnected supports (Khayatkhoei et al., 2018; Salmona et al., 2022), yet they primarily address generative modeling rather than proposal allocation in online RL. Skill and latent-variable methods introduce discrete behavioral structure for imitation and exploration (Ajay et al., 2020; Shafiullah et al., 2022), but do not combine it with one-step generation and value-based selection. Thus, it remains unclear how to represent separated behavioral modes while allocating a finite proposal budget across them for value-based selection.

To address this gap, we propose MODEFLOW, a one-step policy that explicitly represents behavioral modes as a discrete variable. MODEFLOW first discovers K behavioral modes from offline trajectories and assigns each transition its trajectory-level mode label. The MODEFLOW generator is then conditioned on the mode, yielding a family of K continuous maps whose union can have disconnected support. At inference time, we stratify a fixed proposal budget across modes and use a mode-conditioned critic to evaluate the resulting candidates. This design directly addresses the two limitations above: mode conditioning removes the need for a single continuous map to connect separated behaviors, while stratified proposals allocate candidates across modes rather than according to their marginal frequencies. Crucially, MODEFLOW preserves one-step inference: all modes share the same MeanFlow network, with the mode entering only through an embedding.

Figure 1 provides a conceptual illustration of this idea: a single continuous generator must connect separated high-value regions, whereas mode conditioning decomposes the policy into a family of mode-conditioned continuous mappings. With a fixed proposal budget, mode-stratified sampling further allocates proposals across these modes rather than according to their marginal frequencies.

Our contributions are threefold:

- **We identify and quantify a connectivity barrier in one-step flow policies.** A continuous one-step generator necessarily induces bridge mass between separated high-value regions. We then prove that the connectivity barrier can be removed by introducing mode conditioning into the action generator.
- **We propose MODEFLOW, a mode-conditioned one-step MeanFlow policy that overcomes the connectivity barrier.** MODEFLOW combines trajectory-level behavioral mode discovery, mode-conditioned action generator and critic, and mode-stratified best-of-N selection. We further prove a return improvement of MODEFLOW over one-step policies, as the bridge mass induces a nonzero probability of missing the optimal behavior.

- **We demonstrate strong empirical gains on long-horizon multimodal manipulation.** MODEFLOW consistently outperforms existing one-step and flow-based policies across the Robomimic and OGBench benchmarks, with the largest gains on long-horizon triple-cube tasks. On Cube-triple-task5, MODEFLOW achieves 98.4% success under the same proposal budget, compared with 0% for all baselines.

2 PRELIMINARIES

Offline-to-online RL with action chunks. We consider an MDP $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$ with actions $a \in [-1, 1]^{d_a}$, an offline dataset $\mathcal{D}$, and subsequent online interaction (Levine et al., 2020; Lee et al., 2022). Following Li et al. (2026); Zhan et al. (2026), the policy outputs a chunk $A = (a_0, \dots, a_{H-1}) \in \mathbb{R}^{Hd_a}$ that is executed open-loop, which defines a macro-MDP with reward $R_H = \sum_{j<H} \gamma^j r_j$, discount $\beta = \gamma^H$, $V^\pi(s) = \mathbb{E}_{A\sim\pi} Q^\pi(s, A)$, and $J(\pi) = \mathbb{E}_{s_0\sim\rho} V^\pi(s_0)$.

MeanFlow policies. A flow policy transports noise $\epsilon \sim \mathcal{N}(0, I)$ to a chunk along $\dot{x}_t = v(s, x_t, t)$; as in the MVP code, data sits at $t = 0$ and noise at $t = 1$, so $x_t = (1-t)A + t\epsilon$ and $v = \epsilon - A$. The *mean velocity* $u(s, x_t, t, r) = \frac{1}{t-r}\int_r^t v\, d\tau$ satisfies the MeanFlow identity $u = v - (t-r)\frac{d}{dt}u$ (Geng et al., 2026), which yields the loss

$$\mathcal{L}_{\mathrm{MF}}(\theta) = \mathbb{E}\,\ell\big(u_\theta(s, x_t, t, r) - \mathrm{sg}[v - (t-r)(\partial_t u_\theta + \nabla_x u_\theta\, v)]\big), \tag{1}$$

where the derivative is a Jacobian-vector product, sg is the stop-gradient, and ℓ is an adaptive squared loss. MVP additionally imposes the instantaneous velocity constraint $\mathcal{L}_{\mathrm{IVC}} = \mathbb{E}\,\ell(u_\theta(s, x_t, t, t) - v)$, which enforces the boundary condition $u(s, x_t, t, t) = v(s, x_t, t)$ and stabilizes the MeanFlow objective (Zhan et al., 2026). A chunk is then generated in one evaluation:

$$A = F_s(\epsilon) := \mathrm{clip}\,\big(\epsilon - u_\theta(s, \epsilon, 1, 0), -1, 1\big), \qquad \epsilon \sim \mathcal{N}(0, I_{Hd_a}). \tag{2}$$

Best-of-N improvement. The generator is trained by behavior cloning on replay data, while policy improvement is performed through Q-guided selection. The critic is an ensemble of M heads $Q_\phi^{(1)}, \dots, Q_\phi^{(M)}$, and we write $\bar{Q}_\phi = \frac{1}{M}\sum_m Q_\phi^{(m)}$ for their mean, which is the value used both to score candidates and to form TD targets. Given N i.i.d. proposals $A_i = F_s(\epsilon_i)$, MVP executes $A^\star = \arg\max_{i\le N} \bar{Q}_\phi(s, A_i)$, and stores the selected transition in the replay buffer for subsequent critic and behavior-cloning updates (Ghasemipour et al., 2021; Chen et al., 2023). For a proposal policy π_θ, the corresponding best-of-N gain at state s is

$$\Delta_N(s) = \mathbb{E}_{A_{1:N}\sim\pi_\theta}\left[\max_{i\le N} Q(s, A_i)\right] - V^{\pi_\theta}(s). \tag{3}$$

Thus, the benefit of increasing the proposal budget depends not only on N, but also on the distribution of candidates that the generator can produce, and in particular on its coverage of distinct high-value action modes.

3 THE CONNECTIVITY BARRIER OF ONE-STEP FLOW POLICIES

We now formalize why a single continuous one-step generator struggle to represent qualitatively different high-value behaviors. Fix a state s and let $\mathcal{A} \subseteq \mathbb{R}^{Hd_a}$ be the action space. The generator maps a connected Gaussian latent space to this action space, so its proposal support must remain connected. This creates a geometric obstruction when useful actions occupy multiple separated regions: if two high-value regions are both assigned nonzero probability, a continuous generator must also connect them through actions outside these regions. We refer to the resulting probability mass as the *bridge mass*.

The theorem below makes this argument quantitative in three steps. First, the connected-support argument shows that nonzero bridge mass is unavoidable. Second, if the generator is L-Lipschitz, the separation D between high-value regions imposes a lower bound on this bridge mass through the Gaussian isoperimetric inequality. Thus, reducing bridge mass cannot come for free: in the two-region case, achieving bridge mass δ requires the Lipschitz constant to grow as $\Omega(D/\delta)$. This exposes the trade-off faced by a single continuous generator: it must either assign probability to

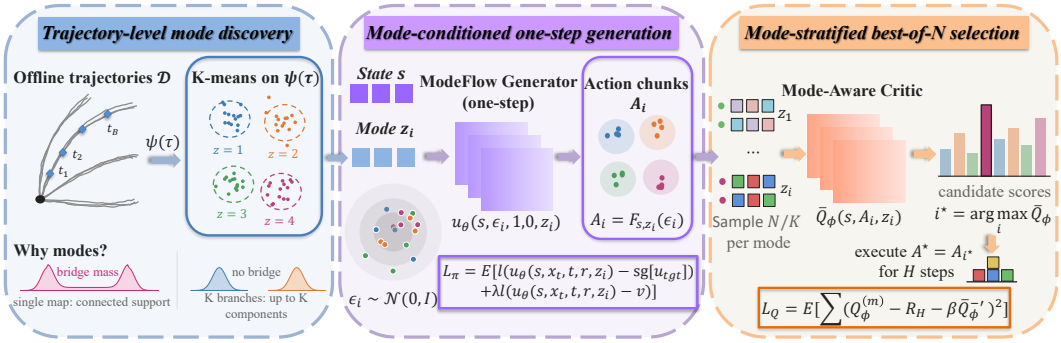

Figure 2: **The framework of ModeFlow.** Offline trajectories are clustered into K behavior modes, whose labels condition both the MeanFlow generator and critic. At each state, the total proposal budget N is evenly allocated across modes, generating N/K proposals per mode in a single batched network evaluation. A mode-aware critic then scores all N candidates and performs best-of-N selection to choose the highest-value action chunk. The policy generator and critic are trained during both offline and online phases.

low-value bridges or become increasingly steep to avoid them. Finally, a discrete mode variable changes the topology of the proposal family. Each mode-conditioned generator can remain continuous while covering only one high-value region, allowing their mixture to have disconnected support and hence eliminating the connectivity-induced bridge mass.

Formally, let $\nu = \mathcal{N}(0, I_m)$ with $m = Hd_a$ and let $\mu_s = (F_s)_{\#}\nu$ denote the proposal distribution generated by a generator $F_s : \mathbb{R}^m \to \mathcal{A}$, where $\#$ denotes the push-forward measure. Suppose the useful actions at s form $K^\star \geq 2$ closed high-value regions $\mathcal{G}_1, \ldots, \mathcal{G}_{K^\star} \subseteq \mathcal{A}$ satisfying

$$\mathrm{dist}(\mathcal{G}_k, \mathcal{G}_\ell) \geq D > 0, \qquad k \neq \ell, \tag{4}$$

where $\mathrm{dist}(\mathcal{G}_k, \mathcal{G}_\ell) := \inf_{x \in \mathcal{G}_k,\, y \in \mathcal{G}_\ell} \|x - y\|_2$. Let $\mathcal{G} = \bigcup_{k=1}^{K^\star} \mathcal{G}_k$ denote the union of all high-value regions, and define the *bridge mass*

$$\delta(\mu_s) := \mu_s(\mathcal{A} \setminus \mathcal{G}), \tag{5}$$

i.e., the probability that the generator proposes an action outside all high-value regions. Let Φ denote the Cumulative Distribution Function(CDF) of the standard normal distribution. We have the theorem as below.

Theorem 1 (Connectivity barrier of one-step generator)**.** *Let F_s be continuous and suppose that $\mu_s(\mathcal{G}_k) > 0$ for at least two regions. Then:*

(i) The support of μ_s is connected:

$$\mathrm{supp}(\mu_s) = \overline{F_s(\mathbb{R}^m)}. \tag{6}$$

Here, $\mathrm{supp}(\mu_s)$ denotes the set of action points whose every neighborhood has positive probability under μ_s. Consequently, $\delta(\mu_s) > 0$.

(ii) If F_s is L-Lipschitz and $p_k := \mu_s(\mathcal{G}_k) \in (0, 1)$, then

$$\delta(\mu_s) \geq \max_k \left\{ \Phi\left(\Phi^{-1}(p_k) + \frac{D}{L} \right) - p_k \right\}. \tag{7}$$

Equivalently, defining $b(p, x) := \Phi\big(\Phi^{-1}(p) + x\big) - p$, we have $\delta(\mu_s) \geq \max_k b\big(p_k, \frac{D}{L}\big) > 0$.

(iii) In the two-region case with equal masses $p_1 = p_2 = (1 - \delta)/2$, any L-Lipschitz generator satisfying $\delta(\mu_s) = \delta$ obeys

$$L \geq \frac{D}{2\Phi^{-1}\left(\frac{1+\delta}{2}\right)} = \frac{D}{\sqrt{2\pi}\,\delta}(1 + o(1)), \qquad \delta \to 0. \tag{8}$$

(iv) The connectivity barrier can be removed by introducing a discrete mixture of continuous generators. In particular, if $K \geq K^\star$ and each $\mathcal{G}_k$ is nonempty, there exist continuous maps $\{F_{s,z}\}_{z=1}^K$ whose images are contained in the corresponding high-value regions. Hence the mixture

$$\mu_s = \sum_{z=1}^{K} \pi_z (F_{s,z})_{\#}\nu \tag{9}$$

can achieve $\delta(\mu_s) = 0$, while its support has at most K connected components.

According to this theorem, for example, in the two-region case, when $p_k = 0.4$ and $D/L = 0.5$, $b(0.4, 0.5) \approx 0.197$, so at least 19.7% of actions sampled by a single continuous map generator will lie outside the high-value regions. The proof of Theorem 1 is given in Appendix A.2.

4 MODEFLOW: MODE-CONDITIONED MEANFLOW POLICIES

MODEFLOW addresses the connectivity barrier of one-step flow policies by replacing the single generator F_s with a discrete family $\{F_{s,z}\}_{z=1}^K$. Each mode captures a distinct behavior, while Gaussian noise models variation within the mode. We obtain the modes once from offline trajectories, condition both the MeanFlow actor and Q-critic on the same mode embedding. Thus, MODEFLOW enables disconnected proposal support while retaining one-step generation.

4.1 MODE-CONDITIONED ONE-STEP GENERATOR TRAINING

Trajectory-level mode discovery. The discrete mode is intended to capture qualitatively distinct strategies rather than instantaneous differences between actions. Such distinctions are often properties of entire trajectories. For example, the order in which objects are manipulated may define different behavioral modes, yet cannot be reliably inferred from an individual transition. For each offline trajectory $\tau = (o_{0:T}, a_{0:T})$, we therefore construct a fixed-dimensional temporal descriptor

$$\psi(\tau) = \big[o_{t_1}, a_{t_1}, \ldots, o_{t_B}, a_{t_B}, o_0, o_T\big], \tag{10}$$

where $B = 50$ temporal locations are uniformly sampled over the normalized trajectory duration. The initial and final observations provide additional trajectory-level context. We apply mini-batch K-means (Sculley, 2010) on at most 3,000 offline trajectories for clustering. Each trajectory is assigned the label $z(\tau) \in \{1, \ldots, K\}$ of its nearest cluster center, and all transitions within the trajectory inherit this label. This procedure is performed once before online interaction and uses neither rewards nor success labels. The resulting modes are therefore identified entirely offline, with no additional mode inference required at deployment. Importantly, the mode label is assigned at the trajectory level rather than to individual transitions, encouraging each mode to capture a coherent behavioral strategy.

Mode-conditioned one-step generation. For each mode z, MODEFLOW defines a continuous proposal branch by conditioning the MeanFlow velocity network on a learned mode embedding $e(z)$. Specifically, $e(z)$ is concatenated with a linear projection of (s, x_t, t, r) and is held fixed throughout the noise-to-action transport. Thus, conditioned on z, each branch is an ordinary MeanFlow policy, and the MeanFlow identity and IVC boundary condition of Zhan et al. (2026) apply independently to every branch. The action generator is trained with the conditional MVP objective

$$\mathcal{L}_\pi(\theta) = \mathbb{E}_{(s,A,z)\sim\mathcal{B}}\Big[\ell\big(u_\theta(s, x_t, t, r, z) - \mathrm{sg}[u_{\mathrm{tgt}}]\big) + \lambda\ell\big(u_\theta(s, x_t, t, t, z) - v\big)\Big], \tag{11}$$

where $u_{\mathrm{tgt}} = v - (t - r)(\partial_t u_\theta + \nabla_x u_\theta v)$, $\lambda = 1$, and each H-step replay window inherits the mode label of its first transition. Generation remains one-step for each candidate,

$$A = F_{s,z}(\epsilon) = \mathrm{clip}\,\big(\epsilon - u_\theta(s, \epsilon, 1, 0, z), -1, 1\big). \tag{12}$$

The mode label therefore specifies the behavioral strategy, while the Gaussian noise captures variation within that strategy.

4.2 ONLINE MODE-AWARE VALUE SELECTION

Mode-stratified best-of-N selection. Given a fixed proposal budget N with $K \mid N$, we allocate exactly N/K proposals to each mode. All N candidates are generated in a single batched network evaluation and jointly scored by the mode-aware critic:

$$(A^\star, z^\star) = (A_{i^\star}, z_{i^\star}), \qquad i^\star = \arg\max_{i \le N} \bar{Q}_\phi(s, A_i, z_i), \qquad A_i = F_{s,z_i}(\epsilon_i). \tag{13}$$

Thus, MODEFLOW retains the same one-step generation cost as MVP at a given N, while explicitly reserving proposal budget for every behavioral mode. The critic $\bar{Q}_\phi(s, A, z)$ conditions on the mode embedding and is trained by chunked TD learning during both offline and online phases. The algorithm is provided in Appendix B.

$$\mathcal{L}_Q(\phi) = \mathbb{E}_{\mathcal{B}}\Big[\sum_{m=1}^{M} \big(Q_\phi^{(m)}(s, A, z) - R_H - \beta\, \bar{Q}_{\phi^-}(s', A'^\star, z'^\star)\big)^2 \Big] \tag{14}$$

Moreover, we theoretically prove the return improvement of MODEFLOW over one-step policies. In particular, the connectivity barrier of a single continuous generator induces a nonzero probability of missing the optimal behavior, which translates into a return gap.

Consider the single-state macro-MDP induced by Section 3, with discount $\beta = \gamma^H$. Among the $K^\star$ viable regions, let $\mathcal{G}^\star$ denote the optimal one, with

$$R(A) = R_+ \ \text{ on } \mathcal{G}^\star, \qquad R(A) \le R_+ - \Delta \ \text{ on } \mathcal{A} \setminus \mathcal{G}^\star, \qquad \Delta > 0. \tag{15}$$

The baseline π_B is policy that generates N i.i.d. candidates from a continuous single-map L-Lipschitz action generator F_s, and selects one according to an arbitrary selection rule. In contrast, MODEFLOW, denoted π_C, allocates proposals across the mode-conditioned branches of Theorem 1(iv), ensuring the branch containing $\mathcal{G}^\star$ contributes a candidate. Then we have the following theorem.

Theorem 2 (Return improvement of MODEFLOW)**.** *Assume the reward model Equation* (15) *and the critic error satisfies* $\big|\bar{Q}_\phi(s, A_i, z_i) - Q^\pi(s, A_i)\big| \le \varepsilon_Q$ *on every proposed candidate, and let* $\rho := 1 - \mu_s(\mathcal{G}^\star)$ *denote the probability that a single baseline proposal misses the optimal region. Then*

$$J(\pi_C) - J(\pi_B) \ge \frac{\Delta \rho^N - 2\varepsilon_Q}{1 - \beta}, \qquad \rho \ge \delta(\mu_s) \ge b\left(\mu_s(\mathcal{G}^\star), \tfrac{D}{L}\right) > 0, \tag{16}$$

whenever F_s places positive mass on a second viable region.

5 EXPERIMENTS

We evaluate MODEFLOW to answer three questions: **(RQ1)** Does MODEFLOW outperform state-of-the-art flow policies, especially on hard long-horizon tasks? **(RQ2)** What makes MODEFLOW more effective? **(RQ3)** How does the number of modes K affect performance?

5.1 EXPERIMENTAL SETTING

Setup. We follow the protocol of Zhan et al. (2026): ten sparse-reward manipulation tasks from Robomimic (`lift`, `can`, `square`; multi-human data) (Mandlekar et al., 2021) and OGBench (`cube-double` tasks 2–4 and `cube-triple` tasks 2–5; play data) (Park et al., 2025a). All the compared methods are trained 1M offline followed by 1M online steps on five seeds. We report the final success rate and the online success rate, i.e., the mean success over the online phase, which measures how quickly interaction is converted into performance. MODEFLOW keeps all hyperparameters of MVP and uses $N = 32$ proposals. Further details are in Appendix C.

Baselines. We compare with the four methods: FQL (Park et al., 2025b), a one-step policy distilled from a flow; BFN, best-of-N sampling from a flow policy (Ghasemipour et al., 2021); QC, which adds action chunking to BFN (Li et al., 2026); and MVP (Zhan et al., 2026), the state of the art on this benchmark.

Table 1: **Results on Robomimic and OGBench** (mean ± SD over five seeds). **Bold**: best; underline: second best. Averages are over the ten benchmark tasks. Modeflow outperforms the baselines on almost all of the tasks.

Task	FQL	BFN	QC	MVP	**MODEFLOW (Ours)**
Robomimic-lift	0.96 ± 0.03	$\mathbf{1.00 \pm 0.01}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$
Robomimic-can	0.74 ± 0.11	0.82 ± 0.03	0.94 ± 0.06	0.92 ± 0.07	$\mathbf{0.98 \pm 0.06}$
Robomimic-square	0.12 ± 0.05	0.34 ± 0.06	0.92 ± 0.01	0.93 ± 0.01	$\mathbf{0.99 \pm 0.01}$
Cube-double-task2	0.95 ± 0.04	0.88 ± 0.05	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$
Cube-double-task3	0.97 ± 0.04	0.90 ± 0.06	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$
Cube-double-task4	0.08 ± 0.04	0.35 ± 0.09	0.93 ± 0.08	0.95 ± 0.04	$\mathbf{1.00 \pm 0.00}$
Cube-triple-task2	0.01 ± 0.02	0.08 ± 0.06	0.82 ± 0.10	0.88 ± 0.03	$\mathbf{1.00 \pm 0.01}$
Cube-triple-task3	0.12 ± 0.13	0.26 ± 0.14	0.69 ± 0.05	0.71 ± 0.06	$\mathbf{0.99 \pm 0.01}$
Cube-triple-task4	0.00 ± 0.00	0.02 ± 0.02	0.46 ± 0.13	0.52 ± 0.11	$\mathbf{0.99 \pm 0.01}$
Cube-triple-task5	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.98 \pm 0.01}$
Average Final Success	0.40 ± 0.05	0.47 ± 0.05	0.78 ± 0.04	0.79 ± 0.03	$\mathbf{0.99 \pm 0.01}$
Average Online Success	0.39	0.35	0.65	0.64	**0.92**

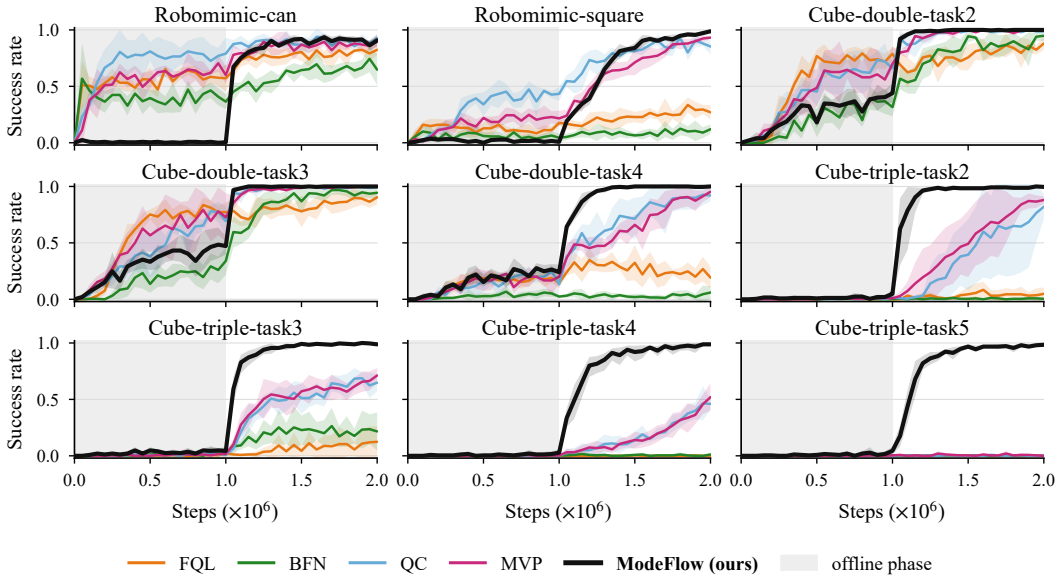

Figure 3: **Learning curves** (mean and 95% band over five seeds; gray: offline pre-training). MODEFLOW converges early in the online phase where the baselines improve slowly or not at all.

5.2 MAIN RESULTS

Answer for RQ1. As Tab. 1 and Fig. 3 shows, MODEFLOW achieves the best average final and online success rates. Its advantage is concentrated on the hardest tasks: it succeeds in 99% of the episodes on Cube-triple-task4, where the best baseline reaches about half, and in 98% on Cube-triple-task5, where every baseline fails. These are precisely the tasks that Section 3 identifies as hard for a single continuous map. Rearranging three cubes requires committing to one of several placement orders, so the useful chunks at a state form separated regions. A single map either spends proposals on the infeasible chunks between them or concentrates on the most frequent order in the data, and best-of-N selection can only choose among what is proposed. MODEFLOW gives every order its own branch and allocates proposals to every branch, so the critic always compares the competing strategies at every decision. Where a single strategy suffices, as on lift and double-cube tasks 2–3, all chunked methods saturate and the two designs coincide in performance.

Answer for RQ2. As Fig. 4 shows, MODEFLOW exhibits a clear advantage early in training: after only 250k online steps, it succeeds in more than 80% of episodes on every hard task, whereas MVP

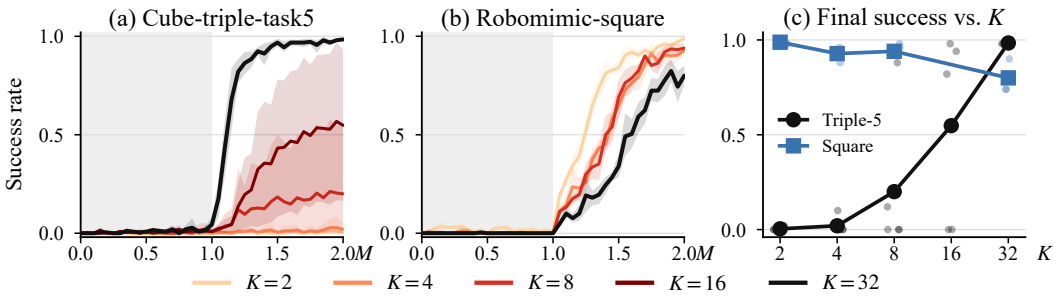

Figure 5: **Effect of the number of modes K at a fixed budget $N = 32$.**

remains substantially lower. Tab. 1 further supports this advantage. The substantially higher online success rate of MODEFLOW indicates faster convergence during online learning.

Theorem 2 makes the mechanism explicit: a single continuous generator loses return whenever all its proposals miss the optimal behavior, and no selection rule can recover from such a candidate set. MODEFLOW avoids this failure by assigning each behavior mode its own branch and reserving proposal budget for each mode. Online learning can then refine these branches through improved proposals and critic estimates. Fig. 1 provides a controlled illustration of the same mechanism, showing how mode separation reduces mass on connecting bridges and improves coverage of disconnected high-value regions.

Answer for RQ3. As shown in Fig. 5, with the proposal budget fixed, performance on Cube-triple-task5 improves sharply as K increases, with only the finest resolution solving the task. In contrast, Robomimic-square performs best with the coarsest resolution, while overly fine resolutions slow learning. This difference reflects the strategic diversity of the two tasks. Cube-triple-task5 admits multiple distinct placement orders. With too few modes, each branch must cover several orders, reintroducing bridge masses within the branch. As a result, discovering a successful order becomes increasingly dependent on stochastic exploration. A finer mode resolution instead separates these orders into distinct branches, allowing the proposal budget to cover them more effectively.

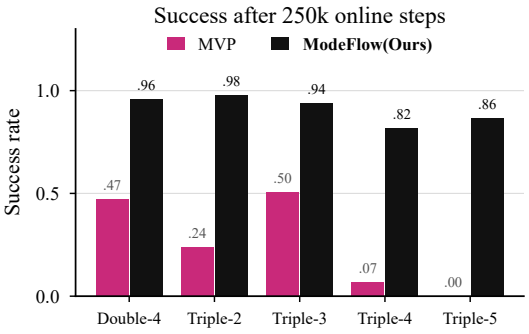

Figure 4: **Success rate after 250k online steps**.

Robomimic-square, by contrast, is dominated by a single strategy with variation mainly in its execution. Excessive mode splitting fragments this behavior across near-duplicate branches, diluting the proposal budget and slowing concentration on the useful strategy. Thus, the appropriate mode resolution should reflect the underlying strategic diversity: finer modes are beneficial for tasks with multiple distinct strategies, whereas coarser modes are preferable when a task is dominated by a single strategy.

Ablation study on Mode-aware critic. To isolate the role of z in the critic, we remove z while keeping the mode-conditioned generator and stratified proposals unchanged, yielding $i^\star = \arg\max_i Q_\phi(s, A_i)$. The candidate set is therefore identical. As the results in Tab. 2, this label is important because candidates from different strategies can be close in action space but differ greatly in value, conditioning on z allows the critic to model their values separately rather than interpolate across strategies.

Inference cost. Mode conditioning adds only a label embedding, and all K strata are generated in a single batched forward pass. Thus, MODEFLOW requires one generator and one critic evaluation per action chunk. As shown in Tab. 3, MODEFLOWachieves faster inference than BFN and QC, which rely on multi-step flows, while maintaining an inference time comparable to that of FQL and MVP.

Table 2: **Ablation results on Mode-aware critic** (final success). *No* z: the critic scores (s, A) only.

Task	No z	**MODEFLOW(Ours)**
Cube-double-task4	0.82 ± 0.03	1.00 ± 0.00
Cube-triple-task4	0.71 ± 0.16	0.99 ± 0.01
Robomimic-square	0.95 ± 0.03	0.99 ± 0.01

Table 3: **Inference time per observation (ms) on Cube-triple-task5.**

FQL	BFN	QC	MVP	**MODEFLOW(Ours)**
12.88 ± 1.22	124.57 ± 16.46	106.63 ± 12.97	10.99 ± 0.97	10.72 ± 0.93

6 RELATED WORK

Generative policies and one-step generation. Diffusion and flow models are expressive policy classes for offline and online RL (Janner et al., 2022; Chi et al., 2025; Wang et al., 2023; Ren et al., 2025), and one line of work uses them purely as proposal distributions that a critic ranks (Ghasemipour et al., 2021; Chen et al., 2023; Li et al., 2026), the mechanism shared by MVP and MODEFLOW. Because iterative sampling is expensive in online RL, recent methods compress generation into a single step by distillation (Park et al., 2025b; Wang et al., 2024), consistency training (Song & Dhariwal, 2024; Ding & Jin, 2024), or direct learning of flow maps and average velocities (Boffi et al., 2025; Frans et al., 2025; Geng et al., 2026), of which MVP (Zhan et al., 2026) is the strongest on manipulation benchmarks. All of them apply a map that is continuous in Gaussian noise and therefore face the barrier of Theorem 1; MODEFLOW keeps MVP's generator and objective and turns the policy into a family of conditional maps.

Topology, multimodality, and discrete structure. The exact flow of a Lipschitz ODE is a homeomorphism (Dupont et al., 2019; Zhang et al., 2020), making disconnected targets difficult for a single push-forward generator and motivating mixtures or additional latent structure (Cornish et al., 2020; Salmona et al., 2022; Khayatkhoei et al., 2018). Tanielian et al. (2020) further characterize the unavoidable off-manifold mass of a single generator via the Gaussian isoperimetric inequality, which underlies Theorem 1. Recent one-step generators introduce discrete structure through mixture noise (Wang et al., 2026), primarily for multimodal sample generation. In RL, however, proposals are evaluated and selected by a learned critic, creating additional questions of how to allocate a fixed proposal budget across modes and how coverage translates into policy improvement. While discrete structure is also central to skill learning and options (Shafiullah et al., 2022; Lee et al., 2024), MODEFLOW re-selects the mode at every chunk boundary rather than committing to a skill or learning a separate high-level policy.

Offline-to-online RL. Offline RL should control distribution shift (Levine et al., 2020) through conservative value estimates (Kumar et al., 2020), in-sample maximization (Kostrikov et al., 2021), or behavior regularization (Fujimoto & Gu, 2021; Tarasov et al., 2023). Online fine-tuning adds advantage-weighted updates (Nair et al., 2020), balanced replay with pessimistic ensembles (Lee et al., 2022), calibrated value initialization (Nakamoto et al., 2023), and symmetric sampling of offline and online data (Ball et al., 2023), while action chunking improves exploration and value propagation on long horizons (Zhao et al., 2023; Li et al., 2026). These methods change how values are learned and regularized. MODEFLOW keeps MVP's learning rules unchanged and modifies the policy class that produces the candidates, which our experiments identify as the factor that governs how quickly online interaction pays off.

7 CONCLUSION

We identify a fundamental connectivity barrier in one-step flow policies, which induces unavoidable bridge mass between separated high-value behaviors, and overcome it with MODEFLOW through mode-conditioned generation. Our theory shows that this advantage cannot be recovered downstream because the barrier forces a single continuous generator to miss the optimal behavior with nonzero probability, MODEFLOW retains a return advantage over any one-step policy of this class, whatever rule it uses to select among its proposals. Across two manipulation benchmarks, MODEFLOW consistently outperforms existing flow policies and solves long-horizon tasks on which all baselines fail. Future work will explore value-aware mode discovery and adaptive mode resolution.

Reproducibility Statement

We will open the source code after the paper is accepted. We have provided theoretical analysis in the Appendix A. We have also provided implementation details in the Appendix C.

AI use statement

Large language models were used to help edit the text, prepare plotting code, and check derivations. All claims, proofs, and experimental results were verified by the authors, who take full responsibility for the content.

References

Anurag Ajay, Aviral Kumar, Pulkit Agrawal, Sergey Levine, and Ofir Nachum. Opal: Offline primitive discovery for accelerating offline reinforcement learning. *arXiv preprint arXiv:2010.13611*, 2020.

Philip J Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning*, pp. 1577–1594. PMLR, 2023.

Nicholas Matthew Boffi, Michael Samuel Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.

Christer Borell. The brunn-minkowski inequality in gauss space. *Inventiones mathematicae*, 30(2): 207–216, 1975.

Huayu Chen, Cheng Lu, Chengyang Ying, Hang Su, and Jun Zhu. Offline reinforcement learning via high-fidelity generative behavior modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

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

Rob Cornish, Anthony Caterini, George Deligiannidis, and Arnaud Doucet. Relaxing bijectivity constraints with continuously indexed normalising flows. In *International conference on machine learning*, pp. 2133–2143. PMLR, 2020.

Zihan Ding and Chi Jin. Consistency models as a rich and efficient policy class for reinforcement learning. In *International Conference on Learning Representations*, volume 2024, pp. 53047–53066, 2024.

Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural odes. *Advances in neural information processing systems*, 32, 2019.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations*, volume 2025, pp. 34668–34684, 2025.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *Advances in Neural Information Processing Systems*, 38:75460–75482, 2026.

Seyed Kamyar Seyed Ghasemipour, Dale Schuurmans, and Shixiang Shane Gu. Emaq: Expected-max q-learning operator for simple yet effective offline and online rl. In *International Conference on Machine Learning*, pp. 3682–3691. PMLR, 2021.

Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9902–9915. PMLR, 17–23 Jul 2022.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the nineteenth international conference on machine learning*, pp. 267–274, 2002.

Mahyar Khayatkhoei, Maneesh K Singh, and Ahmed Elgammal. Disconnected manifold learning for generative adversarial networks. *Advances in Neural Information Processing Systems*, 31, 2018.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191, 2020.

Seunghyun Lee, Younggyo Seo, Kimin Lee, Pieter Abbeel, and Jinwoo Shin. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble. In *Conference on Robot Learning*, pp. 1702–1712. PMLR, 2022.

Seungjae Lee, Yibin Wang, Haritheja Etukuru, H. Jin Kim, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Behavior generation with latent actions. In *Multi-modal Foundation Model meets Embodied AI Workshop @ ICML2024*, 2024.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Qiyang Li, Zhiyuan Paul Zhou, and Sergey Levine. Reinforcement learning with action chunking. *Advances in Neural Information Processing Systems*, 38:55518–55553, 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. *arXiv preprint arXiv:2108.03298*, 2021.

Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Mitsuhiko Nakamoto, Simon Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning. *Advances in Neural Information Processing Systems*, 36:62244–62269, 2023.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations*, volume 2025, pp. 94937–94982, 2025a.

Seohong Park, Qiyang Li, and Sergey Levine. Flow q-learning. In *Forty-second International Conference on Machine Learning*, 2025b.

Allen Ren, Justin Lidard, Lars Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. In *International Conference on Learning Representations*, volume 2025, pp. 77288–77329, 2025.

Antoine Salmona, Valentin De Bortoli, Julie Delon, and Agnes Desolneux. Can push-forward generative models fit multimodal distributions? *Advances in Neural Information Processing Systems*, 35:10766–10779, 2022.

David Sculley. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pp. 1177–1178, 2010.

Nur Muhammad Shafiullah, Zichen Cui, Ariuntuya Arty Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. *Advances in neural information processing systems*, 35:22955–22968, 2022.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *International Conference on Learning Representations*, volume 2024, pp. 15078–15097, 2024.

Vladimir N Sudakov. Extremal properties of half-spaces for spherically invariant measures. *Zap. Nauch. Sem. LOMI*, 41:14–24, 1974.

Ugo Tanielian, Thibaut Issenhuth, Elvis Dohmatob, and Jeremie Mary. Learning disconnected manifolds: a no gan's land. In *International Conference on Machine Learning*, pp. 9418–9427. PMLR, 2020.

Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36:11592–11620, 2023.

Junli Wang, Yinan Zheng, Xueyi Liu, Zebin Xing, Pengfei Li, Kun Ma, Hangjun Ye, Guang Chen, Guang Li, Long Chen, et al. Meanfuser: Fast one-step multi-modal trajectory generation and adaptive reconstruction via meanflow for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17884–17893, 2026.

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

Zhendong Wang, Zhaoshuo Li, Ajay Mandlekar, Zhenjia Xu, Jiaojiao Fan, Yashraj Narang, Linxi Fan, Yuke Zhu, Yogesh Balaji, Mingyuan Zhou, et al. One-step diffusion policy: Fast visuomotor policies via diffusion distillation. *arXiv preprint arXiv:2410.21257*, 2024.

Guojian Zhan, Letian Tao, Pengcheng Wang, Yixiao Wang, Yuxin Chen, Yiheng Li, Hongyang Li, Masayoshi Tomizuka, and Shengbo Li. Mean flow policy with instantaneous velocity constraint for one-step action generation. In *International Conference on Learning Representations*, volume 2026, pp. 157132–157154, 2026.

Han Zhang, Xi Gao, Jacob Unterman, and Tom Arodz. Approximation capabilities of neural odes and invertible residual networks. In *International Conference on Machine Learning*, pp. 11086–11095. PMLR, 2020.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *ICML Workshop on New Frontiers in Learning, Control, and Dynamical Systems*, 2023.

A PROOFS

This appendix proves Theorem 1 (Section A.2) and Theorem 2 (Section A.3), and then states and proves the general policy-improvement bound used to interpret the ablations of Section 5 (Section A.4).

A.1 NOTATION AND STANDARD TOOLS

Throughout, $\nu = \mathcal{N}(0, I_m)$ is the standard Gaussian measure on $\mathbb{R}^m$, which has full support: $\nu(U) > 0$ for every nonempty open U. For a set $B \subseteq \mathbb{R}^m$ and $r > 0$,

$$B^r := \{x \in \mathbb{R}^m : \operatorname{dist}(x, B) < r\} \tag{17}$$

denotes its open r-enlargement. We use two standard results.

Lemma 1 (Gaussian isoperimetric inequality; Borell, 1975; Sudakov, 1974)**.** *For every Borel set $B \subseteq \mathbb{R}^m$ and every $r > 0$,*

$$\nu(B^r) \;\geq\; \Phi\big(\Phi^{-1}(\nu(B)) + r\big). \tag{18}$$

Lemma 2 (Performance difference; Kakade & Langford, 2002)**.** *For policies π, π' in a discounted MDP with discount β and initial distribution ρ_0,*

$$J(\pi') - J(\pi) = \frac{1}{1-\beta}\, \mathbb{E}_{s\sim d^{\pi'}}\, \mathbb{E}_{A\sim\pi'(\cdot|s)}\big[Q^{\pi}(s, A) - V^{\pi}(s)\big], \tag{19}$$

where $d^{\pi'}(s) = (1-\beta)\sum_{t\geq 0} \beta^t \mathbb{P}(s_t = s \mid \pi', \rho_0)$ is the normalized discounted occupancy.

We apply Lemma 2 to the macro-MDP whose actions are chunks, whose reward is R_H, and whose discount is $\beta = \gamma^H$. Episode termination is modeled by an absorbing zero-reward state.

A.2 PROOF OF THEOREM 1 (CONNECTIVITY BARRIER)

Write $F = F_s$ and $\mu = F_{\#}\nu$ throughout this subsection.

(i) Support and connectivity. We first identify the support. If $y \in \overline{F(\mathbb{R}^m)}$ and $U \ni y$ is open, then U meets $F(\mathbb{R}^m)$, so $F^{-1}(U)$ is a nonempty open set and

$$\mu(U) = \nu\big(F^{-1}(U)\big) > 0, \tag{20}$$

because ν has full support. Hence $y \in \operatorname{supp}(\mu)$. Conversely, if $y \notin \overline{F(\mathbb{R}^m)}$, some open $U \ni y$ is disjoint from $F(\mathbb{R}^m)$ and $\mu(U) = 0$. Therefore

$$\operatorname{supp}(\mu) = \overline{F(\mathbb{R}^m)}. \tag{21}$$

Since $\mathbb{R}^m$ is connected and F is continuous, $F(\mathbb{R}^m)$ is connected, and so is its closure.

Now suppose $\mu(\mathcal{G}_1) > 0$ and $\mu(\mathcal{G}_2) > 0$. Then $\operatorname{supp}(\mu)$ meets both $\mathcal{G}_1$ and $\mathcal{G}_2$, since otherwise $\mu(\mathcal{G}_k) \leq \mu(\mathbb{R}^m \setminus \operatorname{supp}\mu) = 0$. Put

$$S_1 = \operatorname{supp}(\mu) \cap \mathcal{G}_1, \qquad S_2 = \operatorname{supp}(\mu) \cap \textstyle\bigcup_{k\geq 2} \mathcal{G}_k. \tag{22}$$

Both sets are closed, nonempty and disjoint, because $\mathcal{G}_1$ and $\bigcup_{k\geq 2}\mathcal{G}_k$ are closed and at distance at least $D > 0$. If we had $\operatorname{supp}(\mu) \subseteq \mathcal{G}$, then $\operatorname{supp}(\mu) = S_1 \cup S_2$ would be a disconnection of a connected set, which is impossible. Hence there exists $y \in \operatorname{supp}(\mu) \setminus \mathcal{G}$. As $\mathcal{G}$ is closed, some open $U \ni y$ is disjoint from $\mathcal{G}$, and therefore $\delta(\mu) \geq \mu(U) > 0$.

(ii) Quantitative bound. Fix k with $p_k = \mu(\mathcal{G}_k) \in (0, 1)$ and let $B_j = F^{-1}(\mathcal{G}_j)$, which is closed by continuity of F. For $x \in B_k$ and $y \in B_j$ with $j \neq k$,

$$D \;\leq\; \operatorname{dist}(\mathcal{G}_k, \mathcal{G}_j) \;\leq\; \|F(x) - F(y)\| \;\leq\; L\|x - y\|, \tag{23}$$

so the preimages inherit the separation,

$$\operatorname{dist}(B_k, B_j) \;\geq\; \frac{D}{L}, \qquad j \neq k. \tag{24}$$

Take any $0 < r < D/L$. By Equation (24), the enlargement B_k^r is disjoint from every B_j with $j \neq k$. A point $x \in B_k^r \setminus B_k$ therefore satisfies $F(x) \notin \mathcal{G}_k$ (since $x \notin B_k$) and $F(x) \notin \mathcal{G}_j$ for all $j \neq k$, that is, $F(x) \notin \mathcal{G}$. Consequently, using $\nu(B_k) = p_k$ and Lemma 1,

$$\delta(\mu) \;\geq\; \nu\big(B_k^r \setminus B_k\big) \;=\; \nu(B_k^r) - \nu(B_k) \;\geq\; \Phi\big(\Phi^{-1}(p_k) + r\big) - p_k. \tag{25}$$

Letting $r \uparrow D/L$ and using the continuity of Φ gives $\delta(\mu) \geq b(p_k, D/L)$, and maximizing over k proves (ii). Clipping the generator output to $[-1, 1]$ is a 1-Lipschitz operation and does not increase L. Finally $b(p, x) > 0$ for $x > 0$, because Φ is strictly increasing.

(iii) Lipschitz blow-up. Let $p_1 = p_2 = p = (1 - \delta)/2$. Part (ii) applied to $k = 1$ gives

$$\Phi\Big(\Phi^{-1}(p) + \tfrac{D}{L}\Big) - p \;\leq\; \delta, \qquad \text{i.e.} \qquad \Phi\Big(\Phi^{-1}\big(\tfrac{1-\delta}{2}\big) + \tfrac{D}{L}\Big) \;\leq\; \tfrac{1+\delta}{2}. \tag{26}$$

Applying the increasing function Φ^{-1} to both sides and using the symmetry $\Phi^{-1}\big(\frac{1-\delta}{2}\big) = -\Phi^{-1}\big(\frac{1+\delta}{2}\big)$ yields

$$\frac{D}{L} \;\leq\; 2\,\Phi^{-1}\Big(\tfrac{1+\delta}{2}\Big), \tag{27}$$

which is the stated bound. Since $\Phi^{-1}\big(\frac{1}{2} + x\big) = \sqrt{2\pi}\,x + O(x^3)$ as $x \to 0$, the right-hand side equals $\sqrt{2\pi}\,\delta + O(\delta^3)$, and therefore $L \geq D/(\sqrt{2\pi}\,\delta)\,(1 + O(\delta^2))$.

(iv) Discrete conditioning removes the barrier. For weights $w_z > 0$ the support of the finite mixture $\sum_z w_z \mu_z$ is $\bigcup_z \operatorname{supp}(\mu_z)$. By part (i) each $\operatorname{supp}(\mu_z)$ is connected and hence contained in a single connected component of the union; every component contains at least one of them, so there are at most K components.

For the construction, pick a closed ball $\overline{B}(c_k, \varrho) \subseteq \mathcal{G}_k$ inside each region, assign every region at least one label z (possible because $K \geq K^\star$), and define

$$F_z(\epsilon) = c_{k(z)} + \varrho\,\frac{\epsilon}{\max(1, \|\epsilon\|)}. \tag{28}$$

The radial retraction onto the unit ball is 1-Lipschitz, so F_z is ϱ-Lipschitz with a constant that does not depend on the separation D, and $F_z(\mathbb{R}^m) \subseteq \overline{B}(c_{k(z)}, \varrho) \subseteq \mathcal{G}$. Hence $\delta = 0$ for every D. □

A.3 Proof of Theorem 2 (return improvement of ModeFlow)

In a single-state macro-MDP every chunk returns to the same state, so for any stationary policy π',

$$J(\pi') = \frac{\mathbb{E}_{A \sim \pi'}[R(A)]}{1 - \beta}, \qquad Q^\pi(s, A) = R(A) + \beta V^\pi(s). \tag{29}$$

In particular $Q^\pi(s, \cdot)$ and $R(\cdot)$ differ by the constant $c := \beta V^\pi(s)$, which does not depend on the action. It therefore suffices to compare the two policies in expected reward and divide by $1 - \beta$.

Step 1: the baseline loses Δ whenever it misses $\mathcal{G}^\star$. Let $A_1, \ldots, A_N \overset{\text{i.i.d.}}{\sim} \mu_s$ be the baseline proposals and let

$$E := \big\{A_i \notin \mathcal{G}^\star \;\text{ for all } i \leq N\big\}, \qquad \mathbb{P}(E) = \big(1 - \mu_s(\mathcal{G}^\star)\big)^N = \rho^N. \tag{30}$$

The selector of π_B is an arbitrary measurable function of the candidate set, so the action it returns is always one of the A_i. On the event E every candidate lies in $\mathcal{A} \setminus \mathcal{G}^\star$ and hence earns at most $R_+ - \Delta$ by Equation (15); on E^c no candidate earns more than R_+. Therefore

$$\mathbb{E}\big[R(A_{\pi_B})\big] \;\leq\; (1 - \rho^N)\,R_+ \;+\; \rho^N\,(R_+ - \Delta) \;=\; R_+ - \Delta\,\rho^N. \tag{31}$$

This holds for *every* selection rule, including an oracle that observes R exactly, because the bound constrains the candidate set rather than the choice made within it.

Step 2: MODEFLOW is within $2\varepsilon_Q$ of the optimum. By Theorem 1(iv) each branch satisfies $\mu_{s,z}(\mathcal{G}_{k(z)}) = 1$, and some branch $z^\star$ is assigned to the optimal region, $\mu_{s,z^\star}(\mathcal{G}^\star) = 1$. Stratification gives $n_{z^\star} \geq 1$, so the candidate set contains at least one index j with $A_j \in \mathcal{G}^\star$ and hence

$$R(A_j) = R_+ = \max_{A \in \mathcal{A}} R(A). \tag{32}$$

Let $\hat{\imath} = \arg\max_{i \leq N} \bar{Q}_\phi(s, A_i, z_i)$ be the index selected by Equation (13). Since $Q^\pi(s, A) = R(A) + c$, the critic-error assumption reads $|\bar{Q}_\phi(s, A_i, z_i) - R(A_i) - c| \leq \varepsilon_Q$ for every candidate. Applying this bound at $\hat{\imath}$, then the optimality of $\hat{\imath}$ for $\bar{Q}_\phi$, then the bound again at j,

$$R(A_{\hat{\imath}}) \;\geq\; \bar{Q}_\phi(s, A_{\hat{\imath}}, z_{\hat{\imath}}) - c - \varepsilon_Q \;\geq\; \bar{Q}_\phi(s, A_j, z_j) - c - \varepsilon_Q \;\geq\; R(A_j) - 2\varepsilon_Q \;=\; R_+ - 2\varepsilon_Q. \tag{33}$$

Step 3: combine. Subtracting Equation (31) from Equation (33) and dividing by $1 - \beta$ using Equation (29) gives Equation (16).

Step 4: the coverage gap is bounded below by the bridge mass. Suppose F_s places positive mass on a second high-value region, so that Theorem 1 applies with $p^\star := \mu_s(\mathcal{G}^\star) \in (0, 1)$. The sets $\mathcal{G}^\star$ and $\mathcal{A} \setminus \mathcal{G}$ are disjoint, so $p^\star + \delta(\mu_s) \leq 1$, and Theorem 1(ii) applied to k with $\mathcal{G}_k = \mathcal{G}^\star$ gives

$$\rho = 1 - p^\star \;\geq\; \delta(\mu_s) \;\geq\; b\Big(p^\star, \tfrac{D}{L}\Big) > 0, \tag{34}$$

which is the second inequality of Equation (16). Raising it to the N-th power shows that Equation (16) also implies the bridge-mass form $\big(\Delta\, b^N - 2\varepsilon_Q\big)/(1 - \beta)$. □

Remark 1 (Branches with residual bridge mass). If the branches are learned rather than constructed and satisfy $\mu_{s,z^\star}(\mathcal{G}^\star) = 1 - \eta$ with $\eta > 0$, and rewards are bounded below by $R_+ - W$, then Step 2 yields $\mathbb{E}[R(A_{\hat{\imath}})] \geq R_+ - 2\varepsilon_Q - W\eta^{\,n_{z^\star}}$, and Equation (16) becomes

$$J(\pi_C) - J(\pi_B) \;\geq\; \frac{\Delta\, \rho^N - W\eta^{\,n_{z^\star}} - 2\varepsilon_Q}{1 - \beta}. \tag{35}$$

The guarantee therefore degrades gracefully: it survives as long as the optimal branch is accurate enough that $W\eta^{\,n_{z^\star}}$ stays below the coverage gain.

A.4 A GENERAL POLICY-IMPROVEMENT BOUND

Theorem 2 is stated in the single-state model, where the effect of coverage can be read off directly. The following proposition is the corresponding statement for the general multi-state macro-MDP. It decomposes improvement into a *coverage* term and a *ranking* term, which is the decomposition we use in Section 5 to interpret the effect of the number of modes and of removing z from the critic. We first record the elementary fact that allocating the budget across branches does not reduce the best-of-N value.

Lemma 3 (Stratified proposals dominate i.i.d. proposals)**.** *Let $\mu_{s,z} = (F_{s,z})_{\#}\nu$, let n_z be the number of proposals drawn from branch z with $N = \sum_z n_z$, let $w_z = n_z/N$ and $\bar{\mu}_s = \sum_z w_z \mu_{s,z}$. Let $q : \mathcal{A} \to \mathbb{R}$ be bounded, and let $G_z(t) = \mathbb{P}_{A \sim \mu_{s,z}}(q(A) \leq t)$. Let M_S be the maximum of q over the stratified proposals and M_I its maximum over N i.i.d. proposals from $\bar{\mu}_s$. Then for all t,*

$$\mathbb{P}(M_S \leq t) = \prod_z G_z(t)^{n_z} \;\leq\; \Big(\sum_z w_z G_z(t)\Big)^N = \mathbb{P}(M_I \leq t), \tag{36}$$

and consequently $\mathbb{E}[M_S] \geq \mathbb{E}[M_I] \geq \mathbb{E}_{\bar{\mu}_s}[q]$.

Proof. The proposals are independent, and a maximum of independent variables is at most t exactly when each of them is, so $\mathbb{P}(M_S \leq t) = \prod_z G_z(t)^{n_z}$. Each i.i.d. proposal from $\bar{\mu}_s$ has CDF $\bar{G}(t) = \sum_z w_z G_z(t)$, so $\mathbb{P}(M_I \leq t) = \bar{G}(t)^N$. For nonnegative g_z and weights w_z summing to one, the weighted AM–GM inequality gives $\prod_z g_z^{w_z} \leq \sum_z w_z g_z$; applying it with $g_z = G_z(t)$ and raising both sides to the power N, using $N w_z = n_z$, gives Equation (36).

Let q take values in $[c, C]$. For such a variable, $\mathbb{E}M = c + \int_c^C \mathbb{P}(M > t)\, dt$, so the pointwise ordering of the CDFs gives $\mathbb{E}M_S \geq \mathbb{E}M_I$. Finally $M_I \geq q(A_1)$ with $A_1 \sim \bar{\mu}_s$, whence $\mathbb{E}M_I \geq \mathbb{E}_{\bar{\mu}_s} q$. □

Proposition 1 (Policy improvement with mode-conditioned proposals)**.** *Consider the macro-MDP with discount $\beta = \gamma^H$. Let the baseline policy π sample $z \sim w$ and then $A \sim \mu_{s,z}$, so that it induces the mixture proposal $\bar{\mu}_s$, and let π_C select among the stratified proposals according to Equation* (13)*. Assume the critic-error bound of Theorem 2 holds with a state-dependent constant $\varepsilon_Q(s)$, and define the coverage gain*

$$g_S(s) = \mathbb{E}\Big[\max_{i \le N} Q^\pi(s, A_i)\Big] - V^\pi(s). \tag{37}$$

Then $g_S(s) \ge g_I(s) \ge 0$, where g_I is the i.i.d. best-of-N gain obtained under the same mixture, and

$$J(\pi_C) - J(\pi) \;\ge\; \frac{1}{1-\beta}\,\mathbb{E}_{s\sim d^{\pi_C}}\big[g_S(s) - 2\varepsilon_Q(s)\big]. \tag{38}$$

Proof. Fix s and a realization of the candidates $\{(A_i, z_i)\}_{i \le N}$. Let $\hat{\imath}$ maximize $\bar{Q}_\phi(s, A_i, z_i)$ and let $i^\dagger$ maximize $Q^\pi(s, A_i)$. Applying the error bound at $\hat{\imath}$, the optimality of $\hat{\imath}$ for $\bar{Q}_\phi$, and the error bound at $i^\dagger$,

$$Q^\pi(s, A_{\hat{\imath}}) \;\ge\; \bar{Q}_\phi(s, A_{\hat{\imath}}, z_{\hat{\imath}}) - \varepsilon_Q(s) \;\ge\; \bar{Q}_\phi(s, A_{i^\dagger}, z_{i^\dagger}) - \varepsilon_Q(s) \;\ge\; Q^\pi(s, A_{i^\dagger}) - 2\varepsilon_Q(s). \tag{39}$$

Taking expectations over the candidates gives $\mathbb{E}_{A\sim\pi_C(\cdot|s)} Q^\pi(s, A) \ge \mathbb{E}\max_i Q^\pi(s, A_i) - 2\varepsilon_Q(s)$. Subtracting $V^\pi(s)$ leaves an expected advantage of at least $g_S(s) - 2\varepsilon_Q(s)$, and Lemma 2 gives Equation (38).

For the properties of g_S, rewards are bounded, so $q = Q^\pi(s, \cdot)$ is bounded. Because π samples $z \sim w$ and then $A \sim \mu_{s,z}$, we have $V^\pi(s) = \mathbb{E}_{A\sim\bar{\mu}_s} Q^\pi(s, A)$, and Lemma 3 gives

$$\mathbb{E}\Big[\max_{i \le N} Q^\pi(s, A_i)\Big] = \mathbb{E}[M_S] \;\ge\; \mathbb{E}[M_I] \;\ge\; V^\pi(s), \tag{40}$$

that is, $g_S(s) \ge g_I(s) \ge 0$. The same CDF argument as in Appendix A.4 of Zhan et al. (2026) shows that g_I is nondecreasing in N and vanishes at $N = 1$. □

Remark 2 (Scope of the analysis). The results above concern the *proposal* distribution and a critic that approximates the value of a reference policy, as in Zhan et al. (2026). They do not describe the joint training dynamics of the actor and the critic. Theorem 1(iv) shows that zero bridge mass is *achievable* by conditional maps; it does not show that K-means labels recover the true regions. We therefore evaluate these assumptions empirically, in the toy study of Figure 1, where the bridge mass can be measured directly, and through the resolution ablation of Figure 5.

B ALGORITHM

Algorithm 1 MODEFLOW: mode-conditioned MeanFlow RL

Require: offline trajectories $\mathcal{D}$; number of modes K; proposal budget N with $K \mid N$; chunk length H; IVC weight λ; Polyak rate ρ; offline and online steps $T_{\text{off}}, T_{\text{on}}$; warm-up T_{start}

Phase 1: trajectory mode discovery (once)

1: Compute $\psi(\tau)$ (Equation (10)) for every $\tau \in \mathcal{D}$ and standardize the descriptors
2: Fit mini-batch k-means with K centers on up to 3,000 trajectories; assign each τ the label $z(\tau)$ of its nearest center
3: Give every transition of τ the label $z(\tau)$
4: Initialize u_θ and Q_ϕ, each with a label embedding $e(z)$; set $\phi^- \leftarrow \phi$

Phase 2: offline pre-training

5: **for** $k = 1, \dots, T_{\text{off}}$ **do**
6: Sample H-step windows (s, A, z, R_H, s', m) from $\mathcal{D}$; z is the label of the window's first transition
7: $z_i \leftarrow i \bmod K$, $\epsilon_i \sim \mathcal{N}(0, I)$, $A_i \leftarrow F_{s',z_i}(\epsilon_i)$ for $i = 1, \dots, N$ ▷ stratified proposals, one batched call
8: $(A'^\star, z'^\star) \leftarrow (A_{i^\star}, z_{i^\star})$ with $i^\star = \arg\max_i \bar{Q}_\phi(s', A_i, z_i)$ ▷ Equation (13)
9: Form $\mathcal{L}_Q(\phi)$ with target $R_H + \beta\, m\, \bar{Q}_{\phi^-}(s', A'^\star, z'^\star)$ ▷ Equation (14)
10: Sample $0 \le r \le t \le 1$ and $\epsilon \sim \mathcal{N}(0, I)$; form $\mathcal{L}_\pi(\theta)$ on (s, A, z) ▷ Equation (11)
11: Take one gradient step on $\mathcal{L}_\pi(\theta) + \mathcal{L}_Q(\phi)$; $\phi^- \leftarrow \rho\, \phi + (1 - \rho)\, \phi^-$
12: **end for**

Phase 3: online fine-tuning

13: $\mathcal{B} \leftarrow \mathcal{D}$; reset the environment to s
14: **for** $k = 1, \dots, T_{\text{on}}$ **do**
15: **if** no actions of the current chunk remain **then**
16: $(A^\star, z^\star) \leftarrow$ stratified selection at s (lines 7–8) ▷ new chunk and mode
17: **end if**
18: Execute the next action a of $A^\star$; observe reward and next state s'
19: $\mathcal{B} \leftarrow \mathcal{B} \cup \{(s, a, \text{reward}, s', z^\star)\}$ ▷ online data carry the selected label
20: $s \leftarrow s'$; if the episode ends, reset s and discard the rest of $A^\star$
21: **if** $k \ge T_{\text{start}}$ **then**
22: One update as in lines 6–11, with windows sampled from $\mathcal{B}$
23: **end if**
24: **end for**

Evaluation executes the chunks chosen by the stratified selection. Relative to MVP, the only changes are the label assignment in Phase 1, the label embedding $e(z)$ in both networks, and the deterministic stratification of the proposals. With $K = 1$ the algorithm reduces to MVP.

C Implementation and Experimental Details

Code base. ModeFlow is implemented in JAX on top of the public MVP code of Zhan et al. (2026). The mode-conditioned actor and critic differ from MVP only in their input layer. The concatenated input (s, noisy chunk, t, r for the actor; s, chunk for the critic) is projected by a dense layer to 512 units and concatenated with a 512-dimensional embedding of z. The remaining MLP layers are those of MVP.

Table 4: **Hyperparameters** of ModeFlow.

Parameter	Value	Parameter	Value
Offline / online steps	1M / 1M	Actor network	4×512 MLP, GELU, no LN
Batch size	256	Critic network	4×512 MLP, GELU, LN
Optimizer, learning rate	Adam, $3 \cdot 10^{-4}$	Critic ensemble / aggregation	5 heads / mean
Discount γ	0.99	Target update rate τ	0.005
Chunk length H	5	UTD ratio	1
IVC weight λ	1.0	Online updates start	after 5,000 steps
Time sampling	$\mathrm{sigmoid}(\mathcal{N}(-0.4, 1)), t \geq r$	Evaluation	50 episodes / 50k steps
Proposals N	32	Modes K	32(OGBench)/2(Robomimic)
Mode embedding size	512	Label assignment	$z_i = i \bmod K$
Feature time points B	50	K-means fit set	3,000 trajectories, seed 0

Differences from the published MVP protocol. We use the configuration of the public MVP code. It differs from Table 6 of Zhan et al. (2026) in two respects: the critic has five heads instead of two, and Robomimic uses $N = 32$ proposals instead of 16. The controlled comparison in Figure 4 (MVP and ModeFlow with identical five-head critics and $N = 32$) shows that these settings alone do not explain the gains: with them, MVP still scores 0% on Cube-triple-task5.

Clustering. Trajectory boundaries come from the dataset terminals (OGBench) or the demonstration boundaries of the HDF5 file (Robomimic). All tasks of one OGBench domain share the same play dataset, so they share the same partition. The labels are therefore task-agnostic and computed without rewards. Table 5 summarizes the partitions. The effective number of modes is $\exp$ of the entropy of the cluster-size distribution.

Table 5: **Mode statistics** (number of trajectories per mode) of the partitions used in the experiments.

Dataset	Traj.	K	Min	Median	Max	Eff. modes
Robomimic lift	300	2	143	150	157	2.0
Robomimic can	300	2	21	150	279	1.3
Robomimic square	300	2	98	150	202	1.9
OGBench cube-double-play	1,000	32	1	34.5	68	26.3
OGBench cube-triple-play	3,000	32	6	103.5	155	28.9
OGBench cube-quadruple-play	5,000	32	1	176–179	296–308	24.6–24.9
OGBench antmaze-giant-navigate	5,000	32	5	110	696	19.1

Environments. *Robomimic* (Mandlekar et al., 2021): `lift` (pick up a cube), `can` (pick a can and place it in a bin), and `square` (insert a square nut on a peg). We use the low-dimensional multi-human datasets, with 50 demonstrations from each of six operators of varying skill. Rewards are -1 until success. *OGBench* (Park et al., 2025a): `cube-double`, `cube-triple`, and `cube-quadruple` require moving two, three, or four cubes to target positions. Tasks 1–5 differ in the initial and goal configurations: single moves, cyclic swaps, and unstacking. The reward is minus the number of misplaced cubes, and the episode ends on success. `antmaze-giant-navigate` requires a quadruped to reach a distant goal in a large maze. The datasets are the default `play` (manipulation) and `navigate` (maze) datasets.

Local baselines on task 5. FQL, BFN, and QC were run with the QC code base of Li et al. (2026) using its default hyperparameters: 10 flow steps, two critic heads, and BC coefficient $\alpha = 300$ (FQL) or 100 (BFN, QC). The chunk length is $H = 1$ for FQL and BFN and $H = 5$ for QC, and the recorded numbers of samples are 32 (FQL, unused), 4 (BFN), and 32 (QC). The local MVP baseline uses exactly the ModeFlow configuration without labels.

Compute. Each run (1M offline + 1M online steps) was trained on a single NVIDIA GPU. Per-step sampling cost equals that of MVP at the same N: one batched generator evaluation and one batched critic evaluation of N candidates.

D TOY STUDY OF THE CONNECTIVITY BARRIER

The toy study isolates the generator. The action space is R^2, and the data are an equal mixture of four isotropic Gaussians with standard deviation 0.12, centered at $(\pm d, \pm d)$. Each high-value region is the disk of radius 3×0.12 around a center. We train the MVP objective, MeanFlow plus IVC with the adaptive loss, the same (t, r) sampling, and the same data-at-t=0 convention as the RL code, with a 4-layer GELU MLP of width 256, Adam ($3 \cdot 10^{-4}$), batch size 1,024, and 15,000 steps. The conditional model concatenates a 256-dimensional label embedding with the projected input, exactly as in MODEFLOW. Labels come from k-means (k-means++ initialization) with $K \in \{4, 8\}$, and sampling uses stratified labels. We draw 20,000 one-step samples and report the fraction outside all regions. For best-of-N, the top-right region has value 1, the other regions 0.5, and the complement 0, and we report the probability that the maximum-value proposal lies in the top-right region. We use three seeds per configuration. The off-support mass of the single map is 1.7 ± 0.3, 3.6 ± 0.2, 6.5 ± 0.8, 7.7 ± 1.5, and $12.1 \pm 1.8\%$ for $d = 0.5, 1, 1.5, 2, 3$, respectively, while both conditional models stay below 0.05% at every separation. At $d = 2$, the single map returns the best region with probability 0.23, 0.41, 0.65, 0.87, 0.98, and 1.00 for $N = 1, \ldots, 32$, whereas stratified MODEFLOW does so in every sampled trial for all $N \geq K$. The toy study measures the generator alone, so its scale is not comparable to the RL tasks. It shows that the trend predicted by Theorem 1 appears when MVP's actual training objective is used.