Understanding Multimodality in Generative Behavioral Cloning

Lorenzo Mazza[1]
lorenzo.mazza@nct-dresden.de

Massimiliano Datres[2]
datres@math.lmu.de

Ariel Rodriguez[1,3]

Sebastian Bodenstedt[1,4]

Gitta Kutyniok[2,5,6]

Stefanie Speidel[1,3,4,7]

Abstract

Behavioral cloning becomes challenging when the same observation admits several valid actions. We study how generative behavioral-cloning policies represent such multimodal expert behavior and identify different bottlenecks across model parameterizations. For latent-variable policies, preserving demonstrated modes requires action-conditioned information in the latent representation. Excessive posterior–prior regularization can suppress this information and prevent the policy from distinguishing demonstrated modes. Weaker or aggregate regularization can preserve mode information, but shifts the challenge to ensuring that the deployment-time prior covers the relevant latent regions. For action-space generative policies, multimodality is constrained by the smoothness of the base-to-action transport: a map with a small Lipschitz constant cannot assign substantial probability to many well-separated modes. Covering many modes therefore requires either sharp transitions in base space or off-support bridge regions in action space. Experiments on synthetic multimodal navigation and a physical-robot bimodal manipulation task support these mechanisms. In contrast, our analysis reveals limited conditional multimodality in standard robotic simulation benchmarks, where deterministic regression remains competitive.

1 Introduction

Imitation learning trains a robot policy directly from expert demonstrations [24, 28, 12]. A standard approach is behavioral cloning (BC), which fits a policy to observation–action pairs collected under an expert policy, without the need for online interaction, reward engineering, or exploration. Due to its practical appeal, BC has become a dominant paradigm for training deep visuomotor policies in robotic manipulation [44, 7, 38, 16, 2]. Recent work on behavioral cloning has highlighted multimodality in the expert's conditional action distribution as a central challenge [10, 30, 7, 19, 31]: for the same observation, the dataset may contain multiple valid actions. This may arise from different expert styles, equally good choices, or latent factors absent from the observation space [20]. Under an L_2 or unimodal-Gaussian behavioral-cloning head, the maximum-likelihood solution regresses to the conditional *mean* of the modes, which is typically not an element of any mode's support [42, 10]. A canonical example is inverse kinematics in robot control[1]: the same end-effector target may admit disconnected joint-space solutions. Averaging can yield a joint command that lies on none of the inverse-kinematics branches, while filtering to one branch removes feasible alternatives needed under constraints such as obstacles or joint limits.

Contemporary multimodal BC methods typically predict short chunks of future actions rather than single-step actions, improving temporal coherence while enabling one-shot non-autoregressive

[1]NCT/UCC Dresden, UKDD Dresden, TU Dresden, DKFZ Heidelberg. [2]Ludwig-Maximilians-Universität München, Munich Center for Machine Learning (MCML). [3]BMFTR Research Hub 6G-Life. [4]Cluster of Excellence CeTI. [5]University of Tromsø. [6]German Aerospace Center (DLR). [7]HZDR Dresden.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

inference over the chunk [44, 7]. There are two broad approaches to making such policies multimodal: *latent-variable methods* and *action-space generative methods*. Latent-variable methods introduce an unobserved code, continuous or discrete, from which different action chunks can be decoded [44, 19, 41]. Action-space generative methods model the distribution over action chunks directly, using flows or diffusion models [7, 2, 23]. Both approaches can generate diverse actions, and their multimodality is typically evaluated by rollout performance.

In both approaches, however, it remains unclear which model- and training-level quantities control the multimodality of the learned policy. Our goal is to fill this gap by providing a precise definition of multimodality. Furthermore, we identify the key quantities that must be controlled to preserve multimodality: the posterior–prior regularization strength in latent-variable policies, and the Lipschitz constant of the base-to-action map in action-space generators.

Contribution. We study how posterior-prior regularization affects multimodality in action-chunking behavioral cloning. In particular, we summarize our contributions as follows:

- For latent-variable policies, we show that preserving demonstrated modes requires action-conditioned latent information; see Proposition 1. We then show that pointwise posterior–prior regularization can suppress this information as its strength increases, giving the multimodality collapse criterion in Corollary 2.
- For action-space generators, we show that a deterministic sampler with a small Lipschitz constant is more likely to assign a small probability to many modes; see Proposition 2, losing multimodality. Therefore, regularization methods that enforce a small Lipschitz constant tend to suppress multimodality.
- We empirically validate these mechanisms on controlled multimodal navigation and manipulation tasks and provide practical heuristics. We also introduce a diagnostic for conditional multimodality in robot demonstration datasets. Applied to standard robotic simulation benchmarks, it detects limited conditional multimodality, while our policy evaluations show that deterministic regression remains competitive. We release the code used in the experiments[1].

Structure of the paper. Section 2 defines multimodal behavioral cloning over actions and the generative policy families considered in the paper. Section 4 formalizes mode preservation and proves that latent policies must carry action-conditioned mode information, leading to a collapse criterion for pointwise prior regularization, a latent-geometry condition for aggregate matching, and a Lipschitz lower bound for action-space generators. Section 5 introduces controlled synthetic benchmarks with ground-truth mode labels and evaluates representative methods on synthetic and robotic tasks to test whether these mechanisms explain practical failures. Notation, proofs, related work, and experimental details are described in the Appendices.

2 Background

For any notation that may be unclear, we refer the reader to Appendix A.

Offline imitation learning. We consider a finite-horizon Markov decision process $(\mathcal{S}, \mathcal{A}, P, r, \rho_0, \mathsf{T})$, where $\mathcal{S}$ is the state or observation space, $(\mathcal{A}, d_{\mathcal{A}})$ is the (metric) action space, $P(\cdot \mid s, a) \in \mathcal{P}(\mathcal{S})$ is the transition kernel, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward, $\rho_0 \in \mathcal{P}(\mathcal{S})$ is the initial-state distribution, and $\mathsf{T} \in \mathbb{N}$ is the horizon. A stochastic policy $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ maps each state s to a probability distribution $\pi(\cdot \mid s)$ over actions. In the offline-imitation-learning setting, the reward signal r is not observed. Instead, we are given demonstrations generated by an expert policy π_E. A trajectory realization is $\tau = (s_0, a_0, \dots, (s_{\mathsf{T}-1}, a_{\mathsf{T}-1}), s_{\mathsf{T}}) \in \mathcal{T}_{\mathsf{T}} := (\mathcal{S} \times \mathcal{A})^{\mathsf{T}} \times \mathcal{S}$. The expert policy, together with P and ρ_0, induces a trajectory law $p_E \in \mathcal{P}(\mathcal{T}_{\mathsf{T}})$. When the relevant densities or probability mass functions exist, this law factorizes as $p_E(\tau) = \rho_0(s_0) \prod_{t=0}^{\mathsf{T}-1} \pi_E(a_t \mid s_t)\, P(s_{t+1} \mid s_t, a_t)$. We denote by $p_{\mathcal{D}} \in \mathcal{P}(\mathcal{S} \times \mathcal{A})$ the time-marginal expert state-action distribution:

$$p_{\mathcal{D}}(s, a) := \frac{1}{\mathsf{T}} \sum_{t=0}^{\mathsf{T}-1} p_E(S_t = s, A_t = a),$$

[1] https://github.com/Lorenzo-Mazza/VersatIL

where $p_E(S_t = s, A_t = a)$ denotes the t-step state-action marginal induced by the trajectory law p_E. The goal of offline imitation learning is to learn a policy $\pi_\theta(\cdot \mid s)$ from samples drawn from this expert state-action law described by $p_{\mathcal{D}}(s, a)$. Let $T_1, \ldots, T_N \overset{\text{iid}}{\sim} p_E$ be independent expert trajectory random variables. The dataset of expert demonstrations is one realization $\tilde{\mathcal{D}} = \{\tau_i\}_{i=1}^N = \left\{(s_0^i, a_0^i, \ldots, s_{\mathsf{T}-1}^i, a_{\mathsf{T}-1}^i, s_{\mathsf{T}}^i)\right\}_{i=1}^N$. Using $\tilde{\mathcal{D}}$, we construct the empirical state-action distribution

$$p_{\tilde{\mathcal{D}}}(s, a) := \frac{1}{N\mathsf{T}} \sum_{i=1}^{N} \sum_{t=0}^{\mathsf{T}-1} \delta_{(s_t^i, a_t^i)}(s, a)\,.$$

Remark 1. In practice, action-chunk BC heads $a_{t:t+\mathsf{H}-1} \in \mathcal{A}^{\mathsf{H}}$ are commonly used. Action-chunk BC is executed open-loop between observations, to retain temporal dependence absent from $p_{\mathcal{D}}$ and reduce rollout error [44, 33]. All definitions and results extend by replacing $\mathcal{A}$ with $\mathcal{A}^{\mathsf{H}}$, so samples become $(s_t^i, a_{t:t+\mathsf{H}-1}^i)$.

Multimodal imitation learning. We formalize *multimodality* as a property of $p_{\mathcal{D}}(\cdot \mid s)$. For each observation $s \in \mathcal{S}$, the support of the expert conditional action distribution is contained in finitely many well-separated valid mode sets. In other words, for each $s \in \mathcal{S}$, there exists a natural number $\mathsf{K}(s) \geq 1$ and a family of covering sets $C_1(s), \ldots, C_{\mathsf{K}(s)}(s) \subseteq \mathcal{A}$ of $\operatorname{supp} p_{\mathcal{D}}(\cdot \mid s)$ such that

$$\operatorname{supp} p_{\mathcal{D}}(\cdot \mid s) \subseteq \bigcup_{k=1}^{\mathsf{K}(s)} C_k(s), \qquad \Delta(s) := \min_{k \neq k'} \operatorname{dist}_{\mathcal{A}}\big(C_k(s), C_{k'}(s)\big) > 0. \tag{1}$$

Each set $C_k(s)$ represents one valid strategy or trajectory mode for observation s. The mode sets have disjoint closures. We further define the *mode-assignment* function $g : \mathcal{S} \times \mathcal{A} \to \mathbb{N}_0$ as

$$g(s, a) = \sum_{k=1}^{\mathsf{K}(s)} k\, \mathbf{1}_{\{a \in C_k(s)\}}, \qquad \mathbf{1}_{\{a \in C_k(s)\}} = \begin{cases} 1, & a \in C_k(s), \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The function g is a mode-level labeling of expert actions used for the analysis. For each realized pair (s, a), $g(s, a)$ is the deterministic index of the mode containing a. Under the data distribution $(S, A) \sim p_{\mathcal{D}}$, we denote the induced mode-label random variable by $M := g(S, A)$. Here $\mathsf{K}(s) = 1$ gives the unimodal case, while $\mathsf{K}(s) \geq 2$ is the setting of interest.

Behavioral cloning. Behavioral cloning instantiates offline imitation learning as supervised learning on state-action pairs extracted from expert trajectories. Let $\pi_\theta(a \mid s)$ be the conditional action distribution given a state s and parametrized by θ. In practice, π_θ is represented by a neural network parametrized by θ, which maps each state s to a distribution over actions. The parameters θ are usually determined by minimizing the standard maximum-likelihood objective

$$\mathcal{L}_{\mathrm{BC}}(\theta) = -\mathbb{E}_{(S,A)\sim p_{\mathcal{D}}}[\log \pi_\theta(A \mid S)]\,, \tag{3}$$

which, in the infinite-data limit, is equivalent [14] up to an additive constant to minimizing the Kullback-Leibler (KL) divergence between the expert data and the model distribution $\mathbb{E}_{S\sim p_{\mathcal{D},S}} \mathrm{KL}(p_{\mathcal{D}}(\cdot \mid S) \,\|\, \pi_\theta(\cdot \mid S))$, where $p_{\mathcal{D},S}$ denotes the state marginal of $p_{\mathcal{D}}$.

Conditional latent-variable policies. Latent-variable BC policies introduce an auxiliary random variable $Z \in \mathbb{R}^{\mathsf{d}_z}$ to represent action ambiguity not resolved by the observation. They use a conditional variational autoencoder (CVAE) architecture [34], with a posterior encoder $q_\phi(\cdot \mid a, s)$, a conditional prior $p_\psi(\cdot \mid s)$, and a latent-conditioned decoder policy $\pi_\theta(\cdot \mid z, s)$. At deployment, $Z \sim p_\psi(\cdot \mid s),\ A \sim \pi_\theta(\cdot \mid Z, s)$. We write their training objective in the generic regularized form

$$\mathcal{L}_{\mathrm{var}}^{\beta}(\theta, \phi, \psi) = \mathbb{E}_{(S,A)\sim p_{\mathcal{D}}}\left[-\mathbb{E}_{Z\sim q_\phi(\cdot|A,S)} \log \pi_\theta(A \mid Z, S)\right] + \beta D(q_\phi, p_\psi), \tag{4}$$

where $D(q_\phi, p_\psi)$ couples the posterior encoder to the deployment prior and $\beta > 0$ controls its strength.

Action-space generative policies. Action-space generative policies sample actions directly by transforming base noise into actions. For fixed s, we write the full deterministic sampler as $G_{\theta,s}(u) := G_\theta(s, u),\ U \sim p_0,\ A = G_{\theta,s}(U)$. The induced conditional policy is therefore the push-forward $\pi_\theta(\cdot \mid s) = (G_{\theta,s})_{\#} p_0$. This abstraction covers deterministic samplers obtained from diffusion or flow-matching policies.

3 Related Work

Multimodal BC has been modeled through several policy families. ACT uses a KL-regularized CVAE with an explicit latent variable to represent trajectory style [44]. VQ-BeT represents multimodality through learned discrete latent tokens and a two-stage training recipe [19, 30]. Q-FAT uses continuous Gaussian-mixture heads in an autoregressive transformer [31], which is structurally slower than parallel action-chunk decoding. Implicit BC parameterizes the policy as a conditional energy function, requiring sampling or gradient-based optimization at inference rather than a single decoder pass [10]. More recent methods shift multimodality directly into action-space denoising or flow matching [7, 26, 25, 2, 23, 32]. Across these approaches, comparisons remain largely task-driven and empirical, with limited theoretical guidance on which training quantities affect policies' ability to represent multimodality.

For one-shot push-forward generators $A = G_\theta(s, U)$, existing results show that separated multimodal targets force a trade-off between approximation error and generator smoothness [29]. For iterative samplers such as DDIM [35, 7] or deterministic flow solvers [2, 23, 32], we study this trade-off for the composed deterministic sampling map induced by the denoising or flow updates. A parallel gap appears in latent-based policies: pointwise KL regularization is standard, but its weight is usually treated as an empirical hyperparameter.

4 Main Results

We analyze how multimodality appears in the action-chunking generative policy families. For latent-variable policies, preserving mode identity requires action information in the latent variable. For action-space generative policies, the constraint is geometric: separated modes must be realized by the sampling map from base noise to action chunks. The reason for these two different analyses is that, for CVAEs, the action information is a natural quantity for studying multimodality, whereas for generative policies it is meaningless, since there is no information bottleneck.

Multimodality in latent-variable policies. Latent-variable policies route multimodal action ambiguity through an explicit latent random variable Z. We first quantify the information required to preserve the demonstrated mode. Let

$$q_{\phi,\mathcal{D}}(s, a, z) := p_{\mathcal{D}}(s, a)\, q_\phi(z \mid a, s) = p_{\mathcal{D},S}(s)\, p_{\mathcal{D}}(a \mid s)\, q_\phi(z \mid a, s)$$

denote the posterior-encoder-induced joint distribution over (S, A, Z). Unless stated otherwise, all probabilities, mutual informations, expectations, and conditional distributions involving (S, A, Z) in this subsection are taken under $q_{\phi,\mathcal{D}}$. We write the state-conditioned aggregated posterior as $q_{\phi,\mathcal{D}}(z \mid s) := \mathbb{E}_{A\sim p_{\mathcal{D}}(a|s)}[q_\phi(z \mid A, s)]$. The conditional mutual information between A and Z given S is denoted by $I(A; Z \mid S)$. From [13], it holds

$$\mathbb{E}_{(S,A)\sim p_{\mathcal{D}}}\mathrm{KL}(q_\phi(\cdot \mid A, S) \,\|\, p_\psi(\cdot \mid S)) = I(A; Z \mid S) + \mathbb{E}_{S\sim p_{\mathcal{D},S}}\mathrm{KL}(q_{\phi,\mathcal{D}}(\cdot \mid S) \,\|\, p_\psi(\cdot \mid S))\,.$$

The first term measures how much information Z must encode about the demonstrated action A beyond what is already determined by the state S, while the second term measures aggregated posterior–prior mismatch. Thus $I(A; Z \mid S)$ captures the action information available to preserve the demonstrated mode: if S already determines A uniquely, then Z need not contain additional information about A.

In the rest of the analysis, we assume that the latent-conditioned decoder policy $\pi_\theta(\cdot \mid z, s)$ is concentrated on the graph of a neural network f_θ, that is $\pi_\theta(\cdot \mid z, s) = \delta_{f_\theta(z,s)}$, with decoded action $a = f_\theta(z, s)$. Therefore, given a sample $(s, a) \in \mathrm{supp}(p_{\tilde{\mathcal{D}}})$, the training-time prediction law is $p_{\theta,\phi}(\cdot \mid s, a) := (f_{\theta,s})_\# q_\phi(\cdot \mid a, s)$, where $f_{\theta,s}(z) := f_\theta(z, s)$. A realized action prediction $\hat{a} \sim p_{\theta,\phi}(\cdot \mid s, a)$ is correct if and only if $\hat{a} \in C_{g(s,a)}(s)$.

For a measurable function $\rho : \mathcal{S} \to [0, 1]$ representing a state-dependent mode-identification error, we define the Fano-corrected lower bound as

$$B_\rho := \mathbb{E}_{S\sim p_{\mathcal{D},S}}[H(M \mid S)] - \mathbb{E}_{S\sim p_{\mathcal{D},S}}\Big[H_b(\rho(S)) + \rho(S)\log(\mathsf{K}(S) - 1)\Big],$$

where $H_b(p) = -p\log p - (1-p)\log(1-p)$ is the entropy of a $\mathrm{Bernoulli}(p)$ random variable, and the bracketed term is taken to be zero when $\mathsf{K}(S) = 1$. The quantity B_ρ measures the portion

of the expected conditional mode entropy $\mathbb{E}_{S\sim p_{\mathcal{D},S}}[H(M \mid S)]$ that remains after applying a Fano correction for $\rho(S)$. A positive value indicates that, under the data distribution, the mode M cannot be resolved from S alone up to the specified error $\rho(S)$, providing a certificate of unresolved conditional multimodality.

The next result lower bounds $I(A; Z \mid S)$ by the Fano-corrected entropy of the demonstrated mode label M.

Proposition 1 (Mode-information lower bound). *Under the setting introduced above, for each $s \in \mathcal{S}$, let*

$$\rho(s) := \Pr(f_\theta(Z, s) \notin \mathcal{C}_M(s) \,|\, S = s)\,. \tag{5}$$

For all $s \in \operatorname{supp}(p_{\mathcal{D},S})$ with $\mathsf{K}(s) > 1$, assume $\rho(s) \le 1 - 1/\mathsf{K}(s)$. Then $I(A; Z \mid S) \ge B_\rho$. Consequently, it holds

$$\mathbb{E}_{(S,A)\sim p_{\mathcal{D}}} \mathrm{KL}(q_\phi(\cdot \mid A, S) \,\|\, p_\psi(\cdot \mid S)) \ge B_\rho\,.$$

The detailed proof of Proposition 1 is given in Appendix B. We note that the assumption of Proposition 1 ensures that, for all $s \in \operatorname{supp}(p_{\mathcal{D},S})$ with $\mathsf{K}(s) > 1$, the induced mode-recovery rule is nontrivial: its error probability $\rho(s)$ is no larger than the error rate $1 - 1/\mathsf{K}(s)$ of uniform random guessing among the $\mathsf{K}(s)$ admissible modes. The result shows that accurate mode recovery forces the posterior latent to retain action-conditioned information: the better the decoder preserves the demonstrated mode, the larger the lower bound B_ρ.

Corollary 1 (Uniform exact-recovery case). *Under the setting of Proposition 1, if $\rho(s) = 0$ for $p_{\mathcal{D},S}$-almost every s, and $M \mid S = s$ is uniform over $\mathsf{K}(s)$ modes, then*

$$I(A; Z \mid S) \ge \mathbb{E}_{S\sim p_{\mathcal{D},S}} \log \mathsf{K}(S).$$

Corollary 1 isolates the maximum-ambiguity case: exact recovery of $\mathsf{K}(s)$ equiprobable modes requires the latent to carry on average the cost of the full mode entropy, $\log \mathsf{K}(s)$ nats.

We recall that CVAEs are usually trained using a regularized objective $\mathcal{L}^\beta_{\mathrm{var}}(\theta, \phi, \psi)$ defined in (4). A widely used regularizer [44] is given by

$$D(q_\phi, p_\psi) = \mathbb{E}_{(S,A)\sim p_{\mathcal{D}}} \mathrm{KL}(q_\phi(\cdot \mid A, S) \,\|\, p_\psi(\cdot \mid S))\,. \tag{6}$$

Corollary 2. *Let us fix $\beta > 0, \mathsf{b}_0 > 0$. Let us denote with*

$$(\hat\theta, \hat\phi, \hat\psi) := \arg\min_{\theta,\phi,\psi} \mathcal{L}^\beta_{\mathrm{var}}(\theta, \phi, \psi), \qquad \hat{\mathcal{L}}^\beta := \mathcal{L}^\beta_{\mathrm{var}}(\hat\theta, \hat\phi, \hat\psi)$$

the solution of the learning problem and the achieved training loss, respectively. Let us assume that:

- *(H1) there exists $(\theta_0, \phi_0, \psi_0)$ such that $D(q_{\phi_0}, p_{\psi_0}) = 0$, and denote by $\mathcal{L}^0 = \mathcal{L}^0_{\mathrm{var}}(\theta_0, \phi_0, \psi_0)$ the corresponding training loss;*
- *(H2) the mode-recovery error $\hat\rho(s) := \Pr\big(f_{\hat\theta}(Z, s) \notin C_M(s) \,\big|\, S = s\big)$ satisfies $\hat\rho(s) \le 1 - 1/\mathsf{K}(s)$ for all $s \in \operatorname{supp}(p_{\mathcal{D},S})$ with $\mathsf{K}(s) > 1$.*

Then there exists a constant $\mathsf{C} > 0$, independent of β, such that $B_{\hat\rho} \le I(A; Z \mid S) \le \frac{\mathsf{C}}{\beta}$. Consequently, a necessary condition for achieving a certificate $B_{\hat\rho} > \mathsf{b}_0$ is $\beta < \frac{\mathsf{C}}{\mathsf{b}_0}$.

Remark 2. We note that the Fano term is what turns the information bound into a mode-recovery statement: $I(A; Z \mid S)$ may encode any action-dependent variation, whereas B_ρ is positive only when unresolved mode entropy $H(M \mid S)$ is recovered with sufficiently small error ρ.

Assumption *(H1)* requires q_{ϕ_0} and p_{ψ_0} to coincide for some parameter values (ϕ_0, ψ_0), so that $D(q_{\phi_0}, p_{\psi_0}) = 0$. For the pointwise KL regularizer, this means that $q_{\phi_0}(\cdot \mid a, s) = p_{\psi_0}(\cdot \mid s)$ on the data support, and therefore Z does not encode additional information about A beyond S. This collapse behavior is observed in practice; see, for instance, [3, 6]. Assumption *(H2)* ensures that the learned mode-recovery rule is nontrivial, since its error probability is no larger than that of uniform random guessing among the $\mathsf{K}(s)$ admissible modes.

Corollary 2 shows that the strength of the variational regularization directly limits the amount of certified conditional multimodality that the learned latent representation can retain. In particular, as β

increases, the conditional mutual information $I(A; Z \mid S)$ is forced to decrease at rate $1/\beta$. Since $B_{\hat{\rho}}$ is upper-bounded by this information term, strong regularization can prevent the latent variable from encoding the action-dependent mode information needed to represent multiple plausible actions at the same state. Therefore, obtaining a nontrivial multimodality certificate $B_{\hat{\rho}} > \mathsf{b}_0$ requires the regularization strength to be sufficiently small, namely $\beta < \mathsf{C}/\mathsf{b}_0$.

Remark 3. Corollary 2 is specific to regularizers of the form (6). For other regularization strategies, an analogous relation between the regularization strength β and the certified multimodality does not follow in general. Aggregate matching objectives provide a counterexample to this behavior; see Appendix B.2 for details.

Multimodality in action-space generative policies. Action-space generative policies remove the explicit posterior latent, but not the geometric problem of routing samples to separated action modes. In this paragraph, we investigate when multimodality is preserved by such generative policies.

We consider a generative policy defined as follows. Let us fix $s \in \mathcal{S}$, $d \geq 1$ a positive integer and let $p_0 := \mathcal{N}(0, I_d)$, $U \sim p_0$, where $I_d \in \mathbb{R}^{d \times d}$ denotes the identity matrix. The transport map, or deterministic sampler, at state s with parameters θ is defined as $G_{\theta,s}(u) := G_\theta(s, u)$. The induced conditional policy is the push-forward of the Gaussian base distribution $\pi_\theta(\cdot \mid s) = (G_{\theta,s})_{\#} p_0 = (G_{\theta,s})_{\#} \mathcal{N}(0, I_d)$. In other words, for any measurable set $B \subseteq \mathcal{A}$, $\pi_\theta(\cdot \mid s)$ is defined by

$$\pi_\theta(B \mid s) = \Pr_{U \sim \mathcal{N}(0, I_d)} (G_\theta(s, U) \in B) = \int_{\{u:\, G_\theta(s,u) \in B\}} \frac{1}{(2\pi)^{d/2}} e^{-\|u\|_2^2/2}\, du.$$

In practice, G_θ is parameterized by a neural network and trained to transform Gaussian samples into actions whose conditional distribution approximates $p_{\mathcal{D}}(\cdot \mid s)$.

We first introduce the notion of a τ-represented mode.

Definition 1. Let $\tau \in (0, 1]$ and $s \in \mathcal{S}$, and assume that $p_{\mathcal{D}}(\cdot \mid s)$ is multimodal, see Eq. 1. We say that a mode $C_k(s)$ is τ-represented by the policy distribution induced by $G_{\theta,s}$ if and only if $\pi_\theta(C_k(s) \mid s) > \tau$.

This definition means that a mode is considered represented whenever the learned policy places sufficiently large probability, greater than τ, on actions belonging to that mode. We define the number of τ-represented modes by

$$N_\theta^{(\tau)}(s) := \big|\, \{k \in \{1, \ldots, \mathsf{K}(s)\} : \pi_\theta(C_k(s) \mid s) > \tau\} \,\big|.$$

The Lipschitz constant of the map $u \mapsto G_{\theta,s}(u)$ strongly controls the number of modes that the policy distribution generates with sufficiently large probability.

Proposition 2. *Let us fix $s \in \mathcal{S}, \tau \in (0, 1)$. Let us assume that the transport map $G_{\theta,s} : u \mapsto G_\theta(s, u)$ is $L_{\theta,s}$-Lipschitz. Then*

$$N_\theta^{(\tau)}(s) \leq 1 + \left\lfloor \frac{2\big(\sqrt{2\pi}\,|\Phi^{-1}(\tau)| + 1\big)}{\sqrt{2\pi}\,\tau} \frac{L_{\theta,s}}{\Delta(s)} \right\rfloor,$$

where $\Delta(s)$ is the mode separation defined above and Φ^{-1} is the quantile of the standard normal distribution.

We note that the $L_{\theta,s}$-Lipschitz assumption on $G_{\theta,s}$ is broadly satisfied in practical applications, as it only requires the activation function of the neural network to be Lipschitz. Most commonly used activation functions satisfy this property. If the modes are well separated, a generator with a small Lipschitz constant cannot assign probability mass larger than τ to many different modes at the same time. In other words, covering many separated modes requires the generator to stretch the latent space sufficiently. The number of τ-represented modes grows with the ratio $L_{\theta,s}/\Delta(s)$. This result can be interpreted as a mode-representation limitation for Lipschitz generators. A smoother or less expansive generator is more constrained in the number of modes it can represent, while a generator with larger Lipschitz constant has more capacity to spread probability mass across distant regions of the action space. In continuous-time diffusion and flow matching policies, separating modes requires extreme spatial stretching near the terminal sampling steps [36]. Consequently, applying any stabilization or regularization technique that overly restricts the network's Lipschitz constant will inevitably compromise this stretching mechanism and force the generative model into mode collapse [29, 36].

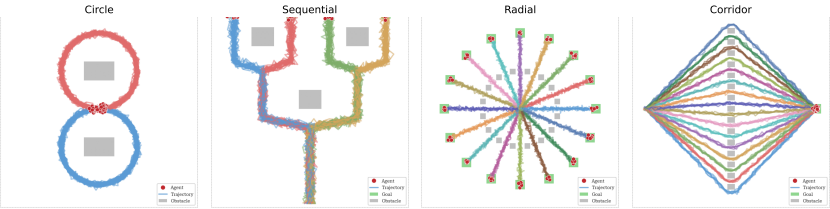


Figure 1: Synthetic datasets. Each panel shows expert trajectories for one task, color-coded by ground-truth mode. Grey rectangles denote obstacles, green rectangles denote goal regions, and red dots denote final agent positions. Policies receive a single RGB observation at $t = 0$ and predict the full open-loop chunk $a_{0:\mathsf{H}-1} \in \mathbb{R}^{60\times 2}$ without replanning.

Remark 4. If the base space U is replaced by the latent space $\mathcal{Z}$ and the policy $\pi_\theta(\cdot \mid s)$ by the decoder $p_\theta(\cdot \mid z, x)$ of the CVAE, Proposition 2 yields the corresponding bound for the decoder, whenever it satisfies the analogous Lipschitz assumption.

5 Experiments

Synthetic navigation tasks. We design four 2D multimodal navigation tasks with ground-truth modes (Figure 1), which allow us to test five research questions from the theory: (RQ1) Is mode coverage necessary for successful rollouts when the true data distribution is multimodal? (RQ2) Do the lower bounds from Proposition 1 and Corollary 1 hold in practice, and how tight are they? (RQ3) How can we provide practical recipes to pick a value for the KL regularizer β? (RQ4, Appendix D.4) Can aggregate posterior matching preserve latent mode information, and how does the geometry of the deployment prior interact with the training posterior? (RQ5) What are the practical consequences of the relationship between $L_{\theta,s}$ and the number of τ-represented modes $N_\theta^{(\tau)}$ of the action-space generators identified by Proposition 2?

Simulated and real-world robot demonstrations. While the synthetic tasks provide ground-truth modes for testing the theory, robot demonstration datasets typically lack such labels. We therefore introduce a heuristic to estimate the conditional action multimodality of a demonstration dataset. This enables us to study the interaction between the multimodality of the policies and the eventual multimodality of the training demonstrations. In simulation, we use Push-T [7], UR3 BlockPush [10], LIBERO[21], Meta-World [43], and Kitchen [11], benchmarks previously used to evaluate multimodal and vision-language policies [7, 19, 30, 31]. For the real-world evaluation, we design a proof-of-concept bimodal tissue-grasping task using a surgical robotic platform [27]. An extensive description of the policy parametrizations, architectures, training recipes, and hyperparameters is given in Appendices C and E.

5.1 Answers to the research questions from the theory

RQ1: when data is multimodal, mode coverage is necessary but not sufficient. Full per-task synthetic results are reported in Appendix D. In the presence of multimodal data, mode coverage proves to be necessary for successful rollouts: the deterministic baseline indeed collapses on the invalid average trajectory on all four tasks (Fig. 6), achieving zero success. However, mode coverage alone is not a sufficient condition for valid deployment: the generative policies recover high valid mode coverage, but on the two hardest $\mathsf{K} = 16$ tasks, success ranges from 0.37 to 0.52 despite near-complete coverage.

RQ2: tightness of the lower bounds and exact mode recovery. In Figure 2, the lower bounds from Proposition 1 hold. The gap between B_{ρ_z} and $I(M; Z \mid S)$ shows the looseness of the Fano bound, which depends on both the mode-recovery error rate and how errors are distributed. The gap between $\mathcal{K}_{\mathrm{pt}}$ and $I(A; Z \mid S)$ is the mismatch term $\mathbb{E}_S(D_{KL}(q_{\phi,\mathcal{D}}(\cdot|S), p_\psi(\cdot|S))$ [13], which measures how well the learned aggregated posterior matches the prior, and gives an indication of how well the model will perform at inference time, when sampling Z from the prior. Consistent with the mode recovery statement in Corollary 1, at small β, the mode information $I(M; Z \mid S)$ is

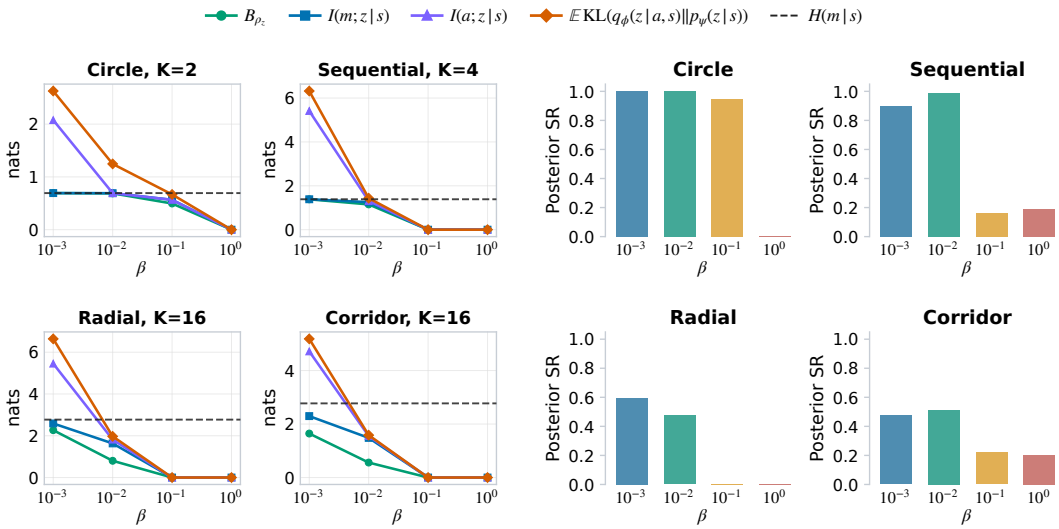


Figure 2: **Pointwise KL can suppress latent mode information.** Information decomposition for KL-CVAE under increasing pointwise KL regularization. For each task, we estimate the latent-space Fano lower bound B_{ρ_z}, the mode information $I(M;Z \mid S)$, the action information $I(A;Z \mid S)$, and the pointwise KL regularizer $\mathcal{K}_{\mathrm{pt}} = \mathbb{E}_{(S,A)\sim p_{\mathcal{D}}}\mathrm{KL}(q_\phi(\cdot \mid A,S) \,\|\, p_\psi(\cdot \mid S))$. The dashed line denotes the empirical mode entropy $H(m \mid s) = \log \mathsf{K}$, since the K modes are balanced. On the right, the success rates (SR) achieved by sampling from the posterior.

close to $H(M \mid S)$, indicating that the latent variable nearly identifies the demonstrated mode. As β increases, both $I(M;Z \mid S)$ and $I(A;Z \mid S)$ decrease sharply beyond $H(M|S)$; consequently, the downstream success rate of the posterior-sampled policy collapses toward zero.

RQ3: selection rule for the value of β. Let us fix a desired error probability $\bar{\rho}$ and compute an estimate of the Fano lower bound $b_0(\bar{\rho})$ as in (19). Corollary 2 provides a threshold $\beta_{\max} := \frac{\mathcal{L}^0}{b_0(\bar{\rho})}$ above which no solution can have the desired mode-recovery accuracy, but for a choice of β below that bound it does not guarantee that training will find a multimodal solution. In Appendix D.3, we provide a heuristic for choosing β for a desired mode-recovery error $\bar{\rho}$. To fulfill the hypothesis of Corollary 2, it is required to check whether the trained policy (with $\beta < \beta_{\max}$) achieves the desired mode-recovery error and a total loss below the collapsed reference $\mathcal{L}^0$. In Table 5 we evaluate several values of β for two of the synthetic tasks. In every run, the empirical inequality$B_{\hat{\rho}} \le D_{\mathrm{KL}} \le \mathcal{L}^0/\beta$ holds. For both tasks, all tested β values below the heuristic recommendation successfully achieve $\hat{\rho} \le \bar{\rho}$. Additionally, consistently with Corollary 2, every run with $D_{\mathrm{KL}} < b_0(\bar{\rho})$ has $\hat{\rho} > \bar{\rho}$, and no run above $\beta_{\max}$ achieves the desired mode recovery. Several runs below $\beta_{\max}$ also fail to achieve it, confirming that satisfying the theoretical threshold alone does not ensure successful mode recovery.

RQ5: relationship between $L_{\theta,s}$ and $N_\theta^{(\tau)}$ in action-space samplers. From Proposition 2 and Corollary 3, it follows that a Lipschitz base-to-action map cannot place non-negligible mass on many well-separated modes unless its Lipschitz constant is sufficiently large, or its bridge (the amount of probability mass in the base space that decodes to invalid trajectories) is large. We aim to test this trade-off empirically. We first estimate the Lipschitz constant using finite differences of the cross-mode transitions (details in Appendix D.5), and then measure the empirical L_{seg}/Δ; see (20) for the definition, and Δ is the mode distance. Figure 3 compares the measured L_{seg}/Δ with the necessary theoretical lower bounds for mode representation at $\tau = 0.01$ (Proposition 2) and for maintaining that representation while limiting bridge mass to 1% (Corollary 3). The benchmarks that can represent all ground-truth modes at $\tau = 0.01$ lie above the bound; Corridor represents 15 of its 16 modes despite exceeding it, as the bound is a necessary condition. The Circle and Sequential runs have high normalized transition sensitivities of approximately 7.9 and 17.7, negligible interpolation bridge fractions, and success rates of 1.00 and 0.99. On the other hand, Radial and Corridor exhibit lower measured sensitivities (2.63 and 1.67), substantial interpolation bridge fractions (0.46 and 0.55), and

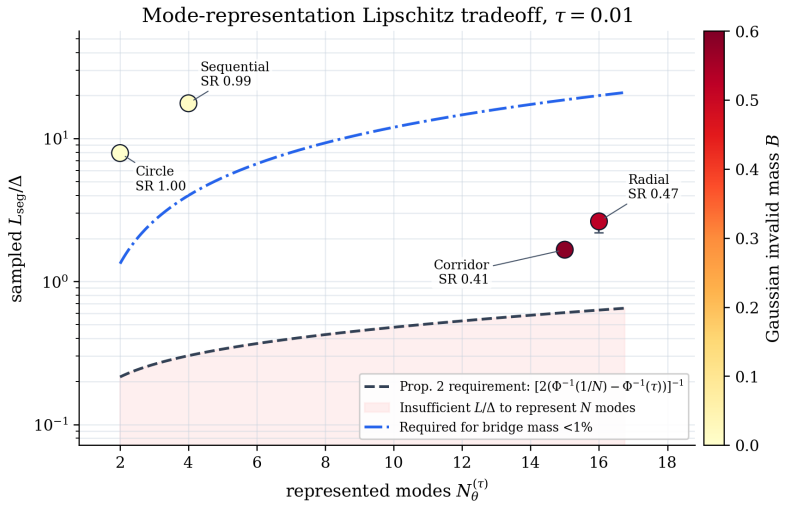


Figure 3: **Action-space sampler tradeoff.** Diffusion and flow-matching-based samplers trade off a high Lipschitz constant for bridge regions. Act-Flow results on the synthetic benchmark are represented.

Table 1: Synthetic conditional-mode estimates, action horizon $= 60$, radius $\varepsilon = 0.5$, and modal-mass cutoff 2.5%. Entropy $\hat{H}$ is in bits.

Benchmark	$\widehat{K}$	$\widehat{H}$
Circle	2.000	0.998
Sequential	4.000	1.986
Radial	15.970	3.972
Corridor	15.920	3.960

Table 2: Dataset-wide conditional-mode estimates at radius $\varepsilon = 0.5$ and modal-mass cutoff 2.5%. Each value is reported per action horizon $= 10/30$.

Dataset	$\widehat{K}$	$\widehat{H}$
UR3 BlockPush	1.136/1.164	0.082/0.103
Kitchen	1.143/1.231	0.083/0.147
PushT	1.068/1.005	0.038/0.004
LIBERO	1.014/1.027	0.007/0.017
MetaWorld	1.037/1.067	0.020/0.043
Bimodal Tissue Grasping	1.468/2.267	0.432/0.760

lower success rates (0.47 and 0.41). These observations associate broad invalid bridge regions with lower Lipschitz constant and lower successful deployment, despite complete or near-complete mode coverage.

5.2 Multimodality in simulation and real robot demonstrations

Estimating the number of modes for unlabeled data. Estimating $K(S)$ and $H(M|S)$ without knowledge about task semantics is a hard problem in general. We propose an estimator based on Gaussian-mixture modal clustering [5] of future action chunks associated with nearby observation states (implementation details in Appendix D.6), which yields estimates for the conditional local mode count and entropy. We validate the estimator on the synthetic benchmarks (Table 1), where it reconstructs the true number of modes. We note here that the multimodality property of the data is horizon-dependent; therefore, the horizons chosen for evaluation are closely related to those used for policy training. Surprisingly, despite their use in prior work to evaluate policies designed to capture multimodality [7, 30, 19, 31], most simulation benchmarks have almost-unimodal support across the dataset states. Push-T, UR3, LIBERO, and Meta-World all give estimates extremely close to 1, while Kitchen is the most multimodal at 1.231 when the action horizon is 30. In our proof-of-concept robot dataset of bimodal tissue grasping, the multimodality is higher, and when the action horizon is long enough, it approximates the ground-truth value of 2 semantic modes.

Policy performance. We evaluate the baseline policies on the simulation and robot benchmarks (Fig. 4 and Appendix D.7). We emphasize here that our goal is not to obtain state-of-the-art performance, but to study how (estimated) multimodality of each robot benchmark's data distribution interacts with different policy parametrizations, and whether policy parametrizations that can represent this

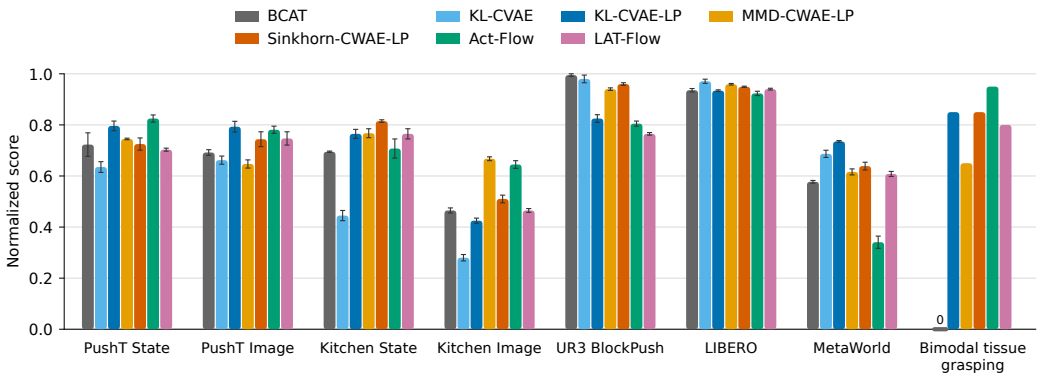


Figure 4: Policy success rate on robot benchmarks, normalized to $[0, 1]$.

property lead to better downstream performance. The largest performance gap between regression and multimodal-capable policies occurs on bimodal tissue grasping, the dataset with the highest estimated multimodality: BCAT averages the two modes and achieves 0% success. A parametrization unable to represent conditional multimodality cannot recover the full conditional expert distribution when the latter exhibits distinct modes. Mode averaging may still yield successful rollouts if the interpolation happens to be valid, but this reflects a fortuitous compatibility with the task constraints rather than faithful, safe imitation. Mode coverage, however, is not a necessary condition for success: Sinkhorn-CWAE collapses to a single mode during training and still retains 85% success. On the other hand, on the estimated nearly unimodal simulation data, deterministic regression remains competitive, while multimodal-capable policies can improve or worsen performance, depending on their training dynamics, and generally increase inference latency. This is consistent with concurrent work [22], which attributes benefits of flow-based policies on common BC benchmarks primarily to supervised iterative computation and training-time noise injection rather than the representation of multimodal action distributions supposedly present in the data.

6 Conclusion

In this work, we formalize multimodality in imitation learning and show that, for both CVAE-based and action-space generative policies, mode preservation is governed by explicit quantities. For CVAEs, the key factor is posterior-prior regularization: stronger regularization improves alignment but can remove the action-conditioned information needed to recover distinct modes. For action-space generative policies with deterministic samplers, the key factor is the Lipschitz constant of the base-to-action map: excessive smoothness prevents assigning significant mass to many well-separated modes. To further emphasize the practical relevance of our analysis, we note that the default choice of β is 10 in ACT [44]. Our analysis reveals that this choice is likely to produce unimodal policies. We empirically validate this in Section 5.1. On the simulation benchmarks, our diagnostics detect little conditional multimodality, and deterministic regression remains competitive. These findings suggest that representing multiple action modes may not be the main bottleneck for policy performance in these settings.

Limitations and future work. Current limitations are that our results are training-independent: Corollary 2 does not ensure that training finds an optimizer $(\hat{\theta}, \hat{\phi}, \hat{\psi})$ with the desired multimodal properties, while Proposition 2 only identifies a structural limitation of low-Lipschitz samplers. Since common regularizers penalize Lipschitz behavior, they may bias policies toward mode collapse. Future work will examine how to estimate the number of modes for datasets where only visual features are available, develop training procedures that balance sampler smoothness and mode preservation, and understand whether the limited conditional multimodality observed in robotic simulation benchmarks also characterizes large-scale real-robot datasets.

7 Acknowledgements

Lorenzo Mazza, Massimiliano Datres, Gitta Kutyniok and Stefanie Speidel acknowledge support by the project "Next Generation AI Computing (gAIn)," funded by the Bavarian Ministry of Science and the Arts and the Saxon Ministry for Science, Culture, and Tourism as well as by the Hightech Agenda Bavaria.

Gitta Kutyniok is also grateful for partial support from the Konrad Zuse School of Excellence in Reliable AI (DAAD).

Additionally, Gitta Kutyniok and Massimiliano Datres acknowledge support by the Munich Center for Machine Learning (MCML).

Gitta Kutyniok furthermore acknowledges support by the German Research Foundation under Grants DFG-SPP-2298, KU 1446/31-1 and KU 1446/32-1, and by the Bavarian Ministry for Digital Affairs.

Ariel Rodriguez, Sebastian Bodenstedt and Stefanie Speidel acknowledge the support by the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as part of Germany's Excellence Strategy – EXC 2050/1 – Project ID 390696704 – Cluster of Excellence "Centre for Tactile Internet with Human-in-the-Loop" (CeTI) of Technische Universität Dresden.

Ariel Rodriguez and Stefanie Speidel would additionally like to thank the Federal Ministry of Research, Technology, and Space (BMFTR) for its support as part of the research program Communication Systems "Souverän. Digital. Vernetzt". Joint project 6G-life, project identification number: 16KIS2413K.

This work has been partially supported by the German Federal Ministry of Research, Technology and Space (BMFTR) under the Robotics Institute Germany (RIG).

References

[1] Christopher M Bishop. Mixture density networks. 1994.

[2] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A Vision-Language-Action Flow Model for General Robot Control. *arXiv preprint arXiv:2410.24164*, 2024.

[3] Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL conference on computational natural language learning*, pages 10–21, 2016.

[4] Alexander Buslaev, Vladimir I Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A Kalinin. Albumentations: fast and flexible image augmentations. *Information*, 11(2):125, 2020.

[5] José E Chacón. Mixture model modal clustering. *Advances in Data Analysis and Classification*, 13(2):379–404, 2019.

[6] Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*, 2016.

[7] 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.

[8] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[9] Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré. Interpolating between optimal transport and MMD using sinkhorn divergences. In *AISTATS*, 2019.

[10] Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Conference on robot learning*, pages 158–168. PMLR, 2022.

[11] Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *arXiv preprint arXiv:1910.11956*, 2019.

[12] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

[13] Matthew D Hoffman and Matthew J Johnson. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in advances in approximate Bayesian inference, NIPS*, volume 1, 2016.

[14] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of machine learning research*, 2013.

[15] H. G. Kenngott, J. J. Wünscher, M. Wagner, A. Preukschas, A. L. Wekerle, P. Neher, S. Suwelack, S. Speidel, F. Nickel, D. Oladokun, L. Maier-Hein, R. Dillmann, H. P. Meinzer, and B. P. Müller-Stich. Openhelp (heidelberg laparoscopy phantom): Development of an open-source surgical evaluation and training tool. *Surgical Endoscopy*, 29(11):3338–3347, 2015. doi: 10.1007/s00464-015-4094-0.

[16] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[17] Ioannis T Konstantinidis, Philip Ituarte, Yanghee Woo, Susanne G Warner, Kurt Melstrom, Jae Kim, Gagandeep Singh, Byrne Lee, Yuman Fong, and Laleh G Melstrom. Trends and outcomes of robotic surgery for gastrointestinal (gi) cancers in the usa: maintaining perioperative and oncologic safety. *Surgical Endoscopy*, 34(11):4932–4942, 2020.

[18] Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. Dart: Noise injection for robust imitation learning. In *Conference on robot learning*, pages 143–156. PMLR, 2017.

[19] Seungjae Lee, Yibin Wang, Haritheja Etukuru, H Jin Kim, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Behavior generation with latent actions. *arXiv preprint arXiv:2403.03181*, 2024.

[20] Yunzhu Li, Jiaming Song, and Stefano Ermon. Infogail: Interpretable imitation learning from visual demonstrations. *Advances in neural information processing systems*, 30, 2017.

[21] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

[22] Chaoyi Pan, Giridharan Anantharaman, Nai-Chieh Huang, Claire Jin, Daniel Pfrommer, Chenyang Yuan, Frank Permenter, Guannan Qu, Nicholas Boffi, Guanya Shi, et al. Much ado about noising: Dispelling the myths of generative robotic control. In *International Conference on Learning Representations*, volume 2026, pages 90575–90614, 2026.

[23] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[24] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

[25] Moritz Reuss and Rudolf Lioutikov. Multimodal diffusion transformer for learning from play. In *2nd Workshop on Language and Robot Learning: Language as Grounding*, 2023.

[26] Moritz Reuss, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. *arXiv preprint arXiv:2304.02532*, 2023.

[27] Ariel Rodriguez, Lorenzo Mazza, Martin Lelis, Rayan Younis, Sebastian Bodenstedt, Martin Wagner, and Stefanie Speidel. An open-source robotics research platform for autonomous laparoscopic surgery. *arXiv preprint arXiv:2603.08490*, 2026.

[28] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[29] 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.

[30] 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.

[31] Ziyad Sheebaelhamd, Michael Tschannen, Michael Muehlebach, and Claire Vernade. Quantization-free autoregressive action transformer. *arXiv preprint arXiv:2503.14259*, 2025.

[32] Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

[33] Max Simchowitz, Daniel Pfrommer, and Ali Jadbabaie. The pitfalls of imitation learning when actions are continuous. *arXiv preprint arXiv:2503.09722*, 2025.

[34] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. volume 28, 2015.

[35] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[36] Arthur Stéphanovitch. Lipschitz regularity in flow matching and diffusion models: sharp sampling rates and functional inequalities. *arXiv preprint arXiv:2604.06065*, 2026.

[37] Vladimir N Sudakov and Boris S Tsirel'son. Extremal properties of half-spaces for spherically invariant measures. *Journal of Soviet Mathematics*, 9(1):9–18, 1978.

[38] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[39] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

[40] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[41] Yating Wang, Haoyi Zhu, Mingyu Liu, Jiange Yang, Hao-Shu Fang, and Tong He. Vq-vla: Improving vision-language-action models via scaling vector-quantized action tokenizers. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11089–11099. IEEE, 2025.

[42] Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. *arXiv preprint arXiv:2208.06193*, 2022.

[43] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020.

[44] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

Appendix Overview

A Notation, Definitions, and Known Results **16**

B Proofs **18**

B.1 Latent-Variable Information Bounds and Collapse 18

B.2 Aggregate Matching 20

B.3 Action-Space Generative Policies 22

C Experimental Details **25**

C.1 Methods 25

C.2 Network Architectures 30

C.3 Benchmarks 32

C.4 Training Details 33

C.5 Evaluation Details 34

D Additional Results **35**

D.1 Complete Synthetic Benchmarks Results 35

D.2 Collapse Diagnostics for Baseline Policies 35

D.3 KL-Regularized Latent Policies 35

D.4 Aggregate Matching and Prior Coverage 37

D.5 Action-Space Bridge–Smoothness Trade-off 40

D.6 Measuring Multimodality in Robot Benchmark Datasets 40

D.7 Simulation and Real-World Robot Results 43

D.8 Relationship Between Policy Performance and Action Horizon 43

E Experiment Hyperparameters **47**

A Notation, Definitions, and Known Results

We denote generic spaces with calligraphic letters, random variables with uppercase letters, their realizations with lowercase letters, and fixed problem constants with sans-serif capital letters. Generic measurable sets are denoted by uppercase letters such as B, C. Given a space $\mathcal{X}$, $\mathcal{X}^{\mathsf{T}}$ denotes the space of length-T sequences with entries in a generic space $\mathcal{X}$, namely $x = (x_1, \ldots, x_{\mathsf{T}})$ with $x_i \in \mathcal{X}$. Event probabilities are written as $\Pr(\cdot)$ while expectations are denoted with $\mathbb{E}$. We use subscripts to indicate the random variable with respect to which $\Pr$ and $\mathbb{E}$ are taken whenever this is not clear from the context. Let $(\mathcal{A}, d_{\mathcal{A}})$ be a metric space. For $B, C \subseteq \mathcal{A}$, their distance is

$$\operatorname{dist}_{\mathcal{A}}(B, C) = \inf_{b \in B,\, c \in C} d_{\mathcal{A}}(b, c).$$

When $\mathcal{A} \subseteq \mathbb{R}^{d_a}$, we set $d_{\mathcal{A}}(a, a') = \|a - a'\|_2$. For notational simplicity, we denote the Euclidean norm simply by $\|\cdot\|$. All logarithms are natural, so information quantities are measured in nats. The support of a random variable X, denoted by $\operatorname{supp}(X)$, is the support of its law μ_X. In particular, in the discrete case, $\operatorname{supp}(X) = \{x : p_X(x) > 0\}$.

We write $X \sim \mu$ to mean that the random variable X has distribution μ; that is, for every measurable set B, $\Pr(X \in B) = \mu(B)$. When μ is absolutely continuous with respect to the Lebesgue measure, we denote its density by $p(x)$. Equivalently, in this case we may write $X \sim p$ meaning that $\mu(B) = \int_B p(x)dx$. If the support of μ is discrete, then μ does not admit a density with respect to the Lebesgue measure. In this case, writing $\operatorname{supp}(\mu) = \{x_i\}_{i \in I}$ where I is finite or countable, we identify μ with its probability mass function

$$p(x_i) = \mu(\{x_i\}), \qquad i \in I.$$

Equivalently, in the sense of measures, one may write $\mu = \sum_{i \in I} p_\mu(x_i)\, \delta_{x_i}$, where δ_{x_i} denotes the Dirac measure at x_i. We write $\mu \ll \nu$ to mean that μ is absolutely continuous with respect to ν, namely, for every measurable set A, $\nu(A) = 0$ implies $\mu(A) = 0$. We denote by $\mu_{X,Y}$ the joint law of a pair of random variables (X, Y), by $\mu_{X|Y=y}$ the conditional law of X given $Y = y$, and by $p_{X,Y}$ and $p_{X|Y}(\cdot \mid y)$ their corresponding densities, with

$$p_{X|Y}(x \mid y) = \frac{p_{X,Y}(x, y)}{p_Y(y)} \qquad \text{for } p_Y(y) > 0.$$

We will measure the discrepancy between probability measures using the Kullback-Leibler divergence, defined as follows.

Definition 2 (Kullback-Leibler divergence). Let $X \sim \mu$ and $Y \sim \nu$ be random variables such that $\mu \ll \nu$. The Kullback–Leibler divergence from ν to μ is defined as

$$\mathrm{KL}(\mu \| \nu) = \int \log\left(\frac{d\mu}{d\nu}\right) d\mu.$$

We define next the notions of Shannon entropy, differential entropy, conditional entropy, and conditional mutual information. For a discrete random variable, the Shannon entropy $H(X)$ measures the uncertainty of the probability mass assigned to its possible values. For a continuous random variable, the differential entropy $h(X)$ measures the spread of its density. Conditional entropy measures the remaining uncertainty after observing another variable. Conditional mutual information $I(X; Y \mid Z)$ measures how much information X and Y share once Z is known.

Definition 3 (Entropy and conditional entropy). Let $X \sim \mu$, $Y \sim \nu$ be random variables. The entropy of X is defined as

$$H(X) := -\int \log p_X(x)\, \mu(dx),$$

with the convention $0 \log 0 = 0$. The conditional entropy of X given Y is defined as

$$H(X \mid Y) := -\int \log p_{X|Y}(x \mid y)\, \mu_{X,Y}(dx, dy)\,.$$

We note that, for any random variable X supported on a finite set $\mathcal{X}$,

$$H(X) \le \log |\mathcal{X}|,$$

with equality if and only if X is uniformly distributed on $\mathcal{X}$. For $p \in [0,1]$, we denote by $H_b(p)$ the binary entropy function,

$$H_b(p) = -p \log p - (1-p) \log(1-p).$$

Equivalently, if $X \sim \mathrm{Ber}(p)$, then $H(X) = H_b(p)$.

Mutual information provides a measure of statistical dependence between random variables, quantifying how much knowing one variable reduces uncertainty about another.

Definition 4 (Conditional mutual information). For random variables X, Y, Z, the *conditional mutual information* between X and Y given Z is

$$I(X;Y \mid Z) := H(X \mid Z) - H(X \mid Y, Z),$$

whenever X is discrete and the conditional entropy terms are well defined. If X is continuous, the same identity holds with differential entropies $h(\cdot)$ in place of $H(\cdot)$, whenever the terms are well defined. When X, Y, Z are continuous and admit densities, this is equivalently

$$I(X;Y \mid Z) = \int p(x,y,z) \log \frac{p(x,y \mid z)}{p(x \mid z) p(y \mid z)} \, dx \, dy \, dz.$$

Two standard properties of mutual information that will be used throughout are the following:

(P1) For random variables X, Y, mutual information is symmetric:

$$I(X;Y) = I(Y;X).$$

(P2) For random variables X, Y, Z, conditional mutual information satisfies

$$I(X;Y \mid Z) = H(X \mid Z) - H(X \mid Y, Z),$$

whenever X is discrete and the entropy terms are well defined. For continuous X, the analogous identity holds with differential entropies.

Of particular interest in this work are Markov chains of random variables.

Definition 5 (Markov chain). Let p be a joint distribution on $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$. We write $X \to Y \to Z$ if X and Z are conditionally independent given Y, denoted by $X \perp Z \mid Y$. Equivalently, the conditional distribution of Z given (X, Y) depends only on Y, that is

$$p(z \mid x, y) = p(z \mid y) \qquad p\text{-a.s.}$$

A fundamental standard consequence of the Markov-chain structure is the following inequality, called data-processing inequality.

Lemma 1 (Data-processing inequality). *Let $(X, Y, Z) \sim p$ such that $X \to Y \to Z$ forms a Markov chain. Then*

$$I(X;Z) \le I(X;Y).$$

Fano's inequality relates the probability of incorrectly estimating a discrete random variable to its remaining conditional entropy given the estimator.

Lemma 2 (Fano's inequality). *Let M be a random variable taking values in a finite set C with $|C| = \mathsf{K}$, where $\mathsf{K} > 1$. Let $\hat{M}$ be an estimator of M, that is, a random variable taking values in C. Define the probability of error*

$$e := \Pr(\hat{M} \neq M).$$

Then

$$H(M \mid \hat{M}) \le H_b(e) + e \log(\mathsf{K} - 1),$$

where

$$H_b(e) = -e \log e - (1-e) \log(1-e)$$

is the binary entropy function.

B Proofs

B.1 Latent-Variable Information Bounds and Collapse

We first prove the latent-variable results from §4. The subsection has three parts. We prove the mode-information lower bound, then specialize it to the uniform exact-recovery case, and finally derive the objective-level collapse criterion for pointwise-KL regularization. Let

$$q_{\phi,\mathcal{D}}(s,a,z) := p_{\mathcal{D}}(s,a)\, q_\phi(z \mid a,s) = p_{\mathcal{D},S}(s)\, p_{\mathcal{D}}(a \mid s)\, q_\phi(z \mid a,s)$$

denote the posterior-encoder-induced joint distribution over (S, A, Z). Conditioning the joint law $q_{\phi,\mathcal{D}}$ on $S = s$ gives

$$q_{\phi,\mathcal{D}}(a, z \mid S = s) = p_{\mathcal{D}}(a \mid s)\, q_\phi(z \mid a, s).$$

Throughout the subsection, all statewise probabilities, entropies, and mutual informations below are taken under this conditional law.

Proof of Proposition 1 We prove Proposition 1. The proof rests on two Markov chains, $M \to A \to Z$ and $M \to Z \to \hat{M}$, each yielding a data-processing inequality. Combining them with Fano's inequality gives the bound. We now make this precise. Fix $s \in \mathrm{supp}(p_{\mathcal{D},S})$ with $\mathsf{K}(s) > 1$. Under the conditional law at $S = s$, we have

$$A \sim p_{\mathcal{D}}(\cdot \mid s), \qquad Z \sim q_\phi(\cdot \mid A, s), \qquad M = g(A,s), \qquad \hat{A} = f_\theta(Z,s).$$

Let $\hat{M}$ be the mode label induced by $\hat{A}$: if $\hat{A} \in C_j(s)$, set $\hat{M} = j$; if $\hat{A} \notin \bigcup_{j=1}^{\mathsf{K}(s)} C_j(s)$, assign $\hat{M}$ arbitrarily in $\{1, \dots, \mathsf{K}(s)\}$. We start by noticing that since $M = g(A,s)$ is a deterministic function of A at fixed s, the conditional law of M given $A = a$ is degenerate:

$$q_{\phi,\mathcal{D}}(M = m \mid A = a, S = s) = \mathbf{1}_{\{m = g(a,s)\}}.$$

Moreover, Z is sampled from $q_\phi(\cdot \mid A, s)$, so conditioning additionally on M does not change its law:

$$q_{\phi,\mathcal{D}}(z \mid A = a, M = m, S = s) = q_{\phi,\mathcal{D}}(z \mid A = a, S = s) = q_\phi(z \mid a, s).$$

Therefore, at fixed s, we have the Markov chain

$$M \to A \to Z.$$

Since $M = g(A,s)$ is a deterministic function of A, we also have $Z \to A \to M$ at fixed s. By data processing,

$$I(Z; M \mid S = s) \le I(Z; A \mid S = s).$$

By symmetry of mutual information, this gives

$$I(M; Z \mid S = s) \le I(A; Z \mid S = s). \tag{7}$$

Similarly, at fixed s, $\hat{M}$ is a deterministic function of Z, hence

$$M \to Z \to \hat{M}.$$

Therefore, by Lemma 1, it holds

$$I(M; \hat{M} \mid S = s) \le I(M; Z \mid S = s). \tag{8}$$

Combining (7) and (8), it holds

$$I(A; Z \mid S = s) \ge I(M; Z \mid S = s) \ge I(M; \hat{M} \mid S = s). \tag{9}$$

Let

$$e(s) := \Pr(\hat{M} \neq M \mid S = s)$$

be the conditional decoded-mode error. By construction of $\hat{M}$, see (2), if $\hat{A} \in C_M(s)$, then $\hat{M} = M$. Hence

$$\{\hat{M} \neq M\} \subseteq \{\hat{A} \notin C_M(s)\} = \{f_\theta(Z,s) \notin C_M(s)\}.$$

Therefore, by the definition of $\rho(s)$ in (5),

$$e(s) = \Pr(\hat{M} \neq M \mid S = s) \leq \Pr(f_\theta(Z, s) \notin C_M(s) \mid S = s) = \rho(s).$$

Given s, since M takes values in $\{1, \ldots, \mathsf{K}(s)\}$, Lemma 2 gives, for $\mathsf{K}(s) > 1$,

$$\begin{aligned} H(M \mid \hat{M}, S = s) &\leq H_b(e(s)) + e(s) \log(\mathsf{K}(s) - 1) \\ &\leq H_b(\rho(s)) + \rho(s) \log(\mathsf{K}(s) - 1). \end{aligned} \tag{10}$$

The last inequality follows from $e(s) \leq \rho(s)$ and from the fact that, if $\rho(s) \leq 1 - 1/\mathsf{K}(s)$, then

$$p \mapsto H_b(p) + p \log(\mathsf{K}(s) - 1)$$

is non-decreasing on $[0, \rho(s)]$.

Combining (P2), (9), and (10), we obtain

$$\begin{aligned} I(A; Z \mid S = s) &\geq I(M; \hat{M} \mid S = s) \\ &= H(M \mid S = s) - H(M \mid \hat{M}, S = s) \\ &\geq H(M \mid S = s) - H_b(\rho(s)) - \rho(s) \log(\mathsf{K}(s) - 1). \end{aligned}$$

Averaging over $S \sim p_{\mathcal{D},S}$ gives

$$I(A; Z \mid S) \geq B_\rho,$$

which proves the first claim of Proposition 1.

By the definition of B_ρ, this gives

$$I(A; Z \mid S) \geq B_\rho.$$

Recall that $q_{\phi,\mathcal{D}}(s, a, z)$ denotes the posterior-encoder-induced joint distribution and $q_{\phi,\mathcal{D}}(z \mid s)$ is the state-conditional aggregated posterior. The pointwise latent regularizer decomposes into a mutual-information term between the action and latent code, and a mismatch term between the aggregated posterior and the prior:

$$\mathbb{E}_{(S,A)\sim p_{\mathcal{D}}} \mathrm{KL}(q_\phi(\cdot \mid A, S) \,\|\, p_\psi(\cdot \mid S)) = I(A; Z \mid S) + \mathbb{E}_{S\sim p_{\mathcal{D},S}} \mathrm{KL}(q_{\phi,\mathcal{D}}(\cdot \mid S) \,\|\, p_\psi(\cdot \mid S)) \,. \tag{11}$$

The full derivation of this decomposition can be found, for example, in Hoffman and Johnson [13]. The second term in (11) is nonnegative. Hence

$$\mathbb{E}_{(S,A)\sim p_{\mathcal{D}}} \mathrm{KL}(q_\phi(\cdot \mid A, S) \,\|\, p_\psi(\cdot \mid S)) \geq I(A; Z \mid S) \geq B_\rho \,,$$

which concludes the proof. □

Proof of Corollary 1. Using $\rho(s) = 0$ for $p_{\mathcal{D},S}$-almost every s, we have

$$H_b(\rho(s)) + \rho(s) \log(\mathsf{K}(s) - 1) = 0$$

for every s with $\mathsf{K}(s) > 1$, where we recall that $H_b(p) = -p \log p - (1-p) \log(1-p)$ and the usual convention $H_b(0) = 0$.

Moreover, since $M \mid S = s \sim \mathrm{Unif}(\{1, \ldots, \mathsf{K}(s)\})$, we have

$$\mathbb{E}_{S\sim p_{\mathcal{D},S}}[H(M \mid S)] = \mathbb{E}_{S\sim p_{\mathcal{D},S}}[\log \mathsf{K}(S)].$$

These two observations, combined with Proposition 1, imply

$$I(A; Z \mid S) \geq \mathbb{E}_{S\sim p_{\mathcal{D},S}} \log \mathsf{K}(S).$$

□

Proof of Corollary 2 Let us denote with

$$\hat{\rho}(s) := \Pr\big(f_{\hat{\theta}}(Z, s) \notin C_M(s) \,\big|\, S = s\big)\,.$$

Assume that

$$\hat{\rho}(s) \le 1 - \frac{1}{\mathsf{K}(s)}$$

for all $s \in \mathrm{supp}(p_{\mathcal{D},S})$ with $\mathsf{K}(s) > 1$. By *(H1)* and the optimality of $(\hat{\theta}, \hat{\phi}, \hat{\psi})$, we have

$$\begin{aligned}\hat{\mathcal{L}}^{\beta} = \mathcal{L}^{\beta}_{\mathrm{var}}(\hat{\theta}, \hat{\phi}, \hat{\psi}) &\le \mathcal{L}^{\beta}_{\mathrm{var}}(\theta_0, \phi_0, \psi_0) \\ &= \mathcal{L}^{0}_{\mathrm{var}}(\theta_0, \phi_0, \psi_0) =: \mathsf{C},\end{aligned} \tag{12}$$

where the equality follows from $D(q_{\phi_0}, p_{\psi_0}) = 0$, and $\mathsf{C} > 0$ is independent of β.

By (H2) and Proposition 1, it holds that

$$I(A; Z \mid S) \ge B_{\hat{\rho}}, \tag{13}$$

On the other hand, by the decomposition in (11), applied to $(\hat{\phi}, \hat{\psi})$, we have

$$\begin{aligned}I(A; Z \mid S) &= \mathbb{E}_{(S,A)\sim p_{\mathcal{D}}} \mathrm{KL}\Big(q_{\hat{\phi}}(\cdot \mid A, S) \,\|\, p_{\hat{\psi}}(\cdot \mid S)\Big) - \mathbb{E}_{S\sim p_{\mathcal{D},S}} \mathrm{KL}\Big(q_{\hat{\phi},\mathcal{D}}(\cdot \mid S) \,\|\, p_{\hat{\psi}}(\cdot \mid S)\Big) \\ &\le \mathbb{E}_{(S,A)\sim p_{\mathcal{D}}} \mathrm{KL}\Big(q_{\hat{\phi}}(\cdot \mid A, S) \,\|\, p_{\hat{\psi}}(\cdot \mid S)\Big) = D(q_{\hat{\phi}}, p_{\hat{\psi}}).\end{aligned} \tag{14}$$

Since the reconstruction loss is nonnegative, the objective satisfies

$$\hat{\mathcal{L}}^{\beta} = \mathcal{L}_{\mathrm{rec}}(\hat{\theta}, \hat{\phi}) + \beta\, D(q_{\hat{\phi}}, p_{\hat{\psi}}) \ge \beta\, D(q_{\hat{\phi}}, p_{\hat{\psi}}).$$

Together with $\hat{\mathcal{L}}^{\beta} \le \mathsf{C}$, this gives

$$\beta\, D(q_{\hat{\phi}}, p_{\hat{\psi}}) \le \mathsf{C}.$$

Therefore,

$$I(A; Z \mid S) \le D(q_{\hat{\phi}}, p_{\hat{\psi}}) \le \frac{\mathsf{C}}{\beta}.$$

Combining this with (13) gives

$$B_{\hat{\rho}} \le I(A; Z \mid S) \le \frac{\mathsf{C}}{\beta}.$$

Consequently, if $B_{\hat{\rho}} > \mathsf{b}_0$, then it follows trivially that

$$\beta < \frac{\mathsf{C}}{\mathsf{b}_0}.$$

□

B.2 Aggregate Matching

We next show why aggregate matching can preserve mode information even when pointwise KL would penalize it. The key observation is that the aggregated posterior is a mixture over mode-conditioned latent distributions.

We write $\mathcal{Z} := \mathbb{R}^{\mathsf{d}_z}$ for the latent space.

Proposition 3 (Aggregate matching can preserve mode information). *Under the setting introduced above, fix $s \in \mathrm{supp}(p_{\mathcal{D},S})$. For each mode index $k \in \{1, \dots, \mathsf{K}(s)\}$ with $p_{\mathcal{D}}(M = k \mid S = s) > 0$, define*

$$q_k(\cdot \mid s) := q_{\phi,\mathcal{D}}(\cdot \mid M = k, S = s).$$

That is, $q_k(\cdot \mid s)$ is the latent distribution induced by the posterior encoder when the demonstrated action belongs to mode k. Then the state-conditioned aggregated posterior is

$$q_{\phi,\mathcal{D}}(\cdot \mid S = s) = \sum_{k=1}^{\mathsf{K}(s)} p_{\mathcal{D}}(M = k \mid S = s)\, q_k(\cdot \mid s).$$

If the prior satisfies

$$p_\psi(\cdot \mid s) = \sum_{k=1}^{\mathsf{K}(s)} p_{\mathcal{D}}(M = k \mid S = s)\, q_k(\cdot \mid s),$$

then the aggregated posterior–prior mismatch is zero. Moreover, if the supports of the components $q_k(\cdot \mid s)$ are mutually disjoint, then there exists a deterministic decoder $h_s : \mathcal{Z} \to \{1, \dots, \mathsf{K}(s)\}$ such that

$$\Pr(h_s(Z) = M \mid S = s) = 1,$$

and therefore

$$I(M; Z \mid S = s) = H(M \mid S = s).$$

Proof. We first compute the state-conditional aggregated posterior and then show that disjoint latent supports make the mode recoverable from Z.

By the law of total probability over the discrete mode label M,

$$q_{\phi,\mathcal{D}}(\cdot \mid S = s) = \sum_{k=1}^{\mathsf{K}(s)} p_{\mathcal{D}}(M = k \mid S = s)\, q_{\phi,\mathcal{D}}(\cdot \mid M = k, S = s).$$

By definition,

$$q_k(\cdot \mid s) = q_{\phi,\mathcal{D}}(\cdot \mid M = k, S = s),$$

and hence

$$q_{\phi,\mathcal{D}}(\cdot \mid S = s) = \sum_{k=1}^{\mathsf{K}(s)} p_{\mathcal{D}}(M = k \mid S = s)\, q_k(\cdot \mid s).$$

Thus, if

$$p_\psi(\cdot \mid s) = \sum_{k=1}^{\mathsf{K}(s)} p_{\mathcal{D}}(M = k \mid S = s)\, q_k(\cdot \mid s),$$

then

$$q_{\phi,\mathcal{D}}(\cdot \mid S = s) = p_\psi(\cdot \mid s),$$

so the aggregated posterior–prior mismatch is zero.

Recall now that, by definition of $q_k(\cdot \mid s)$, for each $k \in \{1, \dots, \mathsf{K}(s)\}$ with $p_{\mathcal{D}}(M = k \mid S = s) > 0$, we have

$$Z \mid (M = k, S = s) \sim q_k(\cdot \mid s).$$

Let

$$E_k(s) := \operatorname{supp} q_k(\cdot \mid s).$$

If the sets $E_1(s), \dots, E_{\mathsf{K}(s)}(s)$ are mutually disjoint, define $h_s : \mathcal{Z} \to \{1, \dots, \mathsf{K}(s)\}$ by $h_s(z) = k$ whenever $z \in E_k(s)$, with arbitrary values outside $\bigcup_k E_k(s)$. Since $Z \mid (M = k, S = s)$ is supported on $E_k(s)$, it follows that

$$\Pr(h_s(Z) = M \mid S = s) = 1.$$

Therefore, at fixed s, the mode label M is determined by Z, and hence

$$H(M \mid Z, S = s) = 0.$$

By Definition 4, it follows trivially that

$$I(M; Z \mid S = s) = H(M \mid S = s).$$

This concludes the proof. □

Remark 5 (Deployment-time prior coverage). Proposition 3 shows that the prior can match the aggregated posterior while the latent variable still identifies the demonstrated mode, avoiding the pointwise-KL pressure on $I(A; Z \mid S)$. The limitation is that aggregate matching constrains only the state-conditioned marginal

$$q_{\phi,\mathcal{D}}(\cdot \mid S = s) = \int q_\phi(\cdot \mid a, s)\, p_{\mathcal{D}}(a \mid s)\, da,$$

and not each action-conditioned posterior $q_\phi(\cdot \mid a, s)$. Thus two encoder families can have identical, or very similar, aggregated marginals while having very different conditionals $q_\phi(\cdot \mid a, s)$. Consequently, matching $p_\psi(\cdot \mid s)$ to the aggregated posterior does not by itself guarantee that prior samples decode to valid actions.

We can see this by defining, for a fixed observation s and mode index $k \in \{1, \dots, \mathsf{K}(s)\}$, the decoder-induced latent region

$$R_k(s) := \{z \in \mathcal{Z} : \ f_\theta(z, s) \in C_k(s)\}.$$

Deployment covers mode k only if $p_\psi(R_k(s) \mid s)$ is sufficiently large. Thus aggregate matching must align the prior with the posterior mass on the decoder-induced regions $\{R_k(s)\}_{k=1}^{\mathsf{K}(s)}$, and not only match aggregate latent marginals in regions irrelevant to the decoder.

B.3 Action-Space Generative Policies

Proof of Proposition 2 The proof relies on the Gaussian isoperimetric inequality from Sudakov and Tsirel'son [37] to avoid dependency on the dimension d of the base space of U. We will denote with γ_d the standard Gaussian measure on $\mathbb{R}^d$, so that, for any measurable set $A \subseteq \mathbb{R}^d$,

$$\gamma_d(A) := \Pr_{U \sim \mathcal{N}(0, I_d)}(U \in A).$$

Let us fix $s \in \mathcal{S}$, $\tau \in (0, 1)$, and a parameter vector θ. For each $k \in \{1, \dots, K\}$, define the set of points in the base space that the generator $G_{\theta,s}$ maps into action mode $C_k(s)$:

$$A_k := G_{\theta,s}^{-1}\big(C_k(s)\big) = \big\{u \in \mathbb{R}^d : G_{\theta,s}(u) \in C_k(s)\big\}.$$

Let $k \neq k'$. If $A_k = \varnothing$ or $A_{k'} = \varnothing$, then by definition $dist(A_k, A_{k'}) = \infty$. If $A_k, A_{k'} \neq \varnothing$, let $u \in A_k$ and $u' \in A_{k'}$. By definition, $G_{\theta,s}(u) \in C_k(s)$ and $G_{\theta,s}(u') \in C_{k'}(s)$. Hence

$$\operatorname{dist}\big(G_{\theta,s}(u), G_{\theta,s}(u')\big) \geq \operatorname{dist}\big(C_k(s), C_{k'}(s)\big) \geq \Delta(s).$$

But, by the $L_{\theta,s}$-Lipschitzianity of $G_{\theta,s}$, it follows that

$$\operatorname{dist}\big(G_{\theta,s}(u), G_{\theta,s}(u')\big) \leq L_{\theta,s}\|u - u'\|,$$

Therefore, combining the bounds, it holds

$$\|u - u'\| \geq \frac{\Delta(s)}{L_{\theta,s}}.$$

Taking the infimum over u, u' proves that

$$\operatorname{dist}(A_k, A_{k'}) = \inf_{\substack{u \in A_k \\ u' \in A_{k'}}} \|u - u'\| \geq \frac{\Delta(s)}{L_{\theta,s}}, \qquad k \neq k'. \tag{15}$$

To avoid a dimension-dependent bound, we introduce the following extended set. Let $0 < r' < \frac{\Delta(s)}{2L_{\theta,s}}$ and defined

$$A_k^{r'} := \{u \in \mathbb{R}^d : \ \operatorname{dist}(u, A_k) \leq r'\}$$

where $\operatorname{dist}(u, A) = \min_{v \in A_k} \|u - v\|_2$. Then, it is easy to see that

$$A_k^{r'} \cap A_{k'}^{r'} = \varnothing, \qquad k \neq k', \qquad 0 < r' < \frac{\Delta(s)}{2L_{\theta,s}}, \tag{16}$$

since $\operatorname{dist}(A_k, A_{k'}) \geq \frac{\Delta(s)}{L_{\theta,s}}$.

By the standard Gaussian isoperimetric inequality, it holds

$$\gamma_d\big(A_k^{r'}\big) \geq \Phi\big(\Phi^{-1}(\pi_\theta(C_k(s)|S = s)) + r'\big), \qquad k = 1, \dots, K. \tag{17}$$

Therefore, since $A_k^{r'} \cap A_{k'}^{r'} = \varnothing$ for $k \neq k'$,

$$\sum_{k=1}^{K} \Phi\big(\Phi^{-1}(\pi_\theta(C_k(s)|S = s)) + r'\big) \leq \sum_{k=1}^{K} \gamma_d\big(A_k^{r'}\big) = \gamma_d\left(\bigcup_{k=1}^{K} A_k^{r'}\right) \leq 1.$$

By continuity of Φ,

$$\sum_{k=1}^{K} \Phi\left(\Phi^{-1}(\pi_\theta(C_k(s)|S=s)) + \frac{\Delta(s)}{2L_{\theta,s}}\right) = \lim_{r' \to \left(\frac{\Delta(s)}{2L_{\theta,s}}\right)^-} F(r') \leq 1.$$

Let us recall the definition of number of τ-represented modes by the sampler as:

$$N_\theta^{(\tau)}(s) = |\left\{k \in \{1,\dots,K(s)\} : \pi_\theta\big(C_k(s) \mid s\big) > \tau\right\}|.$$

By monotonicity of Φ^{-1}, for all $k \in \left\{k \in \{1,\dots,K(s)\} : \pi_\theta\big(C_k(s) \mid s\big) > \tau\right\}$, it holds that

$$\Phi^{-1}(\pi_\theta\big(C_k(s) \mid s\big)) \geq \Phi^{-1}(\tau)$$

and hence

$$\Phi\big(\Phi^{-1}(\pi_\theta\big(C_k(s) \mid s\big)) + \frac{\Delta(s)}{2L_{\theta,s}}\big) \geq \Phi\big(\Phi^{-1}(\tau) + \frac{\Delta(s)}{2L_{\theta,s}}\big).$$

Summing over k gives

$$\begin{aligned}
1 &\geq \sum_{k=1}^{K(s)} \Phi\big(\Phi^{-1}(\pi_\theta\big(C_k(s) \mid s\big)) + \frac{\Delta(s)}{2L_{\theta,s}}\big) \\
&\geq \sum_{k \in \left\{k \in \{1,\dots,K(s)\} : \pi_\theta\left(C_k(s)|s\right) > \tau\right\}} \Phi\big(\Phi^{-1}(\tau) + \frac{\Delta(s)}{2L_{\theta,s}}\big) \\
&\geq N_\theta^{(\tau)}(s)\, \Phi\big(\Phi^{-1}(\tau) + \frac{\Delta(s)}{2L_{\theta,s}}\big) \\
&\geq N_\theta^{(\tau)}(s) \frac{\sqrt{2\pi}\tau\, \Delta(s)/2L_{\theta,s}}{\sqrt{2\pi}\tau\Delta(s)/2L_{\theta,s} + \sqrt{2\pi}|\Phi^{-1}(\tau)| + 1},
\end{aligned}$$

where in the last step we have used the fact that, for all $\tau > 0$ and $r > 0$, it holds

$$\Phi(\Phi^{-1}(\tau) + r) \geq \frac{\sqrt{2\pi}\tau r}{\sqrt{2\pi}\tau r + \sqrt{2\pi}|\Phi^{-1}(\tau)| + 1}.$$

Inverting the previous equation, we conclude that

$$N_\theta^{(\tau)}(s) \leq 1 + \left\lfloor \frac{2(\sqrt{2\pi}|\Phi^{-1}(\tau)| + 1)}{\sqrt{2\pi}\tau} \frac{L_{\theta,s}}{\Delta_s} \right\rfloor,$$

which concludes the proof. □

Corollary: stretch-bridge trade-off The argument underlying Proposition 2 also yields a trade-off between the generator's Lipschitz constant and the bridge mass, i.e., the probability assigned outside the action modes.

Corollary 3. *Under the assumptions of Proposition 2, define the bridge mass at state s as*

$$B_\theta(s) := \pi_\theta\left(\mathcal{A} \setminus \bigcup_{k=1}^{\mathsf{K}(s)} C_k(s) \,\middle|\, s\right).$$

Then,

$$L_{\theta,s}\, B_\theta(s) \geq \big(N_\theta^{(\tau)}(s) - 1\big) \frac{\Delta(s)}{2}\, \varphi\big(\Phi^{-1}(\tau)\big),$$

where $\varphi(x) = (2\pi)^{-1/2} e^{-x^2/2}$ is the standard normal density.

Proof of Corollary 3 Let us fix s. If $N_\theta^{(\tau)}(s) \le 1$, the result is immediate. Otherwise, choose any τ-represented mode with index k_0.

For r' as in (16), (15) and (16) imply that the added regions $A_k^{r'} \setminus A_k$ are pairwise disjoint and lie outside all mode preimages. Thus,

$$B_\theta(s) \ge \sum_{\substack{k \ne k_0 \\ \pi_\theta(C_k(s)|s) > \tau}} \Big[\gamma_d(A_k^{r'}) - \pi_\theta(C_k(s) \mid s)\Big]. \tag{18}$$

For each index in this sum, by (17) and (16),

$$\Phi\big(\Phi^{-1}(\pi_\theta(C_k(s) \mid s)) + r'\big) \le \gamma_d(A_k^{r'}) \le 1 - \pi_\theta(C_{k_0}(s) \mid s) < 1 - \tau.$$

Since $\pi_\theta(C_k(s) \mid s) > \tau$, it follows that

$$\Phi^{-1}(\pi_\theta(C_k(s) \mid s)) + t \in \big[\Phi^{-1}(\tau), -\Phi^{-1}(\tau)\big], \qquad 0 \le t \le r'.$$

The standard normal density on this interval is at least $\varphi(\Phi^{-1}(\tau))$. Applying (17) therefore gives

$$\begin{aligned}\gamma_d(A_k^{r'}) - \pi_\theta(C_k(s) \mid s) &\ge \int_0^{r'} \varphi\big(\Phi^{-1}(\pi_\theta(C_k(s) \mid s)) + t\big)\, dt \\ &\ge r'\, \varphi(\Phi^{-1}(\tau)).\end{aligned}$$

Substituting into (18), which contains $N_\theta^{(\tau)}(s) - 1$ terms, yields

$$B_\theta(s) \ge \big(N_\theta^{(\tau)}(s) - 1\big) r'\, \varphi(\Phi^{-1}(\tau)).$$

Letting $r' \uparrow \Delta(s)/(2L_{\theta,s})$ and multiplying by $L_{\theta,s}$ proves the corollary. □

C Experimental Details

This section specifies the policy parameterizations and training objectives used in the experiments, using the notation introduced in Appendix A. We write objectives with respect to the population expert state-action law $p_{\mathcal{D}}$; in implementation, these objectives are optimized using minibatches $\{(s_i, a_i)\}_{i=1}^{\mathsf{B}}$ from the realized dataset $\tilde{\mathcal{D}}$. All methods use the same observation encoder and action-chunking interface, and differ only in how they parameterize the conditional distribution over action chunks. We write $\mathcal{L}_{\mathrm{rec}}(\hat{a}, a)$ for the action-chunk reconstruction loss, instantiated as an L_1 loss in all deterministic-decoder latent policies unless stated otherwise. For latent-variable policies, training uses posterior latents $Z \sim q_\phi(\cdot \mid A, S)$, while deployment uses prior latents $Z \sim p_\psi(\cdot \mid S)$.

C.1 Methods

We study the following policy parametrizations: deterministic regression (BCAT), KL-regularized conditional variational autoencoder (KL-CVAE), aggregate-matched conditional Wasserstein autoencoder (CWAE), vector-quantized conditional variational autoencoder (Residual VQ-VAE), conditional variational autoencoder with a learned latent flow prior (LAT-Flow), and action-space diffusion/flow-matching-based generative models (Act-Diff, Act-Flow).

BCAT The deterministic regression baseline represents the conditional policy by a Dirac distribution,

$$\pi_\theta(\cdot \mid s) = \delta_{f_\theta(s)}.$$

It is trained with

$$\mathcal{L}_{\mathrm{BC}} = \mathbb{E}_{(S,A)\sim p_{\mathcal{D}}} \left[\mathcal{L}_{\mathrm{rec}}(f_\theta(S), A)\right].$$

KL-CVAE KL-CVAE uses a continuous Gaussian posterior, a Gaussian prior, and a latent-conditioned action decoder. The posterior encoder receives the demonstrated action chunk together with the observation representation and outputs a diagonal Gaussian

$$q_\phi(\cdot \mid a, s) = \mathcal{N}(\mu_\phi(a, s), \operatorname{diag}\exp \ell_\phi(a, s)),$$

where $\ell_\phi(a, s)$ denotes the posterior log-variance. Posterior samples are obtained by reparameterization,

$$z_q = \mu_\phi(a, s) + \exp\left(\frac{1}{2}\ell_\phi(a, s)\right) \odot \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I_{\mathsf{d}_z}),$$

and decoded as

$$\hat{a} = f_\theta(z_q, s).$$

We use two prior parameterizations. The fixed-prior variant uses

$$p_\psi(\cdot \mid s) = \mathcal{N}(0, I_{\mathsf{d}_z}).$$

The learned-Gaussian variant uses an observation-conditioned diagonal Gaussian prior,

$$p_\psi(\cdot \mid s) = \mathcal{N}(\mu_\psi(s), \operatorname{diag}\exp \ell_\psi(s)),$$

where $\mu_\psi(s)$ and $\ell_\psi(s)$ are predicted from the observation representation. In practice, we observe that learning the variance and fixing the variance to I_{d_z} yields similar performance for the simulation benchmarks in Section 5.2. Thus, the reported learned-prior KL-CVAE uses a learned-mean Gaussian prior,

$$p_\psi(\cdot \mid s) = \mathcal{N}(\mu_\psi(s), I_{\mathsf{d}_z}),$$

where $\mu_\psi(s)$ is predicted from the observation representation.

The training objective is

$$\mathcal{L}_{\mathrm{KL}} = \mathbb{E}_{(S,A)\sim p_{\mathcal{D}}} \mathbb{E}_{Z_q \sim q_\phi(\cdot \mid A,S)} \left[\mathcal{L}_{\mathrm{rec}}(f_\theta(Z_q, S), A)\right] + \beta\, \mathbb{E}_{(S,A)\sim p_{\mathcal{D}}} \mathrm{KL}(q_\phi(\cdot \mid A, S) \,\|\, p_\psi(\cdot \mid S)).$$

At inference, actions are generated by sampling from the prior,

$$Z_p \sim p_\psi(\cdot \mid s), \qquad \hat{a} = f_\theta(Z_p, s).$$

Aggregate-matched latent policies MMD-CWAE and Sinkhorn-CWAE use the same latent-conditioned decoder as KL-CVAE, but replace the pointwise KL term by an aggregate posterior matching objective, following the WAE formulation [39]. For these variants, the posterior encoder is deterministic. Given a minibatch $\{(s_i, a_i)\}_{i=1}^{\mathsf{B}}$, the posterior latents are

$$z_i^q = e_\phi(a_i, s_i), \qquad i = 1, \dots, \mathsf{B}.$$

Prior latents are sampled as

$$z_i^p \sim p_\psi(\cdot \mid s_i), \qquad i = 1, \dots, \mathsf{B}.$$

The corresponding empirical posterior and prior latent measures are

$$\widehat{Q}_Z = \frac{1}{\mathsf{B}} \sum_{i=1}^{\mathsf{B}} \delta_{z_i^q}, \qquad \widehat{P}_Z = \frac{1}{\mathsf{B}} \sum_{i=1}^{\mathsf{B}} \delta_{z_i^p}.$$

For MMD-CWAE, the aggregate penalty is the squared maximum mean discrepancy

$$\mathrm{MMD}_k^2(\widehat{Q}_Z, \widehat{P}_Z) = \frac{1}{\mathsf{B}^2} \sum_{i,j=1}^{\mathsf{B}} \Big[k(z_i^q, z_j^q) + k(z_i^p, z_j^p) - 2k(z_i^q, z_j^p) \Big],$$

where k is the latent kernel. For Sinkhorn-CWAE, the aggregate penalty is the debiased Sinkhorn divergence [9]

$$\mathrm{Sink}_{\varepsilon,c}(\widehat{Q}_Z, \widehat{P}_Z) = \mathrm{OT}_{\varepsilon,c}(\widehat{Q}_Z, \widehat{P}_Z) - \frac{1}{2}\mathrm{OT}_{\varepsilon,c}(\widehat{Q}_Z, \widehat{Q}_Z) - \frac{1}{2}\mathrm{OT}_{\varepsilon,c}(\widehat{P}_Z, \widehat{P}_Z),$$

where $\mathrm{OT}_{\varepsilon,c}$ denotes entropic optimal transport with cost c and entropic regularization strength ε.

The aggregate-matched objective is

$$\mathcal{L}_{\mathrm{CWAE}} = \frac{1}{\mathsf{B}} \sum_{i=1}^{\mathsf{B}} \mathcal{L}_{\mathrm{rec}}(f_\theta(z_i^q, s_i), a_i) + \beta\, D(\widehat{Q}_Z, \widehat{P}_Z),$$

where

$$D(\widehat{Q}_Z, \widehat{P}_Z) = \mathrm{MMD}_k^2(\widehat{Q}_Z, \widehat{P}_Z)$$

for MMD-CWAE, and

$$D(\widehat{Q}_Z, \widehat{P}_Z) = \mathrm{Sink}_{\varepsilon,c}(\widehat{Q}_Z, \widehat{P}_Z)$$

for Sinkhorn-CWAE.

The fixed-prior variants use

$$p_\psi(\cdot \mid s) = \mathcal{N}(0, I_{\mathsf{d}_z}), \qquad z_i^p \sim \mathcal{N}(0, I_{\mathsf{d}_z}).$$

The learned-prior variants use an observation-conditioned prior $p_\psi(\cdot \mid s)$, so that

$$z_i^p \sim p_\psi(\cdot \mid s_i).$$

At inference, actions are generated by sampling from the prior and decoding,

$$z_p \sim p_\psi(\cdot \mid s), \qquad \hat{a} = f_\theta(z_p, s).$$

Conditional aggregate matching In the synthetic benchmark, each task is evaluated under a fixed layout, so matching the marginal latent distribution is often sufficient. In the simulation benchmarks, the observation s varies substantially across episodes and determines which action chunks are valid. Matching only the marginal latent distribution can therefore mix latent samples coming from different conditioning states. We address this by matching joint condition–latent samples.

Let

$$h_i = h_\eta(s_i)$$

be the action-free condition embedding emitted by the prior network. In the conditional matching loss, this vector is L_2-normalized and scaled by a state-weight parameter $\lambda_s \geq 0$:

$$\tilde{h}_i = \frac{h_i}{\|h_i\|_2}, \qquad \bar{h}_i = \sqrt{\lambda_s}\, \tilde{h}_i.$$

The empirical joint posterior and prior measures are

$$\widehat{Q}_{\bar{h},Z} = \frac{1}{\mathsf{B}} \sum_{i=1}^{\mathsf{B}} \delta_{(\bar{h}_i, z_i^q)}, \qquad \widehat{P}_{\bar{h},Z} = \frac{1}{\mathsf{B}} \sum_{i=1}^{\mathsf{B}} \delta_{(\bar{h}_i, z_i^p)}.$$

For conditional MMD-CWAE, we use a product kernel on condition–latent pairs,

$$k\big((\bar{h}, z), (\bar{h}', z')\big) = k_s(\bar{h}, \bar{h}')\, k_z(z, z').$$

The conditional MMD penalty is

$$\mathrm{MMD}^2_{\mathrm{cond}}(\widehat{Q}_{\bar{h},Z}, \widehat{P}_{\bar{h},Z}) = \frac{1}{\mathsf{B}^2} \sum_{i,j=1}^{\mathsf{B}} k_s(\bar{h}_i, \bar{h}_j) \Big[k_z(z_i^q, z_j^q) + k_z(z_i^p, z_j^p) - 2k_z(z_i^q, z_j^p) \Big].$$

The objective is

$$\mathcal{L}_{\mathrm{cMMD}} = \frac{1}{\mathsf{B}} \sum_{i=1}^{\mathsf{B}} \mathcal{L}_{\mathrm{rec}}(f_\theta(z_i^q, s_i), a_i) + \beta\, \mathrm{MMD}^2_{\mathrm{cond}}(\widehat{Q}_{\bar{h},Z}, \widehat{P}_{\bar{h},Z}).$$

For relaxed conditional Sinkhorn-CWAE, we perform transport in the joint condition–latent space. Define

$$x_i^q = \begin{bmatrix} \bar{h}_i \\ z_i^q \end{bmatrix}, \qquad x_i^p = \begin{bmatrix} \bar{h}_i \\ z_i^p \end{bmatrix}.$$

Since $\bar{h}_i = \sqrt{\lambda_s}\tilde{h}_i$, the squared Euclidean cost between posterior sample i and prior sample j is

$$c_{ij} = \lambda_s \|\tilde{h}_i - \tilde{h}_j\|_2^2 + \|z_i^q - z_j^p\|_2^2.$$

The conditional Sinkhorn penalty is the debiased Sinkhorn divergence

$$\mathrm{Sink}_{\varepsilon,c}(\widehat{Q}_{\bar{h},Z}, \widehat{P}_{\bar{h},Z}) = \mathrm{OT}_{\varepsilon,c}(\widehat{Q}_{\bar{h},Z}, \widehat{P}_{\bar{h},Z}) - \frac{1}{2}\mathrm{OT}_{\varepsilon,c}(\widehat{Q}_{\bar{h},Z}, \widehat{Q}_{\bar{h},Z}) - \frac{1}{2}\mathrm{OT}_{\varepsilon,c}(\widehat{P}_{\bar{h},Z}, \widehat{P}_{\bar{h},Z}),$$

where $\mathrm{OT}_{\varepsilon,c}$ denotes entropic optimal transport with cost c and entropic regularization strength ε. The objective is

$$\mathcal{L}_{\mathrm{cSink}} = \frac{1}{\mathsf{B}} \sum_{i=1}^{\mathsf{B}} \mathcal{L}_{\mathrm{rec}}(f_\theta(z_i^q, s_i), a_i) + \beta\ \mathrm{Sink}_{\varepsilon,c}(\widehat{Q}_{\bar{h},Z}, \widehat{P}_{\bar{h},Z}).$$

Posterior geometry regularization Aggregate matching constrains the batch-level distribution of posterior latents, but it does not by itself control the local geometry of the posterior codes used by the decoder. In practice, the deterministic posterior can become poorly conditioned, for example by concentrating in thin directions, drifting away from the prior scale, or developing strong correlations across latent dimensions. We therefore add a simple minibatch moment regularizer on the posterior latents, together with optional decoder-side jitter. The regularizer keeps the posterior codes approximately centered, controls their per-dimension scale, caps excessive spread, and penalizes off-diagonal covariance.

For aggregate-matched variants, the posterior encoder is deterministic, so each demonstration pair produces a latent

$$z_i^q = e_\phi(a_i, s_i).$$

Let

$$\bar{z}^q = \frac{1}{\mathsf{B}} \sum_{i=1}^{\mathsf{B}} z_i^q, \qquad \tilde{z}_i^q = z_i^q - \bar{z}^q.$$

For each latent dimension d, define the minibatch standard deviation

$$\hat{\sigma}_d = \left(\frac{1}{\mathsf{B}} \sum_{i=1}^{\mathsf{B}} (\tilde{z}_{i,d}^q)^2 + \varepsilon_{\mathrm{num}} \right)^{1/2},$$

and define the empirical covariance

$$\widehat{C} = \frac{1}{\mathsf{B}-1}\sum_{i=1}^{\mathsf{B}} \tilde{z}_i^q (z_i^q)^\top.$$

The posterior geometry regularizer is

$$\mathcal{L}_{\text{geom}} = \beta_{\text{mean}}\|\bar{z}^q\|_2^2 + \beta_{\text{std}}\frac{1}{\mathsf{d}_z}\sum_{d=1}^{\mathsf{d}_z}(\hat{\sigma}_d - \sigma_\star)^2 + \beta_{\max}\frac{1}{\mathsf{d}_z}\sum_{d=1}^{\mathsf{d}_z}[\hat{\sigma}_d - \sigma_{\max}]_+^2 + \beta_{\text{cov}}\frac{1}{\mathsf{d}_z}\left\|\text{offdiag}(\widehat{C})\right\|_F^2.$$

Here $\sigma_\star$ is the target per-dimension standard deviation, $\sigma_{\max}$ is the maximum allowed standard deviation, and $\varepsilon_{\text{num}} > 0$ is a numerical stabilizer. When this regularizer is enabled, it is added to the corresponding aggregate-matched objective:

$$\mathcal{L} = \mathcal{L}_{\text{CWAE}} + \mathcal{L}_{\text{geom}}.$$

We also optionally apply decoder-side latent jitter during training. The latent used for aggregate matching remains z_i^q, while the latent consumed by the decoder is

$$z_i^{\text{dec}} = z_i^q + \sigma_{\text{dec}}\epsilon_i, \qquad \epsilon_i \sim \mathcal{N}(0, I_{\mathsf{d}_z}).$$

Thus, the aggregate posterior measure is still constructed from $\{z_i^q\}_{i=1}^{\mathsf{B}}$, while reconstruction is computed with $f_\theta(z_i^{\text{dec}}, s_i)$.

Latent flow prior (LAT-Flow) LAT-Flow uses a deterministic posterior encoder and a latent-conditioned decoder,

$$z_i^q = e_\phi(a_i, s_i), \qquad \hat{a}_i = f_\theta(z_i^{\text{dec}}, s_i).$$

The prior $p_\psi(\cdot \mid s)$ is parameterized by a conditional flow-matching model in latent space. During training, sample

$$u_i \sim \mathcal{N}(0, I_{\mathsf{d}_z}), \qquad t_i \sim p(t),$$

and construct the interpolation

$$z_{t_i} = (1-t_i)u_i + t_i z_i^q.$$

For the linear path, the target velocity is

$$v_i^\star = z_i^q - u_i.$$

The flow prior predicts

$$v_\psi(z_{t_i}, t_i, s_i),$$

and is trained with

$$\mathcal{L}_{\text{flow}} = \frac{1}{\mathsf{B}}\sum_{i=1}^{\mathsf{B}} \|v_\psi(z_{t_i}, t_i, s_i) - v_i^\star\|_2^2.$$

The full training objective is

$$\mathcal{L}_{\text{LAT}} = \frac{1}{\mathsf{B}}\sum_{i=1}^{\mathsf{B}} \mathcal{L}_{\text{rec}}\big(f_\theta(z_i^{\text{dec}}, s_i), a_i\big) + \beta\mathcal{L}_{\text{flow}}.$$

When enabled, the posterior geometry regularizer $\mathcal{L}_{\text{geom}}$ is added to this objective.

At inference, sample

$$z_0 \sim \mathcal{N}(0, I_{\mathsf{d}_z})$$

and integrate

$$\frac{dz_t}{dt} = v_\psi(z_t, t, s)$$

from $t = 0$ to $t = 1$. The resulting latent prior sample z_1 is decoded as

$$\hat{a} = f_\theta(z_1, s).$$

Residual VQ-VAE VQ-VAE uses the same latent-conditioned action decoder as the continuous latent policies, but replaces the continuous posterior sample by a residual vector-quantized latent. Given a minibatch $\{(s_i, a_i)\}_{i=1}^{\mathsf{B}}$, the posterior encoder first produces continuous latents

$$z_{e,i} = e_\phi(a_i, s_i) \in \mathbb{R}^{\mathsf{d}_z}, \qquad i = 1, \dots, \mathsf{B}.$$

A residual vector quantizer with L_{vq} layers and C_{vq} codes per layer maps each $z_{e,i}$ to a quantized latent $z_{q,i}$. Let $\mathcal{E}_\ell = \{e_{\ell,1}, \dots, e_{\ell,\mathsf{C}_{\mathrm{vq}}}\}$ denote the codebook of layer ℓ. Starting with residual $r_{i,1} = z_{e,i}$, each layer selects the nearest code

$$c_{i,\ell} = \arg\min_{j \in \{1,\dots,\mathsf{C}_{\mathrm{vq}}\}} \|r_{i,\ell} - e_{\ell,j}\|_2^2, \qquad q_{i,\ell} = e_{\ell,c_{i,\ell}},$$

and updates the residual as

$$r_{i,\ell+1} = r_{i,\ell} - \mathrm{sg}(q_{i,\ell}),$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient. The final quantized latent is the sum of residual code contributions,

$$z_{q,i} = \sum_{\ell=1}^{\mathsf{L}_{\mathrm{vq}}} q_{i,\ell},$$

with straight-through gradients used through the quantization operation.

The action decoder receives $z_{q,i}$ and predicts

$$\hat{a}_i = f_\theta(z_{q,i}, s_i).$$

The minibatch training objective is

$$\mathcal{L}_{\mathrm{VQ}} = \frac{1}{\mathsf{B}} \sum_{i=1}^{\mathsf{B}} \mathcal{L}_{\mathrm{rec}}(f_\theta(z_{q,i}, s_i), a_i) + \beta_{\mathrm{vq}} \frac{1}{\mathsf{B}\,\mathsf{L}_{\mathrm{vq}}} \sum_{i=1}^{\mathsf{B}} \sum_{\ell=1}^{\mathsf{L}_{\mathrm{vq}}} \|r_{i,\ell} - \mathrm{sg}(q_{i,\ell})\|_2^2.$$

The codebooks are updated by exponential moving averages of assigned encoder outputs, with first-batch initialization and dead-code replacement, following vector-quantized autoencoders [40, 19].

At inference, we use a fixed uniform prior over codebook indices. For each residual layer,

$$c_\ell^p \sim \mathrm{Unif}\{1, \dots, \mathsf{C}_{\mathrm{vq}}\}, \qquad \ell = 1, \dots, \mathsf{L}_{\mathrm{vq}},$$

and reconstruct the prior latent by decoding the sampled code indices,

$$z_p = \sum_{\ell=1}^{\mathsf{L}_{\mathrm{vq}}} e_{\ell,c_\ell^p}.$$

The predicted action chunk is then

$$\hat{a} = f_\theta(z_p, s).$$

Action-space flow and diffusion policies Act-Flow and Act-Diff model the action chunk directly in $\mathcal{A}^{\mathsf{H}}$, without an explicit posterior latent variable. For a fixed observation s, both methods define a deterministic sampler

$$G_{\theta,s}(u) := G_\theta(s, u),$$

which pushes forward a simple Gaussian base distribution $p_0 = \mathcal{N}(0, I)$ on the flattened action-chunk space to the conditional action chunk distribution:

$$\pi_\theta(\cdot \mid s) = (G_{\theta,s})_{\#} p_0.$$

Equivalently, sampling from the policy is written as

$$U \sim p_0, \qquad A = G_{\theta,s}(U).$$

For Act-Flow, $G_{\theta,s}$ is the ODE solution map induced by the learned velocity field. For Act-Diff, $G_{\theta,s}$ is the full composition of deterministic DDIM denoising steps. Both variants use the same conditional action transformer backbone: noisy or interpolated action tokens attend jointly with observation tokens, and the network is conditioned on a timestep embedding.

For Act-Flow, we train a conditional flow-matching model in action space. Given a minibatch $\{(s_i, a_i)\}_{i=1}^{\mathsf{B}}$, sample

$$u_i \sim \mathcal{N}(0, I), \qquad t_i \sim p(t),$$

and form the linear interpolation

$$a_{t_i} = (1 - t_i)u_i + t_i a_i.$$

For this path, the target velocity is

$$v_i^\star = a_i - u_i.$$

The model predicts a velocity field

$$v_\theta(a_{t_i}, t_i, s_i),$$

and is trained with

$$\mathcal{L}_{\text{ActFlow}} = \frac{1}{\mathsf{B}} \sum_{i=1}^{\mathsf{B}} \|v_\theta(a_{t_i}, t_i, s_i) - v_i^\star\|_2^2 .$$

At inference, we sample $a_0 \sim \mathcal{N}(0, I)$ and integrate

$$\frac{da_t}{dt} = v_\theta(a_t, t, s)$$

from $t = 0$ to $t = 1$. The resulting a_1 is used as the predicted action chunk.

For Act-Diff, we train a denoising diffusion model in action space. Let $\bar{\alpha}_t$ denote the cumulative noise schedule. Given a minibatch $\{(s_i, a_i)\}_{i=1}^{\mathsf{B}}$, sample a timestep t_i and noise $\epsilon_i \sim \mathcal{N}(0, I)$, and construct

$$a_{t_i} = \sqrt{\bar{\alpha}_{t_i}}\, a_i + \sqrt{1 - \bar{\alpha}_{t_i}}\, \epsilon_i.$$

The model predicts the clean action chunk directly,

$$\hat{a}_i = g_\theta(a_{t_i}, t_i, s_i),$$

and is trained with the sample-prediction objective

$$\mathcal{L}_{\text{ActDiff}} = \frac{1}{\mathsf{B}} \sum_{i=1}^{\mathsf{B}} \|g_\theta(a_{t_i}, t_i, s_i) - a_i\|_2^2 .$$

At inference, we initialize $a_{\mathsf{T}_{\text{diff}}} \sim \mathcal{N}(0, I)$ and apply the deterministic DDIM sampler to obtain the final action chunk. The full composition of denoising steps defines the sampling map $G_{\theta,s} : u \mapsto a$ used in Section 4.

C.2 Network Architectures

This section describes the neural implementations, the benchmarks, and the training and evaluation details of the experiments. All policies predict action chunks non-autoregressively. Within each benchmark, methods share the same observation encoder, action representation, normalization pipeline, and training loop, and differ only in the output parameterization.

Observation encoders. Image observations are encoded with a convolutional encoder, initialized from pretrained weights and fine-tuned during policy training. The encoder preserves the spatial feature map, uses no global pooling, and uses frozen batch-normalization statistics. Proprioceptive and state observations are encoded with a single projection layer.

Action decoder. The deterministic and latent-variable policies use non-autoregressive action transformer decoders. Encoded observations are first converted into a token sequence. Spatial feature maps receive two-dimensional sinusoidal positional encodings, while flat features receive one-dimensional positional encodings. Let

$$Q \in \mathbb{R}^{\mathsf{H} \times \mathsf{d}}$$

denote the learnable action-query sequence, with one query embedding per predicted action timestep. Given the observation-token sequence, the decoder maps Q to H action embeddings in parallel, and task-specific action heads map these embeddings to the predicted action components.

The deterministic baseline uses the non-latent version of this decoder. For an observation s, the action transformer produces action embeddings

$$U_{1:\mathsf{H}} = D_\theta(Q, X_s),$$

where $X_s \in \mathbb{R}^{\mathsf{N}_s \times \mathsf{d}}$ denotes the encoded observation tokens. The action heads then produce a single predicted chunk,

$$\hat{a} = \mathrm{Head}_\theta(U_{1:\mathsf{H}}) = f_\theta(s).$$

LACT. Latent-variable policies use LACT, a latent-conditioned variant of the previous action decoder. Given a latent z, LACT conditions each decoder block through AdaLN-Zero-style modulation, as used in DiT-style transformers. The conditioning network maps z to scale, shift, and residual-gating parameters, which modulate the normalization and residual branches of the self-attention, cross-attention, and feedforward blocks. Thus, the latent affects the full action-query decoding process, and the decoder implements the map

$$\hat{a} = f_\theta(z, s).$$

MMDiT action decoder. Action-space flow and diffusion policies use an MMDiT-style dual-stream transformer [8]. The model receives observation tokens and action tokens as separate streams. The action stream contains either interpolated actions for flow matching or noisy actions for diffusion. A timestep embedding conditions the transformer layers through adaptive normalization. The dual-stream decoder allows bidirectional attention between observation and action tokens, and a final prediction layer maps the action-token outputs to either velocities for Act-Flow or denoising targets for Act-Diff.

For Act-Flow, the MMDiT predicts the action-space velocity field used by the flow solver. For Act-Diff, the MMDiT predicts the diffusion training target and is sampled with a deterministic DDIM scheduler.

Latent encoders. Latent-variable policies use transformer encoders to parameterize the posterior and, when used, the learned prior. The posterior encoder receives a sequence of tokens containing the ground-truth action sequence together with the encoded observation tokens. A learnable class token is appended to the sequence, and a transformer encoder maps the final class-token representation to the latent output.

For KL-CVAE, the posterior projection has output dimension $2\mathsf{d}_z$ and is split into a mean and log-variance,

$$(\mu_\phi(a, s), \ell_\phi(a, s)) = P_\phi(\mathrm{CLS}_\phi(a, s)),$$

which define the diagonal Gaussian posterior $q_\phi(\cdot \mid a, s)$. For aggregate-matched variants and LAT-Flow, the projection has output dimension d_z and directly returns

$$z^q = e_\phi(a, s).$$

The VQ-VAE posterior uses the same action-plus-observation transformer encoder, but replaces the continuous latent output by a residual vector-quantized bottleneck. The class-token representation is first projected to a continuous vector

$$z^{\mathrm{cont}} = P_\phi(\mathrm{CLS}_\phi(a, s)),$$

which is then quantized by a residual codebook stack. The quantized latent z^q is passed to the action decoder.

Learned priors use a separate transformer encoder that receives only observation tokens. A class token is appended to the observation sequence, and the final class-token representation is projected to prior parameters,

$$(\mu_\psi(s), \ell_\psi(s)) = P_\psi(\mathrm{CLS}_\psi(s)),$$

defining

$$p_\psi(\cdot \mid s) = \mathcal{N}(\mu_\psi(s), \mathrm{diag}\exp \ell_\psi(s)).$$

For conditional aggregate matching, the same prior encoder also emits an action-free condition vector $h_\eta(s)$. This vector is obtained by pooling the observation tokens and used to compute the conditional MMD/Sinkhorn losses.

C.3 Benchmarks

All benchmarks use observation horizon $\mathsf{T}_{\text{obs}} = 1$. Thus, each policy conditions on a single current observation, without observation history. The action horizon depends on the benchmark. Synthetic tasks use $\mathsf{H} = 60$, corresponding to the full task trajectory. Simulation and real-world robot benchmarks use $\mathsf{H} = 10$; at evaluation time, we execute the first two actions from each predicted chunk before replanning.

Synthetic benchmark. The synthetic benchmark consists of four 2D navigation tasks. All trajectories lie in the unit square $[0, 1]^2$. Each task contains 1000 demonstrations. Policies receive a single 64×64 RGB rendering at $t = 0$, showing the agent initial position, obstacles, and goal regions, and predict the complete trajectory chunk $a_{0:\mathsf{H}-1} \in \mathbb{R}^{60\times 2}$, with $\mathsf{H} = 60$, without replanning. Thus, each trajectory effectively contains one single training sample pair. The four tasks differ in their mode structure. *Circle* has $\mathsf{K} = 2$ circular trajectory modes around obstacles. *Sequential* has two consecutive binary decisions, yielding $\mathsf{K} = 4$ modes separated by three obstacles. *Radial* has $\mathsf{K} = 16$ modes from a common start to sixteen evenly spaced radial goals, with obstacles between neighboring directions. *Corridor* has $\mathsf{K} = 16$ valid routes from a common start to a common endpoint through sixteen corridors induced by fifteen obstacles. For synthetic evaluation, a generated trajectory is successful if it satisfies all three criteria: it does not collide with obstacles; its endpoint is within distance 0.1 of one valid mode endpoint; and its path length is at least 0.8 times the mean expert path length. The path-length condition rules out trivial collapsed trajectories that reach an endpoint without following a valid path. We also report valid mode coverage and valid mode entropy ratio, computed only over successful trajectories.

Simulation benchmarks. We use PushT, Franka Kitchen, and UR3 BlockPush as robotic simulation benchmarks previously employed in the literature of multimodal policy [10, 7, 30, 19, 31], together with the language-conditioned benchmarks LIBERO and Meta-World, extensively used in the vision-language-action model benchmarking, see for example [2, 32]. PushT requires pushing a T-shaped block to a target pose using 2D end-effector control [7]. We use both image and state variants. The image observation is a single 96×96 RGB agent-view image. The state observation contains the end-effector position, block position, block angle, block keypoints, and contact indicator. PushT performance is measured by final intersection-over-union (IoU) between the block pose and target pose. Franka Kitchen [11] is a long-horizon manipulation environment in which a Franka arm completes a subset of kitchen subtasks in sequence. We use state and image variants. The state observation contains 9 robot-arm configuration variables and 21 object configuration variables. The image observation is a single 112×112 RGB rendering. Performance is reported as the number of completed goals out of four. UR3 BlockPush requires moving two blocks to two target regions. The task is multimodal because the blocks can be pushed in different orders. We use the unconditional state variant. The state contains the 2D end-effector position and the 2D positions of both blocks; the action is a 2D end-effector target. Performance is reported as the number of achieved goals out of two. LIBERO [21] is a language-specified robotic manipulation benchmark. We train one policy on four suites with ten tasks each: LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long. The first three suites vary spatial relationships, manipulated objects, and task goals, respectively, while LIBERO-Long emphasizes longer-horizon manipulation. Meta-World [43] comprises 50 robotic manipulation tasks performed by a simulated Sawyer arm. Its four-dimensional action space consists of three-dimensional end-effector displacements and a gripper command. We train on the MT50 multi-task suite and report results for the Easy, Medium, Hard, and Very Hard difficulty groups, together with aggregate performance across all evaluated tasks.

Real-world benchmark: bimodal tissue grasping To validate the policies on a real-world robotic platform, we design the surgical task of bimodal tissue grasping. During minimally invasive surgery for gastrointestinal cancer, surgical assistants commonly manipulate and retract the small bowel to expose the surgical field for procedures such as anastomosis and tumor resection [17]. In the bimodal bowel tissue grasping task, a segment of bowel tissue must be grasped and displaced while avoiding a sensitive region of interest, representing, for example, a tumor or another area where grasping could cause tissue trauma. Two separate safe grasping regions are available and located on opposite sides of the region of interest. Both grasps are equally valid, but they correspond to distinct trajectory strategies, for example, reflecting left-side versus right-side approach preferences. The resulting conditional action-chunk distribution is therefore bimodal. We collect 90 demonstrations,

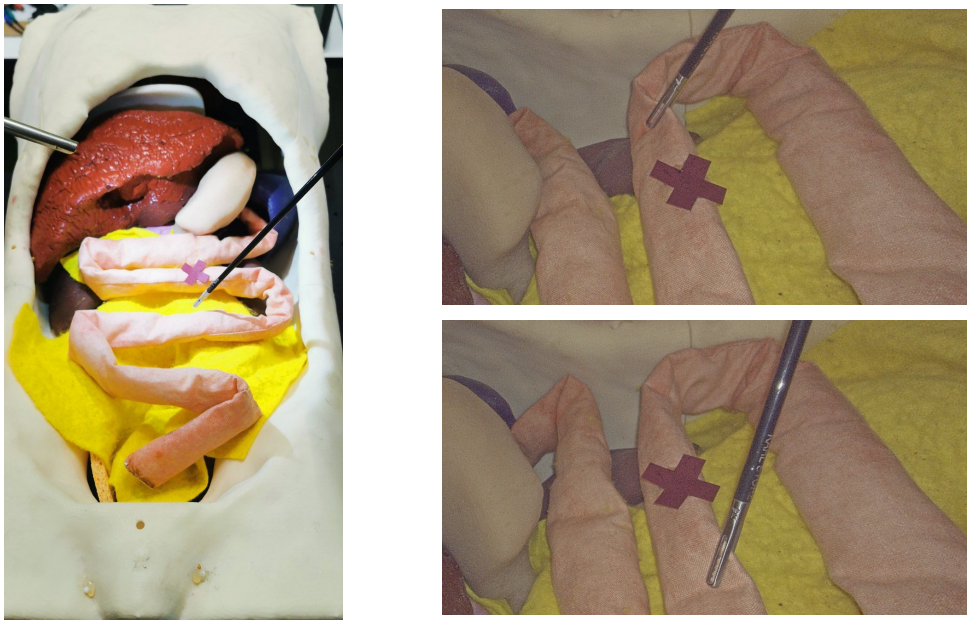


Figure 5: Real-world surgical robot task: bimodal tissue grasping. The robotic instrument needs to grasp the bowel tissue while avoiding a sensitive region of interest marked with a red cross.

Table 3: Benchmark summary. 'L' stands for language instruction.

Benchmark	Demonstrations	Observation	Horizon H	Evaluation
Synthetic	1000/task	RGB 64×64	60	success, coverage, entropy
PushT-State	206	state, 24-d	10	final IoU
PushT-Image	206	RGB 96×96	10	final IoU
Kitchen-State	566	state, 30-d	10	goals achieved out of 4
Kitchen-Image	566	RGB 112×112	10	goals achieved out of 4
UR3 BlockPush	600	state, 6-d	10	goals achieved out of 2
LIBERO	1693	2xRGB 256×256, L	10	success
Meta-World	2500	RGB 480×480, L	10	success
Bimodal Tissue Grasping	90	state, 4-d	10	success rate

balanced across the two modes with 45 demonstrations per mode, using an open-source surgical robotics platform [27] and the Open-HELP abdominal phantom [15]. A picture of the setup can be seen in Figure 5. The observation space comprises the current Cartesian 3D position of the surgical instrument tip and the binary gripper state; the action space is the delta 3D movement of the tip and the binary gripper action.

C.4 Training Details

All runs used 32-bit precision. All policies in a given benchmark share the same observation representation, normalization, action horizon, data loader, optimizer, and training loop. The network architectures are described in Appendix C.2. The full hyperparameter tables are reported in Appendix E. Each experiment was executed as a single-GPU job on an NVIDIA H100 94GB GPU. The available memory was not fully utilized for smaller models; in particular, synthetic runs required less than 4GB of VRAM. Synthetic experiments used 8 CPU workers per job and required approximately 20–40 minutes of wall-clock time. Simulation and robot experiments used 14 CPU workers per job and required approximately 10–20 hours of wall-clock time, depending on the benchmark and model family.

C.5 Evaluation Details

Policies are evaluated by rolling them out in the benchmark environments on which they were trained. For the synthetic benchmarks, we perform 200 rollouts per policy every 100 epochs and report each metric as the average over the last five evaluation checkpoints, corresponding to $5 \times 200 = 1000$ rollouts. For the simulation benchmarks, we evaluate each policy on 200 rollout trials using a reproducible random seed that is varied per each roll-out, in order to sample reproducibly randomized initial conditions. We repeat the evaluation over three seeds. We evaluate policies at fixed and latest checkpoint intervals. For the real-robot experiment, we evaluate each policy for 20 trials.

D Additional Results

D.1 Complete Synthetic Benchmarks Results

Table 4 reports the complete per-task synthetic results. Valid mode coverage is the fraction of ground-truth modes reached by at least one successful rollout, so invalid or colliding trajectories do not increase coverage. The valid mode-entropy ratio (MER) is the entropy of the empirical distribution $\widehat{p}_{\text{valid}}$ over successful mode labels M, normalized by the maximum entropy $\log \mathsf{K}$:

$$\text{MER} = \frac{H(\widehat{p}_{\text{valid}})}{\log \mathsf{K}}.$$

When no rollout is successful, both coverage and entropy ratio are set to zero.

Table 4: Full synthetic results: **success rate (SR) / valid mode coverage (Cov) / valid mode-entropy ratio (MER)**. Values are 3-seed means, each run summarized by the running mean over the last five rollout checkpoints, with 200 episodes per checkpoint. Bold indicates the best value per task and metric.

Method	Circle	Sequential	Corridor	Radial
BCAT (deterministic)	0.00/0.00/0.00	0.00/0.00/0.00	0.00/0.00/0.00	0.00/0.00/0.00
KL-CVAE	0.96/**1.00**/0.99	0.94/**1.00**/**0.99**	0.47/0.94/0.92	0.48/**1.00**/0.95
MMD-CWAE	0.92/**1.00**/**1.00**	0.94/**1.00**/**0.99**	0.40/0.98/0.94	0.34/**1.00**/**0.98**
Sinkhorn-CWAE	0.88/**1.00**/0.98	0.93/**1.00**/**0.99**	0.43/**0.99**/0.95	0.47/**1.00**/0.96
VQ-VAE	0.57/0.60/0.59	0.67/0.67/0.67	**0.52**/0.94/0.93	**0.52**/0.99/0.95
Act-Flow	0.94/**1.00**/0.99	0.91/**1.00**/0.97	0.44/0.98/**0.96**	0.49/0.99/0.96
Act-Diff	**1.00**/**1.00**/0.99	0.89/**1.00**/0.94	0.41/0.95/0.93	0.44/0.97/0.93
LAT-Flow	**1.00**/**1.00**/**1.00**	**0.96**/**1.00**/**0.99**	0.48/**0.99**/**0.96**	0.46/0.99/0.96

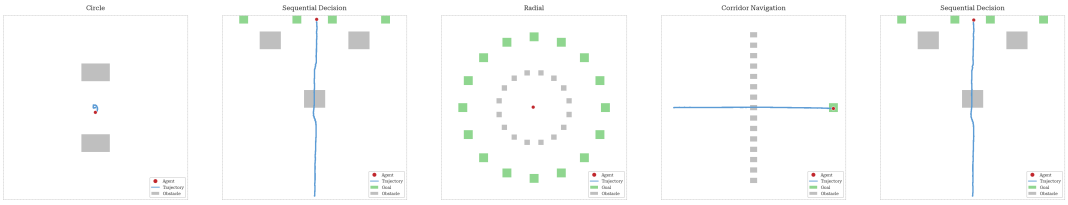
D.2 Collapse Diagnostics for Baseline Policies

Deterministic behavioral cloning exhibits complete collapse on all four synthetic benchmarks. As expected from the unimodal regression discussion in Section 1, the policy predicts an averaged trajectory between valid modes, and consequently fails to produce a single successful rollout on any task. Figure 6 visualizes representative collapsed predictions for the four synthetic tasks. The VQ-VAE baseline was considerably less stable during training. While it achieved some of the strongest success rates on the more challenging $\mathsf{K} = 16$ tasks, it also collapsed completely on the easier Sequential task ($\mathsf{K} = 4$), indicating high sensitivity to optimization dynamics and codebook usage. A representative failure run is shown in Figure 6. In preliminary PushT experiments, the same policy also performed poorly, reaching a score below 0.05 final IoU. We therefore did not pursue a broader simulation evaluation for this baseline. A plausible explanation is training instability in the single-stage setup. More structured two-stage training procedures may be beneficial here, as also adopted in prior vector-quantized imitation learning methods such as VQ-BeT [19], but we leave further investigation to future work.

D.3 KL-Regularized Latent Policies

Mode collapse under too strong regularization. Figure 7 provides a qualitative view of the mechanism quantified in Figure 2. We visualize posterior latent samples $Z \sim q_\phi(\cdot \mid A, S)$ on the synthetic benchmark *Sequential* using PCA, with points colored by the ground-truth mode label. At $\beta = 0.01$, posterior latents form mode-separated clusters, indicating that Z retains information about which demonstrated branch is being reconstructed. At $\beta = 0.1$, the posterior latents collapse into an overlapping cloud, so the latent no longer separates the four modes. This visualizes the loss of $I(M; Z \mid S)$ under stronger pointwise KL regularization.

Heuristic for choosing the value for β. We present here the extended description of the heuristic and the results on the training sweep over the value of β. Let us choose a target posterior same-mode decoding error of $\bar{\rho} = 0.1$, corresponding to at least 90% same-mode reconstruction. The required



(a) BCAT collapse on Circle. (b) BCAT collapse on Sequential. (c) BCAT collapse on Radial. (d) BCAT collapse on Corridor. (e) VQ-VAE collapse on Sequential.

Figure 6: Collapse diagnostics for baseline policies. The first four panels show representative predictions from deterministic BCAT on the four synthetic benchmarks. In all cases, the policy collapses to an averaged trajectory between valid modes and yields no successful rollouts. The fifth panel shows a representative VQ-VAE collapsed run on the Sequential task, illustrating the training instability observed for this baseline.

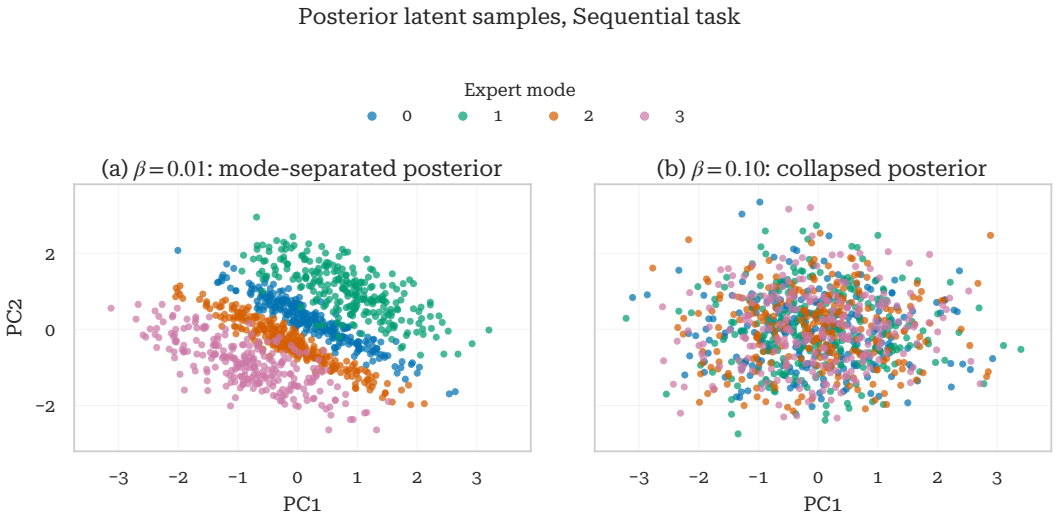


Figure 7: **Posterior latent geometry under pointwise KL pressure.** PCA projection of posterior latent samples $Z \sim q_\phi(\cdot \mid A, S)$ on the synthetic benchmark *Sequential* task, colored by ground-truth mode label. At $\beta = 0.01$, the posterior separates according to the demonstrated mode. At $\beta = 0.1$, the posterior collapses into an overlapping cloud, consistent with the loss of $I(M; Z \mid S)$ in Figure 2.

information level is

$$\widehat{b}_0(\bar{\rho}) = \frac{1}{Q} \sum_{i:\widehat{K}_i > 1} \left[\widehat{H}_i - H_b(\bar{\rho}) - \bar{\rho} \log(\widehat{K}_i - 1) \right]. \tag{19}$$

We next construct an explicit collapsed comparator within the same CVAE decoder family by setting the posterior equal to the prior and disabling the latent conditioning of the decoder. Its pointwise KL is therefore zero by construction. We denote its reconstruction loss by C. We obtain $C = 0.459$ for Circle and $C = 0.347$ for Sequential. Because both dataset modes are exactly balanced, $b_0 = 0.368$ for Circle (K=2), $b_0 = 0.951$ for Sequential (K=4). For any actual trained checkpoint satisfying $L_\beta \le C$, we can directly verify $B_{\hat{\rho}} \le I(M; Z \mid S) \le I(A; Z \mid S) \le D_{\mathrm{KL}} \le \frac{L_\beta}{\beta} \le \frac{C}{\beta}$. It follows from the corollary that achieving $\hat{\rho} \le \bar{\rho}$ is impossible when $\beta > \beta_{\max}, \beta_{\max} := \frac{C}{b_0}$. This gives $\beta_{\max} = 1.25$ for Circle, and $\beta_{\max} = 0.365$ for Sequential.

As mentioned before, this is a necessary condition: values below it are permitted, but are not guaranteed to work. To obtain a suitable value below this threshold, we use one low-β calibration run that correctly retains the mode information. Let its reconstruction loss and pointwise KL be R_p and D_p. Holding this observed solution fixed, its objective remains below the collapsed comparator while $R_p + \beta D_p < C$. This gives the estimate $\beta^* = \frac{C - R_p}{D_p}$. We then use $\beta_{heu} = \beta^*/2$ as a safety margin.

Table 5: Full β-sweep on Circle (K=2, $\beta_{heu} = 0.125$) and Sequential (K=4, $\beta_{heu} = 0.0138$). Success is computed over the last five evaluation epochs ($\pm$ s.d., n=5), measured with 200 rollouts per evaluation.

(a) Circle

β	$\beta/\beta_{\max}$	D_{KL}	$I(M;Z \mid S)$	$\hat{\rho}$	$B_{\hat{\rho}}$	L_β	$L_\beta \le C$	$\hat{\rho} \le 0.1$	Success
0.001	0.00	2.627	0.693	0.000	0.693	0.2658	✓	✓	0.766 ± 0.054
0.01	0.01	1.246	0.691	0.001	0.689	0.2893	✓	✓	0.965 ± 0.015
0.1	0.08	0.667	0.566	0.050	0.496	0.3589	✓	✓	0.901 ± 0.019
0.3117	0.25	0.110	0.108	0.308	0.075	0.4489	✓		0.174 ± 0.214
0.6234	0.50	0.000	0.000	0.502	0.000	0.4608			0.000 ± 0.000
1	0.80	0.000	0.000	0.502	0.000	0.4584	✓		0.000 ± 0.000
1.2468	1.00	0.000	0.000	0.504	0.000	0.4596			0.000 ± 0.000
2.4935	2.00	0.000	0.000	0.495	0.000	0.4591			0.000 ± 0.000
4.987	4.00	0.000	0.000	0.499	0.000	0.4591			0.000 ± 0.000

(b) Sequential

β	$\beta/\beta_{\max}$	D_{KL}	$I(M;Z \mid S)$	$\hat{\rho}$	$B_{\hat{\rho}}$	L_β	$L_\beta \le C$	$\hat{\rho} \le 0.1$	Success
0.001	0.00	6.321	1.386	0.000	1.385	0.2598	✓	✓	0.918 ± 0.018
0.01	0.03	1.437	1.268	0.039	1.179	0.3218	✓	✓	0.938 ± 0.024
0.02	0.05	0.890	0.856	0.217	0.626	0.3385	✓		0.728 ± 0.095
0.04	0.11	0.000	0.000	0.749	0.000	0.3477			0.000 ± 0.000
0.0912	0.25	0.000	0.000	0.751	0.000	0.3481			0.000 ± 0.000
0.1	0.27	0.000	0.000	0.747	0.000	0.3471			0.000 ± 0.000
0.1824	0.50	0.000	0.000	0.747	0.000	0.3480			0.000 ± 0.000
0.3648	1.00	0.000	0.000	0.752	0.000	0.3481			0.000 ± 0.000
0.7296	2.00	0.000	0.000	0.749	0.000	0.3481			0.000 ± 0.000
1	2.74	0.000	0.000	0.748	0.000	0.3471			0.000 ± 0.000
1.4592	4.00	0.000	0.000	0.748	0.000	0.3480			0.000 ± 0.000

For Circle, the calibration run at $\beta_0 = 0.1$ has $R_p = 0.3589 - 0.1(0.667) = 0.2922, \beta^* = \frac{0.459-0.2922}{0.667} = 0.250$. The resulting recommendation is $\beta_{heu} = 0.125$. In the sweep, the target $\bar{\rho}$ is met at $\beta = 0.1$ and missed at $\beta = 0.3117$.

For Sequential, the calibration run at $\beta_0 = 0.01$ has $R_p = 0.3218 - 0.01(1.437) = 0.3074, \beta^* = \frac{0.347-0.3074}{1.437} = 0.0275$. The resulting recommendation is $\beta_{heu} = 0.0138$. The target $\bar{\rho}$ is met at $\beta = 0.01$, missed at $\beta = 0.02$, and complete latent collapse occurs at $\beta = 0.04$.

Additionally, we note that the pointwise KL value provides an early training-time failure diagnostic:

$$D_{\mathrm{KL}} < b_0(\bar{\rho}) \quad \Longrightarrow \quad \hat{\rho} > \bar{\rho}.$$

Therefore, if during training, the KL stabilizes below b_0, the current run cannot meet the specified mode-recovery target and β should be reduced.

Table 5 contains 20 runs across Circle and Sequential. In every run, the empirical inequality $B_{\hat{\rho}} \le D_{\mathrm{KL}} \le C/\beta$ holds. Every run achieving $\hat{\rho} \le 0.1$ has $D_{\mathrm{KL}} \ge b_0$, every run with $D_{\mathrm{KL}} < b_0$ misses the target, and no run above $\beta_{\max}$ meets it. Several runs below $\beta_{\max}$ also fail, confirming that the theoretical threshold is necessary rather than sufficient.

D.4 Aggregate Matching and Prior Coverage

As discussed in Appendix B.2, the critical-β collapse criterion is specific to pointwise KL and does not apply to aggregate matching. For aggregate-matched objectives, the relevant deployment question is whether prior samples cover the decoder-valid latent regions learned from posterior samples.

Posterior–prior valid-mass gap. We test the deployment caveat from Appendix B.2: aggregate matching can preserve mode information in the posterior, but successful inference also requires the prior to place mass on decoder-valid latent regions. We therefore compare posterior-sampled and prior-sampled valid mass. Posterior evaluation samples $Z \sim q_\phi(\cdot \mid A, S)$ after drawing $(S, A) \sim p_{\mathcal{D}}$,

Table 6: Posterior–prior valid-mass gap on synthetic tasks. We report macro averages across all four synthetic tasks and averages on the two hardest $\mathsf{K} = 16$ tasks. Posterior evaluation samples $Z \sim q_\phi(\cdot \mid A, S)$ after drawing $(S, A) \sim p_{\mathcal{D}}$; prior evaluation samples $Z \sim p_\psi(\cdot \mid S)$, as at deployment. A larger Δ_{valid} indicates that posterior latents decode to valid trajectories more often than prior samples.

Method	Post. valid	Prior valid	Avg. Δ_{valid}	Hard Δ_{valid}
KL-CVAE	0.65	0.61	**0.03**	**0.03**
MMD-CWAE	**0.72**	0.57	0.14	0.21
Sinkhorn-CWAE	0.69	0.57	0.12	0.14
LAT-Flow	0.70	**0.64**	0.06	0.09

Table 8: Conditional aggregate-matching ablation for Sinkhorn-CWAE on PushT Image. Posterior L_1 loss evaluates chunks decoded from $Z \sim q_\phi(\cdot \mid A, S)$; prior-sample L_1 loss evaluates chunks decoded from $Z \sim p_\psi(\cdot \mid S)$, as at deployment. We report $\Delta_{\mathrm{L}_1} = \mathrm{L}_{1\text{prior}} - \mathrm{L}_{1\text{post}}$.

OT variant	Final IoU ↑	Post. L_1 ↓	Prior L_1 ↓	Δ_{L_1} ↓
Fixed Gaussian	0.0004	1.9829	38.8889	36.9060
Learned prior	0.4385	2.0476	**16.8767**	**14.8290**
Conditional learned prior	**0.7517**	2.0422	20.1505	18.1084

whereas prior evaluation samples $Z \sim p_\psi(\cdot \mid S)$, as at deployment. Since each synthetic action chunk is a complete trajectory, valid mass is the probability that the decoded trajectory satisfies the synthetic success criterion. We report

$$\Delta_{\text{valid}} = \mathbb{E}_{(S,A)\sim p_{\mathcal{D}}} \Pr_{Z\sim q_\phi(\cdot|A,S)} \left[f_\theta(Z, S) \in \bigcup_{k=1}^{\mathsf{K}(S)} C_k(S) \right] - \mathbb{E}_{S\sim p_{\mathcal{D},S}} \Pr_{Z\sim p_\psi(\cdot|S)} \left[f_\theta(Z, S) \in \bigcup_{k=1}^{\mathsf{K}(S)} C_k(S) \right].$$

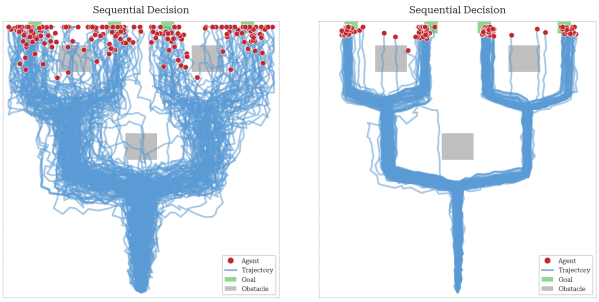
Table 6 shows that aggregate matching changes the failure mode rather than removing it. KL-CVAE has the smallest gap, consistent with pointwise KL keeping posterior latents close to the prior, but it also has lower posterior valid mass. MMD-CWAE attains the highest posterior valid mass, indicating that its posterior latents contain useful decoder-valid regions, but it has the largest posterior–prior gap. LAT-Flow gives the highest prior valid mass and a smaller gap than MMD/Sinkhorn, suggesting that learning the prior improves coverage of decoder-valid regions. We also find that this gap is sensitive to latent-space regularization during training: removing these regularizers substantially increases Δ_{valid}, widening the gap.

Posterior-geometry ablation. Aggregate matching constrains a batch-level latent distribution, but it does not by itself control whether the deterministic posterior codes have a geometry that is easy for the prior to sample. We therefore use dropout, decoder latent jitter, and posterior geometry regularization mentioned in C.1 to stabilize the posterior regions seen by the decoder. Table 7 ablates these regularizers. Removing them increases the posterior–prior valid-mass gap, especially on the $\mathsf{K} = 16$ tasks. The effect is largest for MMD-CWAE and Sinkhorn-CWAE: their posterior latents can still decode to valid trajectories, but prior samples miss the same decoder-valid regions much more often. Figure 8 shows the same failure mode qualitatively for Sinkhorn-CWAE on the synthetic *Sequential* task. Without posterior regularization, prior-sampled reconstructions degrade quickly and become highly noisy. With these regularizers, prior-sampled trajectories become substantially sharper and remain concentrated on the valid branches.

Conditional aggregate matching As mentioned in C.1, in simulation benchmarks the observation s changes the valid action distribution, so marginal latent matching can mix posterior and prior samples across incompatible conditioning states. Table 8 ablates the prior parameterization used by Sinkhorn-CWAE on PushT Image environment. All variants obtain similar posterior-sampled L_1 loss, indicating that the posterior encoder and decoder can reconstruct the demonstrated chunks.

Method	Avg. Δ_{valid}	Hard Δ_{valid}
KL-CVAE	0.04	0.05
MMD-CWAE	0.32	0.57
Sinkhorn-CWAE	0.32	0.58
LAT-Flow	0.18	0.30

Table 7: Effect of removing latent-space regularizers. We remove dropout, decoder latent jitter, and the posterior geometry regularizer, and report the posterior–prior valid-mass gap. Larger values indicate that posterior latents decode to valid trajectories more often than prior samples, revealing a prior-coverage failure.



(a) Without latent-space regularizers

(b) With latent-space regularizers

Figure 8: **Effect of latent-space regularization on Sinkhorn-CWAE.** Prior-sampled trajectories on the *Sequential* task. Each policy is rolled out 200 times. Without dropout, decoder latent jitter, and posterior geometry regularization, prior samples generate many dispersed and invalid trajectories. With the regularizers enabled, prior samples concentrate on valid trajectory branches and reach the goal regions more consistently. This qualitative behavior is consistent with Table 7: removing the regularizers increases the posterior–prior valid-mass gap, revealing that the posterior can define useful latent regions while the deployment prior fails to sample them reliably.

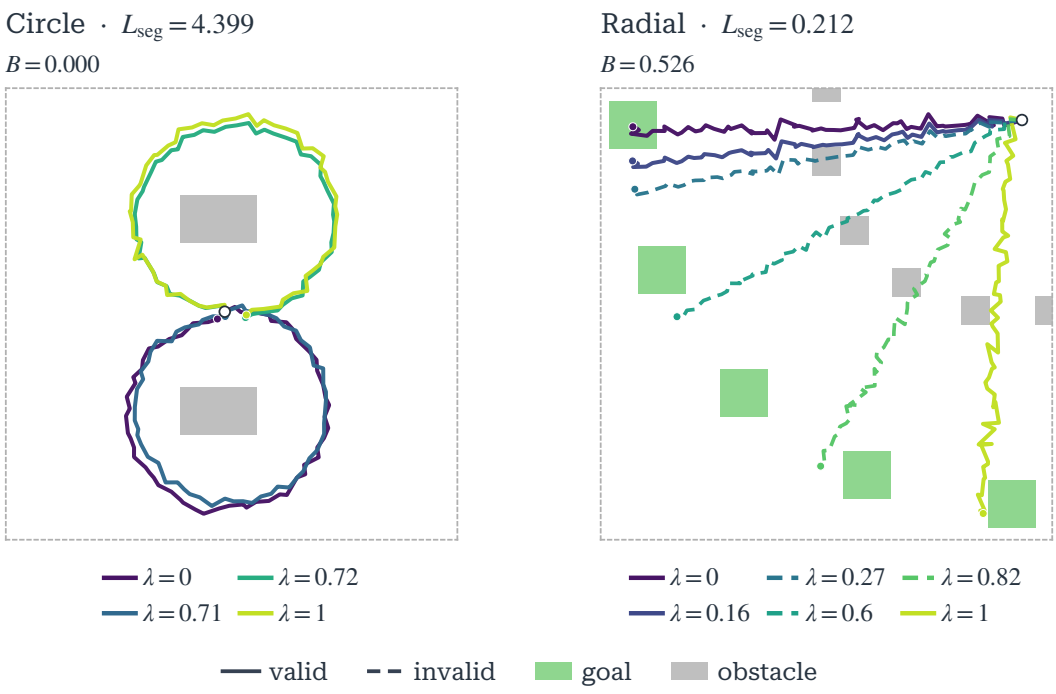


Figure 9: **Action-space bridge–smoothness trade-off.** Cross-mode interpolations in base space. Colors denote λ; dashed curves are invalid trajectories and therefore part of the bridge mass.

The difference appears at deployment: the fixed Gaussian prior produces very poor prior-sampled chunks and near-zero rollout performance. Learning the prior substantially reduces the posterior–prior sample gap and improves final IoU. Adding conditional aggregate matching gives the best closed-loop performance, suggesting that matching the state–latent coupling is more important for deployment than marginal latent matching alone.

Table 9: Bridge-smoothness trade-off for action-space samplers on synthetic tasks.

Method	Task	K	B	$\|u_{\lambda_+} - u_{\lambda_-}\|$	L_{seg}	L_{seg}/Δ
Act-Flow	Circle	2	0.0000	0.155	4.3985	7.92
Act-Flow	Sequential	4	0.0013	1.910	2.3893	17.65
Act-Flow	Corridor	16	0.5851	12.687	0.0405	1.67
Act-Flow	Radial	16	0.5259	10.558	0.2119	2.63
Act-Diff	Circle	2	0.0000	0.155	4.2850	7.71
Act-Diff	Sequential	4	0.0005	1.833	2.7834	20.56
Act-Diff	Corridor	16	0.5779	11.908	0.0520	2.14
Act-Diff	Radial	16	0.5475	10.730	0.2786	3.46

D.5 Action-Space Bridge–Smoothness Trade-off

Estimating the Lipschitz constant. We estimate the Lipschitz constant using finite differences. From a fixed s, we sample a pool of base noises $u \sim \mathcal{N}(0, \mathbf{I})$, and decode trajectories $G_{\theta,s}(u)$. We group the input noises of the valid decoded trajectories by output mode, and from each mode pair we randomly select a pair of noises (u_1, u_2), $G(u_1) \in C_i, G(u_2) \in C_j$. For each sampled cross-mode pair (u_1, u_2), we evaluate $G_{\theta,s}(u_\lambda)$ on a fixed grid of $N = 101$ interpolation coefficients $\lambda \in [0, 1]$, where $u_\lambda = (1-\lambda)u_1 + \lambda u_2$. We identify λ_+ as the first grid point whose decoded trajectory is valid in the second mode, and λ_- as the last preceding grid point whose decoded trajectory is valid in the first mode. We compute the finite-difference sensitivity

$$S_{\text{seg}} = \frac{d_{\text{traj}}\big(G_{\theta,s}(u_{\lambda_+}), G_{\theta,s}(u_{\lambda_-})\big)}{\|u_{\lambda_+} - u_{\lambda_-}\|}, \tag{20}$$

where d_{traj} is the root-mean-square pointwise Euclidean distance between trajectories. The maximum $L_{seg} := \max(S_{seg})$ is a lower bound of the Lipschitz constant $L_{\theta,s}$.

Full table results For each action-space sampler, we use its native inference setting: 10 solver steps for Act-Flow and 100 DDIM steps for Act-Diff. For each task, we fix the initial observation, sample 4096 base noises, select 100 cross-mode valid pairs, and evaluate 101 interpolation points per pair. We obtain the mode separation Δ_{ij} as the d_{traj} from the synthetic ground-truth trajectories for the two modes, generated from the known task geometry. Table 9 reports the results and the Act-Flow values used for Figure 9.

D.6 Measuring Multimodality in Robot Benchmark Datasets

Estimating conditional action multimodality requires distinguishing alternative actions at the same semantic state from action variation caused by differences between nearby but semantically different states. Directly clustering action chunks collected across a neighborhood can conflate these two sources of variation. We therefore fit a local joint state–action Gaussian mixture model (GMM) and condition it analytically on a query state to account for local state–action dependence. We then apply modal clustering to the resulting conditional density, identifying modes by their attraction basins [5]. This method allows several GMM components to describe a single action mode and mitigates over-segmentation caused by treating each Gaussian component as a distinct mode.

For each query proprioceptive state s_i, we retrieve nearby states from other demonstrations (with the same task or language instruction, where applicable). Both the query and its candidate neighbors must have a complete future chunk of H_{diag} recorded actions contained within their respective demonstrations. We denote the set of eligible state indices by $\mathcal{I}_{\mathsf{H}_{\text{diag}}}$. State coordinates are standardized using dataset-level moments within each task group. We use the dimension-normalized distance

$$d_{\text{state}}(i,j) = \frac{\|\widetilde{s}_i - \widetilde{s}_j\|_2}{\sqrt{d_s}},$$

where d_s is the state dimension. Each other demonstration contributes at most its nearest eligible state; the query demonstration is excluded. Let $\mathcal{J}_i$ denote the indices retained by this selection. For a radius ε, the neighborhood is

$$\mathcal{N}_\varepsilon(i) = \{j \in \mathcal{J}_i : d_{\text{state}}(i,j) \le \varepsilon\}, \qquad n_i = |\mathcal{N}_\varepsilon(i)|.$$

The main results use $\varepsilon = 0.5$. Sensitivity checks use $\varepsilon \in \{0.3, 0.5, 1.0\}$.

Local joint density and conditioning. Within each neighborhood, we flatten the future action chunks and fit a local principal component analysis (PCA) projection with $d_y = 2$ components. For $n_i = 2$, we retain one PCA component; neighborhoods with fewer than two demonstrations are excluded from density fitting. State coordinates retain their global standardization and are only centered locally. Let x denote the resulting state coordinates and y the PCA coordinates of the corresponding future action chunk. We fit a GMM to the concatenated state–action vectors $[x^\top, y^\top]^\top$, obtaining the local joint density

$$\widehat{p}_i(x, y) = \sum_{g=1}^{G_i} \pi_g \mathcal{N}\left(\begin{bmatrix} x \\ y \end{bmatrix} ; \begin{bmatrix} \mu_g^x \\ \mu_g^y \end{bmatrix}, \begin{bmatrix} \Sigma_g^{xx} & \Sigma_g^{xy} \\ \Sigma_g^{yx} & \Sigma_g^{yy} \end{bmatrix} \right),$$

where G_i is the number of Gaussian components and π_g are their mixture weights.

Candidate GMMs are fitted by expectation–maximization (EM). We compare $G = 1, \ldots, \min(20, n_i)$ components and two covariance structures: a separate full covariance matrix for each component, or a single full covariance matrix shared across components. For each configuration, we compare two initialization procedures. The first uses three k-means starts in joint state–action space. The second first fits an action-only full-covariance GMM with three starts, uses its component-membership probabilities (soft responsibilities) to initialize the joint mixture parameters, and performs one joint EM fit.

We select the candidate with the smallest Bayesian information criterion (BIC),

$$-2 \sum_{j \in \mathcal{N}_\varepsilon(i)} \log \widehat{p}_i(x_j, y_j) + p \log n_i,$$

where p is the number of free parameters. This selects both the number of Gaussian components and the covariance structure for each neighborhood.

We then analytically condition the selected joint GMM at the query coordinate x_i, obtaining

$$\widehat{p}_i(y \mid x_i) = \sum_g w_g(x_i) \, \mathcal{N}\big(y; \mu_{g|i}, \Sigma_{g|i}\big),$$

where

$$\begin{aligned} w_g(x_i) &= \frac{\pi_g \mathcal{N}(x_i; \mu_g^x, \Sigma_g^{xx})}{\sum_h \pi_h \mathcal{N}(x_i; \mu_h^x, \Sigma_h^{xx})}, \\ \mu_{g|i} &= \mu_g^y + \Sigma_g^{yx} (\Sigma_g^{xx})^{-1} (x_i - \mu_g^x), \\ \Sigma_{g|i} &= \Sigma_g^{yy} - \Sigma_g^{yx} (\Sigma_g^{xx})^{-1} \Sigma_g^{xy}. \end{aligned}$$

Modal clusters and entropy. We identify modes of the conditional action density using [5]'s Algorithm 2. A modal cluster consists of points that converge to the same density maximum and may contain probability mass contributed by several Gaussian components.

We draw $R = 4096$ samples from $\widehat{p}_i(y \mid x_i)$. For each sample, we first select a component according to $w_g(x_i)$ and then sample from its conditional Gaussian. Gaussian-mixture mean shift iteratively moves each sample toward a density maximum; samples reaching the same maximum are assigned to the same modal cluster. Endpoints are checked for negative curvature, and numerically identical maxima are merged. The probability mass of each modal cluster is estimated by the fraction of conditional samples assigned to it:

$$\widehat{m}_{i\ell} = \frac{1}{R} \sum_{r=1}^{R} \mathbf{1}\{y_i^{(r)} \text{ converges to mode } \ell\}.$$

We retain modal clusters with estimated mass at least 2.5%,

$$\mathcal{R}_i = \{\ell : \widehat{m}_{i\ell} \geq 0.025\},$$

and report their count and entropy:

$$\widehat{K}(s_i) = |\mathcal{R}_i|, \qquad \widehat{H}_i = -\sum_{\ell \in \mathcal{R}_i} \widetilde{m}_{i\ell} \log_2 \widetilde{m}_{i\ell}, \qquad \widetilde{m}_{i\ell} = \frac{\widehat{m}_{i\ell}}{\sum_{j \in \mathcal{R}_i} \widehat{m}_{ij}}.$$

The probability cutoff is applied to modal-cluster mass after conditioning and clustering; retained masses are renormalized for entropy.

Table 10: Dataset-wide conditional-mode $\hat{K}$ and conditional entropy $\hat{H}$ estimates. Entries correspond to radii $\varepsilon = 0.3/0.5/1.0$. The modal-mass cutoff is 2.5%; entropy is in bits.

Horizon	$\widehat{K}$	$\widehat{H}$
UR3 BlockPush		
10	$1.128/1.136/1.110$	$0.080/0.082/0.068$
30	$1.159/1.164/1.098$	$0.098/0.103/0.055$
Kitchen		
10	$1.000/1.143/1.111$	$0.000/0.083/0.071$
30	$1.000/1.231/1.145$	$0.000/0.147/0.099$
PushT		
10	$1.022/1.068/1.144$	$0.013/0.038/0.087$
30	$1.000/1.005/1.104$	$0.000/0.004/0.059$
LIBERO		
10	$1.003/1.014/1.028$	$0.003/0.007/0.016$
30	$1.023/1.027/1.033$	$0.015/0.017/0.020$
MetaWorld		
10	$1.009/1.037/1.069$	$0.004/0.020/0.040$
30	$1.022/1.067/1.123$	$0.016/0.043/0.073$
Bimodal Tissue Grasping		
10	$1.486/1.468/1.454$	$0.449/0.432/0.419$
30	$2.267/2.267/2.267$	$0.760/0.760/0.760$

Dataset protocols. For the synthetic validation, we use 100 demonstration-initial query states per benchmark and $H = 60$. The robot dataset fits use query states across demonstration trajectories, with $H \in \{10, 30\}$. We sample 1000 query starts per dataset, without replacement and stratified by task. We allocate at least 16 queries per language task when available, then distribute remaining queries according to the task's state population.

Results. Table 10 shows the complete results with varying neighborhood radius. The simulation estimates are generally close to one, while Bimodal Tissue Grasping, given its proof-of-concept nature, exhibits stronger conditional multimodality. Ambiguity may be concentrated at initial or branching states, with subsequent states admitting largely unambiguous continuations. Once a policy commits to a branch, potentially through small prediction or execution deviations, replanning can guide it along a coherent expert strategy.

D.7 Simulation and Real-World Robot Results

We provide the complete results from Section 5.2. Table 11 reports performance at the final and best checkpoints, allowing the best performance achieved during training to be distinguished from subsequent degradation. Table 12 reports diversity across completed behavior sequences for simulation benchmarks with available ground-truth behavior information or reference entropy values. Table 13 reports performance on the language-conditioned LIBERO and Meta-World tasks.

Table 14 reports the complete results for the bimodal tissue-grasping robot task, including task success, conditional mode entropy, inter-mode trajectory separation, and invalid-rollout counts. A rollout is classified as invalid if the policy grasps the delicate region of interest that should be avoided. A rollout is classified as a failure if it is invalid or does not complete a successful grasp within a reasonable time. Finally, Table 15 reports the inference throughput of the different policy parametrizations under the evaluation settings used in simulation.

These results suggest four observations:

1. BCAT performs reasonably well across the simulation benchmarks, consistent with limited conditional action multimodality in their demonstration data. It also achieves higher average inference throughput than the flow-matching and CVAE policies evaluated here. The choice of a multimodal-capable policy should therefore be informed by whether the demonstration data actually require this representational capacity.

2. The performance of multimodal-capable policies varies with training dynamics. Latent-variable policies involve many hyperparameters, resulting in a large search space for an effective configuration. Action-space flow or diffusion policies may therefore be preferable when reducing this tuning burden is a priority.

3. Some models collapse late in training, including Act-Flow on UR3 BlockPush and KL-CVAE-LP on Kitchen Image. These observations highlight the importance of checkpoint selection and motivate further investigation of the role of covariate shift [28, 18] in performance degradation.

4. As emphasized in the manuscript, the bimodal tissue-grasping task illustrates the consequences of failing to represent certified multimodal demonstrations. BCAT collapses toward an average between modes, whereas Sinkhorn-CWAE-LP collapses to a single demonstrated mode. In this case, single-mode collapse retains a demonstrated strategy rather than interpolating into a region outside the expert support. Although Sinkhorn-CWAE-LP does not recover the full multimodal behavior, it still achieves an 85% success rate.

D.8 Relationship Between Policy Performance and Action Horizon

This ablation examines how policy performance varies with the training and execution action horizons. As noted in the manuscript, the multimodality of the demonstration data is horizon-dependent. We investigate how the choice of training horizon affects policy performance and how this interacts with the execution horizon, which determines the frequency of replanning.

The results in Table 16 show that longer open-loop execution amplifies failure but does not fully explain it. Performance generally deteriorates at longer execution horizons, particularly for long training chunks, consistent with the accumulation of open-loop errors. However, longer training chunks can also reduce performance at a fixed, short execution horizon, suggesting that modeling longer, higher-dimensional, and potentially more multimodal action distributions introduces additional difficulty. Training horizons of around 10 actions and execution horizons of 2–5 actions typically yield the best performance. This suggests a double trade-off: during training, short chunks provide limited temporal consistency but are easier to model, while long chunks teach policies temporal coherence of the actions but need to learn a higher-dimensional conditional distribution. During execution, short horizons enable frequent replanning, improving reactivity and limiting the accumulation of deployment errors, but reduce sustained commitment to a global strategy. Longer execution horizons instead enforce commitment to a longer-term plan, at the cost of accumulating open-loop errors. Future work should investigate how policy modeling interacts with the execution horizon, and how these choices jointly affect covariate shift, multimodality representation, and generalization.

Table 11: Simulation evaluation. 'LP' stands for 'learned prior' variant. When not specified, prior is a fixed standard Gaussian. Values are mean ± standard deviation across three evaluation seeds, with 200 rollouts per seed. Final denotes epoch 1000; Best is selected among checkpoints evaluated every 100 epochs, with its epoch in superscript. A dash denotes Best coinciding with Final. Bold indicates the highest reported mean for each benchmark.

Policy		**PushT State** (/1)	**PushT Image** (/1)	**Kitchen State** (/4)
BCAT	Final	0.723 ± 0.046	0.663 ± 0.012	2.76 ± 0.01
	Best	–	$(0.692 \pm 0.011)^{700}$	$(2.78 \pm 0.01)^{500}$
KL-CVAE	Final	0.625 ± 0.024	0.662 ± 0.016	1.78 ± 0.08
	Best	$(0.635 \pm 0.021)^{900}$	–	–
KL-CVAE-LP	Final	0.765 ± 0.013	0.749 ± 0.022	3.06 ± 0.07
	Best	$(0.796 \pm 0.019)^{400}$	$(\mathbf{0.793 \pm 0.021})^{900}$	–
MMD-CWAE-LP	Final	0.745 ± 0.003	0.647 ± 0.016	2.97 ± 0.04
	Best	–	–	$(3.07 \pm 0.07)^{400}$
Sinkhorn-CWAE-LP	Final	0.694 ± 0.034	0.744 ± 0.029	3.21 ± 0.10
	Best	$(0.725 \pm 0.024)^{500}$	–	$(\mathbf{3.26 \pm 0.02})^{300}$
Act-Flow	Final	$\mathbf{0.825 \pm 0.014}$	0.778 ± 0.014	2.53 ± 0.07
	Best	–	$(0.781 \pm 0.014)^{800}$	$(2.83 \pm 0.15)^{200}$
LAT-Flow	Final	0.682 ± 0.027	0.724 ± 0.023	3.06 ± 0.08
	Best	$(0.703 \pm 0.006)^{500}$	$(0.747 \pm 0.026)^{400}$	–

Policy		**Kitchen Image** (/4)	**UR3 BlockPush** (/2)
BCAT	Final	0.53 ± 0.05	$\mathbf{1.99 \pm 0.01}$
	Best	$(1.86 \pm 0.04)^{600}$	–
KL-CVAE	Final	1.12 ± 0.05	1.92 ± 0.02
	Best	–	$(1.96 \pm 0.03)^{100}$
KL-CVAE-LP	Final	0.94 ± 0.01	1.65 ± 0.03
	Best	$(1.70 \pm 0.04)^{600}$	–
MMD-CWAE-LP	Final	1.28 ± 0.01	1.88 ± 0.01
	Best	$(\mathbf{2.67 \pm 0.03})^{500}$	–
Sinkhorn-CWAE-LP	Final	1.31 ± 0.02	1.63 ± 0.02
	Best	$(2.04 \pm 0.06)^{800}$	$(1.92 \pm 0.01)^{200}$
Act-Flow	Final	2.33 ± 0.04	0.47 ± 0.01
	Best	$(2.58 \pm 0.06)^{900}$	$(1.61 \pm 0.02)^{200}$
LAT-Flow	Final	1.09 ± 0.05	1.49 ± 0.02
	Best	$(1.86 \pm 0.03)^{900}$	$(1.53 \pm 0.01)^{300}$

Table 12: Behavioral entropy with seed-level uncertainty, following the QFAT/VQ-BeT convention: Shannon entropy (bits) of the empirical distribution over completed behavior sequences, $H = -\sum_m \hat{p}(m) \log_2 \hat{p}(m)$. Mean $\pm$ std over 3 evaluation seeds $\times$ 200 rollouts; Final and Best as in Table 11.

	Kitchen State		**Kitchen Image**		**UR3 BlockPush**	
Policy	Final	Best	Final	Best	Final	Best
BCAT	3.43 ± 0.05	$3.04 \pm 0.00^{(500)}$	1.81 ± 0.26	$1.28 \pm 0.62^{(600)}$	0.98 ± 0.00	–
KL-CVAE	3.09 ± 0.14	–	1.52 ± 0.52	–	$\mathbf{0.99 \pm 0.00}$	$0.93 \pm 0.01^{(100)}$
KL-CVAE-LP	$\mathbf{3.76 \pm 0.09}$	–	0.00 ± 0.00	$1.71 \pm 0.26^{(600)}$	0.97 ± 0.02	–
MMD-CWAE-LP	2.48 ± 0.07	$2.48 \pm 0.10^{(400)}$	0.00 ± 0.00	$1.98 \pm 0.12^{(500)}$	0.98 ± 0.00	–
Sinkhorn-CWAE-LP	2.66 ± 0.16	$2.62 \pm 0.04^{(300)}$	0.00 ± 0.00	$1.51 \pm 0.11^{(800)}$	0.71 ± 0.04	$\mathbf{0.99 \pm 0.00}^{(200)}$
Act-Flow	3.08 ± 0.37	$2.84 \pm 0.17^{(200)}$	$\mathbf{2.60 \pm 0.20}$	$2.36 \pm 0.27^{(900)}$	0.00 ± 0.00	$\mathbf{0.99 \pm 0.01}^{(200)}$
LAT-Flow	3.28 ± 0.15	–	0.00 ± 0.00	$2.17 \pm 0.30^{(900)}$	0.07 ± 0.07	$0.03 \pm 0.05^{(300)}$

Table 13: Success rates (%) by LIBERO subtask suite and MetaWorld task difficulty. Policies are all evaluated on the last epoch checkpoint. Values are mean $\pm$ standard deviation over three seeds, 400 rollouts.

Benchmark	BCAT	KL-CVAE	KL-CVAE-LP	MMD-CWAE-LP	Sinkhorn-CWAE-LP	Act-Flow	LAT-Flow
LIBERO							
Long	86.33±1.25	**92.33±0.94**	83.67±0.47	89.00±0.00	90.00±0.82	87.00±3.74	88.67±0.47
Spatial	92.33±1.25	**98.67±1.25**	96.00±0.82	**98.67±0.94**	96.00±0.00	93.67±0.47	94.00±2.16
Goal	96.00±0.82	**97.33±1.25**	95.67±0.47	96.00±0.82	96.00±0.82	91.00±2.16	94.33±1.25
Object	99.67±0.47	**100.00±0.00**	98.33±0.47	**100.00±0.00**	97.67±0.47	97.67±0.94	99.00±0.82
Total	93.58±0.66	**97.08±0.82**	93.42±0.31	95.92±0.31	94.92±0.24	92.33±0.85	94.00±0.35
MetaWorld							
Easy	61.55±0.44	71.78±1.62	**81.31±1.22**	68.45±1.38	72.26±0.73	39.28±1.27	69.41±2.19
Medium	46.97±2.14	55.15±0.42	**58.49±2.81**	43.33±2.14	37.27±3.86	18.79±4.35	37.27±1.97
Hard	69.45±2.08	75.00±1.36	71.67±2.72	57.22±6.98	**78.33±6.81**	34.45±3.93	73.33±2.72
Very Hard	46.00±6.53	**73.33±3.77**	65.33±1.89	68.67±2.49	58.00±4.32	38.00±4.32	49.33±5.25
Total	57.73±0.52	68.67±1.41	**73.53±0.34**	61.60±1.18	63.87±1.52	34.07±2.39	60.80±0.99

Table 14: On-robot rollout metrics on the bimodal bowel-grasping task, with 20 recorded rollouts per policy. Mode entropy $H(M \mid S = s_0)$ is conditioned on the initial state. Inter-mode width is the Euclidean distance between the two per-mode mean trajectories. Invalid denotes a mode-average trajectory into the invalid region of interest, while general failure in reaching/grasping is incorporated in the success rate.

Policy	$H(M \mid S)$	Max width (mm)	Mean width (mm)	Invalid	Success rate
BCAT	0.000	—	—	20	0/20 (0.0)
Act-Flow	1.000	40.0	19.6	1	19/20(0.95)
KL-CVAE-LP	0.881	37.9	15.4	2	17/20(0.85)
MMD-CWAE-LP	0.982	33.1	13.3	3	13/20(0.65)
LAT-Flow	0.764	41.3	16.6	0	16/20(0.8)
Sinkhorn-CWAE-LP	0.000	—	—	2	17/20(0.85)

Table 15: Inference throughput in policy queries per second, measured as the inverse wall-clock time required to generate one action chunk. All methods are evaluated with the same hardware, batch size, and native inference setting used in the simulation experiments. Higher is better.

Method	PushT Img.	PushT State	Kitchen Img.	Kitchen State	UR3 BlockPush	Avg.
Act-Flow	6.24	6.59	4.61	5.79	6.24	5.89
LAT-Flow	13.78	20.58	8.79	12.66	15.19	14.20
KL-CVAE	30.63	69.02	17.50	21.62	19.73	31.70
KL-CVAE-LP	31.97	58.94	12.11	14.75	30.18	29.59
MMD-CWAE-LP	29.89	59.10	13.31	27.05	15.72	29.02
Sinkhorn-CWAE-LP	27.06	33.76	12.69	14.24	29.72	23.49
BCAT	43.86	63.72	18.95	16.86	25.57	33.79

Table 16: Execution-horizon ablation. For each benchmark, method, and training chunk, the best checkpoint is selected once from each training chunk horizon, then held fixed across all execution horizons. Superscripts give the selected training epoch.

Benchmark	Training horizon	Execution horizon	BCAT	KL-CVAE-LP	Act-Flow
PushT State ($\cdot/1$)	5	2	$0.597^{(200)}$	$0.743^{(500)}$	$0.778^{(300)}$
		5	$0.724^{(200)}$	$0.800^{(500)}$	$0.816^{(300)}$
	10	2	$0.723^{(1000)}$	$0.796^{(400)}$	$0.825^{(1000)}$
		5	$0.734^{(1000)}$	$0.827^{(400)}$	$\mathbf{0.854}^{(1000)}$
		10	$0.611^{(1000)}$	$\mathbf{0.829}^{(400)}$	$0.813^{(1000)}$
	30	2	$0.643^{(1000)}$	$0.727^{(800)}$	$0.701^{(800)}$
		5	$\mathbf{0.749}^{(1000)}$	$0.809^{(800)}$	$0.808^{(800)}$
		10	$0.662^{(1000)}$	$0.723^{(800)}$	$0.774^{(800)}$
		30	$0.326^{(1000)}$	$0.473^{(800)}$	$0.615^{(800)}$
Kitchen State ($\cdot/4$)	5	2	$3.19^{(600)}$	$3.04^{(300)}$	$2.56^{(900)}$
		5	$3.02^{(600)}$	$2.93^{(300)}$	$2.23^{(900)}$
	10	2	$\mathbf{3.29}^{(500)}$	$3.21^{(1000)}$	$\mathbf{2.83}^{(200)}$
		5	$3.18^{(500)}$	$\mathbf{3.27}^{(1000)}$	$2.65^{(200)}$
		10	$2.94^{(500)}$	$2.92^{(1000)}$	$2.38^{(200)}$
	30	2	$3.00^{(1000)}$	$2.74^{(900)}$	$2.42^{(300)}$
		5	$2.96^{(1000)}$	$2.84^{(900)}$	$2.21^{(300)}$
		10	$2.83^{(1000)}$	$2.56^{(900)}$	$1.98^{(300)}$
		30	$1.03^{(1000)}$	$1.12^{(900)}$	$1.00^{(300)}$
UR3 BlockPush ($\cdot/2$)	5	2	$1.94^{(700)}$	$1.67^{(900)}$	$\mathbf{1.68}^{(100)}$
		5	$1.93^{(700)}$	$1.52^{(900)}$	$1.57^{(100)}$
	10	2	$\mathbf{1.99}^{(1000)}$	$1.65^{(1000)}$	$1.48^{(200)}$
		5	$1.97^{(1000)}$	$1.53^{(1000)}$	$1.43^{(200)}$
		10	$1.97^{(1000)}$	$1.47^{(1000)}$	$1.45^{(200)}$
	30	2	$1.79^{(500)}$	$\mathbf{1.72}^{(400)}$	$1.35^{(100)}$
		5	$1.81^{(500)}$	$1.69^{(400)}$	$1.35^{(100)}$
		10	$1.87^{(500)}$	$1.63^{(400)}$	$1.24^{(100)}$
		30	$1.77^{(500)}$	$1.44^{(400)}$	$1.18^{(100)}$

E Experiment Hyperparameters

We report the hyperparameters used across synthetic and simulation experiments. A dash denotes a setting that is not applicable or not run. Unless stated otherwise, all models use AdamW with learning rate 5×10^{-4}, weight decay 0.02, observation horizon $\mathsf{T}_{\text{obs}} = 1$, and action horizon $\mathsf{H} = 10$ for simulation and robot tasks. Synthetic tasks use $\mathsf{H} = 60$. Image encoders are fine-tuned jointly with the policy. All transformer layers use ReLU as activation function.

Latent and generative objectives. For KL-CVAE, the pointwise KL weight is $\beta_{\text{KL}} = 0.01$. Learned-prior KL variants use a conditional Gaussian prior with fixed unit variance. For VQ-VAE, the commitment weight is $\beta_{\text{vq}} = 0.25$ and the number of residual quantization layers is $\mathsf{L}_{\text{vq}} = 2$.

For aggregate-matched latent models, posterior geometry regularization uses decoder-side latent jitter $\sigma_{\text{dec}} = 0.075$, mean penalty weight $\beta_{\text{mean}} = 0.01$, standard-deviation penalty weight $\beta_{\text{std}} = 0.05$, covariance penalty weight $\beta_{\text{cov}} = 0.01$, target standard deviation $\sigma_\star = 1$, and maximum standard deviation $\sigma_{\max} = 2$. Learned latent priors are matched to posterior means.

For MMD-CWAE and its conditional variant, the latent aggregate matching weight is $\beta = 1.0$. We use the inverse multiquadratic kernel [39]

$$k_{\text{IMQ}}(z, z') = \sum_{C \in \mathcal{C}_z} \frac{C}{C + \|z - z'\|_2^2}.$$

For synthetic experiments, $\mathcal{C}_z = \{0.5\mathsf{d}_z, \mathsf{d}_z, 2\mathsf{d}_z, 4\mathsf{d}_z\}$. For simulation and robot experiments, $\mathcal{C}_z = \{\mathsf{d}_z, 2\mathsf{d}_z, 4\mathsf{d}_z, 8\mathsf{d}_z\}$.

For Sinkhorn-CWAE and its conditional variant, the latent aggregate matching weight is $\beta = 0.01$, the blur parameter is $\varepsilon_{\text{Sink}} = 0.1$, and the transport cost is squared Euclidean distance.

For both conditional aggregate matching models (Appendix C.1) used in the simulation experiments, the state-conditioning weight is $\lambda_s = 1.0$.

For LAT-Flow, the latent flow-prior loss weight is $\beta_{\text{flow}} = 0.01$. LAT-Flow and Act-Flow both use an Euler solver with 10 inference steps and sample training times from $\text{Beta}(1.5, 1)$.

For Act-Diff, we use a DDIM scheduler with 100 training timesteps and 100 inference steps, the squared-cosine capped noise schedule, sample prediction, sample clipping enabled, $\alpha_0 = 1$, and step offset 0.

Dropout. Synthetic action decoders use dropout 0.2 and attention dropout 0.05. Simulation and robot action decoders use dropout 0.1 and attention dropout 0.01. Posterior encoders use dropout 0.2 and attention dropout 0.05. Learned-prior encoders use dropout 0.1 and attention dropout 0.01.

Data augmentation and normalization. We normalize observations and actions to $[-1, 1]$ using min–max scaling. For image-based simulation experiments, we apply data augmentation with Albumentations [4]. The photometric augmentation pipeline contains ColorJitter with brightness 0.3, contrast 0.4, saturation 0.5, hue 0.1, and probability 0.5; RandomSunFlare with flare region $(0, 0, 1, 0.5)$, white source color, and probability 0.6; RandomBrightnessContrast with brightness and contrast limits 0.4 and probability 0.6; RandomGamma with gamma range $(80, 120)$ and probability 0.3; CLAHE with clip limit 4.0 and probability 0.3; RandomShadow with probability 0.4; and JPEG ImageCompression with quality range $[50, 100]$ and probability 0.2. The spatial and occlusion augmentation pipeline contains GaussianBlur with blur limit $(3, 7)$ and probability 0.5, CoarseDropout with at most 8 holes of size 8×8 and probability 0.3, and ShiftScaleRotate with shift limit 0.0625, scale limit $[-0.5, 0.6]$, zero rotation, and probability 0.5.

Table 17: Common benchmark-level training settings.

Setting	Synthetic	PushT-S	PushT-I	Kitchen-S	Kitchen-I	UR3-S	MBG
Observation Dimension	3x64x64	24	3x96x96	30	3x112x112	6	4
Observation horizon	1	1	1	1	1	1	1
Action horizon H	60	10	10	10	10	10	10
Batch size	256	256	64	256	64	256	256
Validation Ratio	0.05	0.05	0.05	0.05	0.05	0.05	0.05
Epochs	600 (Circle) / 1000	1000	1000	1000	1000	1000	1000
Rollout every	100	100	100	100	100	100	–
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW
Policy LR	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}
Weight decay	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Image-encoder LR	10^{-5}	–	10^{-5}	–	10^{-5}	–	–
Image-encoder WD	0.001	–	0.001	–	0.001	–	–
EMA	–	true	true	true	true	true	true

Table 18: Common training settings for LIBERO and Meta-World.

Setting	LIBERO	Meta-World
Image observation dimension	$2 \times (3 \times 256 \times 256)$	$3 \times 480 \times 480$
Language encoder	all-MiniLM-L6-v2	all-MiniLM-L6-v2
Language encoding dimension	31×384	31×384
Language encoder frozen	true	true
Observation horizon	1	1
Action horizon H	10	10
Batch size	64	64
Validation ratio	0.05	0.05
Epochs	100	100
Optimizer	AdamW	AdamW
Policy LR	5×10^{-5}	5×10^{-5}
Weight decay	0.02	0.02
Image-encoder LR	10^{-5}	10^{-5}
Image-encoder WD	0.001	0.001
EMA	false	false

Table 19: Model sizes used in the experiments. L denotes transformer layers, d hidden width, h attention heads, and FF the feedforward width. All transformers use ReLU as activation function.

Module / family	Synthetic	Simulation/Robot
Image encoder	MobileNetV4 Conv Small 0.50	EfficientNet-B0
State encoder	–	Linear projection (state_dim, embedding_dim)
BCAT decoder	$L = 8,\ d = 256,\ h = 8,\ \text{FF} = 512$	$L = 8,\ d = 256,\ h = 8,\ \text{FF} = 1024$
Latent decoder	$L = 8,\ d = 256,\ h = 8,\ \text{FF} = 512$	$L = 8,\ d = 256,\ h = 8,\ \text{FF} = 1024$
MMDiT decoder	$L = 8,\ d = 256,\ h = 8,\ \text{FF} = 512$, timestep_emb=32	$L = 8,\ d = 256,\ h = 8,\ \text{FF} = 1024$, timestep_emb=128
Posterior encoder	$L = 8,\ d = 256,\ h = 8,\ \text{FF} = 512$	$L = 8,\ d = 256,\ h = 8,\ \text{FF} = 1024$
Learned Gaussian prior	–	$L = 4,\ d = 256,\ h = 8,\ \text{FF} = 1024$
Conditional aggregate prior	–	$L = 8,\ d = 256,\ h = 8,\ \text{FF} = 1024$
Latent flow prior	$L = 8,\ d = 256,\ h = 8,\ \text{FF} = 512$	$L = 8,\ d = 256,\ h = 8,\ \text{FF} = 1024$

Table 20: Latent dimensions and codebook sizes.

Benchmark / task	Modes K	Latent dim. d_z	VQ codebook size C
Circle	2	1	2
Sequential	4	2	4
Radial	16	4	16
Corridor	16	4	16
Simulation benchmarks	–	16	–

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state the paper's scope: an analysis of multimodality in behavioral cloning across latent-variable and action-space generative action-chunking policies. The theoretical claims are formalized in Section 4 and Appendix B, and the empirical claims are evaluated in Section 5 and additionally in Appendix D.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The conclusion highlights the main limitations: the strongest mode-level diagnostics rely on controlled synthetic tasks with known mode labels and simulation benchmarks.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Section 4 states the assumptions and results for the latent-information bounds, aggregate matching, and action-space geometric cost. Full proofs and supporting notation and definitions are provided in Appendix A and B.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 5 describes the benchmark structure, evaluated policy families, and main metrics. Appendices C, D and E provide additional architecture, training, diagnostic, and evaluation details. Exact bitwise reproducibility is not guaranteed because some GPU operations may be nondeterministic, but the reported protocols and aggregate metrics are reproducible in expectation.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide as supplementary material a downloadable ZIP archive for the code, synthetic data-generation scripts, simulation datasets, simulation benchmark evaluation servers, diagnostic scripts, README and reproduction instructions. The archive is anonymized and contains no author-identifying metadata. Upon acceptance, we will release the final public version.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setup is summarized in Section 5. Full details on policy architectures, training settings, hyperparameters, datasets, inference settings, and evaluation metrics are provided in Appendix C and E.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper reports averages over fixed evaluation protocols and, where applicable, multiple rollout samples, but does not provide confidence intervals or formal statistical significance tests for all experiments. Because the study compares several policy classes across multiple benchmarks and diagnostics, repeated multi-seed evaluation is computationally expensive. We therefore emphasize controlled diagnostics and report detailed per-task and aggregate results in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides accounting of GPU type, memory, and runtime for the reported experiment in Appendix C.4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The work is a methodological study of imitation-learning policy classes using synthetic and simulation benchmarks. We do not release high-risk models, collect sensitive data, or conduct human-subject experiments. We will release the code with the appropriate public licenses.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The paper is primarily a foundational and methodological study of multimodal behavioral cloning. It discusses technical limitations, but does not include a dedicated broader-impact analysis of positive and negative societal consequences.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release high-risk pretrained models, scraped datasets, language models, or image generators requiring misuse safeguards.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper cites the original works for existing benchmarks and policy families used as related works, including Push-T, Block Push, Kitchen, ACT, Diffusion Policy, VQ-BeT, and related methods. Existing assets are used only for research benchmarking and are credited in the corresponding sections.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The synthetic multimodal benchmark tasks introduced in the paper are specified in Section 5 and Appendix C, including task geometry, mode definitions, success criteria, and diagnostic metrics. The code to generate the data is released, as previously mentioned.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or human-subject studies.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human-subject research, so IRB approval or equivalent review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used

only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not part of the core methodology, experiments, theory, or scientific contribution of the paper.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.