

When Do Discrete Actions Make Good Continuous Controllers?

Zeji Yi^{1,2,†}, Soham Dasgupta^{1,2,†}, Xinyang Zhang², Koushil Sreenath², Guannan Qu¹, Xi Chen², Max Simchowitz^{1,2,*}, Guanya Shi^{1,2,*}

¹Carnegie Mellon University ²Amazon FAR (Frontier AI & Robotics)

[†]Work done during internship at Amazon FAR. ^{*}Equal advising.

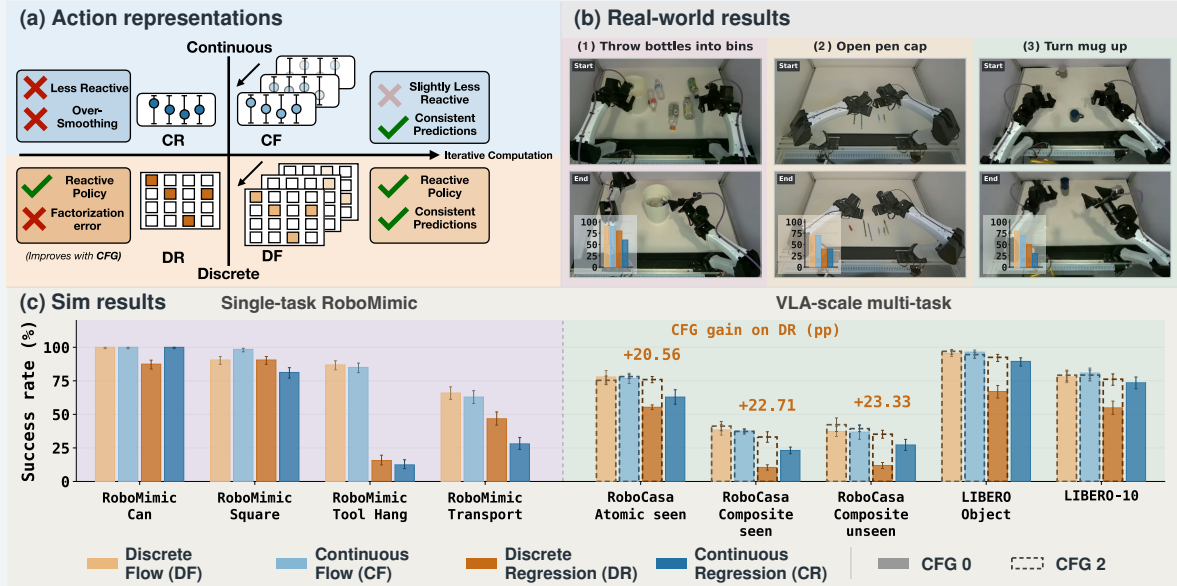


Figure 1: (a) Action Representation and comparison; (b) Closed-loop success on RoboMimic, and VLA-sim (RoboCasa and LIBERO); (c) VLA-real (three real-world tasks); Within each group, heads appear in the order DF, CF, DR, CR; orange denotes discrete heads and blue denotes continuous heads. For RoboCasa and LIBERO DF, CF, and DR, filled bars show CFG 0 and transparent dashed bars show CFG 2. Details are in Appendix C.

Date: October 13, 2026

Website: <https://iscoyizj.github.io/discrete-action-website>

Abstract

Though modern AI models like LLMs predict discrete tokens, today’s robotic foundation models represent actions as continuous vectors. In this paper, we challenge this common consensus by asking if continuous action parametrizations are superior to their discrete counterparts. We conduct an extensive comparison of single-step regression and multi-step flow variants of both continuous and discrete policies in both from-scratch single-task and multi-task VLA training, in both simulation and on real hardware. In all regimes, we find that discrete action prediction, even the most naive, uniform-bin quantization, can match or even outperform continuous-action policies in task success rate. Moreover, we find that the challenge to discrete action policies is *not* action precision, but rather factorization error introduced by product-distribution predictions across time and action

dimensions. We show that, when paired with the correct interventions—iterative computation through discrete flows, or distribution sharpening through classifier-free guidance—discrete flow policies can turn this weakness into a strength, enabling more responsive action changes during a predicted action chunk. Lastly, we show that attempts to explicitly inject geometric information into discrete policies, e.g. through corruption processes that emulate continuous flow matching, do *not* yield consistent improvements. Taken together, our findings question existing assumptions that robotic action prediction requires distinct design decisions compared to other AI domains, and motivates the possibility of discrete action prediction as a compelling alternative to today’s continuous action parametrizations. Link for the webpage <https://iscoyizj.github.io/discrete-action-website>

1 Introduction

A major goal of modern robot learning is to perform complex and varied tasks to high precision. As a result, the community has devoted significant study to the question of how to parametrize the actions predicted by robotic policies and foundation models, such as vision-language-action models (VLAs) [Brohan et al., 2023, Kim et al., 2024, Black et al., 2025, NVIDIA et al., 2025] and world action models (WAMs) [Li et al., 2025, Ye et al., 2026b, Kim et al., 2026, Li et al., 2026b]. The common consensus is that robot models should predict continuous-valued actions during deployment, as these are lossless and therefore support high-precision behavior. For example, most of today’s VLAs predict actions using continuous-space generative models, such as flow [Black et al., 2025, Physical Intelligence et al., 2025b] or diffusion models [Chi et al., 2023], which are believed to better capture training data diversity whilst enhancing the precision of action predictions.

At the same time, modern robot learning is adopting increasingly more features from other AI domains, notably language modeling. These include test-time reasoning [Wei et al., 2022], on-task self-correction [Huang et al., 2022], future goal prediction [Du et al., 2023], and long-term memory [Wang et al., 2023]. In fact, recent work has showcased the possibility that frontier language models can directly control robot embodiments through text generation [Brohan et al., 2023, Kim et al., 2024]. Yet, today’s frontier AI models, even ones that process multiple modalities such as text, images, and audio, both take as input and produce discretely quantized tokens. Whether robot actions should follow the same discrete format therefore raises a key question:

Are robot actions unique among other AI modalities in that they must be predicted as continuous values, or can **discrete action representations** compete with, or even outperform, their continuous counterparts?

1.1 Contributions

In this work, we undertake a systematic comparison between the performance of continuous and discrete action parameterizations for both single-task robot policies and multi-task vision-language-action (VLA) models. In short, we show that **discrete action parametrization can be competitive with, and even uniformly outperform, continuous action prediction**, provided that correct design decisions are made. These findings provide a deeper understanding of the performance of recent discrete diffusion/flow VLA policies [Liang et al., 2026b, Wen et al., 2025b, Chen et al., 2026a].

In greater detail, we consider parametrizations of the *action head*, or part of a robot model responsible for action prediction. Our comparison builds on classical supervised control and imitation learning [Pomerleau, 1988, Ross et al., 2011, Osa et al., 2018]. We focus our study on 4 representative parametrizations: (1) continuous regression (CR), which predicts continuous-valued actions in a single forward pass and is trained with a regression objective [Bojarski et al., 2016, Kim et al., 2025] (2) continuous flow (CF), which generates continuous actions using flow matching, a family applied to

robot control and motion planning [Hu et al., 2024, Chisari et al., 2024, Black et al., 2025, Zhang et al., 2024, Nguyen et al., 2025] (3) discrete regression (DR), which predicts a quantized action via cross-entropy classification [Brohan et al., 2022] and (4) discrete flow (DF), which predicts quantized actions in multiple steps via the discrete flow matching objective [Gat et al., 2024]. Hybrid methods such as Behavior Transformer [Shafiullah et al., 2022] combine discrete action-bin prediction with a continuous offset; in contrast, we study purely discrete and continuous action heads separately to isolate the effect of action representation. Recent work compares regression and flow policies under architecture-matched settings [Pan et al., 2026]. We evaluate these parametrizations from single-task policies on RoboMimic [Mandlekar et al., 2021] to multi-task VLA-scale policies using RoboCasa [Nasiriany et al., 2026], DROID [Khazatsky et al., 2024], and LIBERO [Liu et al., 2023], as well as real-world robot experiments. All are evaluated using closed-loop task success, action prediction accuracy, and within-chunk temporal variation. We identify 3 key findings that upend prior beliefs about the relative performance of discrete and continuous action modalities, challenging the prevailing view that precise robot control requires predicting continuous action prediction:

- **Finding 1: Control Precision** Contrary to prior belief, discrete action parametrizations, even discrete regression (DR) with the simplest bin discretization, can exhibit greater control precision than their continuous counterparts, despite the possibly lossy quantization of actions! We support this claim in both sim and real settings in Section 3. We show that this arises from greater policy reactivity, both to the action-chunk time index and the conditioning observation.
- **Finding 2: Iterative Refinement and Guidance Mitigate Factorization Error** Independently predicted tokens can combine incompatible choices across time and action dimensions, even when individual token distributions are accurate. DF solves this by iterative computation, while classifier-free guidance reduces ambiguity by changing token preferences. Both improve control significantly, supporting joint consistency as main explanation for their gains. In the RoboCasa comparison, guidance raises DR success from 27.1% to 50.2%, approaching DF’s 53.8% and CF’s 52.3%, largely ahead of CR’s 38.8%.
- **Finding 3: Explicit Geometry and Structured Tokenization Provide Limited Additional Gains** (Section 5). Encoding action distances or using alternative tokenizers improves some configurations, but none exceeds our best DF result with uniform bins and the standard mixture path. Adjacent bins also acquire similar output embeddings, suggesting that discrete heads can learn local geometry without encoding it explicitly.

We discuss related work in Appendix A.

2 Preliminaries: Action Representations and Prediction Heads

Notation. Let o denote the policy’s conditioning information, such as images, language, and proprioception, and let $\mathbf{a}_{1:H} = (\mathbf{a}_1, \dots, \mathbf{a}_H) \in \mathbb{R}^{H \times M}$ be an action chunk [Zhao et al., 2023] of horizon H , with M action coordinates per timestep. We represent actions in *end-effector (EEF) space*, with 3 translation coordinates $\mathbf{a}^{\text{tr}} \in \mathbb{R}^3$, rotation coordinates $\mathbf{a}^{\text{rot}}$, and a binary gripper coordinate $\mathbf{a}^g \in \{0, 1\}$, and $\mathbf{a}_i = [\mathbf{a}_i^{\text{tr}}, \mathbf{a}_i^{\text{rot}}, \mathbf{a}_i^g]$. All policies use *relative* actions: every target in the chunk is measured from the EEF pose at prediction time. The VLA policies encode the rotation as an axis-angle vector $\mathbf{a}^{\text{rot}} \in \mathbb{R}^3$, and the single-task RoboMimic policies as a 6D representation $\mathbf{a}^{\text{rot}} \in \mathbb{R}^6$ of a rotation matrix in $\text{SO}(3)$. Appendix D.3 defines delta, relative, and absolute actions and compares them for all four heads; relative actions give the highest success for three of the four heads, and absolute actions the lowest for all four. From training pairs $(o, \mathbf{a}_{1:H}) \sim \mathcal{D}$, we learn (possibly stochastic) policies $\hat{\mathbf{a}}_{1:H} \sim \pi_\theta(o)$ that execute the chunk open loop and replan. All heads share a backbone design; full objectives and implementation details are in Appendix C.

Continuous regression (CR). The policy $\pi_{\theta}^{\text{CR}}(o)$ predicts a chunk in one pass using squared-error regression. Its population minimizer is the conditional mean $\mathbb{E}[\mathbf{a}_{1:H} \mid o]$, which need not be a demonstrated action when several behaviors are plausible.

Continuous flow (CF). Flow matching learns a conditional velocity field v_{θ} [Lipman et al., 2022, Liu et al., 2022]. From $\mathbf{x}_0 = \epsilon_C \sim \mathcal{N}(0, \mathbf{I})$, generation follows $\dot{\mathbf{x}}_t = v_{\theta}(\mathbf{x}_t, t, o)$ over $t \in [0, 1]$. After N Euler integration steps, the policy returns $\hat{\mathbf{a}}_{1:H} \leftarrow \mathbf{x}_{t_N}$, with $t_N = 1$. Here, flow time t indexes generation, rather than the action horizon.

Discrete action representation. We quantize each action coordinate into one of B bins after normalization and clipping. Let $[B] = \{1, \dots, B\}$ and $L = HM$. Flattening the quantized chunk in a fixed order gives $\mathbf{z} = (z_1, \dots, z_L) \in [B]^L$, whose entries we call *tokens*. Thus, B is both the number of bins per coordinate and the action-token vocabulary size; unless otherwise stated, $B = 512$. The quantizer $Q : \mathbb{R}^{H \times M} \rightarrow [B]^L$ and decoder $D_Q : [B]^L \rightarrow \mathbb{R}^{H \times M}$ satisfy $\mathbf{z} = Q(\mathbf{a}_{1:H})$ and $\hat{\mathbf{a}}_{1:H} = D_Q(\hat{\mathbf{z}})$. We primarily use **uniform quantization**. For action coordinate i , let $[\ell_i, u_i]$ be its range and $\Delta_i = (u_i - \ell_i)/B$ its bin width. With scalar action $a_{h,i}$ and token position $j = (h-1)M + i$, encoding and decoding are $Q(\mathbf{a}_{1:H})_j := \text{clip}_{[1,B]} \left(1 + \left\lfloor \frac{a_{h,i} - \ell_i}{\Delta_i} \right\rfloor \right)$, $[D_Q(\mathbf{z})]_{h,i} := \ell_i + (z_j - \frac{1}{2})\Delta_i$. Clipping keeps the bin index in $[B]$, including at the upper endpoint; decoding returns the bin center and restores the chunk shape. Both maps use normalized coordinates, with normalization inverted before execution. Recipe details and alternative tokenizations are described in Appendix C.2.

Discrete regression (DR). In one forward pass, the head outputs $L \times B$ logits whose position-wise softmax defines $p_{\theta,j}^{\text{DR}}(\cdot \mid o)$ [Brohan et al., 2022]. Conditioned on the model inputs, it factorizes the token distribution as $p_{\theta}^{\text{DR}}(\mathbf{z} \mid o) = \prod_{j=1}^L p_{\theta,j}^{\text{DR}}(z_j \mid o)$. For $\mathbf{z} = Q(\mathbf{a}_{1:H})$, we train with tokenwise cross-entropy: $\mathcal{L}_{\text{DR}} = \mathbb{E}_{(o, \mathbf{a}_{1:H}) \sim \mathcal{D}} \left[-\frac{1}{L} \sum_{j=1}^L \log p_{\theta,j}^{\text{DR}}(z_j \mid o) \right]$. The factorization makes maximizing the model’s joint probability tractable: we greedily select $\hat{z}_j = \arg \max_{k \in [B]} p_{\theta,j}^{\text{DR}}(k \mid o)$ at each position [Devlin et al., 2019]. Writing $\hat{\mathbf{z}} = (\hat{z}_1, \dots, \hat{z}_L)$, the policy returns $\hat{\mathbf{a}}_{1:H} := D_Q(\hat{\mathbf{z}})$.

Discrete flow (DF). Discrete flow [Gat et al., 2024] generates a token chunk through iterative updates, starting from a product source $q_0(\mathbf{z}_0) = \prod_{j=1}^L q_{0,j}(z_{0,j})$. Let $\alpha : [0, 1] \rightarrow [0, 1]$ be a continuously differentiable, nondecreasing schedule with $\alpha(0) = 0$, $\alpha(1) = 1$, and $\alpha(t) < 1$ for $t < 1$. For a clean chunk $\mathbf{z} = Q(\mathbf{a}_{1:H})$, the conditional probability path is

$$q_t(\mathbf{z}_t \mid \mathbf{z}) = \prod_{j=1}^L [\alpha(t) \mathbf{1}\{z_{t,j} = z_j\} + (1 - \alpha(t)) q_{0,j}(z_{t,j})]. \quad (2.1)$$

At each position, this path interpolates between the source and the clean target token. The source $q_0(\cdot)$ may be uniform over $[B]$ or concentrated on an auxiliary mask token [Sahoo et al., 2024], which does not decode to an action bin. As in DR, the head predicts a clean-token distribution at each position, but additionally conditions on the current chunk and flow time: $p_{\theta,j}^{\text{DF}}(\cdot \mid \mathbf{z}_t, t, o)$. For each training pair, we sample $t \sim \mathcal{U}[0, 1]$ and $\mathbf{z}_t \sim q_t(\cdot \mid \mathbf{z})$, and minimize clean-token cross-entropy [Gat et al., 2024]:

$$\mathcal{L}_{\text{DF}} = \mathbb{E}_{(o, \mathbf{a}_{1:H}) \sim \mathcal{D}, t, \mathbf{z}_t} \left[-\frac{1}{L} \sum_{j=1}^L \log p_{\theta,j}^{\text{DF}}(z_j \mid \mathbf{z}_t, t, o) \right]. \quad (2.2)$$

At inference, initialize $\mathbf{z}_{t_0} \sim q_0$ and use $0 = t_0 < \dots < t_N = 1$. At step n , draw proposals $\tilde{z}_{n,j} \sim p_{\theta,j}^{\text{DF}}(\cdot \mid \mathbf{z}_{t_n}, t_n, o)$ and update each position, writing $\alpha_n = \alpha(t_n)$, as

$$z_{t_{n+1},j} = \begin{cases} \tilde{z}_{n,j}, & \text{with probability } \eta_n, \\ z_{t_n,j}, & \text{otherwise,} \end{cases} \quad \eta_n = \frac{\alpha_{n+1} - \alpha_n}{1 - \alpha_n}. \quad (2.3)$$

Finally, the policy returns $\hat{\mathbf{a}}_{1:H} \leftarrow (o; \epsilon_D) := D_Q(\mathbf{z}_{t_N})$. Importantly, while each single DF step predicts independently per position (as in DR), DF introduces position-wise correlation through multi-stage computation of the final output tokens.

Classifier-free guidance. CFG [Ho and Salimans, 2022] combines conditional and unconditional predictions. Let c be the encoding of the observation o , and let $\emptyset$ be a learned null condition of the same shape. Training replaces c with $\emptyset$ at random, so one head learns both predictions. We write $\emptyset$ in place of o for a prediction made with the null condition. For discrete heads, suppressing other inputs, $p_{\theta,j}^{\text{CFG}}(k | o) \propto p_{\theta,j}(k | o)^{1+\lambda} p_{\theta,j}(k | \emptyset)^{-\lambda}$, normalized over bins, where $\lambda \geq 0$ is the guidance weight ($\lambda = 0$ means no guidance). We write CFG x for $\lambda = x$. DR applies this rule to its single pass and decodes the guided distributions greedily; DF applies it at every refinement step and samples clean-token proposals from them. CF applies the same combination to its velocity, $(1 + \lambda)v_{\theta}(\mathbf{x}_t, t, o) - \lambda v_{\theta}(\mathbf{x}_t, t, \emptyset)$, at every integration step. The backbone encodes o once per inference call, so guidance adds one action-head pass per head evaluation. We do not guide CR: it outputs a single chunk rather than a distribution, so guidance has nothing to sharpen and only extrapolates that chunk. Details are in [Appendix C.5](#).

3 Discrete Action Prediction Can Match Continuous Parametrizations

In this section, we demonstrate that discrete action prediction matches, and often outperforms continuous action prediction *across* both flow and regression policies. To evaluate performance, and refute possible precision loss under action discretization, we measure (1) task success rate under closed-loop deployment, and (2) mean ℓ_2 action prediction error on held-out validation data.

Evaluation Protocol. To demonstrate that our findings are broadly applicable, we consider a wide breadth of tasks; we defer further implementation details to [Appendix B](#).

- **Single-Task Imitation Learning.** To test whether discrete policies’ advantages depend on pretrained VLM, we train separate policies from scratch, without a pretrained VLM backbone, on the Can, Square, Tool Hang and Transport tasks in the RoboMimic suite [Mandlekar et al., 2021]. In particular Tool Hang and Transport are representative of tasks that demand high control precision.
- **VLA Scale Tasks in sim.** We post-train the PaliGemma Beyer et al. [2024] VLM, used by the π -series models Black et al. [2025], Physical Intelligence et al. [2025b], with new action heads corresponding to each action head parametrization. As representatives of VLA training, we pre-train on a mixture of 3 datasets: RoboCasa Nasiriany et al. [2024, 2026], DROID Khazatsky et al. [2024], and MimicGen Mandlekar et al. [2023], and evaluate on each. Evaluations on RoboCasa are grouped into 3 categories: atomic-seen and composite-seen consisting of atomic and composite tasks, respectively, that are contained in the training data; and composite-unseen, consisting of tasks that require compositions of in-distribution atomic-seen, but in novel combinations.
- **Real World VLA tasks.** We used the same backbone and action head setting in sim but trained on different real datasets and SFT on three tasks: throw bottles into bins, open pen caps, and turn mug up. The rollout results are provided in [Fig. 1\(b\)](#)

Results. We depict the closed-loop performance of all four policy parametrizations : CR, CF, DR, DF in [Fig. 1\(b,c\)](#). We see two surprising findings: (i) across **VLA** and **single-task, from-scratch** training, discrete flow is competitive with, and often slightly surpasses, continuous flow. More surprisingly, (ii) discrete regression *consistently outperforms* continuous regression. The fact that discrete *regression* outperforms its continuous counterpart, even on single-task from scratch training, debunks two widely

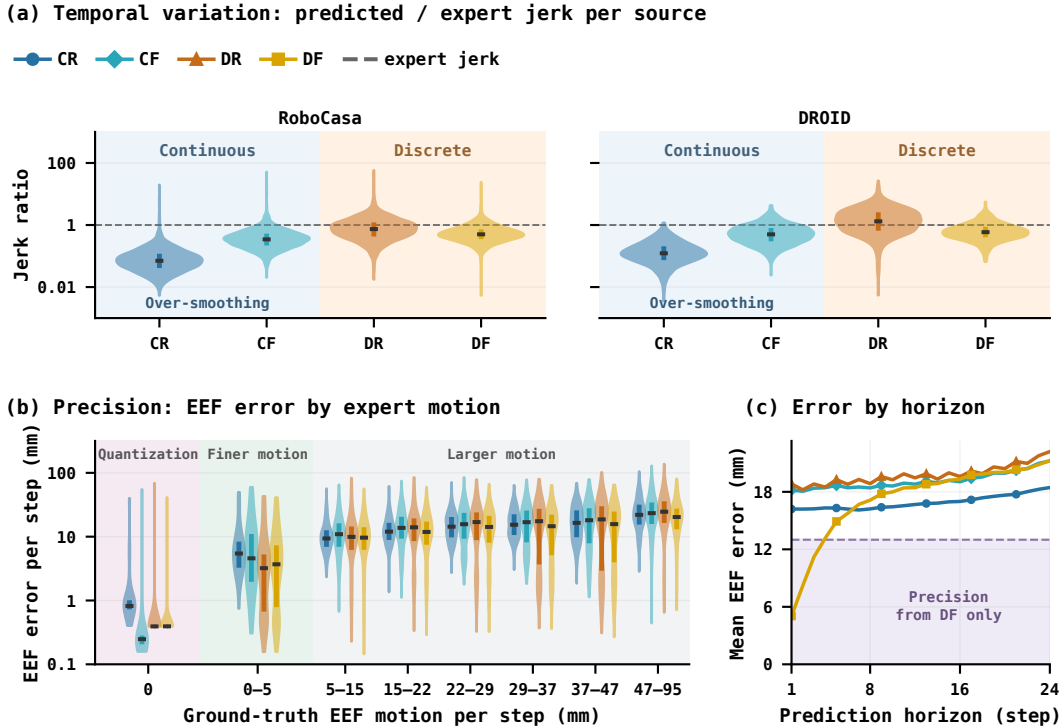


Figure 2: **Temporal variation and precision on held-out chunks.** (a) Jerk-ratio violin plots; Continuous underestimates expert temporal variation. (b) EEf error, grouped by ground-truth motion. “Quantization” highlights the discrete heads’ static-error floor. (c) Mean RoboCasa EEf error by horizon; DF has the lowest error in the first steps of the chunk. More details are in [Appendix E.1](#).

held hypotheses: (a) that precision is a fundamental bottleneck to discrete policies [Shafiullah et al., 2022]; (b) that discrete policies are only beneficial in VLA settings, where it has been hypothesized, that discrete predictions are more compatible with token-based VLM backbones [Brohan et al., 2023, Driess et al., 2025]. As further evidence that discrete policies are not fundamentally limited by precision, we present their end-effector reconstruction error in [Fig. 2](#), showing that discrete policies remain competitive (and in some cases slightly outperforms) their continuous counterparts. In fact, in [Fig. 2](#), we show that discrete flow policies in particular enjoy *lower* end-effector error early on in the action chunk generation and along the finer motion regime, suggesting the potential for greater precision when actions are discrete.

3.1 What Enables the Performance of Discrete Policies?

Paradoxically, our previous findings suggest that the opposite of the common wisdom regarding discrete policies might be true: perhaps, discrete policies are *uniformly better* at high-precision action prediction than their continuous counterparts?

However, this belief is not quite correct. In [Appendix F.2](#), we show find that the ℓ_2 reconstruction errors across all parametrizations are roughly similar, and show *no consistent correlation with task success*. For example, DR exhibits less error than CR on some benchmarks, and more on others, despite *always* enjoying greater closed loop task success.

Temporal Variation, via the Jerk Ratio, Reflects Discrete Policy Competitiveness. Instead, we propose an alternative explanation for the success of discrete policies: that they enable **greater temporal**

variation across action chunks than do their continuous counterparts. To make this rigorous, Fig. 2 (a) depicts each chunk’s **jerk ratio**.¹ For a given action chunk $\mathbf{a}_{1:H}$, let $\mathbf{a}_h^{\text{tr}}$ denote the translation coordinates of the end-effector. We define the difference approximation to the jerk at step h as $\Delta^3 \mathbf{a}_h^{\text{tr}} = \mathbf{a}_{h+3}^{\text{tr}} - 3\mathbf{a}_{h+2}^{\text{tr}} + 3\mathbf{a}_{h+1}^{\text{tr}} - \mathbf{a}_h^{\text{tr}}$, $h = 1, \dots, H-3$. The chunk’s mean jerk magnitude and predicted-to-target jerk ratio are

$$\text{Jerk}(\mathbf{a}_{1:H}) = \frac{1}{H-3} \sum_{h=1}^{H-3} \|\Delta^3 \mathbf{a}_h^{\text{tr}}\|_2, \quad R_{\text{jerk}} = \frac{\text{Jerk}(\hat{\mathbf{a}}_{1:H})}{\text{Jerk}(\mathbf{a}_{1:H})}. \quad (3.1)$$

Thus, jerk ratio of $R_{\text{jerk}} = 1$ means that the learned policy’s jerk matches the ground-truth jerk on a training example; $R_{\text{jerk}} < 1$ means jerk is underestimated, so that the policy has *oversmoothed* the training data with respect to time.

Across all datasets, Fig. 2 (a) reveals that continuous regression enjoys a jerk ratio substantially less than 1, whereas discrete regression is close to (and occasionally above) 1. This suggests that discrete action modalities can more easily represent large temporal variation in a single pass, likely due to the probabilistic *decoupling* of predictions across time steps. These differences reduce under iterative computation, as Fig. 2 (a) shows CF and DF have much closer jerk ratios than do the single-pass counterparts. To identify differences that persist between DF and CF, Fig. 2 (b,c) reveal that discrete policies better fit other subtle variations: in (b), we see that discrete modalities offer more response to smaller degrees of EEf movement, especially the 0–5 mm range (note the logarithmic plot scale); in (c) we see that DF in particular fits actions earlier in the chunk more accurately, which require smaller relative motion. We hypothesize that continuous objectives underweight/over-smooth small action differences compared to categorical prediction. In Appendix H.1, we corroborate these hypotheses in controlled studies of fitting synthetic functions with varying levels of smoothness.

DR/DF Exhibit Greater Responsiveness to Conditioning Changes than CR/CF

Beyond temporal variation within a chunk, we examine responsiveness to observation changes, formally $\frac{\partial \pi}{\partial o}$. Fig. 3 shows that both continuous heads produce smaller changes in their mean predicted chunks than their discrete counterparts, underestimating expert change magnitudes on held-out data. DR tracks the expert response most closely, while DF remains closer than CF at every tested state displacement. On the other hand, additional diagnostics in Appendix E.3 also show that CF stays closer to nearby training actions than DF, corroborating findings from Pan et al. [2026]. These results suggest that stronger adherence to the training data may come at the cost of responsiveness. DF may offer a better balance, retaining the relative stability under disturbances observed for flow heads while responding more closely to expert changes.

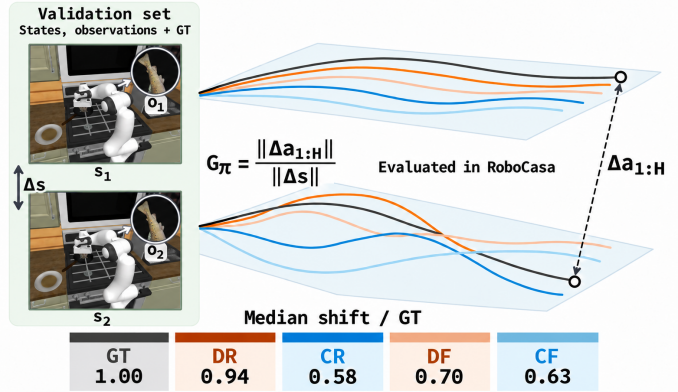


Figure 3: Median state displacements and chunk changes (normalized units). Model columns show raw proposal-mean shifts; sampling floors are not subtracted.

Takeaway 1: Discrete policies generally **match or exceed** the control performance of their continuous counterparts across the evaluated testbeds. They obtain stronger performance in both VLA and from-scratch settings. These gains arise from **better capturing variation** across within-chunk time steps and conditioning observations, rather than outright reconstruction error.

¹Recall that *jerk* refers to a the third-time derivative of its actions, and thus reflects the *variability* of actions during a given action chunk.

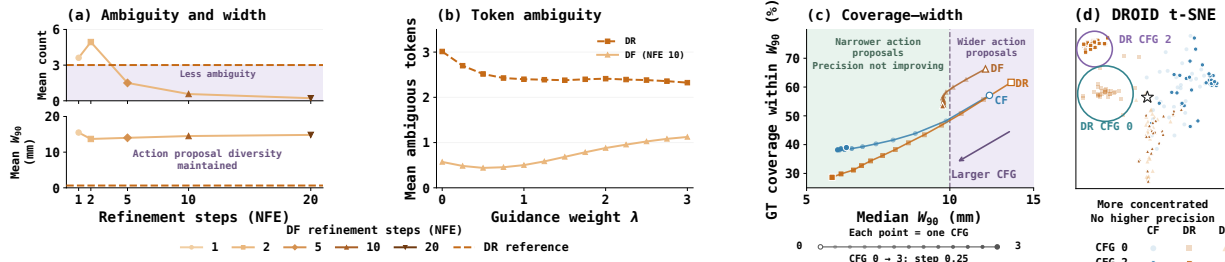


Figure 4: **Iterative refinement reduces token ambiguity.** (a,b) At CFG 0, DF at NFE 1, 2, 5, 10, 20 show separated-mode prevalence (a) ambiguity decreasing, action diversity remains $\exp(\mathcal{H}(p))$ (b); the orange dashed curve is DR. (c) Median central-90% width W_{90} of action proposal and its ground truth coverage, run from CFG 0 to 3. (d) DROID t-SNE; the star marks the GT. ; additional diagnostics and protocols are in [Appendix F.1](#).

4 Factorization Error and Iterative Computation in Discrete Policies

Thus far, [Section 3](#) establishes that discrete regression (DR) outperforms continuous regression (CR), in part due to better capturing temporal variation, subtle action changes, and response to the input variables. In this section, we show that the greater temporal reactivity observed in DR comes at the cost of action inconsistency, or what we call **factorization error** (Finding A), harming policy performance. We then identify that the key interventions which enhance performance—applying classifier-free-guidance (Finding B) and introducing iterative computation by moving to discrete flow (Finding C) —correct this inconsistency via complementary mechanisms described below.

Factorization Error in Discrete Policies. DR predicts a categorical distribution for each action token, giving the factorized distribution $p_{\theta}(\mathbf{z} \mid o) = \prod_j p_{\theta,j}(z_j \mid o)$. When an observation admits several plausible behaviors, different token distributions can assign probability to alternatives behaviors independently. These independent choices can combine incompatible alternatives across time steps and action dimensions. We refer to this failure to preserve dependencies between token choices as **factorization error**. This may occur even when every token marginal is learned correctly, because the product of the marginals need not recover the joint distribution of demonstrated chunks.

Evaluation Protocol for Factorization Error. To isolate token ambiguity as a potential source of incompatible action tokens, we vary (i) the number of function evaluations (NFE), that is, the number of refinement steps N in DF, comparing to DR as a baseline, and (ii) the guidance weight λ of classifier-free guidance (CFG, [Section 2](#)). Note that greater NFE provides more opportunity for self-correction, whereas greater CFG “sharpens” the generated action distribution. Due to space constraints, policy performance under varying CFG/NFE is reported in [Appendix F.3](#); in short, performance improves with NFE and an appropriate guidance weight.

Here, we focus on two metrics, deferring precise details to [Appendix F.1](#): **1. Mode-mixture:** we identify distributions with at least two “incompatible” modes in the same action chunk. We report the median count of such modes within one chunk across NFE [Fig. 4\(a, top\)](#) and across guidance weights [Fig. 4\(b\)](#). **2. Translation Quantiles.** Next, we measure the *diversity* of actions generated across repeated policy samples from the same observation. This lets us test (and ultimately rule out) an alternative hypothesis that DF improves performance over DR by better capturing multimodality. Specifically, we measure the central-90% interval width $W_{90} = Q_{0.95} - Q_{0.05}$ for each of the three translation coordinates per time step, and take the Euclidean norm of the 3 per-axis widths. We plot the norm of median widths against NFE in [Fig. 4\(a, bottom\)](#) and guidance weight in [Fig. 4\(c\)](#). Lastly, [Fig. 4\(d\)](#) qualitatively visualizes action diversity.

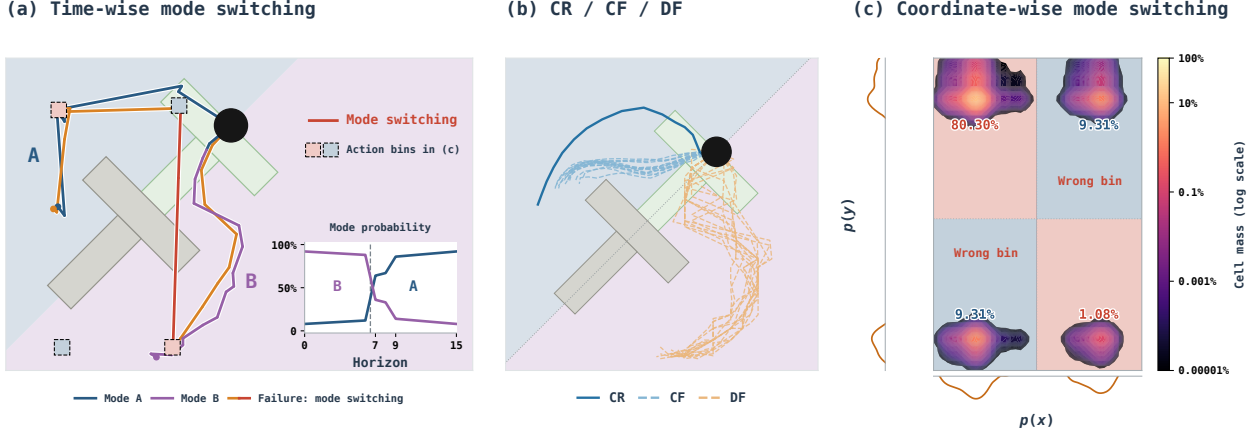


Figure 5: **(a)** Temporal mode switching within an action chunk as relative mode probabilities vary across the prediction horizon; **(b)** Action chunks predicted by CR, CF, and DF from the same starting position. **(c)** Coordinate-wise factorization within a timestep: independently combining coordinates from two valid modes can produce two additional, incompatible actions.

Finding A: Factorization Error Across Horizon and Coordinates Can Limit DR. In Fig. 4(a), we show that DR suffers from more mode-mixture than DF, even worse at zero CFG. As further evidence, Fig. 5 illustrates the mechanism in the simple Push-T environment. Here, DR assigns probability to separated movement alternatives, and their relative mass changes across the prediction horizon. Appendix F.4 presents the analogous ambiguity in RoboCasa. These examples confirm the quantitative findings in Fig. 4, illustrating how factorized decoding can introduce *spurious* temporal variation, even when individual token predictions capture plausible alternatives.

Finding B: Classifier-Free Guidance Reduces Ambiguity and Makes DR Competitive with DF As further evidence of the presence of factorization error, we observe that increasing guidance from CFG 0 to CFG 2 raises pooled RoboCasa success from 27.1% to 49.2% (Appendix F.2). This coincides with drastic reduction in the mean-width of the confidence intervals (Fig. 4(c)), which is consistent with reductions in spurious temporal variation. As noted above, guidance is known to “sharpen” the resulting generated distribution [Ho and Salimans, 2022, Karras et al., 2024], and thus is likely to be resolving mode ambiguity. Moreover, Fig. 4(c,d) shows that CFG reduces coverage of plausible ground-truth EEF positions, suggesting that performance advantages from CFG primarily reflect reduced factorization error rather than improved prediction accuracy. Across sources, CFG almost never improves EEF error by more than 1 mm (Appendix E.2).

Finding C: Iterative Computation Aids Discrete Policies via Reducing Factorization Error. While CFG guidance is a partial-yet-imperfect remedy of factorization error, Here we conclude we show that iterative computation, as in DF, reduces factorization error further. To support this claim, we find that DF reduces the number of inconsistent-mode predictions per chunk by a factor of 5.28 relative to DR, bringing the average below one at NFE ≥ 10 (Fig. 4(a,top)). This coincides with a 4.7% increase in RoboCasa success (Appendix F.3). These suggest that iterative computation makes discrete action predictions more cohesive than guidance alone. We note that, for continuous action predictions, the benefits of iterative computation are somewhat *opposite*, combating over-smoothing (Fig. 2) and policy under-reactivity (Fig. 5). The high level role of iterative computation, however, is the same, in that it performs self-correction on policy actions [Pan et al., 2026].

Does action diversity explain the performance of DF? (Likely not) Despite exhibiting low factorization error, DF exhibits *greater action diver-*

Table 1: EEF error and RoboCasa success

CFG 0 $\rightarrow$ 2	AR	DR (ref.)	DF (ref.)
$\mathbb{E}$ EEF (mm) $\downarrow$	18.4 $\rightarrow$ 24.4	16.4 $\rightarrow$ 16.3	14.1 $\rightarrow$ 18.2
Success (%) $\uparrow$	42.8 $\rightarrow$ 48.3	27.1 $\rightarrow$ 49.2	52.3 $\rightarrow$ 53.9

sity than DR, as shown in all Fig. 4 plots. However, this is unlikely to explain its greater performance. CFG 2 notably reduces DF action diversity, and reduces action diversity in CF and DR considerably more; yet in all three, performance increases, most notably in DR (see Appendix F.3). Moreover, although performance improves up to NFE= 10, action diversity remains relatively constant (Fig. 4(a,bottom)). These findings suggest that preservation of full action diversity in the policy is not essential for task success.

Can temporal autoregression resolve factorization error? Temporal autoregression provides another way to coordinate action choices across time. We implement a discrete autoregressive (AR) policy that predicts the first action from the observation, then predicts each subsequent action conditioned on the observation and all previously generated actions. Coordinates within each time step are predicted in parallel, requiring 24 sequential predictions for a 24-step chunk. Without CFG, temporal AR achieves 42.8% success on RoboCasa, exceeding DR’s 27.1% but remaining below DF. However, with CFG, the AR slightly trails both DR and DF (Tab. 1).

Takeaway 2: We identify **factorization error**, a form of temporal inconsistency, as the cause of DR’s relative lack of performance compared to CF and DF. Iterative computation in DF yields maximal performance, but with appropriate CFG, DR’s performance is **surprising close**; this reveals that iterative computation and distribution sharpening are both effective means for combating factorization errors.

5 Additional Structure Provides Limited Gains for Discrete Policies

Section 4 links the gains from iterative refinement and guidance to reduced ambiguity in discrete action predictions. In this section, we ask *can explicit action geometry or structured tokenization further improve on simple categorical prediction with uniform bins?* We now consider two mechanism for adding structure: encoding metric distances between action values during the DF corruption process, or changing how action chunks are tokenized.

Explicit Geometry Provides Limited Control Gains. We introduce action geometry in two ways. For DR, distance-based label smoothing assigns target probability to nearby action bins as well as the demonstrated bin, with probability decreasing with distance. We combine this with logit soft-capping to limit confidence. For DF, we replace the uniform-source mixture path in Eq. (2.1) with a metric-induced corruption path: $q_t^{\text{metric}}(\mathbf{z}_t \mid \mathbf{z}) = \prod_{j=1}^L \frac{\exp[-\beta(\alpha(t))d(\mathbf{z}_{t,j}, \mathbf{z}_j)]}{\sum_{b=1}^B \exp[-\beta(\alpha(t))d_p(b, \mathbf{z}_j)]}$, where $d(k, b) = |k - b| / ((B - 1)/2)$, and β is an increasing concentration schedule. As generation progresses, the metric path concentrates corrupted tokens near the clean target, whereas the mixture path keeps all incorrect bins equally likely (Appendix G.2). Both DF variants retain the clean-token prediction objective in Eq. (2.2). Thus, label smoothing changes DR’s prediction targets, whereas the metric path changes DF’s corrupted inputs. The schedules and sampling rates are provided in Appendix G.2. The tested DR variant improves RoboCasa success by 2.1%. The best tested metric-path DF reaches 52.8% pooled success, compared with 53.9% for mixture-path DF. Explicit geometry therefore provides a modest gain for DR but does not improve the best observed DF result.

Structured Tokenizers Do Not Improve over Naive Binning. We next test whether changing the action representation improves control. We evaluate Finite Scalar Quantization(nFSQ) [Mentzer et al., 2024], residual vector quantization (RVQ) [Zeghidour et al., 2021], and FAST [Pertsch et al., 2025], alongside quantile-bin baselines. At their best recorded complete operating points, DF reaches 52.4% RoboCasa SR with nFSQ, 46.2% with FAST, and 38.2% with RVQ (Appendix G.1). nFSQ and quantile representation improves over the flat-bin baseline for DR, showing that the tokenizer can matter for a

given configuration. However, none of these exceeds the broader campaign’s best flat-bin DF result of 53.9%. The observed gains limited on DR, and do not raise the best recorded performance.

Discrete Policies Can Learn Local Bin Geometry. Why does encoding structural information yield little benefit? We hypothesize that discrete policy encode geometry through the **embedding vectors** associated with the last logit layer. Specifically, each bin b at coordinate j associated with a last-layer embedding $\mathbf{v}_{j,b}$. For spatially adjacent bins $b, b + 1$, we compute the vector cosine $\cos \theta_{j,b} := \langle \mathbf{v}_{j,b}, \mathbf{v}_{j,b+1} \rangle / (\|\mathbf{v}_{j,b}\| \|\mathbf{v}_{j,b+1}\|)$. We find that adjacent bins exhibit high cosine similarities. These observations suggest discrete policies do not treat distinct bins as isolated categories, and offers one possible explanation for the limited benefit of imposing that geometry explicitly. Embedding plots, probes, and the selected-checkpoint comparison are provided in [Appendix G.3](#).

Takeaway 3: Explicit geometry and structured tokenization can improve individual configurations, but neither exceeds the best configuration of DF with uniform masking and flat-bin tokenization. Learned local geometry offers a possible explanations for these limited gains.

6 Discussion and Future Work

We began by asking whether robot actions are fundamentally different from other AI modalities in requiring continuous prediction. Our results challenge this distinction. Although robots operate in a continuous physical world, simple action tokens can support precise control in both simulation and real-world tasks. Their success shows that discrete action prediction is more than a convenient interface to a language model. It is an effective way to learn robot control in its own right.

The key difficulty exposed by our experiments is deciding which token choices belong together. Discrete heads readily express action variation, but independent predictions can combine incompatible behaviors. Iterative refinement and guidance help resolve these conflicts while retaining responsiveness. This connects action prediction to a challenge also studied in diffusion language models: accurate token predictions must be coordinated to form a coherent joint output.

The need for joint consistency also suggests a different objective for action tokenization. Although the tested reconstruction-based tokenizers do not improve on our best uniform-bin DF result, tokenizers designed to make coupled action choices easier to predict may help reduce factorization error. For example, tokens could jointly encode coherent motion across several time steps. Their value may depend on the prediction head and the number of generation steps available. Finding tokenizations suited to discrete regression, iterative flow, or generation with only a few refinement steps is therefore a promising direction. Such tokenizers should be evaluated by the consistency and control performance of the resulting policies, alongside reconstruction quality.

More broadly, these findings support learning action generation within the same token-based framework as perception and reasoning. A natural next step is to test whether sharing representations and prediction machinery across modalities improves co-training and transfer, while preserving the action consistency needed for closed-loop control. These results make a strong case for treating discrete action prediction as a central component of future robot foundation models.

AI Use Statement

We used generative AI tools to help implement methods, analyze results, make figures, and edit the paper. We did not use them to form hypotheses or design the experiments. We have reviewed all AI-assisted work and take responsibility for the final content of this paper.

Reproducibility Statement

The evaluation protocol and environments are described in [Appendix B](#), and the action heads, training objectives, architectures, and training recipes in [Appendix C](#). Full results and selected settings are in [Appendix D](#). We will release code and model checkpoints upon publication.

References

- Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022. URL <https://arxiv.org/abs/2209.15571>.
- Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pages 17981–17993, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/958c530554f78bcd8e97125b70e6973d-Paper.pdf.
- Suneel Belkhale and Dorsa Sadigh. MiniVLA: A better VLA with a smaller footprint. Stanford AI Lab Blog, <https://ai.stanford.edu/blog/minivla/>, 2024.
- Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pages 449–458, 2017.
- Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.
- Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0 : A vision-language-action flow model for general robot control. In *Robotics: Science and Systems*, 2025. URL <https://arxiv.org/abs/2410.24164>.
- Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Praseen Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016. URL <https://arxiv.org/abs/1604.07316>.
- Arwen Bradley and Preetum Nakkiran. Classifier-free guidance is a predictor-corrector. *Transactions on Machine Learning Research*, 2025.
- Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. URL <https://arxiv.org/abs/2212.06817>.
- Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, 2023. URL <https://arxiv.org/abs/2307.15818>.
- Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising mod-

- els. In *Advances in Neural Information Processing Systems*, volume 35, pages 28266–28279, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b5b528767aa35f5b1a60fe0aaeca0563-Paper-Conference.pdf.
- Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 5453–5512, 2024. URL <https://proceedings.mlr.press/v235/campbell124a.html>.
- Haoqun Cao and Tengyang Xie. Understanding behavior cloning with action quantization. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306, pages 11606–11640, 2026. URL <https://arxiv.org/abs/2603.20538>.
- Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. MaskGIT: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11315–11325, 2022. URL <https://arxiv.org/abs/2202.04200>.
- Huiwen Chang, Han Zhang, Jarred Barber, Aaron Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Patrick Murphy, William T. Freeman, Michael Rubinstein, Yuanzhen Li, and Dilip Krishnan. Muse: Text-to-image generation via masked generative transformers. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 4055–4075, 2023. URL <https://proceedings.mlr.press/v202/chang23b.html>.
- Yevgen Chebotar, Quan Vuong, Alex Irpan, Karol Hausman, Fei Xia, Yao Lu, Aviral Kumar, Tianhe Yu, Alexander Herzog, Karl Pertsch, Keerthana Gopalakrishnan, Julian Ibarz, Ofir Nachum, Sumedh Son-takke, Grecia Salazar, Huong T. Tran, Jodilyn Peralta, Clayton Tan, Deeksha Manjunath, Jaspiar Singh, Brianna Zitkovich, Tomas Jackson, Kanishka Rao, Chelsea Finn, and Sergey Levine. Q-Transformer: Scalable offline reinforcement learning via autoregressive q-functions. In *Conference on Robot Learning*, 2023.
- Jiayi Chen, Wenxuan Song, Pengxiang Ding, Ziyang Zhou, Han Zhao, Feilong Tang, Donglin Wang, and Haoang Li. Unified diffusion VLA: Vision-language-action model via joint discrete denoising diffusion process. In *International Conference on Learning Representations*, pages 139291–139311, 2026a. URL <https://arxiv.org/abs/2511.01718>.
- Jiayi Chen, Wenxuan Song, Jiaxin Fang, Ruiqing Yin, Jingbo Wang, Shuai Chen, Jieyuan Pei, Yikai Qin, Feifan Chen, Haodong Yan, Zhide Zhong, Wen Chen, Yan Wang, Yuxiang Gao, and Haoang Li. DFM-VLA: Iterative action refinement for robot manipulation via discrete flow matching. *arXiv preprint arXiv:2603.26320*, 2026b.
- Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023. URL <https://arxiv.org/abs/2303.04137>.
- Muthu Chidambaram, Khashayar Gatmiry, Sitan Chen, Holden Lee, and Jianfeng Lu. What does guidance do? A fine-grained analysis in a simple setting. In *Advances in Neural Information Processing Systems*, 2024.
- Eugenio Chisari, Nick Heppert, Max Argus, Tim Welschehold, Thomas Brox, and Abhinav Valada. Learning robotic manipulation policies from point clouds with conditional flow matching. *arXiv preprint arXiv:2409.07343*, 2024. URL <https://arxiv.org/abs/2409.07343>.

- Jiwan Chung, SAEJIN KIM, Yongrae Jo, Jaewoo Park, Dongjun Min, and Youngjae Yu. Teaching metric distance to discrete autoregressive language models. In *International Conference on Learning Representations*, pages 11633–11655, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/13b45b44e26c353c64cba9529bf4724f-Paper-Conference.pdf.
- Zichen Jeff Cui, Yibin Wang, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. From play to policy: Conditional behavior generation from uncurated robot data. In *International Conference on Learning Representations*, 2023.
- Chenxu Dang, Sining Ang, Yongkang Li, Haochen Tian, Jie Wang, Guang Li, Hangjun Ye, Jie Ma, Long Chen, and Yan Wang. DriveFine: Refining-augmented masked diffusion VLA for accurate and robust driving. In *European Conference on Computer Vision (ECCV)*, 2026. URL <https://arxiv.org/abs/2602.14577>.
- Haoge Deng, Ting Pan, Fan Zhang, Yang Liu, Zhuoyan Luo, Yufeng Cui, Wenxuan Wang, Chunhua Shen, Shiguang Shan, Zhaoxiang Zhang, and Xinlong Wang. Uniform discrete diffusion with metric path for video generation. In *International Conference on Learning Representations*, pages 135312–135327, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/daf8364f0715a41a469c677c0adc4754-Paper-Conference.pdf.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019. URL <https://api.semanticscholar.org/CorpusID:52967399>.
- Raúl Díaz and Amit Marathe. Soft labels for ordinal regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4733–4742, 2019. doi: 10.1109/CVPR.2019.00487.
- Danny Driess, Jost Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, et al. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. In *Advances in Neural Information Processing Systems*, volume 38, pages 102867–102888, 2025.
- Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023.
- Jesse Farebrother, Jordi Orbay, Quan Vuong, Adrien Ali Taïga, Yevgen Chebotar, Ted Xiao, Alex Irpan, Sergey Levine, Pablo Samuel Castro, Aleksandra Faust, Aviral Kumar, and Rishabh Agarwal. Stop regressing: Training value functions via classification for scalable deep RL. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.
- Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025. URL <https://arxiv.org/abs/2510.13626>.
- Pete Florence, Corey Lynch, Andy Zeng, Oscar Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 158–168, 2021.
- Bin-Bin Gao, Chao Xing, Chen-Wei Xie, Jianxin Wu, and Xin Geng. Deep label distribution learning with label ambiguity. *IEEE Transactions on Image Processing*, 26(6):2825–2838, 2017. doi: 10.1109/TIP.2017.2689998.

- Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In *Advances in Neural Information Processing Systems*, volume 37, pages 133345–133385, 2024. URL <https://arxiv.org/abs/2407.15595>.
- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.
- Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025. doi: 10.1038/s41586-025-08744-2.
- Siddhant Haldar, Zhuoran Peng, and Lerrel Pinto. BAKU: An efficient transformer for multi-task policy learning. In *Advances in Neural Information Processing Systems*, 2024.
- Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024.
- Haoran He, Chenjia Bai, Ling Pan, Weinan Zhang, Bin Zhao, and Xuelong Li. Learning an actionable discrete diffusion policy via large-scale actionless video pre-training. In *Advances in Neural Information Processing Systems*, volume 37, pages 31124–31153, 2024. URL <https://arxiv.org/abs/2402.14407>.
- Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. URL <https://arxiv.org/abs/2207.12598>.
- Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Advances in Neural Information Processing Systems*, volume 34, pages 12454–12465, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/67d96d458abdef21792e6d8e590244e7-Paper.pdf.
- Zhi Hou, Tianyi Zhang, Yuwen Xiong, Haonan Duan, Hengjun Pu, Ronglei Tong, Chengyang Zhao, Xizhou Zhu, Yu Qiao, Jifeng Dai, and Yuntao Chen. Dita: Scaling diffusion transformer for generalist vision-language-action policy. *arXiv preprint arXiv:2503.19757*, 2025.
- Xixi Hu, Bo Liu, Xingchao Liu, and Qiang Liu. AdaFlow: Imitation learning with variance-adaptive flow-based policies. *arXiv preprint arXiv:2402.04292*, 2024. URL <https://arxiv.org/abs/2402.04292>.
- Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner monologue: Embodied reasoning through planning with language models. In *arXiv preprint arXiv:2207.05608*, 2022.
- Ehsan Imani and Martha White. Improving regression performance with distributional losses. In *International Conference on Machine Learning*, pages 2162–2171, 2018.
- Ehsan Imani, Kai Luedemann, Sam Scholnick-Hughes, Esraa Elelimy, and Martha White. Investigating the histogram loss in regression. *Journal of Machine Learning Research*, 27(66):1–54, 2026.
- Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/5ee7ed60a7e8169012224dec5fe0d27f-Abstract-Conference.html.

- Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, Vitor Guizilini, David Antonio Herrera, Minh Heo, Kyle Hsu, Jiaheng Hu, Muhammad Zubair Irshad, Donovan Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. DROID: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems (RSS)*, 2024.
- Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Paul Foster, Grace Tin Yan Lam, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning*, 2024. URL <https://arxiv.org/abs/2406.09246>.
- Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025. URL <https://arxiv.org/abs/2502.19645>.
- Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos Policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026. URL <https://arxiv.org/abs/2601.16163>.
- Seungjae Lee, Yibin Wang, Haritheja Etukuru, H. Jin Kim, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Behavior generation with latent actions. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 26991–27008, 2024.
- Sung-Wook Lee, Xuhui Kang, and Yen-Ling Kuo. TBD-VLA: Temporal block diffusion vision language action model. *arXiv preprint arXiv:2606.07895*, 2026. URL <https://arxiv.org/abs/2606.07895>.
- Ian Li, Zilei Shao, Benjie Wang, Rose Yu, Guy Van den Broeck, and Anji Liu. Breaking the factorization barrier in diffusion language models. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 69716–69732, 2026a. URL <https://proceedings.mlr.press/v306/li26fd.html>.
- Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026b. URL <https://arxiv.org/abs/2601.21998>.

- Pengxiang Li, Yinan Zheng, Yue Wang, Huimin Wang, Hang Zhao, Jingjing Liu, Xianyuan Zhan, Kun Zhan, and Xianpeng Lang. Discrete diffusion for reflective vision-language-action models in autonomous driving. In *International Conference on Learning Representations*, 2026c.
- Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, Xiaofan Wang, Bei Liu, Jianlong Fu, Jianmin Bao, Dong Chen, Yuanchun Shi, Jiaolong Yang, and Baining Guo. CogACT: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024a.
- Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. In *Robotics: Science and Systems (RSS)*, 2025. URL <https://arxiv.org/abs/2503.00200>.
- Xinghang Li, Peiyan Li, Long Qian, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Xinlong Wang, Di Guo, Tao Kong, Hanbo Zhang, and Huaping Liu. What matters in building vision-language-action models for generalist robots. *arXiv preprint arXiv:2412.14058*, 2024b.
- Haotian Liang, Xinyi Chen, Bin Wang, Mingkang Chen, Yitian Liu, Yuhao Zhang, Zhanxin Chen, Tianshuo Yang, Yilun Chen, Jiangmiao Pang, Dong Liu, Xiaokang Yang, Yao Mu, Wenqi Shao, and Ping Luo. MM-ACT: Learn from multimodal parallel generation to act. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 35080–35090, 2026a. URL <https://arxiv.org/abs/2512.00975>.
- Zhixuan Liang, Yizhuo Li, Tianshuo Yang, Chengyue Wu, Sitong Mao, Liuaao Pei, Tian Nian, Shunbo Zhou, Xiaokang Yang, Jiangmiao Pang, Yao Mu, and Ping Luo. Discrete diffusion VLA: Bringing discrete diffusion to action decoding in vision-language-action policies. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306, pages 73258–73274, 2026b. URL <https://arxiv.org/abs/2508.20072>.
- Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. URL <https://arxiv.org/abs/2210.02747>.
- Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.
- Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, Chengkai Hou, Mengdi Zhao, KC alex Zhou, Pheng-Ann Heng, and Shanghang Zhang. HybridVLA: Collaborative diffusion and autoregression in a unified vision-language-action model. In *International Conference on Learning Representations*, 2026a.
- Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: a diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations*, 2025.
- Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. URL <https://arxiv.org/abs/2209.03003>.
- Yang Liu, Pengxiang Ding, Tengyue Jiang, Xudong Wang, Wenxuan Song, Minghui Lin, Han Zhao, Hongyin Zhang, Zifeng Zhuang, Wei Zhao, Siteng Huang, Jinkui Shi, and Donglin Wang. MMaDA-VLA: Large diffusion vision-language-action model with unified multi-modal instruction and generation. In *Proceedings of the 34th ACM International Conference on Multimedia*, 2026b. URL <https://arxiv.org/abs/2603.25406>.

- Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 32819–32848, 2024. URL <https://proceedings.mlr.press/v235/lou24a.html>.
- Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning*, 2021. URL <https://arxiv.org/abs/2108.03298>.
- Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Ireteyayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. MimicGen: A data generation system for scalable robot learning using human demonstrations. In *7th Annual Conference on Robot Learning*, 2023.
- Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: VQ-VAE made simple. In *International Conference on Learning Representations*, volume 2024, pages 51772–51783, 2024.
- Atharva Mete, Haotian Xue, Albert Wilcox, Yongxin Chen, and Animesh Garg. QueST: Self-supervised skill abstractions for learning continuous control. In *Advances in Neural Information Processing Systems*, 2024.
- Luke Metz, Julian Ibarz, Navdeep Jaitly, and James Davidson. Discrete sequential prediction of continuous actions for deep RL. *arXiv preprint arXiv:1705.05035*, 2017.
- Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems (RSS)*, 2024.
- Soroush Nasiriany, Sepehr Nasiriany, Abhiram Maddukuri, and Yuke Zhu. RoboCasa365: A large-scale simulation framework for training and benchmarking generalist robots. In *International Conference on Learning Representations (ICLR)*, 2026.
- Khang Nguyen, An T. Le, Tien Pham, Manfred Huber, Jan Peters, and Minh Nhat Vu. FlowMP: Learning motion fields for robot planning with conditional flow matching. *arXiv preprint arXiv:2503.06135*, 2025. URL <https://arxiv.org/abs/2503.06135>.
- Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *Advances in Neural Information Processing Systems*, volume 38, pages 50608–50646, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/48b383b24230e0e6e649d9c98dae4d8c-Paper-Conference.pdf.
- Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking guidance for discrete state-space diffusion and flow models. In *International Conference on Learning Representations*, pages 36052–36106, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/597254dc45be8c166d3ccf0ba2d56325-Paper-Conference.pdf.
- NVIDIA, Johan Bjorck, Fernando Castañeda, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. URL <https://arxiv.org/abs/2503.14734>.
- Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag

- Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Robotics: Science and Systems*, 2024.
- OpenAI, Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39:3–20, 2020. doi: 10.1177/0278364919887447.
- Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J. Andrew Bagnell, Pieter Abbeel, and Jan Peters. An algorithmic perspective on imitation learning. *Foundations and Trends in Robotics*, 2018. URL <https://arxiv.org/abs/1811.06711>.
- Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *International Conference on Learning Representations*, pages 64972–65009, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/a365e37c18fb91af547a2f0012a89e98-Paper-Conference.pdf.
- Chaoyi Pan, Giri Anantharaman, Nai-Chieh Huang, Claire Jin, Daniel Pfrommer, Chenyang Yuan, Frank Permenter, Guannan Qu, Nicholas Boffi, Guanya Shi, and Max Simchowitz. Much ado about noising: Dispelling the myths of generative robotic control. In *International Conference on Learning Representations*, pages 90575–90614, 2026. URL <https://arxiv.org/abs/2512.01809>.
- Minho Park, Kinam Kim, Junha Hyung, Hyojin Jang, Hoiyeong Jin, Jooyeol Yun, Hojoon Lee, and Jaegul Choo. ACG: Action coherence guidance for flow-based vision-language-action models. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 9943–9950, 2026. doi: 10.1109/ICRA57385.2026.11695974.
- Tim Pearce, Tabish Rashid, Anssi Kanervisto, Dave Bignell, Mingfei Sun, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Ida Momennejad, Katja Hofmann, and Sam Devlin. Imitating human behaviour with diffusion models. In *International Conference on Learning Representations*, 2023.
- Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025. URL <https://arxiv.org/abs/2501.09747>.
- Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, Danny Driess, et al. $\pi_{0.6}^*$: a VLA that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025a.
- Physical Intelligence et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025b. URL <https://arxiv.org/abs/2504.16054>.
- Dean A. Pomerleau. ALVINN: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems*, volume 1, 1988. URL https://papers.neurips.cc/paper_files/paper/1988/file/812b4ba287f5ee0bc9d43bbf5bbe87fb-Paper.pdf.
- Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent. *Transactions on Machine Learning Research*, 2022.
- Moritz Reuss, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. In *Robotics: Science and Systems*, 2023.

- Kevin Rojas, Ye He, Chieh-Hsin Lai, Yuhta Takida, Yuki Mitsufuji, and Molei Tao. Improving classifier-free guidance in masked diffusion: Low-dim theoretical insights with high-dim impact. In *International Conference on Learning Representations*, pages 33722–33761, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/391f1a5319c929173bddce4cd9367056-Paper-Conference.pdf.
- Stephane 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*, 2011. URL <https://proceedings.mlr.press/v15/ross11a.html>.
- Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 130136–130184, 2024. URL <https://arxiv.org/abs/2406.07524>.
- Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin T Chiu, and Volodymyr Kuleshov. The diffusion duality. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 52584–52619, 2025. URL <https://proceedings.mlr.press/v267/sahoo25a.html>.
- Guillaume Sanchez, Alexander Spangher, Honglu Fan, Elad Levi, and Stella Biderman. Stay on topic with classifier-free guidance. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 43197–43234, 2024.
- Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dalla-torre, Bernardo P. de Almeida, Alexander Rush, Thomas Pierrot, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. In *International Conference on Learning Representations*, pages 43776–43821, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/6cc31b44d88dce8380d36e81485cd07f-Paper-Conference.pdf.
- Tim Seyde, Igor Gilitschenski, Wilko Schwarting, Bartolomeo Stellato, Martin Riedmiller, Markus Wulfmeier, and Daniela Rus. Is bang-bang control all you need? solving continuous control with Bernoulli policies. In *Advances in Neural Information Processing Systems*, volume 34, pages 27209–27221, 2021.
- Tim Seyde, Peter Werner, Wilko Schwarting, Igor Gilitschenski, Martin Riedmiller, Daniela Rus, and Markus Wulfmeier. Solving continuous control via Q-learning. In *International Conference on Learning Representations*, 2023.
- Nur Muhammad Mahi Shafiullah, Zichen Jeff Cui, Ariuntuya Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL <https://arxiv.org/abs/2206.11251>.
- Neta Shaul, Itai Gat, Marton Havasi, Daniel Severo, Anuroop Sriram, Peter Holderrieth, Brian Karrer, Yaron Lipman, and Ricky T. Q. Chen. Flow matching with general discrete paths: A kinetic-optimal perspective. In *International Conference on Learning Representations*, pages 16397–16429, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/29cd7f8331d13ede6dc6d6ef3dfacb70-Paper-Conference.pdf.
- Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems*, volume 37,

- pages 103131–103167, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/bad233b9849f019aead5e5cc60cef70f-Paper-Conference.pdf.
- Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. SmolVLA: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.
- Wenxuan Song, Jiayi Chen, Shuai Chen, Jingbo Wang, Pengxiang Ding, Han Zhao, Yikai Qin, Xinhui Zheng, Yan Wang, Donglin Wang, and Haoang Li. Fast-dVLA: Accelerating discrete diffusion VLA to real-time performance. In *European Conference on Computer Vision (ECCV)*, 2026. URL <https://arxiv.org/abs/2603.25661>.
- Lawrence Stewart, Francis Bach, Quentin Berthet, and Jean-Philippe Vert. Regression as classification: Influence of task formulation on neural network features. In *International Conference on Artificial Intelligence and Statistics*, pages 11563–11582, 2023. URL <https://proceedings.mlr.press/v206/stewart23a.html>.
- Andrew Szot, Bogdan Mazouze, Harsh Agrawal, Devon Hjelm, Zsolt Kira, and Alexander Toshev. Grounding multimodal large language models in actions. In *Advances in Neural Information Processing Systems*, 2024.
- Luhan Tang, Longxuan Yu, Shaorong Zhang, and Greg Ver Steeg. Is your diffusion sampler actually correct? A sampler-centric evaluation of discrete diffusion language models. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 119372–119401, 2026. URL <https://proceedings.mlr.press/v306/tang26ad.html>.
- Yunhao Tang and Shipra Agrawal. Discretizing continuous action space for on-policy optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5981–5988, 2020. doi: 10.1609/aaai.v34i04.6059.
- Arash Tavakoli, Fabio Pardo, and Petar Kormushev. Action branching architectures for deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 4131–4138, 2018.
- Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016a.
- Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, pages 1747–1756, 2016b.
- Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. In *Advances in Neural Information Processing Systems*, volume 38, pages 147282–147339, 2025a. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/d877ea0dd78bdb5e4830670618c1de09-Paper-Conference.pdf.
- Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi (Jim) Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *ArXiv*, abs/2305.16291, 2023. URL <https://api.semanticscholar.org/CorpusID:258887849>.
- Pengcheng Wang, Kaiwen Hong, Chensheng Peng, Katherine Driggs-Campbell, Masayoshi Tomizuka, Chenfeng Xu, and Chen Tang. DiscreteRTC: Discrete diffusion policies are natural asynchronous executors. *arXiv preprint arXiv:2604.25050*, 2026a. URL <https://arxiv.org/abs/2604.25050>.

- Ruiheng Wang, Shuanghao Bai, Haoran Zhang, Badong Chen, and Xiangyu Xu. BlockVLA: Accelerating autoregressive VLA via block diffusion finetuning. *arXiv preprint arXiv:2605.13382*, 2026b. URL <https://arxiv.org/abs/2605.13382>.
- 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. *arXiv preprint arXiv:2507.01016*, 2025b.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- Junjie Wen, Minjie Zhu, Jiaming Liu, Zhiyuan Liu, Yicun Yang, Linfeng Zhang, Shanghang Zhang, Yichen Zhu, et al. dVLA: Diffusion vision-language-action model with multimodal chain-of-thought. *arXiv preprint arXiv:2509.25681*, 2025a. URL <https://arxiv.org/abs/2509.25681>.
- Yuqing Wen, Hebei Li, Kefan Gu, Yucheng Zhao, Tiancai Wang, and Xiaoyan Sun. LLaDA-VLA: Vision language diffusion action models. *arXiv preprint arXiv:2509.06932*, 2025b. URL <https://arxiv.org/abs/2509.06932>.
- Shihan Wu, Xu Luo, Ji Zhang, Junlin Xie, Jingkuan Song, Heng Tao Shen, and Lianli Gao. Policy contrastive decoding for robotic foundation models. In *International Conference on Learning Representations*, 2026a.
- Xiao-Ming Wu, Kang Liao, Yihang Luo, Bin Fan, Jian-Jian Jiang, Runze Yang, Zhonghua Wu, Wei-Shi Zheng, and Chen Change Loy. VLANeXt family: A systematic study of VLA models from core recipes to emerging paradigms. *arXiv preprint arXiv:2602.18532*, 2026b.
- Yuhao Wu, Yitian Liu, Weijie Shen, Mishuo Han, Wenjie Xu, Haotian Liang, Zhongshan Liu, Yinan Mao, Lei Xu, Xinping Guan, Ru Ying, Ran Zheng, Wei Sui, Xiaokang Yang, Wenbo Ding, and Yao Mu. dVLA-RL: Reinforcement learning over denoising trajectories for discrete diffusion vision-language-action models. *arXiv preprint arXiv:2606.23623*, 2026c. URL <https://arxiv.org/abs/2606.23623>.
- Mingwang Xu, Jiahao Cui, Feipeng Cai, Hanlin Shang, Zhihao Zhu, Shan Luan, Yifang Xu, Neng Zhang, Yaoyi Li, Jia Cai, and Siyu Zhu. WAM-Diff: A masked diffusion VLA framework with MoE and online reinforcement learning for autonomous driving. *arXiv preprint arXiv:2512.11872*, 2025. URL <https://arxiv.org/abs/2512.11872>.
- Yifang Xu, Jiahao Cui, Zhihao Zhu, Hanlin Shang, Shan Luan, Mingwang Xu, Feipeng Cai, Neng Zhang, Yaoyi Li, Jia Cai, and Siyu Zhu. WAM-Flow: Parallel coarse-to-fine motion planning via discrete flow matching for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24918–24928, 2026. URL <https://arxiv.org/abs/2512.06112>.
- Shengzhuo Yang, Ronghao Yu, Chuanjie Lv, Linpeng Peng, Hang Yu, Jie Ren, Jiajun Lv, and Yong Liu. TS-Mask VLA: 2D temporal-spatial masking for vision-language-action model with effective bridging. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2026. URL <https://arxiv.org/abs/2607.09818>.
- Jiacheng Ye, Shansan Gong, Jiahui Gao, Junming Fan, Shuang Wu, Wei Bi, Haoli Bai, Lifeng Shang, and Lingpeng Kong. Dream-VL & Dream-VLA: Open vision-language and vision-language-action models with diffusion language model backbone. *arXiv preprint arXiv:2512.22615*, 2025a. URL <https://arxiv.org/abs/2512.22615>.

- Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025b. URL <https://arxiv.org/abs/2508.15487>.
- Jinhui Ye, Ning Gao, Senqiao Yang, Jinliang Zheng, Zixuan Wang, Yuxin Chen, Pengguang Chen, Yilun Chen, Shu Liu, and Jiaya Jia. Less is more: Reducing complexity in vision-language-action systems. In *European Conference on Computer Vision (ECCV)*, 2026a. URL <https://arxiv.org/abs/2604.11757>.
- Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026b. URL <https://arxiv.org/abs/2602.15922>.
- Jaehoon Yoo, Wonjung Kim, and Seunghoon Hong. ReDi: Rectified discrete flow. In *Advances in Neural Information Processing Systems*, volume 38, pages 81651–81683, 2025. URL <https://arxiv.org/abs/2507.15897>.
- Annan Yu, Danielle C. Maddix, Boran Han, Xiyuan Zhang, Abdul Fatir Ansari, Oleksandr Shchur, Christos Faloutsos, Andrew Gordon Wilson, Michael W. Mahoney, and Yuyang Wang. Understanding the implicit biases of design choices for time series foundation models. In *International Conference on Learning Representations*, 2026. URL <https://openreview.net/forum?id=5jkzTzV5Ao>.
- Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2021.
- Zhihao Zhan, Jiaying Zhou, Likui Zhang, Qinhan Lv, Hao Liu, Jusheng Zhang, Weizheng Li, Ziliang Chen, et al. E0: Enhancing generalization and fine-grained control in VLA models via tweedie discrete diffusion. *arXiv preprint arXiv:2511.21542*, 2025. URL <https://arxiv.org/abs/2511.21542>.
- Qinglun Zhang, Zhen Liu, Haoqiang Fan, Guanghui Liu, Bing Zeng, and Shuaicheng Liu. FlowPolicy: Enabling fast and robust 3D flow-based policy via consistency flow matching for robot manipulation. *arXiv preprint arXiv:2412.04987*, 2024. URL <https://arxiv.org/abs/2412.04987>.
- Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems*, 2023. URL <https://arxiv.org/abs/2304.13705>.
- Yixiu Zhao, Jiaxin Shi, Feng Chen, Shaul Druckmann, Lester Mackey, and Scott Linderman. Informed correctors for discrete diffusion models. In *Advances in Neural Information Processing Systems*, volume 38, pages 125510–125538, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/b67d9d07faf071532c420ed5b140756b-Paper-Conference.pdf.
- Hongyi Zhou, Weiran Liao, Xi Huang, Yucheng Tang, Fabian Otto, Xiaogang Jia, Xinkai Jiang, Simon Hilber, Ge Li, Qian Wang, Ömer Erdiñç Yağmurlu, Nils Blank, Moritz Reuss, and Rudolf Lioutikov. BEAST: Efficient tokenization of b-splines encoded action sequences for imitation learning. In *Advances in Neural Information Processing Systems*, 2025.

Lipeng Zhuang, Shiyu Fan, Florent P. Audonnet, Yingdong Ru, Edmond S. L. Ho, Gerardo Aragon Camarasa, and Paul Henderson. Masked generative policy for robotic control. In *International Conference on Learning Representations*, pages 118833–118864, 2026. URL <https://arxiv.org/abs/2512.09101>.

Contents

1	Introduction	2
1.1	Contributions	2
2	Preliminaries: Action Representations and Prediction Heads	3
3	Discrete Action Prediction Can Match Continuous Parametrizations	5
3.1	What Enables the Performance of Discrete Policies?	6
4	Factorization Error and Iterative Computation in Discrete Policies	8
5	Additional Structure Provides Limited Gains for Discrete Policies	10
6	Discussion and Future Work	11
A	Related Work	28
B	Environment Details	30
B.1	Evaluation Measures and Comparison Protocol	30
B.2	From-Scratch Training Experiments	30
B.3	VLA-Scale Experiments	31
B.4	VLA Training Data Mixture	31
C	Action Head and Objective Details	32
C.1	Continuous Heads	32
C.2	Scalar Action Quantization	32
C.3	Discrete Training Objectives	33
C.4	Discrete-Flow Source Distribution and Sampler	34
C.5	Classifier-Free Guidance	34
C.6	Omitted Details for Policy Architecture	35
C.6.1	Single-Task From-Scratch Architectures	35
C.6.2	VLA-Scale Architectures	36
D	Full Performance Results	37
D.1	RoboMimic Results	37
D.2	Backbone Variants	37
D.3	Delta, Relative, and Absolute Actions	38
E	Temporal Variation and Policy Responsiveness Experiments	40
E.1	Temporal Variation on Held-Out Chunks	40
E.2	Prediction Error Under Varying CFG	40
E.3	Distributional Robustness and the Cost of Contraction	40
E.3.1	Flows Retain Broader Proposals With Smaller Relative Changes	41
E.3.2	CF’s Stronger Contraction Comes at a Cost in Responsiveness	42
E.4	Disturbance and Expert-Reference Diagnostic Protocol	42
E.4.1	Measuring a Distribution’s Response to Disturbance	42

F	Factorization Error Experiments	43
E.1	Experimental Methodology	43
E.2	Factorization Error and L_2 Loss Diagnostic Results	44
E.3	Task Performance Under Varying NFE and CFG	44
E.3.1	Task Performance Under Varying NFE	44
E.3.2	Task Performance Under Varying CFG	45
E.4	Visualization of Mode Inconsistency in RoboCasa	46
G	Tokenization and Geometric Priors	48
G.1	Performance Under Different Tokenization	48
G.2	Geometry-Aware Probability Paths and Evaluations	48
G.2.1	From a Mixture Path to a Metric-Induced Path	49
G.2.2	Evaluated Metric-Path Variants	50
G.3	Bin Representation and Input-Sensitivity Study	50
G.3.1	Combined Output-Representation Visualization	51
H	Synthetic Fitting Protocol and Results	52
H.1	Which Fine Structure Does Categorical Prediction Preserve?	52

A Related Work

Action representations. Early generalist policies predicted actions as tokens. RT-1 discretizes each action coordinate into 256 uniform bins [Brohan et al., 2022], RT-2 and OpenVLA reuse such bins as tokens of a pretrained vision-language model [Brohan et al., 2023, Kim et al., 2024], and Gato trains one transformer from scratch on text and on actions in 1024 bins [Reed et al., 2022]. The field has since moved to continuous heads, motivated by the precision of continuous outputs and by the ability of diffusion models to represent multimodal action distributions. Diffusion Policy and ACT generate continuous chunks [Chi et al., 2023, Zhao et al., 2023], Octo and RDT-1B use a diffusion head or backbone [Octo Model Team et al., 2024, Liu et al., 2025], and most recent VLAs attach a diffusion or flow-matching module to a pretrained VLM [Li et al., 2024a, Black et al., 2025, Physical Intelligence et al., 2025b, NVIDIA et al., 2025, Shukor et al., 2025] or regress actions directly [Kim et al., 2025]. Methods that keep discrete actions usually add machinery to address a perceived weakness of plain bins. Behavior Transformers add a continuous offset to a discrete cluster to recover precision [Shafiullah et al., 2022, Cui et al., 2023, Lee et al., 2024]. FAST, vector and scalar quantization, and B-spline tokenizers compress chunks into shorter token sequences [Pertsch et al., 2025, Lee et al., 2024, Belkhale and Sadigh, 2024, Wang et al., 2025b, Mete et al., 2024, Zhou et al., 2025]. HybridVLA and knowledge insulation combine a discrete token loss with a continuous action objective in one model [Liu et al., 2026a, Driess et al., 2025]. We ask whether this machinery is needed. Our discrete heads predict plain per-coordinate uniform bins, and Section 5 tests several of these tokenizers.

Comparisons of discrete and continuous heads. The move to continuous heads rests partly on studies that swap the action head within one model and find continuous heads stronger [Octo Model Team et al., 2024, Li et al., 2024b, Kim et al., 2025, Hou et al., 2025, Wu et al., 2026b, Ye et al., 2026a]. Octo, RoboVLMS, and OpenVLA-OFT attribute the gap to bin precision. But these studies change the representation together with other choices. The discrete head usually decodes tokens autoregressively, and FAST shows that autoregressive decoding of per-step bins fails at 20 Hz and 50 Hz [Pertsch et al., 2025]. The discrete head is often a linear projection onto the language-model vocabulary while the continuous head is a new MLP [Li et al., 2024b, Kim et al., 2025], and the discrete baselines in Diffusion Policy and ACT predict one action at a time [Chi et al., 2023, Zhao et al., 2023]. When all tokens are decoded in one pass, the gap shrinks or reverses. RT-1’s one-pass bins outperform a unimodal Gaussian head [Brohan et al., 2022], BEAST’s parallel tokens outperform L1 regression on the same B-spline representation [Zhou et al., 2025], BAKU’s BeT and VQ-BeT heads match an MLP head and match or beat a diffusion head [Haldar et al., 2024], and Discrete Diffusion VLA reports 94.2% LIBERO success for parallel bins, 81.2% for autoregressive bins, and 94.9% for a continuous diffusion head [Liang et al., 2026b]. The evidence is still not one-sided: parallel bins trail L1 regression in OpenVLA-OFT (90.2% vs. 95.3%) [Kim et al., 2025], and the ranking of heads in Dream-VLA changes with the backbone [Ye et al., 2025a]. Separating the representation from the decoding requires varying them independently. Pan et al. [2026] do this for continuous heads, comparing regression and flow under matched architectures. We extend the comparison to discrete heads, crossing continuous and discrete representations with one-pass and iterative generation under one backbone and training recipe.

Discrete diffusion and flow policies. A second line of work keeps discrete actions and replaces autoregressive decoding with iterative parallel decoding. Discrete Diffusion VLA decodes 256-bin action tokens inside the VLM by confidence-ordered unmasking [Liang et al., 2026b]. Some VLAs build on diffusion language models: LLaDA-VLA and MMaDA-VLA start from one [Wen et al., 2025b, Liu et al., 2026b], and Dream-VLA pretrains one on discrete action tokens before fine-tuning a flow head [Ye et al., 2025a]. A shared token space also lets one model generate actions together with text and images. dVLA,

UD-VLA, and MMaDA-VLA denoise these tokens jointly [Wen et al., 2025a, Chen et al., 2026a, Liu et al., 2026b], and MM-ACT co-trains the three generation tasks [Liang et al., 2026a]. Later work targets speed and deployment through block diffusion and caching [Lee et al., 2026, Wang et al., 2026b, Song et al., 2026], structured masks [Yang et al., 2026], inpainting for asynchronous execution [Wang et al., 2026a], and RL fine-tuning [Wu et al., 2026c]. The approach also extends to policies trained without a VLM [He et al., 2024, Zhuang et al., 2026] and to driving [Li et al., 2026c, Xu et al., 2025, Dang et al., 2026]. Closest to our DF head, DFM-VLA and WAM-Flow use discrete flow matching with a uniform source and a metric-induced path [Chen et al., 2026b, Xu et al., 2026], and EO trains a π_0 -style expert with cross-entropy on Gaussian-noised one-hot bins [Zhan et al., 2025]. This work focuses on building faster and more capable models, and leaves open why iterative decoding helps. Several of these papers find that it beats one-pass decoding in the same model [Liang et al., 2026b, Lee et al., 2026, Liang et al., 2026a], but fewer than half train a continuous head on the same backbone, none of the manipulation policies applies classifier-free guidance to action tokens, and DFM-VLA and WAM-Flow do not compare the metric path to a uniform-source mixture path with all else fixed. Section 4 attributes the gain of iterative decoding to reduced factorization error and adds guidance, and Section 5 tests metric-induced paths.

Discrete diffusion and flow models. Our DF head builds on discrete diffusion [Austin et al., 2021, Hoogetboom et al., 2021, Campbell et al., 2022], in particular its masked variant [Lou et al., 2024, Sahoo et al., 2024, Shi et al., 2024, Ou et al., 2025, Chang et al., 2022, 2023], which now scales to large language models [Nie et al., 2025, Ye et al., 2025b], and on discrete flow matching [Gat et al., 2024, Campbell et al., 2024]. Shaul et al. [2025] extend discrete flow matching to arbitrary paths and propose the metric-induced path tested in Section 5. Uniform sources and remasking samplers, which let committed tokens change, are related alternatives [Schiff et al., 2025, Sahoo et al., 2025, Deng et al., 2026, Wang et al., 2025a, Zhao et al., 2025]. Most relevant to us, few-step sampling in these models suffers the same factorization error as one-pass action heads, because each step predicts positions independently. ReDi and CoDD reduce this error [Yoo et al., 2025, Li et al., 2026a], and Tang et al. [2026] show that it persists even with an exact denoiser. Section 4 studies the same error for action chunks.

Discretized control and classification losses. Section 3 shows that discrete heads can match continuous ones despite quantization. Work outside imitation learning suggests why precision is rarely the bottleneck. In reinforcement learning, per-dimension action bins work well on continuous-control benchmarks [Metz et al., 2017, Tavakoli et al., 2018, Tang and Agrawal, 2020, Seyde et al., 2021, 2023, Chebotar et al., 2023], Dactyl uses only 11 bins per coordinate [OpenAI et al., 2020], and value and reward heads are often trained with cross-entropy over bins [Bellemare et al., 2017, Hafner et al., 2025, Hansen et al., 2024]. HL-Gauss, a cross-entropy loss with Gaussian-smoothed bin targets [Imani and White, 2018], improves the performance and scaling of deep RL [Farebrother et al., 2024]. Analyses attribute such gains to training rather than to modelling the target distribution: easier optimization in supervised regression [Imani et al., 2026], robustness to noisy targets and better representations in RL [Farebrother et al., 2024], and different learned features in two-layer networks [Stewart et al., 2023]. MSE also has a known failure mode. It averages distinct modes [Shafiullah et al., 2022] and smooths over discontinuities [Florence et al., 2021], and in forecasting, models trained with MSE or quantile losses regress toward the mean or median when the context is uncertain, while cross-entropy over bins can settle on a mode [Yu et al., 2026]. For behavior cloning, the regret with quantized actions and a log-loss depends on the quantization error only polynomially in the horizon, given stable dynamics and a probabilistically smooth policy [Cao and Xie, 2026]. Categorical outputs are also standard for other continuous signals such as pixels and audio [van den Oord et al., 2016b,a]. For VLAs the evidence is

mixed: 512 uniform bins are worse than MSE regression on CALVIN and better on Meta-World [Szot et al., 2024]. Our distance-based label smoothing follows soft ordinal targets [Gao et al., 2017, Díaz and Marathe, 2019, Imani and White, 2018, Chung et al., 2026], and our logit soft-capping follows Gemma 2 [Gemma Team et al., 2024].

Classifier-free guidance. Section 4 uses classifier-free guidance [Ho and Salimans, 2022] to sharpen discrete action distributions. For categorical outputs, guidance reweights each token’s distribution directly. Muse applies it to token logits [Chang et al., 2023], later work formalizes it for discrete diffusion and flow [Nisonoff et al., 2025, Schiff et al., 2025, Rojas et al., 2026], and on language-model logits it lowers entropy while also reordering the top tokens [Sanchez et al., 2024]. We apply the Muse rule to each coordinate’s clean-token distribution. In continuous diffusion the effect is less direct. Guided sampling does not sample the tilted distribution [Chidambaram et al., 2024, Bradley and Nakkiran, 2025], and in simple mixtures large weights push samples toward the edge of the support [Chidambaram et al., 2024]. Results on continuous robot policies are mixed. Guidance helps CogACT [Li et al., 2024a] and gives a small gain for language-conditioned flow VLAs [Park et al., 2026], but it is slightly worse than plain goal conditioning in BESO [Reuss et al., 2023], hurts when applied to the observation [Pearce et al., 2023], and makes the RDT-1B arm unstable [Liu et al., 2025]. $\pi_{0.6}^*$ notes that large weights push actions to the corners of the support and cause aggressive motions [Physical Intelligence et al., 2025a]. On discrete action tokens, the closest prior use is policy contrastive decoding over OpenVLA’s tokens [Wu et al., 2026a]. In our experiments, guidance raises DR success on RoboCasa from 27.1% to 50.2% (Section 4).

B Environment Details

This section gives the evaluation measures, training recipes, and comparison protocol for the experiments in Sections 3 to 5.

B.1 Evaluation Measures and Comparison Protocol

Settings. Our evidence spans pretrained vision–language–action policies, evaluated on RoboCasa and LIBERO and diagnosed offline on DROID, and policies trained from scratch on RoboMimic and Push-T. Controlled function fitting uses separate models with a common trunk architecture to predict deterministic targets from a scalar input. These settings let us examine precision in robot policies and study fitting behavior where the target and codec error are known.

Evaluating action quality. We distinguish three measurements. *Prediction error* compares predicted and demonstrated actions, reporting training fit and held-out error separately. Robot translation errors use millimeters; Push-T uses pixels. *Temporal variation*, assessed through jerk, describes variation in the commanded trajectory. *Task performance* measures closed-loop success on RoboCasa, LIBERO, and RoboMimic in Figure 1.

Robot diagnostics use 5 training seeds with per-head checkpoints and inference budgets. Synthetic comparisons use 3 initialization seeds. Formal error definitions, benchmark recipes, and the chronology of the synthetic studies are in Appendices B and H.

B.2 From-Scratch Training Experiments

Single-task RoboMimic policies are state-based and trained from scratch, without a pretrained vision–language backbone. Their evaluation protocol and records are in Appendix D.1.

B.3 VLA-Scale Experiments

All VLA-scale policies are trained on one shared data mixture (Appendix B.4). They share one PaliGemma backbone, action horizon 24, quantile normalization statistics, and seed 7. Actions are relative end-effector commands for a Franka arm: target poses with respect to the end-effector pose at prediction time, with axis-angle rotation. For each head, we evaluate the last five saved checkpoints, at 110k, 120k, 130k, 140k, and 150k training steps, and report the best of the five. Each evaluation uses five seeds; we report the mean success rate over the seeds, and $\pm$ denotes the standard deviation over them. Evaluation replans every 24 steps, and every head runs the same evaluation episodes. The backbone, optimizer, and batch size are given in Appendix C.6.2.

RoboCasa. RoboCasa [Nasiriany et al., 2026] is a simulated benchmark of household manipulation tasks in kitchen scenes. We run 10 trials per task on three suites: atomic-seen (180 episodes), composite-seen (160), and composite-unseen (160).

LIBERO. LIBERO [Liu et al., 2023] is a simulated benchmark of language-conditioned tabletop manipulation with a Franka arm. Each suite has 10 tasks with 50 human demonstrations per task. We evaluate closed-loop success on two suites. In LIBERO-Object, the tasks share one layout and differ in which object the robot picks and places in a basket. LIBERO-10 (also called LIBERO-Long) has long-horizon tasks that each chain two subtasks across varied objects, layouts, and scenes. The training mixture contains LIBERO-plus demonstrations (Appendix B.4), and held-out LIBERO-plus episodes are a validation source for the offline diagnostics.

DROID. DROID [Khazatsky et al., 2024] is a real-world Franka teleoperation dataset of 76k trajectories collected across 564 scenes and 86 tasks. It is part of the VLA training mixture. DROID has no simulator, so we do not measure closed-loop success on it. Held-out DROID chunks instead serve as a validation source for the offline diagnostics: the jerk ratio and EEF error in Fig. 2, the t-SNE in Fig. 4(d), and the CFG error sweep in Appendix E.2.

B.4 VLA Training Data Mixture

The VLA-scale policies train on one mixture of eight sources: six RoboCasa365 datasets [Nasiriany et al., 2026], LIBERO-plus [Fei et al., 2025], and DROID [Khazatsky et al., 2024] (Tab. 2). All eight share one action space, a 7-dimensional relative end-effector command for a Franka arm: 3D position and axis-angle rotation with respect to the end-effector pose at prediction time, and gripper. One set of quantile normalization statistics covers the whole mixture.

We filter frames before sampling. For RoboCasa, we drop the frames in which the robot drives its mobile base and keep the arm-control frames. For DROID, we keep the 78,864 successful episodes out of 95,658 and remove idle segments. LIBERO-plus is used in full. Each source is sampled with weight proportional to the square root of its frame count. The RoboCasa MimicGen source has 116M frames of synthetic demonstrations. To keep it from dominating the mixture, its frame count is capped at DROID’s when we compute the weight. We hold out 2% of the episodes of every source, chosen with a fixed seed. These held-out episodes form the eight validation sources for the offline diagnostics.

Table 2: Training mixture for the VLA-scale policies. Frames are counted after filtering. Weights are proportional to the square root of the frame count, with the MimicGen count capped at DROID’s.

Source	Frames (M)	Weight
RoboCasa pretrain atomic	1.38	0.052
RoboCasa pretrain composite	21.89	0.206
RoboCasa pretrain MimicGen	116.25	0.202
RoboCasa target atomic	2.14	0.065
RoboCasa target composite-seen	5.17	0.100
RoboCasa target composite-unseen	5.99	0.108
LIBERO-plus	2.24	0.066
DROID	21.06	0.202

C Action Head and Objective Details

This appendix gives the training objectives, scalar codec, and sampling details for the policies introduced in Section 2. Table 3 summarizes the four head families. Per-experiment settings are given in Appendices B and H.

C.1 Continuous Heads

Continuous regression. For $L = HM$, the continuous-regression policy is trained with

$$\mathcal{L}_{\text{CR}} = \mathbb{E}_{(o, \mathbf{a}_{1:H}) \sim \mathcal{D}} \left[\frac{1}{L} \|\pi_{\theta}^{\text{CR}}(o) - \mathbf{a}_{1:H}\|_F^2 \right], \quad \hat{\mathbf{a}}_{1:H} = \pi_{\theta}^{\text{CR}}(o). \quad (\text{C.1})$$

The unrestricted population minimizer is $\mathbb{E}[\mathbf{a}_{1:H} \mid o]$. When several actions are plausible under the same conditioning, this mean need not itself correspond to a demonstrated action. Auxiliary random inputs are suppressed in this notation; the expectation also averages over them when present.

Continuous flow. Let $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ be independent Gaussian noise with the same shape as $\mathbf{a}_{1:H}$. For $t \sim \mathcal{U}[0, 1]$ and a continuously differentiable, nondecreasing schedule satisfying $\alpha(0) = 0$ and $\alpha(1) = 1$, define the stochastic interpolant [Albergo and Vanden-Eijnden, 2022] $I_t = (1 - \alpha(t))\epsilon + \alpha(t)\mathbf{a}_{1:H}$ and its derivative $U_t(\mathbf{a}_{1:H}, \epsilon) = \dot{\alpha}(t)(\mathbf{a}_{1:H} - \epsilon)$. The velocity field is trained by flow matching [Lipman et al., 2022, Liu et al., 2022]:

$$\mathcal{L}_{\text{CF}} = \mathbb{E}_{\substack{(o, \mathbf{a}_{1:H}) \sim \mathcal{D} \\ t \sim \mathcal{U}[0, 1], \epsilon \sim \mathcal{N}(0, \mathbf{I})}} \left[\|v_{\theta}(I_t, t, o) - U_t(\mathbf{a}_{1:H}, \epsilon)\|_F^2 \right]. \quad (\text{C.2})$$

At inference, use a grid $0 = t_0 < \dots < t_N = 1$ and Euler updates

$$\mathbf{x}_{t_{n+1}} = \mathbf{x}_{t_n} + (t_{n+1} - t_n)v_{\theta}(\mathbf{x}_{t_n}, t_n, o), \quad \mathbf{x}_{t_0} = \epsilon \sim \mathcal{N}(0, \mathbf{I}), \quad (\text{C.3})$$

for $n = 0, \dots, N - 1$. The resulting policy is $\pi_{\theta, N}^{\text{CF}}(o; \epsilon) := \mathbf{x}_{t_N}$.

C.2 Scalar Action Quantization

Actions are normalized and clipped according to the recipe in Appendix B. For coordinate i , let $[\ell_i, u_i]$ be its quantization range and $\Delta_i = (u_i - \ell_i)/B$ its bin width. The ranges may differ across coordinates. With token position $j = (h - 1)M + i$, the uniform quantizer and bin-center decoder are

$$Q(\mathbf{a}_{1:H})_j := \text{clip}_{[1, B]} \left(1 + \left\lfloor \frac{a_{h,i} - \ell_i}{\Delta_i} \right\rfloor \right), \quad (\text{C.4})$$

$$[D_Q(\mathbf{z})]_{h,i} := \ell_i + (z_j - \tfrac{1}{2})\Delta_i.$$

Table 3: Four heads: representation crossed with inference computation. Losses and decoders are described at the head-family level; per-experiment details are in Appendices B and H.

Head	Output	Training loss	Generation
CR	real-valued chunk	squared error on the chunk	one pass
CF	real-valued chunk	velocity or clean-target MSE	N integration steps from noise
DR	one token per coordinate	token cross-entropy (hard or distance-smoothed)	one pass, argmax on every token
DF	one token per coordinate	clean-token cross-entropy on corrupted chunks	N updates, conditioned on the current chunk

Clipping keeps the index in $[B]$, including at the upper endpoint. The quantizer flattens the chunk in a fixed order, and the decoder restores its $H \times M$ shape. Both maps use normalized action coordinates; normalization is inverted before execution. Thus B is both the number of bins per coordinate and the action-token vocabulary size. Unless otherwise stated, $B = 512$. Alternative tokenizations, including quantile-based bins and more compressive encoders, are described in Appendix G.1.

C.3 Discrete Training Objectives

For $\mathbf{z} = Q(\mathbf{a}_{1:H})$, discrete regression is trained by tokenwise cross-entropy:

$$\mathcal{L}_{\text{DR}} = \mathbb{E}_{(o, \mathbf{a}_{1:H}) \sim \mathcal{D}} \left[-\frac{1}{L} \sum_{j=1}^L \log p_{\theta,j}^{\text{DR}}(z_j | o) \right]. \quad (\text{C.5})$$

The head outputs $L \times B$ logits, with a softmax over bins at each position [Brohan et al., 2022]. Conditional on the model inputs, its factorized distribution is $p_{\theta}^{\text{DR}}(\mathbf{z} | o) = \prod_{j=1}^L p_{\theta,j}^{\text{DR}}(z_j | o)$. Consequently, maximizing this model’s joint probability reduces to selecting each token’s mode [Devlin et al., 2019]: $\hat{z}_j = \arg \max_{k \in [B]} p_{\theta,j}^{\text{DR}}(k | o)$. The policy returns $\pi_{\theta}^{\text{DR}}(o) := D_Q(\hat{\mathbf{z}})$.

Discrete flow also predicts the clean token at each position, but conditions on the corrupted chunk and flow time. We sample $t \sim \mathcal{U}[0, 1]$ and $\mathbf{z}_t \sim q_t(\cdot | \mathbf{z})$ using Equation (2.1), and train with clean-token cross-entropy [Gat et al., 2024]:

$$\mathcal{L}_{\text{DF}} = \mathbb{E}_{\substack{(o, \mathbf{a}_{1:H}) \sim \mathcal{D} \\ t \sim \mathcal{U}[0, 1] \\ \mathbf{z}_t \sim q_t(\cdot | \mathbf{z})}} \left[-\frac{1}{L} \sum_{j=1}^L \log p_{\theta,j}^{\text{DF}}(z_j | \mathbf{z}_t, t, o) \right]. \quad (\text{C.6})$$

DR variants with distance-smoothed targets are specified below.

Per-benchmark DR loss. Four action heads cross representation with decoding: continuous regression (CR), continuous flow matching [Lipman et al., 2022] (CF), discrete regression over flat bins with per-token argmax (DR), and discrete flow over flat bins (DF). An autoregressive head (AR) appears in the FAST tokenizer validity and closed-loop comparisons. The DR loss differs between benchmarks. On RoboCasa it is a distance-kernel soft cross-entropy with kernel temperature $\tau = 0.0025$ in normalized action units, so the neighbor bin’s unnormalized weight is $\exp(-0.0078/0.0025) \approx 0.045$ of the target bin’s. On Push-T it is hard-label cross-entropy. Statements about “cross-entropy without a metric” therefore apply per run.

C.4 Discrete-Flow Source Distribution and Sampler

The source factorizes as $q_0(\mathbf{z}_0) = \prod_{j=1}^L q_{0,j}(z_{0,j})$. A uniform source has $q_{0,j}(k) = 1/B$ for $k \in [B]$ [Gat et al., 2024]; a masked source places all mass on an auxiliary mask token [Sahoo et al., 2024]. The mask is an input symbol and does not decode to an action bin. In either case, the clean-token predictor outputs distributions over $[B]$.

The schedule is continuously differentiable and nondecreasing, with $\alpha(0) = 0$, $\alpha(1) = 1$, and $\alpha(t) < 1$ for $t < 1$. For the grid $0 = t_0 < \dots < t_N = 1$, the mixture-path update in Equation (2.3) uses $\eta_n = (\alpha(t_{n+1}) - \alpha(t_n)) / (1 - \alpha(t_n))$. Proposals and replacement decisions are independent across positions conditional on the current chunk. This rule gives an Euler approximation, in the interpolation coordinate α , to the associated continuous-time Markov chain. Because $\eta_{N-1} = 1$, the final update replaces every position with a clean-token proposal, so the output contains no mask tokens.

The policy is $\pi_{\theta,N}^{\text{DF}}(o; \xi) := D_Q(\mathbf{z}_{t_N})$, where ξ collects randomness in initialization, proposals, and replacement decisions. We suppress N and ξ when clear from context. For a uniform source, tokens updated at earlier steps may be revised later. Subsequent predictions condition on the updated chunk, allowing the generation process to capture dependencies across positions despite the conditional independence of updates within each step.

C.5 Classifier-Free Guidance

CFG [Ho and Salimans, 2022] combines a prediction conditioned on the observation o with one made from a null condition $\emptyset$.

Condition and null condition. Let c be the encoding of the observation o . For the VLA policies, c is the set of 64 condition tokens that the resampler produces from the backbone states and the proprioception token (Appendix C.6.2), so it carries the images, the instruction, and the proprioceptive state. The null condition $\emptyset$ is a learned set of 64 tokens with the same shape. During training, each example’s condition tokens are replaced by $\emptyset$ with probability 0.1, so the same head learns the conditional prediction and an unconditional prediction that sees no observation. All other head inputs (the current chunk, the flow time, and the auxiliary random inputs) are unchanged. We write $\emptyset$ in place of o for predictions made with the null condition, as in $p_{\theta,j}(k | \emptyset)$, $v_{\theta}(\mathbf{x}_t, t, \emptyset)$, and $\pi_{\theta}^{\text{CR}}(\emptyset)$.

Discrete heads. Suppressing other model inputs, discrete heads use

$$p_{\theta,j}^{\text{CFG}}(k | o) \propto p_{\theta,j}(k | o)^{1+\lambda} p_{\theta,j}(k | \emptyset)^{-\lambda}, \quad k \in [B], \quad \lambda \geq 0, \quad (\text{C.7})$$

with normalization over bins. Equivalently, if $s_{\theta,j}(o) \in \mathbb{R}^B$ and $s_{\theta,j}(\emptyset) \in \mathbb{R}^B$ are the conditional and unconditional logits at position j , apply softmax to $(1 + \lambda)s_{\theta,j}(o) - \lambda s_{\theta,j}(\emptyset)$. The rule multiplies $p_{\theta,j}(k | o)$ by $(p_{\theta,j}(k | o) / p_{\theta,j}(k | \emptyset))^{\lambda}$, so it moves mass toward bins that the observation makes more likely than the null condition does. Guidance is usually paired with an iterative sampler. Because DR is trained with the same condition dropout, its single forward pass also gives unconditional token distributions, so we apply the rule directly to DR’s one-pass logits. Both DR passes receive the same auxiliary random input tokens, and DR takes the argmax of each guided distribution. DF applies the rule at every refinement step: at step n , both passes read the same current chunk $\mathbf{z}_{t_n}$ and time t_n , and DF samples clean-token proposals from the guided distributions before the update in Equation (2.3).

Continuous flow. At each Euler step, CF evaluates the head at the same state $\mathbf{x}_{t_n}$ and time t_n , once with o and once with $\emptyset$. The guided velocity

$$v_{\theta}^{\text{CFG}}(\mathbf{x}_{t_n}, t_n, o) = (1 + \lambda)v_{\theta}(\mathbf{x}_{t_n}, t_n, o) - \lambda v_{\theta}(\mathbf{x}_{t_n}, t_n, \emptyset) \quad (\text{C.8})$$

replaces $v_\theta(\mathbf{x}_{t_n}, t_n, o)$ in the Euler update of Equation (C.3).

Why CR is not guided. Guidance reweights a distribution. For DR and DF it tilts each token’s distribution, and for CF it changes the velocity field that carries noise to actions. CR has no distribution to reweight: it outputs one chunk, its estimate of $\mathbb{E}[\mathbf{a}_{1:H} \mid o]$, and its unconditional output estimates the dataset mean $\mathbb{E}[\mathbf{a}_{1:H}]$. The same combination, $(1 + \lambda)\pi_\theta^{\text{CR}}(o) - \lambda\pi_\theta^{\text{CR}}(\emptyset)$, only moves the predicted chunk along $\mathbb{E}[\mathbf{a}_{1:H} \mid o] - \mathbb{E}[\mathbf{a}_{1:H}]$, further from the dataset mean. When several behaviors are plausible, the conditional mean already averages them, and moving it does not select one of them. We therefore evaluate CR without guidance.

Inference cost. The backbone and resampler encode o once per inference call. Because $\emptyset$ is a fixed learned tensor, the unconditional prediction needs no backbone pass, only a second pass of the action head. We report NFE as the number of steps N for DF and CF. Guidance doubles the action-head passes, from one to two for DR and from N to $2N$ for DF and CF, and leaves the backbone cost unchanged. Under this convention, $\lambda = 0$ gives the unguided conditional prediction.

C.6 Omitted Details for Policy Architecture

The heads use the same backbone design. CR and DR receive auxiliary random inputs independent of the target action chunk, so random inputs are present for both regression and flow heads. These auxiliary inputs are suppressed in the preceding notation. Within each setting, all heads see the same observations and training data. This section gives the networks that map these observations to actions; the losses and samplers are given in [Appendices C.1](#) and [C.3](#).

C.6.1 Single-Task From-Scratch Architectures

RoboMimic policies observe the object state and the end-effector position, orientation, and gripper state of each arm, over the last two time steps. An MLP encodes the flattened observation. Each policy predicts $H = 10$ actions and executes 8 before replanning. An action is a 10-dimensional relative end-effector command: 3D position, 6D rotation, and gripper (20 dimensions for the two-arm Transport task). Actions are normalized per dimension to $[-1, 1]$. Discrete heads use $B = 512$ bins per coordinate. [Tab. 4](#) lists the two network families.

Table 4: Networks for the single-task RoboMimic policies. Both families are DiT-style transformers conditioned through adaptive layer norm (adaLN) on the observation embedding and the flow time. Parameter counts include the observation encoder and are given for single-arm tasks.

	CR, CF	DR, DF
Blocks	8 encoder + 8 decoder	12
Width / heads / MLP width	256 / 4 / 1024	256 / 4 / 1024
Action tokens	one per time step (10)	one per time step and coordinate (100)
Action input	MLP on the 10-d action	learned embedding of B bins
Action output	linear to 10 values	logits over B bins
Observation MLP width	256	512
Dropout	0	0.1
Parameters	16.4M	15.8M

CF integrates its velocity field with 9 Euler steps. DF runs 100 update steps with the mixture path $\kappa_t = t^3$, followed by one argmax prediction. All models are trained for 300k steps with AdamW (learning

rate 10^{-4} , weight decay 10^{-5} , gradient clipping at 10) and an exponential moving average of the weights with decay 0.995, which is used for evaluation. Continuous heads use batch size 1024 and cosine decay; discrete heads use batch size 512, a 5% linear warmup, and cosine decay. Each run uses one L40S GPU. The synthetic-fitting trunk is described in [Appendix H](#).

C.6.2 VLA-Scale Architectures

All four heads share a PaliGemma-3B backbone [[Beyer et al., 2024](#)]: a SigLIP So400m/14 image encoder and a Gemma-2B language model. We initialize it from the π_0 vision-language weights [[Black et al., 2025](#)] and fine-tune it fully. The backbone reads up to three 224×224 camera images (256 tokens each) and up to 48 text tokens with bidirectional attention. A resampler with 64 learned queries (2 blocks, width 1536) compresses the final-layer backbone states, together with one proprioception token, into 64 condition tokens.

The action head is a separate transformer with 8 blocks, width 1536, 12 attention heads, and MLP width 6144. Each block applies self-attention over the action sequence, cross-attention to the 64 condition tokens, and an MLP. The head does not share attention layers with the backbone. It processes the $H = 24$ action chunk as a sequence with one element per action coordinate ($M = 7$: 3D position change, axis-angle rotation change, and gripper). Continuous heads represent each element as a real value; discrete heads represent it as a token over B bins. Learned position and coordinate embeddings and Fourier features of the flow time are added to every element. [Tab. 5](#) lists what all heads share, what each representation shares, what differs per head, and the training settings.

Bins are uniform over the normalized action range. DF uses a uniform source, and the reported DF policy uses the schedule $\alpha(t) = t^3$. The condition is dropped with probability 0.1 and replaced by a learned null condition, which enables CFG at inference ([Appendix C.5](#)). Training settings are listed in [Tab. 5](#), and the evaluation protocol, including checkpoint selection and seeds, is given in [Appendix B.3](#).

Table 5: Action heads and training settings for the VLA-scale policies. Parameter counts cover the action head and resampler; the backbone adds 2.92B parameters.

	CR	CF	DR	DF
<i>Shared by all four heads</i>				
Backbone	PaliGemma-3B from the π_0 weights, fully fine-tuned			
Condition tokens	64, from a 2-block resampler (width 1536)			
Blocks / width / heads / MLP	8 / 1536 / 12 / 6144			
Action chunk	$H = 24$ steps, $M = 7$ coordinates, one sequence element per coordinate			
<i>Shared within each representation</i>				
Action representation	real-valued vector $\mathbf{a}_{1:H} \in \mathbb{R}^{H \times M}$		token sequence $\mathbf{z} = Q(\mathbf{a}_{1:H}) \in [B]^{HM}$, $B = 512$	
Action input	real-valued vector of the same shape		token sequence of the same length	
Action output	one real value per element		logits over B bins per element	
<i>Per head</i>				
Input content	random input	interpolant I_t	random input	corrupted tokens $\mathbf{z}_t$, uniform source
Output target	clean chunk	velocity	clean tokens	clean tokens
Training loss	squared error	flow matching	distance- smoothed cross-entropy	clean-token cross-entropy
Generation	one pass	Euler integration	one pass, argmax	N updates, $\alpha(t) = t^3$
CFG	no	yes	yes	yes
Parameters	367M	367M	369M	371M
<i>Training, shared by all four heads</i>				
Training steps	150k			
Global batch size	512 (64 per GPU on 8 B200 GPUs)			
Optimizer	AdamW, learning rate 7.5×10^{-5} , $\beta = (0.9, 0.95)$, weight decay 10^{-4}			
Gradient clipping	1			
Learning-rate schedule	linear warmup, then cosine decay to $0.1 \times$ the peak			
Precision	bf16			

D Full Performance Results

D.1 RoboMimic Results

RoboMimic policies are state-based, trained from scratch on each task, and use relative actions. For each task and head family, Fig. 1 reports the best success rate over tokenizer (flat or quantile bins), flow variant, and guidance weight. Each evaluation uses 32 episodes.

Fig. 6 reports how closely each run fits its training data. For each run, we show the end-effector (EEF) position error on training samples at its last 10 evaluations.

D.2 Backbone Variants

We also train each head with two other backbone settings: a fine-tuned backbone initialized from the $\pi_{0.5}$ weights, and a frozen backbone initialized from the π_0 weights. Only the action head and resampler are trained in the frozen setting. All other settings follow Appendix C.6.2. Tab. 6 reports, for each head, the operating point with the highest pooled success over checkpoints, guidance weights $\lambda \in \{0, 2\}$, and NFE $\in \{5, 10, 20\}$ for the flow heads.

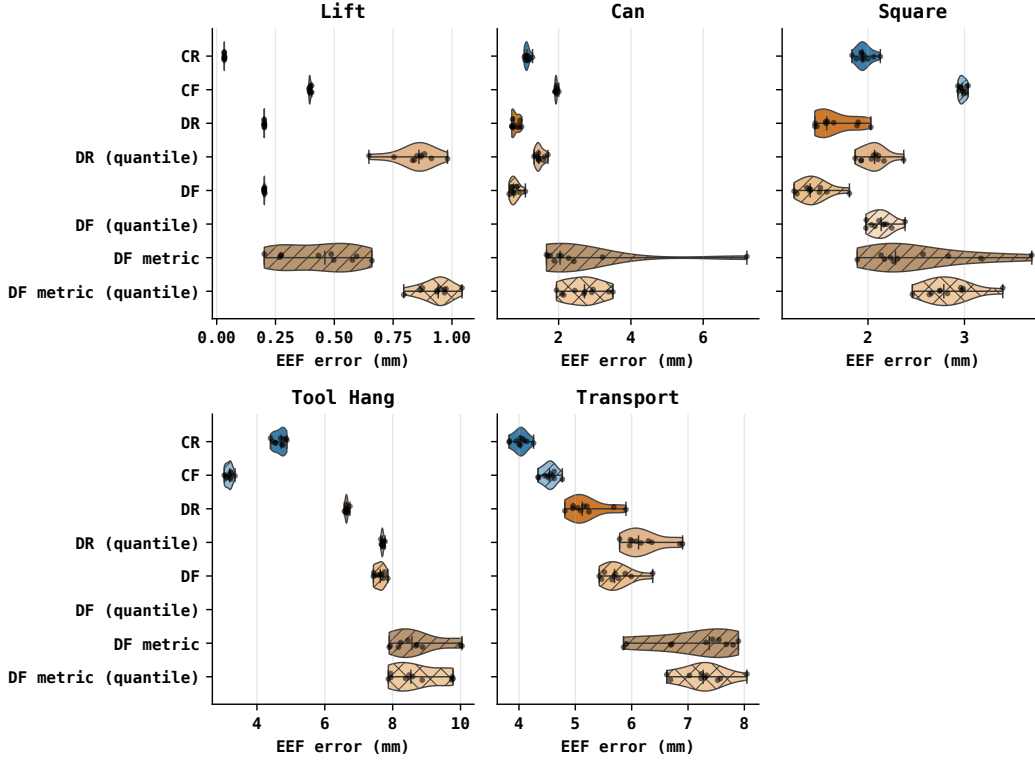


Figure 6: **EEF position error on RoboMimic training samples.** Each violin shows one run’s RMS EEF position error (mm) over its last 10 evaluations; dots are individual evaluations and the vertical line marks the median. The spread reflects late-training variation of the metric. DF denotes the mixture path and DF metric the metric-induced path (Appendix G.2); “quantile” marks equal-frequency bins, otherwise bins are uniform.

D.3 Delta, Relative, and Absolute Actions

Both the VLA and the single-task RoboMimic experiments in the main paper use relative actions. This section defines relative actions together with two alternatives, delta and absolute actions, and compares all three for the four VLA heads on RoboCasa (Fig. 7 and Tab. 7).

Definitions. Let the chunk $\mathbf{a}_{1:H}$ be predicted at environment step t , and let command h be sent at step $t+h-1$ with commanded target pose $(\mathbf{p}_h^*, R_h^*)$, where $\mathbf{p}_h^* \in \mathbb{R}^3$ and $R_h^* \in \text{SO}(3)$ are expressed in the robot base frame. Let $(\mathbf{p}_k, R_k)$ be the end-effector pose reached at step k . Each parameterization encodes the target with respect to a reference pose $(\bar{\mathbf{p}}_h, \bar{R}_h)$:

$$\mathbf{a}_h^{\text{tr}} = \mathbf{p}_h^* - \bar{\mathbf{p}}_h, \quad \mathbf{a}_h^{\text{rot}} = \text{Log}(R_h^* \bar{R}_h^{\top}), \quad (\bar{\mathbf{p}}_h, \bar{R}_h) = \begin{cases} (\mathbf{p}_{t+h-1}, R_{t+h-1}) & \text{delta,} \\ (\mathbf{p}_t, R_t) & \text{relative,} \\ (\mathbf{0}, \mathbf{I}) & \text{absolute.} \end{cases} \quad (\text{D.1})$$

Here $\text{Log} : \text{SO}(3) \rightarrow \mathbb{R}^3$ returns the axis-angle vector, so $R_h^* = \text{Exp}(\mathbf{a}_h^{\text{rot}}) \bar{R}_h$. The gripper coordinate $\mathbf{a}_h^g$ is the same in all three. Delta actions are per-step pose increments, measured from the pose the robot has reached when the command is sent. Relative actions measure every target in the chunk from the end-effector pose at prediction time; all policies in the main paper use them. Absolute actions are target poses in the robot base frame.

Table 6: RoboCasa success counts with other backbone settings. Each row is the best operating point for that head; all three suites are evaluated at that point. Replan horizon 24.

Backbone	Head	CFG / NFE	Atomic	Composite seen	Composite unseen	Overall SR (%)
$\pi_{0.5}$, fine-tuned	CR	0 / -	117/180	39/160	56/160	42.4
$\pi_{0.5}$, fine-tuned	CF	2 / 20	148/180	69/160	74/160	58.2
$\pi_{0.5}$, fine-tuned	DR	2 / -	144/180	62/160	73/160	55.8
$\pi_{0.5}$, fine-tuned	DF	2 / 5	148/180	75/160	78/160	60.2
π_0 , frozen	CR	0 / -	49/180	1/160	1/160	10.2
π_0 , frozen	CF	2 / 20	55/180	6/160	6/160	13.4
π_0 , frozen	DR	2 / -	45/180	4/160	2/160	10.2
π_0 , frozen	DF	2 / 5	49/180	5/160	2/160	11.2

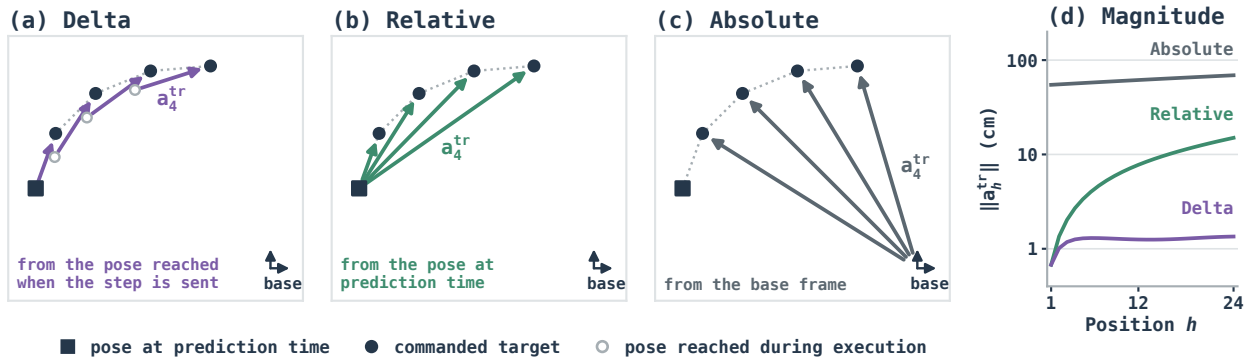


Figure 7: **Delta, relative, and absolute actions.** (a)–(c) One planned chunk in the robot base frame. Each arrow is the translation a_h^{tr} that the parameterization encodes for one command, drawn from its reference pose in Eq. (D.1). Rotations use the same reference poses. (d) Translation magnitude encoded along a 24-step chunk of an illustrative motion, on a logarithmic axis. The trajectories are illustrative, and (a)–(c) are not to scale.

Three properties follow from Eq. (D.1). At $h = 1$, delta and relative actions coincide. Relative and absolute actions fix every target of the chunk at prediction time, while a delta target also depends on the pose reached during execution. Relative actions start near zero and grow along the chunk, while absolute actions also carry the robot’s position in the workspace, so their range is set by the workspace rather than by the motion (Fig. 7(d)).

Results. Each entry of Tab. 7 uses the decoding setting (CFG and, for the flow heads, NFE) chosen separately for that head and action space. Relative actions give the highest success for DF, CF, and CR. DR does slightly better with delta actions (50.2% against 49.2%). Absolute actions give the lowest success for every head. The drop from relative to absolute actions is smaller for the discrete heads, 2.4 points for DR and 6.9 for DF, than for the continuous heads, 13.7 points for CF and 21.4 for CR.

Table 7: Action-space comparison on RoboCasa. Success rate (%) of the best of the last five checkpoints, as mean $\pm$ standard deviation over five evaluation seeds. Bold marks the highest success rate in each row.

Head	Delta	Relative	Absolute
DF	51.6 $\pm$ 2.6	53.9 $\pm$ 1.2	47.0 $\pm$ 1.3
CF	50.2 $\pm$ 3.5	52.7 $\pm$ 3.4	39.0 $\pm$ 2.1
DR	50.2 $\pm$ 2.9	49.2 $\pm$ 1.4	46.8 $\pm$ 1.9
CR	37.3 $\pm$ 1.8	38.8 $\pm$ 3.3	17.4 $\pm$ 3.1

E Temporal Variation and Policy Responsiveness Experiments

This section supplements [Section 3](#).

E.1 Temporal Variation on Held-Out Chunks

Figure 2: offline replay. Each of eight sources supplies 1,000 matched 24-step chunks, giving 32,000 model–chunk observations. All heads use CFG 0; both flow heads use NFE 10. Jerk is the mean norm of third finite differences of translation commands, without physical-time normalization. The ratio divides predicted jerk by target jerk after the same normalization and clipping.

The logarithmic scatter and its coordinate-wise medians omit 488 exact-zero ratios and 400 zero-target-denominator observations. All positive finite ratios remain. EEF violins retain all observations, with independently normalized density widths, medians, and interquartile ranges. Their horizontal positions are for layout only. Vertical scales vary by source. Including zero numerators wherever the denominator is positive, CR’s per-source median ratio ranges from 0.036 to 0.286, and CF’s from 0.231 to 0.549. CR lies below one on 98.8–99.8% of defined-ratio chunks and CF on 85.0–98.1%. The corresponding median ranges are 0.294–1.313 for DR and 0.343–0.589 for DF. DF’s EEF-error 95th percentile is below DR’s on seven sources, with LIBERO-plus the exception.

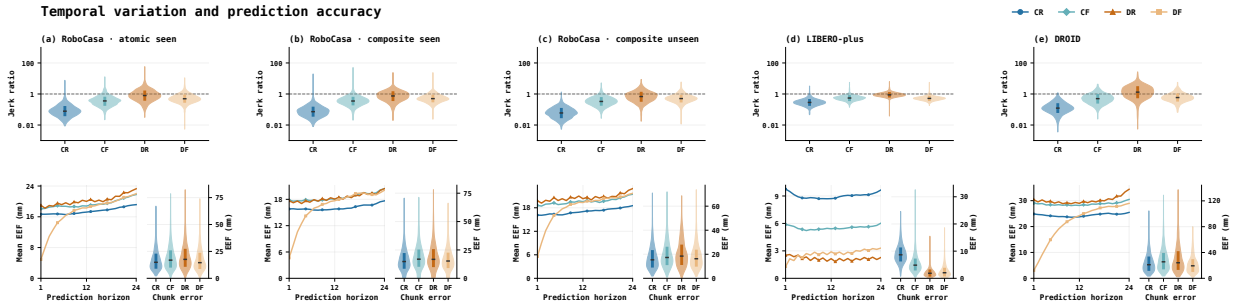


Figure 8: **Per-source view of Figure 2.** Top: jerk-ratio violins for each of the five displayed sources. Bottom: mean EEF error by horizon (left axis) and chunk-error violins (right axis).

E.2 Prediction Error Under Varying CFG

[Tab. 1](#) reports that CFG raises DF’s EEF error. To find the source of this increase, we sweep the guidance weight from $\lambda = 0$ to 3 in steps of 0.25. We use 4,000 held-out observations, 500 from each of 8 validation sources (DROID, LIBERO, and six RoboCasa splits). The predictions come from the dense CFG sweep in [Fig. 4\(c\)](#). Error is the per-step XYZ Euclidean distance to the ground-truth action in physical units, averaged over the valid steps of the chunk, then over draws, then over observations. All heads are scored against the same target.

Pooled over all sources ([Fig. 9a](#)). DF has lower error than DR at small λ : 15.16 mm versus 18.06 mm unguided. DF’s error is lowest at $\lambda = 0.5$ (14.86 mm) and then rises to 17.97 mm at $\lambda = 3$, passing DR at about $\lambda = 2.5$. DR’s error falls slightly with guidance and is nearly flat for $\lambda \geq 1$ (17.44–17.53 mm).

E.3 Distributional Robustness and the Cost of Contraction

[zeji: Rephrase the sub-title]

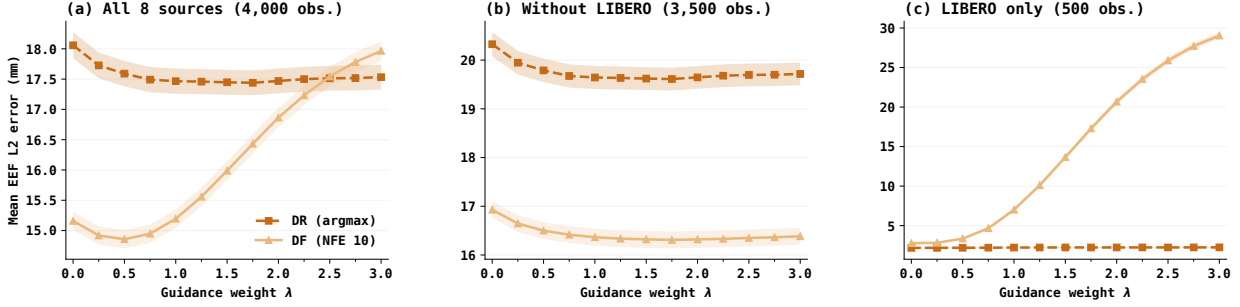


Figure 9: EEf error against ground truth as a function of the guidance weight λ . Mean XYZ end-effector L2 error (mm) over the 24-step chunk, for DR (argmax decoding) and DF (NFE 10, mean over 32 draws). Shaded bands show ± 1 standard error over observations. $\lambda = 0$ is unguided (Appendix C.5). (a) All 8 validation sources. (b) The 7 sources other than LIBERO. (c) LIBERO only.

Table 8: Proposal spread and response to one-step simulator disturbances. σ is mean nominal RMS spread over the full bank. The independent 32-vs-32 transport statistics use the existing 24-snapshot subsample per head and every other perturbation. Triples correspond to $\epsilon = 0.1, 0.3, 1.0$. Relative excess is computed from the displayed rounded transport means. CR’s deterministic law has zero floor, so its relative excess is undefined. Spread and transport use different subsets.

Head	σ	F_h	$T_h - F_h$	R_h
DF	1.217	1.245	0.233 / 0.490 / 1.071	0.19 / 0.39 / 0.86
CF	0.764	0.591	0.036 / 0.102 / 0.403	0.06 / 0.17 / 0.68
DR	0.182	0.141	0.301 / 0.529 / 0.927	2.13 / 3.75 / 6.57
CR	0.000	0.000	0.147 / 0.275 / 0.568	undefined

Question Can a policy preserve diverse actions while remaining robust and responsive to observation changes?

Our tests reveal two sides of the tradeoff: flow heads retain broader proposals with smaller relative disturbance responses, but excessive contraction can suppress the action changes required by real trajectories.

E.3.1 Flows Retain Broader Proposals With Smaller Relative Changes

We perturb executed actions, render the resulting observations, and compare the distributions of predicted chunks at CFG 0 (NFE 10 for flows). Table 8 reports RMS proposal spread σ , same-observation sampling distance F_h , and disturbance-induced distance T_h , measured by empirical Wasserstein distance W_2 . Relative excess transport is $R_h = (T_h - F_h)/F_h$; it describes change relative to the sampling baseline. Definitions and protocol are in Appendix E.4.

Flow proposals are broader than regression proposals, with DF broader than CF ($\sigma = 1.217$ versus 0.764). Their disturbance responses are also smaller relative to the same-observation sampling baseline. At $\epsilon = 0.1$, excess transport is 19% of that baseline for DF and 6% for CF versus 213% for DR. DR’s narrow proposal distribution shifts substantially relative to its own variability; CR also shifts despite having no proposal spread. This suggests a mechanism by which regression errors could compound into out-of-distribution drift.

E.3.2 CF’s Stronger Contraction Comes at a Cost in Responsiveness

CF stays closer to nearby training actions: its expert-calibrated excess distance is -0.100 , versus -0.015 for DF. This apparent adherence can also make a policy too “sticky.” We test that possibility on real frame pairs, comparing the head’s raw proposal-mean change with the expert’s chunk change under the same state displacement. Dividing either change by $\|\Delta s\|$ defines an empirical Lipschitz-like response; the table reports the underlying magnitudes directly. Sampling floors are not subtracted.

DF responds more than CF and is closer to the expert change at every tested lag (Table 9). At lag one, the expert shift is 0.66, compared with 0.45 for DF and 0.36 for CF; at lag 16, the values are 4.32, 3.55, and 3.28. Both flows underfit the observed response magnitude, but CF does so more strongly. DR tracks the expert magnitude most closely, showing that responsiveness on real trajectories and stability under disturbances are distinct requirements.

Table 9: Action changes on held-out expert trajectories. $\|\Delta s\|$ is the state displacement and expert Δchunk is the demonstrated chunk change. Head columns report raw proposal-mean shifts; parentheses show same-observation sampling floors for the stochastic heads. All values are medians over pairs in normalized units; no floor subtraction or response-ratio normalization is applied.

Lag	$\ \Delta s\ $	Expert Δchunk	DR (floor)	CR	DF (floor)	CF (floor)
1	0.018	0.66	0.71 (0.05)	0.35	0.45 (0.38)	0.36 (0.21)
2	0.036	1.13	1.16 (0.05)	0.62	0.77 (0.37)	0.64 (0.22)
4	0.073	1.88	1.77 (0.05)	1.09	1.32 (0.38)	1.18 (0.21)
8	0.149	3.00	2.71 (0.05)	1.93	2.34 (0.38)	2.25 (0.22)
16	0.292	4.32	4.01 (0.05)	3.13	3.55 (0.38)	3.28 (0.22)

Takeaway Flow preserves broader proposals with smaller relative disturbance responses. Within the flow family, CF is more contracted toward training actions but under-responds more to real trajectory changes; DF retains broader proposals and more of the expert’s response. This is the tradeoff between distributional stability and responsive control.

E.4 Disturbance and Expert-Reference Diagnostic Protocol

This appendix gives the measurement definitions and sampling protocol for Appendix E.3. DR uses its deployed argmax decoder; its residual proposal spread comes from varying random source-token inputs. CR is deterministic.

E.4.1 Measuring a Distribution’s Response to Disturbance

Let $p_h(\cdot \mid o)$ be head h ’s law over decoded, normalized action chunks. Its spread and its sensitivity are different quantities:

$$\sigma_h(o) = \left(\mathbb{E}_{A \sim p_h(\cdot \mid o)} \|A - \mu_h(o)\|_2^2 \right)^{1/2}, \quad (\text{E.1})$$

$$W_2(p, q) = \left[\inf_{\gamma \in \Pi(p, q)} \mathbb{E}_{(A, B) \sim \gamma} \|A - B\|_2^2 \right]^{1/2}, \quad (\text{E.2})$$

where $\mu_h(o) = \mathbb{E}[A \mid o]$ and $\Pi(p, q)$ denotes joint distributions with marginals p, q . Spread measures diversity at one observation; $W_2(p_h(\cdot \mid o), p_h(\cdot \mid o'))$ measures how far probability mass must move after a disturbance. Neither measures support cardinality.

We intervene on the simulator, because adding noise only to the proprioceptive input produced no measurable response. At 144 snapshots from DF rollouts across 18 RoboCasa atomic tasks, we restore the simulator state and execute either the nominal action or an action perturbed by $\eta \sim \mathcal{N}(0, \epsilon^2 I)$ in normalized action coordinates. After one control step, we render the resulting observations and query each head at CFG 0 (NFE 10 for flows), drawing 64 chunks per observation.

Relative change requires a same-observation baseline. Even two independent finite samples from the same law have nonzero empirical W_2 . Split the nominal draws into independent groups C_1, C_2 and use perturbed draws P_2 with source noise independent of C_1 . Each group contains 32 chunks. Exact optimal assignment gives

$$F_h = \langle \widehat{W}_2(C_1, C_2) \rangle, \quad T_h(\epsilon) = \langle \widehat{W}_2(C_1, P_2) \rangle, \quad R_h(\epsilon) = \frac{T_h(\epsilon) - F_h}{F_h}. \quad (\text{E.3})$$

Brackets average over the evaluated snapshots and perturbations; R_h is a ratio of aggregate means. We call it *relative excess transport*: how much the measured between-observation distance exceeds the within-observation sampling baseline. For example, $R = 0.2$ means 20% above that baseline. The ratio depends on sample count and distribution shape; a large floor can reduce it. We therefore report spread, absolute excess, and relative excess together.

Reference adherence must be calibrated against real actions. For held-out observation o , let $M(o)$ contain the $k = 5$ nearest training action chunks, with neighbors selected in normalized proprioception and restricted to the same instruction when enough matches exist. Define

$$r_0(A; o) = \min_{B \in M(o)} \|A - B\|_2, \quad E_h = \frac{\mathbb{E}_o[\mathbb{E}_{A \sim p_h} r_0(A; o) - r_0(A_{\text{gt}}; o)]}{\text{median}_o r_0(A_{\text{gt}}; o)}. \quad (\text{E.4})$$

We compute E_h within each source, then average over eight sources (512 held-out observations per source, 16 proposals each). The held-out expert reference matters: genuine demonstrated actions also lie some distance from a finite training set. A negative excess means predictions lie closer to this reference than held-out demonstrations do.

Gripper support. Under a bin-scale tolerance, the fraction of gripper commands outside the demonstrated values is about 0.22 for discrete heads and 0.78–0.94 for continuous heads.

The raw on-data shifts in Table 9 compare change magnitudes. Adjacent chunks also shift their temporal window, so the comparison includes replanning alignment. Sampling floors are reported separately and are not subtracted from these shifts.

The one-step disturbance changes normalized proprioception by about 0.003–0.021, below the reference data’s median neighbor spacing of 0.32. The diagnostics use CFG 0 and NFE 10, while reported control success uses separately selected settings.

F Factorization Error Experiments

This section supplements Section 4.

F.1 Experimental Methodology

Mode-detection criteria. For each translation-token distribution $p_j(b)$, we smooth probabilities along the bin axis with a Gaussian kernel of standard deviation two bins, normalized over the available bins

at each location. We identify local maxima, including endpoint maxima, in the smoothed distribution. Each bin is assigned to its nearest peak, and each peak’s mass is the sum of the *unsmoothed* probabilities assigned to it. We retain peaks with at least 10% mass. Two adjacent retained peaks qualify as separated modes if they are at least eight bins apart and the minimum smoothed probability between them is at most 50% of the smaller peak height. Each translation token represents one coordinate at one predicted time step. We call it ambiguous if it has a qualifying pair of modes, and count it once regardless of how many such pairs it has. The mean count averages ambiguous tokens per chunk. The threshold statistic reports the fraction of chunks containing at least k ambiguous tokens. For DF, detection is performed on each final conditional token prediction before averaging over refinement draws.

NFE sweep in Figure 4(a,b). We use 4,000 validation observations (500 from each of eight sources) and 16 DF draws per observation at each NFE and CFG 0. Each chunk has 39 translation tokens: three coordinates at 13 time steps. Panel (a) reports the mean number of ambiguous tokens per chunk under the mode detector above; Figure 10(a) gives the threshold curves. Panel (b) reports percentiles of $\exp(\mathcal{H}(p))$ over all translation tokens, draws, and observations. DR references use the same observations and detector. Because the final query time is $t = 1 - 1/\text{NFE}$, the sweep changes both the number of refinement steps and the time of the last prediction. The mean count is 4.94 at NFE 2 and falls to 1.50, 0.57, and 0.22 at NFE 5, 10, and 20 (DR: 3.01). Figure 12 reports entropy and action diversity for the same sweep.

E2 Factorization Error and L_2 Loss Diagnostic Results

Tab. 10 reports the L_2 prediction errors referenced in Section 3. The remaining figures give additional mode-ambiguity diagnostics for Fig. 4; Fig. 10(b) breaks W_{90} down by horizon.

Table 10: EEF target-prediction error ($\downarrow$). Slow-validation translation error in mm near 100k training steps; CF uses 100 integration steps and DF 10. Regression uses one pass. These are single-run diagnostics without repeated-seed uncertainty.

Dataset / source	CR	CF	DR	DF
<i>Slow validation (mm)</i>				
DROID	21.60	26.60	25.09	19.67
LIBERO-plus	9.88	7.23	2.53	2.90
RoboCasa pretrain composite	16.44	17.34	15.97	14.84
RoboCasa MimicGen	16.18	16.95	15.76	12.55
RoboCasa target atomic	17.34	18.97	19.55	15.62
RoboCasa composite-seen	12.11	14.87	15.39	12.93

E3 Task Performance Under Varying NFE and CFG

We use the same evaluation protocol as in the rest of the paper (Appendix B.3), with one fixed checkpoint per head. Only the number of decoding steps (NFE) and the guidance weight λ (Appendix C.5) change.

E3.1 Task Performance Under Varying NFE

Fig. 13(a,b) shows that the effect of NFE depends on guidance. Without guidance, both flow heads lose about five points at NFE 2, where DF reaches $47.0 \pm 2.5\%$ and CF $45.6 \pm 1.3\%$. From NFE 5 to 100 they change little: DF ranges from 51.2% to 52.5% and CF from 50.8% to 51.9%, with no consistent gain from more steps. With $\lambda = 2$, DF changes little at every budget, ranging from 52.3% to 53.6%. Guided

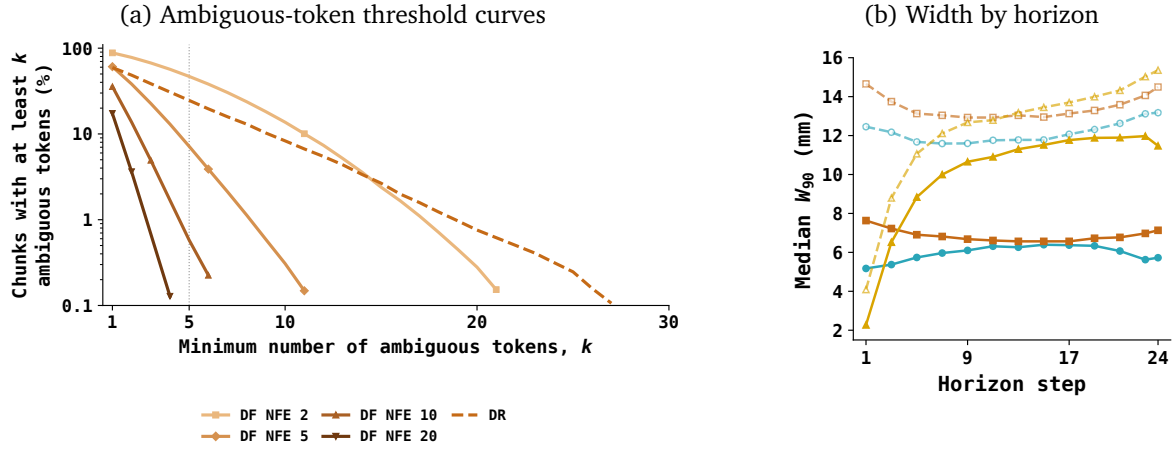


Figure 10: **Ambiguity thresholds and width by horizon.** (a) Percentage of chunks with at least k ambiguous tokens at CFG 0, on the same observations and draws as Fig. 4(a). The log axis stops at 0.1%. (b) Median central-90% width W_{90} at each horizon step for CF (teal), DR (orange), and DF (gold). Dashed lines with open markers are CFG 0; solid lines with filled markers are CFG 2.

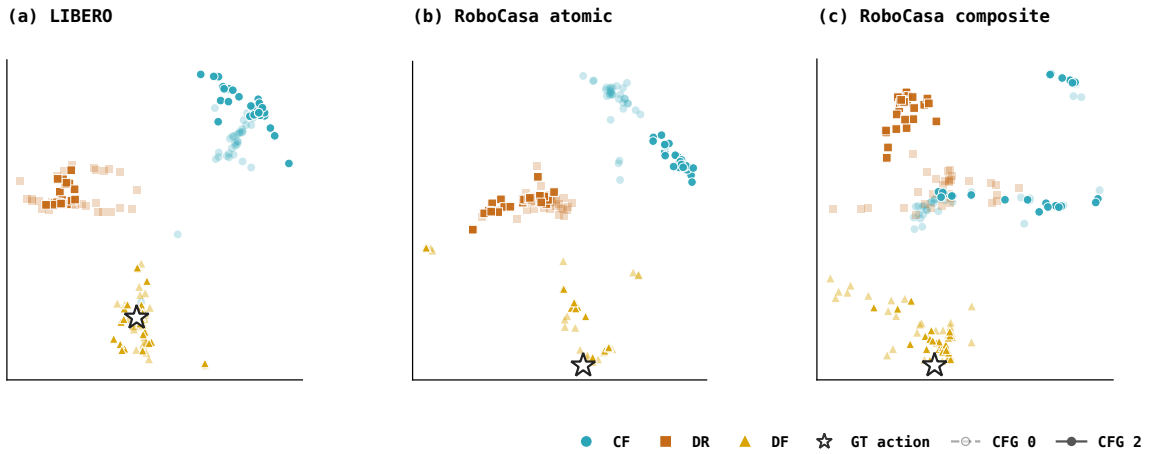


Figure 11: **Additional t-SNE views of sampled action chunks.** LIBERO, RoboCasa atomic, and RoboCasa composite supplement the DROID example in Figure 4. Colors and marker shapes identify CF, DR, and DF; faded and solid marks denote CFG 0 and CFG 2, respectively. Stars denote reference actions. The original embeddings are preserved; these projections provide a qualitative view of concentration under guidance.

CF, however, fails under coarse decoding: its success falls to $0.3 \pm 0.3\%$ at NFE 2 and $41.9 \pm 1.7\%$ at NFE 5, then recovers to $52.4 \pm 2.4\%$ at NFE 10. Only guided CF fails outright; unguided CF still reaches 45.6% at NFE 2.

F.3.2 Task Performance Under Varying CFG

We next vary λ with NFE fixed at 10 for both flow heads (Fig. 13(c)); DR has no NFE axis. Guidance changes DR far more than either flow head. DR improves from $27.1 \pm 1.2\%$ without guidance to $48.9 \pm 3.6\%$ at $\lambda = 2$, and reaches its highest mean, $50.8 \pm 2.8\%$, at $\lambda = 3.5$. Over the same range, its mean success on composite-seen and composite-unseen tasks rises from about 10% and 12% to 34.8%

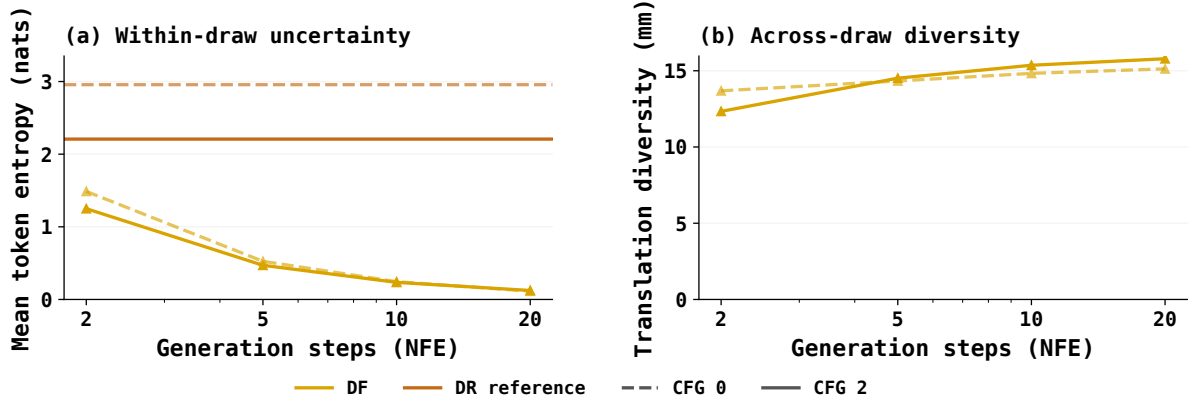


Figure 12: **Refinement sharpens conditionals while retaining across-draw variation.** Mean within-draw translation-token entropy (a) and across-draw translation-command diversity (b) use the same 4,000 validation observations and 16 draws per observation as Figure 4. Gold curves show DF; orange horizontal lines are DR references. Dashed and solid curves denote CFG 0 and CFG 2. From NFE 5 to 20, entropy decreases while action diversity persists. Diversity is the RMS pairwise Euclidean translation-command distance across draws, averaged over horizon positions and observations, in mm.

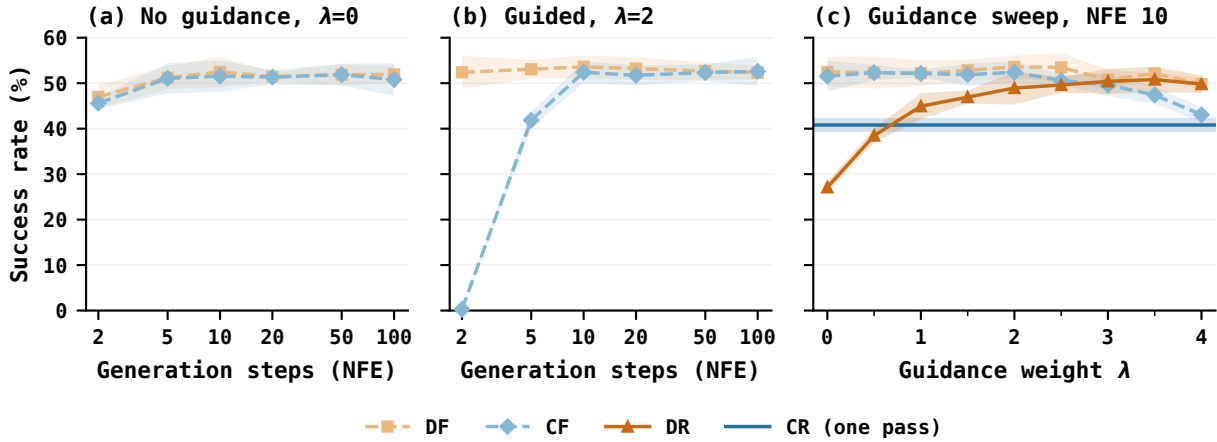


Figure 13: **RoboCasa success rate under varying NFE and guidance.** Each head uses one fixed checkpoint. (a,b) NFE sweep for DF and CF at guidance weight $\lambda = 0$ and $\lambda = 2$. (c) Guidance sweep with DF and CF at NFE 10; DR has no NFE axis. The CR line is an unguided one-pass reference. Lines show the mean and bands the sample SD over three evaluation seeds.

and 39.4%.

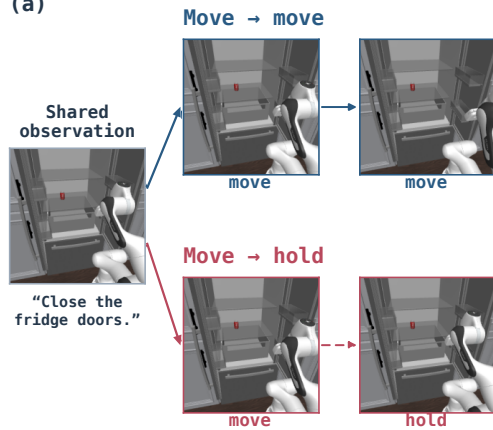
Moderate guidance changes the flow heads only slightly: from $\lambda = 0$ to 2, DF gains 1.1 points and CF 0.9 points. Strong guidance hurts, most of all for CF, which falls to $43.1 \pm 1.6\%$ at $\lambda = 4$, where DF obtains $49.8 \pm 1.4\%$. The best guidance weight therefore depends on the head and the decoding budget. For reference, CR reaches $40.8 \pm 1.4\%$ in one unguided pass.

E.4 Visualization of Mode Inconsistency in RoboCasa

RoboCasa move–hold example. Figure 14 uses CloseFridge marginals and a schematic mode illustration: moved thumbnails are taken from steps 12 and 24 of a DR-argmax replay; the hold thumbnail

RoboCasa

(a)



(b)

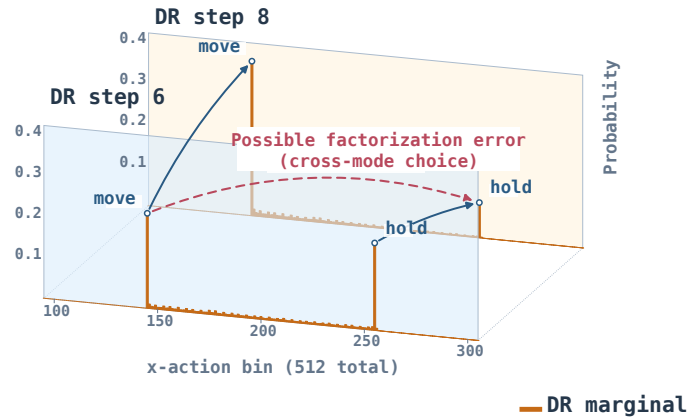


Figure 14: **a** Robocasa example showing that in the same chunk, robot arm has two mode one is move consistently, the second is first move then hold; **b** We show that this independent token sampling could cause factorization error, leads to switching between motions on different timestep, the same as what we have seen in push-T [Fig. 5](#)

repeats the initial image.

G Tokenization and Geometric Priors

This section supplements [Section 5](#).

G.1 Performance Under Different Tokenization

Closed-loop results for tokenizers. Table 11 compares nFSQ, RVQ, and FAST with their frequency-bin baselines. For each tokenizer and head, we report the operating point with the best pooled success, selecting CFG, the replanning horizon, and, for DF, the NFE among 5, 10, and 20. The same setting is used on all three suites. Pooled success combines 180 atomic-seen episodes and 160 episodes from each composite suite.

The nFSQ result (52.4%) is slightly above the frequency-bin metric-path result (51.8%). Neither exceeds the main mixture-path DF result of 53.9%, which uses a different recipe and horizon.

FAST framing and validity. FAST compresses action sequences [Pertsch et al., 2025], so generation must respect its sequence grammar. During training, valid decodes were near 100% for AR, 25–40% for DF, and 0% for DR. The FAST DR run still had 0% validity at the end of training, so it has no closed-loop evaluation. Flat scalar bins do not have this failure: every token sequence decodes. With EOS-terminated framing, FAST AR reaches 48.2% with CFG 2; FAST DF reaches 46.2%.

Tokenizer	Head	CFG	NFE	Atomic	Composite seen	Composite unseen	Overall SR (%)
Frequency bins	DR	0	–	140/180	57/160	52/160	49.8
	DF	2	20	128/180	50/160	52/160	46.0
Frequency bins + metric path	DF	2	20	136/180	67/160	56/160	51.8
nFSQ	DR	2	–	121/180	40/160	47/160	41.6
	DF	2	5	139/180	61/160	62/160	52.4
RVQ	DR	0	–	59/180	12/160	1/160	14.4
	DF	2	50*	113/180	40/160	38/160	38.2
FAST	AR	2	–	131/180	56/160	54/160	48.2
	DF	2	20	123/180	55/160	53/160	46.2

*RVQ needs more NFE to decode, so DF RVQ uses NFE 50.

Table 11: **RoboCasa success by tokenizer and head.** Each row is the best operating point for that tokenizer and head. No tokenizer lifts DF above the best flat-bin DF. Frequency bins behave differently for DR: CFG 0 and CFG 2 give almost the same success (49.8% and 49.0%), and at CFG 0 DR is far above flat-bin DR (27.1%, [Section 4](#)). With frequency bins every token is used equally often, so each token carries more information and should be harder to predict, yet single-pass DR handles it well. We do not yet have a concrete explanation, but the results suggest that when inference computation is limited, how actions are encoded matters more.

G.2 Geometry-Aware Probability Paths and Evaluations

This appendix specifies the metric-path interventions summarized in [Section 5](#). The path illustration is analytical; the success rates below come from trained-policy evaluations.

G.2.1 From a Mixture Path to a Metric-Induced Path

Let $y \in \{0, \dots, B-1\}$ denote a clean action token and let $s \in [0, 1]$ increase from noise toward the clean action. A probability path specifies the distribution of corrupted tokens presented to the model at each stage. With a uniform source, the mixture path is

$$p_s^{\text{mix}}(k | y) = \frac{1-s}{B} + s \mathbf{1}\{k = y\}. \quad (\text{G.1})$$

Here s is the mixture weight; a separate schedule maps inference time to s . Increasing s raises the probability of the clean token, while every incorrect bin retains the same probability. The corruption does not distinguish a neighboring action value from a distant one.

The metric-induced path introduces this distinction through a Gibbs kernel:

$$p_s^{\text{metric}}(k | y) = \frac{\exp[-\beta(s)d_p(k, y)]}{\sum_{j=0}^{B-1} \exp[-\beta(s)d_p(j, y)]}, \quad d_p(k, y) = \frac{|k - y|^p}{(B-1)/2}. \quad (\text{G.2})$$

At $\beta = 0$ the distribution is uniform; as β increases, probability concentrates around the clean bin, approaching a point mass as $\beta \rightarrow \infty$. For finite positive β , nearby errors are more probable than distant errors. We use $p = 1$ for absolute bin distance and $p = 2$ for squared bin distance. Uniform scalar bins make index distance proportional to distance in normalized action values; equal-frequency bins instead impose this structure in quantile space.

Figure 15 isolates the difference at matched expected bin error. The mixture path can achieve a low average error by being exactly correct often, while its remaining errors are spread across the vocabulary. The metric path instead allocates more of its error to nearby bins. This changes the denoising problem even when the average corruption magnitude is the same.

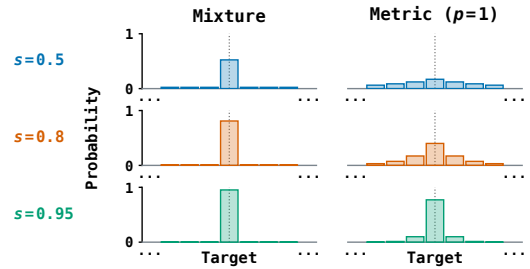


Figure 15: Conditional corruption distributions at matched expected bin error. Colors mark path progress $s = \alpha(t)$; the target is bin 10. Mixture corruption keeps a uniform background, while metric corruption concentrates around the target.

The path and its clock are separate choices. The Gibbs kernel defines a family of distributions; $\beta(s)$ determines how quickly the model moves through that family. Our bin-progress clock is calibrated so that

$$\bar{e}(\beta(s)) = (1-s)\bar{e}(0) + s\bar{e}(\beta_{\max}), \quad \bar{e}(\beta) = \mathbb{E}_{y \sim \pi} \sum_k p_\beta(k | y) |k - y|, \quad (\text{G.3})$$

where the implementation uses a uniform reference histogram π over clean bins. The entropy-progress clock instead linearly decreases the histogram-averaged conditional entropy from $\log B$ to its value at $\beta_{\max}$. Both clocks are numerically inverted and stored as fixed lookup tables. The finite endpoint $\beta_{\max}$ bounds the worst-bin token error using the configured chunk error budget. Thus changing the clock changes the training and inference schedule without changing the form of the Gibbs kernel.

Learning clean tokens and decoding with distance-aware rates. Training retains clean-token cross-entropy, $\mathbb{E}[-\log q_\theta(y | X_s, s, o)]$, summed over action-token positions; X_s is the corrupted chunk and o is the observation. Geometry enters through the corruption path. For a current token i and a sampled clean

endpoint $\hat{y}$ from the network posterior, the implementation uses kinetic-optimal discrete flow matching (KO-DFM) off-diagonal rates

$$u_s(j | i, \hat{y}) = \dot{\beta}(s) p_s^{\text{metric}}(j | \hat{y}) [d_p(i, \hat{y}) - d_p(j, \hat{y})]_+, \quad j \neq i. \quad (\text{G.4})$$

Here $[x]_+ = \max(x, 0)$. Conditional on that endpoint, only moves to closer bins have positive rates. The decoder caps each token’s rate-vector norm at 50 before the discrete update.

G.2.2 Evaluated Metric-Path Variants

The bin-progress $p = 1$ and entropy-progress $p = 2$ variants are completed training runs with RoboCasa evaluations. Their best pooled success rates are 52.8% and 4.6%, respectively. Applying the bin-progress path to the equal-frequency representation reaches 51.8%. Each rate uses one operating point across all three suites. The first result is competitive with, but does not exceed, the mixture-path DF result of 53.9%. The effect of explicit geometry depends strongly on how the path is constructed.

G.3 Bin Representation and Input-Sensitivity Study

We analyze DR and DF policies trained on XDof bus-table (plates) and bottle data. These policies use relative stretch actions, a 16-step horizon, 14 action dimensions, a shared 512-bin vocabulary, and 1,536-dimensional bin representations. The checkpoints are DR step 55,000 and DF step 40,000.

Input tables subtract the global mean of all vocabulary rows, while occupied output tables subtract the occupied-row mean before cosine normalization. The analysis restricts the headline statistics to bins with at least 50 aggregate occurrences in the validation sample: 222 bins (145–366). Occupancy was counted over 100,794 valid frames, or 1,411,116 scalar action values; 290 bins had no observed occupancy. The gate is important because unused tails and common components can distort all-vocabulary similarities. Bin separation is measured in the shared normalized coordinate system.

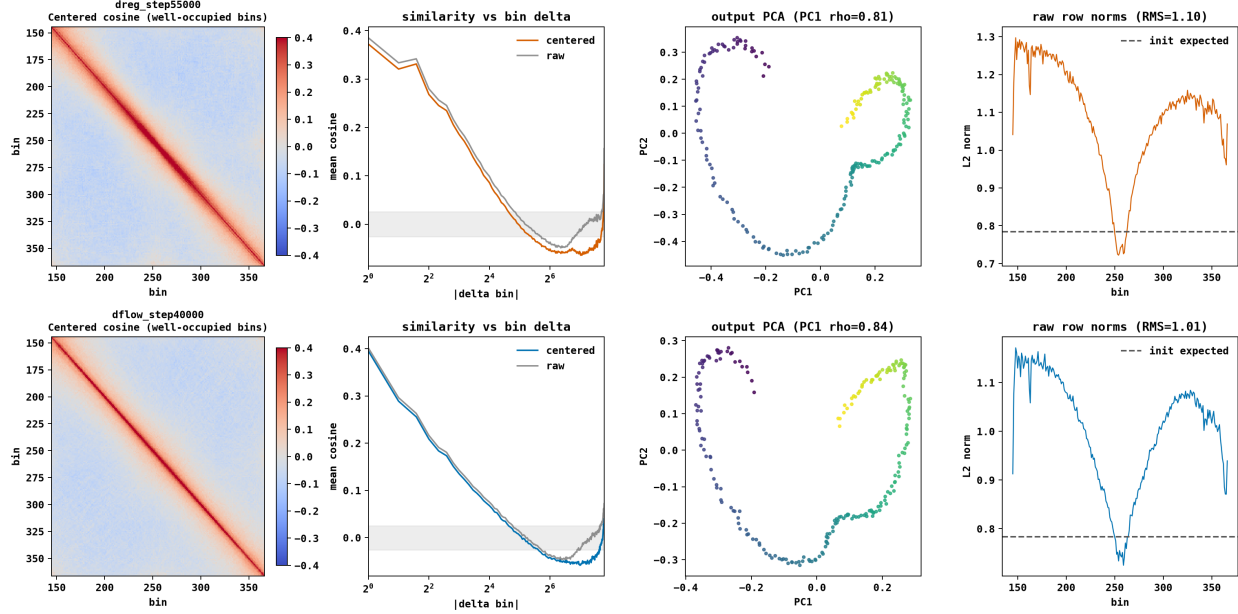


Figure 16: Output-weight diagnostics: DR (top), DF (bottom). Columns show centered cosine, lag-dependent similarity, PCA, and row norms. The banded matrices and decreasing local similarity support ordinal organization in both heads.

DR/DF centered input adjacency is $-0.0038/0.5733$, and output adjacency is $0.3724/0.3953$. Output embedding-distance versus value-distance Spearman correlations are $0.5835/0.5579$. Bin-relabeling permutation tests give input adjacency $p = 0.81/0.001$ and output adjacency $p = 0.0005$ for both heads. DF input distances saturate at larger separations (reported far-to-near slope ratio 0.027), so the result supports local order. Positive timestep-embedding controls and shuffled-cluster negative controls were also run. Multiple DR checkpoints retain near-zero input adjacency.

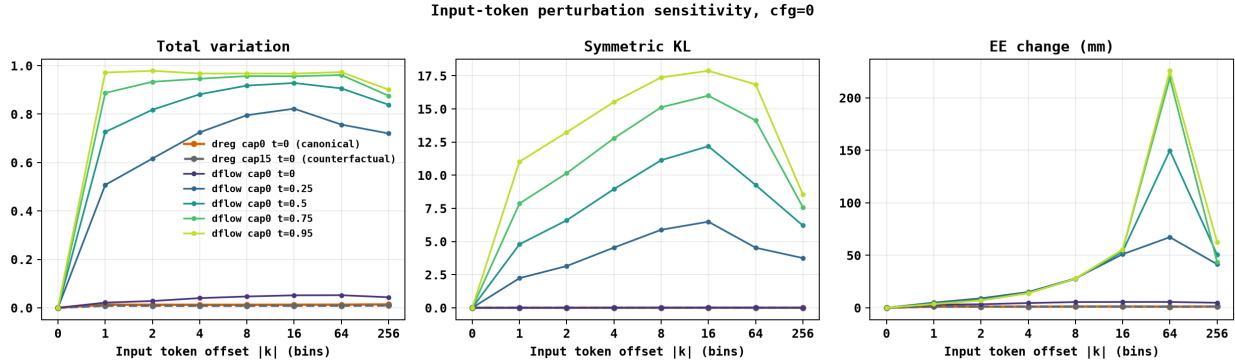


Figure 17: Forward-pass input-offset probes. The plot also shows a counterfactual softcap-15 DR trace; our interpretation uses only the canonical softcap-0 results.

Canonical posterior probes use 64 validation frames per head and report 14,252 valid output distributions per head; the input-sensitivity sweep uses eight frames per head, with no classifier-free guidance and effective logit softcap zero. DR total-variation responses remain approximately 0.0125–0.0141 over offsets of 1–256 bins. Replacing its random tokens with true tokens changes decoded EEF position by 1.18 mm. At DF time 0.75, offsets of 1, 16, and 64 bins change the decoded normalized output by 0.0093, 0.1368, and 0.5121. These interventions establish sensitivity to input tokens.

G.3.1 Combined Output-Representation Visualization

The output PCA projections also organize bins by value, although their first two components explain only about 11% (DR) and 9% (DF) of weight variance. Output biases track the empirical marginal frequency of occupied bins (Pearson $r = 0.809/0.720$ for DR/DF with log-one-plus occupancy).

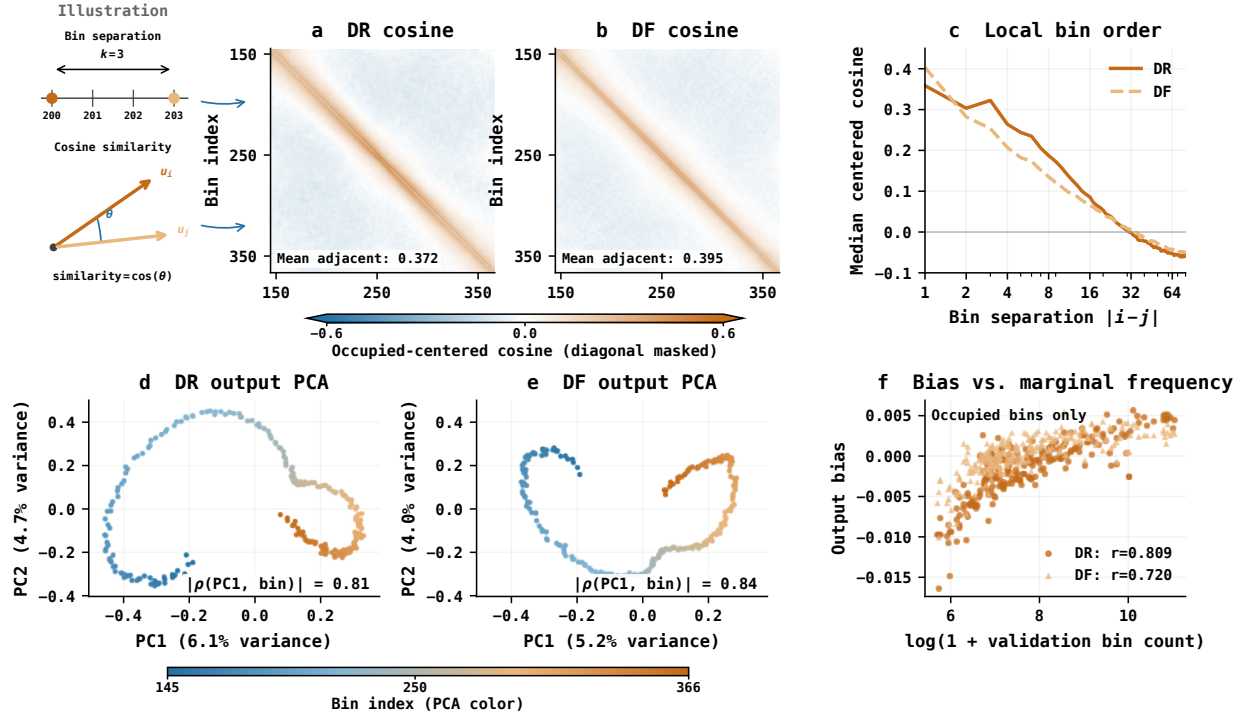


Figure 18: Both discrete heads organize output bins by value. Upper left: generated conceptual illustration defining bin separation and cosine; it contains no empirical measurements. (a,b) Output-weight cosine matrices on the same 222 occupied bins, centered over those rows, with self-similarity masked. (c) Median cosine over pairs at each exact bin separation; only output weights are shown. (d,e) Output-weight PCA, colored by bin index; axes state explained variance, and each head has its own PCA basis. (f) Output bias versus log validation occupancy on the same occupied bins; Pearson $r = 0.809/0.720$ for DR/DF. Zero-count bins are excluded. Scope and controls are in Appendix G.3.

H Synthetic Fitting Protocol and Results

H.1 Which Fine Structure Does Categorical Prediction Preserve?

The robot experiments raise a simple question: can a model get the broad motion right and still miss the fine detail? Robot data mixes too many factors to answer this, so we use synthetic targets where we control the fine structure directly.

Hypothesis 2: categorical prediction can preserve steep fine structure. Following the multiscale construction of Pan et al. [2026], each target is a broad sine plus a small fine-scale term. In the first target the fine term is a fast sine. In the second it is a square wave, which jumps between two values and so has no finite Lipschitz constant. CF and DF both predict clean targets.

Smooth fine detail: continuous regression wins. On the smooth target (Figure 19A), CR’s held-out error is about 250 times lower than DR’s. Both continuous heads go below the codec’s reconstruction floor. That floor limits any output decoded through argmax, but not continuous outputs. The discrete heads stay well above it, so bin resolution alone does not explain their larger error.

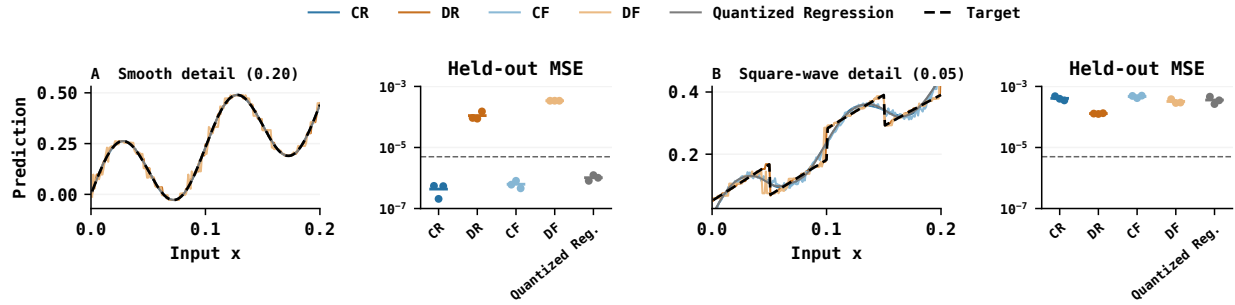


Figure 19: Precision depends on target structure. Each target pairs original predictions (seed 0) with held-out MSE for all three seeds; marks indicate means and the dashed line indicates the hard-bin codec floor. Prediction axes use the local target range; all errors remain included. CF/DF curves show one draw, while stochastic MSE averages eight per-draw errors. Both fine-term amplitude and shape change between targets.

Steep fine detail: the ordering reverses. On the square-wave target (Figure 19B), DR has the lowest held-out error of all heads, with 3.2 times lower MSE and 4.5 times lower MAE than CR. DF also beats both continuous heads, with 31 percent lower error than CF. The prediction panel shows the difference. The regression heads follow the steps but round off the corners, while the categorical heads jump to the next bin right at the edge.

Controls narrow the explanation. Quantized Regression fits the bin centers with squared error. It behaves like CR on both targets, so discrete target values are not what drives DR’s result. CF tracks CR and DF trails DR on both targets, so the reversal shows up with one-pass and iterative prediction alike. That leaves the categorical formulation as a whole: loss, output parameterization, and argmax decoding. One likely reason is argmax itself. The logits can change smoothly while the decoded bin jumps, which lets the model represent sharp transitions.