How Do Latent Actions Generalize? Probing Ordered Transfer Structure in Action-Free Representations

Anonymous authors
Paper under double-blind review

Abstract

Latent action models learn transition representations from visual data without robot-action annotations, providing a potential interface between large-scale visual experience and downstream policy learning. If latent actions are intended to support generalization, however, a key question is how transition information should be organized in the latent space. We study whether action-free latent representations develop an ordered refinement structure that is systematically related to cross-distribution generalization. To make this process observable, we introduce a weak nested-access bias that exposes progressively expanded prefixes of the same transition representation, without action supervision, predefined token semantics, or assumed high-level/low-level skill hierarchies. We then analyze how action-related, state-transition, and environment-specific information evolve across refinements and distribution shifts. Across LIBERO, LIBERO-Plus, and MimicGen, we observe a consistent pattern: refinement increases in-distribution action recoverability, while environment-specific information also becomes more decodable and OOD performance does not improve monotonically. Under stronger distribution shifts, intermediate refinements can outperform the complete latent representation, with similar non-monotonic behavior in closed-loop control. These findings suggest that generalizable latent-action learning depends not only on how much transition information is learned, but also on how that information is organized during learning, offering a useful direction for designing more transferable latent-action objectives.

1 Introduction

A useful embodied agent must generalize beyond the demonstrations on which it was trained. Changes in objects, scenes, robot states, viewpoints, or visual appearance can substantially alter the observations encountered at deployment, while the underlying action-relevant structure may remain similar. Vision-language-action (VLA) models learn mappings from observations and language instructions to robot actions, whereas world models and world-action models explicitly model how the environment evolves and use such predictions to support planning or policy learning (Zitkovich et al., 2023; Kim et al., 2025; Bruce et al., 2024; Huang et al., 2026). Although these paradigms differ in formulation, both ultimately depend on representations that preserve action-relevant transition structure while avoiding over-reliance on incidental properties of the training distribution.

Latent action models (LAMs) provide an appealing interface for learning action-related transition representations from visual experience without robot-action annotations. For example, LAPA (Ye et al., 2025) encodes an observation transition $(o_t, o_{t+\Delta})$ into a discrete latent action, which is used for future prediction and can subsequently supervise VLA pretraining before robot-action finetuning. Because these latent codes need not correspond directly to executable robot commands, they provide a flexible abstraction for connecting large-scale visual experience with robot policy learning. If latent actions are intended to support generalization, however, a more fundamental question is how transition information should be organized within the learned representation.

Existing LAPA-style LAMs typically optimize and use the transition representation as a whole. Future prediction, action decodability, and downstream policy performance provide useful measures

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

of whether the latent is informative, but reveal little about how information is organized within the representation. We hypothesize that action-free latent transitions may admit a *progressive within-transition organization*, in which different refinements capture transition information with systematically different functional and transfer properties. Our central question is whether such an organization can emerge without action supervision, and whether it matters for cross-distribution generalization.

Inspired by Matryoshka representations and related nested learning objectives (Kusupati et al., 2022; Bussmann et al., 2025), we introduce a weak nested-access bias that makes progressively expanded representations of the same transition independently usable. This construction does not assign semantics to individual latent tokens, define high- and low-level skills, or use robot-action supervision. Importantly, the nested bias is not merely a passive visualization device: it changes the learning constraint imposed on the LAM while leaving the semantics of the resulting refinements unconstrained. We therefore use it both as a lightweight intervention on latent-action learning and as an analysis handle for studying how transition information becomes organized.

Our experiments show that this intervention does more than make the latent structure easier to inspect. Nested training can produce more transferable representations than conventional full-code training under several distribution shifts, and at matched full-code length the nested models also improve standard LIBERO closed-loop control over the Flat baseline. The resulting gain is therefore not simply obtained by sacrificing downstream task performance. Probing the learned refinements further reveals a consistent pattern: as the representation is expanded, in-distribution action information becomes increasingly recoverable, while environment-specific information also becomes more decodable and OOD performance does not improve monotonically.

Under stronger LIBERO-Plus shifts and in the MimicGen D2 evaluation, intermediate refinements can outperform the complete latent representation; a similar non-monotonic dependence appears in closed-loop control. We refer to the systematic change in action readability, environment dependence, and cross-distribution transfer across refinements as an *ordered transfer structure*. Here, "ordered" describes an empirical property of the learned representation rather than a semantic hierarchy imposed by construction.

Taken together, these results suggest that generalizable latent-action learning depends not only on how much transition information is captured, but also on how that information is organized during learning. The nested bias is therefore not the end goal of this work, but a simple intervention that exposes and reshapes an otherwise hidden aspect of latent-action learning, providing a direction for designing more transferable latent-action objectives.

We summarize our main contributions and findings below:

- We study latent-action generalization from the perspective of *within-transition information organization*, asking not only what transition information is encoded, but how its organization relates to cross-distribution generalization.
- We introduce a weak nested-access bias that acts as a lightweight intervention on latent-action learning while making progressive refinements of the same transition independently observable, without robot-action supervision, predefined token semantics, or skill hierarchies.
- Across LIBERO, LIBERO-Plus, and MimicGen, we observe an *ordered transfer structure*: refinement increases ID action recoverability while OOD performance becomes non-monotonic and environment-specific information becomes more decodable. Nested training also improves key transfer settings and matched full-code closed-loop control over standard Flat training, with intermediate refinements outperforming the complete representation under stronger shifts.

2 Related Work

VLA and predictive models for robot learning. Vision-language-action models connect visual-language pretraining with robot control by directly predicting actions from observations and instructions (Zitkovich et al., 2023; Kim et al., 2025), while world and predictive models learn visual dynamics for planning or policy learning (Bruce et al., 2024; Huang et al., 2026). Both directions

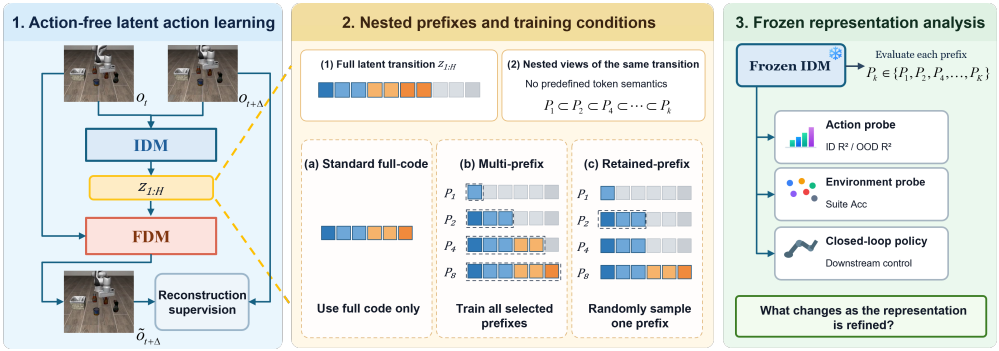


Figure 1: Overview of our framework. **Left:** an action-free LAM learns a latent transition representation from an observation transition. **Middle:** **(1)** the full latent transition is represented by a sequence of latent tokens, and **(2)** progressively expanded prefixes provide nested views of the same transition without predefined token semantics. We compare three training conditions: **(a)** *Standard full-code*, **(b)** *Multi-prefix*, and **(c)** *Retained-prefix*. **Right:** after training, the representation is frozen and each prefix is analyzed through action probes, environment probes, and closed-loop policy evaluation to study how representation refinement relates to transfer.

motivate reusable transition representations from large-scale visual experience. Our work instead focuses on the latent transition representation itself, asking how its internal organization relates to generalization before it is consumed by a downstream policy.

Latent action learning. Latent action models infer compact transition representations from observations and use them for future prediction or downstream policy learning (Ye et al., 2025; Chen et al., 2026). LAPA (Ye et al., 2025) shows that discrete latent actions can be learned without robot-action labels and used for VLA pretraining. Subsequent studies address physical grounding and viewpoint robustness (Chen et al., 2026; Jeong et al., 2026), align latent actions with action signals (Luo et al., 2026), and investigate their use in video-imagination policies and VLA supervision (Li et al., 2026; Lin et al., 2026). Recent analyses also suggest that predictive latent actions may retain transition-correlated distractors (Nikulin et al., 2025; Zhang et al., 2025). Our focus is complementary: we study whether an *action-free representation of the same transition* is progressively organized, and whether different levels of this organization exhibit different transfer behavior under distribution shift.

Structured and nested representations. Nested dropout and Matryoshka-style learning show that a representation can remain useful across multiple nested granularities, with later refinements adding increasingly specific information (Rippel et al., 2014; Kusupati et al., 2022; Bussmann et al., 2025). This observation motivates our hypothesis that action-free latent transitions may exhibit a similar progressive organization. HiLAM (Kim et al., 2026) studies hierarchy across temporal scales, whereas we study representation structure *within the same transition*. FlexLAM (Yoshimoto et al., 2026) is technically closer, using nested dropout for variable-length latent actions and bottleneck adaptation. In contrast, we use nested representations to make different levels of the same latent transition explicitly observable, and analyze how the information revealed through successive refinements transfers from ID to OOD.

3 Approach

Our approach makes different levels of an action-free latent transition explicitly accessible without assigning predefined semantics to individual latent tokens. We first learn a standard latent-action representation from visual transitions, and then introduce nested access to obtain progressively refined views of the same representation. We compare a conventional full-code objective with two forms of nested training, and analyze how the resulting refinements behave under distribution shift. Figure 1 provides an overview.

3.1 Base Latent Action Learning

Given an observation transition $\tau_t = (o_t, o_{t+\Delta})$, an inverse dynamics module (IDM) infers a discrete latent representation of the transition:

$$z_{1:H} = \mathrm{IDM}_\theta\left(o_t, o_{t+\Delta}\right). \tag{1}$$

Here, H denotes the number of tokens in the complete latent representation. The latent is inferred purely from visual change and is learned without ground-truth robot-action supervision. In our implementation, the IDM contains a frozen visual encoder, a transition encoder, and a vector-quantized bottleneck; implementation details are summarized in Sec. 4.1, with full architectural and optimization configurations provided in Appendix B.

A forward dynamics module (FDM) then predicts the future observation from the current observation and the inferred latent:

$$\widetilde{o}_{t+\Delta} = \mathrm{FDM}_\phi\left(o_t, z_{1:H}\right). \tag{2}$$

Following the standard operational definition in latent-action learning, we refer to $z_{1:H}$ as a *latent action*. It summarizes information about the observed transition that supports future prediction, but is not required to correspond one-to-one with an executable robot action.

The action-free training objective is

$$\mathcal{L}_{\mathrm{base}} = \mathbb{E}_{\tau_t}\left[\ell_{\mathrm{rec}}\left(\widetilde{o}_{t+\Delta}, o_{t+\Delta}\right) + \mathcal{L}_{\mathrm{VQ}}\right]. \tag{3}$$

where ℓ_{rec} is the reconstruction loss in representation space, and $\mathcal{L}_{\mathrm{VQ}}$ denotes the vector-quantization objective (van den Oord et al., 2017) used to learn the discrete latent code.

Robot actions, states, task identities, and environment labels are not used to learn the latent representation. They are introduced only after training to analyze what information the latent preserves and how that information transfers across distributions.

3.2 Nested Refinement of Latent Actions

The standard objective in Equation (3) requires only the complete latent representation to support future prediction, providing limited access to how information is organized within the code. We therefore introduce a nested representation in which progressively expanded prefixes of the same latent transition can be evaluated independently. For the complete latent chunk $z_{1:H}$, we define

$$[P_k^\nu(z)]_j = \begin{cases} z_j, & j \le k, \\ \nu, & j > k, \end{cases} \qquad k \in \mathcal{K}, \tag{4}$$

where $\mathcal{K}$ denotes the set of refinement levels and ν represents unavailable suffix positions. The nested construction specifies an ordering over accessible subrepresentations while leaving the semantics of the incremental information unconstrained.

We study three training conditions. *Standard full-code (Flat)* follows Equation (3) and imposes no explicit prefix-wise constraint during training. *Multi-prefix* follows a Matryoshka-style objective (Kusupati et al., 2022; Bussmann et al., 2025) that requires multiple refinements of each transition to support future prediction simultaneously. *Retained-prefix* instead samples one refinement for each training instance, following the stochastic exposure principle of nested dropout and variable-length latent-action training (Rippel et al., 2014; Yoshimoto et al., 2026).

Let

$$\ell_k^\nu(\tau_t) = \ell_{\mathrm{rec}}\left(FDM_\phi(o_t, P_k^\nu(z)), o_{t+\Delta}\right) \tag{5}$$

denote the reconstruction loss associated with refinement k. The three objectives can then be written as

$$\mathcal{L}_{\mathrm{flat}} = \mathbb{E}_{\tau_t}\left[\ell_K(\tau_t) + \mathcal{L}_{\mathrm{VQ}}\right], \tag{6}$$

$$\mathcal{L}_{\mathrm{multi}} = \mathbb{E}_{\tau_t}\left[\frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}} \ell_k^0(\tau_t) + \mathcal{L}_{\mathrm{VQ}}\right], \tag{7}$$

$$\mathcal{L}_{\mathrm{ret}} = \mathbb{E}_{\tau_t,\, k\sim p(k)}\left[\ell_k^\nu(\tau_t) + \mathcal{L}_{\mathrm{VQ}}\right]. \tag{8}$$

For Retained-prefix training, $p(k)$ is uniform over the selected refinement levels.

Multi-prefix and Retained-prefix represent dense and stochastic realizations of nested training, respectively. Multi constrains multiple refinements simultaneously for each transition, whereas Retained exposes them stochastically across optimization steps. With matched suffix handling and uniform $p(k)$, the Retained objective can be viewed as a stochastic estimator of the corresponding multi-prefix objective. Their comparison therefore tests whether the observed representational organization depends on a particular form of nested optimization, rather than defining an ordering of supervision strength. Implementation-specific differences in suffix parameterization and computation are reported in the experimental setup and corresponding ablations.

3.3 Cross-Distribution Transfer of Refinements

The nested construction in Section 3.2 makes different refinements of the same transition independently observable. To understand how refinement can affect cross-distribution transfer, we decompose the target risk of a source-trained readout into a target-intrinsic term and a source-to-target readout mismatch term.

Let $Z^{(k)}$ denote a frozen refinement of the same latent transition, A the robot action used only for evaluation, and $h_{S,k}$ a readout fitted on source distribution S. For target distribution T, define $m_{T,k}(Z^{(k)}) = \mathbb{E}_T[A \mid Z^{(k)}]$. Under squared loss, the target risk decomposes as

$$R_T(h_{S,k}) = \underbrace{\mathbb{E}_T\left[\|A - m_{T,k}(Z^{(k)})\|_2^2\right]}_{\mathcal{A}_T(k)\text{: target Bayes risk}} + \underbrace{\mathbb{E}_T\left[\|h_{S,k}(Z^{(k)}) - m_{T,k}(Z^{(k)})\|_2^2\right]}_{\mathcal{M}_{S\to T}(k)\text{: readout mismatch}}. \tag{9}$$

For nested refinements, $\mathcal{A}_T(k)$ is non-increasing because a target-optimal predictor can always ignore additional information. In contrast, $\mathcal{M}_{S\to T}(k)$ has no corresponding monotonicity: additional features may improve source-domain prediction while inducing different or less stable relationships on the target distribution. A richer refinement may therefore improve ID action readability without yielding an equivalent OOD gain. This observation motivates the refinement-wise ID–OOD analysis in Section 4; the derivation and an illustrative counterexample are provided in Appendix A.

4 Experiments

We study the refinement and generalization properties of action-free latent actions at both the representation and policy levels. Our main analysis is conducted on LIBERO, followed by a stronger LIBERO-Plus shift and an independent D0→D1/D2 evaluation on MimicGen. We then examine whether the observed representation-level differences translate to closed-loop control and inspect selected rollout trajectories.

4.1 Experimental Setup

Benchmarks and transfer settings. We conduct evaluation on LIBERO (Liu et al., 2023), LIBERO-Plus (Fei et al., 2025), and MimicGen (Mandlekar et al., 2023). Table 1 summarizes the corresponding source and target distributions. Following MoLA (Li et al., 2026), LIBERO-90 is used for the main LAM pretraining, while the four standard LIBERO suites provide cross-suite transfer evaluation. LIBERO-Plus introduces a stronger trajectory-level distribution shift. For MimicGen, we leverage its native hierarchy of reset distributions: D0 denotes the default distribution on which the source demonstrations are collected, while D1 and D2 introduce progressively broader initial-state variation (Mandlekar et al., 2023). We train all latent representations exclusively on Core-D0 and use D1 and D2 only as unseen transfer targets, with held-out D0 providing the corresponding in-distribution reference.

Representation probing. Following prior latent-action studies that use linear probing to assess action information in frozen representations (Nikulin et al., 2025; Lee et al., 2026), we evaluate

Table 1: Main benchmarks and transfer protocols. Target distributions are not used for LAM training or source-domain probe fitting.

Setting	LAM train	Source / ID	Transfer target
LIBERO	LIBERO-90	LIBERO-90	Spatial / Object / Goal / Long
LIBERO-Plus	LIBERO-90	LIBERO-90	Plus trajectories
MimicGen	Core-D0	held-out D0	D1 / D2

the learned latent representations with lightweight probes. For action probing, the probe is trained only on the source distribution and then transferred to the target distribution without refitting, allowing us to measure the cross-distribution accessibility of action-relevant information. We report the mean coefficient of determination (R^2) across robot-action dimensions. State-change and environment/suite probes serve as complementary diagnostics. Actions, states, and environment labels are used only for post-hoc probing and never enter the action-free LAM objective.

To compare the relative transfer behavior of different refinements, we additionally report the ID-to-OOD retention

$$\rho_k = \frac{R^2_{\mathrm{OOD}}(Z^{(k)})}{R^2_{\mathrm{ID}}(Z^{(k)}) + \epsilon}. \tag{10}$$

where ϵ is a small constant for numerical stability. We use ρ_k only as a normalized empirical diagnostic rather than an information-theoretic measure. When the corresponding ID R^2 is close to zero or negative, we report and interpret the raw ID/OOD R^2 values directly.

Closed-loop evaluation. We further evaluate the learned latent representations in downstream closed-loop control on LIBERO. All compared policies use the same architecture, robot-action data, and optimization protocol, with task success rate as the primary metric.

Implementation details. Flat, Multi-prefix, and Retained-prefix share the same base LAM architecture and maximum latent capacity. Unless otherwise specified, we use a frozen MAE-Large visual encoder (He et al., 2022), eight 32-dimensional quantized latent tokens with a 128-entry codebook, refinement levels $\mathcal{K} = \{1, 2, 4, 8\}$, and a five-frame transition interval. Complete optimization, probe, training-budget, and compute details are provided in Appendix B.

4.2 Representation Transfer under Distribution Shift

LIBERO. We first study how the information contained in the same latent transition changes as its representation is progressively refined. The complete latent consists of eight query tokens, following the discrete bottleneck structure used by MoLA (Li et al., 2026). We denote by P1, P2, P4, and P8 the representations containing the first 1, 2, 4, and 8 latent tokens, respectively. Token dimensionality, VQ configuration, and detailed optimization settings are provided in Appendix B.

We compare standard full-code training (Flat) with Multi-Prefix and Retained-Prefix, and additionally include the released MoLA IDMs (Li et al., 2026) as external representation references.

Table 2 first shows that explicit nested training makes short prefixes substantially more informative than post-hoc truncation of a full latent. More importantly, although both nested variants expose increasingly action-decodable information as the representation is refined, these gains are not preserved proportionally under distribution shift. For Retained-Prefix, Action ID increases from 18.29 at P1 to 36.98 at P8 and Action OOD from 18.01 to 32.89, whereas ρ_k decreases from 98.47% to 88.94%. Multi-Prefix exhibits the same tendency, with retention decreasing from 88.46% to 82.47%. Thus, higher absolute OOD recoverability does not imply that the additional information introduced by later refinements is equally stable across distributions.

The suite probe provides a complementary view of how the composition of the representation changes with refinement. For both nested variants, suite identity becomes increasingly decodable as more of the latent is exposed: probe accuracy rises from 29.50% to 45.70% for Multi-Prefix and from 29.44% to 42.04% for Retained-Prefix, while their transfer retention simultaneously decreases. This suggests that later refinements add increasingly distribution-specific information

Table 2: Frozen representation analysis on LIBERO. Recon. reports the original reconstruction error. Action ID and Action OOD measure action recoverability on held-out LIBERO-90 and cross-suite transfer, respectively. Action and state-transition scores are reported as $R^2 \times 100$. ρ_k denotes relative ID-to-OOD action retention. Suite Probe measures five-way suite predictability (chance: 20%), where lower values indicate less linearly decodable suite-specific information. Bold indicates the more favorable transfer-oriented value among our three controlled variants at the same prefix: higher for Action OOD and ρ_k, and lower for Suite Probe. MoLA IDMs are included only as external representation references.

Method	**Prefix**	**Recon.** ↓	**Action ID** $R^2\times100$	**Action OOD** $R^2\times100$	$\boldsymbol{\rho_k}$ (%)	**ΔState OOD** $R^2\times100$	**Suite Probe** ↓ (%)
MoLA Flow†	P8	–	3.46	0.07	–	0.45	54.34
MoLA Depth†	P8	–	13.56	-27.96	–	-65.87	72.30
MoLA Semantic†	P8	–	6.68	-17.34	–	-23.55	72.73
MoLA Concat-3†	3×P8	–	19.85	-26.35	–	-69.82	82.61
Flat	P1	.0940	11.83	8.98	75.88	8.17	**25.29**
Multi-Prefix	P1	.0257	17.59	15.56	88.46	14.48	29.50
Retained-Prefix	P1	.0261	18.29	**18.01**	**98.47**	15.85	29.44
Flat	P2	.0855	19.15	16.50	86.17	14.13	**28.44**
Multi-Prefix	P2	.0218	18.84	16.62	88.21	15.56	31.22
Retained-Prefix	P2	.0206	31.00	**28.30**	**91.28**	24.58	31.81
Flat	P4	.0671	26.38	22.49	85.25	20.21	**31.01**
Multi-Prefix	P4	.0198	24.12	20.96	86.89	19.85	35.60
Retained-Prefix	P4	.0198	34.87	**31.59**	**90.60**	28.22	36.17
Flat	P8	.0164	35.78	31.68	88.52	29.82	**35.95**
Multi-Prefix	P8	.0194	29.13	24.03	82.47	22.45	45.70
Retained-Prefix	P8	.0195	36.98	**32.89**	**88.94**	30.06	42.04

together with additional action-relevant content. Notably, however, FLAT exhibits lower suite predictability without consistently achieving higher OOD action recoverability, showing that suppressing suite information alone is insufficient to define a better latent action representation. We therefore treat the co-variation between suite decodability and retention as evidence about representation composition rather than a causal explanation of transfer degradation. Taken together, we refer to the systematic changes in ID readability, cross-distribution retention, and suite dependence across successive refinements as an *ordered transfer structure*. Here, "ordered" describes an empirical organization across prefixes; it does not assign predefined semantics to individual tokens or imply that an early prefix necessarily corresponds to a high-level skill. We next test whether this structure becomes more pronounced under the stronger LIBERO-Plus distribution shift.

Validation under stronger and cross-benchmark shifts. We further test the observed transfer structure under a stronger distribution shift and on an independent benchmark. On LIBERO-Plus, transfer becomes clearly non-monotonic with refinement (Table 7). For RETAINED-PREFIX, P2 achieves the highest action recoverability on all four suites, whereas the full P8 representation drops to negative R^2 on Spatial, Object, and Long (Table 8). For example, Object decreases from 21.48 at P2 to -82.86 at P8 ($R^2\times100$). We observe a similar pattern on MimicGen, where the latent action models are trained exclusively on Core-D0 and the same D0-fitted probe is transferred to D1 and D2 without refitting. As shown in Table 10, the strongest D2 transfer again occurs at an intermediate refinement: MULTI-PREFIX peaks at P2 and RETAINED-PREFIX at P4, rather than at the full P8 representation. At fixed P8, both nested variants outperform FLAT in 23 of 24 D0/D1/D2 task–distribution cells, including 13 of 14 target-only cells (Table 12).

Additional LIBERO-Plus and MimicGen task-level results are provided in Appendix C.

4.3 Closed-Loop Policy Evaluation

Frozen probes characterize the cross-distribution transfer properties of different latent prefixes, but it remains important to test whether these representation-level differences have consequences for downstream control. We therefore evaluate the learned representations in closed-loop policy learning on LIBERO. All conditions share the same policy architecture, robot-action data, training setup, and evaluation protocol, while only the latent interface is changed. For the nested models, we evaluate P1, P2, P4, and P8 to examine whether the transfer differences observed above are reflected in downstream behavior.

Closed-loop performance shows a similar non-monotonic dependence on prefix length. In the reported single-seed evaluation, Multi-Prefix attains its highest observed mean success at P4 (86.1%), whereas Retained-Prefix attains its highest at P2 (88.0%). At the matched P8 capacity, Multi-Prefix and Retained-Prefix achieve 83.5% and 86.3%, respectively, compared with 79.8% for Flat. These results provide a functional complement to the representation analysis: as additional latent information is exposed, neither its cross-distribution behavior nor its downstream consequences improve simply as a function of representation size.

4.4 Sparse Action Supervision as a Diagnostic

Prior work has used limited action labels and auxiliary action prediction to shape LAM representations (Nikulin et al., 2025; Lin et al., 2026). Our main LAMs use no robot-action labels. To assess the OOD action recoverability of our standard action-free Retained-Prefix model (0%), we compare it with variants that add a training-only action head on the full latent and label 0.25%–5% of transitions.

Table 3: Closed-loop success rate (%) on LIBERO using matched latent prefixes. P1/P2/P4/P8 denote the first 1/2/4/8 latent tokens. Each suite contains 10 tasks with 20 closed-loop rollouts per task. The reported results use one training seed and standard suites used in policy post-training.

Prefix	Method	Spatial	Object	Goal	Long	Avg.
P1	Flat	–	–	–	–	–
	Multi-Prefix	87.0	97.5	45.0	53.0	70.6
	Retained-Prefix	90.0	98.0	48.0	56.0	73.0
P2	Flat	–	–	–	–	–
	Multi-Prefix	88.5	98.5	76.0	77.0	85.0
	Retained-Prefix	96.5	98.0	84.0	73.5	88.0
P4	Flat	–	–	–	–	–
	Multi-Prefix	94.5	98.0	83.5	68.5	86.1
	Retained-Prefix	97.5	97.5	83.0	73.0	87.8
P8	Flat	86.5	94.5	81.5	56.5	79.8
	Multi-Prefix	97.5	97.5	83.0	56.0	83.5
	Retained-Prefix	98.0	96.5	80.0	70.5	86.3

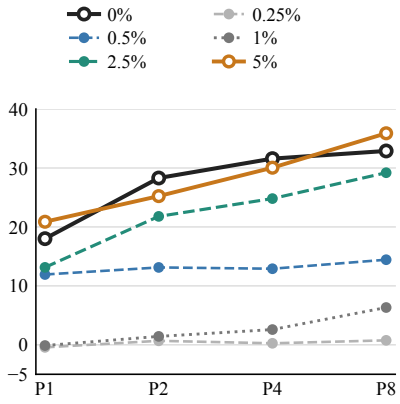


Figure 2: OOD action recoverability ($R^2 \times 100$) across retained prefixes under sparse action supervision. The 0% line is our action-free method. Exact scores in Appendix Table 9.

Figure 2 shows that our action-free method remains competitive under the OOD frozen-probe metric: its P2 and P4 scores (28.30 and 31.59) exceed those of the 5%-label variant (25.23 and 30.05), while the latter leads at P1 and P8. Thus, even without action labels during LAM training, intermediate retained prefixes can preserve action-relevant information that transfers to held-out suites.

4.5 From Latent-Action Transfer to Closed-Loop Behavior

Table 2 shows that Retained-P2 transfers more action-readable information to held-out LIBERO suites than same-length Flat and Multi-Prefix codes (OOD Action $R^2 \times 100$: 28.30 versus 16.50 and 16.62). Figure 3 shows how these latent interfaces behave on a LIBERO-10 target task. All four

policies activate the burner, but only Retained-P2 completes moka-pot placement in the displayed rollout, terminating at frame 258. Across 20 trials of this task, Retained-P2 succeeds 19 times, compared with 10, 11, and 14 for Flat, Multi-P2, and Multi-P8. This gives a behavioral counterpart to the cross-suite probe result: the compact action-free prefix that generalizes in action readout also supports the complete multi-stage task.



Figure 3: Eight-frame visualization of a selected joint-60k rollout on LIBERO-10 task 2 (turn on the stove and place a moka pot). Rows show Flat, Multi-P2, Multi-P8, and Retained-P2; columns share original video-frame indices. Retained-P2 terminates at frame 258, after which its final frame is repeated and marked END. Only Retained-P2 succeeds in the displayed trial. Task-level successes over 20 rollouts are 10, 11, 14, and 19 in row order.

5 Limitations and Conclusion

Our analysis uses controlled simulation benchmarks, one visual backbone, and one LAM training seed. Frozen probes reveal how prefix behavior changes under the tested distribution shifts, while closed-loop policies have only been evaluated on standard LIBERO suites. Testing whether these patterns persist on larger and more diverse robot-trajectory datasets and under policy-level distribution shifts will require replicated training and capacity-matched comparisons. The main Flat, Multi-Prefix, and Retained-Prefix LAMs use no action labels; those labels enter frozen-probe evaluation and downstream policy training.

We study action-free latent-action generalization from the perspective of how transition information is organized within the learned representation. Across progressive refinements of the same transition, we find that in-distribution action recoverability generally increases, while environment-specific information also becomes more decodable and OOD performance does not improve monotonically. Under stronger LIBERO-Plus shifts and in the MimicGen D2 evaluation, intermediate refinements can outperform the complete representation, with a similar non-monotonic pattern appearing in closed-loop control. These findings suggest that generalizable latent-action learning depends not only on how much transition information is captured, but also on how that information is organized during learning. This perspective motivates future latent-action objectives that favor transferable action-relevant structure rather than simply richer latent representations.

AI Use Statement

Generative AI tools were used to assist with experimental code implementation and language editing of the manuscript. The research questions, hypotheses, experimental design, analysis, and scientific conclusions were developed and verified by the authors. The authors take full responsibility for the final code, experimental results, figures, and manuscript.

REFERENCES

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Maria Elisabeth Bechtle, Feryal Behbahani, Stephanie C. Y. Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando De Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4603–4623, 2024. URL https://proceedings.mlr.press/v235/bruce24a.html.

Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning multi-level features with matryoshka sparse autoencoders. *arXiv preprint arXiv:2503.17547*, 2025. URL https://arxiv.org/abs/2503.17547.

Xiaoyu Chen, Hangxing Wei, Pushi Zhang, Chuheng Zhang, Kaixin Wang, Yanjiang Guo, Rushuai Yang, Yucen Wang, Xinquan Xiao, Li Zhao, Jianyu Chen, and Jiang Bian. villa-x: Enhancing latent action modeling in vision-language-action models. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2507.23682.

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.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Huang Huang, Sriram Yenamandra, Arjun Majumdar, Elie Aljalbout, Tushar Nagarajan, Tsung-Yen Yang, Akshara Rai, Michael Rabbat, Li Fei-Fei, Jiajun Wu, Tingfan Wu, and Franziska Meier. Cross-embodiment robot foundation world models with latent actions. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026. URL https://openreview.net/pdf/5c8fa1532215f2ea0717ed9db11ee805185dc7d3.pdf.

Youngjoon Jeong, Junha Chun, and Taesup Kim. Learning to act robustly with view-invariant latent actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6781–6790, 2026.

Hanjung Kim, Lerrel Pinto, and Seon Joo Kim. Hierarchical latent action model. *arXiv preprint arXiv:2603.05815*, 2026. URL https://arxiv.org/abs/2603.05815. ICLR 2026 World Models Workshop.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, 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 *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713, 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. Matryoshka representation learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Jung Min Lee, Dohyeok Lee, Seokhun Ju, Taehyun Cho, Jin Woo Koo, Li Zhao, Sangwoo Hong, and Jungwoo Lee. Mvp-lam: Learning action-centric latent action via cross-viewpoint reconstruction, 2026. URL https://arxiv.org/abs/2602.03668.

Yajie Li, Bozhou Zhang, Chun Gu, Zipei Ma, Jiahui Zhang, Jiankang Deng, Xiatian Zhu, and Li Zhang. From imagined futures to executable actions: Mixture of latent actions for robot manipulation. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2605.12167.

Yihan Lin, Haoyang Li, Yang Li, Haitao Shen, Yihan Zhao, Chao Shao, and Jing Zhang. From pixels to tokens: A systematic study of latent action supervision for vision-language-action models. *arXiv preprint arXiv:2605.04678*, 2026. URL https://arxiv.org/abs/2605.04678.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Hao Luo, Ye Wang, Wanpeng Zhang, Haoqi Yuan, Yicheng Feng, Haiweng Xu, Sipeng Zheng, and Zongqing Lu. Joint-aligned latent action: Towards scalable vla pretraining in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35047–35058, 2026.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. MimicGen: A data generation system for scalable robot learning using human demonstrations. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1820–1864, 2023. URL https://proceedings.mlr.press/v229/mandlekar23a.html.

Alexander Nikulin, Ilya Zisman, Denis Tarasov, Nikita Lyubaykin, Andrei Polubarov, Igor Kiselev, and Vladislav Kurenkov. Latent action learning requires supervision in the presence of distractors. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 46427–46447, 2025. URL https://proceedings.mlr.press/v267/nikulin25a.html.

Oren Rippel, Michael Gelbart, and Ryan Adams. Learning ordered representations with nested dropout. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1746–1754, 2014. URL https://proceedings.mlr.press/v32/rippel14.html.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent action pretraining from videos. In *International Conference on Learning Representations*, 2025.

Takanori Yoshimoto, Yang Hu, Naruya Kondo, and Tatsuya Matsushima. FlexLAM: Resolving the bottleneck trade-off in latent action learning. *arXiv preprint arXiv:2606.19408*, 2026. URL https://arxiv.org/abs/2606.19408.

Chuheng Zhang, Tim Pearce, Pushi Zhang, Kaixin Wang, Xiaoyu Chen, Wei Shen, Li Zhao, and Jiang Bian. What do latent action models actually learn? In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

A Proof and an Illustrative Transfer Counterexample

Proof of Equation (9). For brevity, write $Z = Z_k$, $m(Z) = \mathbb{E}_T[A \mid Z]$, and $h(Z) = h_{S,k}(Z)$. Expanding the squared norm gives

$$\begin{aligned}\mathbb{E}_T\|A - h(Z)\|^2 = \mathbb{E}_T\|A - m(Z)\|^2 + \mathbb{E}_T\|m(Z) - h(Z)\|^2 \\ + 2\mathbb{E}_T\langle A - m(Z), m(Z) - h(Z)\rangle. \end{aligned} \tag{11}$$

The cross term vanishes by conditioning on Z, since $\mathbb{E}_T[A - m(Z) \mid Z] = 0$. The identity holds for any fixed measurable source-trained readout with finite risk. Averaging it over random source training sets preserves the equality, but no such averaging is needed for the stated result.

Why target Bayes risk cannot increase with a longer prefix. For $k < k'$, $\sigma(Z_k) \subseteq \sigma(Z_{k'})$. The tower property and the same orthogonality argument yield

$$\mathcal{A}_T(k) = \mathcal{A}_T(k') + \mathbb{E}_T\|m_{T,k'}(Z_{k'}) - m_{T,k}(Z_k)\|^2 \geq \mathcal{A}_T(k'). \tag{12}$$

The result concerns an unrestricted target Bayes predictor. It does not imply monotonic transfer for finite-sample Ridge readouts, nor does it order separately trained encoders. The second term of Equation (9) includes function-class limitations, estimation effects, regularization, and source–target conditional mismatch; the current experiments do not identify their separate contributions.

A counterexample without token semantics. Let X and ϵ be independent standard normal variables, and let the scalar prediction target be $A = X + \epsilon$. Consider nested features $Z_1 = X$ and $Z_2 = (X, W)$. On the source distribution, set $W = A$; on the target, set $W = -A$. These are abstract statistical coordinates, not claims about the meanings of learned LAM tokens. Population source least squares gives $h_{S,1}(X) = X$ and $h_{S,2}(X, W) = W$.

On the target distribution, $R_T(h_{S,1}) = \mathbb{E}\epsilon^2 = 1$, while $R_T(h_{S,2}) = \mathbb{E}(2A)^2 = 8$. Yet the full target Bayes predictor is $m_{T,2}(X, W) = -W = A$, with zero error. Thus greater available information and worse transferred readout performance can coexist. This example establishes possibility only; it neither diagnoses our learned codes nor justifies assuming such a sign reversal in robot data.

A.1 Quantization, capacity, and sampled prefix training

For continuous encoder outputs $e_j \in \mathbb{R}^d$ and codebook $\mathcal{C} = \{c_1, \ldots, c_V\}$, quantization selects

$$q_j = \operatorname*{arg\,min}_{v \in \{1,\ldots,V\}} \|e_j - c_v\|_2^2, \qquad z_j = c_{q_j}. \tag{13}$$

The index prefix $Q_k = (q_1, \ldots, q_k)$ has the finite-capacity bound

$$I(\tau; Q_k \mid o_t) \leq H(Q_k \mid o_t) \leq k \log_2 V. \tag{14}$$

With $V = 128$, this is at most $7k$ bits for the indices. The bound is not an estimate of the learned entropy or the storage required for $32k$ embedding coordinates. It describes capacity, not semantic granularity.

For fixed parameters and the same suffix fill ν, a sampled reconstruction loss is an unbiased estimator of the weighted multi-prefix objective:

$$\mathbb{E}_{k\sim p}[\ell_k^\nu] = \sum_{k \in \mathcal{K}} p(k)\ell_k^\nu. \tag{15}$$

The direct reconstruction path through token j is retained with probability

$$\rho_j = \Pr(k \geq j) = \sum_{k \in \mathcal{K}:\, k \geq j} p(k). \tag{16}$$

Uniform sampling over $\{1, 2, 4, 8\}$ gives $\rho = (1, 3/4, 1/2, 1/2, 1/4, 1/4, 1/4, 1/4)$. Shared encoder parameters, cross-token attention, and quantization losses can still receive other gradients. Thus ρ_j describes a direct decoder path, not the total update frequency of a parameter. This asymmetry is a property of training, not evidence for a particular interpretation of the tokens.

Table 4: Architecture and training configuration for the action-free comparison.

Component	Setting
Visual input	224×224 RGB; ImageNet normalization
Frozen backbone	ViT-MAE-Large; 303,301,632 frozen parameters
MAE representation	197×1024 features; mask ratio 0
Transition encoder	MFormer; 8 queries; 4 layers; hidden 768; 12 heads
Quantized bottleneck	8 tokens; dimension 32; 128 codebook entries
Commitment coefficient	0.25
Feature decoder	12 layers; hidden 768; 12 heads
Trainable parameters	121,210,944 (Flat/Multi); 121,210,976 (Retained)
Transition interval	5 frames at recorded 20 FPS
Augmentation	Shared current/future replicate-pad random shift, pad 10
Optimizer	AdamW; betas $(0.9, 0.99)$; epsilon 10^{-8}
Learning rate	10^{-4} to 5×10^{-5}; 3,860 warm-up steps
Weight decay	10^{-4} for parameters with at least two dimensions
Gradient clipping / precision	Global norm 1; BF16
Global batch / seed	128 / 0
Training / evaluated checkpoint	20 epochs; 77,200 scheduled steps / 77,000

Prefix-grid hyperparameter. The reported runs use $K = 8$ quantized token slots and the prefix set $\mathcal{K} = \{1, 2, 4, 8\}$. Retained samples these four lengths uniformly per example. Multi evaluates all four lengths and averages the reconstruction losses. These values are experimental hyperparameters, not part of the definition of the nested-access formulation; a different grid can be studied under the same objective.

B Architecture, Optimization, and Data Accounting

The quantized latent tokens are projected back to width 768, flattened, and linearly pooled into a motion embedding. This representation is combined with the current-observation visual features and passed to the feature decoder to reconstruct the future observation in the frozen MAE representation space. The future image is encoded by the frozen MAE backbone to provide the reconstruction target and is not given to the decoder as a direct input. Evaluation and latent extraction disable random shifts.

Flat uses two GPUs with per-GPU batch 32 and gradient accumulation 2; Multi and Retained use per-GPU batch 64 with no accumulation. Archived wall-clock training durations are 25,243, 64,741, and 25,350 seconds for Flat, Multi, and Retained, respectively. These are descriptive timings on the recorded setup, not hardware-independent efficiency claims. The extra decoder passes make Multi materially more expensive despite a shared effective batch and schedule.

Table 5: Dataset accounting. Counts describe local exports, not assumed official benchmark coverage.

Quantity	Source export	Standard target probe	Plus export
Recorded tasks	73	4 named suites	40 base tasks
Episodes in export	3,921	Not aggregated here	14,347
Frames in export	569,249	Not aggregated here	2,238,036
Valid delta-5 pairs	549,644	Not aggregated here	2,166,301
Probe extraction	20,000	$4 \times 5,000$	20,000 initially
Final target probe size	–	20,000	$4 \times 3,500$
Action dimensions scored	7	7	6 (arm only)

Metric aggregation. For an r-dimensional target, the probe score is

$$R^2 = \frac{1}{r} \sum_{d=1}^{r} \left(1 - \frac{\sum_i (a_{i,d} - \widehat{a}_{i,d})^2}{\sum_i (a_{i,d} - \bar{a}_d)^2} \right). \tag{17}$$

Target-domain means are used only for scoring and are not involved in fitting the readout. For pooled evaluation, we first concatenate the balanced target subsets and then compute the coordinate-wise R^2; consequently, the pooled score is not generally equal to the arithmetic mean of the separately evaluated suite scores. Because suite membership alone does not guarantee strict task- or trajectory-level disjointness, we interpret the cross-suite results as transfer under distribution shift rather than as a strictly task-disjoint generalization benchmark.

C Additional Representation Results

Table 6: Full-code Action-7 R^2 by target suite, step 77k. These are individually scored subsets, not components to average into the pooled score.

Training	Spatial	Object	Goal	Long
Flat	.26596	.31870	.26356	.30137
Multi	.19076	.24539	.18350	.24040
Retained	.26868	.33727	.27401	.31758

Table 7: Complete Plus-export probe results. Both ID and target action columns use Arm-6. The target-state column predicts $s_{t+5} - s_t$.

Training	Prefix	Arm-6 ID	Arm-6 Plus	ΔState Plus
Flat	1	.11175	.06625	.05871
Flat	2	.19599	.12714	.10072
Flat	4	.27801	.18532	.14827
Flat	8	.37765	.22245	.18575
Multi	1	.16933	.12265	.10856
Multi	2	.18370	.12403	.11486
Multi	4	.24418	.13877	.12252
Multi	8	.29989	.03269	.03280
Retained	1	.17904	.13082	.11427
Retained	2	.32561	.23419	.18782
Retained	4	.36691	.21467	.17214
Retained	8	.38861	-.03243	-.15942

Table 8: Retained Arm-6 R^2 by base suite within the Plus export. Perturbation labels are unavailable.

Prefix	Spatial	Object	Goal	Long
1	.05361	.16618	.05816	.15033
2	.19729	.21476	.15616	.25096
4	.18808	.15076	.14989	.21182
8	-.08948	-.82861	.05378	-.03766

C.1 Sparse action supervision

Table 9: Exact OOD action recoverability values ($R^2 \times 100$) underlying Figure 2. The 0% row is our standard action-free Retained-Prefix model; bold marks the highest score at each prefix.

Action labels	**P1**	**P2**	**P4**	**P8**
0% (ours)	18.01	**28.30**	**31.59**	32.89
0.25%	-0.43	0.68	0.26	0.76
0.5%	11.93	13.13	12.91	14.44
1%	-0.12	1.43	2.59	6.33
2.5%	13.14	21.79	24.81	29.20
5%	**20.88**	25.23	30.05	**35.89**

Action-supervision comparison protocol. Our standard 0% model uses global batch 128 and 77,000 updates; the 0.25%–5% labeled models use global batch 1,024 and 9,640 updates. Example exposures are similar (9.856M versus 9.871M), but the optimization schedules differ. The masked action auxiliary loss is averaged over the full batch, so label coverage also changes its effective gradient weight. Isolating the effect of label availability or label count requires a matched 0% run and a label-normalized auxiliary-loss control.

Realized action-label coverage. The nominal 0.25%, 0.5%, 1%, 2.5%, and 5% masks select 1,250, 2,489, 4,957, 12,378, and 24,735 training transitions, respectively. Their realized coverages are 0.25296%, 0.50370%, 1.00315%, 2.50495%, and 5.00565%. Label subsets are deterministically nested across rates. This facilitates within-recipe comparisons, but does not remove the need for independent seeds or the batch and auxiliary-normalization controls described above.

C.2 MimicGen cross-distribution action readout

The MimicGen models train on D0 only. For each model and prefix, the D0-fitted frozen action readout is transferred to D1 and D2. D0 evaluates held-out episodes of ten source tasks; D1 and D2 evaluate ten and four target tasks, respectively, without readout refitting. We calculate R^2 within each task, then take an unweighted mean over tasks. Thus the values below are task-macro scores rather than pooled R^2, and the D1 and D2 columns refer to different task sets.

Table 10: MimicGen action-readout $R^2 \times 100$ (task-macro mean; seed 0). D0 is held-out episode evaluation on ten source tasks; D1 contains ten target tasks and D2 contains four target tasks. Task sets differ between D1 and D2. Bold marks the highest prefix within each nested training objective and distribution; it does not indicate statistical significance. The source CSV contains per-task scores but not probe-fit or checkpoint manifests.

Training	Prefix	D0 (10 tasks)	D1 (10 tasks)	D2 (4 tasks)
Flat	P1	0.14	-1.91	-2.42
Flat	P2	0.10	-1.95	-2.45
Flat	P4	0.07	-2.05	-2.47
Flat	P8	-0.06	-2.20	-2.58
Multi	P1	4.03	4.25	5.03
Multi	P2	8.35	8.04	**7.09**
Multi	P4	10.70	9.51	6.51
Multi	P8	**17.04**	**12.07**	3.92
Retained	P1	8.25	6.80	3.02
Retained	P2	12.87	10.48	5.04
Retained	P4	16.20	13.42	**9.02**
Retained	P8	**20.37**	**14.91**	6.32

Table 11: D2 task-level action-readout $R^2 \times 100$ for the nested models (seed 0). The task-macro intermediate-prefix advantage is not universal across tasks; each task has 5,000 target samples in the supplied CSV. The row labeled three-piece assembly corresponds to the CSV task key `three_piece_assembly`.

	Multi				Retained			
Task	P1	P2	P4	P8	P1	P2	P4	P8
coffee	5.94	5.53	2.65	-7.11	0.97	1.91	6.24	-10.12
square	4.71	8.65	9.70	10.11	5.69	9.84	13.64	17.86
threading	1.95	3.00	0.12	-5.21	-3.99	-5.08	-0.50	-5.16
three-piece assembly	7.54	11.19	13.56	17.88	9.40	13.51	16.73	22.68

Table 12: Task-level P8 action-readout $R^2 \times 100$ from the single-seed MimicGen CSV. Each nested condition exceeds Flat on 23 of 24 source-plus-target cells, but on 13 of 14 target-only D1/D2 cells. The only exception for both is D2 coffee. D0 uses held-out episodes; D1 and D2 use the CSV's frozen-OOD evaluation label. This comparison is descriptive, not a significance test.

Distribution	Task	Flat	Multi	Retained
D0	coffee	2.60	15.09	18.82
D0	coffee_preparation	-0.80	18.28	18.54
D0	hammer_cleanup	2.70	21.10	11.69
D0	kitchen	-2.16	16.37	18.28
D0	mug_cleanup	2.39	22.91	26.21
D0	square	1.12	9.09	16.13
D0	stack	-1.13	24.75	34.11
D0	stack_three	1.41	18.32	23.88
D0	threading	-6.14	11.33	21.56
D0	three_piece_assembly	-0.56	13.12	14.45
D1	coffee	-2.19	5.34	1.05
D1	coffee_preparation	-1.51	15.86	19.81
D1	hammer_cleanup	-2.21	15.58	16.72
D1	kitchen	-1.75	13.44	20.69
D1	mug_cleanup	-0.49	16.80	21.40
D1	square	-0.89	6.13	6.46
D1	stack	-1.86	18.98	27.39
D1	stack_three	-0.30	18.08	25.99
D1	threading	-7.68	-2.30	-3.89
D1	three_piece_assembly	-3.10	12.82	13.53
D2	coffee	-1.41	-7.11	-10.12
D2	square	-1.60	10.11	17.86
D2	threading	-6.26	-5.21	-5.16
D2	three_piece_assembly	-1.05	17.88	22.68