TOPACT: TOPOLOGY-AWARE ACTION TOKENS FOR AUTOREGRESSIVE ROBOT POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

An action tokenizer determines both the information retained about a robot trajectory and the categorical prediction problem presented to its policy. We introduce TOPACT, an ordered codec that represents a 16-step action chunk with four finite-scalar-quantized tokens. Beyond reconstruction, its training objective aligns neighborhoods in action space with neighborhoods in latent space, before and after quantization. We evaluate matched topology-aware and reconstruction-only (RECON) codecs through action-only codec training, compact autoregressive policies, and an OpenVLA integration, holding the codec frozen while training each policy. TopAct preserves substantially stronger neighborhood structure after quantization and produces policy targets that are consistently easier to predict: token accuracy is higher for the compact policy on every suite, and under OpenVLA the TopAct arm converges faster and to a stronger token model. The same targets also support better closed-loop behavior: the compact policy reaches higher success on three of four suites, and the selected OpenVLA policy attains a higher success point estimate. Together, the findings motivate topology-aware action representations and the joint evaluation of codec quality, policy learning, and closed-loop behavior.

1 INTRODUCTION

Autoregressive robot policies translate observations and task instructions into discrete action symbols. This interface supports next-token training and integration with pretrained vision-language models (Kim et al., 2024; Pertsch et al., 2025). When actions are predicted in chunks, the tokenizer must compress a continuous trajectory while retaining the control information needed for manipulation. Its design also determines which trajectories share symbols and how difficult those symbols are for a policy to predict.

Reconstruction distortion measures what a codec loses about an individual trajectory, not how different trajectories relate after quantization: two similar action chunks can reconstruct accurately while occupying distant register coordinates, and a representation that makes their neighborhoods agree may allocate fewer distinctions to task-specific details. We therefore study how representation geometry, reconstruction, and downstream control behavior relate.

Existing methods impose complementary structure. FAST compresses temporal redundancy in frequency space (Pertsch et al., 2025). Ordered Action Tokenization (OAT) combines learned registers, finite scalar quantization, causal register attention, and nested dropout to produce progressively refined action chunks (Liu et al., 2026). Register order structures information within a token sequence. We study the additional structure between sequences: whether neighborhoods of different action chunks remain recognizable in the register coordinates after rounding.

TOPACT compresses a 16-action chunk into four ordered tokens from a 1,125-value vocabulary (Figure 1). Each prefix $K \in \{1, 2, 3, 4\}$ reconstructs the same complete 16-action chunk. Its objective aligns neighborhood distributions between actions and both continuous and quantized registers, averaged over all four prefixes.

We evaluate this design through three stages under a single 16-action/four-token contract. Stage 1 trains matched codecs with and without the geometry objective. Stage 2a freezes the final codecs and trains compact autoregressive policies on four LIBERO suites, probing whether the representation already functions in a very light model. Stage 2b post-trains OpenVLA with the same frozen codecs,

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

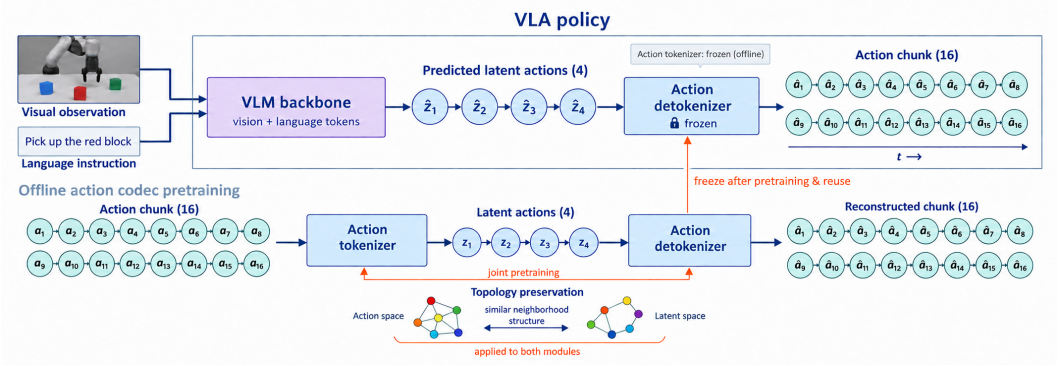

Figure 1: **Overview of the proposed VLA framework.** An action tokenizer and action detokenizer are jointly pretrained to compress a temporally ordered 16-token action chunk into four latent action tokens and reconstruct the original sequence. Topology preservation encourages the latent space to retain a similar, though not necessarily identical, neighborhood structure to the original action space. After pretraining, both modules are frozen. The VLM predicts four latent action tokens from visual observations and language instructions, which the frozen action detokenizer expands into a 16-token action chunk for execution.

predicting the four tokens through a causal head. The Stage 1 codec of each arm is fixed at its final EMA checkpoint rather than chosen after observing policy results, so the downstream comparisons reflect the learned representation rather than checkpoint selection.

Across these stages, the topology-aware codec is consistently easier for a policy to predict: it yields higher autoregressive token accuracy for the compact policy on every suite, and with the OpenVLA backbone it trains faster and to a stronger token model. This advantage carries to closed-loop control: the compact TopAct policy reaches higher success on three of four suites, and under OpenVLA the selected TopAct policy reaches a higher success point estimate (40.8% versus 38.0%), though this single-seed gap is not statistically separated from zero. These benefits come with a codec reconstruction cost. We make three contributions:

- **A topology-preserving action tokenizer/detokenizer.** TopAct translates neighborhood information from action space into latent register space, aligning soft neighborhood distributions before and after quantization at every token prefix. Our experiments show empirically that this topology-preserving codec is more suitable for autoregressive training and inference: its tokens are markedly easier for policies to predict, and the advantage carries to faster training and stronger closed-loop behavior across compact and pretrained VLA backbones.
- **Compact ordered chunk tokenization.** The codec compresses a 16-step action chunk into four ordered latent tokens, so a policy makes four categorical decisions per chunk instead of serializing every action coordinate. The causal register order matches the left-to-right factorization of autoregressive models, shortening the prediction sequence and easing inference.
- **An embodiment-agnostic codec trained once.** The codec is trained once and shared across embodiment setups: learned virtual dimensions let it tokenize and detokenize action chunks from any embodiment whose action dimensionality is at most a configurable width $D_{\max}$, without treating missing coordinates as observed actions.

2 Related Work

Action representations for robot policies. OpenVLA adapts a robot-pretrained vision-language backbone to action prediction (Kim et al., 2024), following a broader line of vision-language-action models such as RT-1 (Brohan et al., 2022), RT-2 (Brohan et al., 2023), and Octo (Team et al., 2023); its native interface serializes scalar bins for individual action dimensions. Diffusion Policy provides a continuous generative alternative for action chunks (Chi et al., 2023), while ACT learns temporally extended action sequences with a conditional variational model (Zhao et al., 2023). These interfaces

either predict continuous trajectories or serialize individual coordinates; TOPACT instead learns a compact, invertible chunk representation that a policy predicts as joint trajectory registers through a separate causal head. Vector-quantized latent models provide a general basis for learned discrete representations (Oord et al., 2017); VQ-BeT applies latent actions to behavior generation (Lee et al., 2024), and PRISE combines quantization with sequence compression to learn temporal action abstractions (Zheng et al., 2024). These approaches motivate treating the action representation as a learned control interface rather than a fixed serialization of coordinates.

Topology and geometry of action representations. Preserving the topology of a data space in its latent embedding is a long-standing goal of representation learning; topological autoencoders, for example, add a topological regularization term so that neighborhoods survive the encoder (Moor et al., 2020). In robot learning, LipVQ-VAE studies how the structure of an action tokenizer affects in-context imitation (Vuong et al., 2025), but neighborhood preservation across quantization has not been an explicit training target for autoregressive action tokenizers. TOPACT addresses this gap by matching soft neighborhood distributions between action space and register space, before and after FSQ rounding, at every token prefix, and by connecting the preserved geometry to autoregressive policy learning and closed-loop control.

Compression and order. FAST uses a discrete cosine transform, coefficient discretization, and byte-pair encoding to compress temporally correlated actions (Pertsch et al., 2025), and FASTer studies neural action tokenization and efficient autoregressive action modeling (Liu et al., 2025). OAT identifies compression, total decodability, and left-to-right causal ordering as desirable properties of an autoregressive action tokenizer (Liu et al., 2026). TOPACT adopts OAT's ordered-register design principle and the fixed scalar quantization of Mentzer et al. (2023), compressing a 16-step chunk into four ordered tokens so that an autoregressive policy needs only four categorical decisions per chunk; its added contribution is the explicit matching of cross-trajectory neighborhoods on both sides of rounding. Hierarchical spatiotemporal tokenizers introduce multi-scale action abstractions for imitation learning (Fateh et al., 2026), and OmniSAT targets compact action sequences for faster autoregressive inference (Lyu et al., 2025). This study does not include locally matched implementations of OAT, FAST, or FASTer, so the empirical comparison isolates the added geometry objective within our codec implementation rather than establishing a ranking of tokenizers.

Embodiment-agnostic action interfaces. Generalist policies aggregate demonstrations across robots with heterogeneous action spaces (Open X-Embodiment Collaboration, 2023) and typically pair a shared backbone with per-embodiment action heads (Team et al., 2023). Learned chunk codecs instead map heterogeneous action dimensions into a shared token vocabulary: NAC learns a neural codec as the action interface of VLA models (Jawaid & Xiang, 2026), and ActionCodec provides a systematic study of tokenizer design choices and evaluation criteria (Dong et al., 2026). TOPACT extends this line with a fixed-width interface and learned virtual dimensions, so a single codec, trained once, tokenizes and detokenizes action chunks from every embodiment whose action dimensionality is at most the configurable width $D_{\max}$.

3 TOPOLOGY-AWARE ACTION TOKENIZATION

3.1 ORDERED TRAJECTORY CODEC

For dataset s, a chunk $a \in \mathbb{R}^{T \times D_s}$ contains $T = 16$ future actions. Per-dataset, per-dimension min–max statistics fitted on training episodes map actions to normalized coordinates

$$x = \mathcal{N}_s(a). \tag{1}$$

Constant training dimensions map to zero; validation uses the same statistics without clipping or refitting. To support embodiments with different action dimensions, we embed each normalized chunk into a fixed-width interface of $D_{\max}$ action dimensions:

$$\tilde{x}_s = \text{concat}\,[x_s,\, v_s] \in \mathbb{R}^{T \times D_{\max}}, \qquad v_s \in \mathbb{R}^{T \times (D_{\max} - D_s)}, \tag{2}$$

where v_s is a learned virtual-dimension vector (broadcast over time and absent when $D_s = D_{\max}$). We set $D_{\max} = 16$ in all experiments; this width is a configuration choice rather than a limitation of the codec. The codec can therefore share its tokenizer and detokenizer across embodiments without

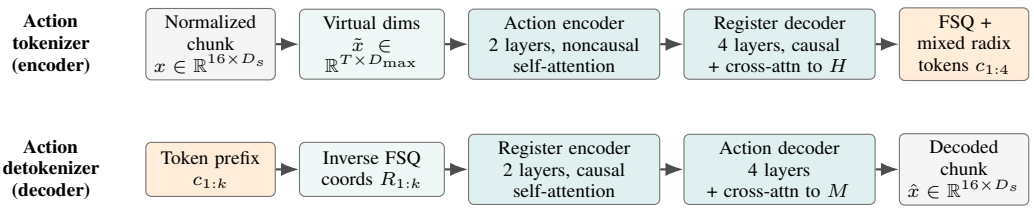

Figure 2: **Tokenizer and detokenizer Transformer architecture.** The upper path encodes a normalized action chunk, appends learned virtual dimensions when $D_s < D_{\max}$, applies action self-attention, and uses register queries that cross-attend to the action sequence before FSQ. The lower path inverts a visible token prefix, applies causal self-attention over registers, and uses action queries that cross-attend to the registers to reconstruct the genuine action dimensions. Virtual dimensions are auxiliary coordinates and are discarded after decoding.

treating missing coordinates as observed actions. The decoder output is restricted to the genuine D_s coordinates for reconstruction, geometry, normalization inversion, and execution; virtual coordinates are discarded. Batches contain only one dataset, so distances never compare incompatible coordinate systems across embodiments. Figure 2 summarizes the tokenizer and detokenizer data flow, including the register attention, finite scalar quantization, prefix masking, and action reconstruction paths.

Tokenizer. The tokenizer maps the padded normalized chunk to an ordered latent sequence. Every Transformer sublayer in the codec is a pre-LayerNorm residual update: it applies its operation to the layer-normalized input and adds the result back. Three operations are used, with compositions shown in Figure 2: self-attention, optionally masked (lower-triangular for causal blocks); cross-attention to a memory sequence from a preceding stage, restricted to the visible register prefix in the detokenizer; and a position-wise multilayer perceptron. Encoder layers chain self-attention with the perceptron, and decoder layers chain self-attention, cross-attention, and the perceptron in that order. A linear embedding and sinusoidal positions produce the action sequence H. Two encoder layers refine H over the T action steps. Four learned register queries $Q^{(0)} \in \mathbb{R}^{K\times E}$ then pass through four decoder layers, cross-attending to H under causal register self-attention. The latent head gives $Z = f_\theta(\tilde{x}_s) \in \mathbb{R}^{K\times L}$ with $K = 4$ and $L = 4$. All codec blocks use width $E = 256$ with four attention heads; causal masking prevents register k from depending on later registers.

Finite scalar quantization (FSQ). For level counts $\ell = (9, 5, 5, 5)$, let $h = (\ell - 1)/2 = (4, 2, 2, 2)$; products and divisions act coordinatewise. Each register row is bounded, rounded to integer coordinates, and renormalized, and the post-quantization objective uses the straight-through estimate:

$$U = \tanh Z, \quad R = \operatorname{round}(U \odot h), \quad Y = R \oslash h, \quad \bar{Y} = U + \operatorname{sg}(Y - U), \tag{3}$$

where $\bar{Y}$ retains the forward value Y and an identity gradient with respect to U. A mixed-radix bijection gives one token per register:

$$c_k = \sum_{l=1}^{L} (R_{k,l} + h_l)\rho_l, \qquad \rho_l = \prod_{m=l+1}^{L} \ell_m, \qquad c_k \in \{0, \ldots, 1124\}. \tag{4}$$

Every valid token sequence has a well-defined coordinate inverse and neural decoder output. This is total decodability, not a guarantee that arbitrary sequences produce useful or safe actions. Distances between numeric token IDs are not used as a geometric metric.

Detokenizer. The detokenizer first inverts the mixed-radix map of Equation 4 for each token in the visible prefix, recovering the signed integer coordinates $R_{1:k}$, and projects them into the decoder width. Two causal register-encoder layers produce the register memory M. Four learned action queries $A^{(0)} \in \mathbb{R}^{T\times E}$ then pass through four action-decoder layers, cross-attending to M. The action head returns only the genuine coordinates: for $k \in \mathcal{K} = \{1, 2, 3, 4\}$, $\hat{x}^{(k)} = g_\phi(R_{1:k}; D_s)$, and each prefix reconstructs the same complete chunk. During training, the prefix k is sampled per example with probabilities $(0.15, 0.20, 0.25, 0.40)$. The nominal full-code rate is $4\log_2(1125) \simeq 40.5$ bits. For a 16-by-7 LIBERO chunk, four register predictions replace 112 scalar tokens under a one-token-per-coordinate serialization, a 28-fold reduction in token count. This arithmetic does not measure wall-clock acceleration, which also depends on the backbone and neural decoder.

3.2 CODEC TRAINING OBJECTIVES

Reconstruction loss. For a sampled prefix k_i, the reconstruction objective compares the genuine normalized action coordinates with the detokenizer output:

$$\mathcal{L}_{\text{rec}} = \frac{1}{B}\sum_{i=1}^{B} \frac{\|x_i - \hat{x}_i^{(k_i)}\|_F^2}{TD_s}. \tag{5}$$

This term preserves the information needed to reproduce each individual action chunk. It does not constrain how distances between different chunks appear in the latent registers, which motivates the topology term below.

Topology loss. For samples $i \neq j$ in the same dataset batch and a prefix $k \in \mathcal{K}$, we compare action-space and register-space distances:

$$d_{ij}^{a} = \frac{\|x_i - x_j\|_F^2}{TD_s}, \qquad d_{ij}^{\Xi,k} = \frac{\|\Xi_{i,1:k} - \Xi_{j,1:k}\|_F^2}{kL}, \qquad \Xi \in \{U, \bar{Y}\}. \tag{6}$$

The distance $d_{ij}^{\Xi,k}$ is evaluated separately for every $k = 1, 2, 3, 4$ in the Stage 1 training objective, both before FSQ ($\Xi = U$) and after FSQ ($\Xi = \bar{Y}$). The primary reported geometry endpoint uses $k = 4$ only; it does not replace the prefix-wise training terms. We use normalized quantized coordinates so the nine-level coordinate cannot dominate solely because of its integer range. For each anchor i, distances become a soft neighborhood distribution:

$$p_{ij}^{a} = \frac{\exp[-d_{ij}^{a}/(\eta\tau_a)]}{\sum_{n\neq i}\exp[-d_{in}^{a}/(\eta\tau_a)]}, \qquad p_{ij}^{\Xi,k} = \frac{\exp[-d_{ij}^{\Xi,k}/(\eta\tau_\Xi)]}{\sum_{n\neq i}\exp[-d_{in}^{\Xi,k}/(\eta\tau_\Xi)]}. \tag{7}$$

The temperatures τ adjust for distance scale and η controls sharpness. The intuition is to preserve relative neighborhood identity: action chunks that are close should induce similar probabilities of selecting one another, while distant chunks should receive lower probability. We formulate this as a soft distribution rather than a hard rank constraint because rounding is itself a discrete, non-differentiable operation, and a distributional comparison degrades gracefully when quantization collapses nearby chunks onto shared coordinates. A symmetric KL divergence penalizes both directions of disagreement:

$$\mathcal{L}_{\text{nbr}}(\Xi) = \frac{1}{|\mathcal{K}|B}\sum_{k\in\mathcal{K}}\sum_{i}\left[D_{\text{KL}}(p_i^{a}\|p_i^{\Xi,k}) + D_{\text{KL}}(p_i^{\Xi,k}\|p_i^{a})\right]. \tag{8}$$

We average this loss equally over all four prefixes because every prefix is an intended detokenization budget; supervising only the full prefix would leave shorter prefixes unconstrained. The pre-FSQ term shapes the continuous register geometry, and the post-FSQ term specifically discourages rounding from changing that geometry. The full codec objective is

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda_{\text{pre}}\mathcal{L}_{\text{nbr}}(U) + \lambda_{\text{post}}\mathcal{L}_{\text{nbr}}(\bar{Y}). \tag{9}$$

Both weights warm linearly from zero to their target value, and the temperatures adapt to the observed distance scales (values and schedules in Appendix A). The RECON arm sets both topology weights to zero and otherwise retains the training recipe.

Applying the objective after quantization explicitly penalizes neighborhood changes that survive into the discrete representation. We use *topology-aware* for this finite-sample neighborhood regularization — the term is not a claim of topological equivalence. A symmetric KL objective on a sampled batch does not prove global rank preservation, continuity, or preservation of task-relevant distinctions, and the straight-through gradient does not exactly differentiate rounding; these are empirical representation constraints, evaluated independently of reconstruction.

3.3 FROZEN-CODEC AUTOREGRESSIVE CONTROL

After codec training, the tokenizer and detokenizer are frozen, and the downstream policy treats action prediction as categorical next-token prediction over the codec's vocabulary. With the codec frozen, action labels are generated in FP32 outside policy autocast.

Token prediction objective. The policy receives an observation o (camera images, robot state, and a task or language instruction, depending on the policy class), written $p_\psi(\cdot \mid \cdot)$ with trainable parameters ψ, and predicts the ordered token sequence $c = (c_1, \ldots, c_4)$ of Equation 4 autoregressively, one factor per register in codec order:

$$p_\psi(c \mid o) = \prod_{k=1}^{K} p_\psi(c_k \mid c_{<k}, o). \tag{10}$$

Each factor is a 1,125-way classification conditioned on o and the previously decided tokens $c_{<k} = (c_1, \ldots, c_{k-1})$; the causal register order learned by the codec is what makes this left-to-right factorization available to the policy. Training maximizes likelihood on N demonstration pairs (o_i, c_i):

$$\mathcal{L}_{\text{policy}} = -\frac{1}{NK} \sum_{i=1}^{N} \sum_{k=1}^{K} \log p_\psi(c_{i,k} \mid c_{i,<k}, o_i), \tag{11}$$

with each factor conditioned on the preceding reference tokens of the demonstration label c_i, so all four factors are supervised in parallel on every example. During rollout, the policy generates greedily token by token, each prediction conditioned on the tokens it has itself generated, and the completed sequence is inverted through the token map and detokenized.

Stage 2a: compact policy. Separate ResNet-18 encoders (He et al., 2015) encode external and wrist images at two observation timesteps. Robot state and a learned task embedding condition a four-layer, width-256, four-head causal Transformer. The codec remains frozen and the token loss is unweighted. This deliberately light design isolates the representation from backbone scale: if topology preservation is effective, its tokens should already be predictable and executable without a large pretrained model, supporting the VLM integration of Stage 2b.

Stage 2b: OpenVLA integration. One OpenVLA-7B contextual pass processes external and wrist views, robot state, language, and four learned action queries. A four-layer causal head predicts the registers using a tied 1,125-way embedding/output table. The vision towers, base language model, and codec remain frozen; rank-16 LoRA (Hu et al., 2021) adapts the language-model attention projections and the multimodal projector, while the state projector, action queries, and token head are trainable. Full architecture, adaptation, and optimization settings appear in Appendix A. As in Stage 2a, all four registers have equal weight in the token objective.

Both policies complete the factorization of Equation 10, invert the token map, decode 16 normalized actions, and apply the stored inverse normalizer. They execute these 16 actions and then replan. Thus four autoregressive decisions describe the complete executable chunk, and new feedback is incorporated every 16 executed actions.

4 Experiments

4.1 Experimental Setup

We use RECON as shorthand for the reconstruction-only baseline throughout. All experiments use the same 16-action/four-token contract: we freeze each Stage 1 codec and train downstream policies on top of it, so every stage compares the two representations under a matched recipe. Stage 1 (Section 4.2) trains the codec pair; Stage 2a trains compact autoregressive policies; Stage 2b trains OpenVLA policies with proxy-based checkpoint selection, which is our primary closed-loop result.

Stage 1 data and metrics. Stage 1 uses 18 canonical action datasets: 14 generalist sources and four LIBERO suites (Open X-Embodiment Collaboration, 2023; Liu et al., 2023), materialized as 561,922 training and 64,566 validation action chunks. Episode-level splits precede chunking, and normalization statistics use training data only. The matched codec arms share architecture, initial weights, data order, seed, global batch 128, and the 800-epoch budget; the final EMA checkpoint is the declared primary artifact for each arm. All codec and policy comparisons use a single training seed, which we treat as a limitation (Section 5). Codec reconstruction is dataset-macro normalized MSE on the full validation set, and geometry is the dataset-macro Spearman correlation between pairwise action distances and register distances.

Stage 2 policies and evaluation. Stage 2a trains one policy per suite and codec arm on LIBERO-Spatial, Goal, Object, and LIBERO-10, selecting each arm's checkpoint by its development-proxy success under a rule applied identically to both arms. Stage 2b trains matched OpenVLA policies on LIBERO-10 (450 training and 50 validation demonstrations) for 10,000 optimizer steps, scoring each arm every 300 steps on a fixed 50-episode development proxy and retaining its best-scoring checkpoint. This selection rule is applied identically and independently to both arms. Closed-loop evaluation uses 50 official initial states for each of ten tasks per suite, giving 500 paired trials per arm and suite with matched task and state identities across codec arms. Policies use greedy four-token generation and execute the decoded 16-action chunk before replanning. The proxy shares initial states with the official evaluation and serves as a development-selection signal, not an independent benchmark. Full training, rollout, and statistical settings appear in Appendix A. These controls support the internal comparisons; differences in preprocessing and execution prevent treating published native-VLA scores as locally matched baselines.

4.2 Stage 1: Codec Training and Representation Results

Both arms complete the full 800-epoch budget under the matched recipe of Appendix A. The primary comparison uses the fixed-final EMA pair, so the full-budget result does not depend on retrospectively choosing a favorable validation epoch.

Table 1: **Stage 1 training outcome at the fixed endpoint.** Both models map 16 actions to four tokens, share a single training seed, and complete the same training budget. Reconstruction and geometry are independently recomputed final validation metrics.

Quantity	Recon	TopAct
Completed epochs	800	800
Optimizer updates	3,506,400	3,506,400
Normalized MSE, $K = 4 \downarrow$	**0.006956**	0.008867
Pre-FSQ Spearman, $K = 4 \uparrow$	0.6664	**0.9443**
Post-FSQ Spearman, $K = 4 \uparrow$	0.6569	**0.9245**
Aggregate observed vocabulary	100%	100%
Effective code count	940.2	815.8

At the four-token endpoint, TopAct increases post-FSQ correlation from 0.6569 to 0.9245, an absolute gain of 0.2676 (Table 1). Normalized MSE increases from 0.006956 to 0.008867, a 27.5% relative cost. The pre-to-post correlation drop is 0.0198 for TopAct versus 0.0095 for Recon. Thus TopAct has much stronger absolute quantized geometry, but does not have a smaller quantization-induced correlation gap in this experiment.

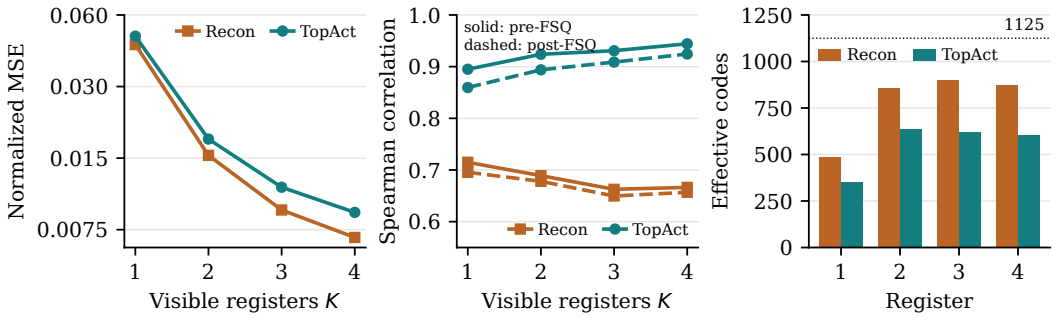

Figure 3: **Final 16-action codec: prefix metrics and per-register code usage.** Left: normalized reconstruction MSE at each visible prefix. Middle: action-space versus register-space Spearman correlation before (solid) and after (dashed) FSQ quantization; every point is an evaluation of the same fixed-final checkpoint, not a different training epoch. Right: effective codes per register (exponential of the register's code entropy); the dotted line marks the 1,125-entry vocabulary. TopAct concentrates usage on fewer codes in every register.

Figure 3 (left and middle) traces both metrics across prefixes. TopAct reconstruction MSE decreases from 0.048856 at one token to 0.018036, 0.011313, and 0.008867 at two, three, and four tokens, with the control 8–22% lower at every prefix. TopAct's post-FSQ correlations increase from 0.8596 to 0.8939, 0.9088, and 0.9245; its pre-FSQ correlations from 0.8950 to 0.9239, 0.9308, and 0.9443. The corresponding control post-FSQ correlations are 0.6956, 0.6781, 0.6497, and 0.6569. Prefix-wise supervision therefore preserves strong geometry even in the earliest register, while every additional register remains useful for reconstruction.

The per-dataset comparison gains geometry on 16 of 18 datasets, ties on `usc_cloth_sim`, and decreases on `columbia_pusht` (0.3371 versus 0.6580). TopAct has higher normalized and raw reconstruction MSE on all 18 datasets, with absolute levels spanning two orders of magnitude across sources and a relative excess of 10–74% per dataset (Figures 4 and 5 in the appendix). The right panel of Figure 3 breaks code usage down by register. Both codecs use all 1,125 vocabulary entries in aggregate, and per-register observed vocabulary is nearly saturated for both arms (minimum 89.8% versus 90.1%). TopAct's per-register effective code counts are consistently lower, most sharply in the first register (354.6 versus 484.4), consistent with its lower aggregate effective count (815.8 versus 940.2). Lower target entropy can itself make categorical prediction easier, so geometry and token accuracy must be interpreted alongside this change in code usage.

4.3 Stage 2a: Compact Policies Across Four Suites

Table 2 reports the selected policies; the four-layer, 27M-parameter policy has no pretrained components, so its results reflect the action representation rather than backbone scale. The TopAct policies have higher autoregressive token accuracy at every evaluated prefix in all four suites, so the topology-aware targets are consistently easier for the compact policy to predict. The same ordering holds in closed-loop success: TopAct reaches higher success on Spatial (65.2% versus 63.2%), Goal (68.6% versus 57.4%), and LIBERO-10 (38.8% versus 35.6%), while the two arms are within one point on Object (29.0% versus 29.8%). Averaged over the four suites, TopAct succeeds on 50.4% of trials versus 46.5% for Recon: the topology-aware targets pair consistently higher token accuracy with higher closed-loop success on three of four suites. Stage 2b evaluates the same codecs under a stronger backbone with proxy-based selection.

Table 2: **Stage 2a selected-policy results.** Success uses 500 paired episodes per suite. Accuracy is task-macro greedy autoregressive token accuracy at $K = 4$ on validation, rather than reference-prefix accuracy. Each arm's checkpoint is selected under the same development-proxy rule.

	Success (%)		AR token accuracy (%)	
Suite	Reconstruction	TopAct	Reconstruction	TopAct
Spatial	63.2	**65.2**	14.23	24.23
Goal	57.4	**68.6**	15.88	26.21
Object	**29.8**	29.0	13.95	26.33
LIBERO-10	35.6	**38.8**	10.50	22.18
All four suites	46.5	**50.4**	—	—

Token accuracy in the table is task-macro greedy autoregressive accuracy at $K = 4$; values at the remaining prefixes appear in Appendix B.2. Neither categorical accuracy nor action MSE alone captures errors accumulated during feedback-driven manipulation.

4.4 Stage 2b: OpenVLA Training and Evaluation

Both OpenVLA arms finish 10,000 optimizer steps. Across matched 500-step training windows, TopAct has lower mean loss and higher mean token accuracy in every one of the 20 windows; Table 3 lists representative windows. TopAct reaches a mean loss of 1.01 and 72.9% token accuracy by the final window, versus 1.76 and 56.2% for Recon, and its advantage is already large in the first window. TopAct therefore converges both faster and to a stronger token model under this pretrained backbone. Checkpoint selection reflects this pace: on the 50-episode development proxy, TopAct reaches its best score by step 2,700 whereas Recon does not peak until step 8,400, and these two selected checkpoints define the closed-loop comparison below.

Table 3: **OpenVLA training with four-token targets.** Mean logged training loss and token accuracy over non-overlapping 500-step optimizer windows; representative windows shown, with TopAct ahead on both quantities in all 20 windows. The values summarize logged training minibatches and are not held-out validation metrics. The final row reports each arm's checkpoint selected on the 50-episode development proxy.

	Training loss ↓		Token accuracy (%) ↑	
Window end step	RECON	TopAct	RECON	TopAct
500	6.00	4.86	7	16
1,500	4.16	2.78	15	32
2,500	3.44	2.27	25	41
3,500	2.92	1.83	32	51
5,000	2.38	1.42	41	61
7,500	1.88	1.10	51	71
10,000	1.76	1.01	56	73
Selected step (dev. proxy)	8,400		2,700	

On all 12,078 LIBERO-10 validation windows, the selected TopAct checkpoint has lower task-macro token CE (3.3668 versus 5.5641), higher reference-prefix accuracy (22.19% versus 9.57%), higher greedy four-token accuracy (22.11% versus 9.53%), and lower greedy decoded MSE (0.042330 versus 0.046713) (Table 4). As in Stage 1, the codec-oracle ordering is reversed: RECON gives lower oracle MSE (0.002355 versus 0.003702), because the two arms decode through different Stage 1 codecs. Even so, TopAct's policy-minus-oracle gap is smaller (0.038628 versus 0.044358), so the topology-aware representation is both easier to predict and closer to its own codec ceiling.

Table 4: **Stage 2b results.** Offline values are task-macro validation metrics on all 12,078 windows. Success uses 500 paired official-state LIBERO-10 episodes per arm. Each arm's checkpoint is selected on the 50-episode development proxy (TopAct step 2,700; RECON step 8,400).

Metric	RECON	TopAct
Validation token CE ↓	5.5641	**3.3668**
Reference-prefix token accuracy (%) ↑	9.57	**22.19**
Greedy AR token accuracy, $K = 4$ (%) ↑	9.53	**22.11**
Decoded action MSE, $K = 4$ ↓	0.046713	**0.042330**
Codec-oracle action MSE ↓	**0.002355**	0.003702
Policy-minus-oracle MSE, $K = 4$ ↓	0.044358	**0.038628**
Closed-loop success	190/500 (38.0%)	**204/500 (40.8%)**

The closed-loop comparison favors TopAct: 204/500 (40.8%) versus 190/500 (38.0%), a paired difference of $+2.8$ percentage points, with 111 TopAct-only and 97 RECON-only successes over identical official initial states. This point estimate favors TopAct. We therefore read Stage 2b as showing that TopAct's stronger token learnability and faster convergence carry over to at least equal, and by point estimate higher, closed-loop success under a pretrained VLM backbone, without claiming a significant success gap at this single seed.

5 DISCUSSION AND LIMITATIONS

Geometry and reconstruction constrain different properties. The completed four-token study shows that an explicit neighborhood objective can preserve distance ranks at every prefix while increasing reconstruction error. Beyond the scope limits stated in Section 3, the neighborhood metric itself is uniform Euclidean distance in normalized action space, which may not emphasize contact transitions, gripper timing, or small task-dependent tolerances.

Codec learning and downstream success. The frozen-codec comparisons connect representation geometry to policy behavior. With the compact policy, TopAct is both easier to predict at the token level on every suite and higher in closed-loop success on three of four suites, with the fourth within

one point. With the OpenVLA backbone, this token-learnability advantage grows: TopAct trains faster and to a stronger token model, and its selected policy reaches a higher success point estimate (40.8% versus 38.0%). Across both policy classes, topology-aware targets benefit policy learning and control.

Offline metrics and closed-loop behavior. With OpenVLA, TopAct also lowers decoded action MSE and its policy-minus-oracle gap. Still, token accuracy depends on target entropy and does not weight errors by their decoded control consequences, and offline MSE measures behavior on demonstration observations, whereas rollout repeatedly conditions on states affected by earlier prediction errors. These distinctions explain why token-space and closed-loop orderings need not coincide; our experiments do not isolate a single causal mechanism.

Training and selection limits. The codec comparison has one completed training seed, and each policy comparison uses a single policy training seed. Stage 1 follows fixed-final selection; Stage 2a and Stage 2b each select a checkpoint independently under the same development-proxy rule. Episode-level paired intervals do not quantify uncertainty across training seeds or unseen tasks, and the Stage 2b success interval includes zero. The retained Stage 1 logs record final training outcomes but not a loss or validation trajectory, so codec convergence is reported only through endpoint metrics rather than learning curves.

Comparison and deployment limits. The study isolates the added geometry objective within our codec and downstream recipes. A four-token chunk shortens the autoregressive sequence, but an end-to-end speedup requires direct latency measurement. All policy results concern simulated manipulation with 16-action execution between observations, rather than physical-robot safety or generalization to other control frequencies.

6 Conclusion

TopAct represents 16 actions with four ordered tokens and aligns action neighborhoods with register neighborhoods before and after quantization at each prefix. The study demonstrates stronger post-quantization geometry and more predictable policy targets. With a frozen codec, compact policies trained on these targets achieve higher closed-loop success on three of four suites, and the targets train faster under an OpenVLA backbone with a higher closed-loop success point estimate. These benefits come with a codec reconstruction cost, and the Stage 2b success gap is not yet significant at a single seed; our results support topology-aware action representations and the joint evaluation of reconstruction, policy learning, and closed-loop behavior.

AI Use Statement

Generative AI assistants supported the preparation of this manuscript in three roles: editing and restructuring text, producing the scripts that render the figures and tables, and auditing reported numbers against the saved experimental artifacts. The experiments themselves — data collection, training, and evaluation — were conducted and controlled entirely by the authors, and no result was produced or modified by an AI system. Every AI-assisted passage was read and revised by the authors, who also verified each citation against the original publication, confirmed the architecture descriptions against the source code, and re-checked all reported values against machine-readable run records. The authors are fully responsible for the final manuscript and everything in it.

Reproducibility Statement

Appendix A records architecture, optimization, and evaluation settings, and Appendix B.3 reports the paired closed-loop counts underlying the Stage 2b statistics. The supplementary material contains the numerical table inputs and the scripts that generate every figure and table from the saved experimental artifacts. Final codec metrics are checked against machine-readable evaluations; policy success counts and pairing are checked against the saved episode records. Missing loss and validation histories are not reconstructed.

Ethics Statement

This study uses existing robot-action datasets and simulated manipulation tasks. The reported experiments do not establish physical-robot safety or generalization to human environments. Executing multiple predicted actions without feedback can amplify errors; physical deployment would require separate validation and safeguards.

References

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, et al. 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, et al. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. *arXiv preprint arXiv:2307.15818*, 2023.

Cheng Chi, Siyuan Feng, Qing Du, Zhen Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

Zibin Dong, Yicheng Liu, Shiduo Zhang, Baijun Ye, Yifu Yuan, Fei Ni, Jingjing Gong, Xipeng Qiu, Hang Zhao, Yinchuan Li, and Jianye Hao. ActionCodec: What Makes for Good Action Tokenizers. *arXiv preprint arXiv:2602.15397*, 2026. URL https://arxiv.org/abs/2602.15397.

Fawad Javed Fateh, Ali Shah Ali, Murad Popattia, Usman Nizamani, Andrey Konin, M. Zeeshan Zia, and Quoc-Huy Tran. A Hierarchical Spatiotemporal Action Tokenizer for In-Context Imitation Learning in Robotics. *arXiv preprint arXiv:2604.15215*, 2026. URL https://arxiv.org/abs/2604.15215.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv preprint arXiv:1512.03385*, 2015. URL https://arxiv.org/abs/1512.03385.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint arXiv:2106.09685*, 2021. URL https://arxiv.org/abs/2106.09685.

Ahad Jawaid and Yu Xiang. NAC: Neural Action Codec for Vision-Language-Action Models. *arXiv preprint arXiv:2606.21372*, 2026. URL https://arxiv.org/abs/2606.21372.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag 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. *arXiv preprint arXiv:2406.09246*, 2024. URL https://arxiv.org/abs/2406.09246.

Seungjae Lee, Yibin Wang, Haritheja Etukuru, H. Jin Kim, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Behavior Generation with Latent Actions. *arXiv preprint arXiv:2403.03181*, 2024. URL https://arxiv.org/abs/2403.03181.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. *arXiv preprint arXiv:2306.03310*, 2023. URL https://arxiv.org/abs/2306.03310.

Chaoqi Liu, Xiaoshen Han, Jiawei Gao, Yue Zhao, Haonan Chen, and Yilun Du. OAT: Ordered Action Tokenization. *arXiv preprint arXiv:2602.04215*, 2026. URL https://arxiv.org/abs/2602.04215.

Yicheng Liu, Shiduo Zhang, Zibin Dong, Baijun Ye, Tianyuan Yuan, Xiaopeng Yu, Linqi Yin, Chenhao Lu, Junhao Shi, Luca Jiang-Tao Yu, Liangtao Zheng, Tao Jiang, Jingjing Gong, Xipeng Qiu, and Hang Zhao. FASTer: Toward Efficient Autoregressive Vision Language Action Modeling via Neural Action Tokenization. *arXiv preprint arXiv:2512.04952*, 2025. URL https://arxiv.org/abs/2512.04952.

Huaihai Lyu, Chaofan Chen, Senwei Xie, Pengwei Wang, Xiansheng Chen, Shanghang Zhang, and Changsheng Xu. OmniSAT: Compact Action Token, Faster Auto Regression. *arXiv preprint arXiv:2510.09667*, 2025. URL https://arxiv.org/abs/2510.09667.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite Scalar Quantization: VQ-VAE Made Simple. *arXiv preprint arXiv:2309.15505*, 2023. URL https://arxiv.org/abs/2309.15505.

Michael Moor, Bastian Rieck, and Karsten Borgwardt. Topological autoencoders. In *International Conference on Machine Learning*, 2020.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural Discrete Representation Learning. *arXiv preprint arXiv:1711.00937*, 2017. URL https://arxiv.org/abs/1711.00937.

Open X-Embodiment Collaboration. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. *arXiv preprint arXiv:2310.08864*, 2023. URL https://arxiv.org/abs/2310.08864.

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.

Octo Model Team et al. Octo: An Open-Source Generalist Robot Policy. *arXiv preprint arXiv:2405.12213*, 2023.

An Dinh Vuong, Minh Nhat Vu, Dong An, and Ian Reid. Action Tokenizer Matters in In-Context Imitation Learning. *arXiv preprint arXiv:2503.01206*, 2025. URL https://arxiv.org/abs/2503.01206.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. *arXiv preprint arXiv:2304.13705*, 2023.

Ruijie Zheng, Ching-An Cheng, Hal Daumé III, Furong Huang, and Andrey Kolobov. PRISE: LLM-Style Sequence Compression for Learning Temporal Action Abstractions in Control. *arXiv preprint arXiv:2402.10450*, 2024. URL https://arxiv.org/abs/2402.10450.

A Training and Evaluation Settings

A.1 Architecture and Optimization

All experiments use the 16-action/four-register codec, with FSQ levels $(9, 5, 5, 5)$, a 1,125-value vocabulary, and prefixes $\{1, 2, 3, 4\}$. The codec has 14,691,876 parameters, a width of 256, four attention heads, SwiGLU expansion four, per-head FP32 query/key normalization, sinusoidal positions, and output dropout 0.05. The encoder has two action layers and four causal register-decoder layers; the detokenizer has two causal register layers and four action-decoder layers. Prefix probabilities for reconstruction are $(0.15, 0.20, 0.25, 0.40)$; the topology objective weights all prefixes equally.

Table 5: Executed training settings. Stage 1 uses a fixed-final EMA checkpoint; Stage 2a and Stage 2b each select a checkpoint on a development proxy, with the same rule applied independently to both arms.

Setting	Stage 1 codec	Stage 2a compact	Stage 2b OpenVLA
Training seeds	one per stage, identical across arms		
Epoch/step budget	800 epochs	100 epochs	10,000 steps
Global batch	128	256	128
GPUs per arm	4	1	4
Main learning rate	$5 \cdot 10^{-5}$	$5 \cdot 10^{-5}$	10^{-4} (head)
Vision learning rate	—	10^{-5}	Frozen
Attention LoRA rate	—	—	$5 \cdot 10^{-5}$
Projector LoRA rate	—	—	$2 \cdot 10^{-5}$
LR warmup steps	100	100	None
Schedule after warmup	Constant	Constant	Cosine
Policy/backbone precision	—	BF16	BF16
Codec precision	FP32, no TF32	FP32	FP32
EMA	Yes	Yes	No

The codec optimizer is AdamW with betas $(0.9, 0.95)$, epsilon 10^{-8}, zero weight decay, and gradient clipping 1.0. The topology weights warm to 0.2 over 350,640 updates, while the sharpness η decreases from 1 to 0.9 over the same span. Distance temperatures are updated every 30 epochs, with a floor of 0.01; in distributed training, local representation queries are differentiated against detached globally gathered references. The same architecture, initialization, training schedule, and data order are used in the Recon control, with both topology weights set to zero. All results use a single codec training seed and a single policy training seed, identical across arms.

Stage 2a uses separate randomly initialized ResNet-18 camera encoders with GroupNorm, 76-pixel crops, and 32 spatial-softmax keypoints per camera. Each camera yields 64 features. Nine state coordinates and a 16-dimensional task embedding complete a 153-dimensional condition vector at each of two observation timesteps. The policy has 27,443,552 trainable parameters; all codec parameters remain frozen. Canonical labels use CPU FP32 with autocast and TF32 disabled and the math attention backend selected.

The Stage 2b state projector is $18 \to 256 \to 4096$ with GELU. Rank-16 LoRA adapts 128 language-model attention projections and three multimodal-projector linear layers. LIBERO-10 yields 110,512 training windows and 12,078 validation windows. Four learned action queries condition the four-layer, width-256 causal head. The head uses LayerNorm, GELU, dropout 0.1, and tied token input/output weights. LoRA rank/alpha/dropout are $16/16/0.05$. Each of four ranks uses microbatch two and accumulation 16, giving global batch 128. Training runs for 10,000 optimizer steps under a cosine schedule with $T_{\max} = 10{,}000$ and no warmup. Every 300 steps the resident model is scored on a fixed 50-episode LIBERO-10 development proxy (five official states per task), and each arm's best-scoring checkpoint is retained for the official evaluation. The logged training loss and accuracy are from the last microbatch on rank zero at each optimizer update; plotted 500-step means are not global-batch averages or full-training-set evaluations.

A.2 Rollout and Statistical Protocol

Codec geometry is evaluated on up to 256 deterministically episode-stratified validation chunks per dataset; distance pairs share chunks and are not treated as independent observations. Each suite has ten tasks and 50 official initial states per task. Evaluation and simulator seeds are fixed once per task environment and shared across arms. Ten settling actions precede the first fresh policy observation. Greedy generation emits four tokens and the decoder supplies all 16 actions for execution before replanning. Episode limits are 220 for Spatial, 300 for Goal, 280 for Object, and 520 for LIBERO-10. State indices, state hashes, task identities, and trial indices are checked before forming paired outcomes.

The effect is the TopAct-minus-reconstruction success-rate difference. Exact two-sided McNemar tests use discordant pairs, and paired bootstrap intervals use 20,000 draws with a fixed seed. The 50-episode selection proxy shares initial states with the official evaluation, so it is a development signal rather than an independent held-out benchmark. Single-seed intervals do not quantify uncertainty across training seeds. Task-level effects are descriptive and not evidence of uniform dominance across task families.

B Additional Codec and Policy Results

B.1 Final Codec Prefixes

Table 6: Final 800-epoch codec evaluation. Every prefix reconstructs the same 16-action chunk; these are not learning-curve samples.

Metric	Arm	$K = 1$	$K = 2$	$K = 3$	$K = 4$
Normalized MSE	Reconstruction	0.045034	0.015405	0.009075	0.006956
	TopAct	0.048856	0.018036	0.011313	0.008867
Post-FSQ Spearman	Reconstruction	0.6956	0.6781	0.6497	0.6569
	TopAct	0.8596	0.8939	0.9088	0.9245

Effective code count is $\exp(H(C))$ and is a count, not a percentage. Aggregate utilization pools observations across registers. Both codecs have 100% aggregate coverage, but this does not imply uniform usage or equivalent information in each register. The geometric metric uses mean squared distances in normalized scalar coordinates, not differences between serialized token IDs. Figure 4 visualizes the per-dataset geometry and MSE differences behind the aggregate counts.

B.2 Selected Stage 2a Offline Results

Table 7: Stage 2a selected checkpoints. Action MSE and greedy autoregressive accuracy use the full four-token prefix and are averaged equally over tasks.

Suite	Arm	Action MSE	AR accuracy (%)
Spatial	Reconstruction-only	0.041735	14.23
Spatial	TopAct	0.041260	24.23
Goal	Reconstruction-only	0.036138	15.88
Goal	TopAct	0.035810	26.21
Object	Reconstruction-only	0.033386	13.95
Object	TopAct	0.033825	26.33
LIBERO-10	Reconstruction-only	0.041464	10.50
LIBERO-10	TopAct	0.044387	22.18

The table reports reevaluated selected policies, which may differ slightly from metrics logged during training. Autoregressive accuracy uses generated prefixes; reference-prefix accuracy uses reference prefixes and is a distinct measurement.

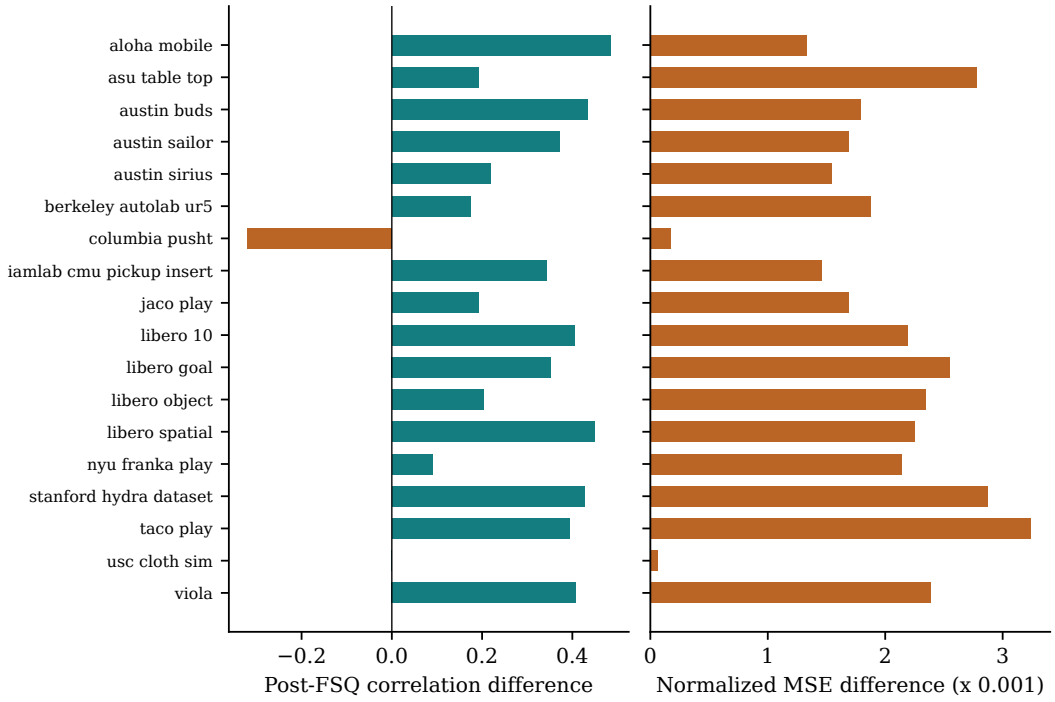

Figure 4: **Stage 1 effects across all 18 datasets.** Each row compares the same final codec pair at $K = 4$. TopAct improves post-FSQ geometry on 16 datasets, ties on one, and declines on one. It increases normalized MSE on all 18. No individual dataset is selected on the basis of a favorable result.

B.3 Stage 2b Paired Counts

The proxy-selected 500-trial comparison contains 93 joint successes, 199 joint failures, 111 TopAct-only successes, and 97 Recon-only successes, giving 204 versus 190 successes and a $+2.8$-point difference. The exact two-sided McNemar test on the 208 discordant pairs gives $p = 0.37$, and the paired bootstrap 95% interval is $[-2.8, +8.4]$ points. The success point estimate favors TopAct, but the interval includes zero at this single training seed.

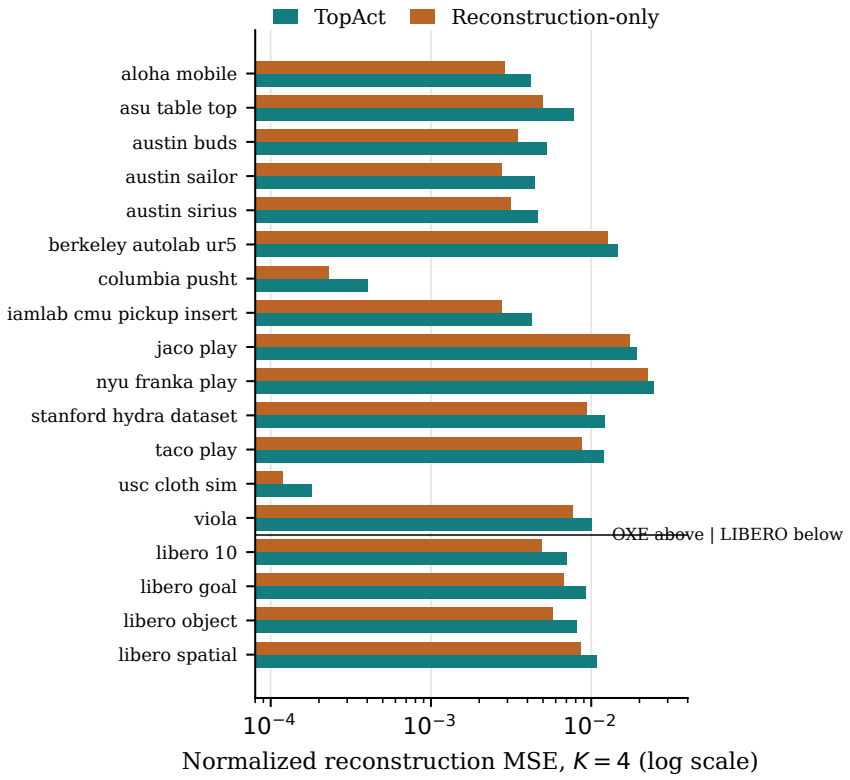

Figure 5: **Per-dataset normalized reconstruction MSE at $K = 4$.** The 14 generalist (OXE) sources are shown above the divider and the four LIBERO suites below it, on a log scale. RECON is lower on every dataset, with dataset-dependent absolute levels spanning two orders of magnitude.