

5 Supplementary Information

This Supplementary Information provides technical detail supporting the analyses in the main text. Supplementary Note 1 summarizes the VLA architectures and action-tokenization schemes used in the study; Supplementary Note 2 defines the neural-collapse metrics; Supplementary Note 3 gives the optimization-level intuition for why action-token prediction can induce neural-collapse geometry; and Supplementary Note 4 describes the latent-space perturbation protocol.

5.1 Vision–Language–Action architectures

We analyze four representative transformer-based Vision–Language–Action (VLA) models: OpenVLA [6], π_0 -FAST [2, 28], BeT [29], and VQ-BeT [30]. These models span the major architectural and action-tokenization choices used in current robotic foundation models. They differ in language backbones, visual encoders, transformer architectures, and action-token representations, yet all formulate policy learning as next-token prediction over discrete action tokens. This shared training objective makes it possible to compare latent geometry across otherwise heterogeneous VLA systems.

The central architectural distinction for this study is how each model converts continuous robot actions into discrete tokens. The resulting tokenizers differ in vocabulary size, token frequency, semantic granularity, and temporal structure. They therefore provide a stringent test of whether the neural-collapse geometry reported in the main text is a general consequence of action-token supervision rather than an artifact of a particular discretization scheme.

5.1.1 OpenVLA

OpenVLA [6] builds on a pretrained vision–language model with a fused DINOv2–SigLIP visual encoder and a Llama-2 language backbone. Continuous robot actions are represented by independently discretizing each action dimension with uniform quantization,

$$z_i = \left\lfloor \frac{a_i - a_i^{\min}}{\Delta_i} \right\rfloor, \quad (10)$$

where $a_i^{\min}$ is the minimum value of the action range and Δ_i is the corresponding quantization interval.

This independent discretization preserves the semantic interpretation of each action dimension: neighboring token indices correspond to nearby continuous motions along the same coordinate. The action tokens are then generated autoregressively by the language-model decoder.

5.1.2 π_0 -FAST

π_0 -FAST [2, 28] extends the π_0 architecture with Frequency-space Action Sequence Tokenization (FAST). Instead of tokenizing individual actions, FAST first compresses short action trajectories in the frequency domain,

$$\tilde{a} = \text{DCT}(a_{t:t+T}), \quad (11)$$

where DCT denotes the discrete cosine transform.

The transformed coefficients are subsequently quantized into discrete action tokens. Compared with per-step discretization, this representation captures temporal correlations and provides a compact description of long-horizon robot motion.

5.1.3 BeT

Behavior Transformer (BeT) [29] represents each continuous action by a single discrete token obtained through k -means clustering,

$$z = \arg \min_k \|a - c_k\|_2^2, \quad (12)$$

where $\{c_k\}$ are cluster centers learned from the demonstration data.

Unlike independent discretization, BeT assigns the complete action vector to one cluster, allowing correlations among action dimensions to be captured directly. Semantic similarity is therefore determined by the geometry of the learned action clusters rather than by individual action coordinates.

5.1.4 VQ-BeT

VQ-BeT [30] replaces the fixed k -means tokenizer with a learned vector-quantized representation. Continuous actions are first embedded into a latent space through an encoder $f(\cdot)$ and then assigned to the nearest codebook vector,

$$z = \arg \min_k \|f(a) - e_k\|_2^2, \quad (13)$$

where $\{e_k\}$ denotes the learned codebook.

Unlike fixed discretization schemes, the codebook adapts to the empirical action distribution. The learned representation also supports hierarchical action organization, which is used in the main text to test whether latent geometry reflects coarse-to-fine action semantics.

5.1.5 Comparison of action-token representations

The four VLA models therefore discretize robot actions using fundamentally different representations. They differ in

- the unit being discretized (individual action dimensions, complete action vectors, trajectory segments, or learned latent embeddings);
- vocabulary size;
- action-token frequency distribution;
- semantic granularity; and
- temporal structure.

As shown in Fig. 2 and Extended Data Fig. 1, these tokenization strategies produce action-token distributions that differ by orders of magnitude in vocabulary size and

class imbalance. Some tokenizers generate only a few dozen frequently used tokens, whereas others produce hundreds or thousands of sparse action tokens. These differences make it far from obvious that a common latent geometry should emerge across models.

Nevertheless, the main text shows that neural-collapse geometry develops across all four tokenization strategies. This observation suggests that the emergence of structured latent geometry is primarily driven by action-token supervision itself rather than by the particular discretization method used to represent robot actions.

5.1.6 Representation extraction

Unless otherwise specified, all analyses use the latent representation immediately before the final action-prediction head. This representation is mapped directly to action-token logits through a linear classifier and is therefore the feature space most directly shaped by action-token supervision. For the layer-wise analyses, hidden representations are additionally extracted from intermediate transformer blocks before the final prediction layer.

5.1.7 Analysis protocol

Neural-collapse metrics are computed with the action-token assignments appropriate to each analysis. For training-dynamics analyses, these are the tokenizer-defined target action tokens. For perturbation analyses performed during closed-loop rollout, the clean model’s predicted action tokens are used to estimate the geometry on which the interventions operate. This convention keeps the observational analyses tied to the behavior-cloning objective while ensuring that online perturbations use information available from the trained policy itself.

For perturbation experiments, latent representations and predicted action tokens are first collected from the trained policy to estimate the learned geometry, including class centers and the class-mean subspace. Perturbations are then applied either offline to the recorded latent representations or online during policy rollout while keeping all network parameters fixed. This protocol separates the effects of latent geometry on representation organization, action-token prediction, and closed-loop robotic performance.

5.2 Computation of neural-collapse metrics

The main text uses neural collapse (NC) to quantify how action-token supervision organizes the latent representations of Vision–Language–Action (VLA) models. This note gives the full definitions of the metrics used throughout the paper and explains how they are adapted to action-token prediction.

Unlike semantic classes in conventional image classification, action tokens are obtained by discretizing continuous robot actions. They can therefore be highly imbalanced, tokenizer-dependent, and heterogeneous in semantic granularity. The metrics below should be interpreted as quantitative descriptors of action-conditioned latent geometry rather than as tests for an exact realization of the ideal neural-collapse limit.

1151 5.2.1 Notation

1152 Consider the latent representation immediately before the action-prediction head for
 1153 a collection of N samples,
 1154

$$1155 \quad \mathcal{H} = \{h_i\}_{i=1}^N, \quad h_i \in \mathbb{R}^d,$$

1156 where each sample is assigned an action-token label $y_i \in \{1, \dots, C\}$.
 1157 For each action token c , the class center is

$$1158 \quad \mu_c = \frac{1}{N_c} \sum_{i:y_i=c} h_i, \quad (14)$$

1159 where N_c is the number of samples assigned to token c .
 1160 The global mean over all latent representations is

$$1161 \quad \bar{\mu} = \frac{1}{N} \sum_{i=1}^N h_i, \quad (15)$$

1162 and the within-class variance for token c is

$$1163 \quad \sigma_c^2 = \frac{1}{N_c} \sum_{i:y_i=c} \|h_i - \mu_c\|_2^2. \quad (16)$$

1164 To compare quantities across models and datasets with different feature scales, we
 1165 use the coefficient of variation,

$$1166 \quad \text{CoV}(x) = \frac{\text{Std}(x)}{|\text{Mean}(x)|}, \quad (17)$$

1167 which provides a scale-independent measure of dispersion.
 1168

1169 5.2.2 NC1: Within-class compactness

1170 NC1 measures the compactness of latent representations associated with the same
 1171 action token relative to the separation between different action-token clusters.
 1172 Following previous work, we compute
 1173

$$1174 \quad \hat{\sigma}_{c,c'} = \frac{\sigma_c^2 + \sigma_{c'}^2}{2\|\mu_c - \mu_{c'}\|_2^{k+2}}, \quad (18)$$

1175 where $k = 2$ throughout this work.

1176 The reported NC1 score is

$$1177 \quad \text{NC}_1 = \text{Mean}_{c \neq c'} \hat{\sigma}_{c,c'}. \quad (19)$$

1178 Lower values indicate stronger concentration of latent representations around their
 1179 corresponding action-token centers.
 1180
 1181
 1182
 1183
 1184
 1185
 1186
 1187
 1188
 1189
 1190
 1191
 1192
 1193
 1194
 1195
 1196

5.2.3 NC2: Global geometric organization

NC2 characterizes the global distribution of action-token centers in latent space. Following the original neural-collapse formulation, we evaluate both the consistency of class-center norms and the regularity of pairwise angles.

The norm variation is

$$\text{NC}_{2,\text{norm}} = \text{CoV}(\log \|\mu_c - \bar{\mu}\|_2), \quad (20)$$

while the angular deviation from the simplex equiangular tight frame (ETF) is

$$\text{NC}_{2,\text{angle}} = \left| \text{Mean}_{c \neq c'} \left(\langle \hat{\mu}_c, \hat{\mu}_{c'} \rangle + \frac{1}{C-1} \right) \right|, \quad (21)$$

where

$$\hat{\mu}_c = \frac{\mu_c - \bar{\mu}}{\|\mu_c - \bar{\mu}\|_2}.$$

Lower values indicate stronger agreement with the ideal ETF geometry.

Because VLA datasets exhibit severe action-token imbalance, exact ETF geometry is generally not expected. We therefore also report a relaxed hyperspherical-uniformity metric,

$$\text{GNC}_2 = \text{CoV}_{c \neq c'}(\log \|\hat{\mu}_c - \hat{\mu}_{c'}\|_2^{-1}), \quad (22)$$

which evaluates the uniformity of action-token centers over all pairwise directions without requiring exact ETF structure.

5.2.4 NC3: Representation–classifier alignment

NC3 measures whether the learned latent geometry is aligned with the linear action-prediction layer:

$$\text{NC}_3 = \text{Mean}_c \|\hat{\mu}_c - \hat{w}_c\|_2, \quad (23)$$

where

$$\hat{w}_c = \frac{w_c}{\|w_c\|_2}$$

is the normalized classifier weight corresponding to action token c .

Lower values indicate stronger agreement between the learned latent geometry and the action-prediction head.

5.2.5 NC4: Geometry-driven prediction

Finally, NC4 evaluates whether the learned action-prediction rule is consistent with nearest-class-center inference in the latent space:

$$\text{NC}_4 = \frac{1}{N} \sum_{i=1}^N \mathbb{I} \left[\arg \max_c w_c^\top h_i = \arg \max_c (2h_i^\top \mu_c - \|\mu_c\|_2^2) \right]. \quad (24)$$

Higher NC4 indicates that the classifier increasingly behaves like a nearest-class-center assignment rule.

1243 5.2.6 Adaptation to action-token prediction

1244 Classical neural collapse assumes balanced semantic classes, whereas VLA models pre-
1245 dict action tokens obtained from continuous robot trajectories. These tokens exhibit
1246 substantial class imbalance, tokenizer-dependent vocabularies, and heterogeneous
1247 semantic structure.

1248 Despite these differences, the NC metrics remain applicable because they depend
1249 only on latent representations and their corresponding action-token assignments. All
1250 geometric quantities are computed independently for each VLA model using its own
1251 action-token vocabulary. Consequently, the reported metrics characterize the intrinsic
1252 geometry learned by each model without requiring a shared action representation
1253 across architectures.

1255 5.3 Why action-token prediction induces neural-collapse 1256 geometry

1258 The main text gives an intuitive account of why neural-collapse (NC) geome-
1259 try emerges during Vision-Language-Action (VLA) training. Here we provide the
1260 corresponding optimization-level explanation in more detail.

1261 VLA models learn from multimodal observations consisting of images, language
1262 instructions, and robot states, while their supervision signal is provided by discrete
1263 action tokens obtained from continuous robot actions. Latent representations are
1264 therefore optimized not only to encode observations but also to arrange those obser-
1265 vations according to the action space. The argument below should be read as a local
1266 gradient-level mechanism rather than a full convergence proof: it explains why the
1267 training objective encourages the geometric organization measured by NC.

1269 5.3.1 Cross-entropy optimization

1270 Let $h(x_{1:t}) \in \mathbb{R}^d$ denote the latent representation immediately before the action-
1271 prediction head. For an action-token vocabulary V , the prediction logits are

$$1274 \quad z_c = w_c^\top h + b_c, \quad c \in V, \quad (25)$$

1275 where w_c and b_c are the classifier weight and bias associated with action token c .
1276 The predicted probability is

$$1278 \quad p_c = \frac{\exp(z_c)}{\sum_{j \in V} \exp(z_j)}. \quad (26)$$

1280 For a sample with target action token y , the cross-entropy loss is

$$1282 \quad \mathcal{L} = -\log p_y. \quad (27)$$

1283 Differentiating with respect to the latent representation gives

$$1285 \quad \nabla_h \mathcal{L} = \sum_{c \in V} p_c w_c - w_y. \quad (28)$$

1287 Thus, the gradient step applied to the latent representation is proportional to
1288

$$\Delta h \propto w_y - \sum_{c \in V} p_c w_c, \quad (29)$$

where the proportionality constant is determined by the learning rate and back-propagation through the preceding layers. Equation (28) therefore provides a geometric interpretation of how action-token supervision progressively organizes latent space.

5.3.2 Within-class contraction

Consider two latent representations assigned to the same action token,

$$h_i, h_j, \quad y_i = y_j.$$

At the level of the final representation, the difference between their gradient steps is

$$\Delta h_i - \Delta h_j \propto - \sum_{c \in V} (p_c^{(i)} - p_c^{(j)}) w_c. \quad (30)$$

As training proceeds, samples associated with the same action token become increasingly confident and produce more similar probability distributions,

$$p_c^{(i)} \approx p_c^{(j)}.$$

Their differential update therefore decreases, while all samples assigned to the same token continue to share attraction toward the same classifier direction. Relative to the growing separation between different action-token centers, same-token representations become increasingly compact. This mechanism explains the emergence of NC1.

5.3.3 Global organization of action-token clusters

Equation (28) contains two competing forces: attraction toward the classifier direction of the target action token and repulsion away from competing action-token directions.

Averaging over all samples belonging to action token y gives the approximate evolution of the corresponding cluster center,

$$\mu_y \leftarrow \mu_y + \eta \left(w_y - \sum_{c \in V} \bar{p}_{c|y} w_c \right), \quad (31)$$

where $\bar{p}_{c|y}$ denotes the average predicted probability of action token c over samples assigned to token y .

As prediction becomes increasingly accurate,

$$\bar{p}_{y|y} \rightarrow 1,$$

the attraction toward the target classifier dominates. Different action-token clusters therefore separate from one another while their directions become more systematically organized, giving rise to the global geometry measured by NC2.

1335 5.3.4 Alignment between representations and classifier weights

1336 The classifier weights evolve according to

$$1338 \quad \nabla_{w_y} \mathcal{L} = (p_y - 1)h, \quad \nabla_{w_c} \mathcal{L} = p_c h, \quad c \neq y. \quad (32)$$

1340 Thus, each classifier weight is repeatedly attracted toward latent representations
1341 assigned to its own action token while being repelled by representations assigned to
1342 competing tokens.

1343 At the same time, Eq. (28) shows that latent representations are attracted toward
1344 the corresponding classifier direction. This bidirectional optimization progressively
1345 aligns latent geometry with the action-prediction layer, producing the self-duality
1346 measured by NC3.

1347 5.3.5 Geometry-driven action prediction

1349 As within-class variation decreases and action-token clusters become increasingly
1350 separated and aligned with the prediction head, the linear prediction rule

$$1352 \quad \hat{y} = \arg \max_c w_c^\top h \quad (33)$$

1354 approaches a nearest-class-center rule

$$1356 \quad \hat{y} = \arg \max_c (2h^\top \mu_c - \|\mu_c\|_2^2). \quad (34)$$

1357 Consequently, action prediction becomes increasingly determined by the geometry
1358 of the latent space itself, giving rise to NC4.

1360 5.3.6 Implications for VLA models

1362 VLA models learn from highly heterogeneous multimodal observations, and their
1363 action tokens are derived from continuous robot trajectories rather than from fixed
1364 semantic classes. Nevertheless, the local optimization dynamics remain governed by
1365 action-token prediction. The analysis above suggests that action supervision naturally
1366 encourages increasingly structured action-aligned latent geometry. The experiments
1367 in the main text verify that this geometry emerges across architectures, tokenization
1368 strategies, datasets, and robotic tasks, and further show that it plays a functional role
1369 in action prediction and policy performance.

1371 5.4 Geometry perturbation experiments

1372 To test whether the learned latent geometry plays a functional role in action pre-
1373 diction, we perform controlled perturbation experiments. Unlike the observational
1374 analyses in the main text, these experiments intervene directly on latent represen-
1375 tations while keeping all model parameters fixed. By selectively modifying different
1376 geometric properties of the latent space, we evaluate their effects on action-token
1377 prediction, neural-collapse metrics, and closed-loop policy performance.

1378 Unless otherwise specified, all perturbations are applied to the latent representation
1379 immediately before the final action-prediction head. For each trained model, we first
1380

collect latent representations together with the clean model’s predicted action tokens. These predictions are used to estimate the learned geometry, including action-token centers and the class-mean subspace. Thus, the perturbations operate on the geometry expressed by the model itself and do not require oracle action labels during rollout.

For each perturbation, we evaluate closed-loop task success and offline action-token prediction using the perturbed representations. We also recompute the neural-collapse metrics to quantify how each intervention modifies the learned geometry.

5.4.1 Norm-preserving isotropic perturbation

As a geometry-agnostic baseline, we inject isotropic Gaussian noise into each latent representation,

$$h' = h + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I), \quad (35)$$

followed by a normalization step that restores the original feature norm:

$$h' \leftarrow \frac{\|h\|_2}{\|h'\|_2} h'. \quad (36)$$

This perturbation changes the direction of the latent representation while preserving its magnitude, providing a baseline intervention that is independent of the learned class structure.

5.4.2 Within-class destructive perturbation

To selectively weaken within-class compactness, we decompose each latent representation into its corresponding action-token center and residual,

$$h = \mu_c + r, \quad (37)$$

where μ_c denotes the center of the predicted action-token cluster.

The residual is then amplified,

$$h' = \mu_c + \alpha r, \quad \alpha \geq 1. \quad (38)$$

This intervention preserves the cluster centers while increasing the spread of representations within each action-token cluster. It therefore primarily targets NC1 while leaving the global arrangement of cluster centers largely unchanged.

5.4.3 Between-class destructive perturbation

To perturb the global organization of latent space without directly modifying within-class residuals, we decompose each centered representation into two orthogonal components,

$$h - \bar{\mu} = p_{\text{mean}} + p_{\perp}, \quad (39)$$

where p_{mean} lies in the subspace spanned by the centered action-token means and $p_{\perp}$ is orthogonal to this subspace.

We then attenuate the component responsible for global class organization:

$$h' = \bar{\mu} + \beta p_{\text{mean}} + p_{\perp}, \quad 0 \leq \beta \leq 1. \quad (40)$$

As β decreases, separation between action-token clusters becomes progressively weaker while the local residual structure is approximately preserved. This perturbation therefore primarily targets NC2, NC3, and NC4.

5.4.4 Geometry-preserving projection

Finally, we consider a constructive intervention that reinforces the learned latent geometry by moving representations toward the center of their corresponding action-token cluster:

$$h' = (1 - \lambda)h + \lambda\mu_c, \quad 0 \leq \lambda \leq 1. \quad (41)$$

Unlike the destructive perturbations above, this operation reduces within-class variation while preserving the global arrangement of action-token centers. It therefore strengthens the learned geometry rather than disrupting it.

5.4.5 Relationship among perturbations

The four perturbations target complementary aspects of the learned latent geometry:

- **Norm-preserving isotropic perturbation** randomly changes representation directions while preserving feature magnitude.
- **Within-class destructive perturbation** increases the spread of individual action-token clusters and primarily weakens NC1.
- **Between-class destructive perturbation** degrades the global organization of action-token clusters while approximately preserving local residual structure.
- **Geometry-preserving projection** reduces within-class variation without altering the global arrangement of action-token centers.

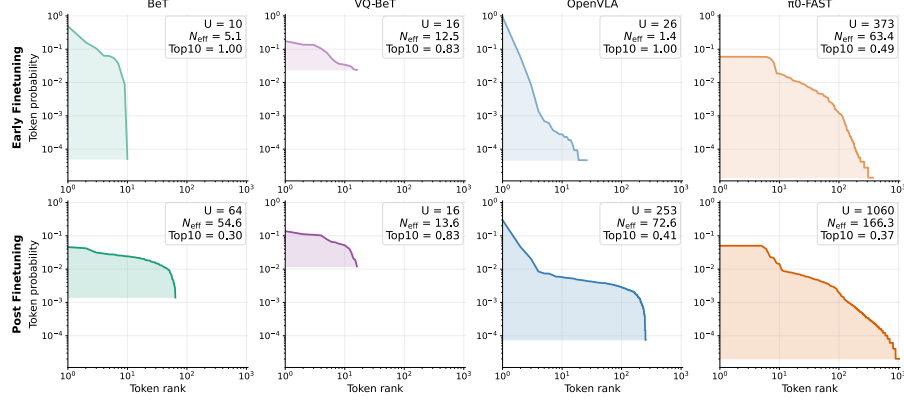
Together, these interventions provide complementary probes of the learned latent geometry and allow us to distinguish the contributions of local compactness, global organization, and feature-classifier alignment to action prediction and closed-loop robotic performance.

6 Extended Data

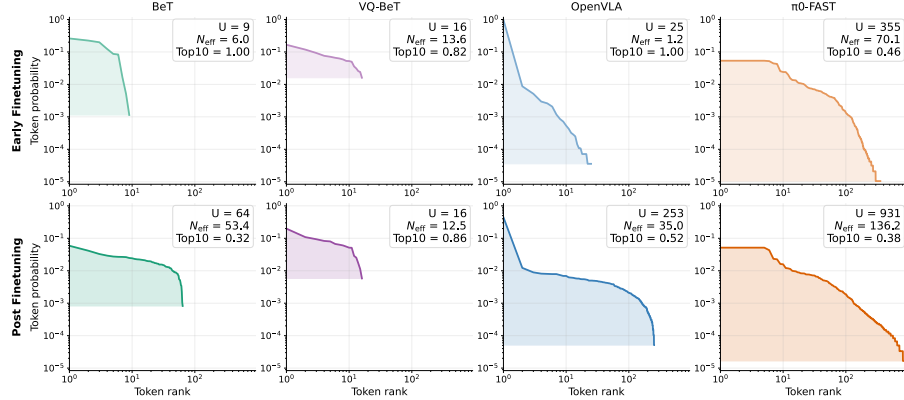
Model	Backbone	Tokenizer	Unit	Learned Token?	Key property
OpenVLA	Llama-2	Uniform bins	Action dimension	No	Semantic bins
π_0 -FAST	Transformer	FAST/DCT	Trajectory	Partially	Temporal compression
BeT	Transformer	k -means	Action vector	No	Action prototypes
VQ-BeT	Transformer	VQ	Latent action	Yes	Hierarchical codebook

Extended Data Table 1 Representative Vision–Language–Action models analyzed in this work. The models span the major architectural and action-tokenization paradigms used in modern robotic foundation models, providing diverse settings for evaluating whether latent geometry emerges across tokenization schemes.

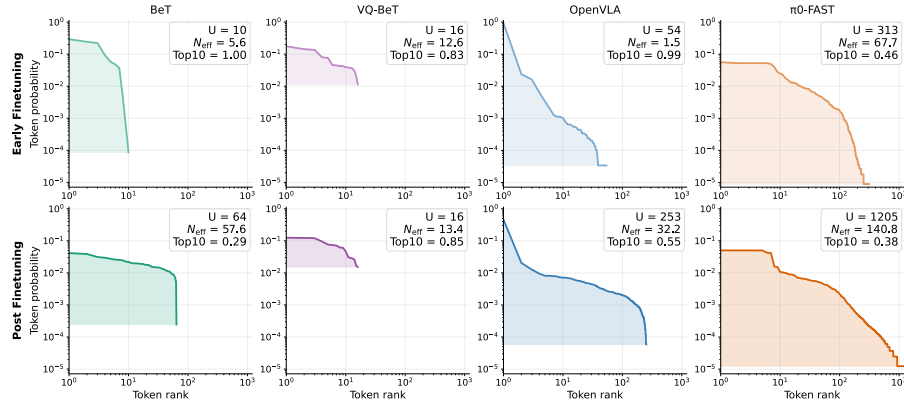
a LIBERO-SPATIAL



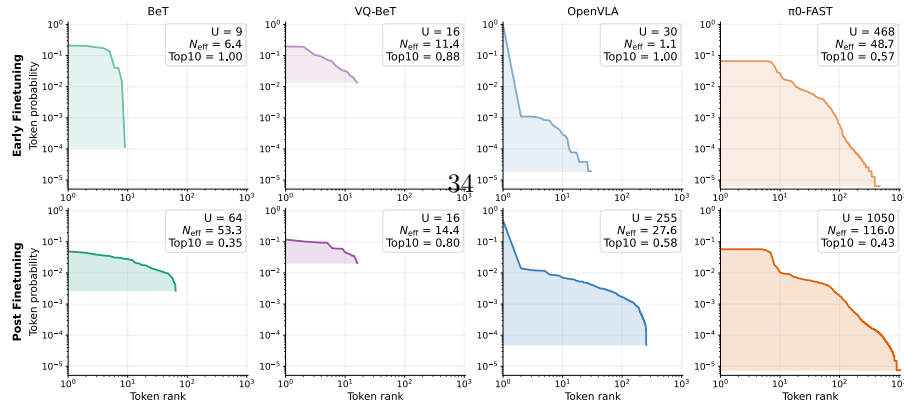
b LIBERO-OBJECT



c LIBERO-GOAL

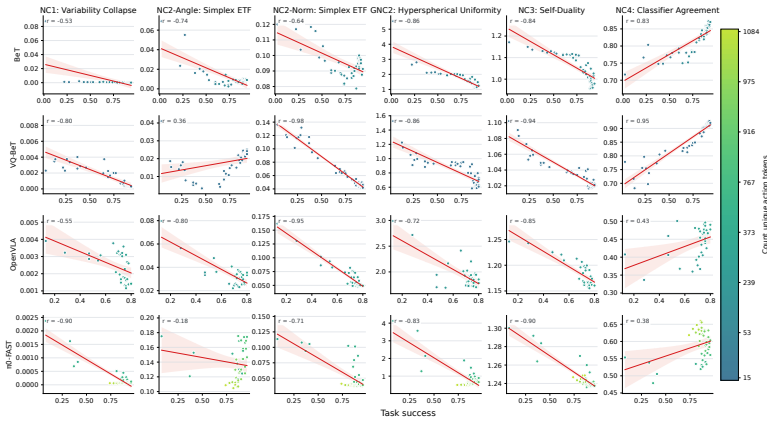


d LIBERO-LONG

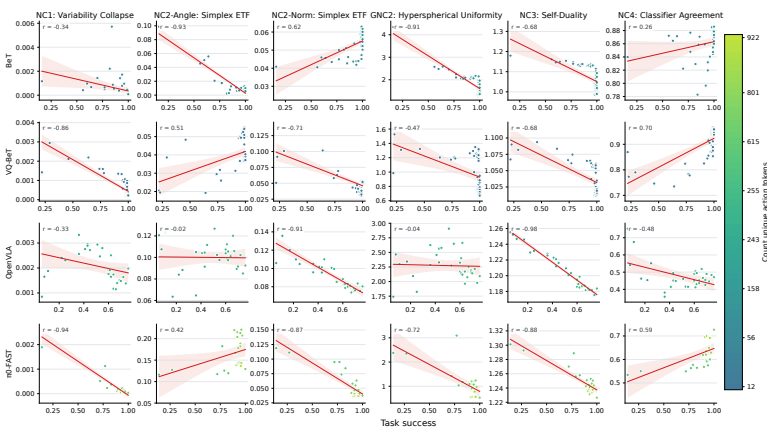


Extended Data Fig. 1 Action-token counts across tokenization strategies and LIBERO task suites. Token-count statistics show substantial differences in vocabulary size and class imbalance across VLA models, providing the empirical basis for testing whether neural-collapse geometry is robust to heterogeneous action-token distributions.

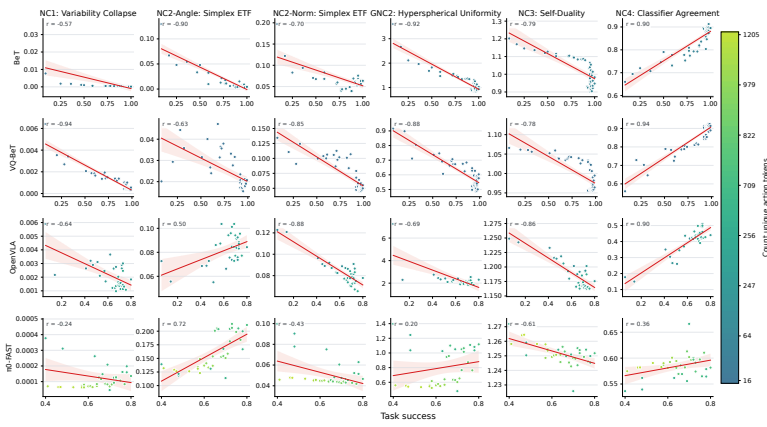
a LIBERO-SPATIAL



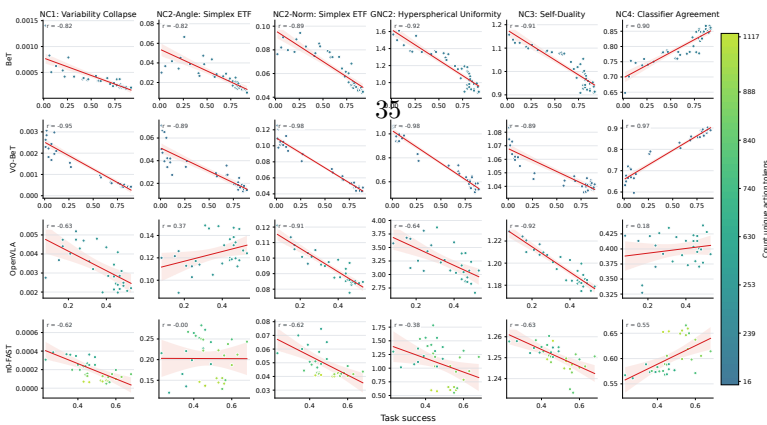
b LIBERO-OBJECT



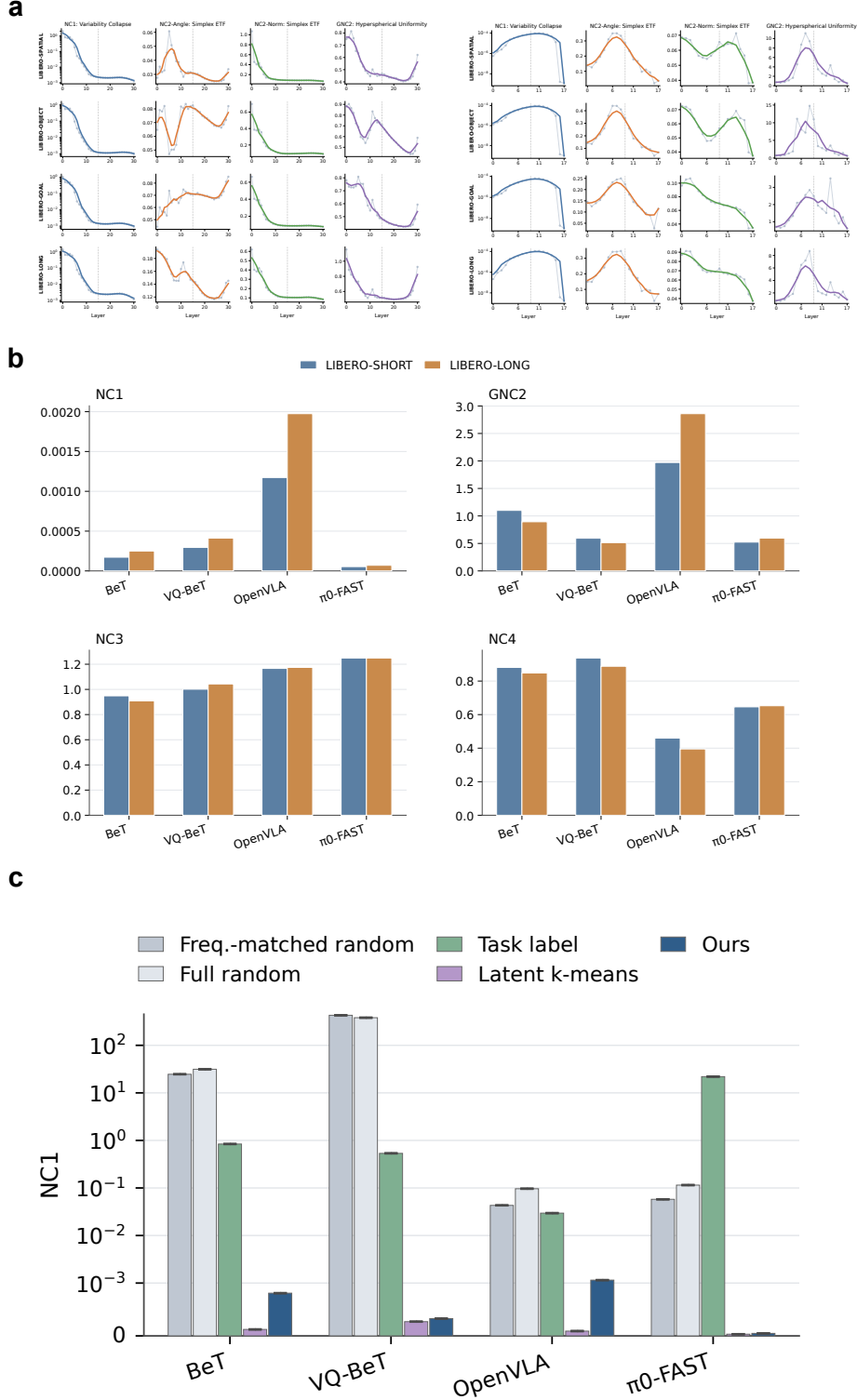
c LIBERO-GOAL



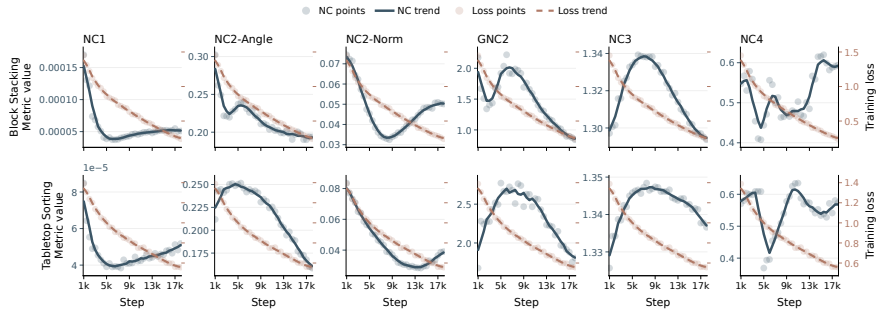
d LIBERO-LONG



Extended Data Fig. 2 Correlations between NC metrics and task success across LIBERO. Across VLA models and LIBERO task suites, NC1, NC2-Norm, NC2-Angle, GNC₂, and NC3 are predominantly negatively correlated with task success, whereas NC4 is positively correlated with task success. These trends support the association between stronger neural-collapse geometry and improved policy performance.

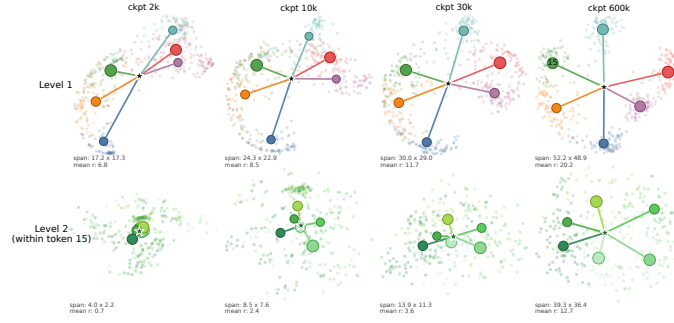


Extended Data Fig. 3 Ablation analysis of neural-collapse geometry in VLA models. **a**, Layer-wise evolution of representation-only NC metrics, showing the progressive emergence of structured geometry across transformer layers. **b**, NC metrics across models and LIBERO task suites. Long-horizon tasks exhibit weaker geometric structure than shorter tasks, consistent with their greater difficulty. **c**, NC1 under alternative labeling strategies. Meaningful collapse arises for action-token labels but not for random, frequency-matched random, task-level, or latent k -means labels, indicating that the geometry is induced by control-level action supervision rather than spurious dataset structure.

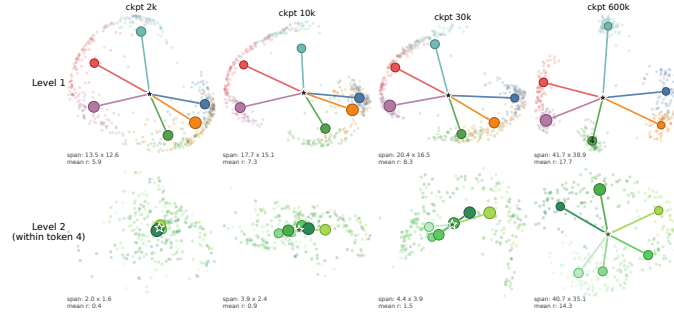


Extended Data Fig. 4 Layer-wise NC analysis for π_0 -FAST during real-world fine-tuning. NC metrics computed across transformer layers show the progressive emergence of structured latent geometry during real-world robot learning, extending the layer-wise analysis beyond the Open-VLA simulation experiments.

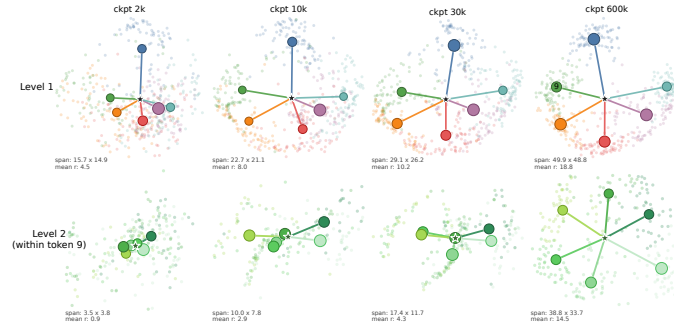
LIBERO-SPATIAL



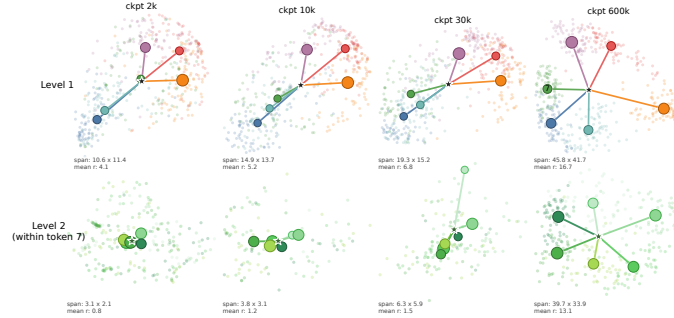
LIBERO-OBJECT



LIBERO-GOAL

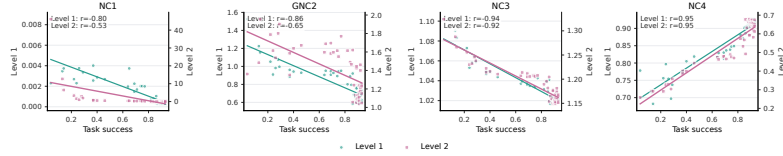


LIBERO-LONG

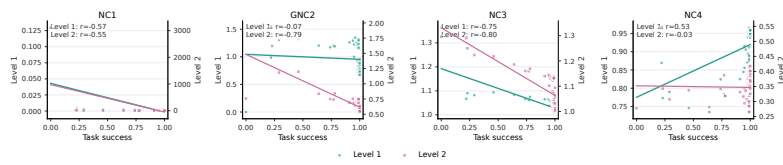


Extended Data Fig. 5 Hierarchical latent clustering in VQ-BeT across LIBERO task suites. Additional VQ-BeT visualizations show that latent representations organize from coarse to fine action-token structure during training, supporting the emergence of hierarchical action semantics beyond the representative example in the main text.

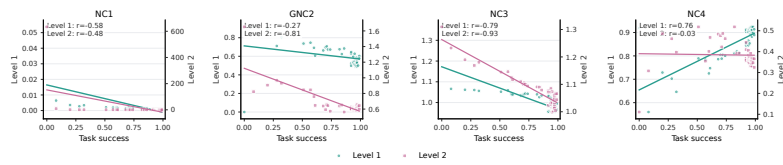
b LIBERO-SPATIAL



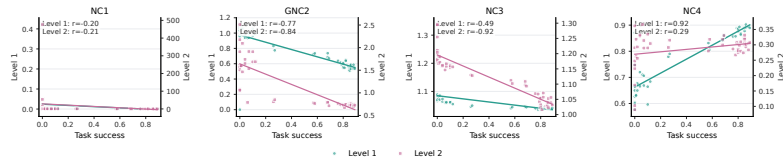
LIBERO-OBJECT



LIBERO-GOAL

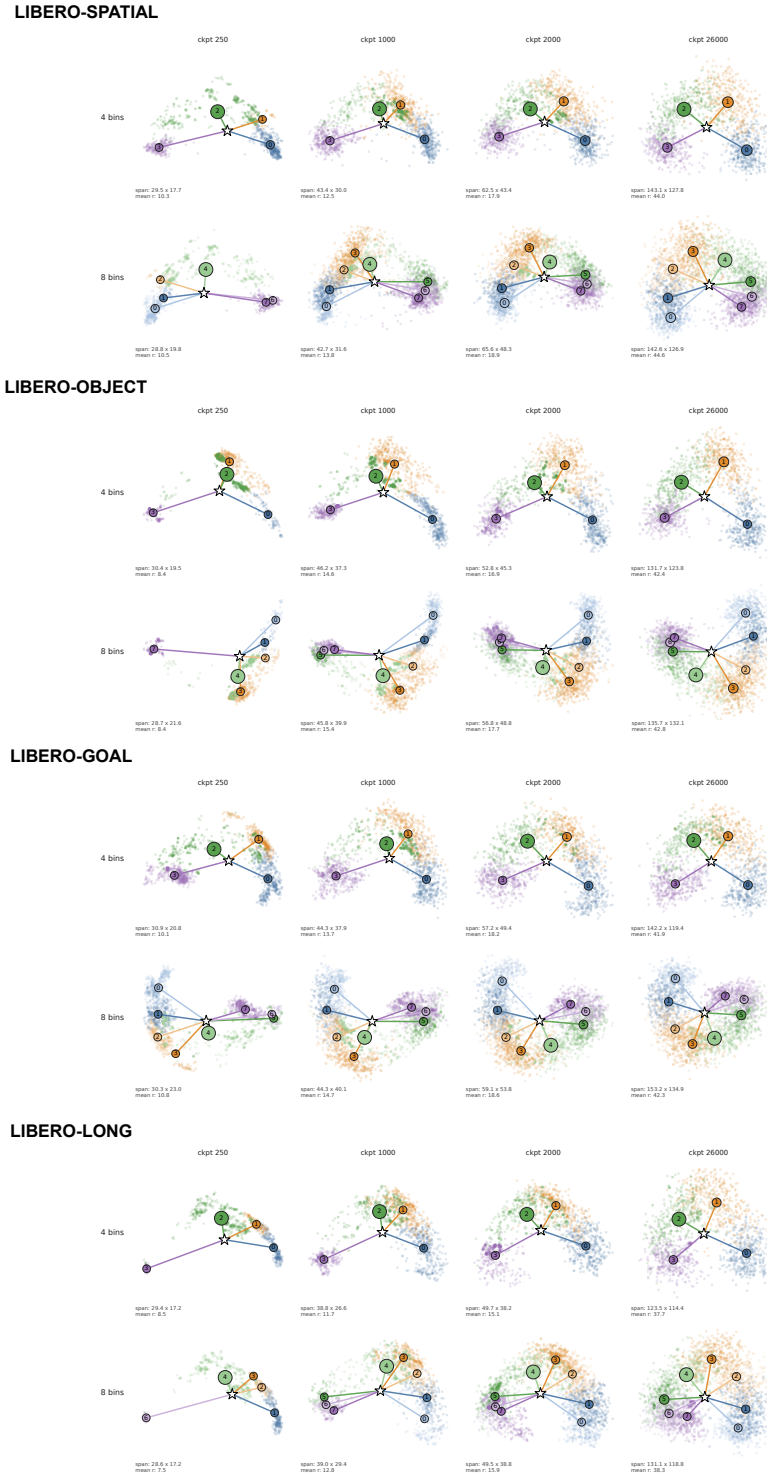


LIBERO-LONG

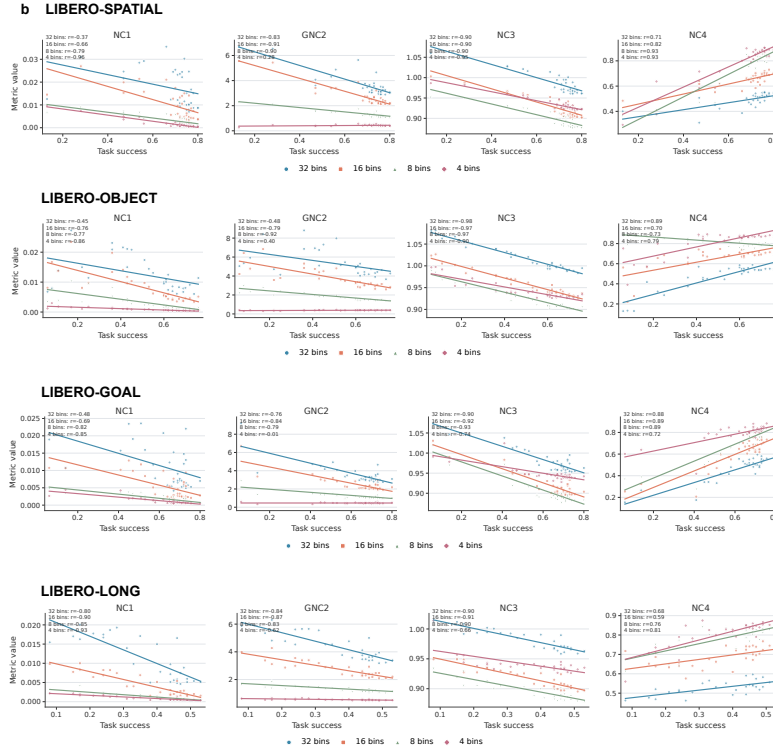


Extended Data Fig. 6 NC metrics for hierarchical VQ-BeT action tokens across LIBERO. Additional NC analyses for VQ-BeT show consistent geometry formation at both coarse and fine action-token levels across LIBERO task suites, confirming the stable emergence of two-level hierarchical latent organization during training.

1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840



Extended Data Fig. 7 Hierarchical latent clustering in OpenVLA across LIBERO task suites. Additional OpenVLA visualizations show latent clustering under different action-token grouping granularities, indicating that semantically related action bins become increasingly organized during fine-tuning.



Extended Data Fig. 8 NC metrics for grouped OpenVLA action tokens across LIBERO. Additional NC analyses for OpenVLA demonstrate consistent hierarchical geometric structure across coarse and fine token groupings, supporting the conclusion that the learned geometry captures semantic relationships among discretized action bins.