How Much of Your VLA is kNN

Denis Grachev[1] **Damien LaRocque**[1] **Omar Moured**[1,2]
[1]Sereact GmbH, Stuttgart, Germany
[2]Karlsruher Institut für Technologie, Karlsruhe, Germany
{denis.grachev,damien.larocque,omar.moured}@sereact.ai

Abstract: Vision-language-action (VLA) models now exceed 95 % success on the LIBERO simulation benchmark, yet it remains unclear how much of this performance requires generalization beyond the training demonstrations. In this work, we show that retrieving nearby demonstrations and reusing their action chunks achieves a success rate of 91.7 % on LIBERO without policy training. Under LIBERO-Plus perturbations, retrieval shows the smallest drop of all evaluated policies. Across nine publicly available VLAs, fine-tuning mostly makes the vision-language backbone a better retrieval key, and the actions of the VLAs remain close to the demonstrations. In addition, we introduce Local Action Consistency (LAC), an offline metric that measures whether nearby observations in representation space are also associated with similar future actions. We show that LAC tracks the success rate of retrieval and can guide the training of a retrieval key. Our evaluation and source code are available at https://github.com/sereact/lerobot_knn.

Keywords: Vision-language-action models, robustness, generalization, retrieval

1 Introduction

Modern vision-language-action (VLA) models typically combine a large pretrained vision-language backbone with an action-generation head (e.g., regression, diffusion, or flow matching). These models are further pretrained on large, heterogeneous collections of robot trajectories and subsequently fine-tuned for downstream manipulation tasks [1]. Their performance is commonly evaluated on standardized benchmarks, which measure closed-loop task success across predefined tasks and initial conditions.

Among these, the LIfelong learning BEnchmark on RObot manipulation tasks (LIBERO) [2] is one of the most widely used simulation benchmarks, consisting of 130 language-conditioned Franka manipulation tasks across five suites: LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, with 10 tasks each, and LIBERO-90 and LIBERO-Long, with 90 and 10 tasks, respectively. Each task comes with 50 demonstrations. VLAs are commonly fine-tuned on these demonstrations and evaluated in closed loop from a fixed set of benchmark initial states.

Recent progress has pushed LIBERO close to saturation, with several modern VLAs reporting success rates around or above 97 % [3, 1, 4], making further gains increasingly difficult to interpret as improved generalization. Yet, recent studies show that policy performance can degrade substantially under changes in viewpoint, robot initialization, language, appearance, or object layout [5]. These findings raise the question of how much benchmark success reflects generalization beyond the demonstrations, and how much can be achieved through demonstration retrieval alone.

To quantify how much benchmark performance can be explained by retrieval, we propose a training-free k-nearest neighbors (k-NN) approach that retrieves demonstration states similar to the current observation and combines the action chunks that follow them, as shown in Fig. 1. Our contributions are as follows:

10th Conference on Robot Learning (CoRL 2026), Austin TX, USA.

1. Quantify the extent to which benchmark performance can be explained by retrieval alone.
2. Use retrieval to evaluate the backbones of nine public VLAs before and after fine-tuning, and compare them with the full policies.
3. Introduce Local Action Consistency (LAC), an offline metric of embedding quality for retrieval, and use it to guide the training of a retrieval key.

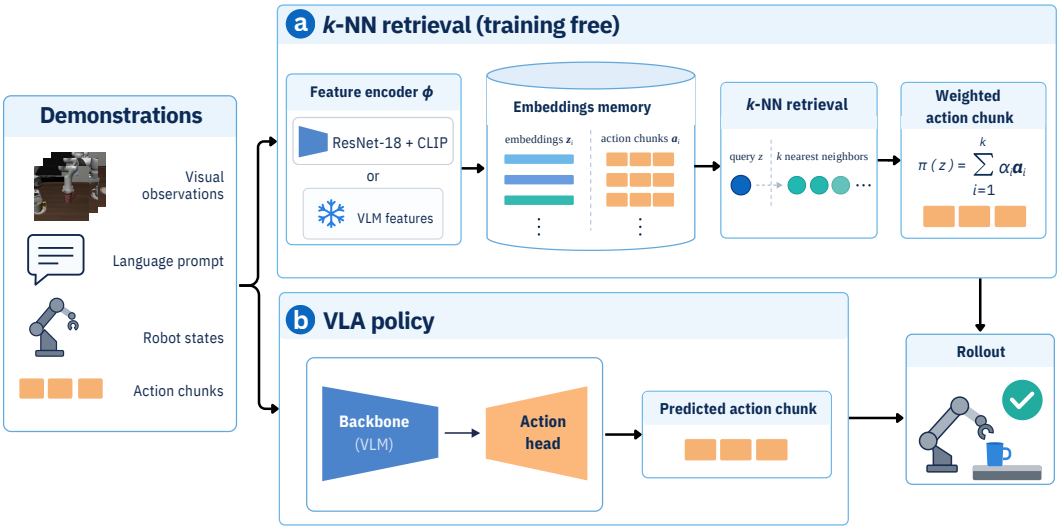


Figure 1: Overview of our evaluation pipeline. Both are evaluated with the same rollout protocol. (a) a k-NN retrieves neighboring demonstration embeddings and combines their associated action chunks; (b) the VLA predicts an action chunk directly through its backbone and action head.

2 Related works

Retrieval in Robot Learning. Retrieval in robot learning has mainly been used for data selection, in-context adaptation, and non-parametric control. For data selection, Behavior Retrieval [6] retrieves task-relevant trajectories from large offline datasets, FlowRetrieval [7] selects demonstrations using optical-flow-based motion similarity, and STRAP [8] retrieves reusable sub-trajectories from offline data to augment downstream policy training. A second line of work uses retrieval at inference time: REGENT [9] conditions a learned agent on retrieved experience, RICL [10] retrieves demonstrations as context for a pretrained VLA, and R+X [11] retrieves relevant behaviors from human videos for in-context imitation. In these approaches, retrieval primarily provides additional context to a learned policy, rather than producing actions. Closer to our setting, VINN [12] showed that nearest-neighbor regression over learned visual features can itself provide a competitive imitation policy, while He et al. [13] showed that Diffusion Policy often generates action chunks close to training trajectories and proposed an explicit lookup-table alternative. These works show that retrieval can either replace learned control or help explain its behavior.

Unlike most of these methods, our lookup requires no additional training. It uses frozen, off-the-shelf encoders or the vision-language backbone of an existing VLA, without updating their parameters on the benchmark. This allows us to evaluate how well a simple retrieval-based method performs without learning an action policy. When using the vision-language model (VLM) backbone of a VLA, we can also compare retrieval-based action selection with the VLA's learned policy using the same underlying visual representations.

VLA Generalization and Memorization. Recent work has questioned whether strong in-distribution VLA performance translates to robust generalization. LIBERO-Plus [5] shows that models exceeding 95 % of success on standard LIBERO can degrade sharply under controlled changes in

camera viewpoint, robot state, language, appearance, and layout, while LIBERO-PRO [14] further shows that some policies continue to execute memorized action patterns despite changes to objects or instructions. A related line of work studies how VLA training affects generalization. These studies indicate that supervised fine-tuning can remain brittle under distribution shift [15], while OTTER [16] and Knowledge Insulation [17] showed that adapting pretrained VLMs can overwrite useful semantics. Our work complements these studies by directly measuring how action-relevant retrieval quality in the VLM backbone changes before and after VLA fine-tuning.

3 Methodology

We use a k-NN directly as a training-free policy. Offline, each demonstration observation is embedded together with the action chunk that follows it. At inference time, the current observation is embedded in the same space, its nearest demonstrated states are retrieved, and their associated action chunks are combined to produce the next action.

3.1 Memory

Let $\mathcal{D} = \{\tau_m\}_{m=1}^{M}$ denote the set of M demonstration trajectories. From these demonstrations, we construct a retrieval memory

$$\mathcal{M} = \{(o_i, \mathbf{a}_{t_i:t_i+H}, m_i)\}_{i=1}^{N}, \tag{1}$$

where each entry contains a stored observation o_i, the action chunk of horizon H that follows it, and the index m_i of its source trajectory. We store every second frame from the 1693 LIBERO demonstrations [2], resulting in approximately 137k memory entries.

3.2 Retrieval Distance

Given a query observation q and memory entry i, we compute a weighted distance $d(q, i)$ across the available modalities:

$$d(q, i) = \sum_r \frac{w_r}{s_r} d_r(q, i), \tag{2}$$

where d_r is the distance for modality r, w_r controls its contribution, and s_r normalizes its scale.

The distance combines visual observations, robot state, language instruction, episode time, and recent observation history. Visual and language terms are computed in their respective embedding spaces, while the exact representations and normalization are described in Section 5.

3.3 Neighbor selection and action prediction

Let $\mathcal{N}_k(q)$ denote the k memory entries with smallest retrieval distance to q. To avoid overreliance on a single demonstration trajectory, we restrict the set to at most one entry per demonstration, such that

$$m_i \neq m_j, \qquad \forall i \neq j,\ i, j \in \mathcal{N}_k(q). \tag{3}$$

Each neighbor is associated with the action chunk that followed it in the demonstration. We assign larger weights to closer neighbors using a softmax:

$$\alpha_i = \frac{\exp\left(-d(q, i)/\tau\right)}{\sum_{j \in \mathcal{N}_k(q)} \exp\left(-d(q, j)/\tau\right)}, \tag{4}$$

where τ is the softmax temperature.

The predicted action chunk is then

$$\pi(q) = \sum_{i \in \mathcal{N}_k(q)} \alpha_i \mathbf{a}_{t_i:t_i+H}. \tag{5}$$

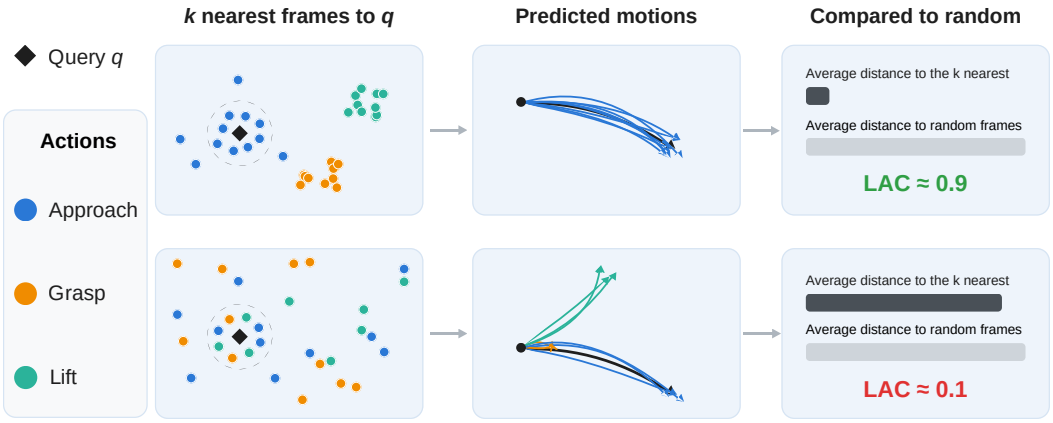


Figure 2: Local Action Consistency (LAC) is the ratio between the average action distance of the retrieved k nearest neighbors and the average action distance over the same-task set.

3.4 Local Action Consistency (LAC)

We propose Local Action Consistency (LAC), an offline metric for evaluating retrieval representations without rollouts, as illustrated in Fig. 2. For each held-out query q, LAC measures whether its nearest neighbors in representation space also exhibit similar future action chunks, relative to other demonstrations from the same task, such that

$$\mathrm{LAC@}k(q) = 1 - \frac{\frac{1}{k}\sum_{i \in \mathcal{N}_k(q)} \|\mathbf{a}_q - \mathbf{a}_i\|_2^2}{\frac{1}{|\mathcal{C}(q)|}\sum_{j \in \mathcal{C}(q)} \|\mathbf{a}_q - \mathbf{a}_j\|_2^2}, \tag{6}$$

where $\mathcal{N}_k(q)$ contains the k nearest neighbors and $\mathcal{C}(q)$ contains all memory entries from the same task as the query q. We report LAC@8 averaged over held-out queries. Higher LAC indicates that nearby representations also correspond to similar future actions relative to typical same-task matches.

4 Experimental Setup

For LIBERO [2], we evaluate on the Spatial, Object, Goal, and Long suites, comprising 40 tasks, and use 1693 training demonstrations. Following the official protocol, we run 50 episodes per task, yielding 2000 rollouts per policy. To assess robustness, we use LIBERO-Plus [5], which introduces seven categories of perturbations to the same tasks. We select a fixed subset of 420 variants, with 60 variants per category, shared across all policies and held out from training and hyperparameter tuning. We evaluate nine publicly available LIBERO VLA checkpoints from LeRobot [18], using each model as released with its corresponding observation, action, and inference conventions. Model-specific settings are provided in Appendix B.

5 Implementation Details

kNN representation. We encode each camera view using a frozen ImageNet-pretrained ResNet-18 [19], extracting a 4×4 spatial feature grid from layer 3. Robot proprioception is standardized using statistics from the demonstration memory, while language instructions are represented by unit-normalized CLIP embeddings [20]. Retrieval combines distances over visual features, robot state, language, episode time, and observation history, following Equation (2). Each term is normalized by its median within-task distance so that modalities with different numerical scales remain comparable.

Hyperparameters. We tune the k-NN hyperparameters with Optuna [21]. The complete search space, normalization variants, history terms, and ablations are reported in Appendix A.

Policy variants. We evaluate three variants of the k-NN policy, which differ in the retrieval distance, the visual representation, and the execution of action chunks. In all variants, executed actions are weighted combinations of demonstrated actions. *k-NN (ResNet)* uses only the image distance, restricts retrieval to the demonstrations of the instructed task, and executes the full weighted action chunk of its $k = 8$ nearest entries before retrieving again. *k-NN (multimodal)* retrieves from all demonstrations with the full distance of Equation (2), which combines image features, robot state, language, episode time, and observation history, with $k = 13$. It additionally applies mode seeking, which down-weights neighbors outside the dominant action mode, and temporal ensembling [22], which re-plans at every control step and combines the overlapping predictions. *k-NN (LAC key)* uses the same retrieval and execution, but applies a learned linear projection to the standardized ResNet features of each grid cell, trained with a differentiable surrogate of LAC (Equation (6)): the chunk of a frame is predicted from the softmax-weighted chunks of its neighbors in other demonstrations.

VLA retrieval features. To probe the representations of each VLA, we run the *ResNet* variant with the hidden states of the image tokens at one VLM layer in place of the ResNet-18 features. These states are pooled to a 2×2 grid per camera and standardized per dimension. The layer maximizes the R^2 of the lookup's action predictions on held-out demonstrations, averaged over the backbone before and after fine-tuning, so that both checkpoints use the same layer.

6 Results

We use retrieval to measure how much of VLA performance can be recovered directly from demonstrations, and how closely VLAs remain tied to them. We first compare k-NN policies with nine public VLAs on LIBERO and under the perturbations of LIBERO-Plus, and identify the components that drive retrieval performance. We then examine how LIBERO fine-tuning changes the VLM backbone as a retrieval key, and how closely the actions of the VLAs follow the demonstrations. Finally, we evaluate whether LAC can estimate the quality of a retrieval key without rollouts.

6.1 Benchmark Results

Table 1 compares the success rates of the three k-NN variants defined in Section 5 with nine vision-language-action (VLA) models on LIBERO and LIBERO-Plus. On average, the *ResNet* variant achieves a success rate of 88.7 % on LIBERO and 75.5 % on LIBERO-Plus. The *multimodal* variant reaches 91.7 % and 82.4 %, respectively, while the *LAC key* variant reaches 93.0 % and 81.7 %. Although $\pi_{0.5}$ achieves the highest success rate on both benchmarks, the k-NN policies perform on par with π_0-FAST on LIBERO and outperform SmolVLA and π_0. These results show that retrieval from demonstrations alone can recover a large fraction of benchmark performance, without learning a control policy.

VLA-JEPA, Evo-1, and X-VLA exceed 94 % on LIBERO, but their success rates on LIBERO-Plus fall below that of the training-free *ResNet* variant. Under perturbation, the *multimodal* k-NN outperforms 7 of the 9 VLAs and shows the smallest performance drop of all evaluated policies (9.3 %pt). These results highlight that strong performance on LIBERO does not necessarily translate to robustness under the perturbations of LIBERO-Plus.

6.2 Ablation study

To identify which components drive retrieval performance, we ablate the *multimodal* k-NN policy one component at a time, as shown by Table 3 in Appendix A.5. Using only 10 % of the demonstrations reduces success from 91.7 % to 77.4 %, and using raw pixels instead of embeddings lowers it to 57.2 %. Ignoring the language prompt causes a drop of 27.1 %pt, whereas removing mode seeking, temporal ensembling, history, robot state, or the time step changes success by at most 1 %pt,

Table 1: Success rate (%) on LIBERO (official protocol, 2000 episodes) and on the same 420 LIBERO-Plus variants. VLAs are ordered by average on LIBERO-Plus. Δ is the difference between the average success rates on LIBERO-Plus and LIBERO, computed before rounding. The *k*-NN rows are training-free, except for the LAC key.

	LIBERO					LIBERO-Plus					
	avg	Spat	Obj	Goal	Long	avg	Spat	Obj	Goal	Long	Δ
$\pi_{0.5}$ [1]	**97.7**	97.0	99.8	**97.8**	**96.2**	**85.5**	**90.5**	83.8	82.9	**84.8**	-12.2
MolmoAct2 [23]	97.5	**97.4**	99.4	97.6	95.6	82.6	89.5	81.9	76.2	82.9	-14.9
GR00T N1.7 [24]	94.9	95.8	98.2	94.6	90.8	80.2	89.5	74.3	79.0	78.1	-14.6
VLA-JEPA [4]	97.2	95.4	**100.0**	97.4	95.8	74.0	81.0	70.5	77.1	67.6	-23.1
Evo-1 [25]	95.4	92.0	98.6	97.4	93.6	71.4	74.3	73.3	69.5	68.6	-24.0
X-VLA [26]	94.5	94.6	98.6	92.6	92.2	64.5	71.4	64.8	64.8	57.1	-30.0
π_0 [27]	71.9	70.0	86.4	85.8	45.2	42.9	54.3	41.9	46.7	28.6	-29.0
SmolVLA [28]	82.2	78.8	92.2	88.0	69.6	27.9	30.5	49.5	11.4	20.0	-54.3
π_0-FAST [29]	90.6	90.4	93.8	90.6	87.4	25.7	39.0	34.3	16.2	13.3	-64.8
***k*-NN** (*ResNet*)	88.7	90.4	95.0	95.2	74.2	75.5	77.1	81.9	79.0	63.8	-13.2
***k*-NN** (*multimodal*)	91.7	88.0	99.8	97.6	81.4	82.4	79.0	**85.7**	**86.7**	78.1	**-9.3**
***k*-NN** (*LAC key*)	93.0	90.6	99.6	97.0	84.8	81.7	87.6	83.8	81.0	74.3	-11.3

Table 2: Lookup success using each VLA's VLM backbone before and after LIBERO fine-tuning, compared with the full VLA. **Bold** indicates cases where retrieval on the fine-tuned backbone outperforms the corresponding VLA. *GR00T freezes its VLM, so before = after.

	LIBERO			LIBERO-Plus		
	***k*-NN** before	***k*-NN** after	VLA	***k*-NN** before	***k*-NN** after	VLA
$\pi_{0.5}$	85.4	81.8	97.7	60.5	65.7	85.5
MolmoAct2	78.6	91.4	97.5	52.6	81.0	82.6
GR00T N1.7*	87.6	87.6	94.9	66.4	66.4	80.2
VLA-JEPA	84.0	83.0	97.2	49.3	65.7	74.0
Evo-1	62.1	87.1	95.4	35.5	**72.6**	71.4
X-VLA	83.3	88.6	94.5	62.4	**78.1**	64.5
π_0	86.9	17.9	71.9	56.7	2.4	42.9
SmolVLA	76.3	**86.5**	82.2	45.0	**69.3**	27.9
π_0-FAST	86.6	**95.0**	90.6	58.3	**74.0**	25.7

within the 95 % confidence interval of $\pm 1.2\,\%\mathrm{pt}$ over 2000 episodes. For k between 4 and 32, success remains between 89.4 % and 92.7 %, and drops to 78.6 % when only the nearest demonstration is replayed ($k = 1$). Overall, the visual representation, the language prompt, and the number of demonstrations account for most of the performance, while the remaining refinements contribute less. Full ablations and hyperparameter studies are reported in Appendix A.

6.3 Representation Analysis

Table 2 shows that LIBERO fine-tuning generally makes the VLM backbone a better retrieval key, mostly under perturbation. Retrieval with the fine-tuned backbone improves for 5 of the 8 fine-tuned VLAs on LIBERO, and for 7 of 8 on LIBERO-Plus, by 5 to 37 %pt. For these VLAs, the drop from LIBERO to LIBERO-Plus also decreases, from 20.9–34.7 %pt before fine-tuning to 10.4–21.0 %pt after. However, the effect is not universal: after fine-tuning, retrieval with the π_0 backbone drops from 86.9 % to 17.9 % on LIBERO.

For some models, retrieval on the fine-tuned backbone alone outperforms the full VLA: for 2 of 9 models on LIBERO and 4 of 9 on LIBERO-Plus (Table 2), by up to 48 %pt. For these models, the results suggest that the action head does not exploit all the action-relevant information encoded in the backbone.

Changes in Local Representation Structure. To understand how fine-tuning affects the geometry of the representations, Fig. 3 shows t-distributed stochastic neighbor embedding (t-SNE) maps [30] of the retrieval keys of four VLAs before and after fine-tuning, and of the ImageNet ResNet-18 before and after LAC training. After fine-tuning, the keys of Evo-1 form trajectories that follow task progress, and the median progress gap between a frame and its nearest same-task neighbor from another demonstration decreases from 0.031 to 0.022. Across all backbones (Fig. 8), this gap decreases for 7 of the 8 fine-tuned VLAs, but the change is small for $\pi_{0.5}$ and MolmoAct2. π_0 is different: its fine-tuned key collapses into one cluster per task, and its nearest neighbors remain of the same task, but its progress gap increases from 0.028 to 0.052, consistent with the collapse of its retrieval performance. After fine-tuning, the main directions of variation of the keys also depend less on the LIBERO suite, and thus on the scene, for all 8 backbones (Fig. 9).

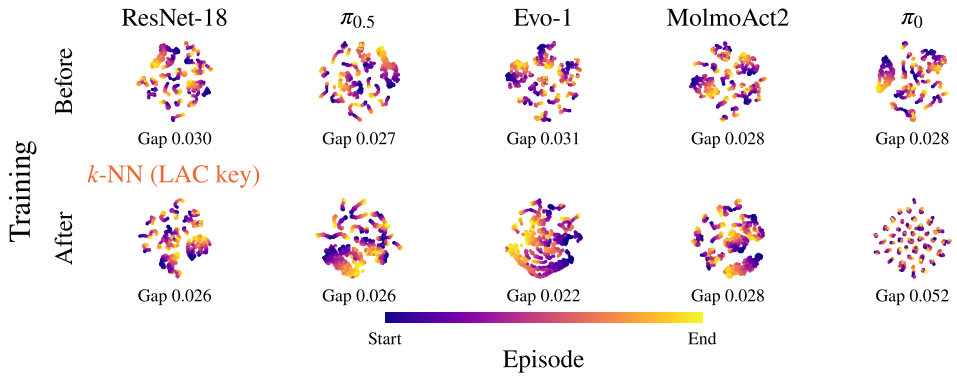


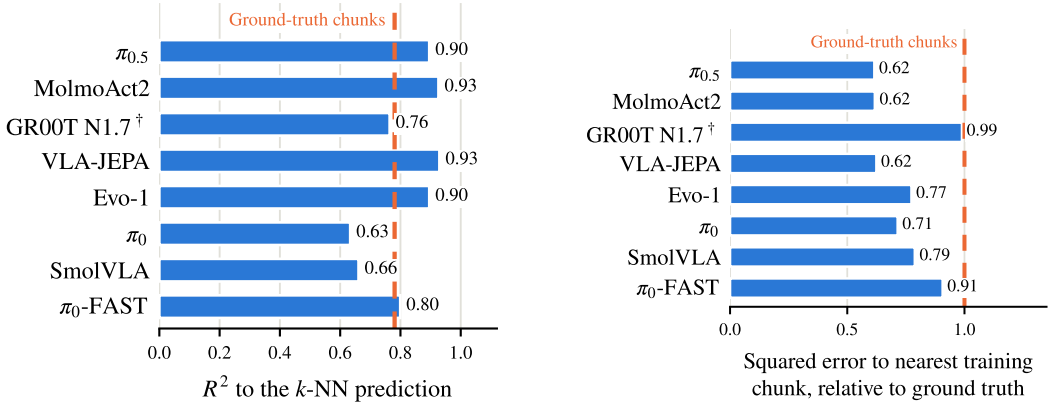
Figure 3: t-SNE maps of retrieval keys for the same 4,150 demonstration frames from LIBERO. Top: before LIBERO fine-tuning (ImageNet ResNet-18 for the first column). Bottom: after fine-tuning (LAC-trained key for the first column). Colors indicate normalized episode progress. Below each map: median progress gap to the nearest frame from another demonstration of the same task.

6.4 Action-Space Analysis

We next examine how closely the actions produced by the VLAs follow the demonstrations. At each state visited in the LIBERO rollouts of a VLA, we compare its action chunk with the prediction of the *ResNet* k-NN variant, the weighted mean of the chunks of its $k = 8$ nearest demonstration frames of the same task. As a reference, the same prediction matches held-out ground-truth chunks with $R^2 = 0.78$. For the VLAs, R^2 ranges from 0.63 to 0.93 (Fig. 4a): the chunks of 5 of the 8 VLAs are closer to the k-NN prediction than the ground-truth chunks are. In addition, the chunks of all VLAs lie closer to a training chunk of the same task than held-out ground-truth chunks do, with a median squared error of 0.62-0.99 times the ground-truth reference (Fig. 4b). Together, these results suggest that, on LIBERO, the actions of the VLAs remain close to the demonstrated actions.

6.5 LAC as an Offline Diagnostic

The previous analyses evaluate retrieval representations through closed-loop execution; we next ask whether LAC (Section 3.4) can estimate their quality from the demonstrations alone. Excluding the fine-tuned VLM backbones, which were trained on the held-out demonstrations, LAC tracks closed-loop retrieval success (Pearson $r = 0.78$) but does not reliably rank keys of similar quality (Spearman $\rho = 0.61$, Fig. 10 in Appendix E). It also flags the degraded fine-tuned π_0 backbone, with an LAC of 0.55 against 0.76-0.88 for all other keys. A linear projection of the ResNet features, trained with a differentiable surrogate of LAC, raises the success rate of a plain k-NN lookup from 88.7 % to 93.0 %.



(a) Agreement with the k-NN prediction.

(b) Squared error to the nearest training chunk.

Figure 4: Comparison of the action chunks of each VLA, in its LIBERO rollouts, with the training demonstrations. (a) R^2 between the VLA chunks and the prediction of the *ResNet* k-NN variant; higher indicates closer agreement. (b) Median squared error to the nearest training chunk of the same task, relative to held-out ground-truth chunks; lower indicates greater proximity to the training data. Dashed lines: held-out ground-truth chunks. †Evaluated on LIBERO-Spatial only. X-VLA is excluded because it predicts absolute poses.

7 Conclusion

In this work, we investigated how much of the performance observed on the LIfelong learning BEnchmark on RObot manipulation tasks (LIBERO) can be recovered directly from demonstrations, without learning a control policy. To this end, we used retrieval-based policies not only as baselines, but also as tools to analyze the representations and actions of existing VLAs. First, retrieval alone recovers a substantial fraction of benchmark performance: a simple k-NN policy is competitive with several VLAs on LIBERO, and its multimodal variant shows the smallest drop of all evaluated policies under the perturbations of LIBERO-Plus. The actions of the VLAs also remain close to the demonstrations, suggesting that, on dense benchmarks such as LIBERO, a large part of successful behavior can be recovered by matching test states to demonstrated ones and reusing their actions. Second, retrieval allowed us to separate the contributions of the VLM backbone and of the action head of a VLA. Across nine VLAs, the backbone already contains action-relevant structure, and fine-tuning mostly makes it a more robust retrieval key, improving retrieval under the perturbations of LIBERO-Plus for 7 of 8 models. This effect is not universal, as some representations degrade after fine-tuning. For some models, retrieval from the fine-tuned backbone alone outperforms the full policy, suggesting that their action head does not exploit all the action-relevant information in the representation. Third, we introduced LAC, an offline metric that tracks closed-loop retrieval success, although it does not reliably rank representations of similar quality. It identifies degraded retrieval keys, and a differentiable surrogate of LAC can be used to train a better retrieval key. Overall, our results suggest that retrieval baselines should be reported alongside VLA performance to distinguish benchmark memorization from generalization.

These conclusions remain limited to LIBERO and related benchmarks. Their relatively dense demonstration coverage may favor retrieval-based methods. LIBERO-Plus perturbations also largely preserve the underlying task and solution trajectories, so our robustness results mainly concern changes in observations and instructions, rather than situations that require new trajectories. In addition, we evaluated the public checkpoints of each VLA as released, and some may perform differently from their published results. More fundamentally, retrieval cannot extrapolate beyond the support of its memory, and the strong performance observed on these benchmarks should not be interpreted as evidence that k-NN policies can generalize broadly across robotic problems.

Future work should extend this analysis to other robot-learning benchmarks, especially settings with sparser demonstrations, novel object placements [14], and tasks requiring behavior outside the demonstrated support. Such evaluations would help determine how much of the observed behavior is specific to LIBERO and how much reflects more general properties of current VLA policies.

Acknowledgments

The authors thank the LeRobot community for developing and maintaining the open-source framework used to evaluate the VLA policies, and the authors of the evaluated models and benchmarks for releasing their checkpoints and data. This work was supported by Sereact GmbH.

References

[1] K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. R. Equi, C. Finn, N. Fusai, M. Y. Galliker, D. Ghosh, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, D. LeBlanc, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, A. Z. Ren, L. X. Shi, L. Smith, J. T. Springenberg, K. Stachowicz, J. Tanner, Q. Vuong, H. Walke, A. Walling, H. Wang, L. Yu, and U. Zhilinsky. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. In J. Lim, S. Song, and H.-W. Park, editors, *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 17–40. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/black25a.html.

[2] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. In *Advances in Neural Information Processing Systems 36*, NeurIPS 2023, pages 44776–44791. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2023. doi:10.52202/075280-1939. URL http://dx.doi.org/10.52202/075280-1939.

[3] M. J. Kim, C. Finn, and P. Liang. Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success. In *Robotics: Science and Systems XXI*, RSS2025. Robotics: Science and Systems Foundation, June 2025. doi:10.15607/rss.2025.xxi.017. URL http://dx.doi.org/10.15607/rss.2025.xxi.017.

[4] J. Sun, W. Zhang, Z. Qi, S. Ren, Z. Liu, H. Zhu, G. Sun, X. Jin, and Z. Chen. VLA-JEPA: Enhancing Vision-Language-Action Model with Latent World Model. In *Computer Vision – ECCV 2026*, pages 478–497. Springer Nature Switzerland, 2026. ISBN 9783032371966. doi:10.1007/978-3-032-37196-6_26. URL http://dx.doi.org/10.1007/978-3-032-37196-6_26.

[5] S. Fei, S. Wang, J. Shi, Z. Dai, J. Cai, P. Qian, L. Ji, X. He, S. Zhang, Z. Fei, J. Fu, J. Gong, and X. Qiu. LIBERO-Plus: A Progressive Robustness Benchmark for Visual-Language-Action Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 38574–38583, June 2026.

[6] M. Du, S. Nair, D. Sadigh, and C. Finn. Behavior Retrieval: Few-Shot Imitation Learning by Querying Unlabeled Datasets. In *Robotics: Science and Systems XIX*, RSS2023. Robotics: Science and Systems Foundation, July 2023. doi:10.15607/rss.2023.xix.011. URL http://dx.doi.org/10.15607/RSS.2023.XIX.011.

[7] L.-H. Lin, Y. Cui, A. Xie, T. Hua, and D. Sadigh. FlowRetrieval: Flow-Guided Data Retrieval for Few-Shot Imitation Learning. In P. Agrawal, O. Kroemer, and W. Burgard, editors, *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 4084–4099. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/lin25a.html.

[8] M. Memmel, J. Berg, B. Chen, A. Gupta, and J. Francis. STRAP: Robot Sub-Trajectory Retrieval for Augmented Policy Learning. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 63813–63832, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/a06e129e01e0d2ef853e9ff67b911360-Paper-Conference.pdf.

[9] K. Sridhar, S. Dutta, D. Jayaraman, and I. Lee. REGENT: A Retrieval-Augmented Generalist Agent That Can Act In-Context in New Environments. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 39385–39414, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/6237474e5a1c58a193af179b2434a86e-Paper-Conference.pdf.

[10] K. Sridhar, S. Dutta, D. Jayaraman, and I. Lee. RICL: Adding In-Context Adaptability to Pre-Trained Vision-Language-Action Models. In J. Lim, S. Song, and H.-W. Park, editors, *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 5022–5038. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/sridhar25a.html.

[11] G. Papagiannis, N. Di Palo, P. Vitiello, and E. Johns. R+X: Retrieval and Execution from Everyday Human Videos. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8284–8290. IEEE, May 2025. doi:10.1109/icra55743.2025.11128322. URL http://dx.doi.org/10.1109/ICRA55743.2025.11128322.

[12] J. Pari, N. M. M. Shafiullah, S. P. Arunachalam, and L. Pinto. The Surprising Effectiveness of Representation Learning for Visual Imitation. In *Robotics: Science and Systems XVIII*, RSS2022. Robotics: Science and Systems Foundation, June 2022. doi:10.15607/rss.2022.xviii.010. URL http://dx.doi.org/10.15607/rss.2022.xviii.010.

[13] C. He, X. Liu, G. Sznaier Camps, J. Bruno, G. Sartoretti, and M. Schwager. Demystifying Robot Diffusion Policies: Action Memorization and a Simple Lookup Table Alternative. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust, editors, *International Conference on Learning Representations*, volume 2026, pages 10890–10917, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/122ea6470232ee5e79a2649243348005-Paper-Conference.pdf.

[14] X. Zhou, Y. Xu, G. Tie, Y. Chen, G. Zhang, D. Chu, P. Zhou, and L. Sun. LIBERO-PRO: Towards Robust and Fair Evaluation of Vision-Language-Action Models Beyond Memorization. *arXiv preprint arXiv:2510.03827*, 2025.

[15] J. Liu, F. Gao, B. Wei, X. Chen, Q. Liao, Y. Wu, C. Yu, and Y. Wang. What Can RL Bring to VLA Generalization? An Empirical Study. In *Advances in Neural Information Processing Systems 38*, NeurIPS 2025, pages 107724–107754. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2025. doi:10.52202/085713-3248. URL http://dx.doi.org/10.52202/085713-3248.

[16] H. Huang, F. Liu, L. Fu, T. Wu, M. Mukadam, J. Malik, K. Goldberg, and P. Abbeel. OTTER: A Vision-Language-Action Model with Text-Aware Visual Feature Extraction. In A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 25566–25580. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/huang25u.html.

[17] D. Driess, J. T. Springenberg, B. Ichter, L. Yu, A. Li-Bell, K. Pertsch, A. Z. Ren, H. Walke, Q. Vuong, L. X. Shi, and S. Levine. Knowledge Insulating Vision-Language-Action Models: Train Fast, Run Fast, Generalize Better. In *Advances in Neural Information Processing Systems 38*, NeurIPS 2025, pages 113969–113990. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2025. doi:10.52202/085713-3439. URL http://dx.doi.org/10.52202/085713-3439.

[18] R. Cadene, S. Alibert, F. Capuano, M. Aractingi, A. Zouitine, P. Kooijmans, J. Choghari, M. Russi, C. Pascal, S. Palma, D. Aubakirova, M. Shukor, J. Moss, A. Soare, Q. Lhoest, Q. Gallouédec, and T. Wolf. LeRobot: An Open-Source Library for End-to-End Robot

Learning. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust, editors, *International Conference on Learning Representations*, volume 2026, pages 122398–122417, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/c702175bc31584c52c28cc20731db371-Paper-Conference.pdf.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE, June 2016. doi:10.1109/cvpr.2016.90. URL http://dx.doi.org/10.1109/CVPR.2016.90.

[20] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

[21] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A Next-generation Hyperparameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 2623–2631. ACM, July 2019. doi:10.1145/3292500.3330701. URL http://dx.doi.org/10.1145/3292500.3330701.

[22] T. Zhao, V. Kumar, S. Levine, and C. Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Robotics: Science and Systems XIX*, RSS2023. Robotics: Science and Systems Foundation, July 2023. doi:10.15607/rss.2023.xix.016. URL http://dx.doi.org/10.15607/RSS.2023.XIX.016.

[23] H. Fang, J. Duan, D. Clay, S. Wang, S. Liu, W. Huang, X. Fan, W.-C. Tsai, S. Chen, Y. R. Wang, S. Xing, J. Cho, J. S. Park, A. Eftekhar, P. Sushko, K. Farley, A. Wadhwa, C. Harrison, W. Han, Y.-C. Lee, E. VanderBilt, R. Hendrix, S. Ellawela, L. Ngoo, J. Chai, Z. Ren, A. Farhadi, D. Fox, and R. Krishna. MolmoAct2: Action Reasoning Models for Real-world Deployment. *arXiv preprint arXiv:2605.02881*, 2026. Accepted to the Conference on Robot Learning (CoRL) 2026.

[24] NVIDIA, J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. J. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, J. Jang, Z. Jiang, J. Kautz, K. Kundalia, L. Lao, Z. Li, Z. Lin, K. Lin, G. Liu, E. Llontop, L. Magne, A. Mandlekar, A. Narayan, S. Nasiriany, S. Reed, Y. L. Tan, G. Wang, Z. Wang, J. Wang, Q. Wang, J. Xiang, Y. Xie, Y. Xu, Z. Xu, S. Ye, Z. Yu, A. Zhang, H. Zhang, Y. Zhao, R. Zheng, and Y. Zhu. GR00T N1: An Open Foundation Model for Generalist Humanoid Robots. *arXiv preprint arXiv:2503.14734*, Mar. 2025. URL https://arxiv.org/abs/2503.14734.

[25] T. Lin, Y. Zhong, Y. Du, J. Zhang, J. Liu, Y. Chen, E. Gu, Z. Liu, H. Cai, Y. Zou, L. Zou, Z. Zhou, G. Li, and B. Zhao. Evo-1: Lightweight Vision-Language-Action Model with Preserved Semantic Alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13397–13406, June 2026.

[26] J. Zheng, J. Li, Z. Wang, D. Liu, X. Kang, Y. Feng, Y. Zheng, J. Zou, Y. Chen, J. Zeng, T. Wang, Y.-Q. Zhang, J. Liu, and X. Zhan. X-VLA: Soft-Prompted Transformer as Scalable Cross-Embodiment Vision-Language-Action Model. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust, editors, *International Conference on Learning Representations*, volume 2026, pages 60580–60606, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/630d293833e09e1ecd892a898a20b074-Paper-Conference.pdf.

[27] K. Black, N. Brown, D. Driess, A. Esmail, M. R. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, L. Smith, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Robotics: Science and Systems XXI*, RSS2025. Robotics: Science and Systems Foundation, June 2025. doi: 10.15607/rss.2025.xxi.010. URL http://dx.doi.org/10.15607/RSS.2025.XXI.010.

[28] M. Shukor, D. Aubakirova, F. Capuano, P. Kooijmans, S. Palma, A. Zouitine, M. Aractingi, C. Pascal, M. Russi, A. Marafioti, S. Alibert, M. Cord, T. Wolf, and R. Cadene. SmolVLA: A Vision-Language-Action Model for Affordable and Efficient Robotics. *arXiv preprint arXiv:2506.01844*, 2025.

[29] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine. FAST: Efficient Action Tokenization for Vision-Language-Action Models. In *Robotics: Science and Systems XXI*, RSS2025. Robotics: Science and Systems Foundation, June 2025. doi:10.15607/rss.2025.xxi.012. URL http://dx.doi.org/10.15607/RSS.2025.XXI.012.

[30] L. van der Maaten and G. Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

[31] F. Hutter, H. Hoos, and K. Leyton-Brown. An Efficient Approach for Assessing Hyperparameter Importance. In E. P. Xing and T. Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 754–762, Beijing, China, 22–24 Jun 2014. PMLR. URL https://proceedings.mlr.press/v32/hutter14.html.

[32] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jégou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

[33] L. Beyer, A. Steiner, A. S. Pinto, A. Kolesnikov, X. Wang, D. Salz, M. Neumann, I. Alabdulmohsin, M. Tschannen, E. Bugliarello, T. Unterthiner, D. Keysers, S. Koppula, F. Liu, A. Grycner, A. Gritsenko, N. Houlsby, M. Kumar, K. Rong, J. Eisenschlos, R. Kabra, M. Bauer, M. Bošnjak, X. Chen, M. Minderer, P. Voigtlaender, I. Bica, I. Balazevic, J. Puigcerver, P. Papalampidi, O. Henaff, X. Xiong, R. Soricut, J. Harmsen, and X. Zhai. PaliGemma: A versatile 3B VLM for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

[34] B. Xiao, H. Wu, W. Xu, X. Dai, H. Hu, Y. Lu, M. Zeng, C. Liu, and L. Yuan. Florence-2: Advancing a Unified Representation for a Variety of Vision Tasks. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4818–4829. IEEE, June 2024. doi: 10.1109/cvpr52733.2024.00461. URL http://dx.doi.org/10.1109/CVPR52733.2024.00461.

A Additional details of the *k*-NN policy

This appendix provides the implementation details and ablations of the *k*-NN policy introduced in Section 3. We use the same notation throughout.

A.1 Distance normalization

The modality-specific scaling factors in Equation (2) are estimated once when constructing the retrieval memory. Under a median normalization, the visual scale for camera c is

$$s_c = \underset{\substack{i,j:\tau(i)=\tau(j)\\ i\neq j}}{\text{median}} \left\|\mathbf{z}_c^i - \mathbf{z}_c^j\right\|_2^2. \tag{7}$$

Similarly, the state and temporal scales are defined as

$$s_x = \underset{\substack{i,j:\tau(i)=\tau(j)\\ i\neq j}}{\text{median}} \frac{1}{n_x}\left\|\tilde{\mathbf{x}}^i - \tilde{\mathbf{x}}^j\right\|_2^2, \tag{8}$$

and

$$s_t = \underset{\substack{i,j:\tau(i)=\tau(j)\\ i\neq j}}{\text{median}} (t_i - t_j)^2. \tag{9}$$

For language, the scale is computed over pairs of distinct task instructions:

$$s_\ell = \underset{\tau\neq\tau'}{\text{median}} \left\|\mathbf{u}_{\ell_\tau} - \mathbf{u}_{\ell_{\tau'}}\right\|_2^2. \tag{10}$$

The gripper term is not normalized by a modality-specific scale.

A.2 History matching and trajectory consistency

The full retrieval policy can incorporate recent observation history to disambiguate states that look similar but occur at different stages of a trajectory.

Let q' denote a previous query observation separated from q by Δ time steps. For a candidate memory entry i, let $i^{-\Delta}$ denote the entry Δ steps earlier in the same demonstration. We define

$$d_1(q,i) = d_0(q,i) + w_h \bar{d}_0(q', i^{-\Delta}), \tag{11}$$

where $\bar{d}_0$ is the base retrieval distance without the language term.

We optionally favor candidates that continue the trajectory selected at the previous control step. Let $(m^\star, t^\star)$ denote the expected continuation of the previous match. The final distance is

$$d(q,i) = d_1(q,i)\left(1 - w_s \mathbf{1}\left[m_i = m^\star, |t_i - t^\star| \le W\right]\right), \tag{12}$$

where W is the temporal consistency window.

A.3 Neighbor weighting

In the implementation, the softmax temperature is scaled by the mean distance of the retrieved neighborhood. Let

$$\bar{d} = \frac{1}{k}\sum_{j\in\mathcal{N}_k(q)} d(q,j). \tag{13}$$

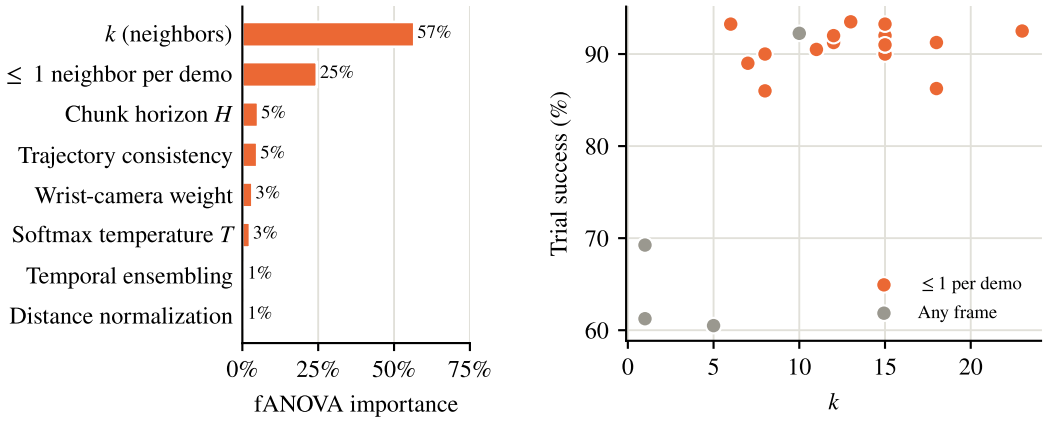
The neighbor weights are then

$$\alpha_i = \frac{\exp\left(-d(q,i)/(T\bar{d})\right)}{\sum_{j\in\mathcal{N}_k(q)}\exp\left(-d(q,j)/(T\bar{d})\right)}. \tag{14}$$

This local rescaling makes the weighting less sensitive to the absolute scale of distances across queries.

A.4 Hyperparameter tuning

We tune the retrieval hyperparameters with Optuna [21], using success rate as the optimization objective. As LIBERO provides no separate validation split, each trial is evaluated on the initial states 40–49 of every task (400 episodes), which are also part of the official evaluation protocol. On the initial states 0–39, never used during tuning, the *multimodal* variant reaches 91.6 %, against 91.7 % on all 50 states, so tuning on these states does not noticeably inflate the reported results. Figure 5a shows that performance is driven primarily by the number of neighbors k and by the retrieval restriction of at most one neighbor per demonstration as defined in Equation (3), accounting for 57 % and 25 % of the fANOVA importance [31], respectively. The remaining hyperparameters each account for about 5 % or less. Figure 5b shows that using the per-demonstration constraint and a number of neighbors k greater than 5 are indicative of high success rates.



(a) Hyperparameter importance estimated with fANOVA [31].

(b) Success rate as a function of k.

Figure 5: Hyperparameter search for the k-NN policy with Optuna, over 24 trials of 400 episodes. (a) Hyperparameter importance estimated with fANOVA [31]. (b) Success rate as a function of the number of neighbors k and of the uniqueness constraint of Equation (3).

A.5 Ablation study

Table 3 evaluates each component independently by changing one element of the *multimodal* variant at a time. The largest drops come from replacing the ResNet-18 features with raw 16×16 pixels (−34.5 %pt), ignoring the language prompt (−27.1 %pt), and reducing the demonstration memory to 10 % (−14.3 %pt). In contrast, mode seeking, temporal ensembling, history matching, proprioception, and the time-step feature each change average success by at most 1 %pt, within statistical noise. For k between 4 and 32, success remains between 89.4 % and 92.7 %, whereas replaying the nearest demonstration ($k = 1$) lowers it to 78.6 %.

B VLA evaluation protocol

We evaluated nine publicly available VLA checkpoints as released. Table 4 lists the Hugging Face repository and revision of each evaluated checkpoint. For $\pi_{0.5}$, π_0, and π_0-FAST, we use the LIBERO fine-tuned checkpoints from LeRobot [18]. X-VLA uses absolute end-effector control and its native observation convention. VLA-JEPA executes 7 actions per re-plan, while Evo-1 executes 14. For Evo-1, we use 256 px observations on both LIBERO and LIBERO-Plus, consistent with the image resolution assumed by the LIBERO-Plus perturbations. MolmoAct2 outputs continuous actions. For GR00T N1.7, we use one checkpoint per LIBERO suite and execute 8 of its 16 predicted actions before re-planning. Some reproduced success rates, notably for π_0 and SmolVLA, are below

Table 3: Ablations of the *multimodal k*-NN variant under the standard LIBERO evaluation protocol (50 rollouts per task, 2000 episodes in total). Δ denotes the change in average success relative to the multimodal variant. Raw-pixel retrieval fails on LIBERO-Object, where the ten tasks share a scene containing several small objects.

Ablation	avg	Spatial	Object	Goal	Long	Δ
None (multimodal variant)	91.7	88.0	99.8	97.6	81.4	
Prompt ignored	64.6	77.6	99.6	8.8	72.4	−27.1
Raw 16×16 pixels	57.2	79.0	0.0	91.0	58.6	−34.5
10% of demos	77.4	83.8	87.0	90.2	48.6	−14.3
25% of demos	82.6	83.6	97.8	92.6	56.4	−9.1
50% of demos	88.5	86.4	99.6	95.4	72.6	−3.2
$k = 1$ (replay nearest demo)	78.6	77.2	90.6	86.2	60.4	−13.1
DINOv2-S [32] instead of ResNet-18	83.4	79.8	97.6	95.0	61.0	−8.3
Agent-view camera only	82.1	78.0	92.8	93.6	63.8	−9.6
Wrist camera only	87.5	87.0	98.8	92.8	71.2	−4.2
Plain weighted mean	90.7	88.4	100.0	96.6	77.8	−1.0
No temporal ensembling	91.1	92.0	98.0	94.2	80.2	−0.6
No history matching	91.5	89.6	98.8	97.6	79.8	−0.2
No proprioceptive state	91.4	87.6	99.6	96.8	81.4	−0.3
No time-step feature	91.3	88.2	99.8	97.4	79.8	−0.4
Hard task filter	91.9	88.4	100.0	97.4	81.8	+0.2
$k = 2$	85.2	83.8	94.6	92.2	70.2	−6.5
$k = 4$	89.4	86.2	98.8	93.8	78.6	−2.3
$k = 8$	89.8	85.4	99.6	96.6	77.6	−1.9
$k = 24$	92.7	89.2	99.8	97.2	84.4	+1.0
$k = 32$	91.7	88.8	99.8	97.4	80.8	+0.0

Table 4: Evaluated VLA checkpoints, hosted on the Hugging Face Hub. Revisions are abbreviated commit hashes. GR00T N1.7 uses one checkpoint per LIBERO suite.

Model	Repository	Revision
$\pi_{0.5}$	`lerobot/pi05_libero_finetuned_v044`	`8e174154`
MolmoAct2	`allenai/MolmoAct2-LIBERO-LeRobot`	`f0c5a956`
GR00T N1.7	`nvidia/gr00t17-lerobot-libero_spatial-640`	`32a6ec78`
	`nvidia/gr00t17-lerobot-libero_object-640`	`1499db35`
	`nvidia/gr00t17-lerobot-libero_goal-640`	`436c57c0`
	`nvidia/gr00t17-lerobot-libero_10-640`	`5ee08ab0`
VLA-JEPA	`lerobot/VLA-JEPA-LIBERO`	`735d9f69`
Evo-1	`zuoxingdong/evo1_libero`	`515921f4`
X-VLA	`lerobot/xvla-libero`	`12e8783e`
π_0	`lerobot/pi0_libero_finetuned_v044`	`45dcc8fc`
SmolVLA	`lerobot/smolvla_libero`	`31d453f7`
π_0-FAST	`lerobot/pi0fast-libero`	`840f4b50`

those reported by their authors, which may reflect differences between the public checkpoints and those used in the original publications.

Before closed-loop evaluation, we performed a simple offline sanity check on each checkpoint. On demonstration frames, the predicted motion must be closer to the corresponding demonstrated motion than to a shuffled pairing, with an error below half of the shuffled baseline. This check is used only to detect clearly invalid model or preprocessing configurations before evaluation.

C Full results

Figure 6 shows the success rate of every policy on each of the 40 tasks of LIBERO and LIBERO-Plus, sorted from easiest to hardest. For both benchmarks, $\pi_{0.5}$ and MolmoAct2 are the best policies.

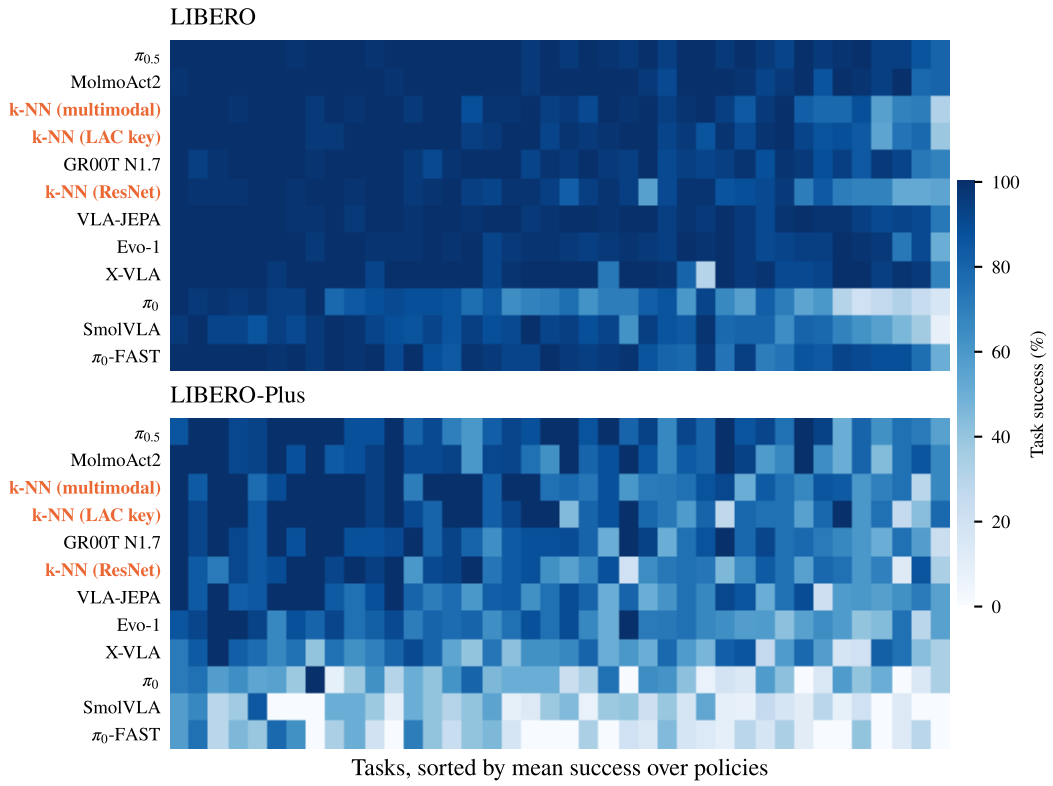


Figure 6: Success of every policy on each of the 40 tasks, tasks sorted from easiest to hardest (mean over policies).

Figure 7 shows the success rate of each policy on LIBERO-Plus, per perturbation category. As in Fig. 6, $\pi_{0.5}$ and MolmoAct2 are the most successful on average, and most policies are resilient to perturbed lighting. Camera perturbations reduce the success of all policies, except for the *LAC key* variant, which reaches 88 % against at most 73 % for the others.

D Embedding space visualization

We visualize the embedding spaces of all keys reported in Table 2, together with the generic visual keys, on the same set of 4,150 demonstration frames as in Fig. 3. We sample six demonstrations per task and retain every tenth frame. Each key is computed exactly as for the *ResNet* variant (Section 5): features are standardized using memory statistics, unit-normalized independently for each camera, and then concatenated across cameras. Figure 8 shows t-SNE embeddings computed from the first 50 principal components with perplexity 30, while Fig. 9 shows the first two principal components.

Before LIBERO fine-tuning, the embedding spaces are strongly structured by suite, and therefore by scene. The suite explains 40-84 % of the variance in the first two principal components across VLM keys, compared with 85 % for an ImageNet-pretrained ResNet-18. After fine-tuning, this fraction decreases to 1-43 %, with a decrease for 8 of the 8 fine-tuned VLMs. These two components capture 9-24 % of the total variance of each key, so the suite dominates the main directions of variation but not the full representation.

For every key, 84-100 % of the nearest neighbors from other demonstrations belong to the same task. Within LIBERO-Goal, where the ten tasks share one scene, the pattern differs. Only representations whose image tokens can attend to the instruction clearly separate its ten tasks. This includes the bidirectional prefix attention used by PaliGemma [33] in π_0, $\pi_{0.5}$, and π_0-FAST, as well as the Florence-2 encoder [34] used by X-VLA. These representations yield 96-100 % same-task nearest

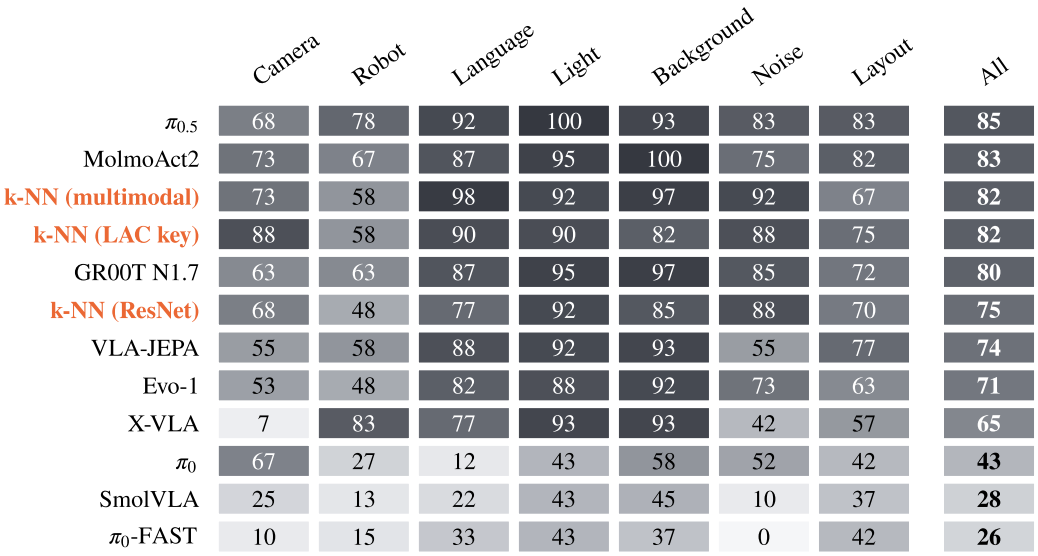


Figure 7: Success rate (%) of each policy on LIBERO-Plus tasks, per type of perturbation.

neighbors. In contrast, VLMs with causal attention, where image tokens precede the instruction and therefore cannot attend to it, achieve only 76-82 %, while ResNet-18 achieves 79 %. The median progress gap is discussed in Section 6.3.

E LAC and closed-loop retrieval

For each visual representation, we compute LAC offline on held-out demonstration queries and independently evaluate the *ResNet* k-NN variant using the same representation as its retrieval key. Figure 10 compares these quantities across generic visual encoders and VLM backbones before and after LIBERO fine-tuning. Higher LAC generally corresponds to higher closed-loop retrieval success, supporting its use as an offline measure of retrieval-key quality.

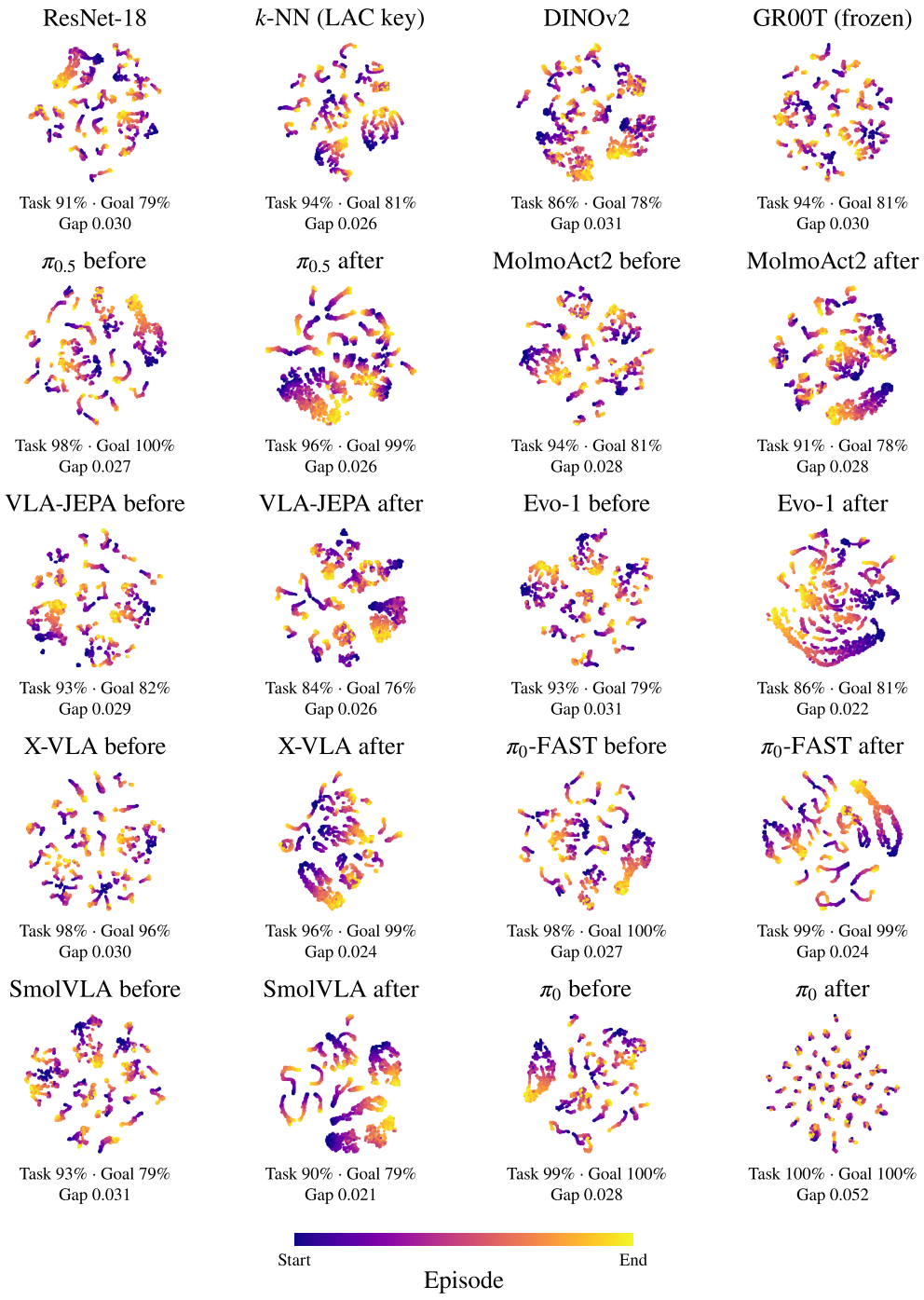


Figure 8: t-SNE visualization of each retrieval space, colored by progress through the episode. Below each map: Task, the fraction of nearest neighbors from another demonstration that belong to the same task; Goal, the same fraction within LIBERO-Goal; Gap, the median progress difference to the nearest frame from another demonstration of the same task. GR00T freezes its VLM, so its representation is unchanged by policy fine-tuning.

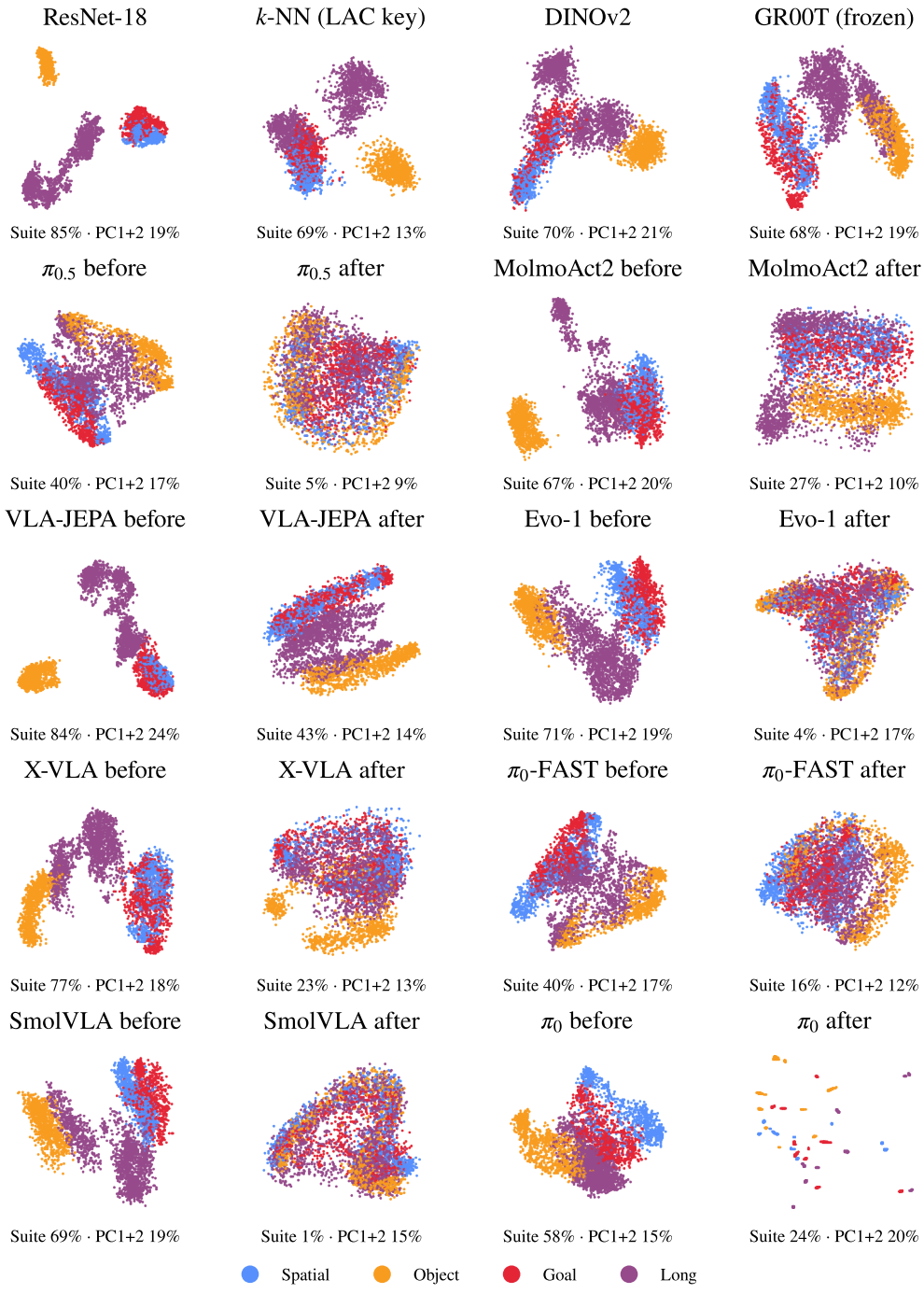


Figure 9: First two principal components of each retrieval key, colored by LIBERO suite. Below each map: Suite, the fraction of the variance of the two components explained by the suite; PC1+2, the fraction of the total variance captured by the two components.

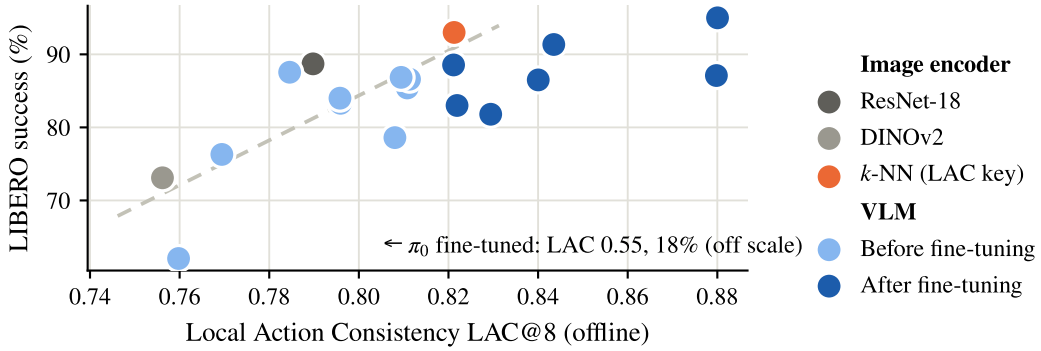


Figure 10: Offline LAC versus closed-loop retrieval success. Each point is a visual representation used as the key of the *ResNet* k-NN variant. The x-axis reports LAC@8 on held-out demonstrations, and the y-axis the LIBERO success rate. The dashed line is fitted on all keys except the fine-tuned VLM backbones, which were trained on the held-out demonstrations.