Near-Zero Forgetting Is Not Enough: Out-of-Distribution Robustness in Continual Learning of Pretrained VLAs

Adrià Ruiz
KU Leuven
Computer Vision Center
Universitat Autònoma de Barcelona
adriarp03@gmail.com

Bartłomiej Twardowski
Computer Vision Center
Universitat Autònoma de Barcelona
IDEAS Research Institute
btwardowski@cvc.uab.es

Tinne Tuytelaars
KU Leuven
tinne.tuytelaars@kuleuven.be

Sergi Masip
KU Leuven
sergi.masipcabeza@kuleuven.be

Abstract: Recent work shows that pretrained Vision-Language-Action (VLA) models exhibit near-zero forgetting in specific continual learning settings with experience replay. However, robustness to domain shift of learned tasks remains unexplored. In this work, we study the robustness of Pi0 fine-tuned with experience replay under out-of-distribution (OOD) data using the LIBERO and LIBERO-Plus benchmarks. While the in-domain (ID) setup achieves near-zero catastrophic forgetting, OOD performance on previous tasks quickly collapses. Specifically, we show that during the continual learning process, the OOD capabilities of the model degrade, following a shared representational pattern across domain shifts, which is also visible behaviorally in the two simpler suites. Representational analysis shows that drift concentrates in the multimodal and action features. Linear probes indicate that OOD features still retain task-relevant information, but appear unreadable, consistent with representation misalignment. These findings suggest that even when ID stability remains intact, severe OOD forgetting can still occur, accompanied by representational drift during continual learning.

Keywords: Continual Learning, Vision-Language-Action, Robotics, Catastrophic Forgetting, Pretrained VLA

1 Introduction

Vision-Language-Action models have emerged as the generalist approach for robot manipulation tasks. These models combine a pretrained Vision-Language Model (VLM) with an action head. This structure allows the transfer of generalist semantic priors to robot control tasks, enabling powerful generalization capabilities across different embodiments, tasks, and environments [1, 2, 3]. To avoid the costly retraining of these policies in changing real-world scenarios, continual adaptation would be desirable. This is the domain of continual learning (CL) [4], where new tasks or domains are learned sequentially. One of the main challenges is catastrophic forgetting [5], where the performance of previous tasks degrades as new ones are learned. This is closely related to the stability-plasticity trade-off [6], which presents the tension between the model's ability to learn new knowledge (plasticity) and retain previous knowledge (stability).

While the broader CL community has already observed that large-scale pretraining can alleviate forgetting [7, 8, 9], recent work on continual learning of VLA models has reported surprisingly strong resistance to forgetting in different regimes by using experience replay (ER) or naïve LoRA fine-tuning [10, 11, 12]. In some cases, forgetting is reduced to nearly zero or even replaced by

backward transfer. If these results extend to deployment settings, they would represent a significant shift in what continual learning can achieve, potentially bringing lifelong robotic systems closer to practical deployment under changing environments.

However, while these studies focus on in-domain (ID) retention, how out-of-distribution (OOD) robustness evolves remains open. This is particularly relevant given that recent benchmarks [13, 14, 15, 16] have demonstrated that VLA performance often degrades when pushed beyond standard benchmark setups. The near-zero forgetting results in Liu et al. [10] and Zhu et al. [11] were obtained with experience replay, where the replay buffer accumulates demonstrations exclusively from the training distribution. This risks narrowing the policy's representations, which could severely degrade its capacity to generalize. This concern motivates the central hypothesis of this work: near-zero forgetting under ER may be specific to ID evaluation and not extend to domain shifts of already-learned tasks.

To test this hypothesis, we reproduce the Liu et al. [10] baseline with Pi0 [1] across all four LIBERO suites, confirming their in-domain results. We then reevaluate the learned LIBERO tasks on LIBERO-Plus [13], defining an OOD evaluation procedure that lets us assess how the model reacts to distribution shifts. Finally, we analyze the model's internal representations to better understand its OOD behavior. Our main findings are summarized as follows:

1. **Despite ID near-zero forgetting, the model incurs heavy OOD forgetting.** We evaluate on OOD scenes, where we observe that the different perturbations induce a widening gap in performance between ID and OOD as the training progresses. This suggests that the model forgets its previous OOD capabilities (Section 4).
2. **Failure structure is shared across domain shifts.** The structural coherence of failures across all perturbations is consistent with a shared representational bottleneck rather than perturbation-specific failures. We also find that most representational drift occurs in the multimodal and action-expert layers, whereas the vision encoder is comparatively robust (Section 5).
3. **OOD features remain informative but misaligned.** Our analysis suggests that the representations from OOD episodes still keep features useful for planning. However, they appear misaligned with the ID representations, so an ID-trained readout cannot read them (Section 6).

2 Related Work

Near-zero forgetting in pretrained VLAs. Catastrophic forgetting is one of the main challenges in continual learning [5, 17]. Large-scale pretraining has been observed to mitigate forgetting beyond robotics, such as in pretrained vision backbones [18, 7], VLMs [8], or parameter-efficient fine-tuning [9]. Recent work pushes this further for large pretrained VLAs, reporting that they show greater resistance to catastrophic forgetting with simple strategies, and obtaining near-zero forgetting or even backward transfer [10, 12, 11].

Domain shift under fine-tuning and continual learning. Fine-tuning can erode OOD robustness; benchmarks such as ImageNet-RIB [19] show that the more diversely pretrained models suffer larger losses in OOD generalization when fine-tuned. In the intersection of continual learning with domain shift approaches for evolving domains, CTDG [20] models continually shifting distributions, and continual test-time adaptation [21] updates models as the environment changes. For robotics, related work addresses domain shift in navigation [22] and visually robust or online adaptation of VLAs [23, 24, 25]. In these works, the central question is whether the model can perform in a world with continual changes, but they do not question whether already known tasks remain robust under distribution shift of their already-learned knowledge.

Robustness evaluation for VLAs. Many datasets and benchmarks exist for robotic evaluation [26, 27, 28, 29, 30] but for the specific case of continual learning, LIBERO [31] remains the gold

standard. In recent years, many works have extended the LIBERO framework to further stress-test VLAs in different aspects such as probing memorization, language conditioning or visual robustness [13, 14, 15, 16, 32]. Specifically, our approach uses LIBERO-Plus [13], which maintains the original LIBERO tasks but expands into different controlled domain perturbations, showing a clear lack of generalization for different popular models.

Positioning. The preceding sections show a gap at the intersection of several active research areas. The empirical reports on near-zero forgetting [10, 12, 11] on pretrained VLAs open the door for new boundaries in continual learning, but none of these studies account for the out-of-distribution robustness of the trained tasks. While some CL mitigation strategies for VLAs, such as ExpReS-VLA [33], report OOD evaluations, these remain adaptation scenarios rather than asking how pretrained VLAs interact with unseen data. This paper addresses this gap by studying whether near-zero forgetting extends to out-of-distribution evaluation in the experience-replay regime.

3 Experimental Setup

3.1 Continual Learning Protocol

We formalize a task suite $S = \{T_0, \ldots, T_{N-1}\}$ consisting of N tasks that the model learns sequentially; after training on task T_k, we save checkpoint θ_k. Each suite has an in-domain (ID) evaluation set $\mathcal{D}_S$ and a family of out-of-distribution (OOD) versions of the same tasks, $\mathcal{P}_S = \{P_S^{(1)}, \ldots, P_S^{(M)}\}$, corresponding to the M perturbation axes applied to that suite. An evaluation cell $C(k, i, p)$ is the evaluation of checkpoint θ_k on task T_i under domain $p \in \{\mathcal{D}_S\} \cup \mathcal{P}_S$.

In all our experiments, we follow the setup in [10] for Pi0 [1], which combines a pretrained PaliGemma VLM backbone with a flow matching action expert. We fine-tune the model using a replay buffer that accumulates 1,000 frames per task. For each subsequent task T_{k+1}, training batches contain a 1:1 ratio of current-task and replay samples.

3.2 Benchmarks

For ID evaluation, we use the four task suites from the LIBERO [31] benchmark: LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-10. To measure domain shift, we use LIBERO-Plus [13], which extends these suites with different domain perturbations applied to the original LIBERO tasks. Specifically, LIBERO-Plus introduces seven perturbation axes that include changes to the background textures, the robot initial state, the external camera viewpoint, the layout of the objects, the light conditions, the rewording of the language instructions, and the addition of sensor noise.

3.3 Evaluation Metrics

To evaluate continual learning performance, we rely on different metrics used in the continual learning literature. We compute the average cumulative accuracy (ACA), which measures the performance on all the tasks the model has seen up to the current checkpoint:

$$ACA_k = \frac{1}{k+1} \sum_{i=0}^{k} SR(k, i, p). \tag{1}$$

Intuitively, a reduction in the ACA translates to a decrease in the general understanding of the tasks.

To measure forgetting and, therefore, stability, we use negative backward transfer (NBT) as described by Liu et al. [31]. This metric tracks performance drops relative to when the task was initially learned. For $N \geq 2$, with $NBT_{N-1} = 0$, it is defined as follows:

$$NBT = \frac{1}{N} \sum_{i=0}^{N-1} NBT_i, \quad NBT_i = \frac{1}{N-i-1} \sum_{k=i+1}^{N-1} \Big(SR(i, i, p) - SR(k, i, p) \Big). \tag{2}$$

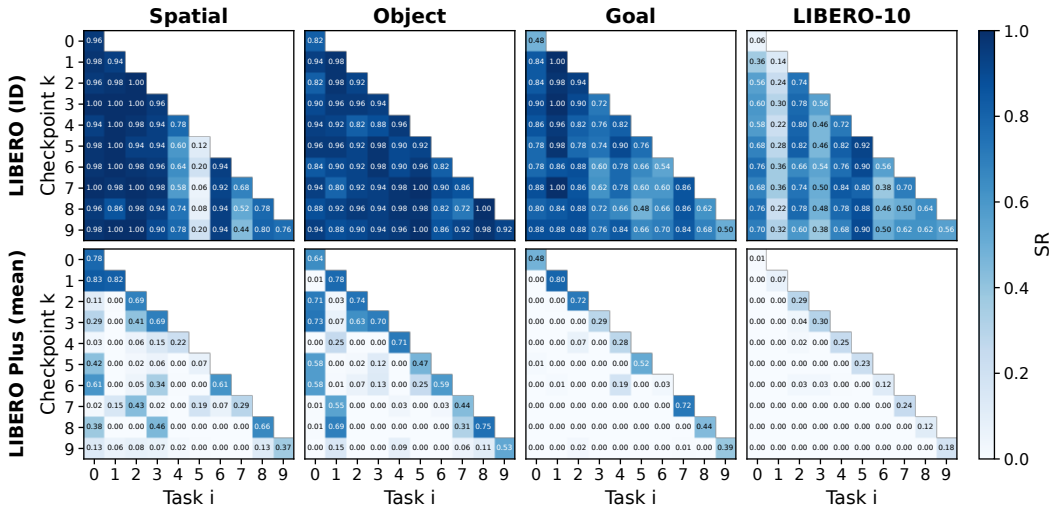


Figure 1: **ID and OOD success rate matrices for LIBERO and LIBERO-Plus (mean across perturbation categories).** Each cell $C(k, i)$ reports the success rate of checkpoint θ_k (row) on task T_i (column); only valid cells with $k \geq i$ are shown. For the first row of subplots, the ID cells are shown; for the second, the mean of the cells across the OOD categories is shown. For ID, the dense, high-SR lower triangles indicate near-zero forgetting across all four suites, whereas for OOD the lower-triangular regions show a clear drop in performance: most cells are close to zero, while some isolated spikes of *active cells* (under-diagonal cells with SR > 0) remain, revealing OOD forgetting.

A low NBT means the model is highly stable and does not forget past tasks, while a negative NBT implies backward transfer. We compute it on both ID and OOD domains p. We use OOD NBT to quantify OOD forgetting.

3.4 Representation Analysis

We analyze the evolution of the model's internal representations at three feature-extraction points: after the vision encoder, after the VLM, and after the first flow matching step at the action module. All features are taken from a single joint forward pass with the flow-matching step locked at $\tau = 0$, and then mean-pooled over non-padding tokens into one vector per frame. These vectors are then used to compare across checkpoints and domains. We use Linear CKA [34] to compute the drift in the same tasks (lower CKA implies more drift), Maximum Mean Discrepancy [35] to measure the distributional gap between ID and OOD (higher MMD implies a larger gap), and linear probes [36] to quantify how much task-relevant information in the representations is linearly accessible (lower R^2 equals less readable). Details are in Appendix D.

4 Near-Zero In-Domain Forgetting Does Not Extend to OOD

We first reproduce the ID near-zero forgetting reported in Liu et al. [10], as shown in Figure 1. The four evaluation matrices for the LIBERO suites show consistently high success rates (SR), indicating that the model retains strong performance while still learning new tasks. The NBT for the suites is slightly negative for Object (-0.005), Goal (-0.017), and LIBERO-10 (-0.036), and a small positive value for Spatial (0.026). These results show mild backward transfer for three of the suites and mild forgetting for Spatial, but driven only by one task, therefore confirming the high ID resistance to forgetting.

In contrast, the OOD evaluation on LIBERO-Plus shows predominantly low success rates, with most entries being exactly zero. Note that OOD performance is already low at task acquisition, with a mean SR fall from ID by about 0.27, 0.27, 0.25 and 0.38 on Spatial, Object, Goal and

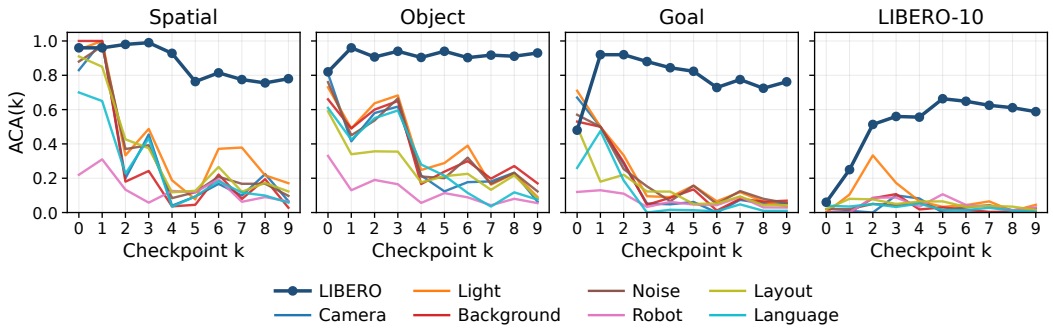


Figure 2: **Average Cumulative Accuracy over checkpoints for LIBERO and each LIBERO-Plus perturbation.** LIBERO ACA (blue) remains stable or increases slightly, while all LIBERO-Plus categories (colored lines) decline, showing a widening ID–OOD performance gap during training.

LIBERO-10 respectively, reflecting the initial domain shift (Appendix E). However, we observe a substantial further degradation in performance after learning subsequent tasks. The mean OOD NBT is $+0.40 \pm 0.13$, $+0.48 \pm 0.17$, $+0.42 \pm 0.15$ and $+0.16 \pm 0.09$ on Spatial, Object, Goal and LIBERO-10 respectively (mean $\pm$std. over the seven Plus categories), with the performance often collapsing to ≈ 0 after just one or two additional tasks.

Interestingly, the observed OOD catastrophic forgetting varies by suite: in Spatial and Object, the less complex suites, scattered cells retain non-trivial OOD performance, creating an irregular pattern of *active cells* (under-diagonal cells with SR > 0). In the hardest suites, Goal and LIBERO-10, these patterns are mostly gone, leaving just a dead under-diagonal triangle.

Figure 2 plots the ACA over checkpoints. Despite the new irregular pattern of scattered active cells, OOD forgetting persists as new tasks are learned. The contrast with ID performance is clear: ID remains high and even slightly improves, indicating backward transfer in most suites, whereas OOD ACA declines continuously for most perturbations. This widening gap between LIBERO and LIBERO-Plus on the ACA shows how OOD robustness is rapidly eroded under continual learning, even while preserving in-domain performance.

5 A Common Failure Pattern Across Perturbations

5.1 The Behavioral Pattern

We observed that OOD performance at acquisition drops for every perturbation, to a degree that depends on its type: lighting changes are the most benign, whereas changes to the robot's initial state are the most damaging (see Appendix E). Interestingly, however, we found that the perturbation type affects the magnitude of the acquisition drop but does not change the location of the *active cell* spike pattern (see Section 4)—these spikes occur at the same tasks and checkpoints.

To illustrate this active cell spike pattern, we construct, for each perturbation category, a vector containing $C(k, i, p)$ across all $i < k$ for a fixed p. We then compute the pairwise Pearson correlation r between these vectors. For Spatial and Object, the correlations are strongly positive, with mean $\bar{r} = 0.73$ and 0.82, respectively, and all pairwise correlations are above 0.5 (minimum $r = 0.53$ for Spatial and 0.60 for Object). Partial correlations given ID cell SR are nearly identical to the raw values (Spatial: 0.72; Object: 0.81), indicating that the cross-perturbation match is not an artifact of easy tasks staying easy. The full correlation matrices are shown in Appendix G. In contrast, for Goal and LIBERO-10, most under-diagonal cells are zero, and besides the complete forgetting, we observe no clear emerging behavioral pattern.

The behavioral pattern seen across perturbations for Spatial and Object suggests that OOD forgetting might stem from the model's internal representations rather than sensitivity to specific perturbations.

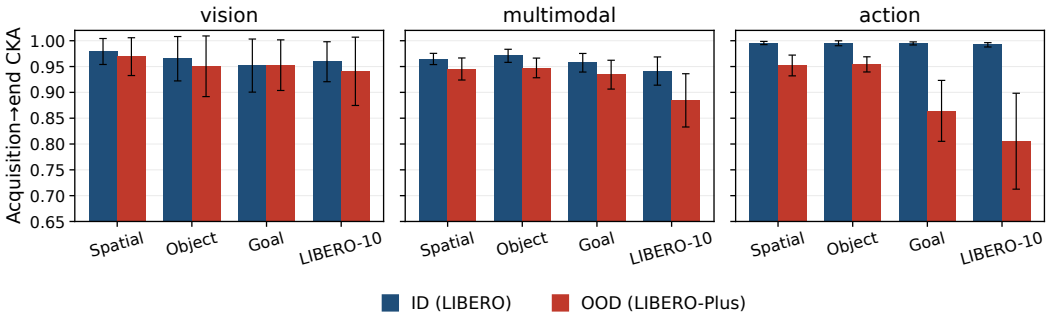


Figure 3: **Acquisition-to-end CKA drift by extraction module (vision, multimodal, action).** Each barplot shows mean CKA(θ_9, θ_i) averaged over tasks, with the task std., for LIBERO (blue) and LIBERO-Plus (red). Vision is comparatively robust (small ID–OOD gap, within the task-to-task spread); multimodal shows a small but consistent gap; the action expert drifts the most, with higher variance on the harder suites.

If the OOD forgetting were driven, for example, by an inability to handle new camera angles, we would expect the camera perturbations to produce a distinct pattern from another category. Instead, the similar patterns across perturbations suggest a representation bottleneck in the ID distribution that limits adaptation to shifts across all domains in the same way.

5.2 A Shared Representational Property

To test whether the observed behavioral pattern reflects a property of the representation space rather than a coincidence, we ask if the structure of the representations is coherent across perturbations, as observed behaviorally with the active cells pattern. We use two metrics to check two complementary representational aspects. First, we look at the multimodal ID–OOD gap: we compute MMD^2 between the cells $C(k, i, \mathcal{D}_S)$ and $C(k, i, p)$ for a specific perturbation type $p \in \mathcal{P}_S$. Secondly, we measure multimodal drift with teacher-forced CKA between the checkpoint θ_i at which task T_i is learned and every later checkpoint θ_k ($k > i$), on T_i under domain p, i.e. between cells $C(i, i, p)$ and $C(k, i, p)$ (see Appendix D). For each suite, we flatten both metrics' per-category profiles into vectors—MMD^2 across all lower-triangular evaluation cells and CKA across all pairs $k > i$—and compute pairwise Pearson correlations across perturbation categories.

Table 1: Cross-perturbation coherence: mean pairwise Pearson $\bar{r}$ between per-category flattened cells for multimodal ID–OOD MMD^2 and teacher-forced multimodal CKA. Positive values across all suites show that the shared structure extends to Goal and LIBERO-10, where behavioral coherence could not be estimated. Full correlation plots can be found in Appendix G.

Suite	Spatial	Object	Goal	LIBERO-10
MMD^2	0.67	0.76	0.76	0.77
CKA	0.64	0.75	0.72	0.84

As illustrated in Table 1, the representational metrics exhibit positive correlations across all suites, with no negative pairs in any perturbation combination. In other words, regardless of the perturbations being compared, the ID–OOD representational gaps and drift follow a similar structure. Notably, this pattern also holds for Goal and LIBERO-10, even where it is not apparent at the behavioral level because the performance collapsed to zero.

These results indicate that the pattern of cells showing larger representational gaps is shared across perturbation categories. While the perturbation type determines the overall magnitude and difficulty of overcoming the domain shift, it does not determine the relative profile of where the gap grows or shrinks across the evaluation matrix. This suggests that the observed structure stems from

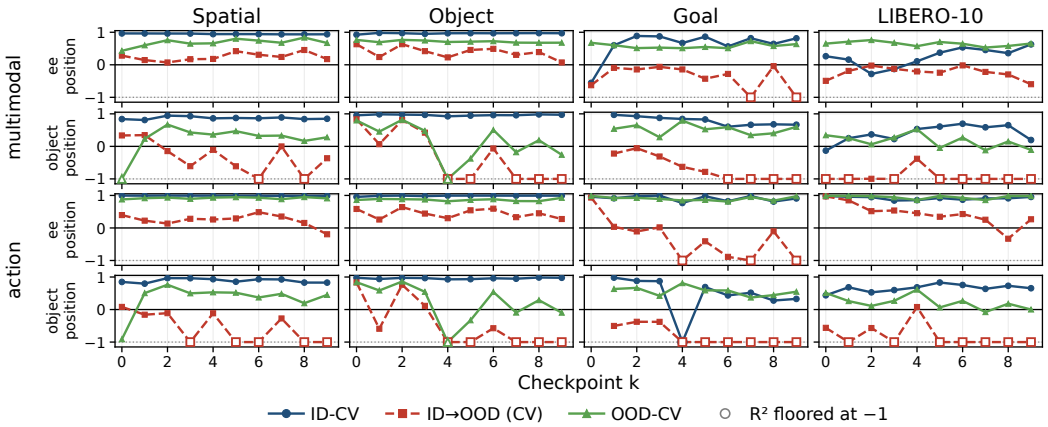


Figure 4: **Linear-probe R^2 for multimodal and action features predicting end-effector and manipulated object position.** Rows pair an extraction point with a regression target; columns are suites. Blue: ID-CV; red dashed: ID→OOD transfer; green: OOD-CV. Open markers are floored at -1 because the R^2 regression metric can yield unbounded negative scores if it is worse than guessing the mean. Across all suites, ID→OOD is consistently the worst score, while OOD-CV recovers, indicating that the target information remains linearly accessible out of distribution but the in-domain readout is misaligned.

training dynamics and how the model represents the data, rather than from features specific to each perturbation.

5.3 Module Localization of the Bottleneck

In Figure 3, we compare the mean acquisition CKA(θ_9, θ_i) for tasks T_i to measure how much the representation of a task drifts from acquisition to the end of continual fine-tuning. On the ID side, all modules exhibit low representation drift, consistent with the observed near-zero in-domain forgetting. In contrast, the OOD side shows different behaviors across extraction modules.

The vision encoder is comparatively robust: its ID–OOD gap is small (at most 0.02 CKA) and within the task-to-task spread. The multimodal representation shows the first signs of consistent decay, with the OOD drift being higher than ID drift and, although the gap is small, it is systematic, low-variance, and consistent with the observed task difficulty. Finally, the action expert shows the largest gap between domains, but also higher variance, especially for Goal and LIBERO-10. Some of this noise can be explained by the fact that the action representation corresponds to the first of ten representations in the flow-matching process, and therefore can contain a significant portion of the noise.

6 OOD Representations Misalign

To obtain a first view of the information encoded in the model's hidden representations, we employed linear probes [36]. The probe's R^2 score quantifies how much of the target information can be linearly decoded from the representation. We fit linear probes on top of frozen features. For each checkpoint θ_k and task T_i, we extracted the pooled representations for multimodal and action extraction points, and set two regression targets: the 3D end-effector position, which serves as an easier baseline since it benefits from Pi0's proprioceptive state input, and the 3D position of the manipulated object, which cannot be directly inferred and provides the purer probe of visual-semantic readability.

To assess information loss, we ran three probe experiments: cross-validation trained and tested in ID (ID-CV), the same cross-validation-trained probes in ID but evaluated on LIBERO-Plus (ID→OOD), and a cross-validated probe trained and tested on OOD (OOD-CV). This way we can

respectively measure how much information is accessible in the original domain, the representation misalignment between ID and OOD, and the accessibility of the same information in OOD.

Figure 4 presents the regression linear probe evaluations. For the end-effector target, the pattern is clear for both multimodal and action features: the in-domain probe (ID-CV) achieves high R^2, while the transferred probes (ID→OOD) show a sharp drop. The OOD-CV probes partially recover performance but do not regain the robustness of the original ID probe. Across all suites and checkpoints, the transferred probe is the worst of the three in 96.6% of the cells for this target. The object-position target follows the same qualitative conclusion, holding for 92.9% of the cells, though it is noisier. The median transfer R^2 for this target is around -1.32 (multimodal) and -3.33 (action), confirming the conclusion is not driven by a few easy cells, and both targets support the same pattern.

In general, the pattern is consistent: the transfer score is the worst, and although ID-CV dominates, OOD-CV recovers in most cases. This means that the information might still be accessible in the OOD data but misaligned, rather than a widespread loss of information.

7 Discussion

Near-zero forgetting is an important step toward practical continual learning in robotics, as it offers the possibility of avoiding the main cost of catastrophic forgetting and deploying policies that are robust under changing environments. In this work, we asked whether this regime still holds when already learned tasks are evaluated under domain shift. To do so, we reproduced the Pi0 experience-replay baseline of Liu et al. [10] on the four LIBERO suites, reevaluated the same tasks under the seven LIBERO-Plus perturbations, and analyzed the resulting representations with CKA, MMD, and linear probes.

We found that despite near-zero forgetting in ID, the model incurs heavy OOD forgetting: as sequential fine-tuning progresses, the ID–OOD performance gap widens even while in-domain retention stays intact. We also found that the pattern of active cells—tasks that keep non-zero OOD success under a given checkpoint—is shared across the seven LIBERO-Plus categories for the suites that do not fully collapse, and that this shared structure shows up in the multimodal representations and potentially in the action representations, whereas the vision encoder remains comparatively robust. Finally, the linear probe experiments suggested that the information in the representations remains mostly accessible in the OOD regime but is misaligned, consistent with the model narrowing its representations to the specific ID data. These results not only reveal heavy OOD forgetting in this regime but also offer a potential explanation for the observed phenomena, such as the scattered active cell pattern: the structure of forgetting is shared across perturbations, which is more consistent with a common underlying cause than with perturbation-specific effects.

Overall, ID retention alone is not sufficient evidence of deployment readiness, since near-zero forgetting can coexist with heavy OOD forgetting. Technical challenges in how the model represents and structures the information still need to be addressed. Future studies in the same regimes should consider OOD-aware testing and mitigation strategies for the domain-shift challenges that appear in real-world scenarios.

Limitations. An important limitation is that we only evaluated one model, Pi0, in a single regime with a fixed replay strategy, and results come from a single seed. Our results are specific to this setting, and it is difficult to know how OOD robustness would look in other near-zero forgetting setups. Another important limitation is that the representational analysis is correlational, meaning the correlations found are consistent with causal relationships but not proven, since establishing causality would require controlled interventions to see how the model reacts. Finally, LIBERO-Plus has a non-uniform distribution of test scenes across tasks and perturbation categories, and although most cells have many samples, some contain very few scenes. This limitation is mitigated by focusing on aggregated metrics, but further discussion can be found in Appendix F.

Acknowledgments

This paper has received funding from the Flemish Government under the Methusalem Funding Scheme (grant agreement n° METH/24/009) and National Science Centre, Poland, grant no 2023/51/D/ST6/02846. Bartłomiej Twardowski acknowledges the grant RYC2021-032765-I.

References

[1] K. Black, N. Brown, D. Driess, A. Esmail, M. 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, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Proc. Robot.: Sci. Syst. (RSS)*, 2025.

[2] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[3] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. OpenVLA: An open-source vision-language-action model. In *Proc. Conf. Robot. Learn. (CoRL)*, volume 270, pages 2679–2713, 2024.

[4] G. M. Van de Ven and A. S. Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

[5] M. McCloskey and N. J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychol. Learn. Motiv.*, volume 24, pages 109–165. Elsevier, 1989.

[6] R. Hadsell, D. Rao, A. A. Rusu, and R. Pascanu. Embracing change: Continual learning in deep neural networks. *Trends Cogn. Sci.*, 24(12):1028–1040, 2020.

[7] Z. Wang, Z. Zhang, S. Ebrahimi, R. Sun, H. Zhang, C.-Y. Lee, X. Ren, G. Su, V. Perot, J. Dy, et al. DualPrompt: Complementary prompting for rehearsal-free continual learning. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pages 631–648, 2022.

[8] D.-W. Zhou, Y. Zhang, Y. Wang, J. Ning, H.-J. Ye, D.-C. Zhan, and Z. Liu. Learning without forgetting for vision-language models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(6):4489–4504, 2025.

[9] D. Biderman, J. Portes, J. J. Gonzalez Ortiz, M. Paul, P. Greengard, C. Jennings, D. King, S. Havens, V. Chiley, J. Frankle, C. Blakeney, and J. P. Cunningham. LoRA learns less and forgets less. *Trans. Mach. Learn. Res.*, 2024.

[10] H. Liu, C. Kim, B. Liu, M. Liu, and Y. Zhu. Pretrained vision-language-action models are surprisingly resistant to forgetting in continual learning. In *Proc. Int. Conf. Mach. Learn. (ICML)*, 2026.

[11] J. Zhu, Y. Hong, X. Sun, Z. Xu, Q. He, H. Chen, Z. Wang, M. Yuan, W. Zeng, and J. Chen. Can VLA models learn from real-world data continually without forgetting? *arXiv preprint arXiv:2605.26820*, 2026.

[12] J. Hu, J. Shim, C. Tang, Y. Sung, B. Liu, P. Stone, and R. Martín-Martín. Simple recipe works: Vision-language-action models are natural continual learners with reinforcement learning. *Reinf. Learn. J.*, 7, 2026.

[13] S. Fei, S. Wang, J. Shi, Z. Dai, J. Cai, P. Qian, L. Ji, X. He, S. Zhang, Z. Fei, et al. LIBERO-Plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

[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] C. Kim, M. Kim, M. Kang, H. Kim, and D. Jung. LIBERO-Para: A diagnostic benchmark and metrics for paraphrase robustness in VLA models. *arXiv preprint arXiv:2603.28301*, 2026.

[16] G. Wang, C. Zhang, Q. Liu, J. Zhang, J. Cai, J. Liu, and X. Liu. LIBERO-X: Robustness litmus for vision-language-action models. *arXiv preprint arXiv:2602.06556*, 2026.

[17] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proc. Nat. Acad. Sci.*, 114(13):3521–3526, 2017.

[18] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. Dy, and T. Pfister. Learning to prompt for continual learning. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 139–149, 2022.

[19] J. Hwang, B. Cheung, Z.-W. Hong, A. Boopathy, P. Agrawal, and I. Fiete. Large pretraining datasets don't guarantee robustness after fine-tuning in image classification. *Trans. Mach. Learn. Res.*, 2026.

[20] Z. Cai, G. Bai, R. Jiang, X. Song, and L. Zhao. Continuous temporal domain generalization. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 37, pages 127987–128014, 2024.

[21] Q. Wang, O. Fink, L. Van Gool, and D. Dai. Continual test-time domain adaptation. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 7191–7201, 2022.

[22] R. Hisada and K. Tanaka. FlatManifold: Robust continual learning under severe label noise and domain shifts via intrinsic manifold flattening. *arXiv preprint arXiv:2607.05201*, 2026.

[23] Y. Peng, J. Fu, C. Zhang, L. Zhao, J. Bian, M. Liu, L. Zhang, J. Zhang, and R. Wang. What to ignore, what to react: Visually robust RL fine-tuning of VLA models. *arXiv preprint arXiv:2605.13105*, 2026.

[24] K. Wang, T. Nguyen, M. H. Nguyen, S. Khan, and F. D. Salim. ROAD-VLA: Robust online adaptation via self-distillation for vision-language-action models. *arXiv preprint arXiv:2606.25800*, 2026.

[25] N. Kachaev, M. Kolosov, D. Zelezetsky, A. Kovalev, and A. Panov. Don't blind your VLA: Aligning visual representations for OOD generalization. In *Proc. Int. Conf. Mach. Learn. (ICML) Workshop on Scalable Learning and Optimization for Efficient Multimodal AI Agents (SCALE)*, 2026.

[26] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, et al. Open X-embodiment: Robotic learning datasets and RT-X models. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 6892–6903, 2024.

[27] H. R. Walke, K. Black, T. Z. Zhao, Q. Vuong, C. Zheng, P. Hansen-Estruch, A. W. He, V. Myers, M. J. Kim, M. Du, A. Lee, K. Fang, C. Finn, and S. Levine. BridgeData V2: A dataset for robot learning at scale. In *Proc. Conf. Robot. Learn. (CoRL)*, pages 1723–1736, 2023.

[28] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *Proc. Robot.: Sci. Syst. (RSS)*, 2024.

[29] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao, X. Yuan, P. Xie, Z. Huang, R. Chen, and H. Su. ManiSkill2: A unified benchmark for generalizable manipulation skills. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023.

[30] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard. CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robot. Autom. Lett.*, 7(3):7327–7334, 2022.

[31] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 36, pages 44776–44791, 2023.

[32] Y. Fang, Y. Feng, D. Jing, J. Liu, Y. Yang, Z. Wei, D. Szafir, and M. Ding. When vision overrides language: Evaluating and mitigating counterfactual failures in VLAs. In *Proc. Conf. Robot. Learn. (CoRL)*, 2026.

[33] S. N. Syed, Y. Ahuja, A. Jakobsson, and J. Ichnowski. ExpReS-VLA: Specializing vision-language-action models through experience replay and retrieval. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2026.

[34] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton. Similarity of neural network representations revisited. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pages 3519–3529, 2019.

[35] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *J. Mach. Learn. Res.*, 13(25):723–773, 2012.

[36] G. Alain and Y. Bengio. Understanding intermediate layers using linear classifier probes. In *Proc. Int. Conf. Learn. Represent. (ICLR) Workshop*, 2017.

A LIBERO Suite Descriptions

Table 2 details the axes of variability for each of the four LIBERO suites referenced throughout the paper and sustains the claims of considering Spatial and Object easier suites with respect to Goal and LIBERO-10.

Table 2: The four LIBERO benchmark suites and their axes of variability.

Suite	Variability	Fixed Elements	Example
Spatial	Source	Environment, Action/Goal, Object, Destination	*pick up the black bowl between the plate and the ramekin*
Object	Object	Environment, Action/Goal, Source, Destination	*pick up the alphabet soup and place it in the basket*
Goal	Action/Goal, Object, Destination	Environment, Source	*open the middle drawer of the cabinet*
LIBERO-10	Environment, Action/Goal, Object, Source, Destination	—	*turn on the stove and put the moka pot on it*

B Training and Evaluation Settings

Table 3 shows the hyperparameters used to fine-tune the model in the Continual Learning setup, which includes an Experience Replay buffer of 1000 frames per task, 1:1 current/replay mixing, LoRA on the VLM and action expert with full fine-tuning of the vision encoder and flow-matching projections. Training ran mainly on NVIDIA A100-SXM4-80GB GPUs; evaluation and representation rollouts mostly ran on A100-SXM4-80GB as well and, in some cases, on RTX 3090s.

Table 3: Continual fine-tuning hyperparameters (Pi0 / openpi low-memory LoRA recipe).

Hyperparameter	Value
Architecture & Compute	
Base checkpoint	π_0 base pretrained
Precision	Mixed (bfloat16)
Hardware	NVIDIA A100-SXM4-80GB
Optimization	
Optimizer	AdamW (β_1=0.9, β_2=0.95, wd=10^{-10})
LR schedule	Cosine (warmup 1000, peak 2.5×10^{-5}, end 2.5×10^{-6})
Schedule length	global 30,000 decay steps
Steps per task	10,000
Batch size	8 (1:1 ER mix within batch)
EMA	Disabled
LoRA Configuration	
VLM / Gemma	r=16, α=16
Action expert	r=32, α=32
Environment & Evaluation	
ER buffer / mix	1000 frames/task; ρ=0.5
Replan / settle	Every 5 env steps / 10 settle steps
ID eval episodes	50 fixed initial states / task
Eval horizons	Spatial: 220, Object: 280, Goal: 300, LIBERO-10: 520

Project page: https://adrrp.github.io/vla-ood-robustness.

C Task Sequences

All suites use the task order of Liu et al. [10] (indices 0–9). Table 4 lists the language instruction of each task; LIBERO-Plus evaluates the same underlying tasks under the seven perturbation categories.

Table 4: LIBERO task order (T_0–T_9) per suite.

(a) Spatial

ID	Instruction
0	pick up the black bowl between the plate and the ramekin and place it on the plate
1	pick up the black bowl next to the ramekin and place it on the plate
2	pick up the black bowl from table center and place it on the plate
3	pick up the black bowl on the cookie box and place it on the plate
4	pick up the black bowl in the top drawer of the wooden cabinet and place it on the plate
5	pick up the black bowl on the ramekin and place it on the plate
6	pick up the black bowl next to the cookie box and place it on the plate
7	pick up the black bowl on the stove and place it on the plate
8	pick up the black bowl next to the plate and place it on the plate
9	pick up the black bowl on the wooden cabinet and place it on the plate

(b) Object

ID	Instruction
0	pick up the alphabet soup and place it in the basket
1	pick up the bbq sauce and place it in the basket
2	pick up the butter and place it in the basket
3	pick up the chocolate pudding and place it in the basket
4	pick up the cream cheese and place it in the basket
5	pick up the ketchup and place it in the basket
6	pick up the milk and place it in the basket
7	pick up the orange juice and place it in the basket
8	pick up the salad dressing and place it in the basket
9	pick up the tomato sauce and place it in the basket

(c) Goal

ID	Instruction
0	open the middle drawer of the cabinet
1	put the bowl on the stove
2	put the wine bottle on top of the cabinet
3	open the top drawer and put the bowl inside
4	put the bowl on top of the cabinet
5	push the plate to the front of the stove
6	put the cream cheese in the bowl
7	turn on the stove
8	put the bowl on the plate
9	put the wine bottle on the rack

(d) LIBERO-10

ID	Instruction
0	pick up the book and place it in the back compartment of the caddy
1	put both moka pots on the stove
2	put both the alphabet soup and the cream cheese box in the basket
3	put both the alphabet soup and the tomato sauce in the basket
4	put both the cream cheese box and the butter in the basket
5	put the black bowl in the bottom drawer of the cabinet and close it
6	put the white mug on the left plate and put the yellow and white mug on the right plate
7	put the white mug on the plate and put the chocolate pudding to the right of the plate
8	put the yellow and white mug in the microwave and close it
9	turn on the stove and put the moka pot on it

D Representation Metric Settings

Feature Extraction

To observe how the model's internal representations evolve, we compute different representation metrics at three feature-extraction points. The first is the vision representation, which is taken at the output of the vision encoder for each camera; the second and third are the multimodal and action representations. For these, all the observation data is fed with the initial action noise $A_t^0 \sim \mathcal{N}(0, I)$, and the final output is separated by prefix, corresponding to the multimodal representation (vision and language) and suffix, corresponding to the action representation taken just before it passes through the flow matching projection layers.

We lock the extraction at the flow-matching step $\tau = 0$ and run a single joint forward pass with all the data, as we do when computing the training loss. Because of the attention mask, the multimodal representation does not see the action tokens, and therefore this is identical to taking the final representation at the multimodal pass at inference.

The noise A_t^0 is the same for a given scene and frame across all checkpoints, and we use $\tau = 0$ because it is the only step where the action expert receives the same input in every checkpoint. At later steps the input depends on the velocities predicted by each checkpoint, so each model would be at a different part of the denoising process and the representations would not be comparable, even starting from the same noise. The downside is that the action representation at this step is closer to the noise, so we interpret it with caution. It is not just noise though, since the linear probes of Figure 4 are able to recover the object position from it.

To reduce to a single vector, we mean-pool all representations over the token axis. The unpooled sequence of tokens is a total of N vectors, where each token $t_j \in \mathbb{R}^d$ has a feature dimension d. A binary mask $m_j \in \{0, 1\}$ is introduced to ignore the padding tokens to avoid merging with irrelevant information, and the tokens are averaged, obtaining a fixed-size vector $v \in \mathbb{R}^d$:

$$v = \frac{1}{\sum_{j=1}^{N} m_j} \sum_{j=1}^{N} m_j t_j \tag{3}$$

Specifically, for the vision representation, to distinguish the spatial identity of each camera view, since the number of cameras is constant, the tokens of each camera are pooled individually and then concatenated. The multimodal representation contains both the image and language tokens, which are pooled jointly into a single vector, and for the action representation the pooling is done across the H action vectors without any masking.

Centered Kernel Alignment (CKA)

With a sample of n frames and a pooled feature dimension of d, the aligned feature matrices $X \in \mathbb{R}^{n \times d}$ and $Y \in \mathbb{R}^{n \times d}$ are used to compute Linear CKA [34], which measures the structural similarity between two sets of representations. Given column-centered matrices X_c and Y_c, CKA is defined as:

$$\text{CKA}(X, Y) = \frac{\|Y_c^\top X_c\|_F^2}{\|X_c^\top X_c\|_F \cdot \|Y_c^\top Y_c\|_F} \tag{4}$$

The CKA value lies between 0 and 1, with lower values showing completely different representations and higher values indicating an identical geometric structure.

We use CKA to study how a specific task T_i evolves as the model learns new tasks. This procedure extracts and compares representations from two checkpoints, θ_k and θ_m, evaluated on the same task data and domain p. In other words, we analyze how the representation varies in cells $C(\cdot, i, p)$.

Normally, evaluation is done in a closed-loop setup, where predicted actions define the robot's behavior and, therefore, the environment's next state. However, this raises a problem: checkpoints behave differently, so the observed environment will differ after just a few frames. To compare how the same data is treated across checkpoints, we need to use the same observations for both models.

Teacher Forcing does exactly that: instead of following the model's actions, a fixed set of expert trajectories is prerecorded from a reference checkpoint for all tasks T_i and domains $p \in \{\mathcal{D}_S\} \cup \mathcal{P}_S$. These trajectories guide the robotic arm, ensuring identical environment perception for both models and guaranteeing frame-by-frame synchronization. For the expert trajectories of $\mathcal{D}_S$, we choose the final checkpoint after continual learning, θ_9, for all tasks and suites. For the perturbed domains, we instead generate the trajectory for each task with the checkpoint trained on it, so for any task T_i we record with θ_i. This selection is done to ensure that both ID and OOD contain diverse trajectories, since there is no clear common set of checkpoints between ID and OOD that contain enough successful trajectories; the main diagonal is too low for some specific tasks of ID Spatial and LIBERO-10, and the last row fails catastrophically in OOD. We acknowledge that using a different reference checkpoint in each domain is a significant difference, even with the justification above. However, two observations suggest that our CKA conclusions are not driven only by this choice.

First, the same frames are fed to the three extraction modules (vision, multimodal and action), and in Figure 3 the vision encoder shows a much smaller ID–OOD gap (at most 0.02 CKA) than the multimodal and action modules, which would be unexpected if the trajectory source alone were driving it. Second, the cross-perturbation coherence of Table 1 also holds for MMD^2, which is computed in closed loop at a single checkpoint and does not use teacher forcing. Still, we interpret the magnitude of the CKA gap with caution.

We do not compute CKA between all frames; instead, whenever the model is triggered, which happens every 5 executed actions, and generates a new action chunk, we save the representations produced by that trigger. For a given scene rollout, we collect these periodically extracted representation vectors into a single feature matrix for each checkpoint. Then we compute CKA between the two scene-level matrices. When comparing two whole cells, we average the CKA produced by comparing each scene in the task that cell contains.

Maximum Mean Discrepancy (MMD)

Maximum Mean Discrepancy [35] is a kernel-based statistic that measures the distance between two distributions embedded in a reproducing kernel Hilbert space. Given samples $X = \{x_i\}_{i=1}^n \sim P$ and $Y = \{y_j\}_{j=1}^m \sim Q$, we estimate this distance with the unbiased MMD^2 estimator and an RBF kernel $k(x, y) = \exp\big(-\|x - y\|^2/2\sigma^2\big)$:

$$\begin{aligned} \mathrm{MMD}^2(X, Y) = &\frac{1}{n(n-1)} \sum_{i \neq j} k(x_i, x_j) \\ &+ \frac{1}{m(m-1)} \sum_{i \neq j} k(y_i, y_j) \\ &- \frac{2}{nm} \sum_{i,j} k(x_i, y_j) \end{aligned} \tag{5}$$

To set σ, we use the median of all pairwise Euclidean distances over the pooled set $X \cup Y$.

In population, $\mathrm{MMD}(P, Q)$ is a metric between distributions in kernel space, so $\mathrm{MMD}^2(P, Q)$ equals zero if and only if P and Q are indistinguishable under the RBF kernel. However, we report its unbiased finite-sample estimate from the formula above. Being unbiased, this estimate can have some noise when $P \approx Q$ (sampling noise), so we read values near zero as indistinguishable domains and larger positive values as a greater distributional gap.

For our specific case, we compute $\mathrm{MMD}^2(k, i, p)$ to measure how far the representation of task T_i departs from the ID manifold when it is evaluated under perturbation p at checkpoint θ_k. Because MMD compares sets of samples evaluated by the same policy, we do not need teacher forcing, so we measure MMD under normal closed-loop evaluation and directly compare the resulting pooled representation vectors of different scenes.

MMD assumes i.i.d. data, so comparing representations every time the model activates would make all frames strongly temporally correlated and violate the assumption. To mitigate the inherent temporal correlation of frames, each rollout only contributes with a fixed number of K frames. Let T be the number of total frames of a rollout, the number of sampled frames is defined by $K = \mathrm{clip}(\lfloor T/20 \rfloor, 6, 12)$, which essentially ensures one frame per second of simulation. The 6 and 12 limits prevent longer episodes from disproportionately dominating the overall distribution.

Linear Probes

Linear probes [36] are linear regressors trained on top of the frozen, extracted representations to predict a specific target variable. Because the probe is only a linear mapping, it serves as a diagnostic tool for identifying what information the representations linearly encode. Based on the probe's R^2 score, we obtain a quantitative measure of how much target information the representation contains.

In our case, the tested probe targets are the 3D positions of the manipulated object and the 3D end-effector (wrist). Since these are regression targets, R^2 is not bounded below because when the probe extrapolates, residual error can dominate total variance and drive the metric to large negatives. For practical reasons, when this happens, we floor and report the R^2 at -1, clearly indicating that the metric has been floored.

Linear probes use ordinary least squares with no ℓ_2 penalty, evaluated with 5-fold cross-validation grouped by episode/scene. The cross-validation partition is done by full episodes (scenes) rather than individual frames.

E OOD Acquisition per Perturbation Category

Looking at OOD at acquisition (OOD-Acq) is the most reliable way to measure how the perturbations directly affect OOD performance because it is not influenced by the confounding effect of training on any posterior task. A drop from LIBERO to LIBERO-Plus then indicates a pure domain-shift penalty, disentangled from forgetting.

Table 5: OOD-Acq: mean diagonal success rate SR(k, k) over the ten tasks, for LIBERO and each LIBERO-Plus category. Colored arrows show the change relative to LIBERO.

Suite	LIBERO	LIBERO-Plus Mean	Camera	Light	Bckg.	Noise	Robot	Layout	Lang.
Spatial	0.79	0.52	0.46	0.86	0.51	0.61	0.26	0.54	0.41
		↓0.27	↓0.33	↑0.07	↓0.28	↓0.18	↓0.53	↓0.25	↓0.38
Object	0.91	0.64	0.68	0.87	0.74	0.76	0.21	0.61	0.57
		↓0.27	↓0.23	↓0.04	↓0.17	↓0.15	↓0.70	↓0.30	↓0.34
Goal	0.72	0.47	0.48	0.63	0.55	0.64	0.26	0.45	0.25
		↓0.25	↓0.24	↓0.09	↓0.17	↓0.08	↓0.46	↓0.27	↓0.47
LIBERO-10	0.56	0.18	0.09	0.38	0.10	0.16	0.22	0.21	0.11
		↓0.38	↓0.47	↓0.18	↓0.46	↓0.40	↓0.34	↓0.35	↓0.45

Table 5 reports the average diagonal SR for each of the suites on ID evaluation, all the OOD perturbations, and the mean OOD-Acq. Domain shift generally imposes a consistent penalty across all suites, resulting in a significant drop in average performance. Notably, the relative degradation grows with baseline difficulty: the hardest in-domain suite, LIBERO-10, loses about 68% of its ID success, versus 30–35% in the other suites, consistent with poor model understanding of the tasks, which makes the harder OOD environment even more challenging.

F Low-Sample LIBERO-Plus Cells

Within a suite, most evaluation tasks contain a healthy number of scenes, typically ranging from 30 to 50. However, the distribution of LIBERO-Plus test scenes is not uniform across tasks. A small subset of cells exhibits a lower sample count (10 or fewer scenes), occurring almost exclusively within the Background and Light perturbations. The minimum in the dataset is $N = 4$ scenes per task, occurring only in four specific instances in the light perturbation (Spatial task 8, Object task 4, Goal task 1, LIBERO-10 task 2). Another few tasks in the background contain only 5 to 10 scenes, and Spatial task 0 for camera contains 6 scenes. Table 6 enumerates them.

The variance these concrete cases can induce is largely mitigated for most metrics and results because we compare across tasks or perturbations and pool results. Where a numerical result isolates a single low-N cell—for example, the Light ACA peak in LIBERO-10 around task 2, which has only 4 scenes—that spike should be read as a small-sample artifact rather than genuine recovery, and the localized uncertainty should be taken into account when interpreting it.

Table 6: LIBERO-Plus evaluation cells with $N \leq 10$ scenes.

Suite	Perturbation	Task	N
Spatial	Light	8	4
Object	Light	4	4
Goal	Light	1	4
LIBERO-10	Light	2	4
Object	Background	4	5
Spatial	Camera	0	6
Spatial	Background	8	6
Goal	Background	1	6
LIBERO-10	Background	2	8
Spatial	Background	1	10
Spatial	Light	1	10

G Additional Behavioral, Coherence, and Probe Figures

In Section 5, we aggregate the LIBERO-Plus success rates over the perturbation categories. Figure 7 shows how each task column evolves under LIBERO and under every perturbation category in LIBERO-Plus individually.

We also reported the mean cross-perturbation coherence but did not show the full correlations; those can be found in Figure 5 (Spatial and Object SR profiles) and Figure 6 (full suite×metric matrices).

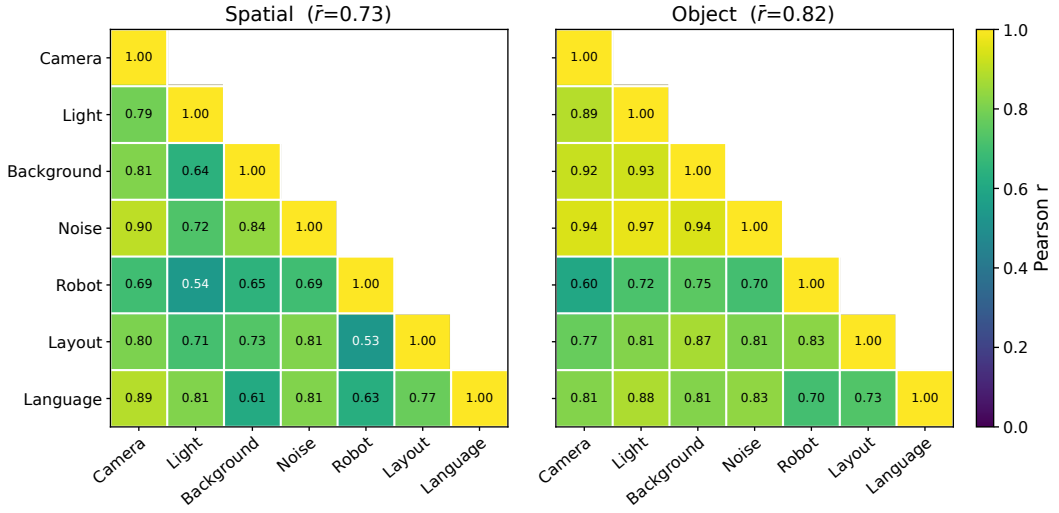


Figure 5: **Pairwise Pearson correlation of under-diagonal SR profiles across perturbation categories (Spatial and Object).** Each entry is the correlation between the flattened retention cells ($k > i$) of two LIBERO-Plus categories. The uniformly positive values mean that categories share the same pattern of active versus collapsed cells and differ mainly in scale.

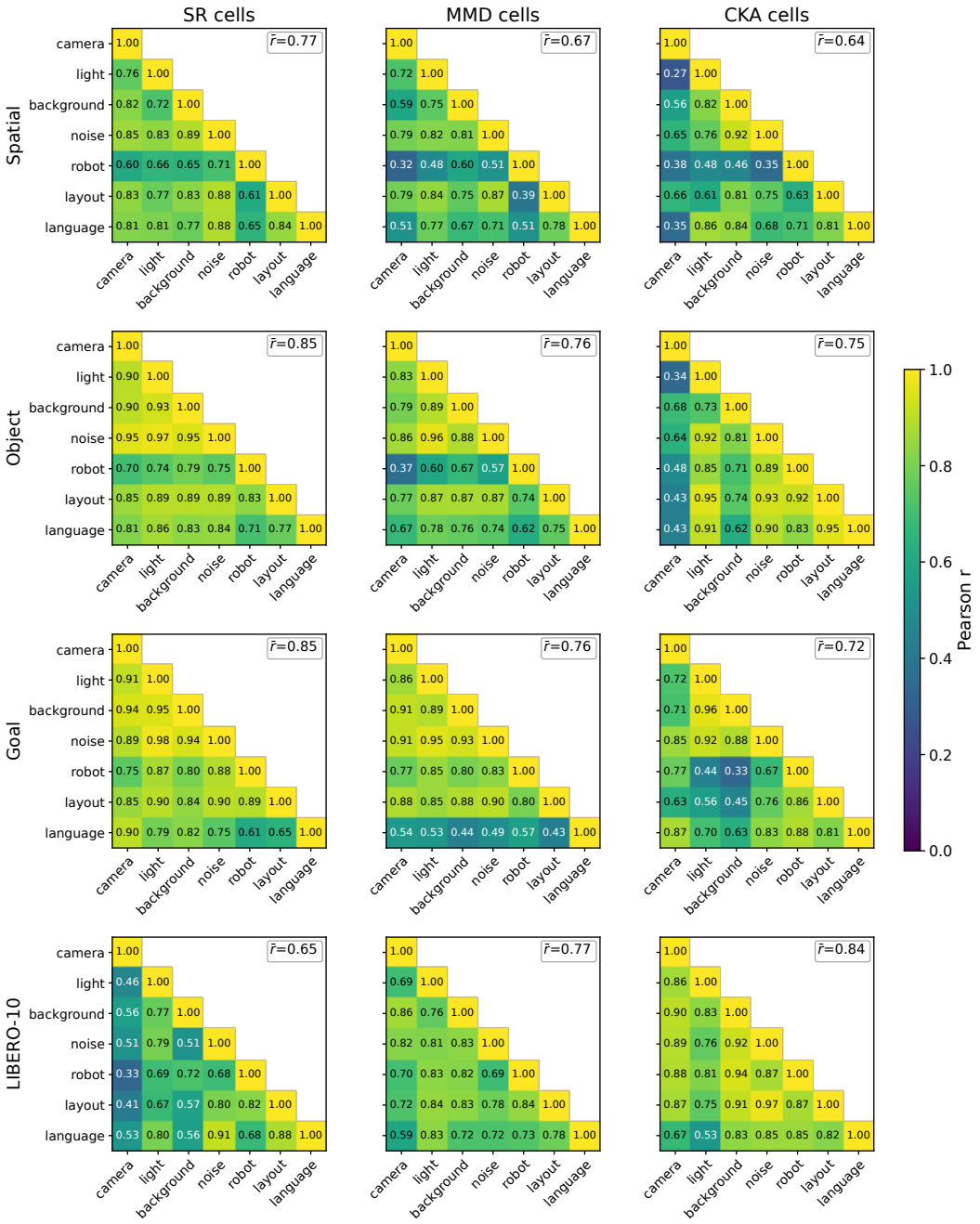


Figure 6: **Full pairwise Pearson correlation of flattened lower-triangular cells across Plus categories.** Rows: suites; columns: success-rate (SR), multimodal ID–OOD MMD^2, and multimodal CKA profiles. Each inset reports the mean pairwise $\bar{r}$. Unlike Figure 5, which only uses the under-diagonal cells ($k > i$), the SR and MMD^2 panels also include the diagonal ($k \geq i$), so the SR values differ slightly. The CKA panel uses the under-diagonal cells only ($k > i$), as it compares θ_k with θ_i on T_i, which is undefined on the diagonal. For Goal and LIBERO-10, most under-diagonal SR cells are zero, so their SR $\bar{r}$ is mainly driven by the diagonal and should be read with care.

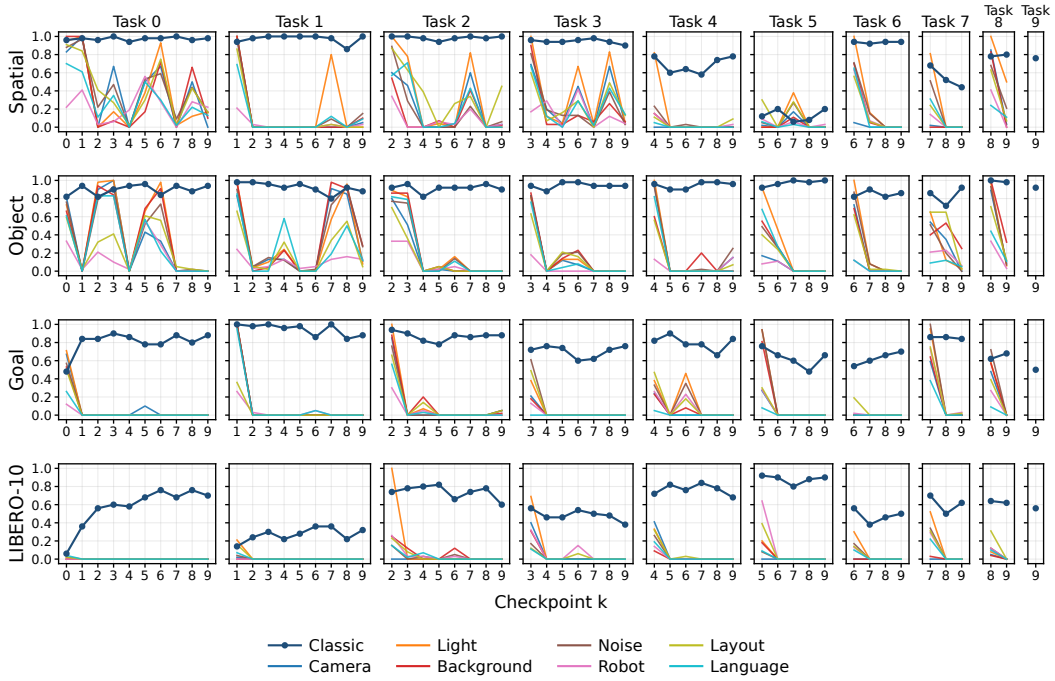


Figure 7: **Per-task column trajectories of success rate under LIBERO and each LIBERO-Plus perturbation.** Each panel tracks task T_i over checkpoints $k \geq i$ (panel width proportional to remaining checkpoints). Dark blue: in-domain LIBERO; other colors: the seven LIBERO-Plus categories. Complements the category-mean heatmaps in the main text by showing per-perturbation collapse and recovery along each task column.