LIMITED LINGUISTIC DIVERSITY IN VLA TRAINING DISTORTS LANGUAGE REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) models inherit rich language representations from pretraining, yet remain brittle to variation in natural-language commands. We study how this brittleness is reflected in the representations themselves. We first show that the Open X-Embodiment Magic Soup mixture relies on a strikingly small and repetitive vocabulary that does not cover the natural variation in robot commands: even simple paraphrases of benchmark instructions often contain words absent from it. We then trace the language embeddings of OpenVLA and π_0-FAST through vision-language and robot training. Robot training changes the embeddings considerably more than vision-language training does, and in both models these changes carry a direct imprint of the data. For OpenVLA, whose training data is public and whose embeddings are untied, only tokens that appear in the training commands change; among them, tokens appearing in more varied commands change less, while tokens seen in only a few commands can still be substantially reshaped. For π_0-FAST, whose training data is not public, the most heavily trained tokens stand out clearly enough from the rest of the vocabulary that we can partially recover its training vocabulary. Rather than adapting the language space as a whole, robot training thus specializes it to the narrow set of commands observed during training. This specialization also shows in how robust VLA models are to small input corruption: a single-character typo in an object name, which preserves the meaning of a command but changes its tokens, alters VLA representations substantially more than those of the corresponding language and vision-language models, suggesting that VLA representations depend more on the exact tokens than on the intended meaning. Together, these results link the narrowness of robot language data to changes in the VLA representations.

1 INTRODUCTION

Vision-language-action (VLA) models extend pretrained vision-language models (VLMs) to predict robot actions, often by treating actions as additional tokens or outputs conditioned on visual observations and natural-language instructions. This design offers a promising path toward generalist robot policies that transfer to new objects, skills, environments, and embodiments (Collaboration et al., 2023; Intelligence et al., 2025; Zhao et al., 2023; NVIDIA et al., 2025; Zheng et al., 2026): instead of learning each task in isolation, a VLA model can inherit broad visual and linguistic representations from large-scale pretraining and adapt them to embodied interaction.

However, recent work has shown that VLA models remain brittle to variation in language commands (Gao et al., 2025; Wang et al., 2024; Moskalenko et al., 2026; Kim et al., 2026b; Hou & Zhao, 2026), and several approaches have begun to explore this limitation via better action representation, input augmentation, co-training, and related interventions (e.g., Zhou et al., 2025b; Hancock et al., 2026; Hou & Zhao, 2026; Grover et al., 2025). In addition, Wanna et al. (2026) document the limited linguistic diversity of open-source datasets used to train VLA models, showing strikingly low numbers of unique words in the VLA training datasets (Collaboration et al., 2023). Few studies examine how adapting VLMs with limited robot-language data affects the inherited LLM representations, and without understanding what adaptation does to the representations it modifies, it is hard to know where to intervene.

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

In this work, we show that the linguistic coverage of robot training data is limited and that this leaves a measurable imprint on the embedding space itself. Extending the analysis of Wanna et al. (2026) from four datasets to the full OXE Magic Soup+ mixture used to train OpenVLA, we find that its almost one million trajectories contain fewer than 200K unique commands and fewer than 4,500 unique words, whereas even instruction-following datasets, which are more limited than the material LLM tokenizers are trained on, contain over 15K unique words (Wanna et al., 2026). This narrow vocabulary is further amplified by repeated exposure, as OpenVLA is trained for 27 epochs on this mixture (Kim et al., 2024), whereas VLM training typically covers its data only once or twice (Karamcheti et al., 2024). Tracking language-token embeddings across the LLM→VLM→VLA adaptation chain in OpenVLA and π_0-FAST, we find that:

- VLA training changes token embeddings considerably more than VLM training, and these changes are concentrated in a small fraction of the vocabulary: in OpenVLA, only tokens appearing in the training commands, roughly 11% of the vocabulary, are updated at all, and in π_0-FAST we estimate that fewer than 3% of language tokens are seen in training.
- In OpenVLA, where embedding changes can be related directly to the training data, frequency alone does not explain how much a token shifts: frequent tokens used across many distinct commands change less, while tokens seen in only a few commands shift more.
- In π_0-FAST, where tied embeddings expose every token to gradient, softmax pressure shifts nearly all tokens along a single shared direction, as expected; the tokens that deviate from it are those most shaped by robot training, and this separation is clear enough to partially recover the model's undisclosed training vocabulary.

These changes show that in-dataset embeddings are adapted jointly with the rest of the model during VLA training, while the remaining embeddings stay close to their pretrained representations. As a result, the model becomes tuned to a small set of adapted tokens, and inputs outside this set are represented by embeddings that were not updated alongside it.

We also test whether this specialization affects how VLA models handle commands that deviate only slightly from their training data. To isolate such deviations, we use single-character typos in object names, which preserve the meaning of a command while changing its tokens. We find that VLA training makes internal representations markedly more sensitive to these minimal corruptions: while the corresponding LLMs and VLMs represent corrupted and original commands almost identically, VLA models map them to substantially different representations.

Together, these results trace a chain from data to model behavior. First, the language in open-source robot datasets is narrow: a small, repetitive vocabulary seen many times during training. Second, VLA training adapts the pretrained language space unevenly: tokens in this vocabulary are reshaped around the limited contexts in which they appear, while the rest of the vocabulary receives little adaptation signal. Third, the way the model uses its tokens changes as well: after VLA training, the representation of a command depends more on its exact tokens, so even minimal deviations from familiar wording are processed differently. This chain offers a representation-level perspective on the language brittleness of VLA models and motivates more careful choices about data coverage, action-token design, and adaptation strategies in future VLA training.

2 Related Work

Recent work has increasingly shown that the language-conditioning abilities of VLA models remain fragile, even when the models are built on strong pretrained VLM backbones. Much of this evidence comes from simulation environments such as LIBERO (Liu et al., 2023a). Extensions such as LIBERO-Para, LIBERO-Plus, and LIBERO-Pro (Kim et al., 2026b; Fei et al., 2025; Zhou et al., 2025a) use this setting to evaluate language robustness, finding that current models often fail to respond appropriately to instruction variation and, in some cases, behave more like vision-action policies than genuinely language-conditioned ones. Other simulation-based studies show that VLA models can be sensitive to irrelevant linguistic context (Moskalenko et al., 2026), struggle with paraphrased instructions and cluttered scenes (Wang et al., 2024; Hou & Zhao, 2026), and fail to reliably infer user intent from language (Fang et al., 2025). Complementing these simulation studies, Gao et al. (2025) conduct a real-robot evaluation and find that semantic generalization remains

limited despite the use of language-model backbones, with models failing on instruction rephrasing, object-property understanding, spatial relations, and broader world knowledge. Together, these studies establish that VLA language conditioning is brittle at the behavioral level; however, they leave open how VLA adaptation changes the inherited language representations that these models rely on. Existing interpretability work on VLAs has examined scene shortcuts (Grant et al., 2026), representation steering (Häon et al., 2025; Swann et al., 2026; Kim et al., 2025; Miao et al., 2026), and related internal mechanisms, but has not examined how the language representations inherited from pretraining are reshaped by VLA adaptation itself.

3 Linguistic Diversity in Open X-Embodiment

Recent work by Wanna et al. (2026) examined the linguistic properties of four major datasets from the OXE collection (Collaboration et al., 2023), which together account for approximately 33% of the OpenVLA (Kim et al., 2024) training mixture according to the reported mixture composition[1]. They found a striking lack of linguistic diversity: two of the four datasets contain fewer than 100 unique words in total. As this kind of extreme repetitiveness can influence the training, we extend this language-statistics analysis to the full OpenVLA training mixture and measure unique words and commands in all the Magic Soup datasets (Table 1 and 4 in Appendix).

Table 1: Top-10 datasets by mixture proportion in the OpenVLA training data mixture, with remaining 17 datasets aggregated as "Other." QT-Opt filtered by success only.

Dataset	Mix %	Eps	Steps	Empty Eps	Uniq cmds	Uniq words
Bridge (Walke et al., 2024a)	13.3%	60K	2.3M	16K	22,198	2,484
RT-1 (Brohan et al., 2023)	12.7%	87K	3.8M	8	578	49
QT-Opt (Kalashnikov et al., 2018)	12.7%	210K	2.5M	0	1	2
DROID (Khazatsky et al., 2024)	10.0%	92K	27.0M	42K	31,303	1,963
BC-Z (Jang et al., 2021)	7.5%	43K	6.0M	0	104	51
FMB Dataset (Luo et al., 2024)	7.1%	9K	1.1M	0	64	31
Language Table (Lynch et al., 2022)	4.4%	442K	7.0M	0	127,583	928
Stanford Hydra (Belkhale et al., 2023)	4.4%	1K	0.4M	0	3	17
Taco Play (Rosete-Beas et al., 2022)	3.0%	4K	0.2M	0	406	74
Furniture Bench (Heo et al., 2023)	2.4%	5K	3.9M	0	9	10
Other (17 datasets)	22.5%	16K	3.3M	0	230	572
Entire mixture	100.0%	969K	57.6M	59K	182,421	4,477

Despite its scale, with nearly one million episodes and over 50 million steps, the mixture contains fewer than 4,500 unique words across all included datasets. Only slightly more than 1,000 appear in more than one dataset, with three datasets accounting for most of the vocabulary. Around 2,500 appear in more than one unique command, and around 1,000 in more than ten (Figure 5, Appendix A). Some of the remaining words stem from typos and a small number of non-English commands in Bridge. The vocabulary available during VLA adaptation is thus not only much smaller than that of standard instruction-following data but also highly uneven: most words are specific to a single dataset and occur in limited linguistic contexts.

VLA models aim to leverage the language representations acquired by pretrained and instruction-tuned language models, whose tokenizers are designed to support a broad range of linguistic variation. However, only a small subset of the tokenizer's vocabulary is likely to be encountered during VLA adaptation. We refer to this subset as the *active vocabulary*: the tokens that receive a direct training signal. This distinction matters because the nominal vocabulary can be orders of magnitude larger than the set of tokens actually updated during VLA training.

It is possible, though, that the robotics instruction domain is narrow enough for the active vocabulary to cover most of the language needed. As a sanity check, we examine LIBERO-Para (Kim et al., 2026b), which provides paraphrases of the 10 LIBERO-Goal commands. We find that 49% of the paraphrases contain at least one token absent from the OXE Magic Soup mixture. This is notable

[1] https://github.com/openvla/openvla/blob/main/prismatic/vla/datasets/rlds/oxe/mixtures.py

because the paraphrases introduce no new objects, environments, or skills; they vary only wording and structure. The gap widens when objects vary: "apple" occurs in Magic Soup, but "kiwi" does not, so a kiwi instruction can fall outside the training vocabulary even when its apple counterpart is covered. The active vocabulary therefore does not exhaust the linguistic variation that arises even within robot instructions.

VLA adaptation thus operates in a substantially different linguistic regime from pretraining. Rather than adapting language representations across diverse contexts, the model receives direct signal for only a small subset of its vocabulary, under highly constrained and repetitive supervision. Understanding how this narrow regime reshapes pretrained token representations is therefore important for characterizing how language representations are transferred and modified during VLA adaptation.

4 Model Factors Shaping Embedding Adaptation

Embeddings are a central component of LLM-based VLA models: they are shaped by architectural choices, including tokenizer design and embedding sharing. These factors determine which tokens receive a training signal and how changes in embedding geometry should be interpreted. Before analyzing embedding drift, we therefore compare the model and data properties of two autoregressive VLA models: OpenVLA (Kim et al., 2024) and π_0-FAST (Pertsch et al., 2025) (see an overview in Table 2).

Table 2: Comparison of OpenVLA and π0-FAST

Model	LLM	VLM	Action	Lang vocab	Act vocab	Embeddings	Data
OpenVLA	Llama 2 7B	Prismatic	Bins (single-action)	32k	256	Untied	OXE
π0-FAST	Gemma 2 2B	PaliGemma	FAST (chunked)	256k	2048	Tied	Unknown

We focus on these models because they represent two prominent open-weight autoregressive VLA models with different embedding-matrix structures. OpenVLA builds on a Llama 2/Prismatic VLM backbone (Touvron et al., 2023; Karamcheti et al., 2024), whereas π_0-FAST builds on a Gemma 2/PaliGemma (Team et al., 2024; Beyer et al., 2024). We choose π_0-FAST from the π family because its model weights are available and, unlike $\pi_{0.5}$ (Intelligence et al., 2025), it does not include additional training beyond VLA conditioning that could further affect the LLM backbone, making it a cleaner comparison point for studying VLA adaptation in isolation. Restricting the comparison to autoregressive policies further reduces variation from different action-generation paradigms, which can independently affect language representations (Driess et al., 2025).

4.1 Tied vs Untied Embeddings

When analyzing changes in language tokens, another important distinction concerns where the embeddings are used. Language models both use language tokens as inputs and outputs; these tokens serve as the language model's head for next-token prediction. In language modeling, tying these weights and having the same embedding matrix for both has often been used to reduce parameters and improve training (Press & Wolf, 2017; Inan et al., 2017). This strategy is chosen by several modern LLMs, including Gemma models (Team et al., 2024) that serve as the backbone for the π_0-FAST model. However, some literature has also expressed a different opinion (Lopardo et al., 2026; Chung et al., 2021), and some LLMs take an approach of untying the embeddings; one example is the Llama 2 model (Touvron et al., 2023) that OpenVLA is built upon.

From an empirical perspective, Bertolotti & Cazzola (2024) argue that input embeddings and output embeddings encode related but distinct similarity structures: input embeddings tend to reflect semantic similarity, whereas output embeddings reflect conditional equivalence, i.e., similarity in the contexts in which tokens are predicted. Weight tying is therefore most beneficial when these two structures align, as they often do in language modeling, where words with similar meanings also tend to appear in similar contexts. Conversely, weight tying can be harmful when this alignment does not hold, as may be the case in VLA adaptation, where language tokens are never predicted and are shaped on the output side only through negative softmax pressure.

4.2 Action Tokenization

Although both VLA models considered here predict actions autoregressively, they use different action tokenization schemes. OpenVLA adds discrete action tokens corresponding to bins over normalized action values in the range $[-1, 1]$. π_0-FAST, in contrast, uses the FAST tokenizer (Pertsch et al., 2025), which represents short action chunks in the frequency domain and compresses them into tokens that capture recurring multi-step movement patterns rather than fixed bins for individual action dimensions. See more discussion in Appendix D.

Predicting action chunks improves inference efficiency, but it also reduces the number of language-conditioned prediction steps per fixed action samples, meaning the language representations receive a less frequent gradient signal relative to single-step prediction. This matters because VLA training already differs from standard VLM adaptation: VLMs are typically trained on more linguistically diverse supervision and often for only one or two passes over the data (Karamcheti et al., 2024). OpenVLA, by contrast, fine-tunes the Prismatic backbone for 27 epochs on a robot-data mixture with relatively repetitive language (Kim et al., 2024). This creates stronger pressure for the language representations to specialize to the limited command distribution observed during VLA training.

5 Language Embedding Geometry: LLM → VLM → VLA

We compare changes in language token embeddings across two stages: incorporating visual input into an LLM during VLM training, and extending the resulting VLM to predict actions during VLA training. While VLM training is generally thought to preserve linguistic robustness, this robustness appears to weaken during VLA adaptation (Gao et al., 2025; Wang et al., 2024; Moskalenko et al., 2026). We first characterize overall embedding drift across the vocabulary, then examine which tokens change and how their shifts relate to training data frequency, and finally probe whether these embedding-level changes correspond to increased sensitivity to minimal instruction perturbations.

5.1 Global Embedding Changes

We compute the percentage of tokens whose embeddings changed and report three complementary measures for changed tokens: cosine distance (directional change), L2 shift (absolute displacement), and signed norm difference (magnitude increase or decrease). A similar methodology is also used to detect undertrained tokens, the ones that receive the least training signal (Land & Bartolo, 2024).

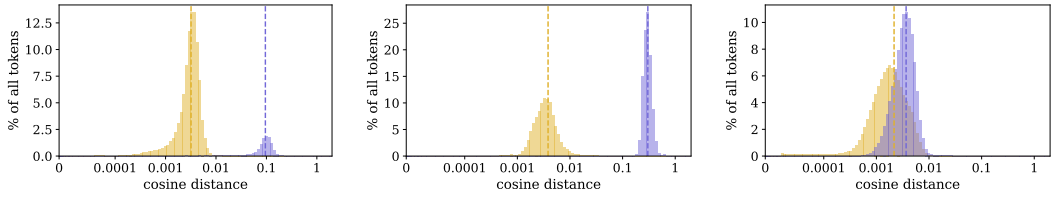
In OpenVLA, untied input embeddings receive gradient only from tokens present in the input, so by construction, only tokens in the instructions (3,411 tokens, $\sim$11% of the vocabulary) can change. The architecture does not, however, determine how far these tokens move. They shift substantially more than during VLM adaptation (Table 3; Figure 1a), although the two stages also differ in training length. The result is a two-tier input embedding matrix whose adapted tier is bounded by the linguistic coverage of the training data: tokens outside it, including paraphrase words absent from Magic Soup, retain representations that were never adapted alongside the rest of the model. This is an embedding-level analog of feature distortion, in which fine-tuning changes representations associated with the fine-tuning distribution more than those outside it (Kumar et al., 2022).

The OpenVLA output embeddings follow a different pattern. Every row of the language-model head receives gradient through the softmax, even though text tokens are never targets during VLA training. This is the known representation-degeneration effect (Gao et al., 2019), and it accounts for the broad geometric compression we observe: mean pairwise cosine distance drops from 0.96 to 0.53 after VLA adaptation (Appendix C).

In π_0-FAST, all tied embedding vectors receive gradients at least from the softmax in the output head, thus π_0-FAST updates all 100% of the embedding parameters (Table 3). The language embeddings get gradients both from participating as input tokens in the prompt and from participating in the output softmax, even though instruction-text tokens are never selected as targets. π_0-FAST's VLA adaptation also produces larger changes than the preceding VLM-adaptation stage, but the difference is less pronounced than in OpenVLA. This may be partly explained by the more anisotropic embedding space of the Gemma family: its mean pairwise cosine distance is around 0.17, compared with around 0.98 for the Llama-2 input embeddings. In this more clustered geometry, small token-wise cosine distances are expected even when the embeddings undergo non-trivial shifts.

Table 3: Summary of language embedding changes across two VLA model families. Llama/OpenVLA uses untied input and output embeddings, while Gemma/π_0-FAST uses a single tied embedding matrix. Cosine distances, L2 displacements, and norm differences are computed over changed tokens only.

Metric	Llama 2 → OpenVLA				Gemma 2 → π_0-FAST	
	Input Embeddings		Output Embeddings		Tied Embeddings	
	LLM→VLM	VLM→VLA	LLM→VLM	VLM→VLA	LLM→VLM	VLM→VLA
Changed tokens	95.8%	10.7%	100.0%	100.0%	100.0%	100.0%
Mean cos. dist.	0.003	0.095	0.004	0.302	0.002	0.004
Mean L2 shift	0.085	0.526	0.091	0.985	0.449	0.628
Mean norm diff.	0.004	0.205	0.000	0.278	-0.001	0.020



(a) Llama 2 → OpenVLA (input) (b) Llama 2 → OpenVLA (output) (c) Gemma 2 → π_0-FAST (tied)

Figure 1: Cosine distances between the LLM and VLM embeddings (yellow) and VLM and VLA model embeddings (purple).

5.2 Which Tokens Are Reshaped by VLA Training?

For OpenVLA, we reconstruct the OXE Magic Soup training mixture and compute token frequencies from extracted language instructions using the Llama tokenizer, then compare these frequencies with embedding change and number of unique commands token appears in. For π_0-FAST, the full training mixture is not publicly available and tied embeddings prevent separating input- and output-side effects, so we instead analyze the shared embedding matrix through L2 norm changes and cosine distances, inspecting tokens with the largest shifts.

In OpenVLA, token frequency in the training mixture is not a direct predictor of embedding change. Instead, the relationship between frequency and the L2 distance between the VLM and VLA embeddings is non-monotonic (Figure 2). Notably, many tokens that occur in only one or a few unique commands still undergo considerable embedding change, showing that substantial shifts can occur even under very limited supervision.[2] Tokens in the long tail of the frequency distribution generally receive smaller updates than more frequently observed tokens, but frequency alone does not determine the magnitude of the change. At the same time, the most frequent tokens tend to be more stable than tokens in the middle of the frequency range. These highly frequent tokens also tend to occur across a larger and more diverse set of instructions, suggesting that repeated exposure across contexts may be associated with greater representational stability. In contrast, mid-frequency tokens may receive enough updates to shift substantially, while still being exposed to a comparatively limited range of linguistic contexts. Overall, the results suggest that embedding drift reflects not simply how often a token is observed, but also the richness of the contexts in which it occurs: tokens can shift substantially from very limited supervision, whereas tokens that recur across many distinct instructions tend to remain more stable (see Appendix F for token examples).

With π_0-FAST, most tokens remain close to their original positions and show only small norm and direction shifts (Table 3). If we examine the change vectors, we find that most of them align in a common direction. We find the mean common direction by normalizing all vectors and finding

[2]Here we use unique commands as a proxy for context diversity, however the commands can be very similar among each other (Wanna et al., 2026).

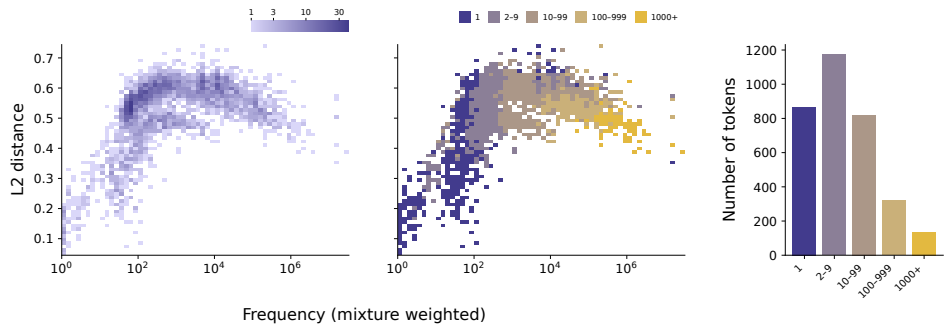
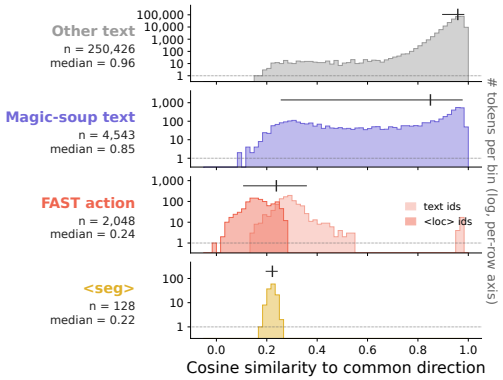
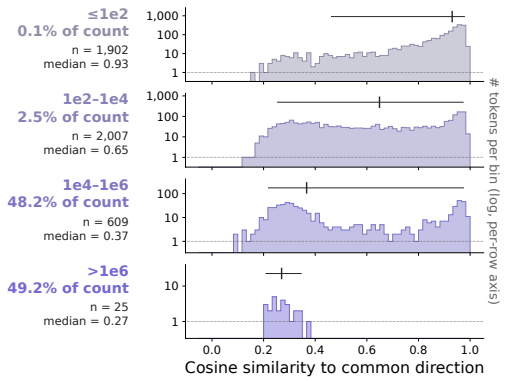


Figure 2: Mixture-weighted token counts versus L2 distance between Prismatic and OpenVLA embeddings: token density (left), average number of unique commands per token (middle), and number of tokens per unique-command tier (right). Many tokens appear in only one or a few commands yet still undergo considerable embedding changes, showing that tokens can shift significantly under limited supervision. Tokens that appear frequently and in more varied contexts remain more stable.



(a) Similarity to common direction over token type for all embedding change vectors.



(b) Similarity to common direction over mixture-weighted token frequency bins in Magic Soup.

Figure 3: Distributions of cosine similarity to common direction among PaliGemma $\rightarrow \pi_0$-FAST embedding change vectors. We differentiate between language tokens (those appearing in the Magic Soup dataset and all the rest), FAST action tokens (adapted from PaliGemma's <loc> and language tokens), and PaliGemma's <seg> tokens. Each plot presents logarithmically scaled counts and shows the sample median and the 0.05th and the 0.95th percentiles above the histograms.

the mean[3] and compute the cosine similarity between each change vector and the found common direction (Figure 3). 252,044 change vectors (98%) have cosine similarity to the common direction of more than 0.8. The non-aligned change vectors consist mostly of FAST action tokens — the only ones used as model outputs and PaliGemma-inherited <seg> tokens not used in π_0-FAST. Across the mixture-weighted count tiers in the Magic Soup data (Figure 3b), tokens with larger training exposure agree less with the common direction; the median of the distribution is moving closer to the non-agreeing part of the distribution as training frequency increases. These findings effectively disentangle input-level updates of used tokens from the output-only updates in tied embeddings and suggest that only a small partition of π_0-FAST's language tokens are meaningfully updated during VLA training, while the absolute majority of tokens are simply suppressed via softmax probability gradients, without real training exposure. In Appendix E, we present a similar plot for the OpenVLA output embeddings. With untied embeddings in OpenVLA, softmax outputs are the only source of gradient for these parameters, hence the demonstrated effect is even cleaner in that case: action tokens are the only tokens disagreeing with the common direction.

[3]This estimate is verified: an alternative iterative procedure of finding the consensual direction for a random subsample (RANSAC-style mean estimation with cosine distances) produced an estimate almost identical to the mean direction (cosine similarity 0.9998).

The observed bimodal similarity distribution enables us to pinpoint the other most-trained tokens in π_0-FAST's VLA adaptation and uncover the dataset distribution beyond Magic Soup. As shown in Figure 3a, a certain portion of other language tokens is also part of the misaligned cluster. These several hundred tokens (250 tokens with similarity below 0.5, and 948 below 0.8) mostly name food, household objects, etc. (e.g., *shampoo*, *taco*, *cashew*, *lipstick*, *screws*). By varying the similarity threshold for outliers, we roughly estimate the total number of tokens used as inputs to be below 7,000, less than $\sim$3% of all language tokens (see analysis and further examples in Appendix G).

Together, the two models show that VLA adaptation meaningfully reshapes only a narrow active vocabulary, with changes depending on contextual diversity rather than frequency alone. The remaining tokens are left untouched or, with tied embeddings, shifted uniformly under softmax pressure.

5.3 Representation Robustness to Minimal Noise

We examine whether the embedding-level changes observed above correspond to differences in how models handle minor variation in language commands. We focus on one-character corruptions of target object nouns: these perturbations preserve most of the intended task, but disrupt the regular tokenization of familiar object words and test whether the model's internal representations remain stable under small deviations from the training-command distribution. This is not intended as a comprehensive robustness benchmark, but as a controlled probe of how the LLM–VLM–VLA adaptation chain changes sensitivity to instruction variation.

We sample 1,000 episodes from the BridgeDataV2 validation set (Walke et al., 2024b) with non-empty English-language commands. For each command, we identify the target object noun and introduce a one-character corruption, which typically breaks the word into a less regular subword sequence. We extract activations from the last token of the autoregressive input encoding across all six checkpoints (LLM, VLM, and VLA for each model family) and compare representational geometry between original and corrupted commands at each layer (see more details in Appendix I).

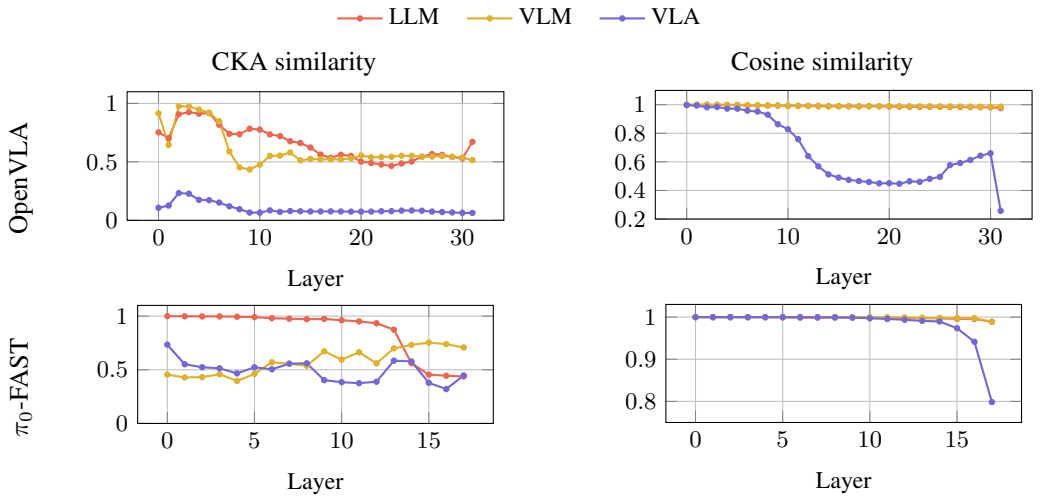


Figure 4: Average layer-wise sensitivity to input corruption across the LLM, VLM, and VLA stages for the Llama/OpenVLA and Gemma/π0-FAST model families. Higher values indicate more similar hidden representations between clean and corrupted inputs.

To estimate layer-wise sensitivity to input corruption, we compute two complementary similarity measures between representations of original and corrupted instructions at each layer. Centered-Kernel Alignment (CKA) measures alignment of representational geometry, with lower values indicating that the two conditions produce more geometrically distinct activation patterns. Cosine similarity provides a pointwise, scale-invariant measure that isolates directional changes in individual activation vectors, independent of their magnitudes.

As shown in Figure 4, CKA decreases most prominently in the final layers for VLAs relative to their VLM and LLM counterparts, while remaining comparable across earlier layers. The cosine

similarity analysis disambiguates the source of this drop: LLMs and VLMs maintain near-unity cosine similarity across all layers, indicating that their CKA decrease stems from magnitude scaling rather than any reorientation of the representations. VLAs, by contrast, show a progressive decline in cosine similarity in deeper layers, indicating that corruption induces both scaling and directional shifts in their activation vectors. Together, these results suggest that while all model types exhibit some representational change under corruption, only VLAs show sensitivity that manifests as a genuine geometric reorganization, consistent with tighter input-to-action mappings that are disrupted by even single-character perturbations in the command text. In Appendix J, we further validate on two established robotics benchmarks that the VLA model's representation drift in response to changes in text commands, results in a decrease in success rate for the downstream tasks.

6 Conclusion and Recommendations

We analyzed how VLA adaptation reshapes language-token embeddings in OpenVLA and π_0-FAST. Our results show that VLA training does not uniformly preserve or overwrite the pretrained language space, but instead concentrates changes around a narrow active adaptation vocabulary. In OpenVLA, this results in a two-tier input embedding matrix, where most tokens remain close to their VLM initialization while a small subset shifts substantially. In π_0-FAST, the largest changes similarly occur among tokens interpretable as relevant to embodied instruction-following, while the remaining majority of the language tokens are equally suppressed by the softmax outputs. Together, these findings suggest that VLA adaptation specializes pretrained language representations to the limited command distributions seen during robot training and makes the language distribution heterogeneous, offering an embedding-level perspective on the language brittleness observed in VLA models. These results suggest several practical directions for future VLA training.

- **Increasing linguistic diversity of the supervision signal:** the narrow active vocabulary observed in current robot datasets (Section 3 and Appendix A) points to the need for richer human annotations, targeted paraphrase augmentation, synthetic instruction generation, or co-training with multimodal reasoning and language-only data.
- **Freezing or regularizing language embeddings:** when VLA training data is linguistically repetitive, it may be useful to explore freezing or regularizing parts of the language embedding space during adaptation to avoid heterogeneity shown in Sections 5.1 and 5.2 and improve robustness to minor perturbations (Section 5.3 and Appendix I).
- **Adapting the output LM head:** VLA models output only action-related tokens, making logits for the rest of the vocabulary redundant. In untied-embedding models like OpenVLA, this stores unnecessary parameters and increases output-layer computations, whereas with tied embeddings, as in π_0-FAST, this implicitly updates all language tokens without any exposure to them as model inputs (Section 5.2).
- **Adapting VLA-specific components before full fine-tuning:** Action embeddings are learned during training and change substantially (Section 4.2, Appendix D). Training them first with a frozen backbone could reduce the number of updates the language model needs and avoid unnecessary representational shifts.

7 Limitations

We analyze released checkpoints rather than controlled training runs. VLA models differ in many aspects at once, including backbone architecture, tokenizer vocabulary, embedding tying, action tokenization, data mixture, training objective, and optimization setup. These factors make it difficult to localize any observed effect to a single cause. For this reason, we interpret our results as evidence of adaptation patterns rather than causal claims about one specific design choice.

Finally, our robustness experiment is intended as a representation-level probe, not a comprehensive behavioral evaluation. One-character corruptions of object nouns provide a controlled way to test sensitivity to small deviations from familiar commands, but they do not cover the full range of linguistic variation that matters for instruction-following robots. Future work should connect embedding-level diagnostics more directly to downstream behavior, ideally through controlled ablations over data diversity, freezing strategies, tokenizer choices, and adaptation schedules.

AI USE DISCLOSURE

In this work, we used generative AI tools to assist with drafting parts of the paper and editing text for cohesion and readability. This included condensing author-written paragraphs and proposing short connecting sentences and finding related work to our experimental setting. All suggestions were reviewed, revised where needed, and accepted or rejected by the authors.

We have not used generative AI tools to generate synthetic datasets, develop the conceptual framework, propose or refine hypotheses, design experiments, or interpret results. We have used AI code writing tools for drafting code, designing plots and reformatting them.

All data statistics, experimental results, and citations were produced and verified by the authors.

REFERENCES

Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. In *CVPR*, 2023.

Suneel Belkhale, Yuchen Cui, and Dorsa Sadigh. Hydra: Hybrid robot actions for imitation learning. *arxiv*, 2023.

Francesco Bertolotti and Walter Cazzola. By tying embeddings you are assuming the distributional hypothesis. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 3584–3610. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/bertolotti24a.html.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. PaliGemma: A versatile 3B VLM for transfer, October 2024. URL http://arxiv.org/abs/2407.07726. arXiv:2407.07726 [cs].

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023. URL https://arxiv.org/abs/2212.06817.

Lawrence Yunliang Chen, Simeon Adebola, and Ken Goldberg. Berkeley UR5 demonstration dataset. https://sites.google.com/view/berkeley-ur5/home.

Hyung Won Chung, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. Rethinking embedding coupling in pre-trained language models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=xpFFI_NtgpW.

Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea

Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Muhammad Zubair Irshad, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Mart'in-Mart'in, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Vitor Guizilini, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

Zichen Jeff Cui, Yibin Wang, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. From play to policy: Conditional behavior generation from uncurated robot data. *arXiv preprint arXiv:2210.10047*, 2022.

Shivin Dass, Jullian Yapeter, Jesse Zhang, Jiahui Zhang, Karl Pertsch, Stefanos Nikolaidis, and Joseph J. Lim. Clvr jaco play dataset, 2023. URL https://github.com/clvrai/clvr_jaco_play_dataset.

Danny Driess, Jost Tobias Springenberg, brian ichter, LILI YU, Adrian Li-Bell, Karl Pertsch, Allen Z. Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, and Sergey Levine. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=cb0xbZ3APM.

Irving Fang, Juexiao Zhang, Shengbang Tong, and Chen Feng. From intention to execution: Probing the generalization boundaries of vision-language-action models. *arXiv preprint arXiv:2506.09930*, 2025.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Jensen Gao, Suneel Belkhale, Sudeep Dasari, Ashwin Balakrishna, Dhruv Shah, and Dorsa Sadigh. A taxonomy for evaluating generalist robot policies, 2025. URL https://arxiv.org/abs/2503.01238. arXiv: 2503.01238 [cs.RO].

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tieyan Liu. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkEYojRqtm.

Kris Wu Ge Yan and Xiaolong Wang. ucsd kitchens Dataset. August 2023.

Bryce Grant, Xijia Zhao, and Peng Wang. Not all features are created equal: A mechanistic study of vision-language-action models. *CoRR*, abs/2603.19233, March 2026. URL https://doi.org/10.48550/arXiv.2603.19233.

Shresth Grover, Akshay Gopalkrishnan, Bo Ai, Henrik I. Christensen, Hao Su, and Xuanlin Li. Enhancing generalization in vision-language-action models by preserving pretrained representations, 2025. URL https://arxiv.org/abs/2509.11417.

Asher James Hancock, Xindi Wu, Lihan Zha, Olga Russakovsky, and Anirudha Majumdar. Actions as language: Fine-tuning VLMs into VLAs without catastrophic forgetting. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=sFO9d6XSlf.

Minho Heo, Youngwoon Lee, Doohyun Lee, and Joseph J. Lim. Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation. In *Robotics: Science and Systems*, 2023.

Yuchen Hou and Lin Zhao. Langgap: Diagnosing and closing the language gap in vision-language-action models, 2026. URL https://arxiv.org/abs/2603.00592.

Bear Häon, Kaylene Stocking, Ian Chuang, and Claire Tomlin. Mechanistic interpretability for steering vision-language-action models, 2025. URL https://arxiv.org/abs/2509.00328.

Hakan Inan, Khashayar Khosravi, and Richard Socher. Tying word vectors and word classifiers: A loss framework for language modeling. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=r1aPbsFle.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. BC-z: Zero-shot task generalization with robotic imitation learning. In *5th Annual Conference on Robot Learning*, 2021. URL https://openreview.net/forum?id=8kbp23tSGYv.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.

Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic VLMs: investigating the design space of visually-conditioned language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML'24*, pp. 23123–23144, Vienna, Austria, July 2024. JMLR.org.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. Droid: A large-scale in-the-wild robot manipulation dataset. 2024.

Chanyoung Kim, Minwoo Kim, Minseok Kang, Hyunwoo Kim, and Dahuin Jung. Libero-para: A diagnostic benchmark and metrics for paraphrase robustness in vla models. *arXiv preprint arXiv:2603.28301*, 2026a.

Chanyoung Kim, Minwoo Kim, Minseok Kang, Hyunwoo Kim, and Dahuin Jung. Libero-para: A diagnostic benchmark and metrics for paraphrase robustness in vla models, 2026b. URL https://arxiv.org/abs/2603.28301.

Ju-Young Kim, Ji-Hong Park, Myeongjun Kim, and Gun-Woo Kim. Explainable adversarial-robust vision-language-action model for robotic manipulation, 2025. URL https://arxiv.org/abs/2512.11865.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An Open-Source Vision-Language-Action Model, September 2024. URL http://arxiv.org/abs/2406.09246. arXiv:2406.09246.

Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=UYneFzXSJWh.

Sander Land and Max Bartolo. Fishing for magikarp: Automatically detecting under-trained tokens in large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 11631–11646, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.649. URL https://aclanthology.org/2024.emnlp-main.649/.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. October 2023a. doi: 10.48550/arXiv.2306.03310. URL http://arxiv.org/abs/2306.03310. arXiv:2306.03310 [cs].

Huihan Liu, Soroush Nasiriany, Lance Zhang, Zhiyao Bao, and Yuke Zhu. Robot learning on the job: Human-in-the-loop autonomy and learning during deployment. In *Robotics: Science and Systems (RSS)*, 2023b.

Antonio Lopardo, Avyukth Harish, Catherine Arnett, and Akshat Gupta. Weight tying biases token embeddings towards the output space, 2026. URL https://arxiv.org/abs/2603.26663.

Jianlan Luo, Charles Xu, Xinyang Geng, Gilbert Feng, Kuan Fang, Liam Tan, Stefan Schaal, and Sergey Levine. Multi-stage cable routing through hierarchical imitation learning. *arXiv pre-print*, 2023. URL https://arxiv.org/abs/2307.08927.

Jianlan Luo, Charles Xu, Fangchen Liu, Liam Tan, Zipeng Lin, Jeffrey Wu, Pieter Abbeel, and Sergey Levine. Fmb: a functional manipulation benchmark for generalizable robotic learning. *arXiv preprint arXiv:2401.08553*, 2024.

Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time, 2022. URL https://arxiv.org/abs/2210.06407.

Ajay Mandlekar, Jonathan Booher, Max Spero, Albert Tung, Anchit Gupta, Yuke Zhu, Animesh Garg, Silvio Savarese, and Li Fei-Fei. Scaling robot supervision to hundreds of hours with roboturk: Robotic manipulation dataset through human reasoning and dexterity. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1048–1055. IEEE, 2019.

Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Structured world models from human videos. *CoRL*, 2023.

Miranda Muqing Miao, Subin Kim, Brandon Yang, and Lyle Ungar. Contrastive conceptor activation steering (coast): Unlocking vision-language-action models through hidden states, 2026. URL https://arxiv.org/abs/2605.17144.

Andrey Moskalenko, Daria Pugacheva, Denis Shepelev, Andrey Kuznetsov, Vlad Shakhuro, and Elena Tutubalina. Bring the apple, not the sofa: Impact of irrelevant context in embodied ai commands on vla models. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pp. 840–860, 2026.

Soroush Nasiriany, Tian Gao, Ajay Mandlekar, and Yuke Zhu. Learning and retrieval from prior data for skill-based imitation learning. In *Conference on Robot Learning (CoRL)*, 2022.

NVIDIA, :, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots, 2025. URL https://arxiv.org/abs/2503.14734. arXiv: 2503.14734 [cs.RO].

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient Action Tokenization for Vision-Language-Action Models, January 2025. URL http://arxiv.org/abs/2501.09747. arXiv:2501.09747 [cs].

Ofir Press and Lior Wolf. Using the output embedding to improve language models. In Mirella Lapata, Phil Blunsom, and Alexander Koller (eds.), *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 157–163, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://aclanthology.org/E17-2025/.

Gabriel Quere, Annette Hagengruber, Maged Iskandar, Samuel Bustamante, Daniel Leidner, Freek Stulp, and Joern Vogel. Shared Control Templates for Assistive Robotics. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7, Paris, France, 2020.

Erick Rosete-Beas, Oier Mees, Gabriel Kalweit, Joschka Boedecker, and Wolfram Burgard. Latent plans for task agnostic offline reinforcement learning. 2022.

Saumya Saxena, Mohit Sharma, and Oliver Kroemer. Multi-resolution sensing for real-time control with vision-language models. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/forum?id=WuBv9-IGDUA.

Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On bringing robots home, 2023.

Rutav Shah, Roberto Martín-Martín, and Yuke Zhu. MUTEX: Learning unified policies from multimodal task specifications. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/forum?id=PwqiqaaEzJ.

Aiden Swann, Lachlain McGranahan, Hugo Buurmeijer, Monroe Kennedy III, and Mac Schwager. Sparse autoencoders reveal interpretable and steerable features in vla models, 2026. URL https://arxiv.org/abs/2603.19183.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Jörn Vogel, Annette Hagengruber, Maged Iskandar, Gabriel Quere, Ulrike Leipscher, Samuel Bustamante, Alexander Dietrich, Hannes Hoeppner, Daniel Leidner, and Alin Albu-Schäffer. Edan - an emg-controlled daily assistant to help people with physical disabilities. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale, 2024a. URL https://arxiv.org/abs/2308.12952.

Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. BridgeData V2: A Dataset for Robot Learning at Scale, January 2024b. URL http://arxiv.org/abs/2308.12952. arXiv:2308.12952 [cs].

Zhijie Wang, Zhehua Zhou, Jiayang Song, Yuheng Huang, Zhan Shu, and Lei Ma. LADEV: a language-driven testing and evaluation platform for vision-language-action models in robotic manipulation, 2024. URL https://arxiv.org/abs/2410.05191. arXiv: 2410.05191 [cs.RO].

Selma Wanna, Agnes Luhtaru, Jonathan Salfity, Ryan Barron, Juston Moore, Cynthia Matuszek, and Mitch Pryor. Limited Linguistic Diversity in Embodied AI Datasets, 2026. URL https://www.semanticscholar.org/paper/b1bafe4f1bf70db30adb58fc68ebca2518a0d50d. arXiv:2601.03136.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.016.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Tai Wang, Ya-Qin Zhang, Jingjing Liu, and Xianyuan Zhan. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kt51kZH4aG.

Gaoyue Zhou, Victoria Dean, Mohan Kumar Srirama, Aravind Rajeswaran, Jyothish Pari, Kyle Hatch, Aryan Jain, Tianhe Yu, Pieter Abbeel, Lerrel Pinto, Chelsea Finn, and Abhinav Gupta. Train offline, test online: A real robot learning benchmark. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. Libero-pro: Towards robust and fair evaluation of vision-language-action models beyond memorization. *arXiv preprint arXiv:2510.03827*, 2025a.

Zhongyi Zhou, Yichen Zhu, Minjie Zhu, Junjie Wen, Ning Liu, Zhiyuan Xu, Weibin Meng, Ran Cheng, Yaxin Peng, Chaomin Shen, and Feifei Feng. ChatVLA: Unified multimodal understanding and robot control with vision-language-action model, 2025b. URL https://arxiv.org/abs/2502.14420. arXiv: 2502.14420 [cs.RO].

Xinghao Zhu, Ran Tian, Chenfeng Xu, Mingyu Ding, Wei Zhan, and Masayoshi Tomizuka. Fanuc manipulation: A dataset for learning-based manipulation with fanuc mate 200id robot, 2023.

Yifeng Zhu, Abhishek Joshi, Peter Stone, and Yuke Zhu. Viola: Imitation learning for vision-based manipulation with object proposal priors. *6th Annual Conference on Robot Learning (CoRL)*, 2022a.

Yifeng Zhu, Peter Stone, and Yuke Zhu. Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation. *IEEE Robotics and Automation Letters*, 7(2):4126–4133, 2022b.

A OXE MAGIC SOUP AUDIT

We further analyze the lexical distribution of the OXE Magic Soup+ mixture used for VLA training. We extract commands from the RLDS files, count unique commands without preprocessing, and compute unique words by removing punctuation and splitting on spaces. Although the mixture contains close to one million episodes, only around 180k episodes have unique language commands, and several datasets contain very few unique instructions. For example, Kuka accounts for a substantial part of the OpenVLA training mixture but contains only one command, *pick anything*; many other datasets contain fewer than 10 unique commands. This shows that the apparent scale of the dataset does not directly translate into linguistic diversity.

Table 4: Per-dataset breakdown of all 27 datasets in the OpenVLA `oxe_magic_soup_plus` mixture. Eps and Steps are exact counts. Empty denotes episodes with no extracted language instruction.

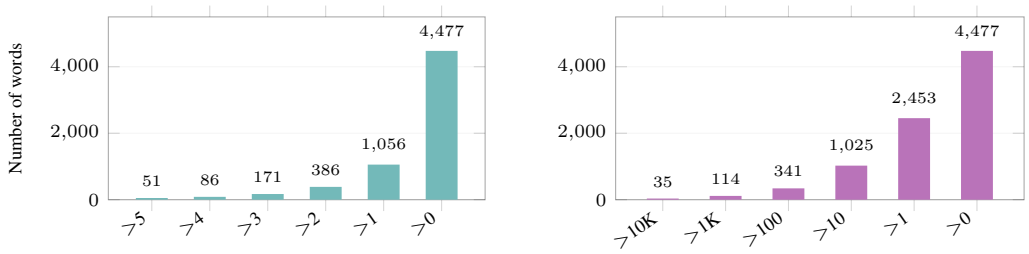
Dataset	Eps	Steps	Empty	Uniq cmds	Uniq words
Bridge (Walke et al., 2024a)	60,064	2,255,591	16,257	22,198	2,484
RT-1 (Brohan et al., 2023)	87,212	3,786,400	8	578	49
QT-Opt (Kalashnikov et al., 2018)	209,880	2,455,879	0	1	2
DROID (Khazatsky et al., 2024)	92,233	27,044,326	42,300	31,303	1,963
BC-Z (Jang et al., 2021)	43,264	6,015,535	0	104	51
FMB Dataset (Luo et al., 2024)	8,611	1,137,340	0	64	31
Language Table (Lynch et al., 2022)	442,226	7,045,476	0	127,583	928
Stanford Hydra (Belkhale et al., 2023)	570	358,234	0	3	17
Taco Play (Rosete-Beas et al., 2022)	3,603	237,798	0	406	74
Furniture Bench (Heo et al., 2023)	5,100	3,948,057	0	9	10
Austin Buds (Zhu et al., 2022b)	50	34,112	0	1	16
Austin Sailor (Nasiriany et al., 2022)	240	353,094	0	4	26
Austin Sirius (Liu et al., 2023b)	559	279,939	0	2	16
Berkeley Autolab UR5 (Chen et al.)	1,000	97,939	0	5	29
Berkeley Cable Routing (Luo et al., 2023)	1,647	42,328	0	1	2
Berkeley Fanuc Manipulation (Zhu et al., 2023)	415	62,613	0	32	63
Cmu Stretch (Bahl et al., 2023; Mendonca et al., 2023)	135	25,016	0	5	15
DLR Wheelchair (Vogel et al., 2020; Quere et al., 2020)	104	8,928	0	13	17
DobbE (Shafiullah et al., 2023)	5,208	1,139,911	0	7	14
Iamlab Cmu Pickup Insert (Saxena et al., 2023)	631	146,241	0	7	12
USC Jaco Play (Dass et al., 2023)	1,085	77,965	0	89	31
NYU Franka Play (Cui et al., 2022)	456	44,875	0	1	4
Roboturk (Mandlekar et al., 2019)	1,995	187,507	0	3	6
TOTO (Zhou et al., 2023)	1,003	325,699	0	1	1
UCSD Kitchen (Ge Yan & Wang, 2023)	150	3,970	0	8	23
Austin Mutex (Shah et al., 2023)	1,500	361,883	0	50	459
Viola (Zhu et al., 2022a)	150	76,324	0	3	15
Entire mixture	969,091	57,552,980	58,565	182,421	4,477

The full training mixture contains 4,477 unique words. However, many of the words that appear only once are spelling errors or noisy fragments, such as *srgerag*, *cicrle*, *geren*, *thke*, *dedplace*, *pput*, and *boaard*. This suggests that the effective semantic vocabulary is even smaller than the raw count of unique words indicates.

Figure 5 summarizes this pattern from two perspectives: how many datasets each word appears in, and how frequently each word occurs in the mixture. The results show that most words are rare, while the shared vocabulary across datasets is dominated by a small set of words. Although the dataset up- and down-sampling changes the exact training frequencies, the overall conclusion remains the same: the active language vocabulary used during VLA training is extremely limited.

B FREQUENT VLA DATASET WORDS

To better characterize the language distribution seen during VLA training, we analyze the words that occur in more than three OXE train datasets. Table 5 groups these frequent words into broad semantic categories. The categories are defined inductively from recurring patterns in the vocabulary,



(a) Number of datasets

(b) Number of unique instructions

Figure 5: Vocabulary distribution in the OXE Magic Soup+ mixture. Left: number of datasets in which each word appears. Right: number of unique commands words appear. Most words are rare, while the shared vocabulary is concentrated in a small set of frequent instruction words.

rather than from a predefined taxonomy. We first use LLM-assisted annotation to assign preliminary labels, and then manually verify and correct all labels. For ambiguous words that could plausibly belong to multiple categories, such as *close* as either an action verb or a spatial indicator, we determine the final label from the word's dominant role in 10 randomly sampled unique commands.

Table 5: Semantic categories for the 169 words that appear in more than three OXE train datasets, bolded words denote ones that appear as a single token in commands tokenized with Llama 2 tokenizer.

Category	N	Words
object description	53/**36**	**table**, drawer, bowl, **object**, **cup**, **door**, oven, **plate**, **square**, **apple**, **box**, **cabinet**, banana, **stick**, **pot**, bottle, **block**, **circle**, **objects**, **handle**, dish, **cube**, rack, **bread**, **round**, **can**, **holder**, sponge, **star**, tray, **pan**, plastic, **board**, **lid**, stove, **sink**, **tower**, **fork**, **water**, **kitchen**, mug, **blocks**, **paper**, **tool**, brush, **rectangle**, oval, **switch**, **button**, cups, **circular**, butter, triangular
action verb	37/**35**	**pick**, **place**, **put**, **open**, **close**, **move**, **take**, **push**, **make**, **rotate**, **pull**, **insert**, sweep, **slide**, **using**, **use**, **turn**, **knock**, **stack**, **lift**, **pour**, **remove**, **go**, **position**, **drag**, **let**, **bring**, arrange, **set**, **separate**, **placed**, **point**, **get**, **adjust**, **moving**, **touch**, **placing**
spatial descriptor	31/**30**	**up**, **on**, **in**, **into**, **top**, **right**, **left**, **inside**, **onto**, **over**, **off**, **down**, **bottom**, **near**, **towards**, **side**, **out**, **above**, **middle**, **front**, **upper**, **edge**, **toward**, **back**, **along**, **direction**, **forward**, **line**, **away**, downward, **under**
color	10/**8**	**blue**, **green**, **red**, **yellow**, **white**, **orange**, purple, pink, **brown**, **black**
stopword	24/**24**	**the**, **and**, **of**, **to**, **with**, **from**, **it**, **a**, **then**, **by**, **is**, **other**, **but**, **or**, **them**, **at**, **you**, **its**, **all**, **that**, **each**, **too**, **both**, **while**
other	14/**13**	upright, **one**, **arm**, **surface**, **hand**, **slightly**, **two**, **together**, **spot**, **s**, **location**, **little**, **space**, **long**

Table 6: Average pairwise cosine similarity of input embeddings within each semantic category (single-token words from the OXE vocabulary), compared to all single-token words and all changed tokens.

Group	**n**	**Prismatic**	**OpenVLA**	**Δ**
Object description	36	0.0563	0.0538	-0.0024
Action verb	35	0.0719	0.0595	-0.0124
Spatial descriptor	30	0.0770	0.0629	-0.0141
Color	8	0.1969	0.1653	-0.0316
Stopword	24	0.0682	0.0563	-0.0120
Other	13	0.0480	0.0410	-0.0069
All single-token freq	146	0.0421	0.0330	-0.0091
All changed	3391	0.0226	0.0200	-0.0026

The resulting vocabulary is concentrated around a small set of instruction-relevant categories: object descriptions, action verbs, spatial descriptors, colors, stopwords, and a small residual category. Most of the frequent vocabulary consists of concrete household objects, manipulation verbs, and spatial relations, reflecting the narrow command structure of current robot training data. We additionally mark which words appear as single tokens under the Llama 2 tokenizer, since only these words can be directly analyzed as individual token embeddings in OpenVLA.

Table 6 reports the average pairwise cosine similarity of input embeddings within each semantic category, using only the single-token words from Table 5. Across all categories, the average within-category similarity decreases from Prismatic to OpenVLA. The largest decreases occur for color words, spatial descriptors, and action verbs, while object descriptions show a smaller change. This suggests that VLA adaptation does not simply make frequent semantically related words more clustered in the input embedding space. Instead, the embeddings of frequent instruction words become more differentiated after VLA training.

For comparison, we also report the average pairwise similarity among all single-token frequent words and among all changed tokens. The frequent OXE words are more internally similar than the full set of changed tokens, both before and after VLA adaptation, which is expected given their shared task-oriented semantics. However, the decrease after VLA training is stronger for the frequent OXE vocabulary than for all changed tokens. This supports the observation that VLA adaptation especially affects the subset of language tokens that are repeatedly used in robot instructions.

C Embedding Spaces

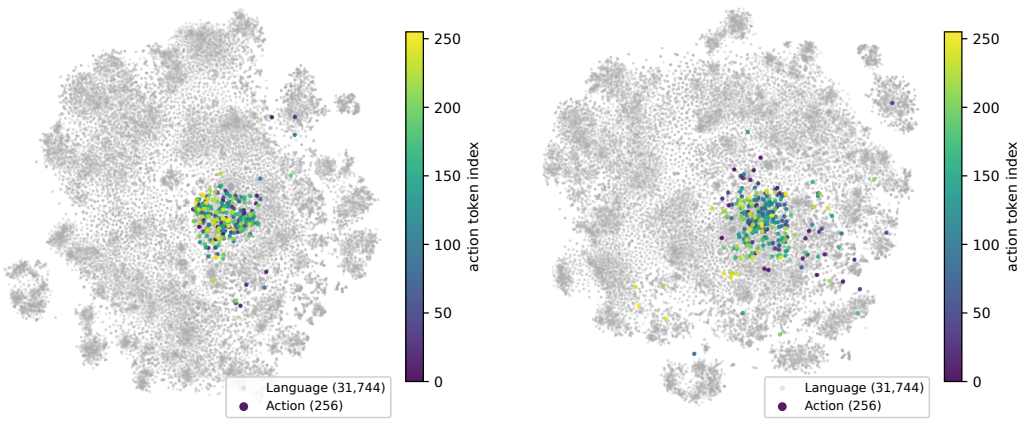
Table 7: Mean pairwise cosine distances between language token embeddings across model stages. The Llama/OpenVLA family uses untied input and output embeddings, reported separately, while the Gemma/π_0-FAST family uses tied embeddings.

Model	Stage	Embedding	Tokens	Mean pairwise cosine distance
Llama-2	LLM	embed_tokens (in)	32,000	0.980752
Prismatic	VLM	embed_tokens (in)	32,000	0.980844
OpenVLA	VLA	embed_tokens (in)	31,743	0.980978
Llama-2	LLM	lm_head (out)	32,000	0.967250
Prismatic	VLM	lm_head (out)	32,000	0.963456
OpenVLA	VLA	lm_head (out)	31,743	0.530488
Gemma-2B	LLM	tied	256,000	0.168236
PaliGemma	VLM	tied	256,000	0.168831
π_0-FAST	VLA	tied	254,976	0.166732

This appendix provides additional visualizations and summary statistics for the language and action embedding spaces analyzed in the main text. Overall, the Llama/OpenVLA and Gemma/π_0-FAST families show very different embedding geometries. The Llama-family models use untied input and output embeddings, while the Gemma-family models use tied embeddings; this architectural difference is reflected in both the t-SNE visualizations and the mean pairwise cosine distances.

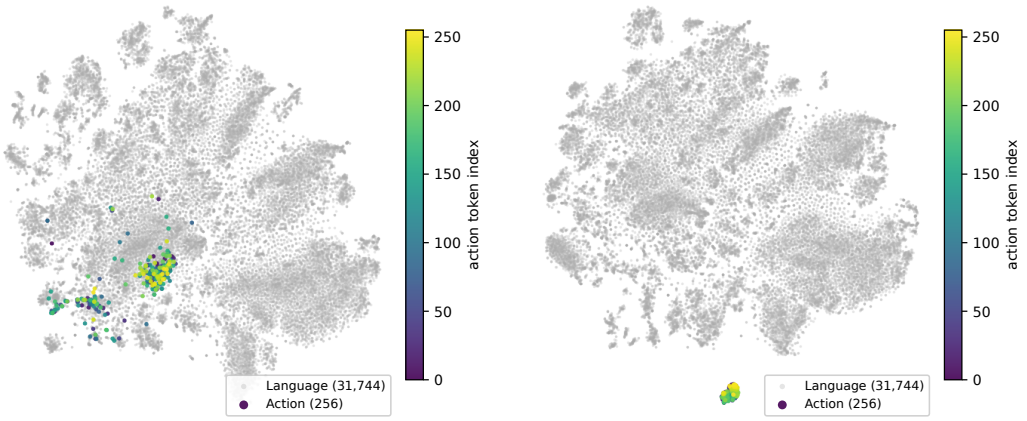
Table 7 shows that the Llama-family input embeddings remain highly dispersed across the LLM, VLM, and VLA stages, with mean pairwise cosine distances close to 0.98. The output embeddings are also dispersed in Llama-2 and Prismatic, but change substantially in OpenVLA, where the mean pairwise distance drops to 0.53. This reflects the strong restructuring of the OpenVLA output space after action-token learning. In contrast, the Gemma-family tied embeddings have much lower mean pairwise cosine distances, around 0.17, and remain relatively stable across Gemma-2B, PaliGemma, and π_0-FAST.

Figure 6 visualizes the relationship between language and action-token embeddings. In the OpenVLA output embedding space, the action tokens form a compact and clearly separated structure, consistent with their role as quantized action outputs. In the input embedding spaces, the action tokens are more embedded within the broader language-token space. For the Gemma-family models, the tied embedding matrix produces a different geometry: π_0-FAST action tokens occupy localized regions of the space rather than forming the same smooth global structure observed in OpenVLA.



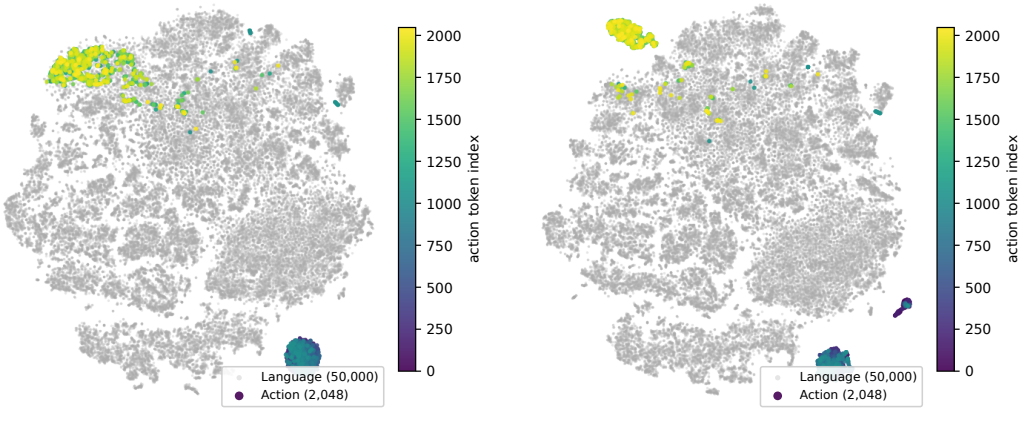
(a) Prismatic input embeddings

(b) OpenVLA input embeddings



(c) Prismatic output embeddings

(d) OpenVLA output embeddings



(e) PaliGemma tied embeddings

(f) π_0-FAST tied embeddings

Figure 6: t-SNE of action and language token embeddings. Gray points denote language tokens; viridis-colored points denote action tokens colored by token index. With Gemma-family models, we sample 50k language tokens.

D ACTION EMBEDDINGS

The action-token embeddings show a clear difference between OpenVLA and π_0-FAST. In OpenVLA, each action token corresponds to a quantized scalar value, which makes the learned action-token structure relatively easy to interpret. As shown in Figure 7, the action embeddings are initially inherited from the Prismatic VLM vocabulary, but after VLA training they become organized according to the action magnitudes they represent, forming a smooth progression from -1 to 1.

In π_0-FAST, the action-token space is harder to interpret globally. Unlike OpenVLA's scalar action bins, FAST tokens represent learned action-tokenizer units, where lower-indexed tokens correspond to more frequent action patterns introduced earlier by the tokenizer merges. Under PCA, the resulting embedding space does not form the same smooth global structure as OpenVLA (Figure 8). However, the t-SNE projection suggests that VLA training still creates local neighborhoods among action tokens (Figure 9).

This difference may partly reflect the initialization of FAST action tokens. Half of the FAST action tokens are initialized from PaliGemma detection tokens, while the other half are initialized from infrequent language tokens. These two groups have different initial geometries and norm distributions. Figure 10 shows that tokens initialized from detection tokens start with very low norms and increase substantially during VLA training, whereas tokens initialized from language tokens start with higher and more variable norms and change less. Together, these results suggest that the less coherent global structure of π_0-FAST action embeddings is influenced not only by the FAST action-tokenization scheme, but also by the heterogeneous initialization of the action-token vocabulary.

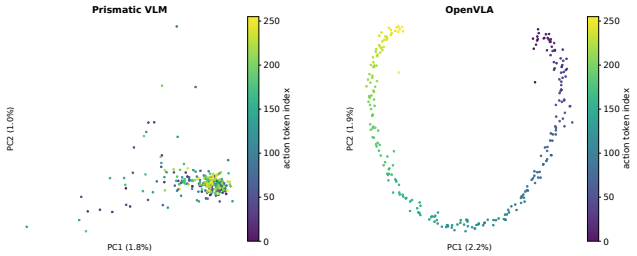


Figure 7: PCA visualization of OpenVLA action-token embeddings before and after VLA training. Before adaptation, these embeddings correspond to language-token embeddings inherited from the Prismatic VLM. After VLA training, the tokens become organized according to the action magnitudes they represent, forming a smooth progression from -1 to 1.

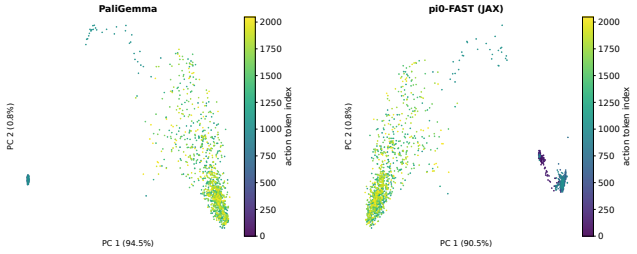


Figure 8: PCA visualization of π_0-FAST action-token embeddings before and after VLA training. The action tokens are initialized from a mixture of PaliGemma detection and language tokens. Lower-indexed FAST tokens correspond to more frequent action patterns in the training data, as these tokens are introduced earlier by the action-tokenizer merges.

E OPENVLA OUTPUT EMBEDDING CHANGES

In Figure 11, we show the cosine similarities to the mean direction of the VLA-adaptation change vectors for output embeddings in the OpenVLA model. The only non-agreeing group is action

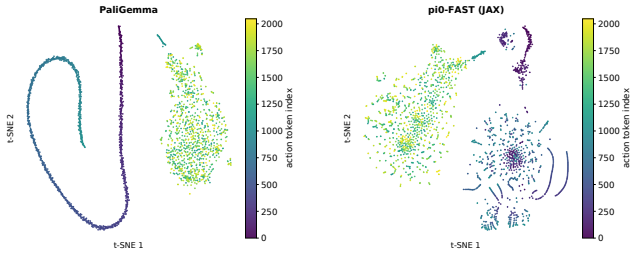


Figure 9: t-SNE visualization of π_0-FAST action-token embeddings before and after VLA training. The action tokens are initialized from a mixture of PaliGemma detection and language tokens.

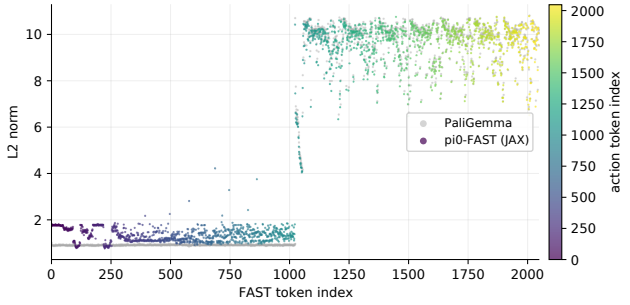


Figure 10: L2 norms of π_0-FAST action-token embeddings before and after VLA training. Tokens initialized from PaliGemma detection tokens start with very low norms and increase substantially during training, whereas tokens initialized from language tokens start with higher and more variable norms and change less.

tokens, the only ones that are trained in the output head, and since OpenVLA has untied embeddings, output probabilities are the only source of gradients for these parameters.

F Examples of Tokens in Unique Command Tiers

Table 8 shows example OpenVLA tokens (frequency > 100, matching Figure 2) for each unique command tier. Low-diversity tokens skew toward specific vocabulary, while the 1000+ tier is dominated by core spatial and action vocabulary (e.g. _pick, _place).

Table 8: Example changed tokens sampled from each contextual-diversity tier (number of distinct instructions the token appears in). The underscore denotes the tokenizer's start-of-word marker.

Unique commands	Example tokens
1	_obtain, _register, uv, _posterior, _efficiently, ering, _applied, _access, _thought, ably, antly, _narrow, anced, _anywhere, _fast, rol, _https, ijn, _kin, ('
2-9	_manner, aj, _baseball, voir, ure, _repeatedly, ith, allow, oot, _outer, vre, _complete, ices, _kept, _orn, _devant, _tour, imi, _thread, _intelligence
10-99	ately, _ca, ree, _direction, de, _once, _pushing, _gra, over, _letter, lower, _puts, ender, _tw, _keys, _stra, _images, _separately, _op, _north
100-999	_piece, _w, _ra, _book, _rectangle, _grey, _chair, _switch, _cable, _lever, _hang, _head, raw, _for, rest, ow, _both, _little, _moves, _form
1000+	., _pick, _place, _on, _blue, _bow, _close, _t, ge, ple, _pan, _hex, _one, ug, _use, _arm, _hand, _them, _below, _adjust

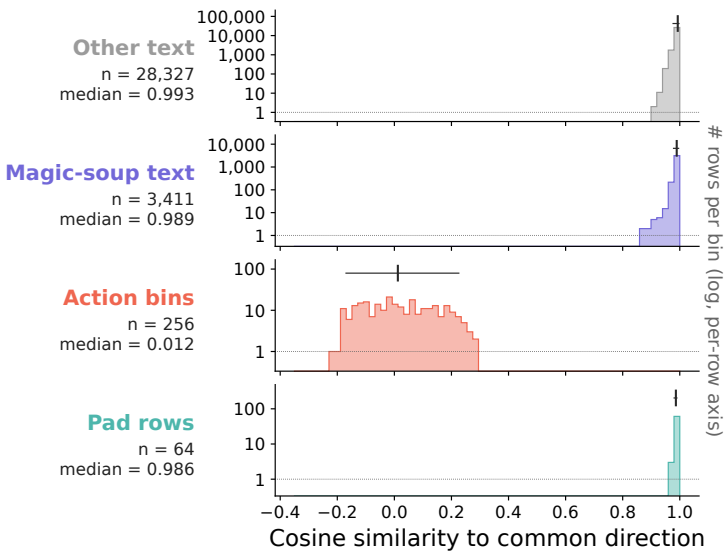


Figure 11: Distributions of cosine similarity to common direction among Prismatic → OpenVLA embedding change vectors. We differentiate between language tokens (the ones appearing in the Magic Soup dataset and all the rest), action bins, and pad rows, unused in the VLA model. Each plot presents logarithmically scaled counts and shows the sample median and the 0.05th and the 0.95th percentiles above the histograms.

G π0-FAST Token Changes

By varying the threshold τ for selecting outliers from 0.3 to 0.8 in terms of similarity to the common direction v, we estimate the total number of tokens used in training commands. We distinguish between in-Magic-Soup inliers $S_i, \forall s \in S_i : \cos(s, v) \geq \tau$ and outliers $S_o, \forall s \in S_o : \cos(s, v) < \tau$ and defined similarly out-of-Magic-Soup inliers N_i and outliers N_o. Under the assumption that $\frac{|N_i|}{|N_o|} = \frac{|S_i|}{|S_o|}$, we estimate $|N_i| = \frac{|N_o| \cdot |S_i|}{|S_o|}$ and compute it for all thresholds. The total value $|N_i| + |N_o| + |S_i| + |S_o|$ varies from 5,234 to 6,516, allowing us to roughly estimate the effective vocabulary size to be below 7,000 and thus amount to less than $\sim$3% of the vocabulary. This number of tokens corresponds well to the vocabulary sizes used for encoding the Magic Soup in π0-FAST (4,543) and OpenVLA (3,411) and gives extra space for the unknown vocabulary in π0-FAST.

We show examples of outlier tokens (cosine similarity below 0.5) that are not present in Magic Soup in Table 9. Tokens mostly include food and other consumables, household objects, verbs related to object manipulation, and words relevant to episode results. Many of these tokens appear in the evaluation cases reported for π-family models (Intelligence et al., 2025), which suggests they are likely common in the training corpus as well.

H Nearest-Neighbor Examples

Figure 12 shows additional nearest-neighbor examples for selected instruction-relevant tokens after VLA adaptation. The examples include color, object, spatial, and action-related tokens, illustrating how local embedding neighborhoods differ across semantic categories.

I Robustness to Minimal Noise: Extended

In Section 5.3, we use CKA to measure the sensitivity of the models' encoded input representations to minimal command corruption. We caution that the CKA analysis may inflate the apparent change: the values are lower for all models because corruption does not produce a uniform change in scale across episodes, due to inherent differences in the nouns being corrupted and their role in each com-

Table 9: Tokens that never occur in the OXE Magic Soup instructions, yet whose PaliGemma → π_0-FAST change vector disagrees with the common direction ($\cos < 0.5$; 250 of 249,608 such tokens), lowest $\cos$ first. They reveal text seen in fine-tuning beyond magic soup: the π_0-FAST prompt template and household, grocery and failure vocabulary. Word pieces are shown joined to the piece they complete (_bus+sing).

Category (n)	Example tokens
Pi0-FAST prompt template (8)	|, <bos>, _State, ., , Task, Action, <eos>,), ,
Food & groceries (58)	_matcha, _nuts, _oatmeal, _taco, _bacon, _donuts, _tart, _zest, _cheddar, _barley, _cashew, _ham, _truffles, _noodles, _sweetener, _bottled, _pretzels, _chocolates, _groceries, _seaweed
Personal care & consumables (21)	_shampoo, _lotion, _wipes, _compost, _lipstick, trash, _gel, _balm, _moisturizer, _pills, _toiletries, _razor, _medication, _buds, _repellent, _eyeshadow, _toner, _mosquito, _sunscreen, _relief
Household objects & tools (30)	_bamboo, _motor, _screws, _collar, _corks, _pipe, _plaid, _athletic, _grinder, _barcode, _compass, _magnifying, _ruler, _reusable, _slotted, _garland, _piano, _pouch, _robe, _batteries
Manipulation & tidying verbs (33)	_scan, _assemble, _tidy, _holding, _folding, _organize, _threw, _grabs, _swap, _dispose, _throws, _unzip, _dump, _knock, _refill, _pouring, _picking, _packing, _select, _unpack
Failure & episode meta (16)	_mistake, _lost, _failed, _missed, _miss, _simulate, _skill, _alternative, _safety, _operator, _tried, _version, _pause, _episode, _wrong, _progress
Pieces of multi-token words (7)	_sk+ittles, _bus+sing, _unt+angle, _trow+el, _pret+zel, _carab+iner
Unassigned (77)	_pat, _unf, _ore, url, ats, _pro, _tw, _sun, _energy, ppers, nickel, _living, _sea, _hack, _sus, _pure, _surfaces, _party, _kat, ade

mand. As a result, the corruption cannot be made explicitly uniform across examples. This makes the average cosine similarity analysis an important complementary view for better understanding the effect of command corruption on the models' representations.

Further, to examine the effect of corruption on the models' produced outputs, and as a sanity check, we show the outputs generated by the models in response to both original and corrupted commands in Table 10. Episodes 3 and 5 show Prismatic generating the same output despite the corrupted noun, suggesting that it may recover the intended object from the image rather than relying only on the text token. In episode 4, Llama-2-7B normalizes the typo during generation, while Prismatic paraphrases *cylindtr* as *cylindrical object*.

J Representation drift predicts robot success rate on LIBERO-Object and LIBERO-Goal

The main experiments (in Section 5.3) measure how far a corrupted instruction moves OpenVLA's and π0-FAST's hidden representations, but we do not have the setups to test these models on the those downstream tasks. To show that this drift corresponds to real degradation in robot behaviour, we reproduce the measurement on two established closed-loop benchmarks, where a simulator judges success against the true goal rather than against the instruction.

Setup. We use LIBERO-Object and LIBERO-Goal, each with 10 manipulation tasks, and the OpenVLA checkpoints fine-tuned on each suite. Rollouts follow the official OpenVLA evaluation procedure and are simulated with robosuite 1.4.1 on MuJoCo 2.3.7, with success determined by the task's



Figure 12: Additional nearest-neighbor examples for selected instruction-relevant tokens after VLA adaptation.

BDDL goal predicate. Our pipeline reproduces the published success rates: 86.8% against 88.4% on LIBERO-Object, and 78% against 79.2% on LIBERO-Goal over 500 episodes (10 tasks × 50 initial states).

Perturbations. On LIBERO-Object we introduce a single-character mistake in the manipulated object's head noun, giving 500 paired trials: each of the 10 tasks is run from its 50 initial states, once with the original instruction and once with the corrupted one. On LIBERO-Goal we instead use lexical paraphrases from LIBERO-Para (Kim et al., 2026a), which rewrite the same tasks without changing their goals: 50 paraphrases per task, 500 in total, each run from the same initial state as the original instruction. Every comparison is therefore made in an identical scene, with the wording as the only difference. Both perturbations degrade behaviour. A one-character noun mistake lowers success on LIBERO-Object from 86.8% to 70.6%. Lexical paraphrases lower success on LIBERO-Goal from 78% to 38.7%.

Drift predicts which perturbations fail. We restrict this analysis to trials the robot completed with the original instruction (434 of the 500 LIBERO-Object trials; on LIBERO-Goal, all 10 tasks, each of which exceeds 50% success with the original instruction), so that a failure can be attributed to the perturbation rather than to an unachievable start. Drift is the cosine distance between the last-token hidden states of the original and perturbed prompts at the first policy step, where both conditions observe an identical image. Because tasks differ in difficulty and in typical drift, we convert drift

Table 10: Outputs of the two text-generating Llama-family models on five Bridge V2 episodes under the original instruction (*orig*) and a 1-character noun corruption (*corr*). Corrupted nouns are shown in **red**. *Unchanged* marks outputs identical to the original, indicating the model did not reflect the corruption. OpenVLA (VLA) is excluded from the table as it produces discretised robot-action tokens rather than natural language; its token sequences differ between original and corrupted conditions across all five episodes.

#		Instruction	Llama-2-7B (LLM)	Prismatic (VLM)
1	*orig*	move the silver **pot** over the upper left stove	The robot should move the silver pot over the upper left stove.	Move the pot using its arm and place it on the stove.
	corr	move the silver **pxt** over the upper left stove	The robot should move the silver **pxt** over the upper left stove.	Move the robot arm and place the silver **pxt** on the upper left stove.
2	*orig*	move the yellow checked **vessel** from front to back	The robot should move the yellow checked vessel from front to back.	Move the yellow checked vessel using the fork and spoon.
	corr	move the yellow checked **versel** from front to back	The robot should move the yellow checked **versel** from front to back.	Move the yellow checked **versel** from front to back using the arm.
3	*orig*	move blue **towel** to lower right corner	The robot should move the blue towel to the lower right corner.	Move the blue towel to the lower right corner using its arm.
	corr	move blue **topel** to lower right corner	The robot should move blue **topel** to lower right corner.	*(unchanged)*
4	*orig*	put the **cylinder** on top of the yellow rectangular block	The robot should move the cylinder to the top of the yellow rectangular block.	The robot should move its arm and carefully place the cylinder on top of the block.
	corr	put the **cylindtr** on top of the yellow rectangular block	*(unchanged)*	The robot should move its arm and carefully place the **cylindrical object** on top.
5	*orig*	place the blue **brush** to the right of the purple towel	The robot should place the blue brush to the right of the purple towel.	Move the blue brush to the right of the purple towel.
	corr	place the blue **brujh** to the right of the purple towel	The robot should move the blue **brujh** to the right of the purple towel.	*(unchanged)*

to its rank within each task before pooling. Figure 13.a shows that success falls monotonically with drift on both benchmarks: across within-task drift quintiles at layer 30, success declines from 76% to 66% on LIBERO-Object and from 59% to 11% on LIBERO-Goal. Figure 13.b shows the same relation as a detector of failure: the within-task AUC is 0.75 on LIBERO-Goal and 0.58 on LIBERO-Object. On LIBERO-Goal, drift is the better predictor of the two, exceeding the keyword similarity that LIBERO-Para itself uses to quantify paraphrase distance (AUC 0.71), while the model's first predicted action carries almost no signal (AUC 0.51–0.57). Figure 13.c shows that this holds at every layer we measure (layers 8–31), so it does not depend on the particular choice of layer 30.

Together these results connect the cosine drift between original and perturbed instruction representations to degraded downstream performance on two established benchmarks, supporting its use in the main experiments as a proxy for behavioral deterioration. The link is correlational: higher drift indicates a perturbation the policy is more likely to fail on, not a demonstrated causal mechanism.

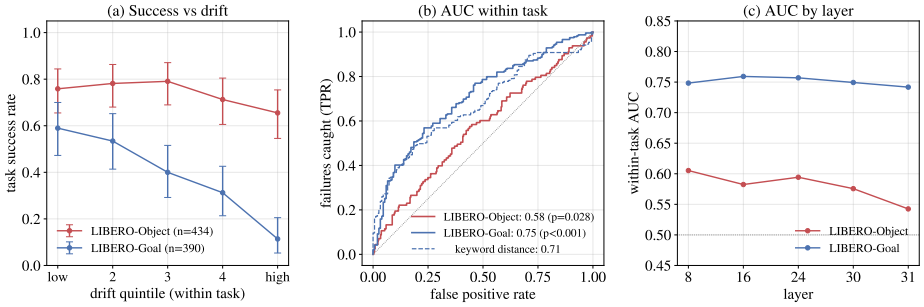


Figure 13: a) More drift in hidden representation (layer 30) indicates lower success rate in task b) predicting task success rate using hidden representation drift for different benchmarks (layer 30) c) Representation drift at all layers predicts task success