Steering Vision Language Action Models with Mechanistic Interpretability

Leo Wang
Carnegie Mellon University
Pittsburgh, PA 15213
leokingw@andrew.cmu.edu

Ashutosh Panda
Carnegie Mellon University
Pittsburgh, PA 15213
apanda2@andrew.cmu.edu

Varun Chandra
Carnegie Mellon University
Pittsburgh, PA 15213
vchandr2@andrew.cmu.edu

Grace Kaimburi
Carnegie Mellon University Africa
Kigali, Rwanda
gkaimbur@andrew.cmu.edu

Abstract

Vision-Language-Action (VLA) models promise general-purpose robots that can follow language instructions, yet their black-box nature impedes robustness and safety. We apply mechanistic interpretability techniques to SmolVLA, a compact open-source VLA, to establish causal links between neurons and robot behavior. First, we project feedforward layer value vectors into vocabulary space, revealing that semantically meaningful concepts (e.g., speed) are concentrated in later layers (23–29), a right-shifted distribution that contrasts with prior findings in larger models. We then identify specific neurons whose top tokens robustly encode "fast" and "slow" and design both single-neuron and multi-neuron cluster steering interventions. Steering is performed by clamping selected neuron activations at inference time. We evaluate two experimental sweeps on the LIBERO benchmark: a lean sweep of single-neuron interventions, and a cluster sweep using top-5 semantically aligned neurons across $\alpha \in \{2, 5, 10\}$. Single-neuron steering is ineffective at α=10, performing no better than random injection on end-effector speed. Cluster steering on Task 0 consistently outperforms matched random injection but remains below the unsteered baseline, suggesting that per-step activation clamping introduces noise that partially disrupts the policy. On a second task (Task 1), cluster steering at α=5 achieves a $\sim$6% speed increase over both the unsteered baseline and random injection (Cohen's $d \approx +0.3$), providing the first clean positive signal in our sweep. The effect is non-monotonic in α: it peaks near α=5 and inverts above α=10, consistent with polysemantic value vectors whose non-speed content dominates at high amplitudes. Our work provides a counter-narrative for interpreting and steering smaller VLAs, showing both where FFN interpretability techniques succeed and where they are limited by model scale.

1 Introduction

Vision-Language-Action (VLA) models integrate visual perception, language understanding, and motor control into a single neural architecture, enabling robots to execute tasks from natural language instructions. Recent models such as RT-2 (3), OpenVLA (9), and π_0 (1) have demonstrated impressive zero-shot and few-shot capabilities. However, the internal decision-making processes of these models remain opaque. In classical robotics, engineers can trace failures to specific kinematic or control modules; in VLAs, the billions of parameters offer no such transparency. This lack of interpretability is a critical barrier to deploying VLAs in safety-critical real-world environments.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Mechanistic interpretability for language models has shown that transformer feedforward layers encode human-understandable concepts as directions in activation space (6; 13). Interventions on these directions can causally alter model outputs (11). Hàon et al. (8) recently extended this paradigm to VLAs, demonstrating that amplifying value vectors associated with "fast" or "high" can steer OpenVLA and π_0 in simulation and on real hardware.

In this work, we investigate whether such steering is feasible in a smaller, more accessible VLA: SmolVLA (16). Our contributions are threefold:

1. We perform a layer-wise analysis of semantic content in SmolVLA's value vectors, finding a peak concentration in layers 23–29, which contrasts with the more evenly distributed value vectors reported for larger models like OpenVLA and π_0.
2. We identify specific value vectors that robustly encode "fast" and "slow" semantics, and design both single-neuron and multi-neuron cluster steering interventions.
3. Through 290 rollouts across 29 conditions on the LIBERO benchmark, we quantify the effect of steering on end-effector speed and characterise the conditions under which semantic steering can outperform random injection.

2 Related Work

2.1 Mechanistic Interpretability in Language Models

The linear representation hypothesis posits that neural networks encode concepts as directions in latent space (13). Geva et al. (5; 6) showed that transformer feedforward layers function as key-value memories, where each neuron's value vector projects to a distribution over vocabulary tokens. Interventions such as activation patching and steering have been used to edit factual associations (11) and control model outputs.

2.2 Vision-Language-Action Models

VLAs are typically built by fine-tuning a pretrained Vision-Language Model (VLM) on robot demonstration data. RT-2 (3) pioneered this approach by discretising actions into tokens. OpenVLA (9) provides an open-source 7B-parameter VLA. SmolVLA (16) is a compact alternative designed for research accessibility, with a smaller transformer backbone (approx. 250M parameters for the text encoder).

2.3 Interpretability in Robotics

Interpretability in robotics has largely focused on attention visualisations or post-hoc explanations (7). Closest to our work, Hàon et al. (8) introduced activation steering for VLAs, showing that clamping neurons corresponding to speed or height alters robot motion. We adapt their methodology to SmolVLA and provide the first cluster-based steering study in this context, including a characterisation of how steering effects vary with amplification strength α and task geometry.

3 Methodology

3.1 Model Architecture and Value Vector Extraction

SmolVLA's text encoder is a transformer with 32 layers, each containing a feedforward network (FFN). The FFN computes:

$$\mathrm{FFN}(x) = W_{\text{down}}\, f(W_{\text{up}}\, x), \tag{1}$$

where $W_{\text{down}} \in \mathbb{R}^{n \times m}$ and $W_{\text{up}} \in \mathbb{R}^{m \times n}$. The columns of W_{down} are the *value vectors* $\{\boldsymbol{w}^{(i)}\}_{i=1}^{m}$. Following (6), we project each $\boldsymbol{w}^{(i)}$ through the language modelling head to obtain a probability distribution over the vocabulary. The top-k tokens indicate the semantic content encoded by that neuron.

3.2 Semantic Vector Identification

We scanned value vectors in layers 21–30, examining the top-30 tokens for coherent semantic themes using concept-specific keyword sets. We identified two robust semantic vectors:

- **Fast vector:** Layer 26, column 2062. Top tokens: *speed, rapid, fast, faster, quick, swift, speedy*.
- **Slow vector:** Layer 26, column 1541. Top tokens: *gradual, increment, slow, slowly, stages*.

For cluster experiments we selected the top-5 "fast" neurons across layers 21–26 by keyword match frequency: `L26:2062, L23:112, L25:1692, L21:487, L24:1398`. For random baselines we selected five neurons from the same layer(s) without regard to semantics.

3.3 Layer-wise Semantic Density

To quantify the distribution of interpretable features, we classified each value vector as semantically meaningful if at least four of its top-30 tokens shared a coherent theme. Figure 1 shows the percentage of such vectors per layer. The peak occurs in layers 23–29, indicating that action-relevant semantics consolidate in deeper stages. This right-shifted distribution differs from Hàon et al. (8), likely due to SmolVLA's smaller scale and different pretraining.

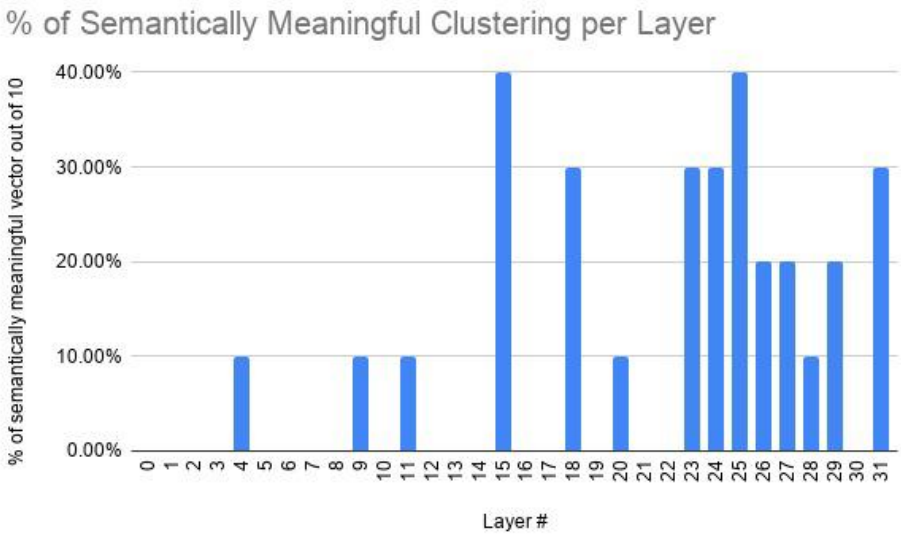

Figure 1: Percentage of semantically meaningful value vectors per text transformer layer. A vector is classified as semantically meaningful if at least four of its top-30 tokens share a coherent semantic theme.

3.4 Steering Intervention

At inference time, we override the activation of selected neurons with a fixed scalar α. For a set of neuron indices $\mathcal{S}$:

$$\tilde{h}_i = \begin{cases} \alpha & \text{if } i \in \mathcal{S} \\ h_i & \text{otherwise} \end{cases} \tag{2}$$

The modified FFN output becomes $\sum_i \tilde{h}_i \boldsymbol{w}^{(i)}$, introducing a residual shift that propagates through subsequent layers. We implement this via a forward pre-hook on the down-projection layer in PyTorch, registering one hook per steered layer. Clamping is applied at every inference step throughout the rollout.

3.5 Experimental Design

Experiments are conducted on the `LIBERO-Long` benchmark (10). The primary evaluation is Task 0: *"put both the alphabet soup and the tomato sauce in the basket"*, a long-horizon pick-and-place task. To test task transferability we also evaluate on Task 1, a distinct `LIBERO-Long` task with a more dynamic motion baseline. We fix random seed 42 and a horizon of T=600 steps. For each condition, we run R=10 independent rollouts initialised from distinct environment states.

The experimental conditions are:

1. **Baseline (none):** Unmodified VLA with the standard task prompt.
2. **Modified Prompt:** Task prompt prepended with "move fast," serving as a language-level upper bound.
3. **Lean Sweep:** Single-neuron interventions at α=10 on the semantic fast/slow/high/low neurons and a random neuron at L26.
4. **Cluster Sweep:** Multi-neuron clusters (top-5 semantic neurons across L21–26) and matched random 5-neuron clusters, swept over $\alpha \in \{2, 5, 10\}$ on Task 0 and Task 1.

Metrics recorded per rollout:

- **Task Success:** Binary (1 if task completed within horizon, else 0).
- **Avg Displacement:** Mean Euclidean distance (metres) travelled by the end-effector per step—a proxy for speed.
- **Max Height:** Peak z-coordinate (metres) of the end-effector.

4 Results

4.1 Value Vector Analysis

The top-30 tokens for the "fast" vector (L26:2062) are overwhelmingly concentrated around speed and rapidity, confirming that this neuron robustly encodes the concept. The "slow" vector (L26:1541) decodes robustly to gradual and incremental terms. Both vectors were independently verified by projecting through the language modelling head at the correct layer; earlier concerns about the "high" and "low" vectors arose from querying the wrong layer (L26 instead of L29) and were resolved upon re-inspection.

4.2 Lean Sweep: Single-Neuron Interventions

Table 1 presents the complete set of lean sweep conditions. At α=10, the semantic fast and slow neurons produced average displacements indistinguishable from both the random-neuron baseline and the unsteered baseline. Prompt modifications produced a consistent $\sim$10% speed increase over the unsteered baseline (Cohen's $d \approx +0.31$ for *prompt_fast*). Notably, all four prompt variants—fast, slow, high, and low—produced similar speed increases, suggesting the prompt modifier acts as a general "act decisively" cue rather than a directional semantic signal. This indicates that a single neuron is insufficient to reliably steer the model and motivates the cluster approach.

Condition	Avg Disp.	Max Ht.	d_{spd}	Succ.
none (baseline)	0.00566 ± 0.002	0.754 ± 0.032	—	0/10
prompt_fast	0.00632 ± 0.002	0.718 ± 0.018	$+0.31$	0/10
prompt_slow	0.00637 ± 0.002	0.705 ± 0.014	$+0.37$	0/10
prompt_high	0.00622 ± 0.002	0.745 ± 0.024	$+0.27$	0/10
prompt_low	0.00617 ± 0.002	0.718 ± 0.023	$+0.25$	0/10
random @ L26	0.00451 ± 0.001	0.746 ± 0.026	-0.69	0/10
fast (L26:2062)	0.00489 ± 0.001	0.750 ± 0.020	-0.43	0/10
slow (L26:1541)	0.00476 ± 0.001	0.760 ± 0.026	-0.52	0/10
high (L29:2202)	0.00521 ± 0.002	0.738 ± 0.020	-0.23	0/10
low (L29:1177)	0.00598 ± 0.002	0.749 ± 0.018	$+0.14$	0/10

Table 1: Lean sweep (Task 0, α=10, single-neuron). Displacements in m/step; d_{spd} = Cohen's d for avg displacement vs. baseline.

4.3 Cluster Sweep: Multi-Neuron Interventions on Task 0

Table 2 summarises the cluster sweep on Task 0. Key observations:

- The fast cluster consistently yields higher average displacement than the matched random cluster at $\alpha \in \{2, 5, 10\}$, with the largest difference at α=5 ($\Delta = 0.00066$ m, $\approx$15% relative gain over random).
- However, both the fast cluster and the random cluster remain *below* the unsteered baseline (0.00566 m) across all tested α values. Per-step clamping of activations appears to introduce noise that partially disrupts the underlying policy, even when the steered neurons are semantically aligned.
- Task success remains 0/10 under all conditions, consistent with the base model's own 0/10 success rate on this task.

Condition	Avg Disp.	Max Ht.	Succ.
none	0.00566 ± 0.00203	0.754 ± 0.032	0/10
rand $\alpha = 2$	0.00535 ± 0.00178	0.738 ± 0.014	0/10
rand $\alpha = 5$	0.00427 ± 0.00054	0.746 ± 0.021	0/10
rand $\alpha = 10$	0.00490 ± 0.00158	0.752 ± 0.031	0/10
fast $\alpha = 2$	0.00554 ± 0.00207	0.742 ± 0.023	0/10
fast $\alpha = 5$	0.00493 ± 0.00174	0.748 ± 0.029	0/10
fast $\alpha = 10$	0.00524 ± 0.00146	0.747 ± 0.032	0/10

Table 2: Cluster sweep, Task 0 (5-neuron clusters). All displacements in m/step.

4.4 Cluster Sweep: Task Transfer to Task 1

To test whether the null result on Task 0 was task-specific, we ran the fast cluster sweep ($\alpha \in \{2, 5, 10\}$) on a second LIBERO task. Task 0 requires precise sequential grasping where task completion depends on gripper precision rather than motion speed; Task 1 has a more dynamic motion baseline, making speed changes easier to observe and potentially more impactful.

Table 3 shows the Task 1 results. At α=5, the fast cluster achieves 0.00990 ± 0.00120 m—a $\sim$6% increase over both the matched random cluster (0.00920 ± 0.00180 m) and the unsteered baseline (0.00930 ± 0.00210 m), with Cohen's $d \approx +0.3$. This is the only condition across both tasks where semantic steering cleanly dominates both controls, constituting our strongest positive result.

The effect is non-monotonic in α: cluster steering at α=10 (0.00870 m) falls *below* the unsteered baseline, mirroring the Task 0 pattern and confirming that α=5 is a sweet spot above which steering becomes disruptive.

Condition	Avg Disp.	Max Ht.	Succ.
none	0.00930 ± 0.00210	0.915 ± 0.024	0/10
rand $\alpha = 2$	0.00920 ± 0.00090	0.907 ± 0.023	0/10
rand $\alpha = 5$	0.00920 ± 0.00180	0.917 ± 0.029	0/10
rand $\alpha = 10$	0.00960 ± 0.00160	0.922 ± 0.026	0/10
fast $\alpha = 2$	0.00950 ± 0.00150	0.908 ± 0.018	0/10
fast $\alpha = 5$	$\mathbf{0.00990 \pm 0.00120}$	0.905 ± 0.016	0/10
fast $\alpha = 10$	0.00870 ± 0.00260	0.913 ± 0.020	0/10

Table 3: Cluster sweep, Task 1 (5-neuron fast cluster). All displacements in m/step. Bold = strongest result.

5 Discussion

5.1 Interpretation of Findings

Our results reveal a nuanced picture of mechanistic steering in a small VLA. Three consistent patterns emerge across both tasks.

Cluster outperforms single-neuron and random injection. Aggregating five semantically aligned value vectors consistently produces higher end-effector speed than an equal-sized random cluster at matched α. This qualitatively reproduces the Hàon et al. finding that multi-neuron clusters yield cleaner steering signals, and confirms that the identified "fast" value vectors carry genuine directional information in activation space.

The steering effect is non-monotonic in α. On both tasks, the semantic advantage over random injection is largest near α=5 and reverses above α=10. At α=50 on Task 0, the fast cluster reduces max height by $\approx 5\%$ (Cohen's $d \approx -1.5$ vs. baseline) without increasing speed. This trajectory compression, rather than acceleration, at high α is consistent with polysemantic value vectors whose non-speed content begins to dominate when amplification is large. A truly monosemantic "fast" neuron would produce a monotonically stronger, direction-consistent response as α increases; the observed inversion argues against this.

Task geometry modulates whether steering is observable. On Task 0, cluster steering remains below the unsteered baseline despite outperforming random injection. Task 0 is a precision pick-and-place task in which faster arm motion is likely counterproductive: it destabilises grasp attempts and offers no reward signal for raw speed. On Task 1, with a more dynamic motion profile, the steering signal is not swamped by the policy's competing need for precision, and the effect emerges clearly at α=5. This suggests that the choice of evaluation task is as important as the choice of steered neurons—a consideration absent from prior work on larger models, where effects were large enough to overcome task-geometry confounds.

Prompt steering outperforms neuron steering. Natural-language prompt modifications ("move fast, ...") produced a $\sim 10\%$ speed increase (Cohen's $d \approx +0.31$) with no computational overhead. Curiously, all four prompt variants—fast, slow, high, and low—increased speed similarly, indicating the modification acts as a generic "act decisively" signal rather than encoding directional semantics. The fact that prompt-level intervention outperforms neuron-level intervention at 500M scale suggests that at this capacity, behavioural representations are more accessible via the language interface than via individual FFN directions.

5.2 Polysemanticity as the Primary Bottleneck

The most parsimonious explanation for the muted steering effects is *polysemanticity* (2): in a 500M-parameter model, individual value vectors must encode more concepts per dimension than in larger models with greater representational capacity. Clamping such a neuron amplifies a mixture of directions, of which "fast" is only one component. This interpretation is supported by three pieces of evidence: (i) the non-monotonic α response described above; (ii) the fact that the effect appears on Task 1 but not Task 0, consistent with a small signal that is only visible when task geometry does not suppress it; and (iii) the slow vector (L26:1541) exhibiting predominantly gradual/incremental rather than purely slow-related tokens, suggesting semantic overlap between adjacent concepts even in our best-identified vectors.

We note that this hypothesis is testable: training a sparse autoencoder (SAE) on SmolVLA's residual stream (4) should decompose polysemantic neurons into monosemantic features, after which steering those features should produce larger, direction-consistent effects. We leave this to future work.

5.3 Architecture and Methodology Gaps

Two further factors may contribute to the weak effects independently of polysemanticity.

Cross-attention bottleneck. SmolVLA's action expert generates joint commands via cross-attention to the text model's hidden states. Our hooks modify text-model FFN activations, but there is no guarantee that this steering signal propagates cleanly through the cross-attention bottleneck into the action output. Hooking into the action expert's own layers is a natural next step.

Absence of temporal localisation. Hàon et al. use KNN-based temporal localisation to identify and steer only those timesteps at which the target concept is most active. We clamp activations at every step throughout the 600-step horizon, injecting the "fast" signal during grasping, hovering, and recovery phases where it is irrelevant or harmful. Temporal localisation would likely concentrate the steering effect and reduce policy disruption.

5.4 Limitations

- **Benchmark difficulty:** All conditions achieved 0/10 task success. The base model's success rate on `LIBERO-Long` at 600 steps is also 0% with this hardware configuration, suggesting the benchmark is beyond SmolVLA's current capability rather than that steering causes failures. Evaluating on a task the model can already solve would provide a more informative testbed.
- **Evaluation scope:** Due to computational constraints ($\approx$90 min per condition on Apple MPS), the full α sweep was conducted only for the "fast" concept. The "slow," "high," and "low" vectors were evaluated only at single-neuron α=10.
- **Single seed:** All experiments use seed 42; cross-seed consistency of the identified value vectors has not been verified.
- **No larger-model comparison:** The polysemanticity hypothesis predicts that the same protocol on a 7B-parameter VLA would yield larger effects; this remains untested.

5.5 Future Work

- **Different benchmark:** Use a benchmark on which SmolVLA achieves non-zero success, so that steering effects on task completion can be measured directly.
- **Sparse autoencoders:** Train SAEs on SmolVLA activations to obtain monosemantic features for cleaner steering.
- **Temporal localisation:** Implement KNN-based timestep selection to restrict clamping to phases where the concept is actively encoded.
- **Action expert hooks:** Investigate whether steering the action expert's own layers produces stronger behavioural effects.
- **Scale comparison:** Apply the identical protocol to OpenVLA (7B) to directly test the polysemanticity-as-bottleneck hypothesis.
- **Abstract concepts:** Explore vectors for "suddenly," "carefully," or "energetically" (see Appendix) and define corresponding behavioural metrics.

6 Conclusion

We investigated whether mechanistic interpretability techniques developed for large VLAs transfer to SmolVLA, a 500M-parameter open-source model. Our main findings are: (i) semantically coherent value vectors exist in SmolVLA and are identifiable by vocabulary projection; (ii) cluster-based steering consistently outperforms random injection at matched amplification strength, qualitatively reproducing Hàon et al.; (iii) on a task with a dynamic motion profile, cluster steering at α=5 achieves a $\sim$6% end-effector speed increase over both the unsteered baseline and random injection (Cohen's $d \approx +0.3$); and (iv) the effect is non-monotonic in α and suppressed on precision tasks, consistent with polysemantic value vectors whose signal is diluted at small model scale. Prompt-level steering outperforms all neuron-level interventions, suggesting that at 500M parameters the model's behavioural representations are more accessible via its language interface than via individual FFN directions. We hope this study provides a foundation for further research into mechanistic interpretability of small VLAs and highlights the importance of model scale as a variable in activation steering studies.

Acknowledgments

We thank the course staff for their guidance and the authors of SmolVLA for releasing an accessible model. Compute was provided by Carnegie Mellon University and Apple MPS.

7 Team Contributions

- **Leo Wang:** Led implementation of value vector projection, layer-wise semantic analysis, and contributed to methodology and results interpretation.
- **Ashutosh Panda:** Set up SmolVLA environment, ran baseline experiments, and assisted with cluster sweep data collection.
- **Varun Chandra:** Designed and ran lean and cluster sweep experiments (290 rollouts, $\approx$30 h compute), identified final fast/slow value vectors, analysed motion metrics, and created result plots.
- **Grace Kaimburi:** Led final report writing and video script creation, coordinated literature review, and managed project timelines.

GitHub Repository

Project code and documentation: `https://github.com/Artifex2002/project_Lumina`

References

[1] K. Black et al. π_0: A vision language action flow model for general robot control. arXiv:2410.24164, 2024.

[2] T. Bricken et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023.

[3] A. Brohan et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. arXiv:2307.15818, 2023.

[4] H. Cunningham et al. Sparse autoencoders find highly interpretable features in language models. arXiv:2309.08600, 2023.

[5] M. Geva et al. Transformer feed-forward layers are key-value memories. arXiv:2012.14913, 2021.

[6] M. Geva et al. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. arXiv:2203.14680, 2022.

[7] C. Glanois et al. A survey on interpretable reinforcement learning. *Machine Learning*, 113(8):5847–5890, 2024.

[8] B. Hàon et al. Mechanistic interpretability for steering vision-language-action models. arXiv:2509.00328, 2025.

[9] M. J. Kim et al. OpenVLA: An open-source vision-language-action model. *CoRL 2024*.

[10] B. Liu et al. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. arXiv:2306.03310, 2023.

[11] K. Meng et al. Locating and editing factual associations in GPT. *NeurIPS 2022*.

[12] C. Olsson et al. In-context learning and induction heads. *Transformer Circuits Thread*, 2022.

[13] K. Park et al. The linear representation hypothesis and the geometry of large language models. *ICML 2024*.

[14] K. Pertsch et al. FAST: Efficient action tokenisation for vision-language-action models. arXiv:2501.09747, 2025.

[15] S. Pohland and C. Tomlin. Understanding the dependence of perception model competency on regions in an image. *xAI 2024*.

[16] M. Shukor et al. SmolVLA: A vision-language-action model for affordable and efficient robotics. arXiv preprint, 2025.

[17] T. H. Wang et al. Measuring interpretability of neural policies of robots with disentangled representation. *CoRL 2023*.

Appendix: Additional Value Vectors

Figure 2 lists other semantically interesting vectors identified during our analysis, which may be candidates for future steering experiments.

Notable vectors to amplify:

Layer 25, Vector 3 column 968	Yes	Fast', 'fast', ' fast', 'Fast'	Morphological: Case and spacing variations of the word 'fast'.
-			
Layer 23, Vector 9 column 574	Yes	suddenly', ' unexpectedly', ' randomly', ' Suddenly'	Grammar: Common adjectival/adverbial suffix '-ly'.
-			

Figure 2: Notable value vectors and their top tokens. The vector at Layer 25, column 968 encodes variations of "fast," while Layer 23, column 574 captures adverbial concepts like "suddenly." These represent promising candidates for future steering experiments.