Controlling Robot Behavior at Inference Time: Weights based SAE steering in Vision Language Action models

Sweta Jena*, Shruthi Muthukumar*, Adithya Casichetty Addepalli, Likhit Bhogadi, Ronit Jalihal, Hemang Jain, Ponnurangam Kumaraguru
Department of Computer Science
International Institute of Information Technology Hyderabad
Hyderabad, India

***Abstract*—Vision Language Action (VLA) models map multimodal inputs to robot actions, yet their interpretability remains critically underexplored (unlike in LLMs and VLMs) despite the direct physical consequences of failures. Furthermore, the scarcity of robotics data constrains conventional interpretability methods. We address both challenges via inference-time steering. We apply a Sparse Autoencoder (SAE) based steering method that operates directly on weight vectors rather than activations, disentangling polysemantic features to precisely amplify target concepts without additional training data. Using OpenVLA-Libero10 on long-horizon tasks, we first benchmark prompt based and prior weight space steering baselines. Our proposed SAE based steering reduces execution time by 0.5-1.42 seconds and improves task success at certain steering strengths on LIBERO-10 for the concept *fast*. Our work establishes the effectiveness of SAE-based weight-space steering framework for VLA models and surfaces fundamental asymmetries in concept controllability, offering key insights for safe, interpretable robot deployment. The code is available here.**

***Index Terms*—Vision-Language-Action Models, Sparse Autoencoders, Interpretability, Robotics, Concept Steering**

I. Introduction

Vision Language Action (VLA) models have emerged as a promising paradigm for embodied AI, enabling robots to map visual observations and natural language instructions directly to low-level control actions [1], [2]. Despite their versatility across physical manipulation tasks, a critical question remains underexplored: *how do VLA models internally encode linguistic concepts when generating robot actions?* Unlike LLMs, whose failures produce incorrect text, VLA failures may manifest as unsafe physical behaviors, making interpretability safety-critical.

Recent work has shown that FFN value vectors in transformers encode interpretable motion-relevant concepts [3]. A fundamental challenge in VLA interpretability is the scarcity of large, diverse robot demonstration datasets, which limits activation-based approaches that require the same to identify and steer based on concept representations. Weight-space steering sidesteps this bottleneck entirely: by operating directly on the FFN weight vectors, it enables inference-time concept control without any additional data collection. However, individual FFN neurons are often polysemantic - encoding multiple unrelated concepts simultaneously, making direct interventions unreliable and difficult to control precisely.

We propose a three-stage framework for precise concept steering in VLA models. We first benchmark two baselines: **(i)** prompt-based adverbial steering, augmenting task instructions with modifiers such as "quickly" or "slowly", and **(ii)** direct FFN weight-space steering following [3]. We then introduce our main method closely following [4]: an **SAE-based steering method** that decomposes polysemantic FFN neurons into monosemantic features via a Matryoshka Sparse Autoencoder, identifies concept-aligned features through LM-head vocabulary projection, and selectively amplifies them at inference time—entirely without additional training data. Concurrent work [5] applies SAEs to VLA activation vectors; our method instead operates directly on weight vectors, requiring no auxiliary datasets.

We evaluate all methods on OpenVLA-Libero10 [1] across long-horizon manipulation tasks on LIBERO-10. Our contributions are:

- SAE-based weight-space steering framework that disentangles polysemantic FFN weights for interpretable, training-free concept control.
- Empirical demonstration of steering across motion-based concepts for *fast* yields a **0.5-1.42 second reduction** in execution time and **varying improvement** in task success rate (depending on steering strength) on LIBERO-10.
- Feature visualization covering OpenVLA across the final four layers.

II. Related Works

a) Interpretability and Steering in VLA models.: VLA models map visual observations and language instructions directly to robot actions, but their internal representations remain difficult to interpret or control [6], [7]. Recent work [3] introduces a framework for analyzing VLA internals and

*Equal contribution.
Correspondence: sweta.jena@research.iiit.ac.in, shruthi.muthukumar@research.iiit.ac.in.

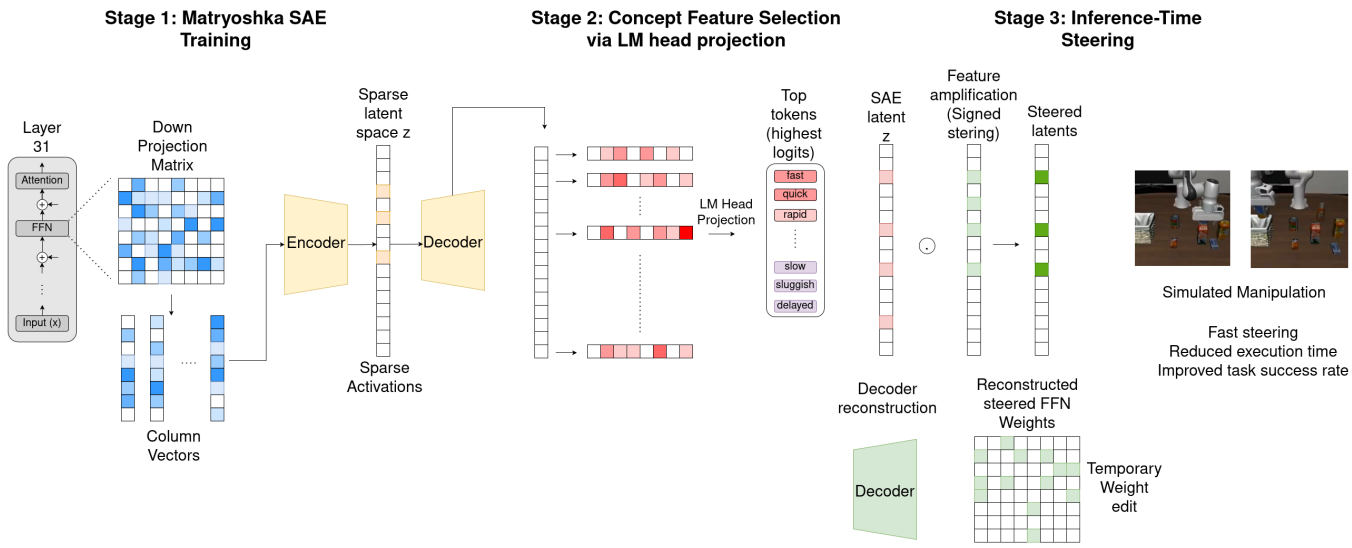

Fig. 1. Overall Methodology

steering behavior through activation interventions. The method projects FFN vectors into vocabulary space [8] to identify semantic directions such as speed or direction and modulates the corresponding activations at inference time to influence robot actions.

b) Concept-Level Editing and Sparse Feature Disentanglement.: [4] introduces a framework for identifying and removing conceptual knowledge directly within model parameters. The method first decomposes MLP parameter vectors into interpretable features, then identifies directions associated with a target concept and edits them to suppress the concept while preserving unrelated knowledge.

III. Methodology

A. Baseline: Prompt Based Steering

As an initial baseline, we explore prompt-based behavioral modulation. Task instructions are augmented with descriptive adverbial modifiers, with the intuition that the VLA's language encoder encodes such modifiers in its internal representations, propagating their effect to downstream action predictions. For example, *"pick up the cup"* becomes *"pick up the cup slowly"*. We experiment with the modifiers: *quick*, *rapid*.

B. Baseline: Weight-Space Steering of FFN Representations

We adopt a weight-space steering approach based on the FFN structure of transformer layers, following prior work by [3]. Given an input representation $x \in \mathbb{R}^{d_{\text{model}}}$, the FFN can be written as:

$$\text{FFN}(x) = W_{\text{down}}\, f_\theta(x) \tag{1}$$

where $f_\theta(x) \in \mathbb{R}^{d_{\text{ff}}}$ are the intermediate activations, and $W_{\text{down}} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{ff}}}$ is the down-projection matrix.

Let $w_i \in \mathbb{R}^{d_{\text{model}}}$ denote the i-th column of W_{down}. The FFN output can be expressed as:

$$\text{FFN}(x) = \sum_{i=1}^{d_{\text{ff}}} f_\theta^{(i)}(x)\, w_i \tag{2}$$

This formulation interprets the FFN output as a linear combination of fixed *value vectors* (down-projection vectors) w_i, weighted by input activations.

We analyze FFN semantics by projecting value vectors (columns of $W_{\text{down}}^{(\ell)}$) into vocabulary space via the LM head, $\ell_i^{(\ell)} = W_{\text{lm}}\, w_i^{(\ell)}$, and extracting top-$k$ tokens as interpretable concept descriptors. To steer toward a target concept c, we construct a concept vector v_c from modifier embeddings, rank value vectors by cosine similarity,

$$\text{sim}((\ell, i), c) = \frac{\ell_i^{(\ell)} \cdot v_c}{\|\ell_i^{(\ell)}\|\, \|v_c\|}, \tag{3}$$

and modulate the top-p activations in middle-to-late layers during inference—either by replacement ($a_i' = \alpha$) or scaled modulation ($a_i' = \alpha\, a_i$). While effective, individual vectors are often polysemantic, risking unintended side effects.

C. SAE-based steering

Inspired by recent concept-level interpretability and steering work with MLP weight vectors [4], we use a three-stage pipeline: (i) train a Matryoshka Sparse Autoencoder (SAE) [9] on the final layer MLP, (ii) identify concept-linked SAE features by projecting decoder directions into vocabulary space through the language-model head, and (iii) amplify the concept during inference by reconstructing steered MLP weights from the modified SAE activations. SAEs are well-suited for this task because their sparsity constraint encourages each latent feature to capture a single, coherent concept, thus reducing the polysemanticity inherent in raw MLP neurons where unrelated concepts are superimposed. This yields a "clean" decomposition of the MLP layer, thus enabling targeted editing instead of inadvertently perturbing unrelated concepts.

1) Stage 1- Matryoshka SAE Training: We treat each column of the MLP down-projection $W_{\text{down}} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{ff}}}$ as a training sample and fit a Matryoshka SAE [9] with encoder E and decoder D on the last layer (we also try this for the

penultimate layer separately), ensuring basis-compatibility for the vocabulary projection in Stage 2.

2) Stage 2- Concept Feature Selection via LM head projection: For each SAE feature f, we take the decoder direction $d_f \in \mathbb{R}^{d_{\text{model}}}$ and project it through the LM head:

$$\ell_f = W_{\text{lm}} d_f + b_{\text{lm}} \in \mathbb{R}^{|V|}.$$

We define a target lexicon $\mathcal{C}_{\text{fast}}$ of the target concept "fast" and its close synonyms. A feature is marked concept-relevant if at least m (we consider 1) of its top- or bottom-k (we consider 10) projected tokens appear in $\mathcal{C}_{\text{fast}}$, yielding positive and negative candidate sets $\mathcal{F}^+$ and $\mathcal{F}^-$.

3) Stage 3- Inference-Time Steering: First, the MLP neuron weight vectors are encoded into the SAE latent space,

$$Z = E(W_{\text{down}}^{\top}) \in \mathbb{R}^{d_{\text{ff}} \times F},$$

where F is the SAE dictionary size. We obtain the static concept feature mask $m \in \mathbb{R}^F$, where

$$m_f = \begin{cases} +\alpha & f \in \mathcal{F}^+ \\ -\alpha & f \in \mathcal{F}^- \\ 0 & \text{otherwise,} \end{cases}$$

with scale $\alpha > 0$. We decompose the SAE activations into steered and unmodified components: $Z^{\text{steer}} = Z \odot m$, $Z^{\text{original}} = Z \odot (1 - \mathbf{1}_{m \neq 0})$, where $\odot$ denotes element-wise multiplication and $\mathbf{1}_{m \neq 0}$ is an indicator vector that is 1 for selected concept features and 0 otherwise. Thus, Z^{steer} contains only the selected concept-related features, scaled by $\pm\alpha$, while Z^{original} contains all remaining SAE features unchanged. At each forward pass, a dynamic sign $s_i = \text{sign}(\sum_{b,t} h_{b,t,i})$ (where b is the batch size and t is the number of tokens) is computed from the the input activation (determined at inference time), and the steered neuron activation is

$$\tilde{z}_i = Z_i^{\text{original}} + s_i \, Z_i^{\text{steer}},$$

from which a temporary steered weight matrix $\tilde{W}_{\text{mlp}} = D(\tilde{Z})$ is reconstructed and applied only for that forward pass. This amplifies "fast"-associated latent directions while leaving the remaining functional subspace intact. Fig. 1 explains the Weights based SAE steering. The methodology is slightly different from [4] in two ways: 1) we modify the steering formulation for amplification of the target concept as opposed to erasure, 2) the MLP vectors to edit are determined at inference time depending on the overall neuron sign.

IV. Experiments

Setup. We evaluate all methods for the model OpenVLA-Libero10 on the **LIBERO-10** [10] Long Horizon benchmark. We perform 5 independent trials per task to account for execution variance. The base model is evaluated under consistent inference settings using a fixed random seed for reproducibility.

SAE Training. The Matryoshka SAE is trained on layers 28–31 (separately) for 100 epochs; we utilize layer 31 for all subsequent steering experiments due to its basis alignment with vocabulary space. We release a comprehensive collection of all trained Sparse Autoencoders (SAEs) used in our experiments here.

Steering Evaluation. To assess steering efficacy, we sweep the steering strength parameter, α. We evaluate the "fast" concept across a broad range ($\alpha \in [11, 39]$) and a window of ($\alpha \in [15, 25]$) for the "slow" concept.

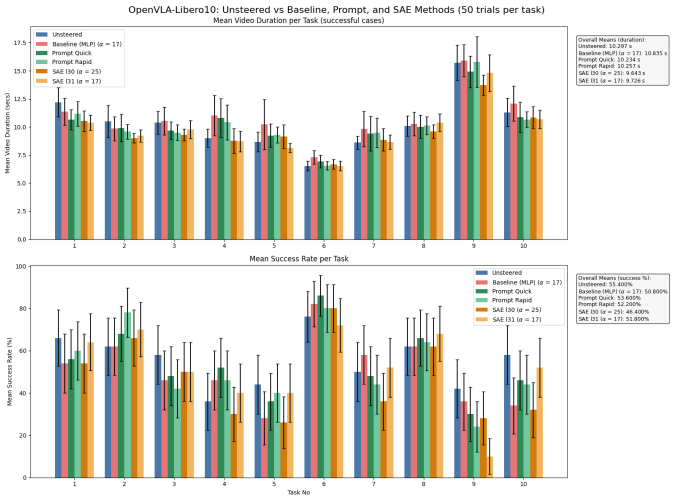

Fig. 2. Comparison of different steering methods in OpenVLA-Libero10 for the concept 'Fast': Unsteered vs Baseline (MLP-steered) vs Prompt-steered (appending 'quickly' or 'rapidly') vs SAE-steered. Layer steered is indicated in the legend (eg., l31 for layer 31). The time duration is measured on successful tasks only.

V. Results

We steer for target concepts such as 'Fast' and 'Slow'. As shown in Fig. 2, we compare the average success rates and time duration for 'Fast' steering. Since we are steering for the concept 'Fast', we expect the time duration (on successful trials) to be lesser than that of the unsteered model. We see that for prompt based steering, the average success rate drops, and for MLP based steering (baseline), the average time duration increases instead of decreasing. We see that with last-layer SAE steering, the time duration reduces by 0.571 seconds on average across different tasks (multiple trials). We also see a reduction in time by 0.654 seconds with penultimate layer (layer 30) SAE steering. However, for both layers, we see variation across all the tasks wherein the time duration increases for certain tasks. Additionally, we also see the average success rate for many tasks increasing for SAE steering. We are unsure as to why this is the case since we expect the success rate to drop post-steering. One possible hypothesis is that the SAE acts as a denoiser, thereby reducing any learned noise in the MLP parameters.

As shown in Table I and Table II, we also do a detailed analysis of steering strengths for the concepts 'Fast' and 'Slow'. We measure the average time across tasks (including successful trials only) as well as the maximum increase/decreased across tasks for a given steering strength. For the 'fast' sweep, we see the most reduction in time at strength 17 with a reduction of **1.42 seconds**.

TABLE I

OpenVLA-Libero10 SAE steering (layer 31): Task Performance by Strength across 10 tasks - Concept Fast (5 trials per task)

Strength	Success (%)	Avg. Time (s)	Avg. Reduction (s)	Max Reduction (s)
0 (unsteered)	46	10.54		
11	60	9.32	1.21	4.57
13	52	9.9	0.633	3.47
15	62	9.56	0.97	4.74
17	52	8.81	**1.4244**	5.97
19	54	9.99	0.546	2.95
21	56	9.42	1.117	**6.27**
23	46	9.2	0.51	5.05
25	56	9.49	1.046	3.46
27	46	9.53	0.5422	4.12
29	58	10.04	0.5	3.03
31	38	9.49	0.7389	5.21
33	26	10.31	-0.0811	5.07
35	0	0	NA	NA
37	0	0	NA	NA
39	0	0	NA	NA

TABLE II

OpenVLA-Libero10 SAE steering (layer 31): Task Performance by Strength across 10 tasks - Concept Slow (5 trials per task)

Strength	Success (%)	Avg. Time (s)	Avg. Increase (s)	Max Increase (s)
0 (unsteered)	46	10.54		
15	66	11.955	1.42	5.6
18	53	10.834	0.299	6.16
≥ 20	0	0	NA	NA

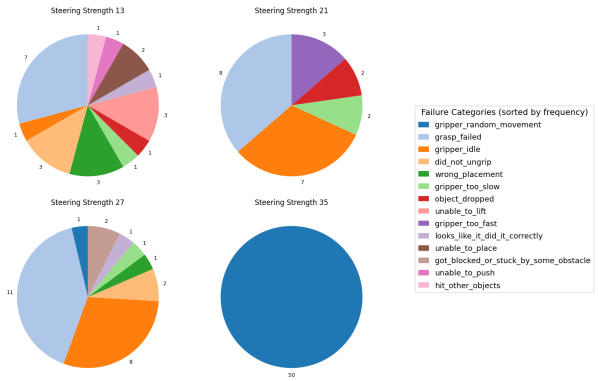

Fig. 3. Distribution of failure categories across steering strengths. Model: OpenVLA-Libero10, SAE Steering (layer 31), Target concept: Fast

Additionally, we perform a a detailed failure analysis (human evaluation) across steering strengths, as shown in Fig. 3. In general, a variety of broad failure categories were observed. At lower steering magnitudes, failures remain relatively diverse and often preserve partial task structure. As the steering coefficient increases, failures become increasingly dominated by repetitive and degenerate behaviors. Grasp failure is the most common cause of failure across steering strengths. Gripper being idle was a failure cause which also increased at higher steering strengths, primarily because higher speeds caused trouble in the precise positioning of the gripper around the object. At steering strength 35, the model collapses and the gripper arm starts doing random movements.

To facilitate analysis of learned representations, we develop an interactive visualization tool for inspecting the semantic structure of the SAE decoder directions, similar to [11]. Check the codebase for details.

VI. Conclusion, Limitations, and Future Work

We present a framework for inference-time concept steering in VLA models via SAE-based weight-space editing, requiring no additional training data. While our method successfully steers behavioral attributes such as execution speed: achieving a 0.5–1.42 second reduction and improvement in task success rate at certain steering strengths for *fast* steering, behavioral changes remain modest in magnitude and inconsistent across tasks. The observed increase in success rate under steering is itself an open question warranting further investigation.

A key limitation is an asymmetry in concept controllability: steering for *fast* is robust across a broad operational range, whereas steering for *slow* produces reliable deceleration only within a narrow window of coefficients (15–18), with behavioral collapse outside it. This asymmetry has direct safety implications – miscalibrated interventions intended to make a robot more cautious may instead destabilize its behavior entirely, manifesting as unsafe physical actions rather than benign errors.

Future work should explore alternative disentanglement methods beyond SAEs, and whether steering in the visual encoder space produces complementary behavioral effects, and develop principled criteria for selecting steering strength to avoid instability.

REFERENCES

[1] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi *et al.*, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[2] M. J. Kim, C. Finn, and P. Liang, "Fine-tuning vision-language-action models: Optimizing speed and success," *arXiv preprint arXiv:2502.19645*, 2025.

[3] B. Häon, K. Stocking, I. Chuang, and C. Tomlin, "Mechanistic interpretability for steering vision-language-action models," *arXiv preprint arXiv:2509.00328*, 2025.

[4] Y. Gur-Arieh, C. H. Suslik, Y. Hong, F. Barez, and M. Geva, "Precise in-parameter concept erasure in large language models," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025, pp. 18 997–19 017.

[5] A. Swann, L. McGranahan, H. Buurmeijer, M. Kennedy III, and M. Schwager, "Sparse autoencoders reveal interpretable and steerable features in vla models," *arXiv preprint arXiv:2603.19183*, 2026.

[6] C. Mitra, Y. Luo, R. Saravanan, D. Niu, A. Pai, J. Thomason, T. Darrell, A. Anwar, D. Ramanan, and R. Herzig, "Mechanistic finetuning of vision-language-action models via few-shot demonstrations," *arXiv preprint arXiv:2511.22697*, 2025.

[7] H. Buurmeijer, C. A. Alonso, A. Swann, and M. Pavone, "Observing and controlling features in vision-language-action models," *arXiv preprint arXiv:2603.05487*, 2026.

[8] M. Geva, R. Schuster, J. Berant, and O. Levy, "Transformer feed-forward layers are key-value memories," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 5484–5495.

[9] B. Bussmann, N. Nabeshima, A. Karvonen, and N. Nanda, "Learning multi-level features with matryoshka sparse autoencoders," *arXiv preprint arXiv:2503.17547*, 2025.

[10] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, "Libero: Benchmarking knowledge transfer for lifelong robot learning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 44 776–44 791, 2023.

[11] B. Grant, X. Zhao, and P. Wang, "Not all features are created equal: A mechanistic study of vision-language-action models," *arXiv preprint arXiv:2603.19233*, 2026.