Mechanistic Interpretability for Steering Vision-Language-Action Models

1st Bear Häon
Electrical Engineering and Computer Sciences
UC Berkeley
bear.haon@berkeley.edu

1st Kaylene Stocking
Electrical Engineering and Computer Sciences
UC Berkeley
kaylene@berkeley.edu

3rd Ian Chuang
Electrical Engineering and Computer Sciences
UC Berkeley

4th Claire Tomlin
Electrical Engineering and Computer Sciences
UC Berkeley

***Abstract*—Note: This is a shortened version of work perviously published at the 2025 Conference on Robot Learning. Vision-Language-Action (VLA) models are a promising path to realizing generalist embodied agents that can quickly adapt to new tasks, modalities, and environments. However, methods for interpreting and steering VLAs fall far short of classical robotics pipelines, which are grounded in explicit models of kinematics, dynamics, and control. This lack of mechanistic insight is a central challenge for deploying learned policies in real-world robotics, where robustness and explainability are critical. Motivated by advances in mechanistic interpretability for large language models, we introduce the first framework for interpreting and steering VLAs via their internal representations, enabling direct intervention in model behavior at inference time. We project feedforward activations within transformer layers onto the token embedding basis, identifying sparse semantic directions - such as *speed* and *direction* - that are causally linked to action selection. Leveraging these findings, we introduce a general-purpose activation steering method that modulates behavior in real time, without fine-tuning, reward signals, or environment interaction. We evaluate this method on two recent open-source VLAs, π_0 and OpenVLA, and demonstrate zero-shot behavioral control in simulation (LIBERO) and on a physical robot (UR5). This work demonstrates that interpretable components of embodied VLAs can be systematically harnessed for control—establishing a new paradigm for transparent and steerable foundation models in robotics.**

***Index Terms*—Mechanistic Interpretability, Vision-Language-Action Models, Foundation Models for Robotics**

I. Introduction

Large language-conditioned models represent a paradigm shift for robotics. By leveraging Internet-scale natural language and image data, these models promise to yield generalist robot policies that can follow directions, exhibit commonsense reasoning about their environments, and quickly adapt to novel tasks and scenarios.

One leading type of frontier robotics model is the Vision-Language-Action (VLA) model, where a pre-trained Vision-Language Model (VLM) is trained to produce robot actions conditioned on a task description and image observations. While most VLAs are designed to be fine-tuned for specific tasks, future models may even be able to perform a variety of tasks without the need for fine-tuning. Indeed, the π_0 model [1] can often complete zero-shot evaluation tasks on the DROID platform [2] and Gemini Robotics demonstrates zero-shot control via code generation [3]. This promise of generality and ease of deployment signals a large shift in how we deploy robot systems. In contrast to rigorous evaluation and certification on individual tasks in factories or warehouses, robots will be placed in novel environments and expected to behave well "out-of-the-box". This raises important questions about safety, robustness, and transparency. How and when can we trust that VLA policies will behave safely? When they fail, how can we diagnose and fix the underlying problems?

A similar set of questions and challenges is being faced by Large Language Model (LLM) researchers. While they have by no means been solved, the field of *mechanistic interpretability* has yielded a number of exciting insights about the inner workings of LLMs. For example, researchers have discovered the functional role of specific attention heads [4] and identified semantically interpretable concepts in model activations [5], [6]. There is also evidence that mechanistic interpretability can be useful for diagnosing problems with models in practice [7]. These insights lead us to ask: can mechanistic interpretability also help with understanding and shaping the behavior of frontier models in robotics?

In this work, we present a series of experiments that together allow us to provide a resounding affirmative answer. We focus our efforts on an interpretability method that analyzes the semantic meanings of vectors in the output weight matrix of the transformer block's feed-forward layer (FFN) [8]. Although this technique has less power to disentangle concepts that might be spread across multiple neurons in the model than alternatives such as sparse autoencoders [5], it allows us to interpret and steer the model without any additional training data. This is a key advantage in robotics where training data is expensive and difficult to acquire, and where available open datasets are small compared to what is available in the natural language domain. Our chosen method leads us to several

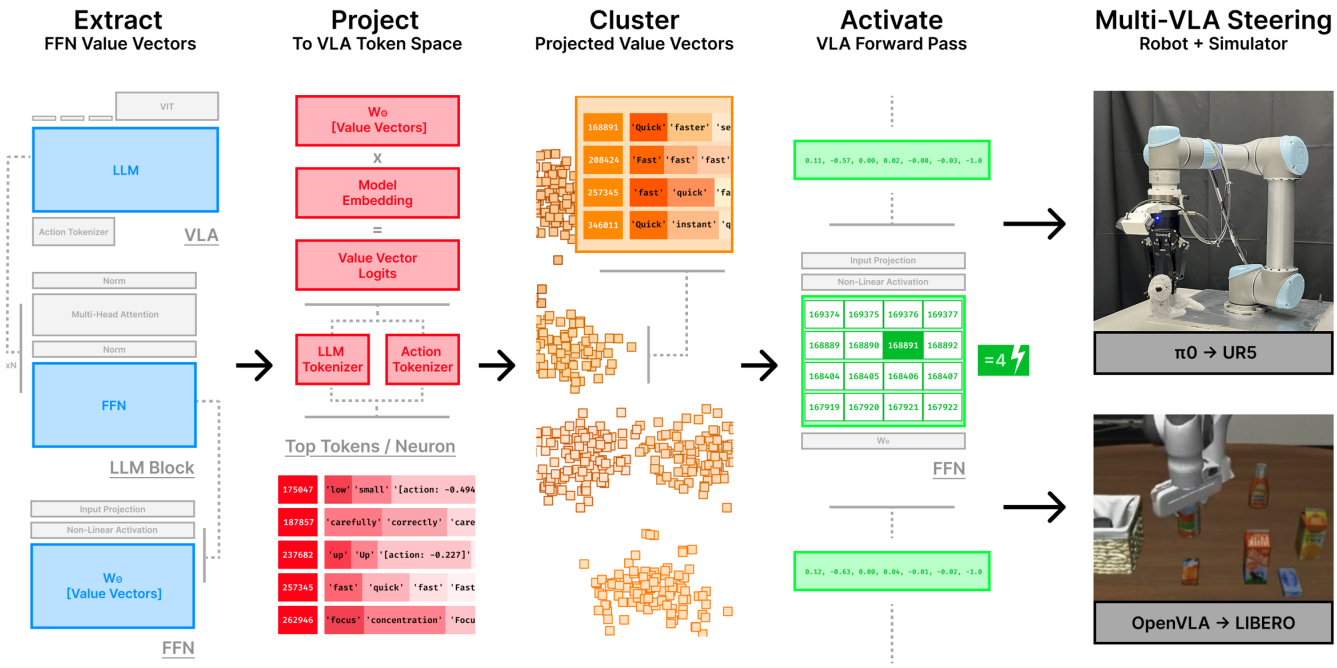


Fig. 1: **We present a framework for steering Vision-Language-Action (VLA) models.** We extract FFN vectors, project them to the VLA token space, cluster them by semantic alignment, and inject activations at inference time to modulate behavior. Our experiments demonstrate interpretable zero-shot control in both simulation (OpenVLA in LIBERO) and on a physical robot (π_0 on a UR5).

surprising conclusions about VLAs, including:

1) Despite being trained to produce only robot actions, the internal activity of the VLA model is largely semantic, with less than 25% of FFN neurons being rewired for action prediction and a large fraction of the remainder representing clear, semantically interpretable concepts.
2) Internal model concepts such as "slow" are causally linked to robot actions that express these concepts (i.e. moving the end effector more slowly), despite the model being trained without any feedback on the meaning or consequences of different action tokens.
3) Targeted interventions on model concepts allow VLA behavior to be modified at inference time out-of-the-box, without any fine-tuning data or environment interaction required to select the intervention.

These contributions represent the first attempt to use mechanistic interpretability techniques developed for LLMs with large robotics models. Our exciting initial results speak to the great promise of adapting these techniques to the robotics context, where there is an especially strong need for AI models that are safe and transparent.

II. Interpreting VLAs

Deep neural networks build their predictions layer by layer, with features in one layer supporting further computations and representations in the next. According to the "linear representation hypothesis", intermediate network layers often represent different features or concepts as directions in the latent embedding space [9]. This means that if, for example, a model represents the concept of "fast", if we can identify the corresponding direction in its latent space, we can understand when and how the model uses this concept to produce its final output. Serendipitously, it turns out that for many large models, semantically meaningful directions align with individual neurons. This fact has been used to help interpret both vision [10] and language [8], [11] models.

Specifically, we follow the approach outlined in [8] and focus on the feedforward layer of each transformer block:

$$\text{FFN}(x) = f_\theta(x)^T W_\theta \tag{1}$$

where $x \in \mathbb{R}^n$ is the input to the layer, $f_\theta(x) \in \mathbb{R}^m$ is an input-dependent set of activations, and $W_\theta \in \mathbb{R}^{m \times n}$ is an input-independent parameter matrix. Using $w_\theta^{(i)} \in \mathbb{R}^n$ to refer to the ith row of W_θ, Equation (1) can be rewritten as follows:

$$\text{FFN}(x) = \sum_i [f_\theta(x)]_i w_\theta^{(i)} \tag{2}$$

This allows us to interpret the FFN as a weighted sum over the $w_\theta^{(i)}$ vectors, which we refer to as *value vectors*. Since these vectors are independent of the inputs, they can be thought of as basis functions for the output of the FFN. Furthermore, as they belong to the same linear space as the final output of the transformer, they can be interpreted as probability distributions over possible output tokens. Therefore, we can assign semantic meanings to value vectors based on the set of tokens to which they assign the highest probability.

A. Value vector analysis

VLA Value Vectors Retain Semantic Meanings. First, we aimed to quantify whether the VLM value vectors represent semantically interpretable concepts, and if so, whether this is preserved after VLA training. We focused on π_0 and its base

pretrained VLM, PaliGemma-3B [12]. We randomly selected 10 value vectors from each layer of the VLM and VLA models and examined whether at least 4 of the top 30 tokens followed a common pattern (adopting the methodology from [8]. Value vectors throughout both the VLM and VLA models exhibit identifiable patterns, many of them semantic, at similar rates. Therefore, VLA training does not cause FFN outputs to lose meaningful organization of concepts, despite producing only non-semantic tokens at the final output.

Action Tokens Appear in All VLA Layers. For both OpenVLA and π_0, action tokens are most common at the final layer, but make up at least a few percent of every layer. This suggests that there is not a hard transition from "thinking about the task" in earlier layers to "thinking about control" in later layers. Instead, it appears VLA models learn to reason about and refine their control action predictions continuously, starting from their earliest computations.

III. Steering VLAs

We formalize our method for *interpretable activation-level steering* of VLAs (Figure 1). This technique modifies selected neurons within the FFN submodules of transformer blocks inside the VLA's base VLM, steering robot behavior in real time - without fine-tuning or environment reward signal.

Let $x \in \mathbb{R}^n$ be the residual input to a transformer FFN. As shown in Equation (2), the FFN output is a sum of fixed value vectors $w_\theta^{(i)} \in \mathbb{R}^n$ weighted by input-dependent activations $f_\theta(x) \in \mathbb{R}^m$. Motivated by evidence that individual FFN neurons encode semantic features [5], [11], we override a subset $\mathcal{S} \subseteq \{1, \ldots, m\}$ of activations using a fixed scalar $\alpha \in \mathbb{R}$. Each $\mathcal{S}$ corresponds to an interpretable neuron cluster aligned with a control concept - e.g., *fast*, *up*, *careful* - identified by grouping neurons with similar token projections (Figure 1, steps 1–3), either manually or via kNN over semantic embeddings.

$$\tilde{f}_\theta^{(i)}(x) = \begin{cases} \alpha & \text{if } i \in \mathcal{S} \\ [f_\theta(x)]_i & \text{otherwise} \end{cases} \tag{3}$$

$$\text{FFN}_{\text{steered}}(x) = \sum_{i=1}^{m} \tilde{f}_\theta^{(i)}(x) \cdot w_\theta^{(i)} \tag{4}$$

This induces a residual shift $\Delta x = \text{FFN}_{\text{steered}}(x) - \text{FFN}(x)$ that propagates through the transformer and modulates the final VLA action token distribution.

A. Simulation Experiments

We evaluate our VLA steering method within the suite of ten long-horizon manipulation tasks making up the LIBERO-Long benchmark [13]. All simulation experiments use the 7B-parameter OpenVLA model with the fine-tuned LIBERO-Long checkpoint.

Steering Motion Interventions. To evaluate whether interpretable semantic structure in VLA value vectors can steer physical behavior, we construct *fast* and *slow* aligned clusters by manually selecting value vectors whose top-projected tokens reflect motion magnitude semantics. These clusters are injected into the model's residual stream with a fixed coefficient, sweeping over cluster sizes $[10, 20]$ and activation strengths $[2, 4, 6, 8, 10, 20]$. Each configuration is evaluated over 10 long-horizon tasks, using 10 rollouts per task. The fast clusters consistently induce larger end-effector displacements across tasks, with an average improvement of 27.73% over slow clusters and maximum gains of 148.54% in some configurations. All 10 matched comparisons yield statistically significant differences ($p < 0.001$, paired t-test), and effect sizes ($d = -0.091$ – 1.419) suggest consistent directional influence. These results demonstrate that directional behavior can be modulated by activating semantically interpretable value vectors alone.

B. Robot Experiments

We also evaluate our VLA steering method using the 3B-parameter π_0-FAST VLA on a real robot. Specifically, we test the ability to steer binary control opposites - *slow vs. fast* and *low vs. high* - in two pick-and-place tasks with a UR5 robot arm.

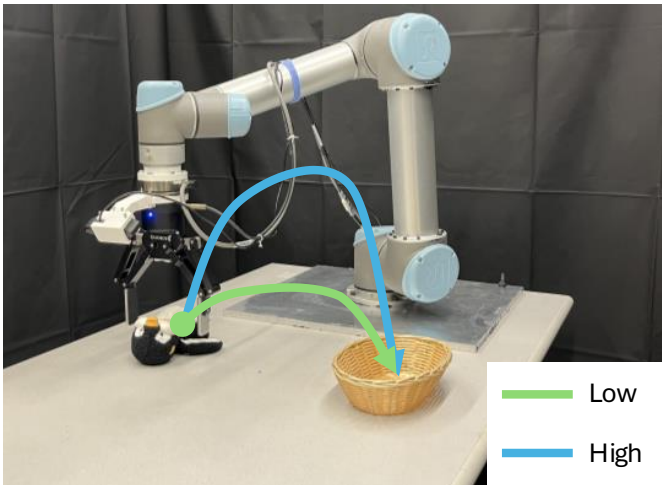
Fine-Tuning. To enable meaningful evaluation, we fine-tune with LoRA [14] on small datasets collected from our robot with variations in behavior during task completion. Crucially, we do not explicitly label these variations in the VLA prompt during fine-tuning, allowing us to test whether the VLA can reproduce these specific behaviors through steering by associating them with high-level semantic concepts.

Robot Tasks. We evaluate steering on two tasks (see Figure 2). The first task, *Low/High Transport*, involves the robot picking up a toy penguin and placing it in a basket and includes 75 episodes with unlabeled variations in how high the penguin is lifted during transport. The second task, *Slow/Fast Transport*, involves the robot picking up a toy seal and placing it on a specified plate and includes 120 episodes with variations in the speed of the robot's motion.

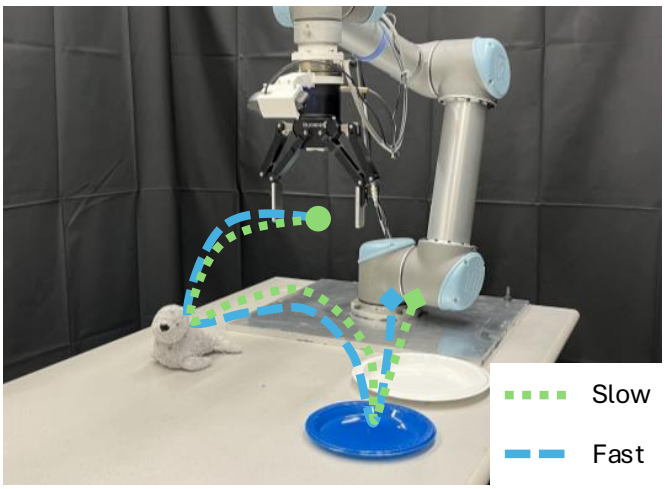
For both tasks, we evaluate steering interventions where we hand-select and upweight a cluster of semantically meaningful vectors. We create a *low* and *high* themed cluster for *Low/High Transport* and *slow* and *fast* themed cluster for *Slow/Fast Transport*. We also compare these interventions against several baselines: (1) running the model without any intervention, (2) adding qualitative descriptors to the prompt (e.g. appending *low* to the beginning of the prompt), and (3) applying steering with randomly upweighted vectors.

Task Evaluation. For *Low/High Transport*, we perform 10 independent rollouts for each steering variant and baseline. We record the maximum height across 10 rollouts for each variant in Figure 3a. In *Slow/Fast Transport*, we focus on measuring the robot's speed. To do this in a consistent manner we run inference for all steering variants and baselines simultaneously, but only execute the baseline predicted action with no intervention. We measure average end-effector displacement between each step along the trajectory across 10 rollouts for each variant and plot results in Figure 3b.

Robot Results. Our results suggest that steering interventions can meaningfully influence the actions of π_0-FAST. In the *Low/High Transport* task, the *low* intervention generated

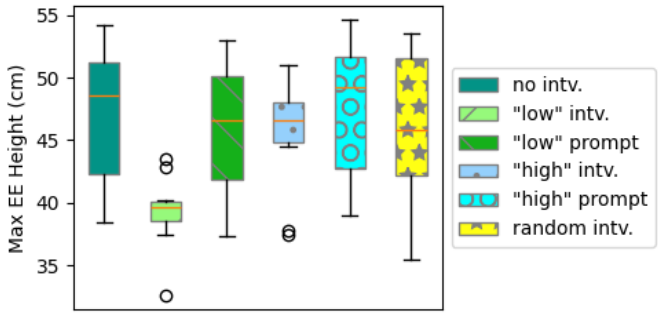


(a) **Low/high transport.** The robot picks up a toy penguin and places it into a basket, with variations in the robot's trajectory height during data collection.

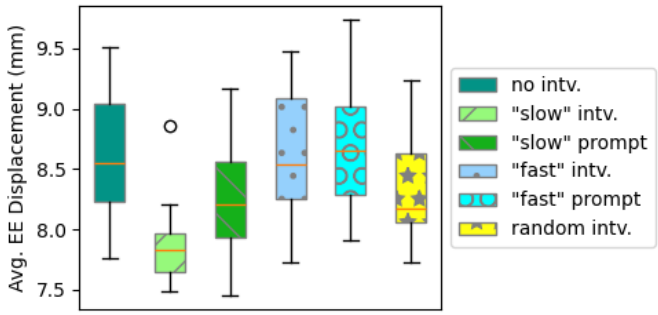


(b) **Slow/fast transport.** The robot picks up a toy seal and places it onto a plate, with variations in the robot's movement speed during data collection.

Fig. 2: **Physical robot experiments:** Steering π_0 on a UR5.



(a) **Low/high transport.** Box plots for maximum end-effector height (cm) showing distribution across 10 rollouts for each steering intervention (intv.) and baseline.



(b) **Slow/fast transport.** Box plots for average end-effector displacement (mm) between each successive action showing distribution across 10 rollouts for each steering intervention (intv.) and baseline.

Fig. 3: **Physical robot experiments:** Binary control opposites.

the lowest overall trajectories. Similarly, the *slow* intervention in *Slow/Fast Transport* resulted in the slowest overall movements. However, while *low* and *slow* interventions clearly altered behavior, their opposites (*high* and *fast*) resembled the baseline without intervention. This may be because the model already "considers" the baseline trajectory to be fast and high. Nevertheless, the minimal difference between using a random intervention compared to no intervention suggests that choosing semantically meaningful vectors can more effectively guide the model towards desired behavior. Finally, we found that steering interventions were more impactful than simply adding target descriptive words to the prompt. In both tasks, modifying the prompt had weaker impact than our steering intervention.

IV. Discussion and Conclusion

This work shows that VLAs can be steered via interpretable internal components. By activating sparse, semantically meaningful neurons in FFN layers—aligned with concepts like "up" and "slow"—we modulate robot behavior at inference time. Experiments demonstrate that our method generalizes across models and tasks, influencing both simulated and real-world behavior, including long-horizon LIBERO tasks and binary control opposites on a UR5.

From a mechanistic perspective, this work provides causal evidence that VLAs retain structured, compositional semantics even after fine-tuning. That activating a small set of semantically aligned neurons can shift motion behavior suggests control-relevant features remain accessible and manipulable. These effects emerge without explicit behavioral supervision, indicating that VLAs internalize semantic concepts during pretraining in a way that supports compositional reasoning across layers.

Looking ahead, activation steering opens a new interpretable control interface for guiding VLA behavior. As embodied foundation models become more capable, steering techniques could play an important role in inspecting, auditing, and intervening in how models reason and act. We see this approach—using just one tool from the mechanistic interpretability toolbox—as a step toward a broader class of transparent, steerable, and semantically grounded control tools for robotics.

REFERENCES

[1] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky, "π_0: A vision-language-action flow model for general robot control," 2024. [Online]. Available: https://arxiv.org/abs/2410.24164

[2] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine, "FAST: Efficient action tokenization for vision-language-action models," 2025. [Online]. Available: https://arxiv.org/abs/2501.09747

[3] G. R. Team, S. Abeyruwan, J. Ainslie, J.-B. Alayrac, M. G. Arenas, T. Armstrong, A. Balakrishna, R. Baruch, M. Bauza, M. Blokzijl, S. Bohez, K. Bousmalis, A. Brohan, T. Buschmann, A. Byravan, S. Cabi, K. Caluwaerts, F. Casarini, O. Chang, J. E. Chen, X. Chen, H.-T. L. Chiang, K. Choromanski, D. D'Ambrosio, S. Dasari, T. Davchev, C. Devin, N. D. Palo, T. Ding, A. Dostmohamed, D. Driess, Y. Du, D. Dwibedi, M. Elabd, C. Fantacci, C. Fong, E. Frey, C. Fu, M. Giustina, K. Gopalakrishnan, L. Graesser, L. Hasenclever, N. Heess, B. Hernaez, A. Herzog, R. A. Hofer, J. Humplik, A. Iscen, M. G. Jacob, D. Jain, R. Julian, D. Kalashnikov, M. E. Karagozler, S. Karp, C. Kew, J. Kirkland, S. Kirmani, Y. Kuang, T. Lampe, A. Laurens, I. Leal, A. X. Lee, T.-W. E. Lee, J. Liang, Y. Lin, S. Maddineni, A. Majumdar, A. H. Michaely, R. Moreno, M. Neunert, F. Nori, C. Parada, E. Parisotto, P. Pastor, A. Pooley, K. Rao, K. Reymann, D. Sadigh, S. Saliceti, P. Sanketi, P. Sermanet, D. Shah, M. Sharma, K. Shea, C. Shu, V. Sindhwani, S. Singh, R. Soricut, J. T. Springenberg, R. Sterneck, R. Surdulescu, J. Tan, J. Tompson, V. Vanhoucke, J. Varley, G. Vesom, G. Vezzani, O. Vinyals, A. Wahid, S. Welker, P. Wohlhart, F. Xia, T. Xiao, A. Xie, J. Xie, P. Xu, S. Xu, Y. Xu, Z. Xu, Y. Yang, R. Yao, S. Yaroshenko, W. Yu, W. Yuan, J. Zhang, T. Zhang, A. Zhou, and Y. Zhou, "Gemini robotics: Bringing ai into the physical world," 2025. [Online]. Available: https://arxiv.org/abs/2503.20020

[4] C. Olsson, N. Elhage, N. Nanda, N. Joseph, N. DasSarma, T. Henighan, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, D. Drain, D. Ganguli, Z. Hatfield-Dodds, D. Hernandez, S. Johnston, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. Brown, J. Clark, J. Kaplan, S. McCandlish, and C. Olah, "In-context learning and induction heads," *Transformer Circuits Thread*, 2022, https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

[5] T. Bricken, A. Templeton, J. Batson, B. Chen, A. Jermyn, T. Conerly, N. Turner, C. Anil, C. Denison, A. Askell, R. Lasenby, Y. Wu, S. Kravec, N. Schiefer, T. Maxwell, N. Joseph, Z. Hatfield-Dodds, A. Tamkin, K. Nguyen, B. McLean, J. E. Burke, T. Hume, S. Carter, T. Henighan, and C. Olah, "Towards monosemanticity: Decomposing language models with dictionary learning," *Transformer Circuits Thread*, 2023, https://transformer-circuits.pub/2023/monosemantic-features/index.html.

[6] H. Cunningham, A. Ewart, L. Riggs, R. Huben, and L. Sharkey, "Sparse autoencoders find highly interpretable features in language models," 2023. [Online]. Available: https://arxiv.org/abs/2309.08600

[7] S. Marks, J. Treutlein, T. Bricken, J. Lindsey, J. Marcus, S. Mishra-Sharma, D. Ziegler, E. Ameisen, J. Batson, T. Belonax, S. R. Bowman, S. Carter, B. Chen, H. Cunningham, C. Denison, F. Dietz, S. Golechha, A. Khan, J. Kirchner, J. Leike, A. Meek, K. Nishimura-Gasparian, E. Ong, C. Olah, A. Pearce, F. Roger, J. Salle, A. Shih, M. Tong, D. Thomas, K. Rivoire, A. Jermyn, M. MacDiarmid, T. Henighan, and E. Hubinger, "Auditing language models for hidden objectives," 2025. [Online]. Available: https://arxiv.org/abs/2503.10965

[8] M. Geva, A. Caciularu, K. R. Wang, and Y. Goldberg, "Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space," 2022. [Online]. Available: https://arxiv.org/abs/2203.14680

[9] K. Park, Y. J. Choe, and V. Veitch, "The linear representation hypothesis and the geometry of large language models," in *International Conference on Machine Learning*. PMLR, 2024, pp. 39 643–39 666.

[10] C. Olah, A. Mordvintsev, and L. Schubert, "Feature visualization," *Distill*, 2017, https://distill.pub/2017/feature-visualization.

[11] M. Geva, R. Schuster, J. Berant, and O. Levy, "Transformer feed-forward layers are key-value memories," 2021. [Online]. Available: https://arxiv.org/abs/2012.14913

[12] L. Beyer, A. Steiner, A. S. Pinto, A. Kolesnikov, X. Wang, D. Salz, M. Neumann, I. Alabdulmohsin, M. Tschannen, E. Bugliarello, T. Unterthiner, D. Keysers, S. Koppula, F. Liu, A. Grycner, A. Gritsenko, N. Houlsby, M. Kumar, K. Rong, J. Eisenschlos, R. Kabra, M. Bauer, M. Bošnjak, X. Chen, M. Minderer, P. Voigtlaender, I. Bica, I. Balazevic, J. Puigcerver, P. Papalampidi, O. Henaff, X. Xiong, R. Soricut, J. Harmsen, and X. Zhai, "PaliGemma: A versatile 3b vlm for transfer," 2024. [Online]. Available: https://arxiv.org/abs/2407.07726

[13] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, "Libero: Benchmarking knowledge transfer for lifelong robot learning," 2023. [Online]. Available: https://arxiv.org/abs/2306.03310

[14] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," 2021. [Online]. Available: https://arxiv.org/abs/2106.09685