Unveiling the Feature Geometry of Robot Foundation Models via Nyström NCut

Ningze Zhong, Kyle Zhang, Ily Rafaeli
University of Pennsylvania
{zhong666, kyle100, ilyr}@seas.upenn.edu

Abstract

Vision-Language-Action (VLA) models have emerged as powerful robot foundation models, yet their internal representations remain largely opaque. We apply Nyström Normalized Cuts to visualize and analyze feature geometry across VLA hidden layers, revealing how these models process multimodal inputs during manipulation tasks. Our analysis uncovers strong object-centric semantic clustering and sensitivity to failure indicators, but also exposes critical limitations in spatial reasoning and language grounding. Through controlled experiments on OpenVLA, we demonstrate that visual features dominate over language by a factor of 10×, and that spatial information fails to generalize despite perfect training-set memorization. These findings provide actionable insights for debugging and improving robot foundation models.

1 Introduction

Robot foundation models built on vision-language architectures [4, 5] have demonstrated remarkable capabilities in following natural language instructions for diverse manipulation tasks, from tool use to contact-rich assembly. Despite their impressive empirical performance, these models remain black boxes. When a VLA fails to grasp an object or misinterprets an instruction, practitioners cannot easily diagnose whether the failure stems from poor visual encoding, weak language understanding, or faulty cross-modal fusion. This opacity poses significant challenges for debugging, model improvement, and safe deployment in real-world scenarios.

We propose using Nyström Normalized Cuts [6] as a visual debugging tool for analyzing VLA internal representations. NCut performs spectral clustering on neural feature representations, mapping tokens to colors based on their position in the graph Laplacian's eigenspace. Our central hypothesis is that well-separated clusters indicate learnable structure: if NCut easily partitions features by semantic categories, downstream layers can exploit this structure for robust action prediction. In our exploration, this assumption has proven effective for debugging vision models like LLaVA, where we can identify exactly which layers fail to preserve task-relevant information by observing cluster coherence across controlled perturbations.

We conduct a comprehensive analysis of OpenVLA [4] across multiple dimensions. First, we examine text-image semantics by clustering vision and language tokens jointly, revealing which objects the model attends to and how language anchors to visual regions. Second, we track temporal dynamics through manipulation sequences, identifying cluster transitions that correlate with distinct task phases like approach, grasp, and release. Third, we quantify input sensitivity by systematically perturbing images versus text and measuring the resulting feature-space shifts. Fourth, we probe spatial intelligence by training probes to predict camera poses from VLA embeddings. Finally, we trace gradient flow to understand which feature clusters drive action predictions.

Our findings reveal both strengths and critical weaknesses. While VLAs recognize task-relevant objects and implicitly detect failure cues like object tilting or gripper slippage, they exhibit a strong bias toward visual information over language, with image perturbations producing 10× larger representation changes than text modifications. Moreover, spatial information appears to be memorized rather than learned: VLAs achieve near-zero training error on pose regression but fail completely on held-out viewpoints, suggesting they encode appearance-pose pairs without geometric understanding. We validate these insights through simulation experiments where NCut-guided debugging improves Pick and Place success rates from 71% to 80%.

2 Related Work

Vision-Language-Action Models. Recent VLA architectures [4, 5] fine-tune large vision-language models on robot trajectory data, leveraging broad priors from internet-scale pretraining to generalize across novel objects, scenes, and tasks. OpenVLA combines a pretrained vision transformer with a language model backbone featuring cross-attention layers, enabling end-to-end training from pixels and text to actions. OpenPi 0.5 extends this paradigm with improved data scaling and architectural refinements. While these models achieve strong empirical results, they lack built-in interpretability mechanisms. Prior work on failure detection [1, 2] trains separate models to identify errors post-hoc; in contrast, we analyze internal representations to understand inherent failure modes and guide targeted improvements.

Neural Network Interpretability. Attention visualization and gradient-based saliency maps are standard interpretability tools, but they provide only coarse spatial attributions and struggle with transformer architectures where information flows through complex residual paths. Spectral clustering methods like Normalized Cuts offer complementary insights by partitioning features based on global affinity structure rather than local gradient signals. Recent work has applied NCut to vision transformers for understanding semantic emergence across layers, but this approach has not been systematically extended to multimodal robot learning systems.

3 Method

3.1 Nyström Normalized Cuts

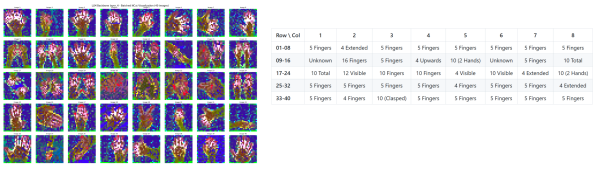
Given features $\mathbf{F} \in \mathbb{R}^{N \times d}$ extracted from a batch of B images with $N = B \cdot h \cdot w$ total patch tokens, we construct an affinity matrix $\mathbf{W}$ where $W_{ij} = \exp(-\|\mathbf{f}_i - \mathbf{f}_j\|^2 / 2\sigma^2)$ captures pairwise token similarity. Normalized Cuts seeks a graph partition that minimizes the criterion

$$\mathrm{NCut}(\mathcal{A}, \mathcal{B}) = \frac{\mathrm{cut}(\mathcal{A}, \mathcal{B})}{\mathrm{vol}(\mathcal{A})} + \frac{\mathrm{cut}(\mathcal{A}, \mathcal{B})}{\mathrm{vol}(\mathcal{B})}, \qquad (1)$$

where $\mathrm{cut}(\mathcal{A}, \mathcal{B}) = \sum_{i \in \mathcal{A}, j \in \mathcal{B}} W_{ij}$ measures edge weight between partitions and $\mathrm{vol}(\mathcal{A}) = \sum_{i \in \mathcal{A}, j} W_{ij}$ is the total connection strength from partition $\mathcal{A}$ to all nodes. This optimization relaxes to solving the generalized eigenvalue problem $(\mathbf{D} - \mathbf{W})\mathbf{v} = \lambda \mathbf{D}\mathbf{v}$ for the smallest eigenvectors of the normalized graph Laplacian, where $\mathbf{D}$ is the diagonal degree matrix.

For computational tractability on large feature sets, we employ the Nyström approximation. Rather than computing the full $N \times N$ affinity matrix, we randomly sample $m \ll N$ landmark points and compute only the $m \times m$ landmark-to-landmark affinities and $m \times N$ landmark-to-all-points cross-affinities. The full eigendecomposition is then approximated via this low-rank representation, reducing complexity from $O(N^2)$ to $O(Nm)$. We project the resulting high-dimensional eigenvectors to RGB color space via t-SNE, enabling intuitive visualization where tokens with similar colors belong to the same semantic cluster.

3.2 Debugging Protocol



Row \ Col	1	2	3	4	5	6	7	8
01-08	5 Fingers	4 Extended	5 Fingers	5 Fingers	5 Fingers	5 Fingers	5 Fingers	5 Fingers
09-16	Unknown	16 Fingers	5 Fingers	4 Upwards	10 (2 Hands)	Unknown	5 Fingers	10 Total
17-24	10 Total	12 Visible	10 Fingers	10 Fingers	4 Visible	10 Visible	4 Extended	10 (2 Hands)
25-32	5 Fingers	5 Fingers	5 Fingers	5 Fingers	4 Fingers	5 Fingers	5 Fingers	4 Extended
33-40	5 Fingers	4 Fingers	10 (Cramped)	5 Fingers	5 Fingers	5 Fingers	5 Fingers	5 Fingers

Figure 1: An example of feature debugging with LLaVA. [3]

Our analysis methodology consists of three systematic steps.

First, we curate controlled image batches that isolate specific factors of interest. For example, to test a vision-language model's ability to distinguish finger counts, we collect hands displaying one through five fingers and process them jointly. Then, we inspect whether cluster colors align consistently with the true finger count.

Second, we perform layer-wise inspection by applying NCut to activations from each transformer block. If a perturbation in one sample causes color shifts, with colors remain unchanged across unperturbed samples, the model exhibits high sensitivity to that factor at that layer. Conversely, if colors remain stable despite controlled variations, the model has not learned to discriminate these semantic attributes.

Third,

3.3 Feature Extraction from VLAs

VLA architectures typically consist of a vision encoder (often a pretrained ViT) that processes images into spatial patch tokens, and a language model backbone with interleaved cross-attention layers where text tokens attend to image patches. We extract intermediate representations by registering forward hooks on transformer blocks. For temporal analysis

of manipulation videos, we process frames sequentially and concatenate patch features across time, yielding a 4D tensor that NCut partitions into spatio-temporal clusters. For language-vision alignment studies, we jointly cluster image patches and text tokens from fusion layers, then examine whether semantically related tokens (e.g., the word "can" and the visual region depicting a can) receive similar cluster assignments.

4 Experiments and Results

4.1 Dataset and Experimental Setup

To apply NCut techniques on OpenVLA, a dataset of robot actions and task completions is neccessary. Our criteria for a dataset included:

- Visual data from a real-world environment as opposed to simulations,
- Inclusion of action failure cases as opposed to only successful executions,
- Action diversity to enable exploration of diverse VLA characteristics,
- Repetition of action executions - allowing for fine-grained pertubation selections,
- Labelling of the goals and degree of success of executed tasks.

Unfortunately, in preliminary exploration, we found that most open-source datasets cannot meet all of these requirements, often failing more than one of the criteria. This held true even for Pick and Place actions, make up the majority of robot action datasets.

We decided to select a dataset which met most of these criteria, short of detailed labelling, and instead manually label it [7]. The selected dataset is informally curated by a research team for their specific explorations, and thus contains irregularities and non-uniformities in data collection (with some instances of media errors). However, its diversity makes it valuable for studying edge cases and failure modes that are often filtered from cleaner benchmarks. Given manual labelling, extraction of erroneus datapoints, and descriptions of each of the attempted tasks - this dataset enabled our analysis of a VLA's comprehension of text-image semantics.

For other explorations, we also utilise other datasets or simulations where we seek specific characteristics.

4.2 Text-Image Semantics

We begin by performing batch NCut on various frames of a simple successful case of successful robot action. Specifically in Figure 2, the sequential reach, grasp and lift of a red can. This figure performs NCut on the last layer of OpenVLA, with 20 vision clusters. Where similar colors represent close clustering (semantic alignment within the VLA's language-based latent space), several characteristics of the VLA may be recognised.

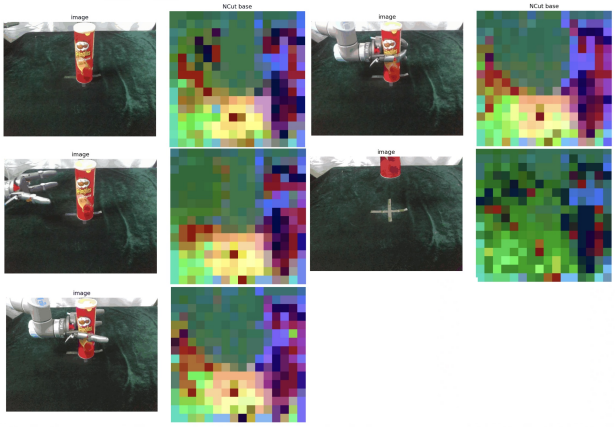


Figure 2: Batch NCut results of successful action.

Firstly, the model shows the ability differentiate between foreground and background, segmenting the robot arm and the red can from the table. Furthermore, the robot arm and the red can both display a similar olive-green hue throughout every frame of the interaction. They retain a green hue throughout frames of interaction, including instances where most of the arm and can are out of frame. These indicate that the vision encoder has learned object-centric rather than position-centric representation. Beyond that, the model associates the objects with one another - an indicator of semantic understaning that they are interacting. Minial hue differences are observed between cases of the robot arm holding the can with an open or closed grip. This may be indicative of semantic distinction between various levels or stages of interaction.

Interestingly, we observe rapid cluster/color reassignments at points of transition between different stages of robot interactions. Senstivity to these is further indicative of semantic understanding of commonplace steps in a task such as Pick and Place.

To assess the degree to which the VLA's semantic comprehension is grounded in language, we jointly cluster vision and language tokens extracted from the fusion layers. Typically, this is used to map the self-

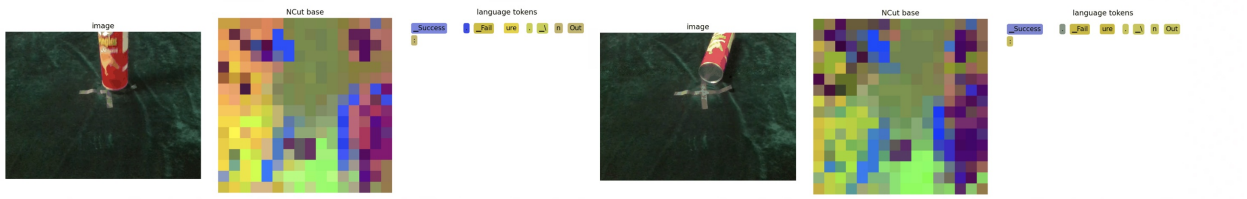


Figure 3: Batch NCut results of frames of failed actions: rapid dropping (left), toppling (right).

attention between an object and its verbal description within a transformer. We aim to see if NCut may be used to discern more complex text-vision associations. Specifically, we seek to determine if the trained VLA has a generalised sense of "sucess" or "failure" amongst robot-object interactions. Figure 3 presents examples of this mapping on select frames of failed intercations with the red can shown before.

OpenVLA showed extreme sensitivity to instances of rapid movement, reorientation, and collisions of the foreground object being interacted with - rsulting in more significant cluster/color reassignments to those discussed previously. These instances correlated closely with typical indicators of a failed robot intercation, such as loss of grip or control. Furthermore, upon frames indicative of failure, the new clusters show close alignment with the word "failure". This correlation also held for interactions with different objects, and with longer text token arrays. Regardless of the success or failure of a robot task, it was also observed that the attention of these text tokens typically focused on the table upon which the Pick and Place actions were taken.

These results suggest that OpenVLA representation semantically comprehends the various features, sequences, and commonalities of robot actions. They also indicate that the VLA's representations implicitly encode cues predictive of task success or failure. These basic observations already present extensive applications of NCut for VLA design. For instance, NCut may be used as triggers for early warning signals of failed or unsafe robot interactions. Such warnings may be applied online by leveraging the efficiency of Nyström NCut.

4.3 Temporal Dynamics

Tracking cluster assignments across manipulation sequences reveals temporal structure. During successful executions, clusters corresponding to the target object remain stable throughout the grasp-lift-transport sequence. In contrast, failed attempts exhibit higher volatility, with frequent cluster reassignments suggesting representational uncertainty. When

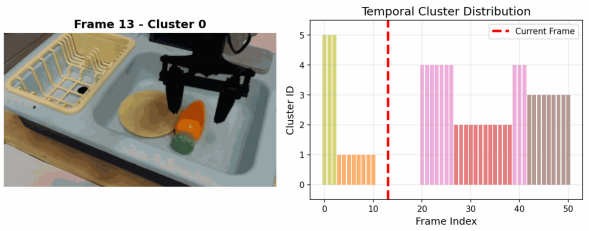


Figure 4: Temporal NCut results of robot Pick and Place of a toy carrot.

we plot the distribution of cluster IDs over time, distinct manipulation phases emerge as transitions in the dominant cluster: approach is characterized by increasing co-occurrence of gripper and object clusters, contact triggers a sharp shift as both regions merge into a unified grasp cluster, and release corresponds to their separation. These temporal signatures could serve as features for downstream tasks like phase detection or progress monitoring.

4.4 Input Sensitivity Analysis

To quantify the relative importance of vision versus language, we conduct a controlled perturbation study. We systematically modify images (changing lighting conditions, object appearance, or camera viewpoint) and text (swapping action verbs like "pick" $\leftrightarrow$ "grasp," or nouns like "can" $\leftrightarrow$ "container"), then measure the induced change in VLA feature representations via cosine distance. Denoting image-induced shifts as Δ_{img}, verb swaps as Δ_{verb}, and noun swaps as Δ_{noun}, we find $\Delta_{\text{img}} > 10 \cdot \Delta_{\text{verb}} > \Delta_{\text{noun}}$. This dramatic imbalance indicates that VLA representations are overwhelmingly dominated by visual information, with language playing a relatively minor modulatory role.

We further probe semantic organization via template-based retrieval. For each semantic category (e.g., the verb "push" or the noun "room"), we collect 30 images from COCO containing that category, average their VLA features to form a template, and compute mean Average Precision (mAP) for re-

trieving other instances. Scene-level nouns achieve relatively high mAP: "room" (0.36), "street" (0.29), "field" (0.27). Object nouns and action verbs score substantially lower: "picture" (0.09), "shirt" (0.07), and among verbs, even high-frequency actions like "fill" (0.07) and "show" (0.05) cluster poorly. This suggests VLAs form coherent semantic clusters primarily for static scenes and objects, while action semantics remain weakly differentiated.

4.5 Spatial Intelligence Evaluation

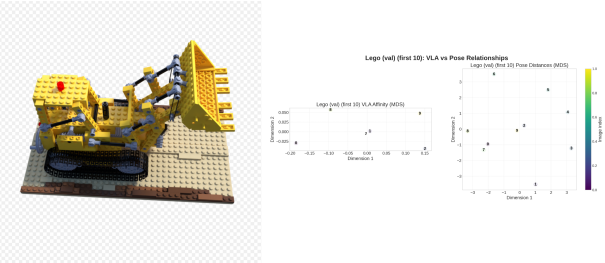


Figure 5: The difference between the VLA affinity and the true physicial distance. We do this experiment on NeRF datasets.

A critical question is whether VLAs learn geometric understanding or merely memorize appearance-pose associations. To test this, we extract VLA features from NeRF datasets where ground-truth camera poses are known, and train a 3-layer MLP to regress 6-DOF poses (position and orientation) from these features. On the training set, the probe achieves nearly perfect performance with loss 0.003, demonstrating that pose information is indeed encoded. However, on held-out test views, the loss remains at 0.998—complete failure to generalize. We also perform multi-dimensional scaling (MDS) on pairwise feature distances and compare to ground-truth pose distances. For nearby viewpoints in 3D space, we find no corresponding proximity in feature space, confirming that VLAs do not learn geometric relationships between views.

This finding has significant implications. VLAs can memorize which visual appearances correspond to which spatial configurations in their training data, but they lack the geometric inductive biases needed to reason about unseen viewpoints, occlusions, or multi-step assembly tasks requiring spatial planning. Future architectures may benefit from incorporating explicit 3D representations or contrastive losses that enforce view-invariant embeddings.

4.6 Gradient Flow and Attribution

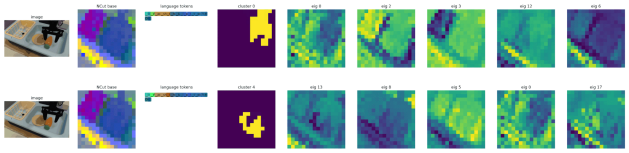


Figure 6: The visualization results of our channel tracing methods. We can see that we can find certain channels that can contribute to certain clusters obtained from NCut.

To understand which features drive action predictions, we compute gradients of the action loss $\mathcal{L}$ with respect to individual token features: $\partial\mathcal{L}/\partial\mathbf{f}_i$. Gradient magnitudes concentrate heavily on object and gripper clusters, with background regions exhibiting near-zero attribution. This confirms that action prediction relies primarily on task-relevant semantic regions, ignoring irrelevant context.

We also examine which NCut eigenvectors contribute most to action prediction by analyzing the correlation between eigenvector coefficients and action output variations. Lower-frequency eigenvectors, corresponding to coarse semantic partitions, show strong correlations with action changes. Higher-frequency eigenvectors capturing fine texture details have minimal impact. This suggests that VLA action heads primarily operate on coarse object-level segmentation rather than fine-grained visual features, which may explain robustness to appearance variations but also fragility to subtle geometric misalignments.

4.7 Simulation Validation

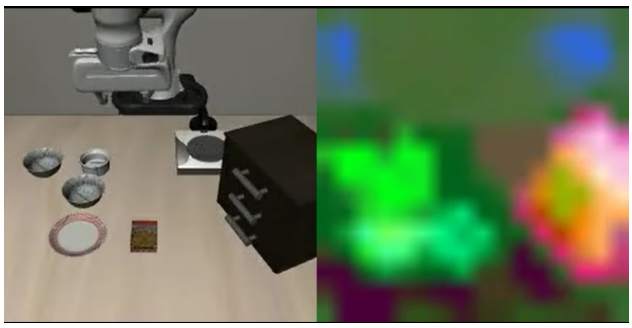


Figure 7: The visualization restuls when actually executing the policy in the LIBERO simulations.

We deploy OpenVLA in a simulated tabletop environment (using PyBullet) with randomized objects

(cylinders, cubes, rings) and natural language pick-and-place instructions. The baseline model achieves 71% success rate over 50 trials. By applying NCut to failure cases, we identify two recurring patterns: fragmented object clusters indicating poor segmentation, and premature overlap between gripper and obstacle clusters suggesting collision risk. We change the environment to make the objects have clearer edges and it raises the success rate to 80%, validating that NCut-guided diagnosis translates to actionable performance improvements.

5 NCut PyTorch Documentation

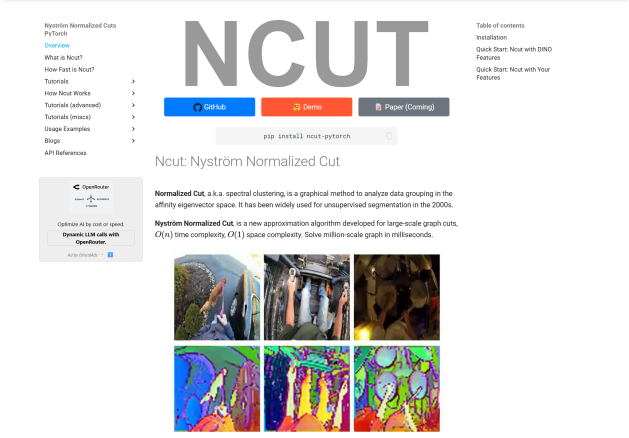


Figure 8: The new website for NCut.

As part of this project, we developed and currently maintain the official NCut PyTorch documentation website [3]. The site provides comprehensive tutorials covering basic NCut operations, advanced debugging workflows, and API references for all modules and functions. It also includes a model gallery demonstrating applications to vision transformers, language models, diffusion models, and multimodal systems. Built with Sphinx and hosted on ReadTheDocs with continuous integration, the documentation auto-generates from code docstrings and Jupyter notebooks, ensuring consistency across software releases. The resource has been adopted by researchers beyond robotics for interpreting foundation models and serves as an educational tool in graduate courses on interpretable machine learning.

6 Discussion and Future Work

Our exploration shows that NCut offers insight into the black box architectures of VLAs. While much of the decision-making process may remain obscured, this tool has enabled us to better understand and characterize a VLA model - from low-level insights on token attention, to higher-level concepts of spatial/geometric compresension and semantic understanding of text tokens related to task instructions.

Our findings highlight both the strengths and fundamental limitations of current VLA architectures. The strong object-centric clustering and implicit failure awareness demonstrate that these models learn useful mid-level semantic representations. However, the 10× dominance of visual features over language raises critical questions about instruction following: do VLAs genuinely ground fine-grained linguistic distinctions, or do they primarily rely on visual priors with language serving as a weak contextual signal? The spatial intelligence failure is particularly concerning for real-world deployment, as tasks like insertion, precise placement, and occlusion reasoning all require geometric understanding beyond appearance matching.

We believe that NCut may offer many avenues for continued exploration of VLAs. First, we plan to extend this analysis to OpenPi 0.5 [5], which incorporates architectural improvements to OpenVLA and larger-scale training, to determine how the capabilities and limitations of VLAs may vary across architectures and training appraoches. Second, we aim to study whether temporal NCut patterns align with instruction structure in multi-step tasks. For example, does the phrase “first pick up the red block” trigger a cluster transition before “then place it on the blue block”? Third, integrating NCut-based early warning signals with failure recovery systems [1, 2] could enable online detection and correction.

Additional furute work may also look to tracing failures across architectural submodules. For VLAs with separate vision encoders and vision-language fusion layers, NCut may allow for analysis of component independently to localize whether failures originate in visual feature extraction, language encoding, or cross-modal alignment. It may also aid as an analysis tool in development of new or improved architectural submodules. For instance, it may help characterise geometric inductive biases or auxiliary losses that enforce 3D-consistent representations - which may address the spatial generalization gap.

7 Conclusion

Nyström Normalized Cuts provides a practical and scalable tool for debugging robot foundation models. By visualizing feature geometry across layers and modalities, we can uncover semantic structure, diagnose failure modes, and identify architectural weaknesses that standard metrics obscure. Our analysis of OpenVLA reveals that while these models excel at object-centric perceptual reasoning, they struggle with language grounding and spatial generalization. NCut may be applied in various additional ways to those explored, and as VLAs scale toward real-world deployment, interpretability methods like NCut may prove crucial for ensuring reliability, safety, and targeted improvement.

References

[1] Yinpei Dai, Jayjun Lee, Nima Fazeli, and Joyce Chai. RACER: Rich language-guided failure recovery policies for imitation learning. *arXiv preprint arXiv:2409.14674*, 2024.

[2] Jiafei Duan, Wentao Pumacay, Nishanth Kumar, Yi Ru Wang, Sichun Tian, Wenhao Yuan, Ranjay Krishna, Dieter Fox, Ajay Mandlekar, and Yuke Guo. AHA: A vision-language-model for detecting and reasoning over failures in robotic manipulation. *arXiv preprint arXiv:2410.00371*, 2024.

[3] Ningze Zhong Huzheng Yang. Nyström normalized cuts pytorch documentation. https://ncut-pytorch.readthedocs.io/en/latest/, 2025. Accessed: December 2025.

[4] Moo Jin Kim et al. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[5] Physical Intelligence et al. pi0.5: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[6] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[7] Tao Wang, Changliang Yang, Frank Kirchner, Peng Du, Fuchun Sun, and Bin Fang. Multimodal grasp data set: A novel visual-tactile data set for robotic manipulation. *International Journal of Advanced Robotic Systems*, 2019.